Production scheduling at MPS-level under nonlinear-capacity constraints

Master thesis IEM at the University of Twente

Date:

29-04-2021

UNIVERSITY OF TWENTE.

Author:

Joey Klein Koerkamp

Supervisors University of Twente

Marco Schutten, Gréanne Leeftink

Supervisor SINTEF Digital:

Hans de Man

Management summary

Introduction and research objective

Brynild Gruppen AS is one of Norway's largest confectionery manufactures. Brynild is located in Fredrikstad, and mainly sells its products in the Scandinavian countries. In this research we focus on the planning of the confectionery production at Brynild at the Master Production Schedule (MPS)-level. Presently, the MPS is created manually. This is a time consuming endeavour, which leaves limited time for optimization. Although the current method suffices for now, an increase in demand requires a more efficient planning in the near future. At the same time the costs of production need to be kept within bounds, explaining the need for a cost-efficient MPS.

Our objective is to construct a method for creating a cost-efficient MPS for one of Brynild's production lines, the Støperi Sukker line. This method takes the form of an algorithm. The benefits of using an algorithmic method over a manual approach are twofold. First, an algorithmic method, run on a computer, can process much more information than a human. This allows it to consider a larger number of possible improvements to make a more cost-efficient MPS. Second, an algorithmic method can create an MPS much faster than a human. We show that our method creates an MPS within 30 seconds, compared to multiple hours by hand. This leaves the planner with much more time to finetune the schedule provided by our method.

Problem solving approach

Many variants of the problem of creating a cost-efficient MPS are studied in literature. We can classify the Støperi Sukker line as a process with multiple capacitated resources, since we need to take into account the capacity of both the moulding machine and the drying cabinets. In our literature review we encountered a heuristic proposed by Boctor & Poulin (2005). Their research shows that this heuristic performs well for planning problems with multiple capacitated resources. We therefore build our own method based on this heuristic.

The biggest hurdle we needed to tackle when building our candidate planning method was how to deal with the capacity constraints of the drying cabinets in which the confectionery is dried. We came up with an efficient method, based on a partition algorithm, to check whether a certain schedule is feasible with respect to these drying cabinets. This method essentially tries to distribute the production orders over the different drying cabinets without violating the capacity constraints on any of them.

This partition algorithm enabled us to do the following: We used Boctor & Poulin's heuristic in combination with a linear approximation of the capacity constraints of the drying cabinets to create an MPS that is approximately feasible. We then used our partition algorithm to locate the infeasibilities and solved them through small tweaks to the schedule, thus obtaining a feasible MPS.

Solution validation

We use our proposed method to create 32 schedules from historical input data and compared those with the current method historical production numbers. We found similar behaviour between the schedules created by both methods with respect to the number of products per week, the average lot size and the average workload per week.

Discussing the schedules created by both methods with the human planner provided the insight that the candidate method works ahead only the minimum amount to meet demand, to save holding costs.

In contradiction, the human planner tends to work ahead more in order to prepare for the uncertainty of the future.

The real world instances of the problem are too large to find near-optimal solutions against which we can benchmark our candidate method. We measured the performance through smaller instances of 5 products and 5 weeks (as opposed to the 41 products and 26 weeks of a real world case). We used our method to create approximately feasibly schedules of those instances and compared these schedules with those of an MIP-solver. We found that our candidate method approaches the best solutions, with on average 4% higher costs than the best found solution. We expect this performance to translate well to larger cases, since Boctor & Poulin’s heuristic, on which our method is based, is known to achieve better result when the problem size increases.

Recommendation

Our proposed method provides feasible, cost-efficient MPSs. These MPSs require only minor tweaking by the planner before they can be implemented. We want to emphasize that the our method should be viewed as a support tool for the human planner and not as a replacement. The schedules created by our method need some finetuning to comply with certain aspects of reality that are not captured in our model. The most prominent of these aspects are the availability of operators. This is especially true for the decision in which weeks to employ night shifts.

Preface

They say a thesis is an individual project, but there are many people who helped me along the way and who deserve to be mentioned here:

I want to thank my supervisors from Twente, Marco Schutten and Gréanne Leeftink for their feedback and guidance in the writing of this thesis.

I want to thank the people at Brynild, especially Mathias Holm and Eirik Blå, for sharing their data, insights and enthusiasm towards this project and for taking the time to answer a thousand or so questions.

I want to thank Hans de Man for his feedback and advice, not only regarding my thesis, but also on surviving in the harsh Norwegian weather and life in general. To share some of his wisdom with the reader: "When waiting out a thunderstorm, sit with your feet and butt on the ground so that when lighting strikes, it passes through your legs instead of your vital organs."

I want to thank Charlotte for the countless discussions regarding our theses and for putting up with me playing Christmas music in our office for six weeks straight.

This thesis marks the end of my time as a student at the University of Twente. I met so many great people through my studies. I want to thank Annabel, Edwin, Fabian, Veerle, Yoran and all the others I did not mention here for making studying in Twente an amazing experience.

List of abbreviations

BOM	Bill Of Materials
COGS	Costs Of Good Sold
ERP	Enterprise Resource Planning
GA	Genetic Algorithm
LP	Linear Programming
MIP	Mixed Integer Programming
MPS	Master Production Schedule
MRP	Material Resource Planning
MTS	Make To Stock
SA	Simulated Annealing
TS	Tabu Search
WIP	Work In Progress

Table of contents

1. Introduction

This thesis researches the long term planning of sugar confectionary at Brynild. This chapter introduces the research. Section 1.1 introduces the parties involved in this project. We describe the research motivation and problem identification in Sections 1.2 and 1.3 respectively. We present the research objective and the research questions in Section 1.4 and 1.5 respectively.

1.1 Parties involved

The two parties involved in this project are SINTEF and Brynild.

SINTEF

SINTEF is a Norwegian research organization headquartered in Trondheim. SINTEF is currently working on a project for Brynild Gruppen AS, a Norwegian confectionery manufacturer. The goal of the project is to improve planning & scheduling of Brynild's sugar confectionary production process. Within this project, we conduct this research.

Brynild Gruppen A.S.

Brynild Gruppen is a Norwegian company that produces and sells branded consumer goods like sweets, chocolate, nuts and dried fruits. The products are sold in the four Scandinavian countries (Norway, Sweden, Finland and Denmark). The company operates one production facility, located in Fredrikstad, Norway.

Within this production facility, they produce confectionery, chocolate and nuts. There is some interaction between the production processes of chocolate and nuts, since some products contain chocolate covered nuts. The production of confectionery is completely separated from the production processes of chocolate and nuts to ensure that the nuts do not contaminate the confectionery. There is an exception for some confectionery items as they that are covered in chocolate as well. These items are moved to the chocolate section of the facility after being moulded and further processed there. In this research, we only consider the confectionery production.

Brynild sells its confectionery products to wholesalers, who in turn supply one or multiple chains of grocery stores. The wholesalers demand short lead times. They expect orders to be sent to them within two or three days after placing them. These lead times are in fact shorter than the production time of the confectionery, so Brynild needs to fulfil its orders from its finished goods warehouse. Therefore Brynild produces most of its confectionery in a make-to-stock (MTS) fashion.

1.2 Research motivation

Brynild sells its confectionery products mainly on the Norwegian market. They experience a lot of competition from larger, international companies on this market. To stay competitive, Brynild has to ensure it can meet the demands set by the wholesalers they supply. The wholesalers demand 98% of their orders to be delivered on time and in full. To meet this requirements, Brynild requires accurate demand forecasts and cost-efficient planning.

Brynild is currently focusing on improving its planning processes. The main goal is to reduce overcapacity and secure shorter lead times. Planning for confectionery production at Brynild is mostly done by experience based techniques. This task is complex because they produce more than 30 different candies and therefore many possible schedules exist. The combination of experience based methods and the complexity of the task makes them believe that this approach does currently not result in the most cost-efficient planning.

The workload at the confectionery production process is variable. The current production capacity is not sufficient for certain peaks in expected workload. Some extra capacity is available, but this comes at a cost. It also makes planning ahead difficult. In turn, this leads to problems when creating the work rosters because the operators need to be informed when they have to work at least two weeks in advance. In addition, there is a stage in the production process where no extra capacity is available. This is the drying of the confectionery products. During peaks in workload, the drying stage limits the total production capacity, which threatens the fulfilment of demand. Therefore the motivation of this research is to find a way to lower the variation in workload so that demand can be fulfilled and so that this happens in a cost-effective manner.

1.3 Problem identification

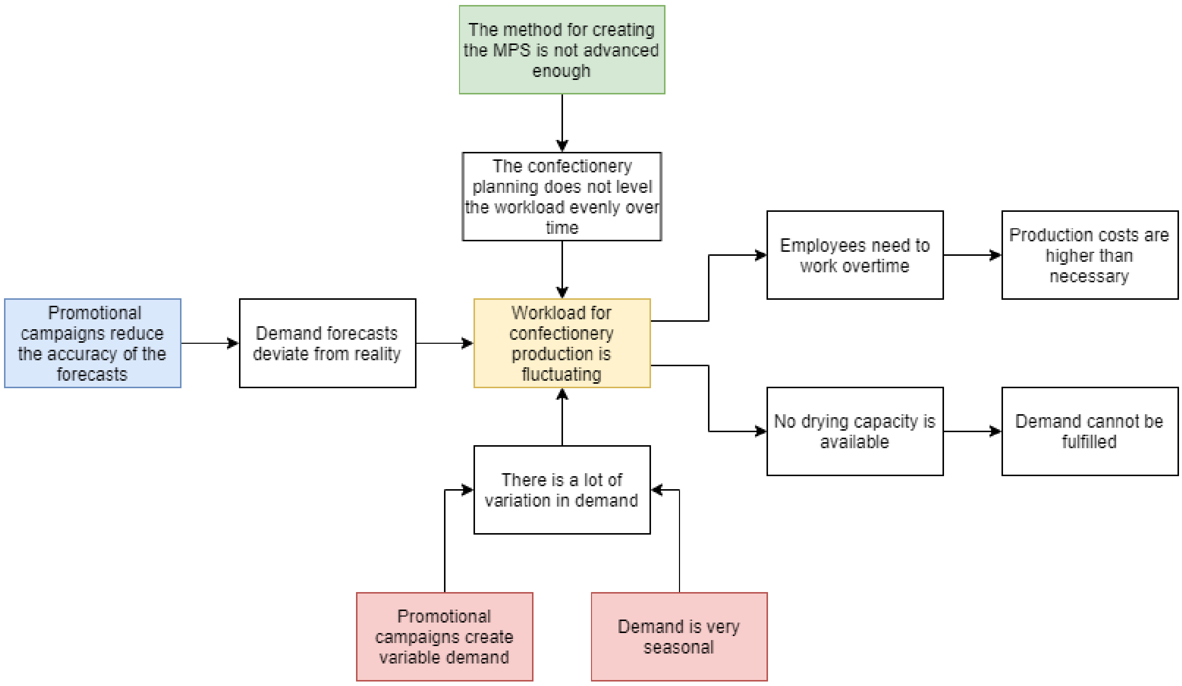

Figure 1.1 Problem cluster.

There are many problems related to the fluctuating workload. Figure 1.1 visualizes the relationship between the different problems in a problem cluster. Since not all problems can be dealt with at the same time, we select one problem to be solved for this research. We select this problem through the method of Heerkens & Van Winden (2012). The selected problem is called the core problem. A core problem is not caused by another problem and must be influenceable.

Figure 1.1 shows that the fluctuating workload (in yellow) is the cause of other problems. Employees need to work overtime to fulfil demand during peaks in the workload. This overtime comes with increased salary costs, which in turn increase the production costs. Another consequence is that sometimes, the fixed capacity of the drying of the confectionery is reached. This limits the entire production, causing demand to remain unfulfilled.

The fluctuating workload is caused by three other problems. We explore each of these causes further, to identify the core problem.

The first cause (below yellow) is that there is a lot of variation in demand for the products. This variation is caused by promotional campaigns, which temporarily increase demand, and the seasonal nature of demand for confectionery products. Demand for confectionery products peaks around

Easter, Halloween and Christmas and to a lesser degree during the summer. The seasonality is hard to influence. The temporal increase in demand from promotional campaigns can be influenced, but only by turning down campaigns proposed by the wholesalers. Turning down campaigns is not recommended because these campaigns have a positive effect on sales. This excludes both problems (in red), from being the core problem. Therefore, the variation in demand is something that needs to be dealt with, rather than something that needs to be solved.

The second cause (left of yellow) is that the demand forecasts deviate from the actual demand. Since Brynild produces its confectionery on a MTS basis, forecasts of demand are highly important for capacity planning. Deviation between the forecasts and realization of demand can disrupt the planning, leading to an fluctuating workload. The forecasts for regular and seasonal demand are accurate, but those for promotional campaigns have a relative high error. This problem (in blue) would be a good candidate for the core problem, but another research project is already focussing on this. Therefore, we do not choose it as the core problem.

The third cause (above yellow) is the performance of the planning method. At Brynild, production planning more than one week into the future is done through the use of a master-production-schedule (MPS). The MPS specifies how much of each product needs to be produced in each week, taking into account demand forecasts for the next 28 weeks. The MPS does not create exact schedules. These are created afterwards and only for the upcoming week, as the production quantities of later weeks might change later. The function of the MPS is to level the workload between different weeks. This reduces peaks in workload so that demand can be met in a cost-effective manner.

The MPS is created by the planner. The planner uses his knowledge of the production process to estimate whether a set of production quantities can fit together in a certain week. Hereby he aims to fit as much production volume into the upcoming one or two weeks to maximize capacity utilization and leave capacity in the weeks that are further ahead for future orders. This is done by manually moving orders between weeks in a spreadsheet program. Creating the MPS manually has two big disadvantages. First, it takes a lot of time to create a schedule since all moves must be thought of and executed manually. Second, the number of options that can be considered by a human is very limited so cost-efficient improvements to the schedule are overlooked. Therefore the created MPS is suboptimal at best.

The lack of an advanced planning method is the cause of the MPS not fulfilling its potential with respect to levelling the workload. This problem (in green) has no other causes and can be influenced. Therefore we identify it as the core problem.

1.4 Research objective and scope

The core problem is that the method for creating the MPS is not advanced enough. The current method of manually moving orders does not consider and evaluate many different alternative solutions. Based on this problem, the main research objective is defined:

To construct a method for creating a cost-efficient MPS for Brynild's confectionery production process.

Through the research objective we determine the scope of this research. We focus on improving the MPS with MPS-like planning methods. We conduct no research into improving other areas of the planning process like forecasting and demand management.

1.5 Research questions and approach

To achieve the research objective, we formulate the following four research questions:

1) What is the current situation regarding the planning of the confectionery production process at Brynild?
    a. What does each stage of the confectionery production process look like?
    b. Which stages of the confectionery production process impose which constraints regarding planning?
    c. What determines the performance of an MPS?
    d. What is the performance of the current planning process?

The answer to this research question provides insights in the planning problem at Brynild. This information is required to create a solution that is tailored for Brynild's needs and can be implemented within the existing planning process. The research question is answered through supplied information and analysis of data provided by Brynild. This research question is discussed in Chapter 2.

2) What methods for the creation and optimization of an MPS are there in literature that are applicable to Brynild's confectionery production?
    a. How can the problem of creating an MPS be defined and how is this problem studied in literature?
    b. What methods for the creation and optimization of an MPS are there in literature?
    c. What are the (dis)advantages of the different methods?

This research question provides an overview of the ideas and solutions that exist in literature. This information is used as a basis to design a solution for the research objective. The research question is answered through a literature review. The results are discussed in Chapter 3.

3) What method or methods are promising candidate solutions for achieving the research objective?
    a. How can the task of creating an MPS for Brynild's confectionery process be formulated as a model?
    b. Which of the methods found in literature is best suited for solving the formulated model?
    c. How can these methods be tailored or combined into promising candidate solutions for Brynild?

This research question provides the creation of candidate solutions. To create solutions for the general task of creating an MPS for Brynild's confectionery process, we need an exact formulation of this task as some type of model. The sub-questions, are answered by combining the information obtained in Chapters 2 and 3. The candidate solutions are described in Chapter 4.

4) How do the candidate solutions perform compared to the current method used at Brynild?
    a. How to evaluate the candidate solutions and the current method?
    b. How do the performances of the candidate solutions and the current method compare to each other?
    c. How does the performance of the candidate solutions compare to the optimal solution?

This last research question is concerned with evaluating the performance of the candidate solutions and the current method. We evaluate these solutions based on their scores on predefined performance indicators. This process is described in Chapter 5.

Chapter 6 describes the main conclusions of this research and the recommendations to Brynild.

2. Current situation

In this chapter, research question 1 is answered: *What is the current situation regarding the planning of the confectionery production process at Brynild?* Section 2.1 describes each stage of the confectionery production process and the constraints each step puts on planning. Section 2.2 describes the current planning process. This chapter concludes with the performance of the current planning process in Section 2.3.

2.1 Confectionery production process

In this section, a detailed description of the confectionery production process at Brynild is provided. Brynild currently produces 38 different candies, which are referred to as intermediates. Each confectionery product that Brynild sells consists of one or more intermediates combined into a packaging type. Some intermediates are only sold solo (one kind of intermediate in a packaging type), while others are only sold in a mix with others. Lastly, there are also intermediates that are both sold solo and in a mix.

All intermediates are produced by the same production line called Støperi Sukker. Some intermediates are afterwards coated on the coating line called Drage Sukker. Finally, all intermediates are packed by one of the four packaging lines. Between the lines, intermediates are referred to as work-in-progress (WIP) and are stored at designated parts of the shop floor. The storage space for WIP is limited to 200 storage pallets. For comparison, the average daily production is 20 storage pallets. Figure 2.1 gives a schematic overview of the production process. Each production step is described in detail in the remainder of this section. Hereby the focus is on the considerations each step induces on the planning process.

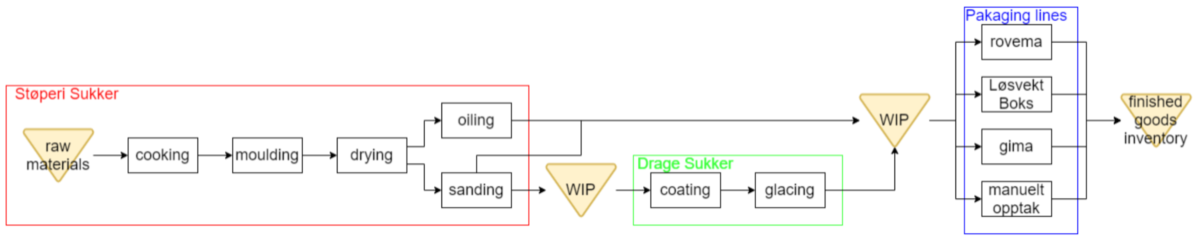

Figure 2.1 Confectionery production process at Brynild.

