K. Lekkerkerk

Scheduling optimization in a refinery for vegetable oils

TU Delft

THIS PAGE IS LEFT BLANK INTENTIONALLY.

Scheduling optimization in a refinery for vegetable oils

By

K. Lekkerkerk

Master Thesis

In partial fulfilment of the requirements for the degree of:

Master of Science
in Mechanical Engineering

At the Department Maritime and Transport Technology of Faculty Mechanical, Maritime and Materials Engineering of Delft University of Technology.
To be defended publicly on Thursday November 26th, 2020 at 03:00 PM

Student number:	4304144	
MSc track:	Transport Engineering and Logistics	
Report number:	2020.MME.8460	
Supervisor:	Dr. F. Schulte	
Thesis committee:	Dr. ir. D. L. Schott,	TU Delft Committee Chair, Faculty Mechanical, Maritime and Materials Engineering
	Dr. F. Schulte,	TU Delft Committee member, Faculty Mechanical, Maritime and Materials Engineering
	Dr. H. S. Seyffert,	TU Delft Committee member, Faculty Mechanical, Maritime and Materials Engineering
	Drs. J. Vernooy,	Company Supervisor, Sime Darby Oils Zwijndrecht Refinery
Date:	November 26th, 2020	

TU Delft

THIS PAGE IS LEFT BLANK INTENTIONALLY.

Preface

In the period of October 2019 until November 2020, I have been working on my graduation assignment for the Mechanical Engineering master of Transport Engineering and Logistics at the Delft University of Technology. The assignment has been performed in cooperation with Sime Darby Oils Zwijndrecht Refinery as a part of their bigger goal to improve the competitiveness of sustainable and certified vegetable oils. Analysis of Sime Darby Oils Zwijndrecht Refinery showed that an improved production scheduling can contribute to that. Therefore, this report focuses on the development of a scheduling optimization program to apply in a refinery for vegetable oils.

Readers of this report that are interested in the refining processes in the refinery of Sime Darby Oils Zwijndrecht Refinery, the equipment used for that or the current scheduling procedure at Sime Darby Oils Zwijndrecht Refinery are referred to Chapter 2. If the readers are more interested in the state-of-the-art in production scheduling solutions or the selection method for the basic scheduling optimization model used for the development of the final mathematical model, they are referred to Chapter 3. The elaboration of this final mathematical model is given in Chapter 4. Readers that are interested in the building of the mathematical model for Sime Darby Oils Zwijndrecht Refinery and its validation are referred to Chapter 5. This chapter also performs further investigation to some of the aspects discussed in the validation by means of an experimental study. The final chapter, Chapter 6, will discuss the conclusions, the contributions of this research to the company, the academic contributions and recommendations for future research.

I would like to conclude by thanking my supervisors from Sime Darby Oils Zwijndrecht Refinery, at first Wijnand van der Tempel and later on Jons Vernooy, for sharing their knowledge, familiarizing me with the company and their support. I would like to thank Dr. F. Schulte as my daily supervisor from the TU Delft and Dr. Ir. D. Schott as my TU Delft Committee Chair for their optimism, support and very useful feedback. I would also like to thank the operators of the refinery, the schedulers, the team members of the Transformation Office and all other employees of Sime Darby Oils Zwijndrecht Refinery who gave me very useful input and helped me to bring this graduation assignment to a good end.
Finally, I would like to thank my family and friends for their unconditional faith in my abilities to finalize my graduation assignment.

K. Lekkerkerk
Delft, November 2020

THIS PAGE IS LEFT BLANK INTENTIONALLY.

Summary

This thesis has been performed in cooperation with Sime Darby Oils Zwijndrecht Refinery (SDOZR), a company that refines palm oil, palm kernel oil, soy bean oil, sunflower oil, coconut oil and rapeseed oil. In the years to come, the world population will grow. It is the expectation of the United Nations that the population will grow from over 7.6 billion in 2018 to almost 9.8 billion people in 2050 (United Nations, Department of Economic and Social Affairs, Population Division, 2019). A big concern is how to feed all these people. Besides the growth of the world population, also the growth in welfare contributes to the demand for a higher food production (van Kasteren, 2013). Companies that process vegetable oils - such at SDOZR - for among others food and care products experience this growth in world population already or in the near future by an increasing demand for their products. At some point these companies may have trouble with continuing meeting the demand of their customers and need to expand their capacity. In the case of SDOZR it is expected that a change of scheduling tactics might be sufficient to unlock at least a part of the newly required capacity. The main research question of this paper to answer is therefore: "Is there a way to optimize the current way of scheduling in order to improve the performance part of the OEE formula?"

In order to answer this main research question, a research has started to the current situation at SDOZR concerning the production processes, the equipment used for these processes and the current way of scheduling. From this investigation, restrictions came forward that should be considered when the production schedule is created. With this information, it has been investigated in literature if there is a mathematical model that suits these restrictions best. After a selection procedure where both the scheduling optimization problem of SDOZR and the literature works were subjected to a classification for batch processes, scores were given when a particular restriction from the SDOZR situation was met by a particular literature work. Based on this selection procedure, the mathematical model discussed by the work of Kondili et al. (1993) named "A General Algorithm for Short - Term Scheduling of Batch Operations - I. MILP Formulation" was used as the basic theoretical model. Since the model did not meet all the restrictions given by SDOZR, adaptions, modifications and additions have been made to the mathematical model. After the development of the mathematical model, a simplified version of the scheduling optimization model has been implemented in the final mathematical model. A validation has been performed to compare the functioning of this mathematical model compared to the current way and results of scheduling. By means of a sensitivity analysis, some findings done during validation were further investigated.

The conclusion to the main research question is that there is a way to optimize the current way of scheduling in order to improve the performance part of the OEE formula. The mathematical model developed during this research is not (yet) able to replace the scheduling software currently used by SDOZR, but it definitely shows potential. The mathematical model can be seen as a basis for further development to make it suitable to replace the scheduling software currently used. This not only holds for the mathematical model itself, but also for the approach that has been used to come to this point. Recommendations for future research following from this thesis include a more thorough research to disabilities of the mathematical model developed in this paper that were addressed during the validation of the mathematical model. For this, help from SDOZR is required in the collection of the required data.

THIS PAGE IS LEFT BLANK INTENTIONALLY.

Contents

1

Introduction

1.1. Motivation for this report

In the years to come, the world population will grow. It is the expectation of the United Nations that the population will grow from over 7.6 billion in 2018 to almost 9.8 billion people in 2050 (United Nations, Department of Economic and Social Affairs, Population Division, 2019). A big concern is how to feed all these people. Besides the growth of the world population, also the growth in welfare contributes to the demand for a higher food production (van Kasteren, 2013). An ingredient processed in many products is palm oil. Palm oil can be processed in for example margarine, different oils for cooking, shampoo, make-up, replacers for animal fat and milk fat and animal nutrition (de Vré, 2011; MVO, n. d.; Sime Darby Plantation Berhad, 2019a, 2019b). When the consumption of these products increases, the amount of palm oil required to produce the products will also increase. However, palm oil is a product that has a negative effect on the conscience of a lot of people. This is because, among others, nowadays most of the palm oil plantations are created by (illegally) claiming areas of rain forests (de Vré, 2011; Slingerland, 2016).

However, it is questionable if changing to other sources of vegetable oil will solve this issue. As for example can be seen in Figure 1.1, palm oil has, compared to other plants that produce a vegetable oil, the highest yield per hectare and it acts as a replacement for trans fats. Trans fats were usually present in fats that were industrially partly hardened to improve its shelf-life and create a more solid structure. Since palm oil is used instead of trans fat-containing fats, the intake of trans fats is no longer a public health concern in the European Union (European Palm Oil Alliance, 2017; Voedingscentrum, n. d.). There are also economical reasons for the use of palm oil: by producing palm oil, millions of small farmers in for example Indonesia and Malaysia get the opportunity to create more prosperity for themselves. In Indonesia, these millions of small farmers are responsible for 40% of the overall palm oil production. However, they don't have the knowledge to improve the yield of their plantations, because of which they would like to develop parts of the rain forest into palm oil plantations (Claassen, 2017). There are initiatives to turn this negative trend around. One of these initiatives is the founding of the "Roundtable on Sustainable Palm Oil" (RSPO). The RSPO is a non-profit organization that has set up a number of criteria on the environmental and social aspects of palm oil production. Companies that adhere to these criteria are allowed to sell their palm oil as "Certified Sustainable Palm Oil" (CSPO). The RSPO maintains seven principles that are divided over three impact areas (Roundtable on Sustainable Palm Oil, 2019, 2020):

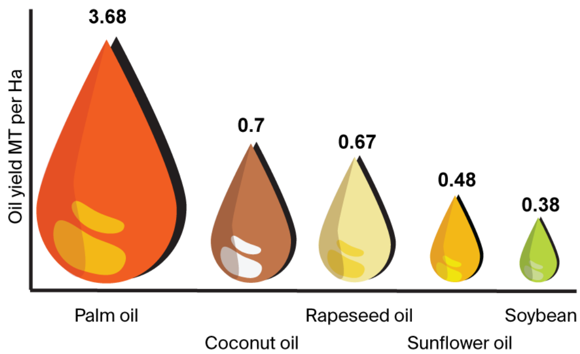

Figure 1.1: Oil yield of major crops in MT per Ha

1. Impact area: Prosperity
    (a) Behave ethically and transparently

   (b) Operate legally and respect rights
   (c) Optimise productivity, efficiency, positive impacts and resilience
2. Impact area: People
   (a) Respect community and human rights and deliver benefits
   (b) Support smallholder inclusion
   (c) Respect workers' rights and conditions
3. Impact area: Planet
   (a) Protect, conserve and enhance ecosystems and the environment

The objective of the first impact area is to improve the prosperity for everybody involved by creating and maintaining a sustainable, competitive and resilient palm oil sector. This will ensure a long-term profitable supply chain and shared benefits that can be used for both the private sector as the livelihoods of communities where oil palm is grown and cultivated. Complying to the principles of this impact area should ensure a strong and healthy relationship between the stakeholders without any violation of laws. It should also motivate the involved parties for a continuous improvement towards a bigger share in sustainable palm oil. (Roundtable on Sustainable Palm Oil, 2020).
In relation to the first impact area, the objective of the second impact area is to take good care of the people involved in palm oil production by respecting and protecting human- and community rights, equality and a healthy working and living environment. People participating in palm oil production processes are able to support themselves and their families in a sustainable way (Roundtable on Sustainable Palm Oil, 2020).
The objective of the last impact area is to protect and restore the ecosystems and their services. This will be achieved by a sustainable consumption and production, controlling air and water pollution, sustainable management of natural resources and converse biodiversity. The result will be a resilient food and fibre production, cleaner water and air and a reuse of carbon for regeneration of soils, for now and in the future (Roundtable on Sustainable Palm Oil, 2020).

One of the companies that is a member of the RSPO organization is Sime Darby Oils (SDO) (Roundtable on Sustainable Palm Oil, 2019; Sime Darby Plantation Berhad, n. d.). SDO is the leader in plantation sustainability, the largest supplier of certified sustainable and traceable palm oil and palm kernel oil and has several factories in the Asian Pacific, Europe, Africa and Papua New Guinea (Sime Darby Plantation Berhad, 2019c n. d.). One of the factories in Europe, Sime Darby Oils Zwijndrecht Refinery (SDOZR), is a factory that, besides palm oil and palm kernel oil, also processes rapeseed oil, sunflower oil, coconut oil and soybean oil (Sime Darby Plantation Berhad, 2019d). From the total of six crude oils more than 350 end products can be made in the same plant. Figure X shows a schematic overview of the plant of SDOZR. It should be noted that not all physical connections between the various departments of the plant are visualised in this figure. For example, the high level of interaction between the equipment of the Chemical Interesterification (C. I.) and Bleaching Line (BL) [...] is not indicated.
[...]
In the plant, various processes are performed to refine the raw oils and create the great variety of end products. In short, the processes do the following. During DF, vegetable oils are separated based on their crystallization properties. This happens by controlled cooling of the oil, whereby the oil that crystallizes last is removed by filtering or centrifuging (Calliauw, 2020). During centrifuging the oils, performed on the Centrifuge Lines (CL's), the oil is split from unwanted particles, whereas the goal of bleaching is to bring the color of the oil within the required specifications (Gibon et al., 2007). In most of the cases, bleaching on one of the BL's is followed by deodorization on one of the Deodorizers (DEO's). This process brings the level of free fatty acids (FFA) within the required specifications. It is also possible to slightly adapt the color of the oil during deodorization (Gibon et al., 2007). During chemical or enzymatic interesterification the FFA's are rearranged by a chemical or enzymatic additive respectively (Gibon et al., 2007; Laning, 1985; Verleyen et al., 2002). The hardening (or hydrogenation) process either converts a liquid oil into a solid fat, changes the consistency of the fat or stabilizes an oil or fat by adding hydrogen under near-vacuum circumstances (Ariaansz, 2020).

SDOZR, as it agrees with a prominent company, wants to keep up with the growth in demand for vegetable oils in a sustainable way. SDOZR would like to do that by improving the competitiveness of sustainable and certified vegetable oils and their share in, among others, the European market. In order to increase the share in sustainable and certified vegetable oils, the competitiveness of SDOZR should increase. However, the variety of oils refined by SDOZR and the fact that from the six crude oils more than 350 end products can be made, results in a high complexity in increasing the production (personal communication). To overcome this, operational improvements in the plant should be made.

1.2. Aim and scope of the report

A key element in the operational improvements in the plant is the Overall Equipment Efficiency (OEE). It is an analysis tool that is used to find potential improvements for a machine or a group of machines in the sense of producing more or better products. The OEE is calculated as a product of the availability of the equipment (A), the performance of the equipment (P) and the quality of the produced volume (Q) (Blom, 2018; The Lean Six Sigma Company, 2019):

$$OEE = A * P * Q * 100\% \tag{1.1}$$

where:

$$A = \frac{\textit{actual operation time}}{\textit{planned operation time}} \tag{1.2}$$

$$P = \frac{\textit{actual amount of produced volume}}{\textit{maximum amount of produced volume}} \tag{1.3}$$

$$Q = \frac{\textit{approved amount of produced volume}}{\textit{total amount of produced volume}} \tag{1.4}$$

Based on the formula, there are different elements that can cause a loss in the OEE. First, loss in availability can be the result of the failure of equipment, setups and adjustments. Next, loss in performance can be caused by idling, minor stops and reduced speed. Finally, quality reduction can be a result of production- and startup rejections (The Lean Six Sigma Company, 2019; Vorne Industries Inc., 2019). Improving these factors will improve the OEE.

An analysis of the company showed that improvements can be made on all three elements of the OEE formula. Based on a stakeholder analysis within the company, this paper will focus on improvements that can be made in the performance part of the OEE formula. The analysis of the company showed that a big part of the current performance is the result of a way of scheduling that does not unlock the full potential of the plant. In this context, the following research questions have been formulated:

Main research question: Is there a way to optimize the current way of scheduling in order to improve the performance part of the OEE formula?

Sub-questions:

1. What are the restrictions for scheduling at SDOZR concerning the production processes, the equipment and the current way of scheduling?
2. What is the state-of-the-art in production scheduling solutions?
3. What optimization model would be appropriate to use?
4. What adaptions or improvements should be made to the mathematical model in order to meet the problem description of SDOZR best?
5. How to build the scheduling optimization model for SDOZR?
6. How does the mathematical model function compared to the current scheduling program?

In order to answer the questions, the whole plant and it's processes should be evaluated. However, for the duration of the graduation assignment, this would be a challenging task. In previous research for SDOZR, the ROS has been considered in the context of the scheduling optimization. To be more precise, a scheduling optimization has been proposed to improve the allocation of the storage tanks in the ROS (Baart, 2020). Because of the size of this assignment and the computational power required, the rest of the plant was not taken into account. This paper will focus on the preceding equipment with respect to the ROS, a part of the plant called the refinery. The refinery includes the CL's, BL's, DEO's

[...] and the C. I., but it are the DEO's with the preceding BL's as shown in Figure X that cover the final processing steps before the end products are transported to the ROS. Because the highest level of interaction occurs between BL [...] and C. I., these two parts of the refinery will form the scope of this report.

Besides the physical scope sketched above, there is also a scope that occurs by making assumptions for the execution of this research. It is assumed that the crude oil that is used to feed the BL's and the C. I. will be present at the moment that it is needed. Also the other raw materials, e. g. the materials required for the BL's and C. I. other than the vegetable oils, will be present at the moment that it is needed. Moreover, it is assumed that the ROS has the required storage capacity at any time. In other words, the demand for the refinery is generated by the ROS.

1.3. Structure of the report

The remainder of this report consists of five chapters. First, the situation at SDOZR will be discussed. Therefore, the processes, equipment and the current way of scheduling will be elaborated in Chapter 2. Based on that information, literature is gathered to describe the state-of-the-art in production scheduling solutions. Together with the selection of the appropriate mathematical model to be used this will be discussed in Chapter 3. The selected mathematical model will be elaborated in Chapter 4, together with any adaptions or improvements to the mathematical model. In Chapter 5 the scheduling problem of SDOZR is implemented in the mathematical model and validated. By means of an experimental study some of the aspects discussed during the validation will be further investigated. Finally, Chapter 6 will address the conclusions, the contributions of this research to the company, the academic contributions and recommendations for future research.

2
The situation at Sime Darby Oils Zwijndrecht Refinery

This chapter will discuss the situation at Sime Darby Oils Zwijndrecht Refinery (SDOZR) in order to get a better and more complete understanding of the scheduling problem of SDOZR. Therefore, the processes that take place in the refinery of SDOZR, the equipment used for these processes and the current way of scheduling will be elaborated. The three elements will be discussed in Section 2.1, Section 2.2 and Section 2.3 respectively.

2.1. Refining vegetable oils in the refinery

During the refining of vegetable oils, several components are removed from the crude oil. The components to be removed include among others free fatty acids (FFA), phosphorus occurring in a variety of forms, traces of metals such as iron and copper and oxidized carotenoids. FFA is a measure for the quality of vegetable oils. A lower level of FFA in the oil corresponds to a better quality of oil. Phosphorus is removed from vegetable oils due to the oil losses it causes during the refining of the oil (Galhardo, F. and Dayton, C., 2020). Metals can effect the flavor and stability of the vegetable oil (Beal, R. E. and Eisenhauer, R. A.). Finally, carotenoids give vegetable oils their color. When they oxidize, rancidity is produced accompanied with off flavours and smells (Miller, M., n. d.).
The refining of vegetable oils can be divided into different steps. What steps to take depends on the type of refining, because a distinction can be made between physical and chemical refining, where FFA is a key element. At what processing step the FFA will be removed determines what type of refining is used. When FFA is removed in the deodorizing unit, one speaks of physical refining. This will be discussed in Section 2.1.1. In the case of chemical refining, which will be discussed in Section 2.1.2, FFA will be removed during the alkali neutralization step, where the oil will also be cleared from gums and where soapstocks are produced. Besides physical and chemical refining, also chemical interesterification happens in the refinery of SDOZR and will be discussed in Section 2.1.3.

2.1.1. Physical refining

Physical refining starts with degumming, followed by bleaching and is finished by deodorization. During degumming, phosphorus in the form of phosphatides are being removed. Depending on the type of phosphatide (hydratable or non-hydratable) and the amount of phosphatide content in the crude oil, degumming can be performed in a variety of ways. Two of the processes take place by mixing water and crude oil or by mixing the crude oil with phosphoric or citric acid at a temperature between 80°C and 90°C. These two processes are called water degumming and wet acid degumming respectively. To the latter, water is added after a prescribed retention time. Then, the oil goes into a centrifugal separation unit. A third process for degumming is called soft degumming. During this process, the oil is heated to a temperature between 75°C and 85°C. Then it is mixed with a water solution that contains a complexing molecule and a wetting compound. The mixture will be sent to a centrifugal separation unit after a prescribed retention time. The last possible process for degumming, called dry degumming, starts with mixing the crude oil with concentrated phosphoric acid. Acid-activated bleaching earth is added under vacuum and at a temperature between 80°C and 120°C after a short retention time. The mixture will then be filtrated when the contact time has been sufficiently long enough (Gibon et al., 2007).

The next step in physical refining contains bleaching. Here, a distinction can be made between adsorptive bleaching and thermal bleaching. During adsorptive bleaching, part of the coloring pigments and some other components of the vegetable oil will be removed by means of Van der Waals surface attraction forces, covalent or ionic bonds to the bleaching earth that will be added to the oil. Thermal bleaching is done during deodorization, where the heating effect takes care of the thermal destruction of pigments.

The process of adsorptive bleaching consists of several steps. To prevent the oil from oxidation and thus deterioration, both the oil and the bleaching earth are deaerated before they are added to the process. The oil will also be dried under a reduced pressure, after which the bleaching earth is directly added to the oil or as a pre-mix of oil with bleaching earth. This happens under controlled temperature, acidity and humidity. Then, the oil is heated further under reduced pressure and it is intensively mixed. To ensure intimate contact between the oil and the bleaching earth, sometimes steam is injected. At last, the bleaching earth is filtered out of the mixture of oil and bleaching earth (Gibon et al., 2007).

The final step in physical refining is deodorizing. During Deodorizing, FFA and oxidized carotenoids are removed. The thermal bleaching process discussed in the previous paragraph is also performed during deodorization. The parameters required to set up a deodorizer, such as temperature, operating pressure and amount of stripping gas, are determined by the type of oil to be refined, but also by the design of the deodorizer. Deodorizers exist for batch, semi-continuous and continuous deodorization processes. The batch deodorizer is mostly used for small capacities and irregular production, or for processing small batches of different oils. For larger batches, semi-continuous deodorizers are often used. In the case of large capacities to be processed and few changes of stock, the continuous deodorizer is used. The general layout of a deodorizer consists of a number of vertically stacked compartments or trays, through which the oil is sent through by gravity (Gibon et al., 2007).

2.1.2. Chemical refining

Chemical refining only differs from physical refining at the beginning of the process. Just as with physical refining, the oil to be refined with the chemical process will be degummed, bleached and deodorized. However, between degumming and bleaching an extra step is added. This is known as alkali neutralization (Gibon et al., 2007; Verleyen et al., 2002). During this process, FFA and non-hydratable phosphatides are removed by mixing caustic soda with the degummed oil at a temperature between 90°C and 95°C. By adding the caustic soda, soapstock will be formed. The mixture of soapstock and oil will be delivered to separators, where the mixture will be centrifuged. By centrifuging, the mixture is separated into a light phase with the neutralized oil that also still contains traces of soaps, free caustic, phosphatides and other soluble impurities. These will be removed by washing with soft water of 90°C and another centrifuging step. The oil is then separated from soapy water. The neutralized oil will be dried, bleached and deodorized, the soapy water will be further processed in a soap splitting unit (Gibon et al., 2007).

2.1.3. Chemical interesterification

Every type of oil has a unique fatty acid composition and the fatty acids are distributed within the triglyceride molecules in a particular way. With chemical interesterification, the orientation of these fatty acids in the triglyceride molecule is rearranged in order to influence some physical and functional characteristics of the oil. By rearranging the fatty acids, the oils can be used in products as margarines, cooking oils, frying oils, salad oils and confectionery fats. Chemical interesterification can be combined with processing techniques as hardening and/or fractionation. The characteristics involved are melting characteristics, crystalline properties, texture and the dropping point (Laning, 1985). During crystallization, a material changes from a solid to a liquid (Editors, B. D., 2018). The dropping point refers to the softness of a fat and is defined as the temperature at which the fat is soft enough that a drop can be formed that is able to fall from the fat mass. This temperature is much higher than the temperature at which the fat is normally used (Laning, 1985).

2.2. Refinery layout

From the three processes described in Section 2.1 and performed in the refinery, degumming is performed on the centrifuge lines. Amounts of material that need to be bleached after degumming are bleached on Bleaching Line (BL) [...], whereas the other materials are bleached on either BL [...]. Deodorizing is performed on either Deodorizer (DEO) [...]. Chemical interesterification (C. I.) is performed on separate cauldrons located near BL [...], where some of the processes of chemical interesterification interfere with the processes of BL [...]. A more detailed description of the latter and the other processing lines are given in the sections below (personal communication).

2.2.1. Centrifuge Line

On Centrifuge Line (CL) [...], Certified Sustainable Palm Kernel Oil (PKcs) and Certified Sustainable Palm Oil (POcs) are processed, mostly in batches [...]. Figure X gives a schematic overview of the equipment.
[...]
From the working tanks, the oil passes a kneader mixer, depending on the oil processed: a residence vessel and again a kneader mixer before the oil ends up in the [...] centrifuge. When palm oil is being processed, [...] residence vessels are used. In the case of liquid oils, one residence vessel is used and when palm kernel oil is processed, no residence vessels are used. The latter is because when palm kernel oil is ordered to stay in a residence vessel, it will start to emulsify. A kneader mixer does not have a residence capacity and could be interpreted as a piece of a pipeline.
When the oil goes to the first kneader mixer, chemicals are added to the oil. After the required residence time in the residence vessel, water and caustic soda will be added when the oil goes to the second kneader mixer. When the oil arrives in the [...] centrifuge, it will be deacidified. In the [...] centrifuge, it will be washed. The fatty acids that are separated from the oil by the centrifuge will go to the soap splitting department. This drainage of fatty acids is a continuous process. When the oil has been washed, it will go to the dryer. After the dryer, the oil will go to Tank Park [...] (TP[...]) or back to the working tanks. The latter only happens when the oil does not have the desired specifications or when there are problems. Any leftovers of water and fatty acids that free up during drying will also be separated and will be sent to the soap splitting department. The residence capacity of the dryer is negligible: it is only about [...] at a maximum. This volume is not needed during the process.

The fatty acids arrive at a soap tank with a mixer first. The mixture of water and fatty acids will then go to the decanter via various kneader mixers, where in the mean time sulphuric acid is added. The fatty acids are separated from other substances by means of an overflow system in the decanter, where the latter is also known as a splitting vessel. [...].

When there is a changeover on CL [...], the whole processing line will be flushed with water.

2.2.2. Centrifuge Line and Bleaching Line

CL [...] has the same structure as CL [...], apart from the fact that there is an extra centrifuge to wash the oil. A schematic overview of CL [...] is given in Figure X and a schematic overview of BL [...] is given in Figure X. The [...] centrifuges that are used to wash the oil are smaller than the washing centrifuge on CL [...]. However, the kneader mixers on this CL are different than the kneader mixers on CL [...]. [...]. [...] CL have their own soap splitting department, because fatty acids from different oils are actually different from each other and should not be mixed. For the decanters it is possible to exchange them with the soap splitting departments.

From CL [...] the oil can go to both TP [...] and BL [...]. BL [...] consists of [...] buffer vessels, a small tank where the bleaching earth is added to the oil, a dry bleacher, a drip tank, an amafilter and a break tank. [...]. CL [...] is fully responsible for the input of BL [...]. To be more precise: the input comes from the dryer of CL [...]. Before the oil goes to the dry bleacher, water and nitrogen are added inline and bleaching earth and/or norrit are added in a small separate tank. From that tank, the mixture goes to the dry bleacher. This tank has a capacity of [...], including the volume that has been present in the piping before the dry bleacher. From the dry bleacher, the mixture goes to the drip tank. This tank has a capacity of [...]. In the process, the drip tank does not have a function other than that it acts as some sort of intermediate storage. From the drip tank the oil goes to the amafilter. The oil with bleaching

earth will be pumped through the filter. Every time it goes through the filter, a little bit of bleaching earth sticks to the filter. The more bleaching earth sticks to the filter, the more efficient the filtering will be. As a consequence, the first oil that passes through the filter will not be clean when it leaves the filter, since a filter is always clean when a batch is started. Depending on the amount of bleaching earth or norrit needed, the filter might need a clean-up during the process. This might take approximately [...].
Oil coming from the filter will be passed to a so called break tank, from where the oil will be pumped to TP [...]. This is only the case when the clarity of the oil is sufficient enough. When the clarity of the oil is not sufficient, the oil will be returned via the blackrun to the drip tank. The decision whether the oil can go to TP [...] is made by the system controlling BL [...]. The maximum capacity of the break tank is [...].
Once the last bit of oil is pumped from the dry bleacher and the drip tank, the BL will start cleaning itself. The last bit of oil going from the drip tank to the filter is called Rest Volume of Filtration (RVF). This last bit of oil will, just like the other oil, be pumped through the filter as long as needed to reach the required clarity. However, this oil will not be pumped back into the drip tank, but will be pumped alongside the drip tank directly back into the amafilter.
With a changeover, BL [...] will be flushed with nitrogen. A full clean up of the line will take about [...] hours.