2.1.1 Cooking & moulding

The process begins by cooking the raw materials together. While cooking, colour and taste are also added. Once ready, the cooked mass is transported to the moulding machine through pipes. There, it is sprayed into trays that contain the shape of the produced intermediate. Depending on the intermediate, each tray contains 96 to 800 pieces. The trays, now containing moulded intermediates, are stacked on pallets that contain 150 trays each. These pallets are then automatically transported to the drying stage. The cooking, moulding and transporting of pallets to the drying section, all happen at the same time in a continuous fashion.

When switching from producing one intermediate to another, the process needs to be stopped so that the settings of some machines can be changed and the pipes can be cleaned. The duration of these so called changeovers depends on both the previous and the subsequent intermediate, i.e., they are sequence dependent. The changeover times vary between 60 and 120 minutes. In addition to the loss of production time, there is also a loss of product material at each changeover. This is due to material that stays behind in the pipe from the cooking machine to the moulding machine. This loss is around

50 kg of material per changeover. Cooking and moulding is done in shifts, from Monday to Friday and sometimes on Saturdays. The number of 8-hour shifts per day varies between one and three.

From a planning perspective, there are two main things to consider regarding the cooking and moulding. The first consideration is the sequence in which intermediates are produced. The loss of capacity due to changeovers and the loss of material are incentives to minimize the number of required changeovers.

The second consideration is the number of shifts that is needed in a certain week. The cooking and moulding is done on a 2-shift or 3-shift basis, depending on how busy it is. In a 2-shift system, there are 9 available shifts in a week (5 morning shifts on Mondays to Fridays and 4 afternoon shifts on Mondays to Thursdays). In a 3-shift system, there are 14 available shifts (5 additional night shifts are added Sunday to Thursday nights). Each week the cooking and moulding is done on either a 2-shift or 3-shift basis. It is not possible to switch during the week. The decision to run a 2-shift or 3-shift system in a certain week must be made at least two weeks in advance to comply with labour regulations.

It is possible to plan an extra shift on Friday afternoon and on Saturday morning, but these come with increased salary costs. Table 2.1 shows the different shifts systems with the 2-shift system (in blue), the additional shifts of the 3-shift system (in green) and the possible extra shifts (in orange).

	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
Morning 06:30-14:30							
Evening 14:30-22:30							
Night 22:30-06:30							

Table 2.1 The different shift systems used by Brynild.

2.1.2 Drying

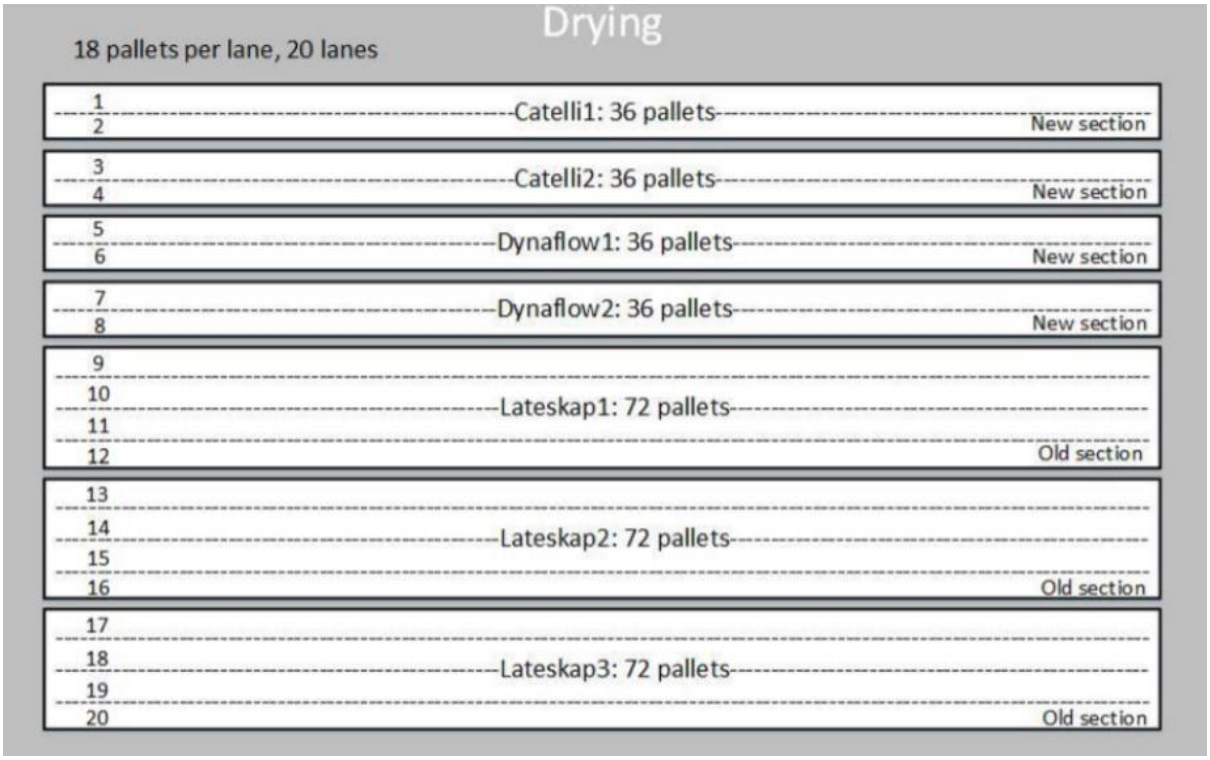

Figure 2.2 The 5 drying cabinets of Brynild.

After moulding, the intermediates need to be dried to reduce their moisture content. Drying happens in temperature controlled cabinets, of which Brynild has 5. Figure 2.2 shows that there are two new cabinets and three old cabinets, each of which has a capacity of 72 pallets. The new cabinets are heated and the air within is circulated, while the old cabinets are only heated. Once all pallets of a batch have been automatically transported from the moulding machine to the allocated drying cabinet, the cabinet closes and the drying begins.

The drying of the intermediates is the most time consuming step in the confectionery production process. The drying takes between one and four days depending on the intermediate. Due to the air circulation most intermediates need a shorter time to dry in the new cabinets than they would in the old ones. Additionally, there are some intermediates that have stricter drying needs and can therefore only be dried in the new cabinets. Lastly, there are intermediates that need to dry at room temperature. For these intermediates, no time can be saved by drying them in the newer sections. They do however still need to be dried in a drying cabinet, because no other space is available for this.

Once the drying is finished, the cabinets are opened and the trays are automatically transported back to the moulding machine. There the moulding trays are automatically emptied and the trays are directly fed back to the moulding machine. Each moulding tray is therefore always either in a drying cabinet, at the moulding machine or travelling to one of these locations.

The drying stage creates two challenges for the planning of the whole process. There should be suitable drying space available for intermediates once they finish the cooking and moulding. If this cannot be achieved, production needs to be stopped until drying space becomes available, resulting in a loss of productivity. Additionally, the moulding of intermediates can only happen while a drying cabinet is being emptied. This is because there are no spare trays available and otherwise no trays are available at the moulding machine.

The constraints of the drying stage must be taken into account when sequencing the production of intermediates at the moulding stage. This is to ensure the availability of moulding trays and suitable drying cabinets. On the level of MPS planning this means that besides the capacity constraints of the moulding machine the capacity of the drying cabinets must be taken into account. The planner must ensure that the capacity constraints of the drying cabinets are respected.

2.1.3 Oiling and sanding

After being separated from the trays. The intermediates are automatically conveyed to either the oiling drum or the sanding drum. All intermediates are either oiled or sanded depending on the type of intermediate. Both the oiling and the sanding are completely automatic steps and take a negligible amount of time. After this step, the intermediates are stored into boxes. All oiled intermediates are now ready for packaging. However, some of the sanded intermediates also need a coating. They are sent to the coating line.

2.1.4 Coating and glancing

The intermediates that need a coating are further processed on Drage Sukker, the coating line. The process consists of two steps, coating and glancing. First the intermediates are put into a coating drum where colour and flavour are added. There are two types of coating: sugar and sugar-free. Once this is finished the intermediates are put in a glancing drum where some wax is added to polish their surface. The coated products are afterwards ready for packaging.

The Drage Sukker line can be run with a 2-shift or a 3-shift system similar to Støperi Sukker (see Table 2.1). During the day shifts there are 2 operators, while during the night shift there is only one operator. The coating involves a lot of manual labour and therefore the available capacity of a night shift is only half the capacity of a day shift. Each operator can process 1400 kg of sugar coated intermediates or 700 kg of sugar free coated products per shift. Setup costs and changeover times are negligible for this stage.

The storage space for WIP between the Støperi Sukker and Drage Sukker lines is limited. This has been a problem in the past; therefore the planner must ensure that WIP between these lines stays under the maximum capacity of the storage space.

2.1.5 Packaging

All intermediates need to be packed to become finished products. The type of packaging differs between products. As mentioned earlier, some types of intermediate need to be packed into multiple products. Brynild has multiple packaging lines, each of which is dedicated to a different type of packaging. Each product can be packed only at a specific packaging line. The intermediates are packed into consumer packages, which are in turn packed into distribution packages. These are then stacked on pallets and transported to the finished goods warehouse.

Before packaging, the intermediates are stored as WIP on the shop floor. The storage space is limited and so there is a restriction on the amount of intermediates that can be waiting to be packed.

2.2 Current production planning & scheduling process

In this research, the focus is on the method of creating the MPS for Brynild's confectionery process. However, to fully understand the MPS, its limitations and potential, it is necessary to understand how it interacts with the other components of the planning & scheduling process. In this section, the entire planning process that is currently used for confectionery production at Brynild is described. At the end of this section, a visualization of the process is given in Figure 2.5.

2.2.1 Types of product demand and product demand forecasting

The confectionery production of Brynild is mostly MTS, so the quantities that need to be produced are based on forecasts. Therefore demand forecasting is paramount for planning. Four different types of product demand are distinguished. We now describe each type of demand and the method used for forecasting this demand:

1) **Regular demand**
   Regular demand consists of the regular sales of products. These products are produced in an MTS fashion. Brynild's enterprise-resource-planning (ERP) provides demand forecasts for six months into the future.
2) **New product introductions**
   For new products, demand is forecasted based on experience. Before the product is sold in stores, the supply chain needs to be saturated with the new product. Achieving this takes approximately 20% of the expected annual demand.
3) **Campaign demand**
   Promotional campaigns temporarily increase sales of a certain product at a specific retail chain. To anticipate this, an agreement is made with the retail chain 5 to 10 weeks in advance.
4) **Seasonal demand**
   Seasonal products are products made for a specific season or holiday. These products have a high demand in the couple of weeks and cannot be sold afterwards. Brynild makes these products for Christmas, Easter, Halloween and the summer season. In addition to seasonal products, certain regular products also see a sharp increase in demand during these holidays. Figure 2.3 shows the weekly sales of such a product (Supermix) from the start of 2012 until week 9 of 2014. Supermix is a regular product, which sees sharp increases in demand during

Easter (påske) and Christmas (jul). The demand for seasonal product and increased demand for regular products are called seasonal demand. Seasonal demand must be produced well in advance because the demand is many times higher than regular demand and production capacity. Volumes for seasonal demand are determined four months in advance. The forecasts are based on historical numbers and regular demand.

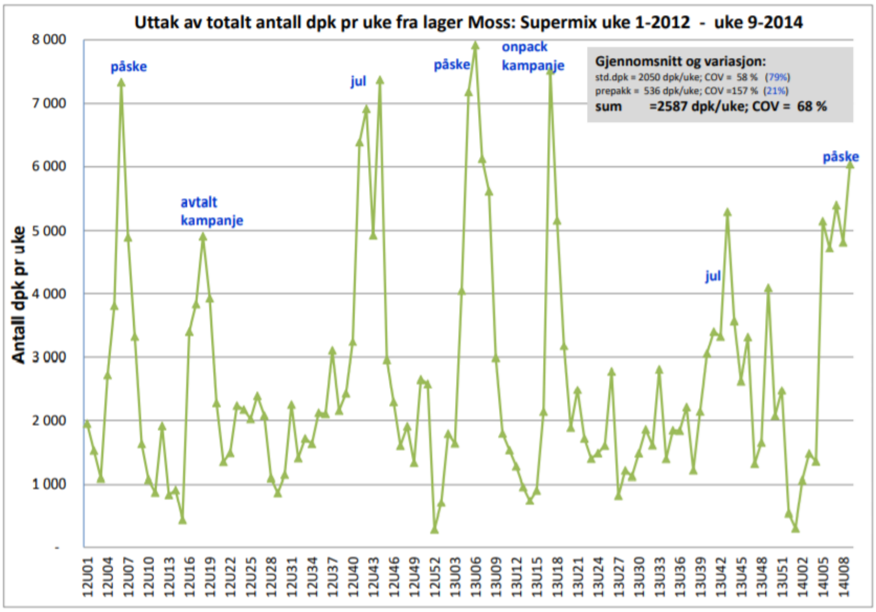

Figure 2.3 Demand for Supermix, a regular product, from the start of 2012 till the beginning of 2014. Figure provided by Brynild.

2.2.2 From product demand to production demand

This section describes the process of converting the different types of product demand to the demand for the different intermediates.

The ERP only creates forecasts for regular product demand. These numbers are pulled out of the ERP system. The new product, campaign and seasonal demand forecasts are then added to create the total weekly demand for all products. From this point in the process, the different types of demand are no longer distinguished.

The forecast for the total weekly product demand spans six months. However, it is important to note that this forecast is not complete. As previously described, campaign demand is only known five to ten weeks ahead and seasonal demand is decided upon four months in advance. Demand that lies further into the future is therefore systematically forecasted too low.

The total weekly product demand, the so called external demand, is then fed back into the ERP system. The ERP combines the weekly product demand with the bill-of-materials (BOM) to determine the demand for each product and intermediate at respectively the packaging and the production line. The demand for the intermediates that is generated by the external demand is the so called internal demand. To do this the ERP uses a systematic procedure that assumes a fixed lead time of one week between each of the production and packaging lines. Other than assuming a fixed lead time, the ERP system uses no planning logic and neither does it consider any capacity constraints.

The procedure that the ERP uses consists of the following steps:

1) For each week t, the product demand in that week is set as demand for that product at the corresponding packaging line.
2) The uncoated intermediates that are part of these demanded products are set as demand at the Støperi Sukker line in week t-1.
3) The coated intermediates that are part of these demanded products are set as demand at the Drage Sukker line in week t-1.
4) The demanded coated products at the Drage Sukker line are set as demand at the Støperi Sukker line in week t-2.

The output of this procedure is called the material-resource-planning (MRP) data and is a database where each entry specifies a product or intermediate, a due date, an order size in kilogram and a production line. An entry in this database is referred to as a production order.

Figure 2.4 shows the production orders generated for one product in one week (the actual data consists of all products for six months). The blue production order shows a demand of 6300 kilogram for a product called “Jelly Beans XXL” in week 14 of 2021. This product is packed on the Løsvekt Boks packaging line, so this is where the demand is located. Jelly beans are a coated product, therefore this induces another production order (in green) of 6300 kilogram for the corresponding coated intermediate at the Drage Sukker line in week 13. In turn, this induces a production order (in red) of 4900 kilogram of the corresponding uncoated intermediate at the Støperi Sukker line in week 12. The demand at the Støperi Sukker is lower because the coating adds weight to the intermediates.

material	material description	due date	order size (kg)	line
107270	Korpus Frutti beans xxl	2021-12	4900	Støperi Sukker
113540	Drage Frutti Beans XXL	2021-13	6300	Drage Sukker
107246	Brynild LV Jelly Beans XXL 2,70kg	2021-14	6300	Løsvekt Boks

Figure 2.4 An example of the input data for the MPS, with production orders for Støperi Sukker in red, for Drage Sukker in Green and production orders for the packaging lines in blue.

The output data as shown in Figure 2.4 is essentially a so called lot-for-lot schedule. A lot-for-lot schedule plans the production of an intermediate or product in the week that it is due, while ignoring capacity constraints. Typically, such a schedule is almost always infeasible. The planner uses this lot-for-lot schedule as the main input for the creation of the MPS.

2.2.3 Creating the MPS

The MPS dictates the amount of each intermediate that needs to be produced and the amount of each product that needs to be packed in each week. The MPS is made manually by the planner, who uses his experience and knowledge of the production process to create a feasible planning. The MPS is updated at irregular intervals varying from multiple times a week, to once a month. The decision when to update the planning is made by the planner.

The ERP data, like the example in Figure 2.4, is loaded into a spreadsheet program. There it is displayed as a lot-for-lot schedule. The planner tries to make the schedule feasible by dragging (partial) production orders to earlier weeks, hereby he takes the capacity constraints into account.

The planner first makes the schedule for the Støperi Sukker line. He tries to fit as many production orders as possible into the first week, by looking at which future production orders can be dragged to this week, either in whole or partially. This process is repeated for the second week and the third week. The philosophy behind this approach is that by producing as many production orders as possible in the upcoming three weeks, enough capacity remains in future weeks to fulfil demand that lies further into the future.

The schedules for the Drage Sukker line and the packaging lines follow from the schedule of the Støperi Sukker line. For these lines the planner applies a simple policy where they package what has been produced earlier, which is now WIP on the factory floor. This policy works because the capacity of the packaging lines is higher than that of the production lines. The scheduling for the Støperi Sukker line is much more complex than that of the packaging lines and consumes therefore the most of the available time.

The MPS is created for the upcoming three weeks based on the forecasts for the next six months. It is necessary to create the MPS for at least the upcoming three weeks because the decision whether to run a 2-shift or 3-shift system on a production line needs to be made at least two weeks in advance. This decision is made by the planner and is part of the MPS planning.

The planner is hesitant to create an MPS for more than three weeks into the future because this is a time consuming task and the demand forecasts are incomplete for future weeks. Instead, it is considered more useful to spend this time on the creation of weekly schedules based on the MPS.

2.2.4 From MPS to weekly schedule

The planner uses the MPS to create a detailed schedule for the next week. In this schedule he specifies the sequence in which batches are produced by the different production lines (Støperi Sukker, Drage Sukker and the packaging lines) and assigns drying cabinets to each batch produced at the moulding machine.

The MPS is created based on estimates of the available capacity and productivity rates. In practice there are slight deviations. The planner can tweak the schedule by slightly increasing the length of shifts by asking people to work overtime or slightly reduce the production quantities. When necessary, he can run an extra overtime shift on Friday afternoon or Saturday. From the perspective of the labour regulations, these overtime shifts are different from the night shifts. Overtime shifts do not have to be announced two weeks in advance and are therefore the operators are compensated with an increased wage. This makes overtime shifts more expensive than night shifts.

2.2.5 Summary of the planning process

A summary and visualization of the different steps in the planning process and their interaction is given in Figure 2.5. Decisions are made on two levels, the MPS level and the weekly schedule level. Figure 2.5 shows that the planning process is a collaboration between the ERP system and manual interventions. In general, the manual interventions cover the tasks the ERP system is not able to perform.