2.2.3. Bleaching Line and Deodorizer

BL [...] consists of a dryer, a wet bleacher, a dry bleacher, a drip tank, a set of amafilters, a break tank, a cricket bat and a dust filter. A schematic overview of BL [...] and DEO [...] is shown in Figure X.
Naturally, oil contains water. For the efficiency of the wet bleaching process, it is required that the oil contains just the right amount of water. Because of that, the oil will be fully dried in advance, after which the exact right amount of water is added, together with some chemicals. The dryer has, just as the other dryers that have been discussed so far, a negligible residence capacity of [...].
Afterwards, the bleaching earth will be added via a small separate tank from where the mixture goes to the wet bleacher. After processing in the wet bleacher, the dry bleacher will remove the water from the bleaching earth. The capacity of the wet and the dry bleacher together is [...]. From the drip tank right behind the dry bleacher, the mixture will be pumped to the filter set. The drip tank has a capacity of [...]. The amafilters of the filter set are alternately used. When one of them is in use, the other one is cleaned and vice versa. This alternate use of the filters creates a continuous outflow of the filters, which is required for DEO [...]. [...]. The cricket bat is a vertical barrel with several hanging cloths. The oil is pumped through the cloths. It should obstruct the final pieces of matter that slipped through the earlier filtrations.
When the oil has not been filtered enough, the oil will be pumped back from the break tank to the drip tank via the blackrun. The break tank has again a capacity of [...]. When the drip tank contains the next batch already, the RVF will be pumped back in the direction of the drip tank, but will go alongside the drip tank directly to the filter.
When there is a changeover, BL [...] is flushed with nitrogen. After flushing, only a minimum amount of oil will stay behind in the equipment.

[...].
[...]. Next, the oil is pumped to the desired destination, which could be for example the receiving tank of the Enzymatic Interesterification (E. I.), TP [...], Tank Park [...] (TP[...]) or the Refined Oil Storage (ROS).
[...]. Due to this continuous flow, the dust filters are operable at the same time. Back in the days the dust filters were operable one at the time, because one of them was always in reserve. Because of an increase in the load on the dust filters, the operators usually choose to use both dust filters at the same time to divide the pressure on both dust filters. Typically this all depends on the situation. It will take [...] to change a dust filter.
When there is a changeover, the operators ensure that there is an empty bucket in between the two varieties of oils. It may be clear that the final bucket of the current batch cannot be released from the DEO by means of the overflow. Because of that, each bucket is equipped with a drain. When the oil has stayed in the bucket sufficiently long enough the DEO will be drained from bottom to top. However, the DEO will not become empty. When the operator is sure that the new variety of oil will not be able to mix with the old variety when the new variety is about to overflow the first bucket – so that there is

an empty bucket in between – the new variety will be pumped into the first bucket.

2.2.4. Bleaching Line and Deodorizer

BL [...] consists of [...] bleaching sub-lines, each consisting of a bleaching tank, an amafilter, a break tank and a buffer tank. A schematic overview of BL [...] together with DEO [...] is shown in Figure X. For the [...] bleaching tanks, the input material comes from TP [...]. Both tanks have a capacity of [...]. The oil is pumped into the bleaching tanks after which the bleaching earth is added to the oil. After a certain residence time, the mixture is passed through the amafilter, from where the mixture goes to the break tank. Again, the latter has a capacity of [...]. The amafilters have a capacity of [...]. This means that the amafilters are able to filter [...] of bleaching earth out of the oil before they have to be cleaned. A clean up takes [...].
When it appears that the oil is not sufficiently filtered, the oil will be pumped from the break tank via the blackrun back to the bleaching tank, from where it will go through the amafilter again until it is sufficiently filtered. When the oil is sufficiently filtered, it will flow through the cricket bat and the dust filter to the buffer of DEO [...]. The capacity of the cricket bat, the dust filter and the buffers are [...].

[...]. The oil is pumped from the buffers directly into the first bucket in which the oil will be deaerated. The result is the same as when the oil will be dried by a separate dryer as is the case with DEO [...]. With the deaeration, the bucket is also brought to a vacuum of [...]. In the same bucket, the oil is heated to the desired temperature – which depends on the oil being processed – and when both requirements of pressure and temperature are met, the oil is drained to the second bucket. [...].
[...]. This is how the process continues. It may be clear that bucket [...] should be drained before bucket [...] can drain its contents into bucket [...]. When drained from bucket [...] the oil is pumped through the dust filters to its desired destination. Just as the dust filters mentioned earlier, both dust filters have a capacity of [...] and will be used in the same way as with DEO [...].
However not desirable, it is possible that buckets may fall empty due to for example delays or problems with any of the upstream processes. In the situation where the DEO has to deal with more than three empty buckets in a row, the control program will decide that the first three batches of oil – each with a batch-size of one bucket – in bucket 1 should be heated to a higher temperature than with the regular process, before the oil is allowed to be drained to the second bucket. [...].

[...] of oil in the piping between the buffer of the BL and the first bucket. The program of the DEO takes this amount into account by adding it to the amount of oil that is present in the buffer of the BL. The diagnostics for the filling rate of the bucket also takes into account the amount of oil that is still in RVF. The diagnostics can only be performed when the control program has released the buffer to start pumping to bucket [...]. This implies that there should be no inflow into the bucket anymore and that the level of oil in the buffer should be stable. This means that it is not possible to run diagnostics when the oil is still in a blackrun for the filtration, but it is possible when the oil is in RVF.

In case of a changeover, the BL and the piping between the BL and the DEO will be flushed with nitrogen. The amount of oil that stays behind in the BL is negligible. The DEO is not cleaned.

2.2.5. Bleaching Line and Deodorizer

BL [...] is comparable with BL [...]. However, BL [...] consists of [...].
DEO [...] is equal to DEO [...]. As can be seen in Figure X, there is a physical connection between BL [...] and DEO [...] and BL [...] and DEO [...]. However, this is a connection that is only rarely used in practice. This is due to the properties of the different oils that are processed on these lines. Normally, BL [...] and DEO [...] are dedicated to the processing of the so called lauric oils. BL [...] and DEO [...] are dedicated to the processing of the so called non-lauric oils. Lauric oils have a C12:0 content of 47-50%. Lauric oils that SDOZR processes are (mixtures of) coconut oil (CN) and palm kernel oil (PK). Non-lauric oils only have a C12:0 content of about 0,2-0,3%. Non-lauric oils that SDOZR processes are (mixtures of) palm oil (PO), sunflower oil (SF), rapeseed oil (RP) and soybean oil (BO). Big problems occur when the end product requires a low C12:0 content and the non-lauric product is processed after a lauric product. That illustrates why the production of these two kinds of oils is so strictly separated. It is the best way to minimize the contamination of these oils.
[...]. To prevent contamination, the processing order of the products is very important. As mentioned

before, a distinction can be made between the lauric and non-lauric oils. On top of that, the non-lauric oils can be divided in a group of oils that have a so called C12:0 specification and a group of oils that does not have the C12:0 specification. When the oils do have the C12:0 specification, it is really important that the C12:0 content of the oil is within the specifications provided by the customer. For oils without this C12:0 specification, this is less important.

Table 2.1: Possible processing orders with respect to contamination

	Lauric (0)	**Non-lauric, no C12:0 (1)**	**Non-lauric, C12:0 (2)**
Lauric (0)	✓	✓	X
Non-lauric, no C12:0 (1)	✓	✓	✓
Non-lauric, C12:0 (2)	✓	✓	✓

To summarize, there are three families of products at SDOZR: the lauric oils (Family 0), the non-lauric oils without a C12:0 specification (Family 1) and the non-lauric oils with a C12:0 specification (Family 2). With these three families of products it is possible to create a processing order where contamination is minimized. The different orders of product families is shown in Table 2.1. This table should be red as follows: if there is a checkmark in a cell, then the product family in the column can be processed after the product family in the row. If there is a cross, it means that this processing order is not possible if there is no cleaning operation in between.

2.2.6. Chemical Interesterification

Next to the sub-lines, BL [...] also consists of [...]. The processes that are performed on the C. I. cauldrons can be divided in direct interesterification and indirect interesterification. With a direct interesterification the oil will only be interesterified, after which the oil is pumped to Buffer 24. From Buffer 24, the oil can be used as input for BL [...] or BL [...]. With an indirect interesterification, the oil will first be interesterified, after which a pre-bleaching process will be started - similar to the bleaching processes described earlier. Also the filtering process will be similar to the filtering process described with the earlier described bleaching processes and is performed on sub-line [...] of BL [...]. After filtering, the oil will not go to the buffer of sub-line [...], but will be pumped to the intermediate storage of TP [...]. It is not explicitly addressed in Figure X, but the blackrun of the filter process for the C. I. is not to the bleaching tank of sub-line [...], but to the C. I. cauldron the bleaching process was performed in.

It should be noted that the processes regarding the C. I. require a high level of interaction with the equipment of sub-line [...] from BL [...]. When the filter equipment is occupied by a filter process for the Chemical Interesterification, the bleaching tank from sub-line [...] cannot be used and vice versa. Besides, the C. I. cauldrons process both lauric and non-lauric oils. This means that, even though BL [...] is dedicated to the processing of non-lauric oils, sub-line [...] does process lauric oils. This causes a high risk of contamination, resulting in the same situation as when DEO [...] processes oils that are usually processed on DEO [...].

2.3. Scheduling at Sime Darby Oils Zwijndrecht Refinery

SDOZR uses a scheduling software program named [...]. The interface of this scheduling program is a Ganttchart, from which an example is given in Figure X. [...]. For the understanding of the scheduling process of SDOZR, the scope sketched in Section 1.2 is disregarded for the moment. [...]. Typically, when the ROS is scheduled, the order of scheduling of the rest of the plant is scheduling the mixing tanks in the ROS, DEO [...], the E. I., DEO's [...], BL's [...], the mixing tanks in TP [...], the Hardening, the centrifuge lines, the Dry Fractionation and the crude oil delivery. Parallel to scheduling the hardening, the Chemical Interesterification will be scheduled. A flowchart of the scheduling order is shown in Figure X.

2.3.1. Basic principles of the scheduling program

The scheduling cycle starts with the import of customer orders. The scheduler is given information on the product to deliver, the amount and delivery time. This information comes from the software program in which the orders are entered after they have been sold. It takes a few clicks of the mouse to load the orders into [...]. New orders or orders that have been changed are listed in a pop-up screen to notify the scheduler.

For the scheduling of the production, there are different ways to implement the processing steps in the schedule. For example, it is possible to load the batch into the time-slot that is available as soon as possible at the moment of scheduling, but it is also possible to set a date and time from which point the batches should be loaded into the Ganttchart. After loading the batch into the Ganttchart, it is possible to drag it through the chart manually. It depends on the scheduler, but usually, the batch is loaded into the Ganttchart on a preset date and time. In this way, the newly scheduled batches are easy to find and can be easily dragged through the chart to the place the scheduler desires to have it.

The batch that is loaded into [...], receives a certain duration of time. In the interface of [...], the Ganttchart, this is logically displayed as a block with a certain width and it also receives a certain color. Batches that produce the same oil naturally receive the same color in the chart. The time duration is determined based on among others the batch-size. Also the flow rate takes part and is on its turn determined by the recipe of the product to be produced. This recipe determines how long a certain oil should stay in the DEO to obtain the desired specifications.

Every colored block has two white blocks attached to both sides of it. This corresponds to the time required to execute a change between the batches to be produced and is the same for all DEO's and products. However, when two batches of the same oil are scheduled after each other - in other words, the end of the first batch and the start of the next batch are in between a certain time from each other - the white block at the end of the first batch and the white block at the start of the next batch automatically disappear. This means that it is not needed to take the change in production batches into account.

The time of the delivery of the order is the start and basis for the scheduler to determine when a batch should be produced. The exact time when to start the production of a batch is determined by the scheduler by means of a line graph. This type of line graph is used throughout [...] and usually consists of three lines. An example of such a line graph is given in Figure 2.1. One red line is placed on the 0-level, another red line indicates the maximum capacity of the tanks dedicated to the same product and the third line indicates the gradient of the tank-level. Because the tanks are not dedicated to one single product, it is possible that the second red line indicates another maximum value every day. [...] reads the tank-levels every morning as being the 0-indicator for the rest of the day.

The third line of the graph shows when a product is added to the tanks (or: produced) and when a product is extracted from the tanks. A condition of the graph is that the third line does not go below the 0-level and does not exceed the maximum indicator too much. Because of these conditions, it is possible to determine if there is enough stock to meet the demand from the orders based on these lines. On top of that, the graphs take into account the stock that was already produced in the last days before the scheduling started or was already scheduled to be produced.

It might happen that the tank-level-indicator-line will go below the 0-indicator-line when a batch for an order is extracted from the tank. This means that there is too little stock to supply the complete order and that actions should be taken to prevent a negative tank-level. The schedulers have several options. The first option is to reschedule the batch for this order to an earlier point in time to make sure that there is enough stock at the time of extraction. When there is no free spot available for the batch, other batches will, if that is possible for the schedule, be rescheduled to a later point in time to create a spot for the

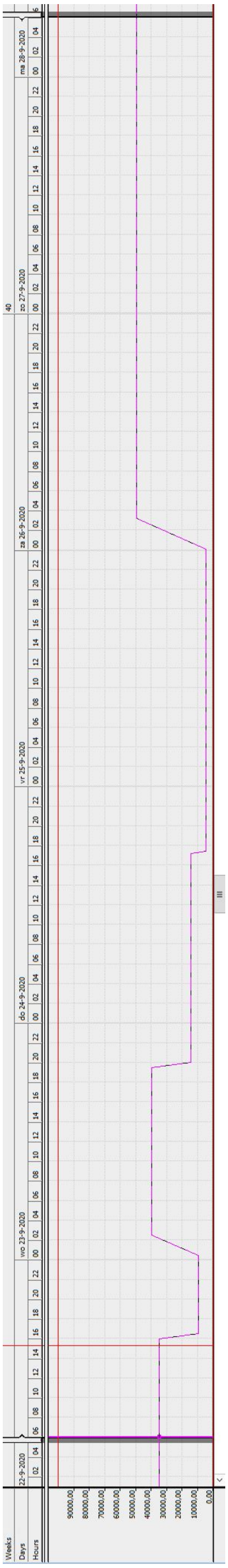

Figure 2.1: Example of a tank level indicator [Internal documents].

batch. The second option is to reschedule the time of delivery to a later point in time. In this way, more time is created on the equipment to be able to produce the batch. The third option would be to adapt the size of a batch that was scheduled earlier than the batch belonging to the concerning order. It is also possible to combine the first and the third option, or to do nothing. What option to choose depends among others on the amount of oil that is falling short at the time of the delivery of the order, but also on the capacity of the equipment. When the tank-level is only 200 kg below the 0-indicator on a batch-size of for example 45 tonnes, no action will be taken. The second option is for example only possible when an earlier scheduled batch is not scheduled too far in advance of the concerning order due to deterioration or maximum shelf-life and when the maximum capacity of the equipment is not exceeded. When it appears that the maximum capacity of the equipment will be exceeded, it is possible to create a new batch from the remainder. However, when the batch-size of the remainder is below the minimum capacity of the equipment, it would be better to divide the total over the two batches.

When the tank-level-indicator-line goes over the maximum-indicator-line, the schedulers have more or less similar options to solve the situation compared to the situation where the tank-level-indicator-line will go below the 0-indicator-line. It is for example possible to gamble to do nothing when only a little surplus is expected. It is possible that the expected surplus will actually not be produced due to losses in production. This gamble will be taken based on the experience of the schedulers. High interaction between the schedulers and the operators from the ROS is also possible here. When there will be produced a small amount more than fits the tank-capacity at that moment, an operator of the ROS can decide to temporarily allocate a small tank and extract from this tank first at the time of delivery. When larger quantities are involved, the operator from the ROS can decide to add an extra tank to the product group. The consequence will be that the maximum tank-level for this product will be increased in [...] for that moment. Furthermore, it is also possible to interact with batches that were scheduled earlier. It is possible to (also) reduce the size of this batch - keeping in mind the minimum batch-size - to spread the production over a longer period of time. It is also possible to increase the size of this preceding batch - keeping in mind the maximum batch-size - to be able to cancel the current batch.

In the situation where the equipment is scheduled to produce the end products, the equipment is scheduled in such a way that the level of the tanks are as low as possible, preferably even 0. The reason is that the less residual material there is in the tanks, the fewer tanks are being occupied by a product that will not be picked up within the scheduling horizon. Besides, also less cleaning operations are required at the end of the week. This way of scheduling can also be referred to as the make-to-order principle. In the situation where the equipment is scheduled for intermediate products, the equipment is scheduled in such a way that the level of the tanks are maintained at a certain threshold value. The latter is also due to the possible presence of sediment. When the tank-level reaches a certain minimum, there is a possibility that this sediment enters the process downstream. Because of that, extra processing might be required - such as extra bleaching or deodorization - to be able to produce the batch within its required specifications. This method of scheduling can also be referred to as the make-to-stock principle.

2.3.2. Scheduling the ROS

As mentioned in the introduction of this section, the process of scheduling starts with the scheduling of the [...] loading berths in the ROS, where the customers or the transporters will pick up the order. For this part of the refinery, the [...] is used. First, the order check as described in Section 2.3.1 is performed. The new orders are loaded into the bars of the Ganttchart of [...] dedicated to the locations of unloading in the ROS. Date, time and batch-size are coupled to the order. It is up to the scheduler to check if there are no overlaps. If there are, the scheduler should drag the overlapping orders manually to a fitting spot on the Ganttchart and inform the corresponding department to communicate the new unloading time to the customer.

Among the products that SDOZR delivers to here customers are mixtures. These mixes are mixed in among others the [...] mixing tanks present in the ROS. [...] of the mixing tanks have a heating spiral. Because of that, these [...] mixing tanks are dedicated to mixtures that consist of hardened products. As soon as the orders are loaded into the Ganttchart of [...], the scheduler manually schedules the batches to be mixed and places them in the Ganttchart. The allocation of storage in the ROS is currently not scheduled (Baart, 2020).

2.3.3. Scheduling the Deodorizers and the Enzymatic Interesterification

Typically, the most products that will be delivered from or mixed in the ROS, will be processed by one of the DEO's as a final processing step. This final processing step on the DEO's is loaded into [...] and is already linked to the order that will be delivered from the ROS. The link with the order monitors if the batch will be produced on time for delivery by the DEO. When the batch is placed in the Ganttchart at such a time that it will not be finished on time, the corresponding delivery turns red. As soon as this batch is dragged to an earlier time in the chart, the red color disappears. However, this principle does not always work, because the production of one batch can be enough for more than one particular order. Because of that, it is possible that due to various reasons the batch corresponding to an order will be changed to a smaller batch. When this size is smaller than the order asks for, the delivery will turn red. However, it is possible that the size difference is taken into account in a batch that has been scheduled earlier. So even though [...] tells the scheduler that there is a problem, there will be enough stock to deliver the order. This phenomenon can be solved by decoupling the batches from the orders and couple them again. The batches are then automatically linked to the required order.

[...] contains a list saying which oil should be produced on which DEO, as desired by SDOZR. For example, the lauric products on DEO [...] and the non-lauric products on DEO [...]. Because of this list, each batch is automatically assigned to the desired DEO. However, at all times, the schedulers are authorized to change this to their own insight and wishes. A reason to change from DEO could be the batch-size. With large batches, it is the preference of SDOZR to produce large batches on DEO [...] instead of on DEO [...] or DEO [...]. However, [...] is not able to automatically assign a batch to DEO [...] when the batch-size is above a certain threshold, so this has to be changed manually. Once the preferences of the schedulers are implemented and the batches are loaded into the Ganttchart by [...], it is still possible to drag the batches around the chart, also to other DEO's.

The interaction between DEO [...], the E. I. and DEO [...] can be found in the scheduling process in the order of scheduling. First, DEO [...] will be scheduled, after which the E. I. checks what it needs to produce to supply DEO [...] with its input material. This list of input material is implemented in the schedule for the E. I.. On its term, DEO [...] checks what oils it needs to pre-refine to supply the E. I.. Then again, this list of input material is implemented in the schedule for DEO [...]. The batches that should be scheduled on DEO [...] should be implemented manually, since these batches are not linked to the orders. In this case it is determined based on the tank-levels if and if yes, how much needs to be produced.

When it appears with the scheduling of DEO [...] and DEO [...] that there are periods in the schedule of DEO [...] and DEO [...] where no production will take place, there are various options for the schedulers to deal with these periods of non-production. One of the possibilities is to reschedule the batches that are scheduled later right after the batch after which this period of non-production initially occurs. However, the ROS only has limited storage possibilities. When a certain product is produced on Sunday, is stored in the ROS, but is actually needed only on Wednesday in the late afternoon, the product will only occupy a tank. Besides, the product might exceed its shelf-life in the meantime. Another option would be to shut down the DEO. However - especially in the case of DEO [...] - you don't just shut down a DEO and it costs a lot of time and money to restart it. The last option would be to take over production from the CL's. That would mean that the oil will be physically refined instead of chemically. There are only a few oils for which this is possible, namely nPKcs and nPK. Because of situations in which there is a staff shortage in the refinery - where the CL's shall be shut down first - there is a win-win situation. When the DEO's take over production from the CL's, free time is created in the schedule of the CL's to overcome the staff shortage. However, the products that have to be processed by DEO [...] and DEO [...] will always have priority on the batches that are originally processed by the CL's. Because of that the DEO's will always process the minimum-sized batches to preserve the flexibility in rescheduling the batches to the CL's. In this way, the batches from the CL's can also be used to build in buffer time in the schedule in case that that is needed. The previously mentioned issue with a lack of storage capacity in ROS is not an issue here, because batches originally processed by the CL's will be stored in TP [...]. The situation described above will occur more at DEO [...] than at DEO [...], because DEO [...] usually has enough batches to process.

With the scheduling of the DEO's, a minimum amount of time or quarantine time should be scheduled between the end of the DEO process and the time the batch should be present in the ROS for delivery. This has for example to do with the quality analysis that is performed in the ROS. Some of the required tests take quite some time that should be considered in the scheduling of the batch. The schedulers take this into account, but it is not automatically added to the relevant product. However, when the batch is scheduled in such a way that there is too little time between the delivery to the ROS and the delivery to the customer or transportation company, this batch will turn red.

2.3.4. Scheduling the Bleaching Lines

Once all the DEO's are scheduled, the schedule of DEO [...] up to and including DEO [...] will be confirmed. The consequence of this confirmation is that the BL's will also be scheduled. This schedule is linked to the schedule of the DEO's and apart from switching the batch to another cauldron, no changes can be made to the batch. The batches are automatically scheduled sufficiently in advance before the DEO starts to extract oil for its process.

2.3.5. Scheduling the Mixing tanks in TP

TP [...] has several mixing tanks that mix blends for DEO [...]. After these blends have been processed by DEO [...], they will be processed by the E. I.. Obviously, the making of the blends is also scheduled. DEO [...] is the 'customer' for the mixing tanks in TP [...] and therefore determines how much of what blend should be made at what time. The base products for the blends could be crude oils from Tank Park [...] (TP[...]), hardened or centrifuged oils. Just as with some final products, it is important to reserve some time between the finishing of the blend and the start of the process in the DEO to be able to analyse the blends. Because the E. I. only has [...] receiving tank, the deodorization of the blends is altered with the deodorization of batches that don't go to the E. I.. This creates some time for the E. I. to process the receiving tank before a new blend will be delivered. Because of the time required between the blends on DEO [...], the mixing of the blends should not be scheduled tight behind one another on the mixing tanks of TP [...].

The required time between the blends are not automatically implemented in the Ganttchart of [...]. It is also not considered as a changeover as with the white blocks for the batches on the DEO's. The schedulers should be aware that they should reserve time between the blends. However, when there is too little time scheduled between the batches, the latest batch will turn red. The implementation of the blending batches into [...] is done manually.

2.3.6. Scheduling the Hardening and the Chemical Interesterification

The scheduling process of the Hardening also starts with loading in the customer orders. Besides that, the schedule of the DEO's and BL's are also loaded into the [...] environment for the Hardening. Based on these schedules and the experience of the schedulers, it is possible for the scheduler of the Hardening to estimate how much of what product is needed from the Hardening as input material for the Chemical Interesterification. The Hardening produces batches that will be stored in TP [...] and TP [...] as input material for the DEO's, but there are also orders for which deodorization is not needed. These batches will be delivered to the customer - [...]. The time required for each batch to produce depends on the flow rate set per batch by the scheduler.
[...].

For both the Chemical Interesterification and the Hardening, the check for sufficient stock to be able to provide the equipment further downstream the factory with input material is based on the line graph described in Section 2.3.1. Just as with this line graph, there is a red line indicating the maximum capacity of the tanks allocated to the relevant product, a red line indicating the 0-level of the tank and a line indicating the tank-level. At this part of the factory, it is required to schedule and produce according to the make-to-stock principle instead of scheduling for the make-to-order principle as was required for the ROS.

When it appears - based on the line graph - that the equipment is running out of stock, a batch of the concerning oil should be implemented into the schedule by hand. This is not linked to the orders to be delivered. A known handicap of the Hardening is a delay in production. The schedulers schedule some buffer time to overcome this. Constant monitoring might show that more buffer time is required or that batches can be started earlier than initially scheduled.

2.3.7. Scheduling the Centrifuge Lines and the Dry Fractionation

The next step in the scheduling process is the scheduling of the CL's. The CL's produce among others the input material for the Hardening. Here, a make-to-stock principle is maintained as well. Extra production batches required here should also be implemented manually. The same procedure holds for the Dry Fractionation, which produces input materials for DEO [...].

As can be seen in Figure X, it is possible for CL [...] to skip BL [...]. Whether a batch will be bleached on BL [...] depends on the product and is pre-set in [...]. The scheduling of BL [...] is linked to CL [...] in the same way as BL [...] up to and including BL [...] are linked to DEO [...] up to and including DEO [...]. When a batch should be bleached on BL [...] after it has been centrifuged on CL [...], a block will be automatically created in the Ganttchart in [...].

CL [...] processes a certain type palm oil for [...]% of the time. When it is really necessary, certain types of palm kernel oil could also be processed on CL [...]. These oils can also be processed on CL [...]. In case it concerns the palm oil, this only happens when it is really necessary, when for example maintenance is required for CL [...]. Except for the earlier mentioned oils, CL [...] can also process coconut oil, rapeseed oil, sunflower oil and soy bean oil.

2.3.8. Scheduling the crude oil and finishing the schedule

When all the above is scheduled, a check on the stock of crude oils is all that is left to do before finishing the schedule. The stock of the crude oil is shown in a same sort of line graph as described earlier. Based on these line graphs it is determined if there will be enough stock until the next delivery of crude material or that it is required to ask for an extra delivery from the external storage. The decision for the latter is based on the complete schedule and the schedulers' experience. Deliveries of crude oils from outside the plant are shown in the same Ganttchart as the other equipment. This is because when a vessel from an external location is unloading its load, pipelines are occupied to pump the load to its desired location.

However, when the delivery of crude oil is scheduled, the schedulers should also take into account the amount of stock. They should check if there is enough storage capacity for the particular product in TP [...] at the moment of delivery. When this is not the case, the schedulers should process the concerning product into semi-finished product, which can be stored in TP [...] or TP [...].