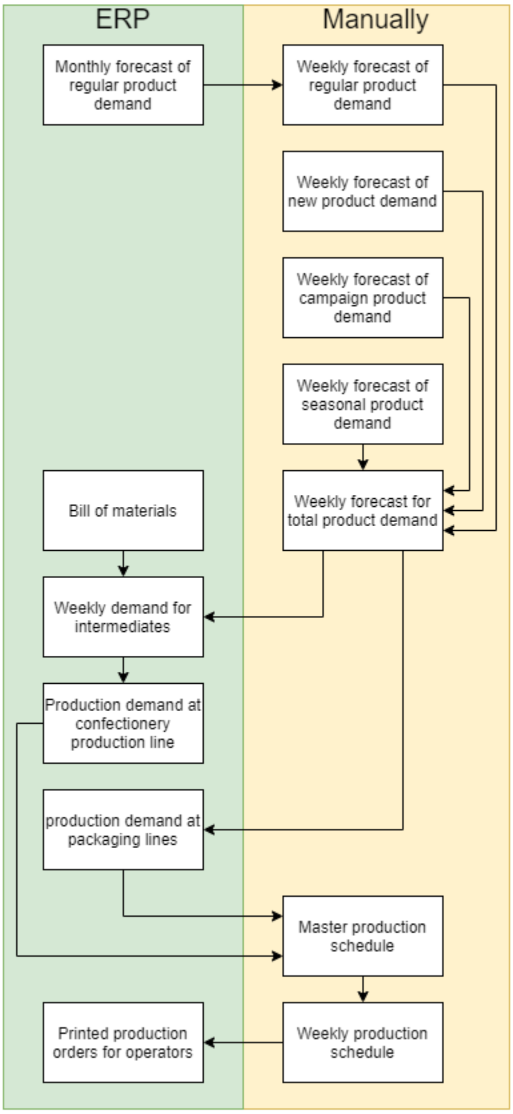

Figure 2.5 The entire planning & scheduling process of confectionery production at Brynild.

2.3 Performance of the planning process

In Section 2.2 the planning process is described in detail. In this section, the performance of this process is discussed. The most interesting performance metrics to production schedules are: the feasibility of the schedule, the quality the schedule, and the time needed to generate and report a schedule. The quality of the schedule is the extent to which the schedule achieves the objective (Harjunkoski, et al. 2014). In our case, this objective is the cost-effective fulfilment of the demand. Each of these performance metrics are discussed separately in relation to the MPS.

2.3.1 The feasibility of the MPS

In the current planning process the feasibility of the MPS is guarded by the planner. He uses his knowledge of the process to check whether capacity constraints are violated. On the MPS level he estimates whether a set of production orders fits into one week. In the weekly schedule he can solve small infeasibilities by allowing for some overtime or by reducing certain production quantities. The high degree of human control provides flexibility and makes it possible to solve infeasibilities in the planning in an ad hoc manner. However, it also makes the process time consuming. This leaves little time for improving the quality of the MPS.

2.3.2 The quality of the MPS

In the current situation the quality of the MPS is not evaluated. Therefore a method for evaluating the MPS is not directly available. In this section we propose a method for evaluating the quality of an MPS. The planner gets a set of production orders that serve as input for the MPS. His task is to create an MPS that ensures the fulfilment of these production orders in a cost-efficient manner. The quality of an MPS can therefore be evaluated by calculating the costs associated with an MPS.

Not all costs of production are influenced by the MPS. The costs of the production equipment and the procurement costs of the packaging materials are examples of costs that are not influenced by the MPS. When calculating the costs of an MPS, only those costs influenced by the MPS should be taken into account. For Brynild's confectionery production process, we consider the following costs to be influenced by the MPS:

1) **Changeover costs**
   When changing from the production of one intermediate to another at Støperi Sukker, the pipeline between cooking and moulding needs to be cleaned, resulting in the loss of material. The number of changeovers and therefore the height of this loss is influenced by the MPS.
2) **Holding costs**
   Holding costs are incurred for the finished product that lay in the warehouse. The MPS determines when products are produced and therefore influences the time products spend in the finished goods warehouse. Therefore these holding costs are influenced by the MPS too.
3) **Costs of lost sales**
   When demand exceeds the available capacity, not all demand can be fulfilled. Costs are incurred for these lost sales. In such a case the MPS determines which demand is fulfilled, therefore the costs of lost sales are influenced by the MPS.
4) **Operator salaries**
   The MPS determines the production quantities in each week and simultaneously whether a 2-shift or 3-shift system is used for each production line. Therefore the additional salary costs of running a 3-shift system are influenced by the MPS.

2.3.3 The time needed to generate a schedule

In the current situation the planning of the confectionery process, which spans the creation of the MPS and the weekly schedules is a full-time job executed by one person. The MPS is only created for the next three weeks because of time constraints. For both the MPS and the weekly schedules most of the available time is spent on the creation of a feasible schedule. This leaves little or no time for the optimization of these schedules.

2.3.4 Summary of the performance of the current planning process

The current planning process consists mostly of manual tasks. This makes the planning process time consuming. Since most of the available time is needed to create a feasible schedule, there is little or no time for improving the quality of the schedule. The quality of the MPS of Brynild's confectionery production process is not evaluated in the current planning process and a method for evaluating the MPS is non-existent. Therefore we propose such a method here. We use this method to evaluate our candidate solution in Chapter 5.

3. Literature review

In this chapter, research question 2 is answered: *What methods for the creation and optimization of an MPS are there in literature and which are applicable to Brynild's confectionery production?* Section 3.1 formally defines the problem of constructing an MPS and relates it to problems and models studied in literature. Section 3.2 describes different methods for the creation and optimization of an MPS that are found in literature. This chapter is concluded with a tabular overview of which solution methods excel when applied to different versions of the problem in Section 3.3.

3.1 Problem definition and extensions

According to our research objective, the problem of our interest is that of constructing an MPS that levels the workload at multiple stages of a production process. An MPS is a plan with the goal of scheduling production quantities in each period of the planning horizon, minimizing the cost and maximizing the bottleneck utilization (Herrera & Thomas, 2009). By this definition, an MPS should dictate when to produce a product and how much to produce of it. The production quantity is referred to as a so called lot. Constructing an MPS is equivalent to lot sizing, which seeks to determine the optimal timing and level of production (Clark, Bernardo, & Almeder, 2011).

Lot sizing problems can be divided into two classes based on whether the lot sizes to be determined should be static or dynamic. When the lot size of a product is determined once and then this lot size is used frequently, this is called static lot sizing. Static lot sizes are normally used when the underlying model assumes constant demand. An example of such a model is that of the economic-order-quantity (EOQ) model. When lot sizes are determined based on current (non-constant) demand, the lot size can change every time a product is produced. This is called dynamic lot-sizing (Broecke, Landeghem, & Aghezzaf, 2007).

Another criterion based on which lot sizing problems can be divided into two classes is whether the resources needed for production have a maximum capacity or unlimited availability. These problem classes are referred to as capacitated-lot-sizing-problems (CLSP) and uncapacitated-lot-sizing-problems respectively (Hein, Almeder, Figueira, & Almada-Lobo, 2018).

When resources have unlimited availability, determining the optimal lot size of a product is a trade-off between the setup costs for production and the holding costs for holding finished products, while taking into account the constraints imposed by demand. There is no interaction between products, allowing the problem to be solved one product at a time (Buschkühl, Sahling, Helber, & Horst, 2010).

When resources have limited capacity, determining the optimal lot size becomes harder. Florian et al. (1980) have proven that a single product CLSP without setup times is NP-hard. In the case of multiple products, the lot sizes of the different products can no longer be determined independently, since all products are competing for the same limited resource.

The confectionery production process at Brynild that is the subject of this research has capacitated resources and demand for products is variable. The variable demand implies dynamic lot sizes. Therefore the remainder of this chapter focusses on the dynamic CLSP.

3.1.1 The dynamic capacitated-lot-sizing-problem

The dynamic CLSP describes the following scenario. A finite planning horizon is divided into a number of discrete time periods (weeks). There are multiple products with period specific demands that must be met without delay. All products are produced on the same machine with limited capacity. The goal is to determine the production amounts for each product in each period such that the sum of setup and holding costs are minimized while capacity restrictions are respected (Dixon & Silver, 1981).

The dynamic CLSP has been intensively studied in the last decades so that different mixed-integer-programming (MIP) formulations, model extensions and solution approaches exist (Hein, Almeder, Figueira, & Almada-Lobo, 2018).

The original dynamic CLSP as formulated by Dixon and Silver (1981) is based on the following assumptions:

1) All products are produced on the same capacitated resource.
2) All demand is deterministic and must be fulfilled on time.
3) Production in a time period can be used to satisfy demand in this period.
4) Each time that production of a product is initiated, a setup costs is incurred.
5) Unit production costs are constant for each product. (Total production costs excluding setup costs are therefore constant and not included in the model.)
6) Setups have fixed costs per product and do not influence available capacity.
7) Holding costs are incurred for each unit of product that is carried from one period to the next.

These assumptions often do not hold in real world production processes. Numerous extensions to the original model have been proposed in order to make the model applicable to real world applications (Buschkühl, Sahling, Helber, & Horst, 2010). In the remainder of this section, we describe extensions proposed in literature that are relevant for making the dynamic CLSP applicable to Brynild's confectionery production process.

3.1.2 Multi-level extension and different product structures

The original dynamic CLSP assumes a singular product structure. Products with a singular structure are produced on one machine. In reality, products often need to be processed on multiple machines in a specific order (serial product structure). Products can also consist of multiple intermediate parts that also need to be produced (assembly structures) and those intermediate parts can in some cases even be used in different products (general structures). Figure 3.1 illustrates these different product structures.

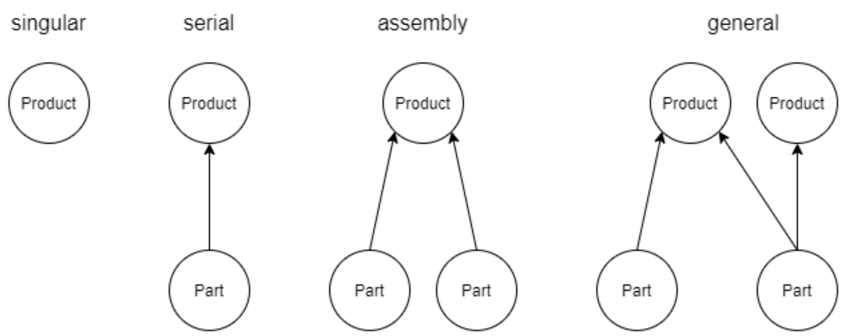

Figure 3.1 The 4 different types of product structures.

The extension of the CLSP to the multi-level CLSP (MLCLSP), sometimes referred to as the multi-stage CLSP, was introduced by Billington et al. (1983). They assume a fixed product structure in which the number of intermediate parts needed for an end-product or other intermediate part is known. This fixed structure is used to convert demand for an end-product, so called external demand, into demand for the intermediate parts, so called internal demand. Producing an end-product lowers the inventory levels of the required intermediate parts by the necessary amounts. Since inventory levels cannot be negative, this ensures availability of the intermediate parts.

This extension can also take into account fixed lead times for the intermediate parts. This is achieved by modifying the inventory constraints so that produced parts are added to the inventory in a later period, instead of in the time period that their production starts.

The extension proposed by Billington et al. (1983) makes it possible for the model to account for general product structures and fixed lead times. Since all other product structures are simplifications of the general structure, the MLCLSP can deal with those as well.

3.1.3 Capacitated resources extensions

Both the original CLSP and the MLCLSP assumes that only one capacitated resource is needed to produce a part or product. In reality, multiple capacitated resources might be needed. This can be incorporated in the model by adding additional capacity constraints for the operations that require multiple resources (Xie & Dong, 2002).

An original application of multiple capacitated resources has been researched by Boctor & Poulin (2005). They show that a multi-stage production process with serial product structures can be reduced to a single-stage production process with multiple capacitated resources by adding the assumption that lot sizes are the same for each stage. This reduces the complexity of the model, since the number of binary and continuous decision variables (for the setup costs and the production quantities) is reduced by a factor equal to the number of stages. The reduced complexity makes it easier to find good solutions for the model.

Another extension relates to the nature of the capacity constraints. The majority of CLSP related research assumes that the amount of a resource needed for production is proportional with the production quantity. This allows for linear capacity constraints. In reality this might not be the case. A MLCLSP with nonlinear capacity constraints is researched by Koken et al. (2017). They conclude that the nonlinearity of the constraints makes the problem much harder to solve.

3.1.4 Setup costs and setup times extensions

The original CLSP assumes fixed setup costs for each product, but no resources are used in this setup. In reality, this is not always the case. When a resource is capacitated by its available time, a time consuming setup reduces the remaining capacity. To model this, the capacity constraints can be altered to include the setup times for each product (Buschkühl, Sahling, Helber, & Horst, 2010). Some model formulations do not include setup costs at all. Instead, setup costs are only indirectly induced by the reduction of the available capacity through setup times. This approach is common when the costs of a setup are negligible compared to the labour costs and capacity loss caused during the setup time (Özdamar & Bozyel, 2007).

Regardless of whether setup costs, setup times or both are included in the model formulation, the standard assumption is that setups cannot be carried over to the next period. It is assumed that the setup needs to be done twice when the same product is produced at the end of a time period and the beginning of the subsequent period. In reality it might be possible to continue production of this product without a setup.

Integrating setup carryovers into the model is not straightforward because the CLSP model and its variations do not provide any information regarding the sequence of production within a time period (Buschkühl, Sahling, Helber, & Horst, 2010). Setup carryovers have been integrated into the CLSP (Dillenberger, Escudero, Wollensak, & Zhang, 1993) and into the MLCLSP (Sürie & Stadtler, 2003). However, both extensions add a significant number of additional variables to the model. This adds to the complexity of these models, making them harder to solve.

The absence of information regarding the sequence of production becomes an even bigger problem when sequence dependent setups are considered. Haase & Kimms (1999) study a standard CLSP with sequence dependent setup times. They were able to find good solutions by predetermining and storing efficient product sequences, but only for relative small problem instances of up to 10 items and up to 3 periods.

3.1.5 Possible overtime extension

In the original CLSP and all extensions discussed so far, the capacity of one or multiple recourses was assumed to be fixed. In reality, it is sometimes possible to increase the capacity of certain recourses in a given time period against increasing costs (Özdamar & Bozyel, 2007). An example of such a resource is manpower. Employees can work overtime to increase capacity, but usually against an increased hourly wage.

When the capacity of the resource is time based, this extra capacity is referred to as overtime capacity. Özdamar & Bozyel (2007) provide a formulation of a CLSP model with overtime decisions. In their model, capacity is constrained by the sum of the normal capacity and the maximum overtime capacity. The used overtime capacity is a decision variable and the total overtime costs are part of the cost function to be minimized. Özdamar & Bozyel (2007) apply their extension only to time based capacitated resources. However, their formulation can also be used for capacitated resources that are not time based, but whose capacity can still be increased against extra costs.

3.1.6 Extending the scale of the model

The scale of the model refers to the number of variables and constraints needed to formulate the model. Increasing the scale of the model is technically not an extension, since it does not change the structure of the model. However, like the extensions covered in the preceding part of this section, increasing the scale of the model increases the runtime, making it more difficult to find a good solution. This is why it is included in this section.

The reason that a larger model takes more time to solve is because more variables and constraints need to be taken into account. A common metric for the scale of a CLSP is the product of the number of items and the number of time-periods included in the model, because this largely determines the number of decision variables and constraints (Buschkühl, Sahling, Helber, & Horst, 2010).

3.1.7 Summary of the extensions for the CLSP

In this section, the original dynamic CLSP was introduced, after which the numerous extensions to this model that are studied in literature were described. These extensions cover different aspects of the model: product structure, capacity, setups and overtime. These extensions aim to make the model more realistic for real world applications. Thereby, they make the model more complex, making it harder to find feasible and (sub-)optimal solutions. The choice “What to include in the model?” is therefore almost always a trade-off between realism and solvability. Table 3.1 combines the different extensions covered in this section into a single overview. The extensions are grouped by aspect and ordered based on the complexity they add to the model.

The original CLSP is marked in yellow. Table 3.1 shows that all extensions discussed in this section lead to an increase of the models complexity. It possible to freely combine extensions for different aspects together into new models.

<table>
<tr><th></th><th colspan="2">Less complex</th><th colspan="2"></th><th colspan="2">More complex</th></tr>
<tr><th>Product structure</th><td>Singular</td><td colspan="2">Serial</td><td colspan="2">Assembly</td><td>General</td></tr>
<tr><th>Capacity</th><td colspan="2">Capacitated single resource</td><td colspan="2">Capacitated multiple resources</td><td colspan="2">Nonlinear capacity constraints</td></tr>
<tr><th>Setups</th><td>Setup costs</td><td colspan="2">Setup times</td><td colspan="2">Setup carryovers</td><td>Sequence dependent setups</td></tr>
<tr><th>Overtime possible</th><td colspan="3">No overtime possible</td><td colspan="3">Overtime possible</td></tr>
<tr><th>Scale of the model</th><td colspan="3">Small-scale</td><td colspan="3">Large-scale</td></tr>
</table>

Table 3.1 Extension to the CLSP, grouped by aspect and ordered by added complexity. The original CLSP is marked in yellow.

3.2 Solution methods for the CLSP and its extensions

In the previous section, the CLSP and several extensions were introduced. Many different solution methods for the CLSP have been studied in literature. In this section, these solution methods are listed. Since there are many different solution methods and numerous versions of the CLSP, the focus of this section is on identifying the combinations of methods and problems that have shown to yield good results. This knowledge is critical for the construction of a solution for Brynild in Chapter 4.

Buschkühl et al. (2010) studied more than 100 papers, published between 1970 and 2010, regarding different solution methods for CLSPs. They classify the different solution methods studied in literature in one of five categories: Mixed integer programming (MIP)-based approaches, Lagrangean heuristics, decomposition and aggregation heuristics, metaheuristics and problem-specific heuristics. MIP-based methods, metaheuristics and problem specific heuristics are identified as the most promising categories for solving real world applications. Meanwhile, Lagrangean and decomposition heuristics show a decline in research after the year 2000.

Therefore, the latter two categories are left out of this section. Instead, the focus is on MIP-based methods, metaheuristics and problem-specific heuristics. In the remainder of this section we discuss each of these 3 categories in detail.

3.2.1 MIP-based methods

MIP-based methods cover a broad category of solution methods. Every approach that models the problem as a MIP-model can be considered to fit into this category. MIP-models can be solved to optimality, but the time required to do so grows unacceptably large for models of realistic size. Therefore most MIP-based methods focus on strategies to reduce the computation time. Examples of such strategies are relaxing the binary variables such that they can take real values and pruning the solution space to exclude unpromising regions. A disadvantage of these methods is that when the model is too large, either the computation time increases or the solution quality decreases beyond acceptable bounds.

Another disadvantage is of MIP-based methods is that they are not applicable to all variations of the CLSP. This is for example the case when a CLSP has complex constraints that cannot be expressed in a MIP-formulation.

LP-based methods

LP-based methods are a distinguished class within the set of MIP-based methods. The LP-based approach is to identify a set of promising values for the binary variables of the MIP-model and fixate these variables. Once all binary variables are fixated, the MIP-model becomes a linear programming (LP)-model. LP-based methods exploit the fact that LP-models are easier to solve than MIP-models (Buschkühl, Sahling, Helber, & Horst, 2010).