After the check on crude oils, the schedule for the scheduling horizon of [...] is ready to be turned into the final schedule. The schedule can then be sent to the ones who work with the schedule, such as the operators in the refinery. However, because the customers are allowed to adapt and cancel their orders at all times, the schedule will no longer be up to date from the moment it is turned into the final schedule. Because of this, the schedulers should go through the described process of scheduling every day. As more changes are made during the scheduling horizon, the quality and efficiency of the schedule will continue to decrease.

2.4. Conclusion

In the refinery of Sime Darby Oils Zwijndrecht Refinery (SDOZR), the processes of physical refining, chemical refining and chemical interesterification take place. Physical refining can be divided in degumming, bleaching and deodorisation. Chemical refining can be divided in degumming, alkali neutralisation, bleaching and deodorisation. The degumming and alkali neutralisation processes are executed on a centrifuge line (CL). Bleaching is executed on a bleaching line (BL) and deodorisation is executed on a deodoriser (DEO). The Chemical Interesterification (C. I.) is executed on C. I. cauldrons. SDOZR features [...] CL's, from which [...] is connected to BL [...]. The other [...] bleaching lines, BL [...], are connected to a deodorizer, DEO [...] respectively. The C. I. cauldrons are lined up near BL [...]. Depending on the C. I. process, the C. I. requires to filter its material on the filtering department of the [...] sub-line of BL [...]. The combination of BL [...] and DEO [...] are dedicated to processing oils from the lauric product family. The combination of BL [...] and DEO [...] are dedicated to processing oils from the non-lauric without a C12:0 specification-product family and the non-lauric with a C12:0 specification-product family. Due to the characteristics of these three product families, it is not desirable to contaminate products of the lauric product family with products of the non-lauric with a C12:0 specification-product family. However, C. I. processes both lauric and non-lauric products and because this department requires filtering on the [...] sub-line of BL [...], there is a high risk on contamination.

On top of this, minimum and maximum batch-size and/or equipment capacity, piping, deterioration or maximum shelf-life and minimum storage or quarantine time are important. The equipment capacity of the filters is determined by the amount of residue it is able to filter out of the emulsion.
Scheduling at SDOZR is executed from back to front and is at some points highly iterative. In this way, sub-problems are solved. Scheduling starts by scheduling the loading berths of the Refined Oil Storage (ROS), followed by the mixing tanks in the ROS, Deodorizer [...], Enzymatic Interesterification, DEO [...], BL [...], the mixing tanks in Tank Park [...] (TP[...]), Hardening, CL's, Dry Fractionation and finally the crude oil delivery. [...]. At the moment, the schedules are created manually. Preferences are implemented in the scheduling software used, but the desired start of the processes has to be addressed manually. Improvements to the schedule are made by manually shifting around the tasks through the scheduling software interface, which consists of a Ganttchart and a line graph indicating stock levels. Deviation from the preferences is done because of for example large batch-sizes, overload on the preferred processing unit, etc.. The decisions are made and the simultaneous scheduling of the hardening and C. I. are based on the experience of the schedulers and are not registered or documented.

3 State-of-the-art in production scheduling solutions

In this chapter, the state-of-the-art in production scheduling solutions is given. Based on this information, a mathematical model will be selected from literature that will be used as a basis for the development of the mathematical model for Sime Darby Oils Zwijndrecht Refinery (SDOZR). First Section 3.1 will briefly go into the history of production scheduling. The state-of-the-art in production scheduling solutions will then be elaborated in Section 3.2. Next, Section 3.3 will classify the problem in a structured way. Then, in Section 3.4 this classification is applied to the literature elaborated in section 3.2 in order to make a motivated decision on what mathematical model to use as a basic model for this scheduling problem.

3.1. History of production scheduling

Production scheduling dates back to the 19th century when the first simple, small factories appeared (Dawande et al., 2006). The factories produced only a small amount of different products and produced them in large batches. The shop floor was run by foremen who hired operators, purchased materials, planned and managed production and delivered the product (Dawande et al., 2006). The schedules at that time, if they were even present, consisted of a list when the work on an order should start and when the order was due, without further information on how long individual operations on the order should take. The main objective at that time was to utilize the equipment as much as possible (Dawande et al., 2006).
Around 1890, factories started making more different products in the same factory. The factories became so complex, that the foremen could not handle it on their own anymore. At that point, planners took over the scheduling and coordination from the foremen (Dawande et al., 2006). It was around the First World War that the first real scientific scheduling techniques were recognized. For example, Gantt discussed the scheduling activity by introducing the so called Gantt chart. The Gantt chart can be applied to the operators, the machines, the orders and the products and are a way to visualize the schedules and shop status. An example of a Gantt chart is given in Figure 3.1 and it still is one of the most common tools for scheduling and scheduling visualization (Dawande et al., 2006).

Production scheduling greatly improved when the use of computer algorithms was introduced around the 1950s (Dawande et al., 2006; O'Brien, 1969). By gaining more experience on the use of computers and the required software, the foundation was laid for the future of computer-based production scheduling (Dawande et al., 2006). The early computer-based scheduling systems automated the collection of data and the processing of functions that already existed in previous scheduling activities. The system was able to generate a dispatch list (a task-to-be-assigned list) for each work station. The next generation of scheduling systems, which arose around the 1980s, was developed to function as decision supporting software, where its aim was to reduce the needed time for the development of a schedule. Later, scheduling decision making computer-based systems were introduced. Such systems were for example able to prioritize jobs in the queue for correct sequencing (Dawande et al., 2006). Nowadays scheduling is used in several areas, such as transportation, maintenance, unit or mass production, process plants, hospitals and the chemical industry (Dawande et al., 2006; O'Brien, 1969; Williams, 2000). On top of that, the goal of scheduling is no longer limited to a maximization of utilization. Other objectives could for example be a minimization on cost or inventory (O'Brien, 1969).

Figure 3.1: Schematic overview of a Gantt chart (Team Superside, 2018)

3.2. Current scheduling techniques and applications

Many scheduling problems are nowadays solved by a technique named linear programming. Linear programming is a technique covered by an approach named Operations Research (OR), which was used during the Second World War by the British for military problems involving man-machine systems. The approach was applied to the commercial-industry areas by the United States in the postwar era (O'Brien, 1969). An OR application attempts to find the best practical solution to a problem, for which one could think for example of the minimization of costs or the maximization of utilization. Due to the fact that OR emerged during the years the computer was developing as well, OR is a computer oriented approach (O'Brien, 1969).

The research area of linear programming has received a lot of attention in the last few decades. The researches vary from different representations of time, namely discrete or continuous time, to different representations of a scheduling problem, for example, linear problems, nonlinear problems and mixed integer problems. Also in the different areas of applications mentioned before, a lot of research has been done and published. For example in the area of crude oil, crude oil blending and pipeline scheduling.
Pinto et al. (2000) discusses a planning and scheduling application for refinery operations. The model represents a general refinery topology and is able to handle nonlinear process models and blending relations. Lee et al. (1996) specifies a mixed-integer linear programming (MILP) optimization model for short-term crude oil unloading, tank inventory management and a Crude Distillation Unit (CDU) charging schedule. The example discussed consists of one docking station, one CDU and several storage and charging tanks. Before the oil is charged into a CDU, it is mixed for the right composition in the charging tanks. The equations required for the mixing of the oil will turn the optimization model into a nonlinear model, but linearization of the equations prevents this. Wenkai et al. (2002) expanded the problem discussed in Lee et al. (1996) with extra berths and CDU's. However, according to Wenkai et al. (2002), the nonlinear mixing equations should be treated differently since the linearization used

by Lee et al. (1996) often leads to inconsistent solutions. Wenkai et al. (2002) deals with the nonlinear equations by solving a MILP problem, a nonlinear programming (NLP) problem and again a MILP problem iteratively. On top of that, Wenkai et al. (2002) also transformed tri-indexed binary variables into bi-indexed binary variables in order to reduce the number of binary decisions. Reddy et al. (2004) approaches the nonlinear terms discussed in Lee et al. (1996) for the crude composition in a different way. Here, the composition of each tank is divided into two blocks. For one block, the composition is known because no crude oil is added to or extracted from the tank. That makes the constraint on the composition in the tank linear. For the second block, the known composition of the other blocks is used to linearize the equation. On top of that, Reddy et al. (2004) also considers crude oil transfer lines with non-negligible volumes. Moreover, important features such as demurrage, changeovers and settling times are also taken into account. Méndez et al. (2006b) puts more emphasis on the interaction between simultaneous gasoline blending and scheduling.

Moro and Pinto (2004) discuss a problem of crude oil inventory management where the crude oil is delivered by a pipeline and the transfers from the pipeline to the crude tanks, the settling time, interface separation between the different types of oil and the charging of the CDU's are considered. Here as well, nonlinear equations appear when the crude composition for mixing is taken into account. Moro and Pinto (2004) discuss two solutions to handle the nonlinearity. One solution is to solve the mixed-integer nonlinear programming (MINLP) problem that occurs, the other solution is to maintain the linearity of the problem through a discretization scheme applied to the fractions of tank volume that are sent to the CDU.

Magatão et al. (2004) discusses a problem where a long bidirectional pipeline connects a harbor to an inland refinery. In the example discussed, the length of the pipeline is defined to be almost 100 km long. The pipeline transfers a limited set of products, where some orders of transfer are not recommended based on product specifications. To overcome this limitation, a plug (small volume of product) can be used to avoid specific interfaces. It does, however, increase the operational cost. A more complex pipeline scheduling system is described by Cafaro and Cerdà (2008). Here, a unidirectional multi-product pipeline connects a single origin to multiple distribution terminals, where the products are stored for further distribution. The amount of products transferred by the pipeline varies and the distance over which the product is transferred depends on its destination. The scheduling problem has to update the sequence and volumes of new product batches to be pumped in the pipeline dynamically throughout a multi-period rolling horizon. Research has also been done to crude oil scheduling in situations that could be considered further downstream than has been discussed earlier. For example in Göthe-Lundgren et al. (2002), where a scheduling problem is described for an oil refinery company that has one CDU and two hydro-treatment units. In this problem, inventory is also taken into account. In order to produce the required amounts of the products the refinery is able to produce, the CDU can run in 10 modes, and the hydro-treatment units can run in 10-15 modes. Changing modes is expensive, so long sequences and few changeovers are preferred. However, longer sequences require more storage capacity, resulting in higher storage costs. It is up to the scheduling optimization model to find a balance between these two factors. Taking into account inventory and inventory management is accompanied with how long a product can be stored before it needs to be sold or used. The phenomenon might not specifically apply to crude oil scheduling, but when perishable products are considered, this might be a point of interest. At the moment of writing, Entrup et al. (2005) are the only one who explicitly addressed this phenomenon. They have developed a MILP model that takes into account restrictions on shelf-life in the production planning. The restrictions on shelf-life also consider degradation of the product when it is in storage for a longer time.

So far, a particular area of application has been discussed regarding the research area of linear programming and most of the processes taking place in the crude oil refineries discussed so far can be classified as a batch processes. However, literature also discusses this combination of research area and classification in a much more general way. For example, Maravelias (2012) discusses a more general framework and modelling approach in the field of chemical production scheduling. The framework and approach is based on the general classification for scheduling problems of batch processes presented in Méndez et al. (2006a). The classification consists of thirteen categories such as process topology and demand patterns. Literature writes about linear programming problems that focus on a particular element of the classification in Méndez et al. (2006a). Examples are Birewar and Grossmann (1990) and Lin et al. (2002), discussing a problem focusing on the process topology, a flow shop

scheduling problem to be more precise and Ierapetritou et al. (1999), discussing a scheduling problem with multiple product demands. Apart from research focusing on a particular element of the batch problem classification, some research focuses on the industrial applicability of existing scheduling solutions and the set-up of guidelines on how the gap toward industrial applicability could be reduced or even closed in the end, such as Harjunkoski et al. (2014).
Besides the batch process classification, Méndez et al. (2006a) also describes a mathematical model. The model is based on the mathematical model presented in Kondili et al. (1993) and enhances the computational performance compared to the model described in Kondili et al. (1993). The model discussed in Kondili et al. (1993) is a general algorithm for short-term scheduling of batch operations and consists of a MILP model. The model is based on a network representation defined as a state-task-network (STN), newly presented in this work. The model in both papers is described in discrete time. Ierapetritou and Floudas (1998a) give an elaboration of the same model, but then in continuous time. On its turn, Lin and Floudas (2001) extend the model of Ierapetritou and Floudas (1998a) with the ability to simultaneously consider design, synthesis and scheduling.
Apart from batch processes, there are also semi-continuous and continuous processes. A MILP problem of a semi-continuous and a continuous process is given by Ierapetritou and Floudas (1998b). This particular paper extends the model formulation given by Ierapetritou and Floudas (1998a). Castro et al. (2004), Mockus and Reklaitis (1999) and Papageorgiou and Pantelides (1996) all describe batch and (semi-) continuous problems in the same research. The resulting models of Castro et al. (2004) and Mockus and Reklaitis (1999) are an expansion of an existing batch scheduling problem discussed in Castro et al. (2001) and Mockus and Reklaitis (1997) respectively. The mathematical model formulated in Papageorgiou and Pantelides (1996) is based on Kondili et al. (1993) and Shah et al. (1993). Castro et al. (2004) extended the model described in Castro et al. (2001) with among others the ability to handle continuous tasks. They are treated in a very similar way as the batch tasks. Also Mockus and Reklaitis (1999) extended the model described in Mockus and Reklaitis (1997) with the ability to handle both batch and continuous tasks. Kondili et al. (1993), as mentioned before, discusses a mathematical model for the short-term scheduling of a multipurpose batch plant. Shah et al. (1993) has used this information to formulate a model for a cyclic scheduling problem in a batch plant. Mockus and Reklaitis (1999) extend both models for batch plants with the ability to handle continuous tasks. However, not as a short-term or cyclic scheduling problem, but as a scheduling problem for campaign modes.

3.3. Problem classification

In order to make an informed choice on what mathematical model to use as the basic model for the scheduling problem of SDOZR, the scheduling problem of SDOZR should be classified in a structured way. The problem classification described by Méndez et al. (2006a) is used to perform this. The problem classification consists of thirteen categories and is based on the requirements elaborated in Chapter 2.