The quality of the solution found by LP-based methods depends largely on the quality of the set of values used to fixate the binary variables. Determining this set can be done through other optimization routines like most metaheuristics. Metaheuristics are discussed in Section 3.2.3. Regardless of the method used to find a set of values for the binary variables, each candidate solution needs to be evaluated by solving the resulting LP-model. Therefore the efficiency with which the resulting LP-model can be solved is paramount for the success of LP-based methods (Jans & Degraeve, 2007).

When the resulting LP-model is very large, solving it might take a considerable amount of time. Since LP-based methods require solving multiple LP-models the total runtime can become quite long. In this case, the strategy of fixing the binary variables can still be applied, but heuristics need to be used to determine the production quantities. This strategy is studied by Xie and Dong (2002) amongst others. The use of these heuristics means deviating from an LP-based approach.

3.2.2 Problem-specific heuristics

The majority of the problem-specific heuristics are simple and intuitive methods. Because these methods are relative simple compared to other solution categories, they are by far the fastest way for creating a feasible solution to the CLSP (Hein, Almeder, Figueira, & Almada-Lobo, 2018). Their speed makes these heuristics suitable for solving large scale CLSPs.

Problem-specific heuristics can be split into 2 categories: Constructive and improvement heuristics.

Constructive heuristics

Constructive heuristics generate a solution from scratch, starting with an empty schedule. For the original CLSP, a constructive heuristic converts the given demand matrix into a feasible production plan. Most heuristics do this in a period-by-period fashion through either a forwards or backwards routine. Forward routines start at the first time period and work their way towards the last time period, while backward routines start at the last time period and work their way back towards the first time period (Buschkühl, Sahling, Helber, & Horst, 2010).

Most forward heuristics consider adding future demand orders to the production in the current week, thereby combining them and reducing the total number of lots. The idea behind this approach is that combining lots reduces the number of setups required and therefore the setup costs and time.

Moving production of future demand orders to earlier weeks increases the holding costs and might therefore not be profitable overall. Therefore most forward heuristics use some sort of savings index. A savings index is a formula that assigns a value to each future demand order based on certain parameters of that order. Common parameters include the setup and holding costs for the product and the number of time periods between the demand order and the current week. The assigned value indicates the attractiveness of adding the demand order to the current week's lot (Hein, Almeder, Figueira, & Almada-Lobo, 2018). Some forward heuristics also increase production in the current week because future capacity might not be sufficient to satisfy all future demand.

Backward routines work by moving demand from the current week to earlier weeks in order to solve infeasibilities in the schedule. An advantage of backward routines compared to forward routines is that they do not need to consider all lots in the weeks they have not worked through yet because moving lots past their due date is in general not possible in a CLSP (unless backorders are allowed).

Improvement heuristics

Improvement heuristics generate a better feasible solution from an inferior and possibly infeasible starting solution (Buschkühl, Sahling, Helber, & Horst, 2010). The starting solution is often a so called lot-for-lot production schedule. In a lot-for-lot schedule, the lot size of a period equals the demand in that period. Since a lot-for-lot schedule does not consider the capacity constraints, it is typically infeasible. An advantage of a lot-for-lot schedule is that it can be instantly created from the demand matrix. Therefore it is a simple starting solution for improvement heuristics that do not rely on the starting solution being feasible. This approach is for example used by Boctor & Poulin (2005). Improvement heuristics that do rely on the starting solution being feasible usually use a constructive heuristic to obtain this feasible starting solution, after which the improvement heuristic tries to improve this solution.

As opposed to constructive heuristics, that often work either backwards or forwards through the planning horizon. Improvement heuristics do not limit themselves to a specific order, but rather make changes all over the schedule. Improvement heuristics try to improve the schedule by shifting partial complete lots backwards or forwards, while chasing two objectives: make the schedule feasible and reduce the costs. There are heuristics that focus on reducing the costs first (Dogramaci, Panayiotopoulos, & Adam, 1981), while others first ensure a feasible schedule (Boctor & Poulin, 2005). Some heuristics pursue both goals simultaneously (Franca, Armentano, Berretta, & Clark, 1997).

Applying problem-specific heuristics to other CLSPs

As the name suggests, problem-specific heuristics are problem specific. Not all heuristics can be applied to all the variations of the CLSP. For example, a heuristic that relies on the assumption that total required capacity is independent of the schedule cannot be applied directly to a CLSP that includes setup times, since the total required capacity in a model with setup times depends on the number of setups, that in turn depend on how the production schedule combines lots together.

Therefore care must be taken when applying an existing heuristic to another problem. Many heuristics for the CLSP have been researched in literature and successfully applying a heuristic to a specific CLSP is a matter of choosing the right heuristic (Buschkühl, Sahling, Helber, & Horst, 2010). The right heuristic is should be applicable to the target problem and should preferable have shown good performance in previous research.

3.2.3 Metaheuristics

Metaheuristics are high level searching strategies for solving optimization problems (Buschkühl, Sahling, Helber, & Horst, 2010). They are high level strategies in the sense that they only guide the process of exploring the search space. They do not specify what the neighbourhood structure must be, how candidate solution are generated or how these candidate solutions should be evaluated. Therefore, metaheuristics do not contain any problem specific knowledge. This makes them applicable to a large variety of problems that are relative large and complex.

Metaheuristics need to incorporate other heuristics for generating neighbouring solutions and evaluating them. These are the so called lower level heuristics. The lower level heuristics can contain problem specific knowledge. Through this path, metaheuristics can exploit problem specific knowledge, while still being applicable to many different problem types (Buschkühl, Sahling, Helber, & Horst, 2010).

Most metaheuristics contain mechanisms that allow them to escape local optima. This enables them to explore a broader section of the search space. The search space of metaheuristics can also include infeasible solutions. When this is the case, a penalty function regarding the infeasibility is usually included in the objective function (Xie & Dong, 2002).

Genetic Algorithms (GA), Tabu Search (TS) and Simulated Annealing (SA) are three of the most well-known metaheuristics and are widely applied for solving complex combinatorial problems in general (Jans & Degraeve, 2007). Their popularity comes from the fact that they can be applied to a broad spectrum of optimization problems. They are also the most researched metaheuristics in lot-sizing literature (Buschkühl, Sahling, Helber, & Horst, 2010). In the remainder of this subsection, we describe these three methods and their appliances to the CLSP and related problems.

Genetic Algorithm (GA*)*

GAs are inspired by the principles of natural selection. They consider multiple solutions simultaneously, that are together referred to as a generation. In each iteration, a new generation of solutions is created by performing certain operations on the previous generation. These operations are carried out in such a way that the best solutions from the previous generation have a higher chance of reappearing and influencing the solutions in the new generations (Buschkühl, Sahling, Helber, & Horst, 2010).

Since GA considers multiple solutions simultaneously, it has the advantage that its performance is not heavily influenced by the quality of the original solutions. It is common that the first generation of a GA consist of random generated solutions. The lack of dependence on a good starting solution is a great advantage when it is hard to create good starting solutions.

A drawback of GA is that the operations that create a new generation of solutions become computationally heavy when a large number of variables is needed to represent a solution. This is often the case in CLSP problems of realistic size and the main reason why GA's fail to find good solutions for these problems (Xie & Dong, 2002).

Özdamar and Bozyel (2007) used a GA to solve a CLSP with overtime decisions and setup times. They concluded that the GA gave good solutions for small problem instances, but that the heavy data structure needed made it impossible to achieve good solutions for larger instances. Xie and Dong (2002) solve a similar problem, but only include the binary setup variables of the solution into GA. They use another heuristic to derive a complete solution from a set of fixed binary variables. This reduces the computational burden somewhat, but they are still only able to solve small and medium problem instances of up to 21 items and 6 time periods.

Tabu search (TS)

TS explores a search space from a starting solution by moving to the best of its current neighbouring solutions, regardless of whether this neighbour is better than the current solution. To prevent itself from hopping between two neighbouring solutions forever, it uses a so called Tabu list. The Tabu list stores information on recent moves to prevent their reversal. This allows TS to escape local optima. The length of the Tabu list is decisive for the performance of TS. The optimal length depends on problem specific characteristics (Buschkühl, Sahling, Helber, & Horst, 2010).

A drawback of TS is that it evaluates all neighbouring solutions before making a move. Depending on the neighbourhood structure this can be a computationally heavy procedure. Within CLSP, neighbours are usually generated by moving a particular lot of one item, either partially or completely (Jans & Degraeve, 2007). This means that large CLSP problems have equally large neighbourhood structures.

Hung et al. (2003) partially solve the problem of a large neighbourhood by defining a neighbourhood structure based on minor changes to the setup variables. They solve the remaining LP model to optimality to evaluate a set of fixed setup variables. This approach means that less neighbours have to be evaluated before making a move. However, an LP model needs to be solved for the evaluation of each neighbour. Therefore the scale to which this approach can be applied depends on how efficient the resulting LP model can be solved (Jans & Degraeve, 2007).

Simulated Annealing (SA)

SA is a metaheuristic for optimizing combinatorial problems that was first proposed by Kirkpatrick et al. (1983). SA starts with an initial solution and tries to improve on this solution by exploring random neighbouring solutions. Better solutions are always accepted, while worse solutions are accepted with a certain probability. This probability depends on how much worse the solution is and a parameter called the temperature. At the start, the temperature is high and almost all solutions are accepted. Slowly, the temperature decreases so that in the end only better solutions are accepted. This simple mechanism allows SA to escape local optima and explore the search space, while still ensuring that local optima are exploited in the end.

SA holds the advantage over GA that it can solve larger instances of the CLSP (Özdamar & Bozyel, 2007). SA requires only a relative simple neighbourhood structure and it only needs to remember one solution, the best solution so far. GA needs to remember all solutions in a generation and requires relative complex operations to create the new generation.

Both SA and TS need to be able to roam the solution space in order to obtain good results. They are best abled to do so when the solution space is continuous. Although the original CLSP has a fairly continuous solution space, certain extensions make the solution space fragmented. Examples of such extensions are more complex product structures. More complex product structures increase the constraints on inventory and availability of parts that in turn cause discontinuities in the solution space.

3.2.4 Summary of solution methods for CLSP

We conclude with a summary of the (dis)advantages of the 3 categories of solution methods we discussed in this section. Table 3.2 shows this summary.

Category	Advantages	Disadvantages
MIP-based methods	• Less dependent on problem characteristics.	• Not all CLSP variations can be formulated as MIP-models. • Not suitable for solving large problems
Problem-specific heuristics	• Faster than methods from other categories. Therefore suitable for solving large problems.	• Choosing the right heuristic might be difficult. • Not clear what approach works best.
Metaheuristics	• General methods that can be applied to many different variations of the CLSP.	• GAs are not suitable for solving large problems. • SA and TS are not suitable for very discontinuous solution spaces.

Table 3.2 Overview of the (dis)advantages of the different solution categories.

4. The proposed planning method

In Chapter 3, we showed that many variants of the CLSP are studied in literature and that multiple solution method are available. In this chapter, we answer research question 3: *What method or methods are promising candidate solutions for achieving the research objective?* To do this we combine the insights from Chapter 3 with the knowledge of Brynild's confectionery production from Chapter 2 to achieve two things. First, in Section 4.1, we define a variant of the CLSP that represents the planning of Brynild's confectionery production. Second, in Section 4.2 we choose an existing solution method from literature as a basis for our own method. In Section 4.3 we make modifications to this method to make it applicable with the capacity constraints. In Section 4.4 we make other modifications to make this method applicable to our problem. We conclude this chapter with a summary of the proposed planning method in Section 4.5.

4.1 Brynild's confectionery production planning as a CLSP

The first step in the construction of a new planning method for the MPS is to decide which parts of the production process should be included in this method. We then determine which of the extensions to the original CLSP, we need to include in our method to represent the chosen parts of the production process.

4.1.1 The process parts covered by the method

In Section 3.1 we concluded that the decision of what to include in our method is a trade-off between realism and solvability. We need to include enough in the method so that the solution is useful in practice, but not so much that the method becomes too complex. When the method becomes too complex, the runtime can become too long. As a consequence, the method might fail to find a good solution.

As described in Section 2.1, the confectionery production process consists of two production lines (Støperi Sukker and Drage Sukker) and four packaging lines. In the current method, the MPS for the Støperi Sukker line is created first. The MPS of the subsequent lines is then created based on that of the Støperi Sukker line. The assumed independence of the MPS of the Støperi Sukker line from the MPS's of the other lines does not cause significant problems in practice. The practice of scheduling the Støperi Sukker line prior to the packaging lines makes the scheduling of the latter simpler, since this is now mostly dictated by what is produced at the production lines. Because of these reasons, a planning method for the MPS that only includes the Støperi Sukker line can provide a solution that is useful in practice.

As explained in Section 3.1.6, a common metric for the scale of a CLSP type model is the product of the number of items and the number of time periods. Excluding the packaging lines and the Drage Sukker line from our model reduces the scale of the model by around half, since it reduces the number of items from 85 to 48. This reduction moves our model closer to the range of those we encountered in our literature review, since largest model we encountered had 34 items. Therefore we consider a planning method for the MPS that only includes the Støperi Sukker line, a good trade-off between usefulness and solvability. We therefore exclude the packaging lines from our model and include only the two production lines Støperi Sukker and Drage Sukker.

4.1.2 Notation

Before we go into the details of the proposed planning method we introduce some notation for the decision variables and parameters used through this chapter. For the sake of consistency all product quantities are in kilos and all time units are in shifts (1 shift = 8 hours). The only exception are the periods in the planning horizon, which are weeks.

Sets:

T	set of all weeks in the planning horizon, weeks are numbered $_{111}$ to T, $t \in T$
K	set of all intermediates, $k \in K$

Decision variables:

Q_{kt}	production quantity of intermediate *k* at in week t in kilos
Y_{kt}	is 1 if intermediate *k* is produced in week *t*, 0 otherwise
O_t	is 1 if night shifts are enabled in week *t*, 0 otherwise
I_{kt}	inventory level of intermediate k at the end of week *t* in kilos
U_{kt}	lost sales of intermediate *k* in week *t* in kilos

Parameters:

c_{change}	costs of 1 changeover
c_{night}	additional costs of running a 3-shift system for 1 week
cu_k	costs of a kilo of lost sales of intermediate *k*
a_t	number of shifts available in week t in 2-shift system
an_t	this parameter has value 1 if night shifts are available in week *t*, 0 otherwise
ns	extra shifts available if a 3-shift system is used
d_{kt}	demand for intermediate *k* in week *t* in kilos
h_k	holding costs of 1 kilo of intermediate *k* for 1 week
p_k	kilo of intermediate *k* that can be produced during a full shift
l_k	the amount of intermediate *k* that fits into 1 drying cabinet
nd_k	the drying time of intermediate *k* in a new drying cabinet in shifts (1 shift = 8 hours)
od_k	the drying time of intermediate *k* in an old drying cabinet in shifts (1 shift = 8 hours)
st	setup time in shifts

4.1.3 Objective function

The objective of the method is to create a cost-efficient MPS, meaning that the objective is to minimize the sum of the related costs. We identified the related costs in Section 2.3. With the notation introduced in the previous section, we can formulate the objective function. The total costs are the sum of four different types of cost: The setup costs (in red), the holding costs (in yellow), the costs for using night shifts (in green) and the costs of lost sales (in blue).

$$\min z = c_{change} \cdot \sum_T \sum_K Y_{kt} + \sum_T \sum_K h_k \cdot I_{kt} + c_{night} \cdot \sum_T O_t + \sum_T \sum_K cu_k \cdot U_{kt}$$

4.1.4 The drying constraints

Most of the CLSP formulations we encountered in the literature only use linear capacity constraints. This means that there is a limited amount of a resource available and that the amount of the resource that is needed to produce a certain lot size of a product is proportional with the lot size.

In Section 2.1.2 we described the workings of the drying process. We now show that this leads to nonlinear capacity constraints. We do have a capacitated resource since there are a limited number of drying hours available each week. There are only 5 drying cabinets, that have finite capacity, each of which can dry products for at most 24 hours a day. As explained in Section 2.1.2, the number of drying hours consumed by the production of a certain lot equals the moulding time plus the drying time. The drying time depends on the number of cabinets that is required to fit all the produced products and the drying time depends on whether these cabinets are old or new cabinets. We need to take into account this uncertainty, since at the moment we create the MPS, it is not yet known in what type of cabinet the product will be dried. The number of drying hours consumed by the production of a certain lot of product k is therefore equal to:

$$\text{drying hours consumed by product } k \text{ in week } t = 8 \cdot \frac{Q_{kt}}{p_k} + 8 \cdot \left\lceil \frac{Q_{kt}}{l_k} \right\rceil \cdot \begin{cases} nd_k & \text{if dried in a new cabinet} \\ od_k & \text{if dried in an old cabinet} \end{cases}$$

As an example, we plotted the drying hours required for a lot of a product called “HF Jordbærfisker” in Figure 4.1. The blue and orange lines show the drying hours required if all products are dried in new and old cabinets respectively. However, once at least two drying cabinets are required, a mixture of old and dry cabinets is also possible. Therefore these lines should be viewed as the lower and upper bound of the required drying hours. The plot shows the nonlinear relation between lot size and consumption of the capacitated resource.

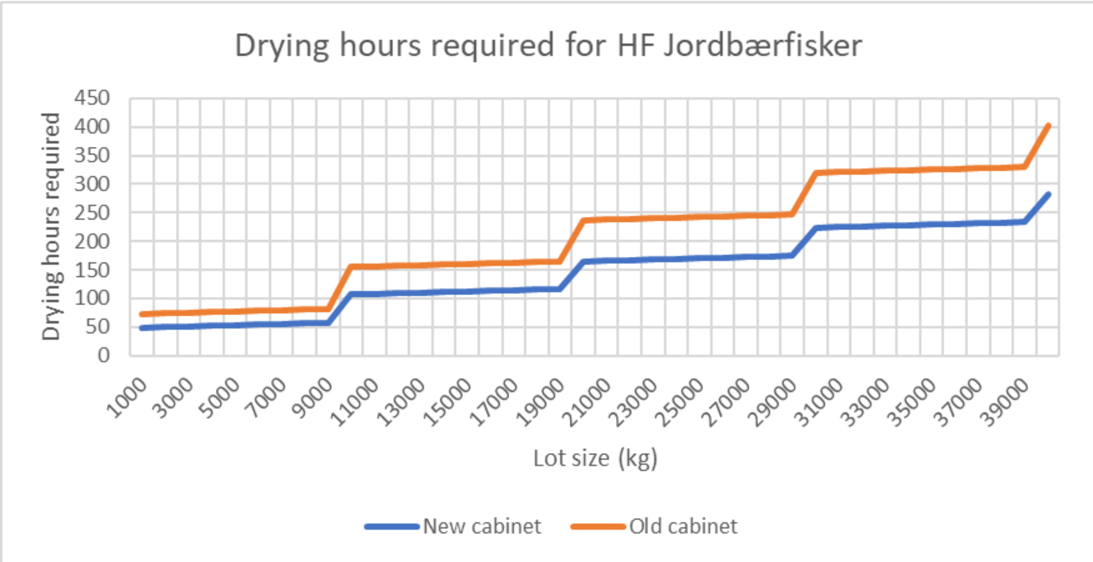

Figure 4.1 The drying hours required vs the lot size for "HF Jordbærfisker".