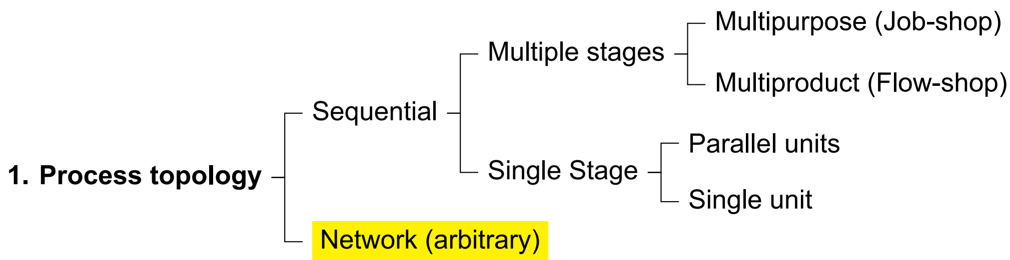

Figure 3.2: Problem Classification, category 1 (Méndez et al., 2006a).

The first category, shown in Figure 3.2, defines the process topology, which can be divided into sequential processes and network processes. A sequential process is defined as a process where each batch needs to follow a sequence of processing steps for production. The sequence is defined by the product to be produced or by the product recipe (Méndez et al., 2006a). It is important to notice here, that with a sequential problem, that it is not allowed to mix a batch with another batch and that it is not allowed to split a batch into several other batches that can be used for different downstream processes.

This means that for example the output of a single batch should be consumed by a single batch (Harjunkoski et al., 2014). A network process is defined as a process that has an arbitrary sequence to be followed. Here, it is allowed to mix and split batches. The mixed batches can act as an input for another batch and a split batch can be consumed by several other batches in the downstream process (Harjunkoski et al., 2014). The process topology at SDOZR can be classified as an arbitrary network.

The second and third category discuss the equipment assignment and connectivity respectively. The equipment assignment, shown in Figure 3.3, can be either fixed or variable. When it is fixed, it is known before scheduling what batch will be processed on what equipment. This could be for example determined by restrictions and characteristics of equipment. When the equipment assignment is variable, the mathematical model will determine what equipment will process what batch.
The equipment connectivity, shown in Figure 3.4, can be either partial (restricted) or full. In the first case, not all equipment is connected to each other and the latter means that all equipment is completely connected to each other. The equipment assignment and connectivity in the scheduling problem of SDOZR can be classified as variable and partial (restricted) respectively.

Figure 3.3: Problem Classification, category 2 (Méndez et al., 2006a).

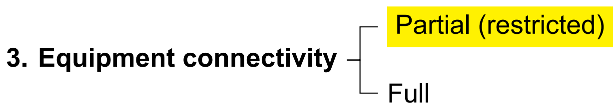

Figure 3.4: Problem Classification, category 3 (Méndez et al., 2006a).

The fourth category of the problem classification, shown in Figure 3.5, defines the inventory policies and can be divided in four subcategories. These are unlimited intermediate storage (UIS), non-intermediate storage (NIS), finite intermediate storage (FIS) and zero-wait (ZW). For both NIS and ZW policy, no intermediate storage tanks are present. However, for a zero-wait policy the processed batch requires to be immediately processed by the next process or task once the current process or task has finished. Otherwise, the batch could for example deteriorate. In case of NIS, the equipment that executes the current task or process could be used as intermediate storage without the batch deteriorating. UIS would imply that there is no capacity restriction in between two processes or tasks. The opposite would be FIS. The latter can be divided into dedicated storage units, that only stores one particular product, and shared storage units, that can store a variety of products. However, not at the same time. When considering the layout of the refinery described in Section 2.2, it is assumed that the problem has a FIS policy with shared storage units. On top of that, the deodorizers ideally have a ZW policy.

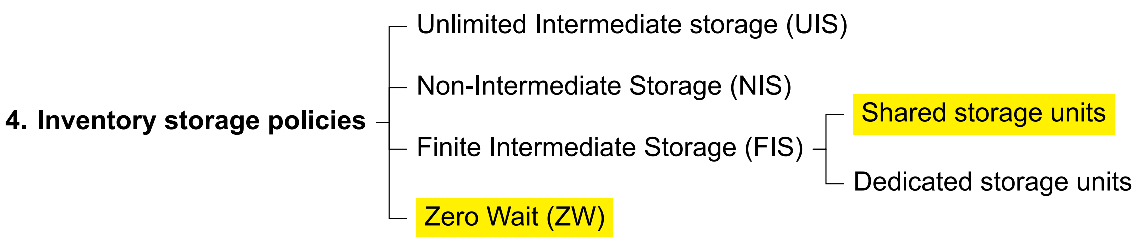

Figure 3.5: Problem Classification, category 4 (Méndez et al., 2006a).

The fifth category discusses the material transfer. As can be seen in Figure 3.6, the category material transfer can be divided into instantaneous and time-consuming. When the material transfer is instantaneous, it means that it is neglected. In the case of time-consuming, three subcategories can be described. This are no-resources, pipes and vessels. In the case of no-resources, the model only takes into account extra time required to actually transport the batch by manually lengthening the process or task with the time required for transport. This could only be assumed when there is no shared equipment on transport. When there is shared equipment for material transport, a distinction is made between continuous transport, often modelled as pipes, and batch transport, often modelled as vessels. Since the equipment in the refinery of SDOZR is connected to each other by pipes, the material transfer can logically be classified as 'Pipes'.

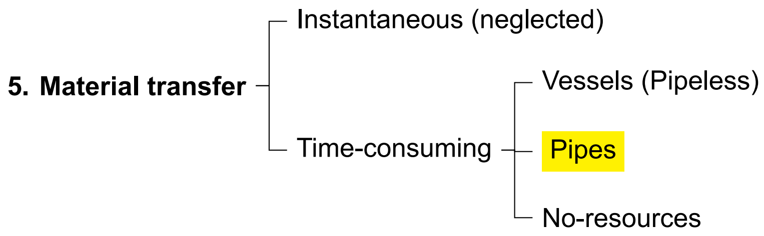

Figure 3.6: Problem Classification, category 5 (Méndez et al., 2006a).

The sixth and seventh category define the batch-size and batch processing time respectively. Both are defined as either fixed or variable as can be seen in Figure 3.7 and Figure 3.8 respectively. The size of a batch is fixed when the size is determined before the scheduling starts. If the scheduling application determines the batch-size, it is variable. The latter could provide a more optimal and feasible schedule than the first. In the case of a fixed batch processing time, it can be either unit independent or unit dependent. When the batch processing time is variable, it depends on the size of the unit or the size of the batch. All is related to the batch-size. For the considered problem at SDOZR, both the batch-size and the batch processing time will be variable.

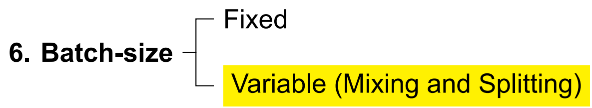

Figure 3.7: Problem Classification, category 6 (Méndez et al., 2006a).

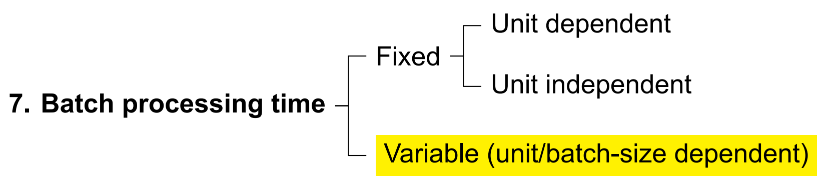

Figure 3.8: Problem Classification, category 7 (Méndez et al., 2006a).

The eighth category discusses the demand patterns concerned with the problem, which can be divided into due dates and a scheduling horizon. An overview is given in Figure 3.9. With a due date, the batch should be finished at a certain time. With a scheduling horizon, it is required to process a fixed or minimum amount over a time horizon. For the problem at SDOZR, a multiple product demand for due dates is considered.

The ninth category defines the changeovers. As shown in Figure 3.10, it is possible that there are no changeovers, but when there are, they can be unit dependent or sequence dependent. A sequence dependent changeover can depend on a product or a product and a unit. A product dependent changeover is likely occur in case products are known to contaminate with each other. The sequence becomes important here, because for example contamination can occur when product A follows product B in the production process, but no contamination will occur when product B follows product A in the production process. When it is decided to let product A follow product B, a cleaning step might be required during a changeover. A unit dependent changeover might occur for example when the molds on the equipment need to be changed for the next process. When a unit dependent changeover is also related to a product dependent changeover, it is for example possible that changing a mold for one type of product takes more time than changing a mold for another type of product. The problem at SDOZR will classify this category as a product dependent changeover.

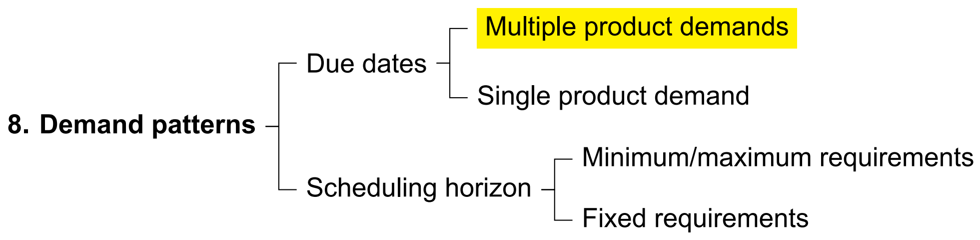

Figure 3.9: Problem Classification, category 8 (Méndez et al., 2006a).

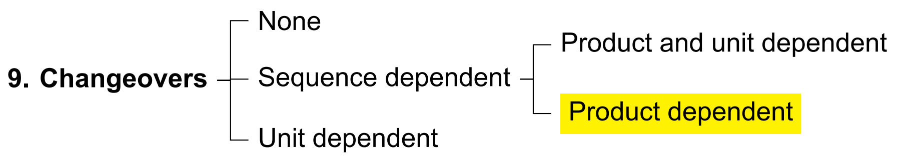

Figure 3.10: Problem Classification, category 9 (Méndez et al., 2006a).

The tenth and eleventh category discuss resource constraints and time constraints respectively. Resource constraints imply all resources required to perform a task or execute a process, except for the crude material or batch and the equipment used. Examples would be employees, steam and caustic soda. As a consequence, as can be seen in Figure 3.11, there can be no resource constraints, discrete resource constraints (employees) or continuous resource constraints (steam and caustic soda).
Time constraints, shown in Figure 3.12, imply periods in which no production can be done. When there are no time constraints, it is assumed that the equipment is available for production 24/7. Examples of non-working periods could be holidays, evenings and weekends. Maintenance has to be performed anyways, but it should be considered whether preventive maintenance (which can be scheduled) is done, or corrective maintenance (maintenance performed when equipment is not working) is done. When a production site operates with shifts, it could be taken into account that during the handover between shifts, it might not be desirable that a batch will start or finish, because all employees would then be occupied by the handover. The situation at SDOZR requires both discrete and continuous resource constraints and the time constraints will include Non-working periods and Maintenance.

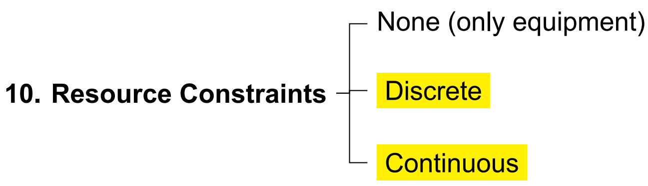

Figure 3.11: Problem Classification, category 10 (Méndez et al., 2006a).

The twelfth and thirteenth category, shown in Figure 3.13 and Figure 3.14, discuss the costs and the degree of certainty respectively. Costs can be made on equipment, utilities, inventory and changeovers. It usually depends on the objective of the optimization problem what costs will or will not be taken into account. Two plausible costs for this problem are changeover costs and inventory costs.
The degree of certainty is divided into deterministic and stochastic. When a problem is deterministic, no random factors are included. For example, all material properties are assumed to be known. In case of a stochastic problem, for example not all material properties are exactly known because they

could change over time. This means that there are some uncertainties in the problem that should be taken care of. The problem at SDOZR concerns a deterministic problem.

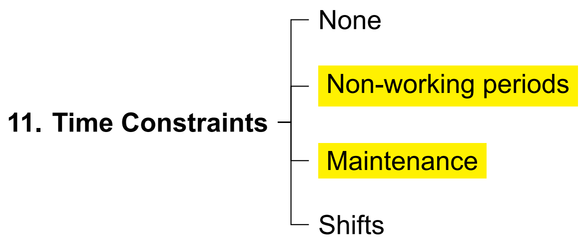

Figure 3.12: Problem Classification, category 11 (Méndez et al., 2006a).

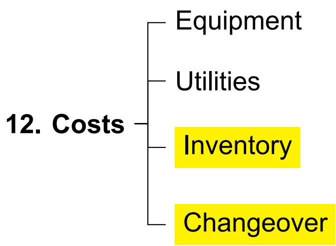

Figure 3.13: Problem Classification, category 12 (Méndez et al., 2006a).

Figure 3.14: Problem Classification, category 13 (Méndez et al., 2006a).

3.4. Selecting an optimization model

In Section 3.3, the scheduling problem has been classified according to the batch problem classification described by Méndez et al. (2006a). However, some aspects of the SDOZR scheduling problem are not covered by this problem classification. In order to make a motivated decision on what mathematical model suits the restrictions of the SDOZR problem best to function as a basic mathematical model, the classification categories described in Section 3.3 are listed in Table 3.1. Just as in Figure 3.2 up to and including Figure 3.14, the items of the classification that are of interest for the current problem are highlighted in yellow. However, there are three extra elements the literature could be classified on. These elements are highlighted in orange and are based on the information given in Chapter 2. They will be elaborated more extensively in Chapter 4. From all the literature that has been discussed in 3.2 a selection of fourteen papers has been made that were considered to be the most promising in describing a basic mathematical model. These fourteen papers all have their own column in the table. When a paper discusses one of the items of the problem classification required for the SDOZR scheduling problem, the box in the table has been ticked with an 'X'. When the 'X' appears in one of the yellow rows, a value of 1 will be assigned to the corresponding box instead of an 'X'. When the corresponding item is discussed, but not extensively, the box on the yellow row will be assigned with a value of 0,5. At the bottom of the table, all the values assigned to the cells in the table are summed up per column or paper and are considered to be a score. As can be seen, the work of Kondili et al. (1993) has the highest score. Therefore, the mathematical model that will be used as a basic model for this scheduling problem is provided by Kondili et al. (1993).
As can be seen in the table, a fifteenth column has been added named 'Contribution of this paper'. In this column, checkmarks were applied to the elements of the problem classification to clarify what

elements were eventually implemented in the mathematical model. In this way, a clear overview of the contribution of this paper to literature has been given.