This section shows that the capacity constraints imposed by the drying cabinets, make Brynild’s confectionery production are more complex than the linear capacity constraints we encountered in our literature review. In Section 4.3 we propose how to deal with this added complexity.

4.1.5 Included extensions

<table>
<tr><th></th><th colspan="2">Less complex</th><th colspan="2"></th><th colspan="2">More complex</th></tr>
<tr><td>Product structure</td><td>Singular</td><td colspan="2">Serial</td><td colspan="2">Assembly</td><td>General</td></tr>
<tr><td>Capacity</td><td colspan="2">Capacitated single resource</td><td colspan="2">Capacitated multiple resources</td><td colspan="2">Nonlinear capacity constraints</td></tr>
<tr><td>Setups</td><td>Setup costs</td><td colspan="2">Setup times</td><td colspan="2">Setup carryovers</td><td>Sequence dependent setups</td></tr>
<tr><td>Overtime possible</td><td colspan="3">No overtime possible</td><td colspan="3">Overtime possible</td></tr>
<tr><td>Scale of the model</td><td colspan="3">Small-scale</td><td colspan="3">Large-scale</td></tr>
</table>

Table 4.1 Extension to the CLSP, grouped by aspect and ordered by added complexity. The extensions needed to model Brynild’s confectionery production process are marked green.

We concluded Section 3.1 with an overview of the different extensions to the CLSP. Table 4.1 provides a similar overview, with the extensions we need to incorporate in our model marked in green.

Product structure

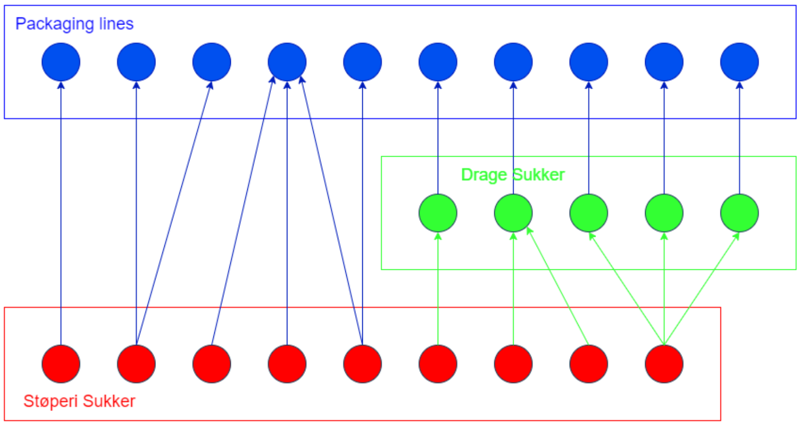

Figure 4.2 The different product structures present in the confectionery production process.

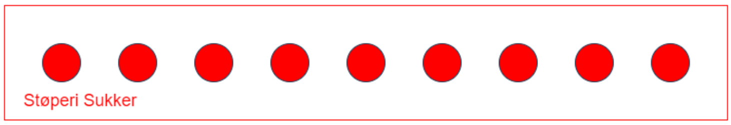

Figure 4.3 The different product structures present in the model.

Figure 4.2 shows the different product structures that can be found within Brynild's confectionery production. We only include the Støperi Sukker line in our model. This simplifies the product structures found in our model to only singular products, as is shown in Figure 4.3.

Capacity

The Støperi Sukker line has two capacitated resources, the moulding machine and the space in the drying cabinets. Therefore the model needs to incorporate multiple capacitated resources. In addition the capacity constraints of the drying cabinets are nonlinear as we have shown in Section 4.1.3.

Setups

The moulding machine at the Støperi Sukker line incurs both setup costs (through the loss of material) and setup times (because the machine needs to be stopped). Therefore we incorporate both into the model.

Overtime

It is possible to run night shifts against additional costs. The model should therefore incorporate the possibility of overtime. All literature we found that incorporated overtime into CLSP models considered the costs of overtime to be proportional with the extra time needed. However, in our model this is a binary decision, since it is only possible to run an entire week with night shifts or an entire week without night shifts.

Scale

The Støperi Sukker line produces 41 different intermediates and the timespan of the MPS is 26 time periods (weeks). This places the model in the large-scale category, since the largest models we found in our literature review consists of 34 items and 15 time periods.

4.1.6 Positioning of our problem to CLSP literature

The main thing that separates our problem from the CLSP variants we encountered in the literature is the set of capacity constraints for the drying cabinets. Ensuring that the proposed planning method can work with these constraints is the main challenge of this research. Besides this major difference, there are 4 other differences between our problem that we did not encounter in literature:

1) The scale of our model, measured in the number of items and number of time periods, is larger than any of the models we encountered in our literature review.
2) The decision to use overtime, through the use of night shifts, against additional costs is binary. The reviewed papers that incorporated overtime decisions into their models made this a proportional decision where it is possible to include just enough overtime to finish the required production.
3) The possibility to include lost sales into the model. We have not found a paper that incorporates the possibility to not fulfil all demand in a CLSP. The papers that we found either check the feasibility beforehand or stop the method if it discovers that a feasible schedule does not exists. In both cases they deem the problem infeasible and do not provide a (partial) solution. This is not possible in our case because even if demand is higher than the available capacity, a production schedule is still needed.
4) Our model has one production line with multiple capacitated resources: the moulding machine and the drying cabinets. There are papers that cover multiple capacitated resources, but their number is significantly smaller than those covering singe capacitated resources. Not all solution methods that are proposed for model with single capacitated resources can be applied to models with multiple capacitated resources. Therefore this characteristic must also be taken into account when choosing a solution method.

4.2 Choosing a solution method

In Chapter 3 we described the three main categories of solution methods for the CLSP and their respective (dis)advantages: MIP-based methods, metaheuristics and problem-specific heuristics. To decide in which of these 3 categories we look for a solution method, we first take a closer look at the model we formulated in the previous section and how this model is different from the models we encountered in our literature review. We then consider all 3 categories and look for a solution method in the most suitable category.

4.2.1 The most suitable solution category

In this subsection we identify the most suitable solution category for our model. We consider the three main categories of solution methods for the CLSP: MIP-based methods, metaheuristics and problem-specific heuristics.

All MIP-based methods require that the underlying model is a MIP-model. Because of the capacity constraints of the drying cabinets, our model is not a MIP-model. A possible course of action would be to approximate the nonlinear capacity constraints with a linear approximation. This would result in an unfeasible schedule that would have to be adjusted to meet the nonlinear capacity constraints. However, this MIP-model would be considerably larger, in terms of the number of variables and constraints, than any MIP-model we encountered in the reviewed papers. Our literature review showed that MIP-based methods are the least suitable of the three categories to solve large models. This means that finding a good solution within reasonable time is a difficult task. These arguments makes us consider the category of MIP-based methods to be not suitable for our model.

Metaheuristics can be applied to a wide variety of problems, including the CLSP. The most common used metaheuristics are GA, TS and SA. With the exception of GA, these metaheuristics are better able to solve large models than MIP-based methods. However, in order to achieve good results TS and SA need to be able to explore the entire solution space. They are best able to do this when the solution space is continuous or close to continuous. The need for a close to continuous solution space is why these metaheuristics are less suitable for our model. Even when the binary variables are fixed, the solution space of feasible production quantities is discontinuous. This discontinuity is a result of the nature of the capacity constraints of the drying cabinets. This nature is illustrated by the following example: If the size of a lot currently fills exactly one drying cabinet, increasing the lot-size by just 1 kilo would almost double the amount of drying capacity required, as now 2 cabinets are needed. The discontinuous behaviour is also illustrated in Figure 4.1. This discontinuous solution space makes metaheuristics less than ideal solution methods for our model.

Problem-specific heuristics are in general faster than the methods from the other two categories. This is an important advantage because our model is larger than any model we encountered in literature. Many different problem-specific heuristics exists in literature. Which one performs best depends on the characteristics of the model and the pattern of the demand. The logic behind most problem-specific heuristics is relative simple compared to that of metaheuristics. This makes it easier to modify the method to fit another model. This is a useful characteristic for our case, since our model has features that are not encountered in literature. Therefore some modification is necessary to fit our model, regardless of which method is chosen.

Their ability to handle large scale models and the relative easiness with which they can be modified make the category of problem-specific heuristics the most suitable to find solution methods for our model.

4.2.2 A basis for the proposed planning method

Our solution method should incorporate the unique characteristics of our model that are not found in literature. However, we can use a method from literature and use it as a basis for our own method. In the previous section we concluded that the category of problem-specific heuristics is the most suitable category for looking for a solution method for our model. In Section 3.2.2 we explained the importance of choosing the right heuristic when applying a problem-specific heuristic to another model. The choice must be made based on the performance of the heuristic in the original model and the similarities between the original and the target model.

There are many problem-specific heuristics researched in literature. We compare the planning problems that these heuristics are applied to with our own planning problem, that we formulated in Section 4.1. The problem that we found in our literature review that seems closest to our own problem is studied by Boctor & Poulin (2005). They study a production process that consists of a number of sequential machines, but show that, under the constraint that the lot sizes must be equal for all machines, they can model the process as a single level CLSP with multiple capacitated resources. This is similar to our problem, which is also a single level CLSP with multiple capacitated resources.

Boctor & Poulin propose a constructive heuristic to solve this problem. They test the performance of their heuristic for small to medium sized problem instances. Their experiments show that their heuristic finds solutions that are on average 6.6% worse than the optimal solution for small problem instances. When the size of the problems increases the average solution gap decreases to 3.3% for the medium sized problem instances. These results show that the performance of the heuristic increases with the problem size. This is a useful characteristic for our research because the size of our problem instances is larger than any problem we encountered in literature.

Because the problem of Boctor & Poulin is similar to our problem and because their heuristic shows good performance for increasing problem sizes we choose to use the heuristic proposed by Boctor & Poulin as a basis for our own proposed planning method. In the remainder of this section we give a detailed description of their proposed heuristic.

Boctor & Poulin's heuristic

The heuristic developed by Boctor & Poulin is a constructive heuristic. It is based on a forward routine. This means that the heuristic starts by considering the first week of the planning horizon, after which it considers the second week and continues in this way until it reaches the last week of the planning horizon.

In each week, the production quantities are initially set to the net demand. For each item, the net demand is the gap between the demand for that week and the inventory level at the beginning of the week. Or in other words, what needs to be produced in that week to meet demand.

Once the initial production quantities are set, there are two possibilities: Either there is some residual capacity in this week or the initial production exceeds the available capacity.

When there is residual capacity, the heuristic tries to increase the lot sizes in the current week. This is done by calculating the savings index for each future demand order. The heuristic then tries to incrementally add the orders with a positive savings index to the current week's production, starting with the order with the highest savings index. Once all orders with a positive savings index have either been added or disregarded the heuristic continues with the next week.

When the initial production exceeds the available capacity the heuristic uses the following procedure. First it sets all production quantities in the current week to zero. Then it tries to add this week's demand orders incrementally to the production in the current week. The orders are added in decreasing order of their unit holding costs. Since we initially did not have enough capacity in this week, some orders cannot be added to the current week. For these orders the heuristic applies a so called backtrack procedure. In this backtrack procedure the heuristic tries to allocate the remaining orders to earlier weeks, given that there exists available capacity in those weeks. First it tries to do this by increasing the lot sizes in previous weeks, since increasing the size of existing lots does not create extra setup costs. When it is not possible to allocate all remaining orders by increasing existing lot sizes, the procedure tries to allocate the remaining orders by creating new lots in the previous weeks.

This process is repeated for all weeks in the planning horizon. The result is a feasible production schedule, given that one exists (Boctor & Poulin, 2005). The pseudo code for the heuristic is given in Box 4.1.

```
Boctor & Poulin heuristic:

For t = 1 to T {
        Set initial production in week t equal to net demand
        If (used capacity < available capacity) {
                Calculate the savings index for all orders in future weeks
                Sort all future orders in decreasing order of savings index
                For each order with a positive savings index {
                        Add the order to this week's production if it does not break capacity
                        constraints
                }
        } Else If (used capacity > available capacity) {
                Set production in the current week to zero

                For each unallocated order {
                        Add the order to this week's production if it does not break capacity
                        constraints
                }

                For each unallocated order {
                        For w = t-1 to 1{
                                If (there exists a lot of this product in week w) {
                                        Increase the lot size with the minimum of the
                                        unallocated amount and the residual capacity in week w
                                }
                        }
                }

                For each unallocated order {
                        For w = t-1 to 1{
                                If (there does not exists a lot of this product in week w) {
                                        Create a lot with a size the minimum of the unallocated
                                        amount and the residual capacity in week w
                                }
                        }
                }
        }
}
```

Box 4.1 Pseudo code for the Boctor & Poulin heuristic.

4.3 Boctor & Poulin heuristic and the drying constraints

In this section we show how we make the heuristic of Boctor & Poulin work with the capacity constraints of the drying cabinets. A detailed explanation of this heuristic is given in Section 4.2.2. From this description it becomes clear that the heuristic needs to calculate the remaining capacity for a product in a week, given an initial set of production quantities. With the remaining capacity of a product, we mean the maximum amount of that product that can be added to that week's production without violating the capacity constraints. From the explanation in Section 4.1.3 it is clear that this is not a straightforward calculation as the drying hours consumed by the initial set of production quantities depend on the drying cabinets that they are allocated to.

Even determining whether a set of weekly production quantities is feasible with respect to the drying constraints is not straightforward as this requires that one checks whether the products can be distributed over the 5 drying cabinets such that the capacity of none of the 5 cabinets is exceeded.

To solve this problem, we created an efficient method to determine whether a set of weekly production quantities is feasible. We describe this method in Section 4.3.1. Through the use of this method we also find a method to make the Boctor & Poulin heuristic work for solving our problem. We describe this method in Section 4.3.2.

4.3.1 Determining the feasibility of a set of weekly production quantities

In this section we describe our proposal for an efficient method to determine whether a set of weekly production quantities is feasible. We first define Q_t as a vector containing all production quantities Q_{kt} for a certain week t. We represent our method as a Boolean function that takes Q_t as input and that returns true when this set can be dried within a week and false otherwise:

$$f(Q_t) = \begin{cases} true & if\ Q_t\ can\ be\ dried\ within\ a\ week \\ false & if\ Q_t\ cannot\ be\ dried\ within\ a\ week \end{cases}$$

From a mathematical point of view checking whether a given set of weekly production can be dried within a week is somewhat similar to a more complex version of the optimization version of the partition problem. In the optimization version of the partition problem, one aims to partition a given multiset S of positive integers over k different buckets such that the difference between the sum of the elements of the different buckets is minimized.

In our function, we cut the production quantities in the function's input into smaller batches that each fit into a drying cabinet. The number of batches for each intermediate can be calculated by dividing the production quantity Q_{kt} of intermediate k in week t, by the amount of intermediate k that fits into one drying cabinet. In formula notation:

$$\#batches_{kt} = \left\lceil \frac{Q_{kt}}{l_K} \right\rceil \quad \forall k \in K$$

The time each batch occupies a drying cabinet is equal to the sum of the drying time of the intermediate and the moulding time of the batch. The moulding time, but not the setup time, needs to be included because the drying cabinet must be available while moulding. Assuming all batches of k are of the same size, the total occupation time of each batch can be calculated with the following formula:

$$occupation\ time\ \ for\ a\ batch_{kt} = \frac{1}{p_k} \cdot \frac{Q_{kt}}{\#batches_k} + \begin{cases} nd_k\ if\ dried\ in\ a\ new\ cabinet \\ od_k\ if\ dried\ in\ an\ old\ cabinet \end{cases} \quad \forall k \in K$$

Note that our assumption that all batches of k are of the same size only affects the moulding time of each batch, not the drying time. As the drying time is many times higher then the moulding time, the effect of this assumption is limited.

We now have a list of batches and the occupation time of each batch on this list when it is dried in a new cabinet and when it is dried in an old cabinet. If we can divide the batches over the 5 drying cabinets in such a way that the sum of the occupation time of the batches in each drying cabinet is lower than a week, the production in this week is assumed to be feasible. We make this assumption because each drying cabinet is available 24 hours a day, 7 days a week. When this is not possible the production in this week is infeasible. If the production is feasible, the function $f(Q_t)$ returns true, otherwise it returns false.

As shown in the above formula, the occupation time of a batch depends on whether it is put in a new or and old drying cabinet. This makes checking whether a feasible partition exists more complex.

To prove that a feasible partition exists, one only needs to provide a feasible partition, but too prove no feasible partition exists, one must in general enumerate over all possible partitions and show that each one is infeasible. However, we consider the approach of total enumeration too time consuming for our purpose. Instead we let a greedy partition algorithm try to find a feasible allocation. If this algorithm succeeds the function returns true, otherwise the function returns false. This approach leaves the possibility of a false negative, where the function returns false while the production in that week is in fact feasible. However, a positive is always a true positive.

The greedy partition algorithm

The greedy partition algorithm tries to partition this set of batches over the 2 new and the 3 old drying cabinets such that the sum of the occupation time of the batches in each cabinet does not exceed a week. A complication of this task is that the occupation time of a batch is dependent on whether it is put in a new or and old drying cabinet.

To work around this complication our version of the greedy partition algorithm uses a priority ranking, where the batches with the highest priority are allocated first. The idea behind this priority ranking is that the batches that profit the most from being dried in a new drying cabinet are allocated first. The priority of a batch is equal to $nd_k - od_k$.

For each batch the algorithm first tries to allocate the batch to the new drying cabinet with the most available time. If it is not possible to add the batch to this new drying cabinet without exceeding the seven days cap, it is also not possible to add it to the other new drying cabinet and the algorithm tries to allocate the batch to the old drying cabinet with the most available time. When it is also not possible to allocate the batch to this old drying cabinet without exceeding the seven days cap, the algorithm stops and the function returns false. When all batches can be allocated through this method the function returns true. The pseudo code for this greedy partition algorithm is shown in Box 4.2.

Greedy partition algorithm

```
For each batch in order of decreasing priority {
        If (current batch fits in new drying cabinet) {
                Add current batch to emptiest new drying cabinet
        } Else If (current batch fits in emptiest old drying cabinet) {
                Add current batch to emptiest old drying cabinet
        } Else {
                Return: false
                Stop
        }
}
Return: true
```

Box 4.2 Pseudo code for the greedy partition algorithm.

4.3.2 The linear approximation and backwards feasibility routine

With the method described in Section 4.3.1. we can determine whether a set of weekly production quantities is assumed to be feasible. It can however not calculate the remaining capacity of a product in a week, which is a requirement for Boctor & Poulin's heuristic.

If the capacity constraints of the drying cabinets were linear, calculating the remaining capacity would be straightforward. Therefore we propose the following approach:

1. Run the Boctor & Poulin heuristic, while using a linear approximation of the capacity constraints of the drying cabinets. This results in an approximately feasible MPS.
2. Run a backwards feasibility routine that uses the method described in Section 4.3.1 for determining the feasibility of a set of weekly production quantities. This results in a feasible MPS.

In the remainder of this section we describe both steps in more detail:

Boctor & Poulin heuristic with linear approximation

The pseudocode for the linear approximation is the same as that of the original heuristic that is shown in Box 4.1. The only difference is that we use a linear approximation for the remaining capacity of the drying cabinets. This linear approximation works as follows:

- There are 24 hours in a day, 7 days in a week and 5 drying cabinets. Therefore the total available drying hours each week is $24 \cdot 7 \cdot 5 = 840\ drying\ hours$.
- The filling of a drying cabinet starts during the moulding, therefore the moulding time is included in the drying time.
- In the linear approximation we pretend that the drying time of each item to be as if it is dried in a new drying cabinet.
- In the linear approximation we pretend that the drying hours used by an item to be proportional with the occupation of the cabinets. So if a half filled drying cabinet is used for an hour this consumes half a drying hour.
- Example: If an item needs to dry for 48 hours and we produce enough to fill 1.5 drying cabinet, which takes 4 hours of moulding time. This consumes a total of $4 + 48 \cdot 1.5 = 76\ drying\ hours$.

From the linear approximation method described above, we can derive the following formula for the drying hours used by a set of production quantities in a certain week t:

$$Drying\ hours\ used(t) = \sum_{K} \frac{Q_{kt}}{p_k} + nd_k \cdot \frac{Q_{kt}}{l_k}$$

From the above formula we can deduce the formula for the remaining drying capacity. We have defined the remaining drying capacity as the kilos of intermediate k we can add to the production of week t without breaking the linear approximation of the drying capacity constraint. This results in the following formula:

$$remaining\ capacity(k,t) = \frac{840\ hours - drying\ hours\ used(t)}{\frac{1}{p_k} + \frac{drying\ time_k}{l_k}}$$

Figure 4.4 is an expansion of Figure 4.1. It shows the linear approximation of the drying hours used plotted against the lot size for the product “HF Jordbærfisker”. It also shows the actual drying hours, which are the same as those in Figure 4.1. From this figure it becomes clear that the linear approximation is always lower or equal to the actual drying hours used.

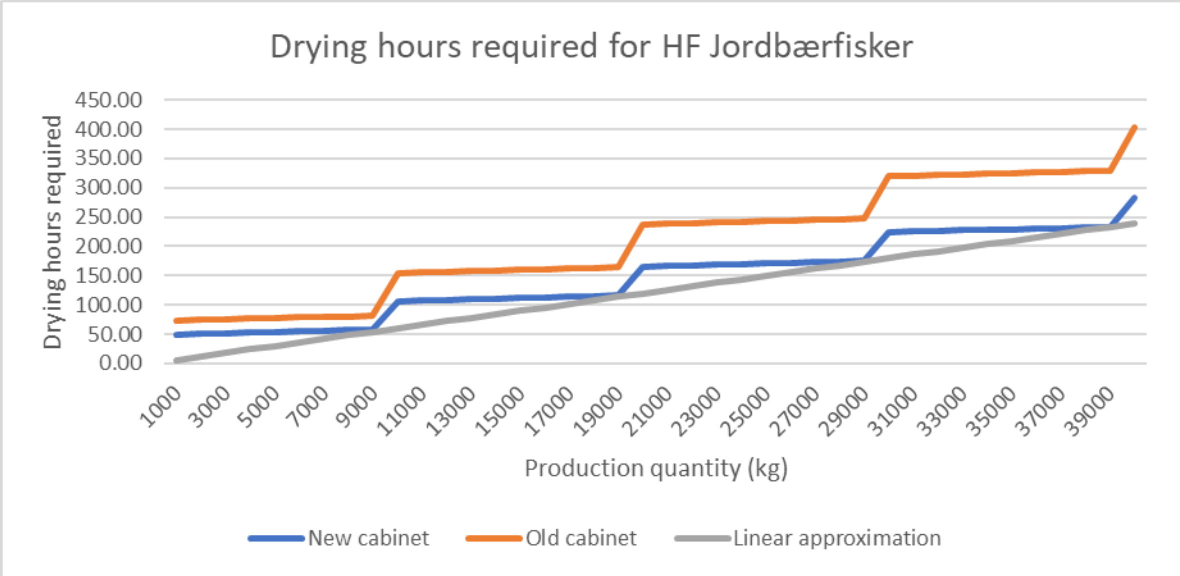

Figure 4.4 The actual drying hours and the linear approximation vs the lot size.

Backwards drying feasibility routine

This procedure fixes possible infeasibilities in the schedule that occurred because a linear approximation of the capacity constraints of the drying cabinets was used to create this schedule.

The procedure starts at the last week of the planning horizon and checks the feasibility of the production in each week through the method described in Section 4.3.1. If the production in a certain week turns out to be infeasible, the procedure calculate the surplus of all intermediates produced in that week from the next full drying cabinet. For example: If the production of a certain intermediate in a certain week is enough to fill 3.015 drying cabinet, the surplus of this intermediate is 0.15.

It then identifies the intermediate with the lowest surplus and tries to move this surplus to an earlier week. It firs tries to move the surplus to week t-1 and then backwards to week 1. When it is not possible to move this surplus to an earlier week, it is instead regarded as lost sales. Moving the surplus reduces the number of times a cabinet is needed in the current week by one. After each move the procedure checks the feasibility again. The procedure repeats this loop until the production in this week is feasible, after which it continues with the previous week. Once the procedure has looped over all weeks, the resulting schedule is feasible. Pseudo code for this procedure can be found in Box 4.3.

```
Backwards_Drying_Feasiblity_routine:

For t = T to 1{
        While (f(Q_t) = false) {
                Find the intermediate k in week t with the lowest surplus

                For w = t-1 to 1 {
                        If (Q_kw > 0) {
                                Increase the lot size with the minimum of the unallocated surplus
                                and the residual capacity in week w
                        } Else If (Q_kw = 0) {
                                Create a lot with size the minimum of the unallocated surplus and
                                the residual capacity in week w
                        }
                }
        }
}
```

Box 4.3 Pseudo code for the backwards drying feasibility routine.

4.4 Other modifications

In the previous section we described how we modified the heuristic proposed by Boctor & Poulin to work with the capacity constraints of the drying cabinets. Although this is the major modification, there are some additional modifications necessary to the original heuristic. These minor modifications are described in this section.

4.4.1 Deciding on night shifts

In addition to deciding on the production quantities in each week, creating an MPS for Brynild's confectionery production also involves deciding in which weeks to employ night shifts. Since the Boctor & Poulin heuristic does not involve decisions on whether to use additional capacity, we need another way to make this decision.

Our solution is the following: We decide beforehand in which weeks we use night shifts and fixate this decision. Once the capacity in each week is fixed we can run the modified Boctor & Poulin heuristic.

This creates a new problem because now we need to decide beforehand in which weeks to use night shifts. To make this decision we propose a heuristic based on the following two principles:

1) To reduce labour costs we want to use night shifts in the minimum number of weeks that is required to meet demand. We calculate the minimum number of night shift by calculating the accumulated demand in terms of moulding time for each week.
2) We want to use this minimum number of weeks with night shifts as late as possible in the planning horizon, while still on time to meet demand, to reduce holding costs.

Night shift heuristic

The idea of this heuristic is to calculate the accumulated demand in terms of moulding time for all weeks and use this to determine the minimum number of weeks in which night shifts must be enabled before a certain week to meet all demand. This results in a set of constraints of the form:

To meet all demand up to week t, at least x weeks up to week t must have night shifts.

These constraints still allow some degree of freedom regarding which exact weeks have night shifts. Within this limited freedom we want to have the night shifts as late as possible in the planning horizon. The idea behind this approach is that production is higher in weeks with nights shifts and that holding costs can be reduced by planning this increased production as late as possible

Since the setups of the moulding machine also consume capacity and no production schedule has yet been made, the exact demand for moulding capacity is unknown. Therefore we use a lot-for-lot schedule to estimate this demand. Furthermore it is possible that the required number of weeks with night shifts before a certain week is higher than the number of weeks where night shifts are possible. In that case the maximum number of night shifts is planned before this week is planned. Box 4.4 shows the pseudo code for this heuristic.

Night_shift_heuristic:

Create a lot-for-lot schedule and calculate the total moulding time for each week:

$$Q_{kt} = d_{kt} \quad \forall k \in K \quad \forall t \in T$$

$$Y_{kt} = \begin{cases} 1 \; if \; Q_{kt} > 0 \\ 0 \; if \; Q_{kt} = 0 \end{cases} \quad \forall k \in K \quad \forall t \in T$$

$$moulding\ time(t) = st \cdot \sum_{k \in K} Y_{kt} + \sum_{k \in K} \frac{Q_{kt}}{p_k}$$

Calculate the accumulated moulding time and from there the minimum number of accumulated night shifts weeks required for each week:

$$acc.moulding\ time(t) = \sum_{w=1}^{t} moulding\ time(w)$$

$$acc.req.night\ shift\ weeks(t) = \left\lceil \frac{(acc.moulding\ time(t) - \sum_{w=1}^{t} a_t)}{ns} \right\rceil$$

While (max(*acc. req. night shift weeks*) > 0) {

    Find the t that maximizes *acc. req. night shift weeks(t)* and plan a night shift in the first possible week, starting at week t and moving backwards to week 1

    If (impossible to add night shift in week t or earlier){

        $acc.req.night\ shift\ weeks(t) = acc.req.night\ shift\ weeks(t) - 1$

    } Else {

        $acc.req.night\ shift\ weeks(w) = acc.req.night\ shift\ weeks(w) - 1 \quad \forall w \geq t$

    }

}

Box 4.4 Pseudo code for Procedure 1.

4.4.2 Lost sales

The Boctor & Poulin heuristic tries to allocate all the demand, inexplicitly assuming that this is possible. However, this might not be the case in some of our scenarios. Demand that cannot be fulfilled is regarded as lost sales.

We want to minimize the lost sales and the costs associated with it. Therefore we propose a straightforward greedy algorithm that can be run after the Boctor & Poulin heuristic. This algorithm is reduces lost sales by filling the gaps between production and capacity after the whole schedule is created.

4.5 Overview of the proposed planning method

In this chapter we described the various parts of our proposed planning method. We end this chapter with an overview of how the various parts work together to create an MPS for Brynild's confectionery production. Figure 4.4 shows this overview.

The night shift heuristic is run first, which produces a decision in which weeks to use night shifts. Once this is decided, the capacity in each week is fixed and the modified Boctor & Poulin's heuristic can be used to create an approximately feasible MPS. This MPS is only approximately feasible because it uses a linear approximation for the capacity constraints of the drying cabinets.

This approximate feasible MPS is than given to the backwards drying feasibility routine, which fixes the small infeasibilities resulting from the approximation. The result is a feasible MPS. This MPS is than given to a greedy algorithm that aims to reduce the lost sales by filling unused capacity. The final result is a feasible MPS with fewer lost sales.

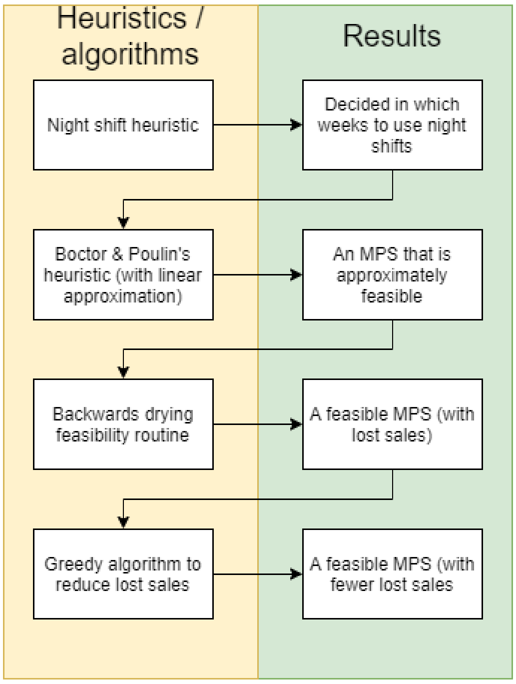

Figure 4.4 Overview of the various parts of the proposed planning method.

5. Evaluation

In this chapter we answer research question 4: *How do the candidate solutions perform compared to the current method used at Brynild?* To do this we first determine values for all parameters required by the candidate method in Section 5.1. In Section 5.2 we compare the schedules created by the candidate method with those created by the current method. In Section 5.3 we benchmark the performance of the candidate method against the best solutions found by an MIP-solver for smaller instances. Section 5.4 presents the conclusions of our evaluation.

5.1 Parameter values

The planning method we proposed in Chapter 4 uses a number of decision variables and parameters. The decision variables are used to represent the solution, while the parameters provide information to the planning method. In this section we list the decision variables and parameters that are used in our planning method and provide values for the parameters.

Changeover costs (c_{change})

The changeover costs are incurred at the moulding machine and consists of the value of the raw material that is thrown away when the pipe between the cooking and moulding machine is cleaned between the production of different intermediates. A previous data collection initiative at Brynild has calculated the average amount of raw material that is thrown away and the average value of this raw material. The average costs of a changeover can therefore simple be calculated: $c_{change} = 1{,}301.8\ NOK$

Additional costs of running a 3-shift system (c_{night})

The bulk of the additional costs of running a 3-shift system are the salary costs of the operators, other additional costs are considered insignificant and not included in the model. The salary of an operator varies depending on the education he received and the years of employment. Brynild estimates the total costs of an operator to be 2850 NOK for a normal shift. However, night shifts pay 33% extra.

The Støperi Sukker line needs 3 operators and therefore the additional costs of running a 3-shift system at this line are $2850\ NOK \cdot 1.33 \cdot 5\ shifts \cdot 3\ operators = 56857.5\ NOK\ per\ week$. So: $c_{night} = 56857.5$.

Costs of lost sales (cu_k)

Only a small fraction Brynild's customers allow backlogging. To simplify the model we therefore assume all demand that cannot be met to be lost. We estimate the costs of lost sales of an end product by multiplying the costs-of-goods-sold (COGS) for that product with an estimated profit margin that is the same for all products. This results in the following formula:

$$costs\ of\ lost\ sales\ for\ product\ x\ per\ kilo = COGS_x \cdot profit\ margin$$

Our model however, does not include the end products. It only includes the intermediates produced at the Stoperi Sukker line. Therefore we need to transfer the costs of lost sales from the end products to the intermediates. This is straightforward for the intermediates that are only packaged into one product, but for those intermediates that are packaged into multiple products it is unclear from our model in which product they will end up. For those intermediates we use the weighted average of the COGS of those products.

In Appendix A we provide an example of these calculations for an intermediate that is packaged into multiple end products. These calculations result in a positive value cu_k for each k.

Available capacity in a 2-shift system in week t (a_t)

As described in Section 2.1.1, the moulding machine is available for 9 shifts a week in a 2-shift system. We add the index t to this parameter to be able to take into account the reduced capacity during certain weeks of the year, for example during Christmas or the summer holidays. In all normal weeks $a_t = 9$.

Availability of night shifts in week t (an_t)

When it is not possible to fill all the normal day shifts in a week because of, for example, holidays. It is also not possible to have night shifts in this week. This parameter has value 1 if night shifts are available in week t, otherwise it has value 0. So $an_t = \begin{cases} 1 \; if \; a_t \geq 9 \\ 0 \; if \; a_t < 9 \end{cases} \quad \forall t \in T$

Additional capacity when running a 3-shift system (ns)

As described in Section 2.1.1, running a 3-shift system increases the number of shifts in a week from 9 to 14. We do not add the index t to the parameter for the extra capacity because in practice it is not possible to run a 3-shift system in the weeks where even the 9 shifts of a 2-shift system cannot be filled. So $ns = 5$.

Demand (d_{kt})

Section 2.2.2 describes the MRP data that is the input for the MPS. The MRP data consists of the demand on all production and packaging lines. The input of our model consists of the demand at only the Støperi Sukker line. Therefore we filter the MRP data and convert it to a demand matrix d of dimensions $K \times T$ where the element d_{kt} is the demand for intermediate k in week t. Figure 5.1 shows part of such a demand matrix for illustration.

Material	Materialkorttekst	2019-50	2019-51	2019-52	2020-1	2020-2	2020-3	2020-4
113717	HF Barnetime skumgelé	0	0	0	0	0	0	0
113680	HF Dent Bærmiks	900	0	0	0	0	0	900
111916	HF Dent Eukalyptus	0	0	0	4800	0	0	0
113723	HF Dent Oi Cherry	0	100	0	0	0	0	0
112812	HF Dent Oi Fuzz	0	0	0	0	0	0	0
111918	HF Dent Salt Lakris	0	0	0	0	0	400	0
112189	HF Dent Trio	0	0	0	4500	0	0	0
113492	HF Figurskumgele Jul	0	0	0	0	4500	0	0
113482	HF Fruktgele jul	0	0	0	0	0	0	0
105994	HF Gelepynt rød, grønn, gul Freia	0	0	0	0	0	0	0
113541	HF Gompegele	0	0	0	0	0	0	0
107935	HF Jellymen, ekstra tykke	0	0	0	0	0	1000	0
112958	HF Jordbærfisker	0	0	0	0	0	3400	0
113515	HF Lakrisbåter	0	0	0	0	0	22000	0
113771	HF Løsvekt Myke Rakkere	0	0	0	0	0	0	0
113713	HF Løsvekt Sure Rakkere	0	0	0	0	0	3300	1200

Figure 5.1 Part of a demand matrix used in the model.

Holding costs (h_k)

As explained in Section 2.3.2, holding costs are only incurred for the finished products. The finished products are not included in our model. However, the MPS for the packaging lines is created based on the MPS for the production lines, which is part of our model. Therefore we must take these holding costs into account. We can assume that if a certain intermediate is produced a week earlier than that it is due, it is also packaged a week earlier and an extra week of holding costs is incurred.

Under this assumption we can transfer the holding costs incurred by the finished products to the intermediates that are packaged into these products. We calculate the holding costs per ton of finished product per week and transfer these to the intermediates that make up this finished product. This is straightforward for the intermediates that are only packaged into one product, but for those intermediates that are packaged into multiple products it is unclear from our model in which product they will end up. For these intermediates we use a weighted average of the holding costs of these finished products. This is similar to the procedure we used to determine the costs of unfulfilled demand for these intermediates.

Brynild considers two types of costs to make up the holding costs for the finished products. The storage costs and the capital costs. The finished goods warehouse is owned by a third party and charges Brynild 2 NOK per day per pallet for all storage costs. The costs of capital are estimated by Brynild to be around 5% per year. This can be combined with the costs-of-goods-sold (COGS) for each finished product to calculate the capital costs. We can therefore use the following formula for the total holding costs of each finished product:

$$holding\ costs\ for\ product\ k\ per\ kilo\ per\ week = \frac{2\ NOK\ \cdot 365 + (kilo\ per\ pallet)_k \cdot COGS_k \cdot 0.05}{52}$$

In Appendix A we provide an example of the calculation of the holding costs for an intermediate that is packaged into multiple end products. These calculations result in a nonnegative value h_k for each k.