Table 3.1: Contribution of this paper compared to the literature listed in Table 3.1.

		Méndez et al. (2006a)	Ierapetritou and Floudas (1998a)	Göthe-Lundgren et al. (2002)	Lin and Floudas (2001)	Castro et al. (2004)	Ierapetritou et al. (1999)	Reddy et al. (2004)	Mockus and Reklaitis (1999)	Méndez et al. (2006b)	Wenkai et al. (2002)	Moro and Pinto (2004)	Lee et al. (1996)	Kondili et al. (1993)	Entrup et al. (2005)	Contribution of this paper
1	**Process topology**															
	Network (arbitrary)	1	1		1		1		1	1				1		✓
	Sequential - Single stage - Single unit	X														
	Sequential - Single stage - Parallel unit															
	Sequential - Multiple stages - Multiproduct (flow-shop)															
	Sequential - Multiple stages - Multipurpose (job-shop)															
2	**Equipment assignment**															
	Fixed						X									
	Variable	1	1		1	1		1		1	1		1	1		✓
3	**Equipment connectivity**															
	Partial (restricted)													0,5		✓
	Full	X						X				X	X	X		
4	**Inventory policies**															
	UIS	X				X								X		✓
	NIS					X								X		
	FIS - Dedicated Storage Units	X	X	X	X	X		X	X	X	X		X	X		
	FIS - Shared Storage Units (voluntary)			0,5										1		
	FIS - Shared Storage Units (mandatory)															✓

Table 3.1: Contribution of this paper compared to the literature listed in Table 3.1.

		Méndez et al. (2006a)	Ierapetritou and Floudas (1998a)	Göthe-Lundgren et al. (2002)	Lin and Floudas (2001)	Castro et al. (2004)	Ierapetritou et al. (1999)	Reddy et al. (2004)	Mockus and Reklaitis (1999)	Méndez et al. (2006b)	Wenkai et al. (2002)	Moro and Pinto (2004)	Lee et al. (1996)	Kondili et al. (1993)	Entrup et al. (2005)	Contribution of this paper
	ZW	1			0,5	1								1		✓
5	**Material transfer**															
	Instantaneous (neglected)	X												X		✓
	Time-cons. - No-resources															
	Time-cons. - Pipes							0,5			1	0,5				
	Time-cons. - Vessels (Pipeless)															
6	**Batch-size**															
	Fixed	X						X								
	Variable (Mixing and splitting)	1	1		1	1	1	1	1					1		✓
7	**Batch processing time**															
	Fixed - Unit independent								X					X		
	Fixed - Unit dependent	X							X					X		
	Variable - (unit/batch-size dependent) range		0,5		0,5	1	1	1						0,5		
	Variable - (unit/batch-size dependent) single value															✓
8	**Demand patterns**															
	Due dates - Single product demand	X														
	Due dates - Multiple product demands	0,5		1			1	1	1	1				1	1	✓
	Scheduling horizon - Fixed requirements	X	X		X			X								

Table 3.1: Contribution of this paper compared to the literature listed in Table 3.1.

		Méndez et al. (2006a)	Ierapetritou and Floudas (1998a)	Göthe-Lundgren et al. (2002)	Lin and Floudas (2001)	Castro et al. (2004)	Ierapetritou et al. (1999)	Reddy et al. (2004)	Mockus and Reklaitis (1999)	Méndez et al. (2006b)	Wenkai et al. (2002)	Moro and Pinto (2004)	Lee et al. (1996)	Kondili et al. (1993)	Entrup et al. (2005)	Contribution of this paper
	Scheduling horizon - Min/Max requirements	X														
9	**Changeovers**															
	None	X								X						
	Unit dependent	X						X			X					
	Sequence dependent - Product dependent								1					1	0,5	✓
	Sequence dependent - Product and unit dependent													X		
10	**Resource constraints**															
	None (only equipment)	X					X	X		X						✓
	Discrete	0,5					0,5		0,5					0,5		
	Continuous						0,5		0,5					0,5		
11	**Time constraints**															
	None	X						X		X						✓
	Non-working periods															
	Maintenance								1					1		
	Shifts															
12	**Costs**															
	None	X														
	Equipment															
	Utilities													X		
	Inventory							1						1		✓
	Changeover							1								
13	**Degree of certainty**															
	Deterministic	1	1	1	1		1	1	1	1				1	1	✓
	Stochastic															

Table 3.1: Contribution of this paper compared to the literature listed in Table 3.1.

	Méndez et al. (2006a)	Ierapetritou and Floudas (1998a)	Göthe-Lundgren et al. (2002)	Lin and Floudas (2001)	Castro et al. (2004)	Ierapetritou et al. (1999)	Reddy et al. (2004)	Mockus and Reklaitis (1999)	Méndez et al. (2006b)	Wenkai et al. (2002)	Moro and Pinto (2004)	Lee et al. (1996)	Kondili et al. (1993)	Entrup et al. (2005)	Contribution of this paper
14 Storage time															
Maximum shelf-life														0,5	✓
Quarantine time														0,5	
TOTAL	6,0	4,5	2,5	5,0	4,0	6,0	7,5	7,0	4,0	2,0	0,5	1,0	**12**	3,5	

3.5. Conclusion

The introduction of computer algorithms improved the production scheduling and resulted in the linear programming technique that is nowadays used to solve many scheduling problems. It is a research area that has received a lot of attention in the last few decades, where different representations of time, different representations of scheduling problems and the different areas of applications are the subject of the investigation. Time could be represented in a continuous or discrete manner, the scheduling problems could be represented in a linear or nonlinear, integer, continuous or mixed integer manner and examples of areas of application are crude oil, crude oil blending and pipeline scheduling. Based on the great amount of researches, classifications of the researches can be made and general frameworks can be developed. From the general frameworks, more specific requirements required by the industries such as multiple product demand are developed and implemented.

A selection of fourteen papers has been made out of all the literature resources discussed at the beginning of this chapter that were considered to be the most promising in describing a basic mathematical model. Both the scheduling optimization problem of Sime Darby Oils Zwijndrecht Refinery and the selected papers were submitted to a problem classification described by Méndez et al. (2006a), based on which the mathematical optimization model discussed by Kondili et al. (1993) has been chosen as the basic mathematical model.

4

The scheduling optimization model

In Section 3.4, a mathematical model has been chosen that will be used as the basic mathematical model. This mathematical model has been developed by Kondili et al. (1993). In this chapter, the basics of this mathematical model will be elaborated, after which adaptions and improvements will be made in order to meet the problem description of Sime Darby Oils Zwijndrecht Refinery (SDOZR) best. In order to achieve that, a list with parameters, variables and indices that are used for the formulation of the mathematical model is presented in Section 4.1 first. Then the objective function of the mathematical model is formulated and explained in Section 4.2. Finally, Sections 4.3 and 4.4 give the constraints accompanied with a short explanation.

4.1. Nomenclature

4.1.1. Parameters

i	Standard subscript for processing tasks; $i \in I_j$.
j	Standard subscript for equipment units; $j \in K_i$.
k	Standard subscript for a family of tasks; $k \in NF_j$.
l	Standard subscript for a family of tasks; $l \in NF_j$, $l \neq k$.
t	Standard subscript for absolute time. Relative to the start of the horizon; $t \in H$.
s	Standard subscript for states.
H	Number of time intervals. The length of the time interval is taken to be the highest common factor of the processing times involved in the problem.

4.1.2. Decision variables

B_{ijt}	Amount of material which starts undergoing task i in unit j at the beginning of time period t.
BS_{st}	Equals 1 if state s is stored at the beginning of time t.
R_{st}	Amount of material of feed state s received from external sources at time t.
S_{st}	Amount of material stored in state s, at the beginning of time period t.
W_{ijt}	Equals 1 if unit j starts processing task i at the beginning of time period t.

4.1.3. Sets

I_j	Set of processing tasks that can be performed by unit j.
$I_j^{(k)}$	Set of processing tasks which can be performed by unit j and belong to family k.
K_i	Set of units capable of performing task i.
NF_j	Number of disjoint families of tasks on unit j.
S_{in}	Set of input states, the states that are required to start a task i.
S_{out}	Set of output states, the states that can be sold.
S_i	Set of input states of task i.
$\overline{S}_i$	Set of output states of task i.
T_s	Set of tasks requiring material from state s.
$\overline{T}_s$	Set of tasks producing material in state s.
SSL	Set of states that can be stored and are subjected to a shelf-life of sl_s time periods.

4.1.4. Variables

C_s	Maximum storage capacity dedicated to state s.
C_{st}	Unit cost or price of material in state s at time t.
C_{st}^S	Running cost of keeping in storage a unit of material of state s at time t.
D_{st}	Amount of material in product state s due for delivery at time t.
p_i	Processing time of task i.
p_{is}	Processing time for the output of task i to state $s \in \overline{S}_i$.
S_{st_0}	Amount of material stored in each state s, at the beginning of time period $t = t_0$.
sl_s	Shelf-life time for state s.
$V_{ij}^{max}/V_{ij}^{min}$	Maximum/minimum capacity of unit j when used for performing task i.
ρ_{is}	Proportion of input of task i from state $s \in S_i$.
$\overline{\rho}_{is}$	Proportion of output of task i in state $s \in \overline{S}_i$.
τ_{jkl}	Cleaning time required when a task of family l is performed after a task of family k, both in unit j.

4.2. Objective function

The goal of the mathematical model described by Kondili et al. (1993) is to maximize the profit. The definition of the profit for the SDOZR situation is slightly adapted to the objective function given by Kondili et al. (1993) and is given in Equation 4.1. In Equation 4.1b, $H+1$ represents the end of the time horizon. With this objective function, it is possible to implement several requirements as desired by the real-world situation. For example, if it is not desirable for certain states of material to be left in storage at the end of the time horizon, the corresponding values of $C_{s,H+1}$ can be set to large negative values. It is also possible to explicitly add a constraint stating that the storage amount of that particular state is zero at time period $H+1$: $S_{s,H+1} = 0$. In Equation 4.1c it is assumed that the initial storage level at the start of the time horizon for each state, S_{s0}, is known in advance and therefore is a constant.

$$Profit = Value\,of\,products - Cost\,of\,feedstock - Cost\,of\,storage \tag{4.1a}$$

$$Value\,of\,products = \sum_{s}\left(C_{s,H+1}S_{s,H+1} + \sum_{t=1}^{H} C_{st}D_{st}\right) \tag{4.1b}$$

$$Cost\,of\,feedstock = \sum_{s}\left(C_{s0}S_{s0} + \sum_{t=1}^{H} C_{st}R_{st}\right) \tag{4.1c}$$

$$Cost\,of\,storage = \sum_{s}\sum_{t=1}^{H} C_{st}^{S}S_{st} \tag{4.1d}$$

4.3. Primary constraints

In mathematical modelling, there are three fundamental or primary constraints. The primary constraints in this section are described by Kondili et al. (1993). The first primary constraint concerns the allocation constraint represented by Equation 4.2, where M represents a very large number. Any item of equipment, or unit, in the refinery can only start and perform at most one task at the same time. The next task to be performed in the same unit can only start after the current task has been finished. The constraint is only binding if $W_{ijt} = 1$, forcing all other $W_{i'jt'}$ to be zero.

$$\left(\sum_{i' \in I_j}\sum_{t'=t}^{t+p_i-1} W_{i'jt'}\right) - 1 \leq M\left(1 - W_{ijt}\right) \qquad \forall j,\, t,\, i \in I_j \tag{4.2}$$

The second fundamental constraint covers the capacity limitations of the units and storage. The amount of material that can be processed at time t depends on the combination of task i and unit j, as shown in Equation 4.3a. The maximum and minimum capacity of each combination of task i and unit j, V_{ij}^{max} and V_{ij}^{min} respectively, is known in advance. The constraint forces the batch-size B_{ijt} to be zero if $W_{ijt} = 0$. Similarly, the amount of material stored in a state s is not allowed to exceed the maximum storage capacity C_s. Equation 4.3b represents this constraint. However, this constraint only takes into account the situation where the storage for each state s is dedicated to that particular state, the so called Unlimited Intermediate Storage (UIS) policy - when C_s equals large numbers - or the so called Finite Intermediate Storage (FIS) policy for dedicated storage units - when C_s is small enough to limit the amount of storage. Other storage policies to be implemented in this mathematical model are elaborated in Section 4.4.2.

$$W_{ijt}V_{ij}^{min} \leq B_{ijt} \leq W_{ijt}V_{ij}^{max} \qquad \forall i,\, t,\, j \in K_i \tag{4.3a}$$

$$0 \leq S_{st} \leq C_s \qquad \forall s,\, t \tag{4.3b}$$

The third fundamental constraint in mathematical modelling concerns the material balance within the model. The material balance used within the current model is represented by Equation 4.4. Here, the storage level of state s at time t (S_{st}) equals the storage level of the same state s at time $t-1$ (S_{st-1}), added to the amount of state s produced at time t, subtracted with the amount of state s used at time t, added to the amount of state s delivered from an external supplier (R_{st}) at time t, subtracted with the amount of state s that was delivered to an external customer (D_{st}) at time t. The initial storage level of state s at time $t = t_0$, S_{st_0}, is known in advance.

$$S_{st} = S_{s,t-1} + \sum_{i \in \overline{T}_s}\left(\overline{\rho}_{is}\sum_{j \in K_i} B_{ij,t-p_{is}}\right) - \sum_{i \in T_s}\left(\rho_{is}\sum_{j \in K_i} B_{ijt}\right) + R_{st} - D_{st} \qquad \forall s,\, t \neq t_0 \tag{4.4}$$

4.4. Secondary constraints

Apart from the primary constraints, mathematical models might also have additional or secondary constraints that represent among others requirements or limitations that emerge from the situation to be described or modelled. One could think of temporary unavailability of units, the use of a different inventory storage policy, limited equipment connectivity, batch-size-dependent processing times, production order due to possible contamination and a maximum shelf-life. In this section, these secondary constraints will be addressed and explained.