Production rates (p_k, l_k)

The production rates per shift and the capacity per drying cabinet differ between the intermediates. These production rates are known and are provided by Brynild. Based on this data we determine p_k and l_k for each k.

Drying times (n_k, o_k)

The drying times differ between the intermediates and between the new and the old drying cabinets. These times are known and are provided by Brynild. Based on this date we determine n_k and o_k for each k. For the intermediates that can only be dried in the new cabinets we use $o_k = \infty$.

Setup times at the moulding machine (st)

As explained in Section 2.1.1 the setup times at the moulding machine vary between 1 and 2 hours and the exact time of a setup is hard to predict in advance. To ensure a feasible schedule we assume a fixed setup time of 2 hours. In our model we measure the available capacity in shifts. 1 shift is 8 hours long, so $st = 0.25$

5.2 Comparison with the current planning method

We want to know how our candidate method compares with the current planning method used by Brynild. We describe the current planning method in Section 2.2. The clear advantage of the candidate method compared to the current method is the time it takes to generate an MPS. With the current method it takes the planner a day of work to create an MPS. The runtime of the candidate method is around 30 seconds. Of course, data preparation and minor tweaks to this schedule mean the human planner requires more than only the run time. Regardless, the candidate method a huge time saver.

In the remainder of this section we compare both methods by the schedules they generate. We first motivate our choice for the test data and the performance indicators. After this we discuss the results. Finally we conclude this section with the findings of a discussion with the human planner regarding the differences between the schedules.

5.2.1 Test data

In this section we motivate why we use historical data and why we choose the time period from September 2019 until August 2020.

Historical data

The candidate planning method is completely defined in Chapter 4. It is an algorithm that takes a set of parameters and demand data as input and gives a production schedule as output. The algorithm does not involve any randomness, therefore, given the same input data, the output data is always the exact same.

The current planning method however, is based on the process specific knowledge of the human planner. It consists of a set of habits and routines that are not explicitly defined. Since the current method is not explicitly defined, it is not possible for us to create new production schedules from the input data that resembles the schedule the planner would create. The human planner would be able to create these schedules, but he is not available to do so. This limits us to the use of historical schedules that the planner has created in the past.

The chosen time period

Since we rely on historical data, we need to select a time period from the past to be used for our comparison. Brynild produces multiple products with seasonal demand. This seasonality is mainly caused by events that occur annually like Christmas, Easter and summer. We want to know how our method performs in these different situations. To capture this seasonality in our comparison, the time period should have a length of one year. We want to use historical data that is as recent as possible, because this better resembles the current situation than data from longer ago. Therefore we use most recent year of which we have all required data. This is the period from September 2019 until August 2020.

5.2.2 Overwriting schedules

The candidate planning method creates a schedule with a timespan equal to that of the MRP data that is used as its input. However, not the entire schedule is used in practice. As time passes, forecasts change and after a while, a new schedule is made that overwrites the current schedule. Therefore the only part of any schedule that is actually executed is the part until the next schedule is created. The schedule is updated when the planner deems it necessary. The length of the executed part of a schedule varies between 1 and 8 weeks.

Figure 5.2 Illustrates how the schedule is constantly overwritten. It shows that in week 37 of 2019 schedule number 1 was made, which spans the time period until at least week 2 of 2020. In week 45 of 2019 however, a new schedule was created that overwrote the previous schedule. Therefore only the first eight weeks of schedule number 1 were executed.

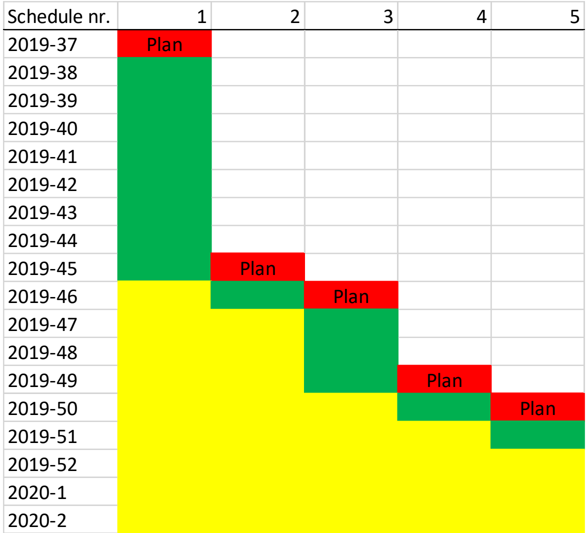

Figure 5.2 A visualization of when schedules are made, in red, and what part of the schedules are actually executed, in green.

In Section 5.2.1 we described why we need to rely on historical data for the comparison of both planning methods. Complete historical production plans are not available, but the historical production quantities are. These historical production quantities are de facto the parts of the historical schedules that were executed in practice (the green parts in Figure 5.2).

5.2.3 Performance indicators

Since we only have data of the parts of the historic schedules that were executed in practice, we can only compare partial schedules of the candidate and the current method. As the planning methods try to minimize the costs of the entire schedule, a fair comparison on the costs of partial schedules is not possible. Instead we measure the performance of our candidate planning method in Section 5.3 and focus here on performance indicators that say something about the general character of the schedules generated by a method with respect to lot size and workload.

1) **The average number of different products per week.** A setup is required before the production of every type of product, hence this indicator also measures the average number of setups per week in each partial schedule.
2) **The average lot size per week in shifts.** The tons of a product that can be produced within one shift vary with more than a factor four. We measure the average lot size in shifts instead of tons to allow for a better comparison in workload.
3) **The average production per week in shifts.** This includes all time that was spend producing products. Setups are therefore not included.

5.2.4 Results of the comparison

Figure 5.3 shows the boxplots of the three indicators. The number of products per week shows a similar distribution between both methods. The average lot sizes have a similar mean, but the candidate method shows less variation in lot sizes. A possible explanation for this is that, in reality, unexpected events like machine failure cause lots to be smaller than their planned size. These smaller lots need to be compensated afterwards by lots that are larger than originally scheduled. This could explain the increased variation.

The average production shows a similar mean, but once again the candidate schedules show less variation. This is a consequence of the larger variation in lot sizes.

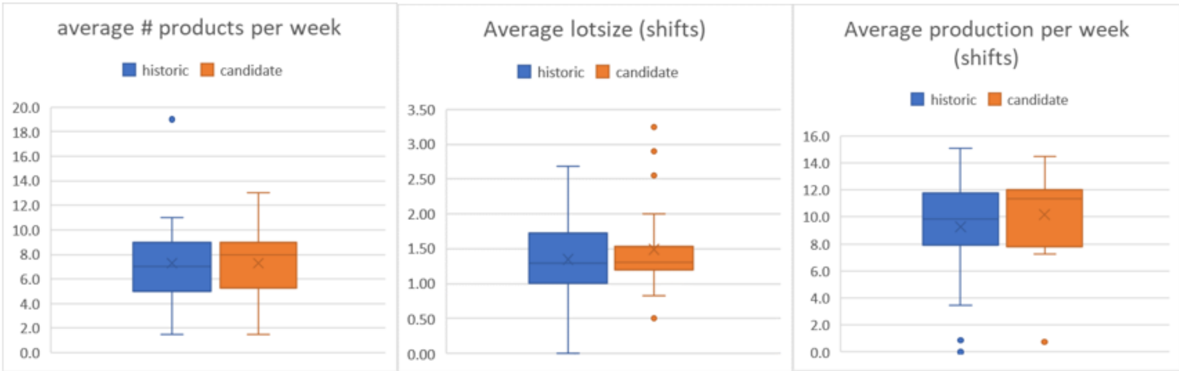

Figure 5.3 Boxplots for the three behavioral indicators.

5.2.5 Discussion with the human planner

In addition to the comparison method discussed in the preceding parts of this chapter, we discussed the schedules created by the candidate planning method with the human planner. Our discussion focussed on the differences between the candidate's schedules and his own. During this conclusion the following three main differences where identified:

- When the planner sees that future capacity is not enough to meet future demand, he tries to maximize the production in the early part of the schedule. The candidate planning method only increases early production to the level required to meet future demand, as to not incur more holding costs than necessary. This can be seen as a trade-off between saving holding costs and preparing for uncertainty. In this case, the candidate method focusses more on saving holding costs, while the human planner focusses more on preparing for uncertainty.
- When demand needs to be produced early because of capacity constraints, one must decide which product should be produced early. The planner bases this decision mainly on which product can be added to the current schedule the easiest, while taking into account the available moulding time and the space in the drying cabinets. The candidate planning method bases this decision mainly on the holding costs. It first tries to fit the products with the lowest holding costs for early production. Both methods result in the early production of some products, but the candidate method has the advantage that it explicitly considers the holding costs in the decision which product is produced early.
- When demand needs to be moved between different weeks, the human planner is inclined to move or merge entire lots. The main reason for this behaviour being that this is the most simple and straightforward way to do so. The candidate planning method prefers to increase existing lots compared to creating new lots since this reduces the total setup costs. To do so the candidate planning method will spread demand over multiple existing lots. Thereby increasing all lots a bit so that future demand is met. This is an advantage of the candidate method, but its effects are reduced by the relative low setup costs, relative to the holding costs, of the production process.

5.3 Performance measurement

In Section 5.2 we compared our candidate method with the current method. This does however not tell us anything about the performance of the candidate method compared to the optimal solution. In this section we compare the performance of the candidate method with that of near-optimal solutions.

5.3.1 Method for performance measurement

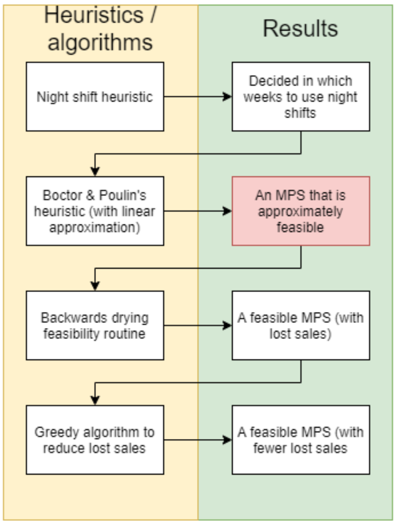

Figure 5.4 Solution overview, with the linear approximated schedule marked in red.

There is no known method for finding optimal solutions for nonlinear or large scale CLSPs. However, up to the point where the approximate feasible schedule is created, see Figure 5.4, our problem is assumed to be linear and can be modelled as an MIP-problem. The MIP-formulation can be found in Appendix B.

The solver can find good solutions to MIP-problems of smaller scale. We can reduce the scale of our problem by reducing the number of weeks in the planning horizon and the number of products. We can then use the candidate method to solve these smaller instances up to the red-marked step in Figure 5.4 and compare this solution with the solution provided by an MIP-solver.

We perform this comparison through the following method. We used the candidate method to find solutions for each of the instances. We put these solutions in Excel and looked for improvement to these solutions with an MIP-solver. We evaluate the quality of a solution through the total costs of the schedule. We measure the performance of the candidate method as the percentage gap between the total costs of the candidate method's solution and the total costs of the best found solution.

There are two main drawbacks to this method of performance measurement. The first one being that the smaller instances may not be representative for the larger instances encountered in reality. The second one being that this method only measures the candidate methods ability to create approximately feasible schedules and not completely feasible schedules. However, given that the backwards drying feasibility routine only moves small amounts of production quantities, this problem is mitigated.

5.3.2 Creating smaller instances

To create smaller instances we reduce the scale of the problem to 5 weeks and 5 products. The 5 products are randomly chosen from the 41 existing products. These 5 products are shown in Figure 5.5. All parameter values are consistent with those provided in Section 5.1. For the sake of simplicity we assume that all 5 weeks in the planning horizon have 9 normal shifts available and that night shifts can be enabled in all 5 weeks.

product_code	product_name	p	h	cu	l	nd	od
104452	HF søte gelehjerter	8500	0.853	0.913013	13162	3	3
112812	HF Dent Oi Fuzz	4000	2.9615	3.191445	7271	4.5	Inf
112958	HF Jordbærfisker	7500	0.7345	1.218397	9754	6	9
111918	HF Dent Salt Lakris	4500	2.6535	2.052817	7271	7.25	Inf
113515	HF Lakrisbåter	8000	0.972517	0.959015	14283	6	9

Figure 5.5 Parameters of the 5 chosen products.

Demand generation

With these 5 products, we generate 10 different scenarios, or demand patterns, to test the performance. Amongst other things, we are interested in our candidate method's ability to make efficient decisions regarding when to use night shifts. Therefore, scenarios in which night shifts are never, or almost always required are deemed less interesting. Since these scenarios leave less room for the method to make decisions.

Without night shifts, there are a total of 45 shifts available in our smaller instance (9 shifts for each of the 5 weeks). We create demand patterns with a total amount such that the production including setups takes around 50 shifts. This way, one or two night shifts will be necessary to meet demand. For each scenario, the total demand is randomly distributed over the 5 weeks and the 5 products. The 10 scenarios can be found in Appendix C.

5.3.3 Results of the performance measurement

		costs per category in NOK				
Scenario	Method	setup	holding	lost sales	night shifts	total
1	Candidate	27338	5509	0	113715	146562
	Solver	27338	5509	0	113715	146562
	Difference	0	0	0	0	0
2	Candidate	22131	67422	0	113715	203267
	Solver	23432	47332	0	113715	184479
	Difference	-1302	20090	0	0	18788
3	Candidate	27338	11018	0	113715	152070
	Solver	28640	7321	0	113715	149676
	Difference	-1302	3696	0	0	2394
4	Candidate	26036	27544	9138	113715	176433
	Solver	27338	27544	0	113715	168597
	Difference	-1302	0	9138	0	7836
5	Candidate	29941	23635	0	113715	167291
	Solver	29941	23635	0	113715	167291
	Difference	0	0	0	0	0
6	Candidate	26036	12395	0	113715	152146
	Solver	27338	5509	0	113715	146562
	Difference	-1302	6886	0	0	5584
7	Candidate	26036	0	20560	170573	217169
	Solver	27338	0	4569	170573	202479
	Difference	-1302	0	15991	0	14690
8	Candidate	23432	49579	0	113715	186726
	Solver	24734	39938	0	113715	178388
	Difference	-1302	9640	0	0	8339
9	Candidate	27338	20658	2284	113715	163995
	Solver	28640	17903	0	113715	160258
	Difference	-1302	2754	2284	0	3737
10	Candidate	26036	27260	18276	113715	185287
	Solver	27338	27260	2284	113715	170598
	Difference	-1302	0	15991	0	14690

Figure 5.6 Comparison of the costs of the solutions found by the candidate and the solver.

Figure 5.6 shows the costs of the solutions found by the candidate method and the solver compare. In scenario 1 and 5, the candidate method and the solver find the exact same solution and therefore the cost differences are zero for these scenarios. In the other scenarios, the solver is able to improve the solution found by the candidate method, resulting in lower total costs. The solver improves the solution in scenarios 2, 3, 6 and 8, by increasing the number of lots to reduce the holding costs. This increases the setup costs, but the total costs are still lower.

In scenarios 4, 7, 9, the solver lowers the total costs by lowering the costs of lost sales. Again, this is achieved by increasing the number of lots, but the total costs are still lower. In scenario 10, the solver reduces both the holding and the lost sales costs. In all scenarios where the solver improves the solution of the candidate method, it does so by increasing the total number of lots. This can be seen in Figure 5.6, as the total setup costs are proportional with the number of lots.

The solver did not find a better night shift pattern than the candidate method for any of the 10 scenarios. This indicates that the candidate's heuristic for deciding on the night shift pattern is near optimal, at least for these small instances.

In Figure 5.7 we measure the performance of the candidate method by comparing the total costs of its solution with those of the solution found by the solver. On average, the solution found by the candidate method is 4% more expensive than the best solution found by the solver. Assuming this performance translates to the larger real world instances, we believe this shows the quality of the schedules created by our method.

Scenario	% of best solution
1	100%
2	110%
3	102%
4	105%
5	100%
6	104%
7	107%
8	105%
9	102%
10	109%
average	**104%**

Figure 5.7 Candidate method's solution as percentage of the best solution.

5.4 Conclusions of the evaluation

In this chapter we evaluated the candidate planning method through two different ways. We compared its schedules with the current planning method and found it to be a huge time saver for the human planner. Through historical data we compared the schedules of both methods and found that the candidate method creates schedules similar to those created by the current method, but much faster.

The comparison with the current method did not provide us with any insights regarding the absolute performance of the candidate method. Therefore, we also measured the performance of our method against an MIP-solver for smaller instances with a linear approximation. We found the solutions provided by the candidate method to be of good quality, as they are on average, 4% more expensive than those found by the solver. A point of attention here is that the best solutions found by the solver often involved a higher number of lots than the solutions found by the candidate method.

6. Conclusion, discussion and recommendations

Section 6.1 states the conclusions of this research. Section 6.2 discusses the limits of this research. Section 6.3 and 6.4 respectively state the recommendation for future research and Brynild.

6.1 Conclusion

The objective of this research is to construct a method for creating a cost-efficient MPS for Brynild's confectionery production process. Presently, the MPS is created manually. This is a time consuming endeavour, with limited room for optimization. For now the current method suffices, but an increase in demand means a more efficient planning is required in the near future.

From a literature review we concluded that the problem of creating a cost-efficient MPS is similar to solving a capacitated-lot-sizing-problem. Combining our insights from this literature review and those of Brynild's confectionery production we found problem-specific heuristics to be the recommended type of solution method for this problem.

We took an existing heuristic that literature has shown to achieve good results for problems similar to ours. We modified this heuristic such that it is applicable to our problem. The biggest hurdle in this process was the set of capacity constraints of the drying cabinets. To make the heuristic applicable, we came up with a partition algorithm that can efficiently check whether a given schedule is feasible. Our method first creates an approximately feasible schedule, after which we use our partition algorithm to solve small infeasibilities.

The clear advantage of the candidate planning method compared to the current planning method is that it is a huge time saver. With the current method it takes the planner a day of work to create an MPS. The runtime of the candidate method is around 30 seconds and requires only some data preparation and minor tweaks by the planner.

We compared the schedules created by our candidate planning method with 32 partial historical schedules. The behaviour of the schedules in terms of number of products per week, lot size and workload per week is similar between both methods.

A discussion with the human planner regarding the main differences between both methods showed the main difference to be that when it comes to producing demand early because of a lack of future capacity, the candidate planning method does just enough to meet demand, while the human planner maximizes the early production as a form of protection against uncertainty. The necessity of this protection is outside the scope of this research, but it is important to be aware of this difference.

To measure the performance of our candidate method we compared its approximately feasible schedules against those of an MIP-solver for smaller instances of 5 weeks and 5 products. We found the solutions of our candidate method to be, on average, 4% more expensive than the best solutions found by the MIP-solver.

6.2 Discussion

The results of the performance measurement of our method are positive, as our method is only 4% away from the best solution found. However, it should be noted that there is no guarantee that this performance translates well to larger instances. Despite this uncertainty, there are reasons to believe that it translates well. Our method is based on the Boctor & Poulin's heuristic and their research shows that the performance of this heuristic increases with the problem size (Boctor & Poulin, 2005).

Two of the five drying cabinets consists of two halves that can be operated separately. The partition algorithm we used to check the feasibility of a set of production quantities with respect to the drying cabinets does not include the possibility to exploit this feature. Extending the partition algorithm to allow this would make it much more complex, while providing a limited amount of flexibility. Therefore we chose not to do so. The impact of this is that in reality there is a bit more flexibility than in our model.

In this research we assumed that machines, raw materials and personnel are always available, while in reality there are bound to be some productivity losses. We did also not consider the possibility for overtime, which is used in reality to make up for these losses. The main thing regarding the gap between this research and reality is that reality is both more uncertain and more flexible.

In this light we want to stress that the planning method proposed in this research is not to be viewed as a replacement for a human planner. Rather it is to be viewed as a support tool that requires the steering of a person to navigate the uncertainty inherit to the real world.

6.3 Recommendations for further research

Varying demand patterns

The performance of our method has been tested for a limited number of small scenarios. To provide insights in the performance of our method in different situations, one could apply the evaluation method used in this research to scenarios that represent those different situations. Some interesting situations in which the performance of our method can be researched are:

- Demand increases beyond capacity and lost sales must be accepted to some extent.
- Demand decreases and efficient production becomes more important.
- Demand becomes more irregular.
- The volume of products relative to total production changes.
- Products with completely different characteristics are introduced (characteristics like production rate, drying time etc).

Robustness of the schedules

Since the planning method proposed in this research is to be used in a rolling-horizon fashion, the variation between different versions of the created schedules is an interesting aspect. When this variation is low, the schedule is called robust. We did not research the robustness of the schedules in this thesis because the required data is not available. However, we recommend the collection of the necessary data and the further research of this aspect of our method, since a more robust planning brings stability to a production process. The data that needs to be collected for this purpose consists of:

- The absolute demand data. Presently only the net demand data (demand minus production) is stored.
- The historical schedules. Presently the schedules are lost, once they are overwritten by new schedules.

6.4 Implementation by Brynild

Implementation

The candidate planning method determines the efficient use of night shifts based on accumulated demand. The model does not include all the labour restrictions that are applicable in reality. Therefore we recommend to view the suggest night shift pattern as an ideal, that should be adjusted to the possibilities of reality by the human planner. Since the night shift pattern is an important part of the input for the actual MPS we recommend the following workflow:

1) The algorithm gives a suggestion when it is efficient to use night shifts.
2) The human planner adjusts this suggestion so that it is feasible with the labour regulations.
3) The adjusted suggestion is fed back to the algorithm.
4) The algorithm gives a suggestion for an efficient MPS.
5) The human planner tweaks the MPS according to his wishes.

References

Aghezzaf, E.-H., Sitompul, C., & Broecke, F. V. (2011). A robust hierarchical production planning for a capacitated two-stage. *Computers & Industrial Engineering*, 361-372.

Billington, P., McClain, J., & Thomas, L. (1983). Mathematical programming approaches to capacityconstrained MRP systems: review, formulation and problem reduction. *Management Science*, 1126-1141.

Boctor, F. F., & Poulin, P. (2005). Heuristics for the N-product, M-stage, economic. *International Journal of Production Research*, 2809-2828.

Broecke, F. V., Landeghem, H. V., & Aghezzaf, E.-H. (2007). An application of cyclical master production scheduling in a multi-stage, multi-product environment. *Production Planning and Control*, 796-809.

Broecke, F. V., Landeghem, H. V., & Aghezzaf, E.-H. (2008). Implementing a near-optimal solution for the multi-stage, multi-product capacitated lot-sizing problem by rolling out a cyclical production plan. *International journal of production economics*, 121-137.

Buschkühl, L., Sahling, F., Helber, S., & Horst, T. (2010). Dynamic capacitated lot-sizing problems:. *OR Specturm*, 231-261.

Clark, A., Bernardo, A.-L., & Almeder, C. (2011). Lot sizing and scheduling: industrial extensions. *International Journal of Production Research*, 2457-2461.

Dillenberger, C., Escudero, L., Wollensak, A., & Zhang, W. (1993). On solving a large-scale resource allocation. *Operations research in production planning and control*, 105-119.

Dixon, P., & Silver, E. (1981). A Heuristic Solution Procedure for the MultiItem, Single-Level, Limited Capacity, LotSizing Problem . *Journal of Operations Management*, 23-39.

Dogramaci, A., Panayiotopoulos, J., & Adam, N. (1981). The dynamic lot-sizing problem for multiple items. *AIIE Trans*, 294-303.

Florian, M., Lenstra, J., & Rinooy Kan, A. (1980). Deterministic production planning and complexity. *Management Science*, 669-679.

Franca, P. M., Armentano, V. A., Berretta, R. E., & Clark, A. R. (1997). A heuristic method for lot-sizing in multi-stage systems. *Computers Operations Research*, 861-874.

Haase, K., & Kimms, A. (1999). Lot sizing and scheduling with sequence-dependent setup costs. *International Journal of Production Economics*, 159-169.

Harjunkoski, J., Maravelias, C. T., Bongers, P., M., C. P., Engell, S., Grossmann, I. E., . . . Wassick, J. (2014). Scope for industrial applications of production scheduling models and. *Computers and Chemical Engineering*, 161-193.

Heerkens, H., & Van Winden, A. (2012). *Geen probleem.* Nieuwegein: Business School Nederland.

Hein, F., Almeder, C., Figueira, G., & Almada-Lobo, B. (2018). Designing new heuristics for the capacitated lot sizing problem by genetic programming. *Computers and Operations Research*, 1-14.

Herrera, C., & Thomas, A. (2009). Simulation of less master production schedule nervousness model. *IFAC Proceedings Volumes*, 1585-1590.

Hungh, Y.-F., Chen, C.-P., Shih, C.-C., & Hung, M.-H. (2003). Using tabu search with ranking candidate list to solve. *Computers and Industrial Engineering*, 615-634.

Jans, R., & Degraeve, Z. (2007). Meta-heuristics for dynamic lot sizing: A review. *European Journal of Operational Research*, 1855-1875.

Kirkpatrick, S., Gelatt, C., & M.P., V. (1983). Optimization by Simulated Annealing. *Science*, 671-680.

Koken, P., Seok, H., & Yoon, S. W. (2017). A simulated annealing algorithm with neighbourhood list for capacitated dynamic lot-sizing problem with returns and hybrid products. *International Journal of Computer Integrated Manufacturing*, 1362-3052.

Olhager, J. (2010). The role of the customer order decoupling point in production andsupply chain management. *Computers in Industry*, 863-868.

Özdamar, L., & Bozyel, M. A. (2007). The capacitated lot sizing problem with overtime decisions and setup times. *IIE Transactions*, 1043-1057.

Silver, E., & Meal, H. (1973). A heuristic for selecting lot size quantities for the case of a deterministic time-varying demand rate and discrete opportunities for replenishment. *Production Inventory Management*, 64-74.

Sürie, C., & Stadtler, H. (2003). The capacitated lot-sizing problem with linked lot sizes. *Management Science*, 1039-1054.

Xie, J., & Dong, J. (2002). Heuristic Genetic Algorithms for General Capacitated Lot-Sizing Problems. *Computers and Mathematics with Applications*, 263-276.

Zijm, W. (2000). Towards intelligent manufacturing planning. *OR Spektrum*, 313-345.

Appendix A: Calculating cu_k and h_k

In this appendix we show how we calculate the value of the lost sales parameter cu_k and the holding costs parameter h_k for the intermediate called Korpus Knatter (code = 112815). The data used for this calculation is from the time span September to December 2019.

Lost sales parameter (cu_k)

The intermediate Korpus Knatter is packaged into 5 different end products. Table A.1 shows the data relevant for the calculation of $cu_{Korpus\ Knatter}$ for these 5 end products. The numbers in the last column that show the profits per kilo are calculated by assuming a 5 percent profit margin over the COGS.

product code	product name	production quantity (kg)	COGS/kg (NOK)	profit/kg (NOK)
113844	Brynild Knatter Frukt 80g 15st	5068	29.39	1.47
113843	Brynild Knatter Skogsbær 80g 1	4435	29.13	1.46
111843	Brynild LV Bærmix 2,00kg	36533	18.27	0.91
113300	Brynild Frukttoppar 140g 15stk	3024	27.06	1.35
104364	Brynild LV Skogsbærmix 2,00kg	72701	19.43	0.97

Table A.1 Required data of the 5 products that Korpus Knatter is packaged into.

The value of $cu_{Korpus\ Knatter}$ can now be calculated by taking the weighted average of the last column of Table A.1 the weights are the production quantities in the third column. Therefore:

$$cu_{Korpus\ Knatter} = \frac{5068 \cdot 1.47 + 4435 \cdot 1.46 + 36533 \cdot 0.91 + 3024 \cdot 1.35 + 72701 \cdot 0.97}{5068 + 4435 + 36533 + 3024 + 72701} = 1.00\ NOK/kg$$

Holding costs parameter (h_k)

product code	product name	production quantity (kg)	COGS / kg (NOK)	kg / pallet	value / pallet (NOK)	storage costs / pallet / year (NOK)	captial costs / pallet / year (NOK)	holding costs / pallet / year (NOK)	holding costs / kilo / week (NOK)
113844	Brynild Knatter Frukt 80g 15st	5068	29.39	210.6	6189.53	730.00	309.48	1039.48	0.09
113843	Brynild Knatter Skogsbær 80g 1	4435	29.13	210.6	6134.78	730.00	306.74	1036.74	0.09
111843	Brynild LV Bærmix 2,00kg	36533	18.27	272	4969.44	730.00	248.47	978.47	0.07
113300	Brynild Frukttoppar 140g 15stk	3024	27.06	136.8	3701.81	730.00	185.09	915.09	0.13
104364	Brynild LV Skogsbærmix 2,00kg	72701	19.43	300	5829.00	730.00	291.45	1021.45	0.07

Table A.2 Required holding costs data of the 5 products that Korpus Knatter is packaged into.

Table A.2 shows the data relevant for the calculation of $h_{Korpus\ Knatter}$ for the 5 end products that Korpus Knatter is packaged into. The holding costs in the 9th column are the sum of the storage and the capital costs in the 7th and 8th column. The value of $h_{Korpus\ Knatter}$ can now be calculated by taking the weighted average of the last column of Table A.2. The weights are again the production quantities in the third column. Therefore:

$$h_{Korpus\ Knatter} = \frac{5068 \cdot 0.09 + 4435 \cdot 0.09 + 36533 \cdot 0.07 + 3024 \cdot 0.13 + 72701 \cdot 0.7}{5068 + 4435 + 36533 + 3024 + 72701} = 0.07\ NOK/kg$$

Appendix B

The MIP-formulation of the smaller instances used for the performance measurement in Section 5.3.

Sets:

T set of all weeks in the planning horizon, weeks are numbered 1 to T, $t \in T$
K set of all intermediates, $k \in K$

Decision variables:

Q_{kt} production quantity of intermediate *k* at in week t in kilos
Y_{kt} is 1 if intermediate *k* is produced in week *t*, 0 otherwise
O_t is 1 if night shifts are enabled in week *t*, 0 otherwise
I_{kt} inventory level of intermediate k at the end of week *t* in kilos
U_{kt} lost sales of intermediate *k* in week *t* in kilos

Parameters:

c_{change} costs of 1 changeover
c_{night} additional costs of running a 3-shift system for 1 week
cu_k costs of a kilo of lost sales of intermediate *k*
a_t number of shifts available in week t in 2-shift system
an_t this parameter has value 1 if night shifts are available in week *t*, 0 otherwise
ns extra shifts available if a 3-shift system is used
d_{kt} demand for intermediate *k* in week *t* in kilos
h_k holding costs of 1 kilo of intermediate *k* for 1 week
p_k kilo of intermediate *k* that can be produced during a full shift
l_k the amount of intermediate *k* that fits into 1 drying cabinet
nd_k the drying time of intermediate *k* in a new drying cabinet in shifts (1 shift = 8 hours)
od_k the drying time of intermediate *k* in an old drying cabinet in shifts (1 shift = 8 hours)
st setup time in shifts

Objective function:

$$\min z = c_{change} \cdot \sum_T \sum_K Y_{kt} + \sum_T \sum_K h_k \cdot I_{kt} + c_{night} \cdot \sum_T O_t + \sum_T \sum_K cu_k \cdot U_{kt} \qquad (1)$$

Subject to:

$$I_{k,t-1} + Q_{k,t} - d_{kt} + U_{kt} = I_{kt} \quad \forall k \in K,\ \forall t \in T \qquad (2)$$

$$\sum_K \left(st \cdot Y_{kt} + \frac{1}{p_k} \cdot Q_{kt} \right) \leq 9 + 5 \cdot O_t \quad \forall t \in T \qquad (3)$$

$$\sum_K \frac{Q_{kt}}{p_k} + nd_k \cdot \frac{Q_{kt}}{l_k} \leq 105 \quad \forall t \in T \qquad (4)$$

$$Q_{kt} \leq M \cdot Y_{kt} \qquad (5)$$

$$Q_{kt}, I_{kt}, U_{kt} \geq 0 \quad \forall k \in K,\ \forall t \in T \qquad (6)$$

$$I_{k0} = 0 \quad \forall k \in K \qquad (7)$$

$$Y_{kt}, O_t \in \{0,1\} \quad \forall k \in K,\ \forall t \in T \qquad (8)$$

1) The objective function minimizes the sum of the costs affected by the MPS. These costs consists of the changeover costs (in red), the holding costs (in yellow), the additional costs of running a 3-shift system (in green) and the costs of not fulfilling demand (in blue).
2) Ensures that the holding level of next week is equal to that of this week plus the production of this week minus the demand for this week. Together with constraints (6), that state that

the holding levels must be nonnegative, this constraint forces the shortage variable U_{kt} to increase to fill the gap between production and demand.

3) Capacity constraints for the moulding machine. Both setups and the actual moulding use capacity. Available capacity depends on whether a 2 or 3-shift system is used.
4) Linear approximation of the capacity constraints of the drying cabinets
5) Ensures that setups are included at the moulding machine.
6) Ensures that production quantities and holding levels are nonnegative.
7) Assume that the starting inventory for each product is zero. If this is not the case, the demand can be adjusted accordingly to meet this assumption.
8) Ensures that the binary decision variables are binary.

Appendix C

The 10 different scenarios used for the performance measurement in Section 5.3 are shown in Figure C.1.

1	product_code	product_name	2020-7	2020-8	2020-9	2020-10	2020-11
	104452	HF søte gelehjerter	8500	25500	34000	8500	0
	112812	HF Dent Oi Fuzz	12000	8000	4000	0	16000
	112958	HF Jordbærfisker	15000	15000	7500	22500	7500
	111918	HF Dent Salt Lakris	4500	4500	9000	9000	13500
	113515	HF Lakrisbåter	16000	32000	0	8000	8000
2	product_code	product_name	2020-7	2020-8	2020-9	2020-10	2020-11
	104452	HF søte gelehjerter	0	8500	8500	25500	25500
	112812	HF Dent Oi Fuzz	8000	24000	16000	4000	4000
	112958	HF Jordbærfisker	0	7500	7500	0	7500
	111918	HF Dent Salt Lakris	22500	13500	0	13500	4500
	113515	HF Lakrisbåter	0	32000	0	16000	16000
3	product_code	product_name	2020-7	2020-8	2020-9	2020-10	2020-11
	104452	HF søte gelehjerter	17000	8500	25500	17000	25500
	112812	HF Dent Oi Fuzz	12000	12000	8000	4000	0
	112958	HF Jordbærfisker	7500	0	22500	0	7500
	111918	HF Dent Salt Lakris	4500	9000	4500	13500	9000
	113515	HF Lakrisbåter	40000	8000	8000	8000	24000
4	product_code	product_name	2020-7	2020-8	2020-9	2020-10	2020-11
	104452	HF søte gelehjerter	25500	17000	25500	8500	17000
	112812	HF Dent Oi Fuzz	0	0	8000	12000	0
	112958	HF Jordbærfisker	0	7500	30000	30000	0
	111918	HF Dent Salt Lakris	4500	9000	9000	4500	13500
	113515	HF Lakrisbåter	16000	16000	16000	24000	16000
5	product_code	product_name	2020-7	2020-8	2020-9	2020-10	2020-11
	104452	HF søte gelehjerter	8500	17000	25500	17000	8500
	112812	HF Dent Oi Fuzz	4000	12000	12000	12000	16000
	112958	HF Jordbærfisker	0	7500	22500	7500	15000
	111918	HF Dent Salt Lakris	9000	4500	4500	4500	9000
	113515	HF Lakrisbåter	8000	16000	8000	24000	8000
6	product_code	product_name	2020-7	2020-8	2020-9	2020-10	2020-11
	104452	HF søte gelehjerter	8500	17000	8500	17000	17000
	112812	HF Dent Oi Fuzz	4000	16000	0	4000	16000
	112958	HF Jordbærfisker	45000	0	30000	7500	15000
	111918	HF Dent Salt Lakris	22500	4500	4500	4500	4500
	113515	HF Lakrisbåter	0	24000	8000	8000	0
7	product_code	product_name	2020-7	2020-8	2020-9	2020-10	2020-11
	104452	HF søte gelehjerter	34000	42500	17000	8500	0
	112812	HF Dent Oi Fuzz	12000	12000	8000	8000	4000
	112958	HF Jordbærfisker	15000	15000	15000	0	7500
	111918	HF Dent Salt Lakris	4500	4500	9000	4500	0
	113515	HF Lakrisbåter	24000	16000	24000	0	16000
8	product_code	product_name	2020-7	2020-8	2020-9	2020-10	2020-11
	104452	HF søte gelehjerter	17000	8500	17000	0	0
	112812	HF Dent Oi Fuzz	4000	16000	4000	8000	8000
	112958	HF Jordbærfisker	22500	22500	22500	22500	15000
	111918	HF Dent Salt Lakris	4500	22500	9000	0	0
	113515	HF Lakrisbåter	0	8000	24000	24000	8000
9	product_code	product_name	2020-7	2020-8	2020-9	2020-10	2020-11
	104452	HF søte gelehjerter	17000	25500	25500	0	8500
	112812	HF Dent Oi Fuzz	4000	4000	0	8000	8000
	112958	HF Jordbærfisker	22500	30000	7500	7500	37500
	111918	HF Dent Salt Lakris	9000	9000	4500	4500	13500
	113515	HF Lakrisbåter	8000	8000	16000	24000	0
10	product_code	product_name	2020-7	2020-8	2020-9	2020-10	2020-11
	104452	HF søte gelehjerter	0	25500	17000	8500	8500
	112812	HF Dent Oi Fuzz	12000	20000	16000	8000	8000
	112958	HF Jordbærfisker	0	7500	15000	22500	0
	111918	HF Dent Salt Lakris	0	4500	9000	9000	4500
	113515	HF Lakrisbåter	8000	16000	24000	8000	24000

Figure C.1 The 10 different scenarios.