4.4.1. Temporary unavailability of units

So far, it has been assumed that the considered units will be available throughout the whole time horizon. However, it can occur that due to, for example, maintenance or non-working periods a unit is not available for some time during the time horizon. A way to deal with this is to assign the value zero to an appropriate subset of W_{ijt} variables (Kondili et al., 1993).

If equipment item j is unavailable between times t_1 and t_2 $(t_2 > t_1)$, then it is not able to start processing any task at the start of the intervals t_1 to $t_2 - 1$. Furthermore, the item must already be idle at time t_1, which implies that it must not have started any task i at any time after $t_1 - p_i + 1$ (Kondili et al., 1993).

$$W_{ijt} = 0 \qquad \forall i \in I_j,\ t = t_1 - p_i + 1,\ \ldots,\ t_2 - 1 \tag{4.5}$$

4.4.2. Shared storage units

In this section about the mathematical model, the ability for storing states has been addressed several times. The maximum storage capacity has been mentioned in Equation 4.3b and the amount of stored material is included in the material balance of Equation 4.4. The inventory storage policy assumed and described so far can be classified as an Unlimited Intermediate storage (UIS) policy and a FIS policy with dedicated storage units. As classified by Figure 3.5, the situation at SDOZR asks for a Zero Wait (ZW) policy and a Finite Intermediate Storage (FIS) policy with shared storage units.

A ZW storage policy can be easily implemented by not allowing the mathematical model to store any of the particular state at all. This can be accomplished by setting the maximum storage capacity C_s for that state to zero (Kondili et al., 1993).

The FIS policy with shared storage units has to be implemented in a different way. So far a State-Task-Network (STN) for the processes of this mathematical model could generally be described by an STN shown in Figure 4.1 (Kondili et al., 1993).

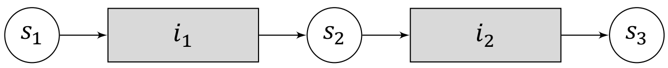

Figure 4.1: A simplified straight forward STN (Kondili et al., 1993).

In the situation where a mathematical model needs to take into account the FIS policy with shared storage units, a storage task is created that produces the same state and the same amount as it consumes. This will take exactly one time interval and the task can be performed by a shared storage unit. The corresponding STN will then look like the STN shown in Figure 4.2. The storage capacity C_s for this state should only include the capacity that is specifically dedicated to this state (Kondili et al., 1993).

It should be noted that the storage policy described above is a policy where storage is voluntary. However, there might be situations where a state is to be stored mandatory before it can be used as an input state for another task. In that case, the method described by Kondili et al. (1993) should be slightly adapted to a situation where there is a 'pre-storing' state and a 'after-storing' state. The method described above can then be applied to the after-storing state.

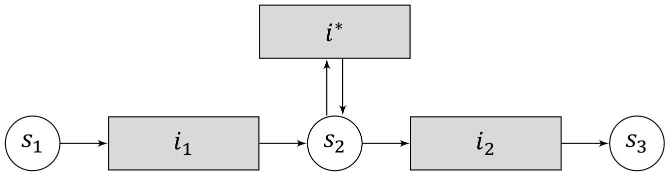

Figure 4.2: A simplified STN representing a Finite Intermediate storage policy with shared storage units (Kondili et al., 1993).

4.4.3. Limited equipment connectivity

In a real-world problem, usually not all equipment is connected to each other as it has been assumed so far. Often there are production lines or there is one single station to be addressed by each product. In the case of SDOZR - as has been discussed in Section 2.2 - there is a mix of both. However, it is not necessary to implement extra constraints to the mathematical model to be able to take this into account. The mathematical model deals with this requirement by splitting both task i_1 and i_2 in tasks i_{11}, i_{12}, i_{21} and i_{22} respectively and by splitting state s_2 in state s_{21} and s_{22}. The values of variables p_{is}, ρ_{is} and $\overline{\rho}_{is}$ link for example task i_{11} and state s_1 and task i_{12} and state s_{22} to each other in the same way they do for the original task i and state s. The STN showing the described principle is shown in Figure 4.3 (Kondili et al., 1993).

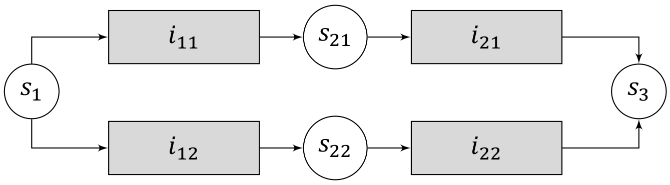

Figure 4.3: A simplified STN representing limited equipment connectivity (Kondili et al., 1993).

4.4.4. Batch-size dependent processing times

So far, it has been assumed that the processing times of the tasks are fixed and do not depend on the batch-size. This situation has been illustrated by the red line plotted in Figure 4.4, where the batch-size has been plotted in relation to the processing time.

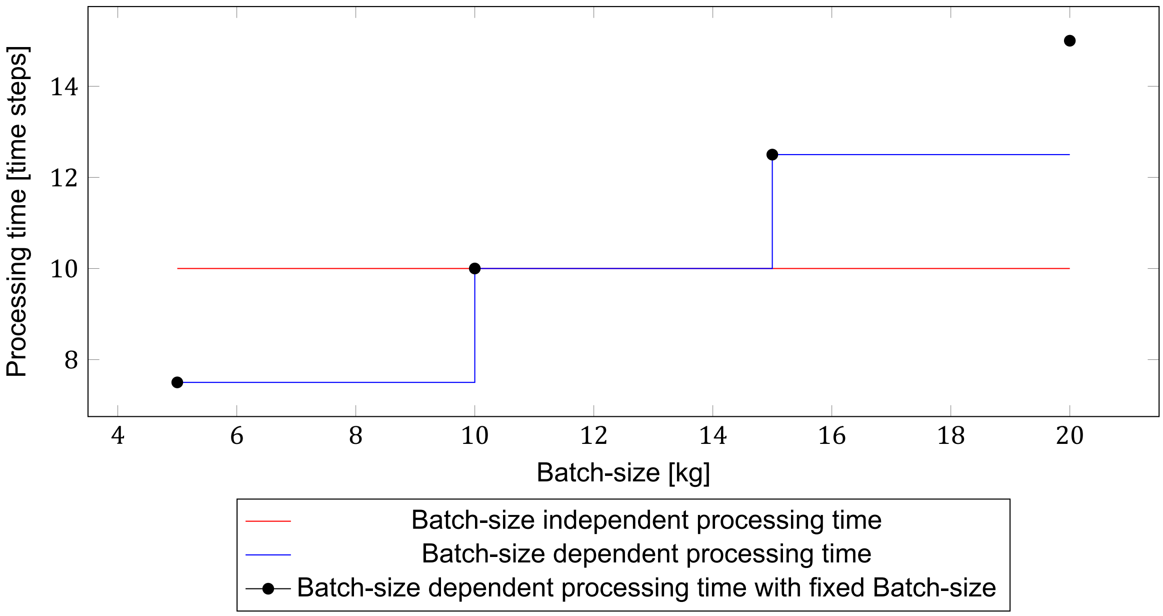

Figure 4.4: Batch-size independent processing time vs. Batch-size dependent processing time [Own visualisation].

However, there are operations for which this assumption does not hold, such as filtration. The blue line in Figure 4.4 shows this alternative case, where the processing time does depend on the batch-size (Kondili et al., 1993). This case can be approximated by a so called piecewise constant function (Kondili et al., 1993). Mathematically it can be written down as follows:

For batch-size B	∈	$[0, B^1]$	Processing time p^1
For batch-size B	∈	$(B^1, B^2]$	Processing time p^2
For batch-size B	∈	$(B^2, B^3]$	Processing time p^3
For batch-size B	∈	$(B^3, B^4]$	Processing time p^4

Each of the instances written above can be seen as a separate task. Just as with the limited equipment connectivity described in the previous paragraph, the tasks performs the same action, but have different batch-sizes - and thus minimum and maximum capacities V_{ij}^{min} and V_{ij}^{max} and different processing times. The STN visualising this situation is shown in Figure 4.5 (Kondili et al., 1993).

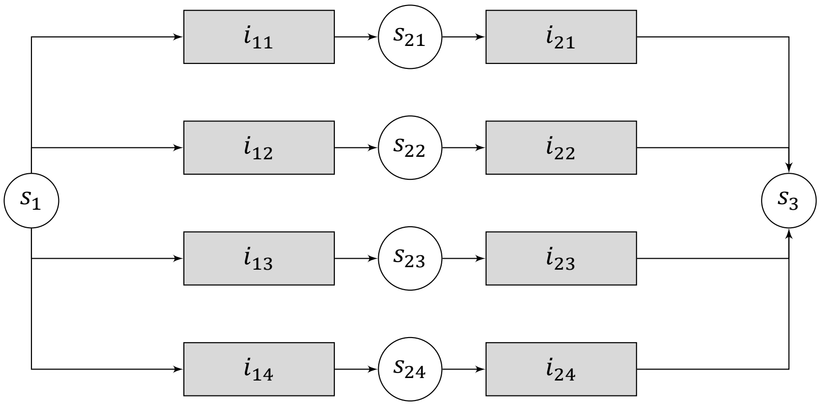

Figure 4.5: A STN representing batch-size dependent processing times (Kondili et al., 1993).

It is also possible that the batch-size can only take certain values. For example the batch-size can only be 5, 10, 15 or 20 kg and no value in between. In that case, the method described by Kondili et al. (1993) should be slightly adapted to a situation where the minimum and maximum capacities V_{ij}^{min} and V_{ij}^{max} will be equal to each other for each task. The mathematical formulation can then be written as follows:

For batch-size B	∈	$[B^1, B^1]$	Processing time p^1
For batch-size B	∈	$[B^2, B^2]$	Processing time p^2
For batch-size B	∈	$[B^3, B^3]$	Processing time p^3
For batch-size B	∈	$[B^4, B^4]$	Processing time p^4

4.4.5. Production order

As has been described in Section 2.2.5, contamination is a restriction of the refinery that needs to be taken into account. The situation where big problems occur when the end product requires a low C12:0 content and the non-lauric (low C12:0) product is processed after a lauric (high C12:0) product is comparable with the dye example that is often used to explain similar situations (Kondili et al., 1993). This dye example is used to describe the need of the cleaning of equipment in some, but not all, situations. In a dye manufacturing plant, little or no cleaning is required when a dark paint is produced after a light paint. However, when a light paint needs to be produced after a dark paint, extensive cleaning is needed. This concept is described as sequence dependent cleaning.

In mathematical modelling, sequence dependent cleaning can be approached in two ways. For the first method, an actual cleaning task is created, for which the model should decide when the task needs to be performed. For the second method, only the time required for the cleaning task is taken into account. For both methods, I_j is split into NF_j disjoint families I_j^k, $k = 1, ..., NF_j$. Then a (virtual) cleaning task i_{jkl} is created with a processing time of τ_{jkl}, where $l = 1, ..., NF_j$, $l \neq k$.

For the situation at SDOZR, the second method is assumed to be sufficient. More information on the first method can be found in Kondili et al. (1993). For the second method, it is sufficient to add a constraint stating that when unit j has processed a task of family k, no task of family l is started within

τ_{jkl} time units after the task of family k in unit j has finished. This constraint is similar to the allocation constraint of Equation 4.2 and is written as follows:

$$\sum_{i' \in I_j^{(l)}} \sum_{t'=t+p_i}^{t+p_i+\tau_{jkl}-1} W_{i'jt'} \leq M\left(1 - W_{ijt}\right) \qquad \forall j, i \in I_j^{(k)}, t \tag{4.6}$$

The concept described here for sequence dependent cleaning can also be applied to the contamination restriction discussed earlier. In Figure 3.10 this situation is referred to as a Product Dependent Changeover. The application of the constraint works the same. Namely, for an oil of family l - for example the family of non-lauric oils with a C12:0 specification - that is processed after an oil of family k - for example the family of lauric oils, the value of 'cleaning-time' τ_{jkl} can be made sufficiently large to force the model not to create a production order where a product of family l follows a product of family k. With the presence of the third family of products, the non-lauric oils without the C12:0 specification, an ideal production order can be created.

4.4.6. Maximum (and minimum) storage time

The oils that are processed by SDOZR can deteriorate. To be more precise, before an oil (or state s) can be pumped to the equipment to be processed or can be marked 'ready-for-delivery', various analysis are required. These analysis take time, are expensive and are only valid for a certain amount of time. If the time is expired a new analysis should be performed. Therefore, it is assumed for this mathematical model that the time each result of the analysis is valid is considered as a maximum storage time or shelf-life. Entrup et al. (2005) discusses a maximum shelf-life application in mathematical modelling. For the products considered by Entrup et al. (2005) deterioration is assumed to happen gradually. However, as the description of the maximum shelf-life for SDOZR states, the products produced by SDOZR deteriorate from one moment to the next. Therefore, the maximum shelf-life as it has been modelled by Entrup et al. (2005) can not be applied here. Instead, two additional constraints are formulated. First, the storage time should be tracked. This is done by adding a decision variable to the model, BS_{st}, which is equal to 1 when $S_{st} > 0$ as described by Equation 4.7. It should be noted that this constraint does not force BS_{st} to be 0 when $S_{st} = 0$. Second, the number of time periods t should not exceed the maximum shelf-life of the oil or state s, sl_s. This is covered by the constraint shown by Equation 4.8. This equation states that in the range of time periods from t up until $t + sl_s + 1$ the number of BS_{st} is not allowed to exceed the maximum shelf-life sl_s. Because the time range is one time period larger than the shelf-life itself, at least one of the BS_{st} in this range of time periods is forced to be zero, forcing the mathematical model not to exceed the shelf-life of a state s. This way of monitoring does not take into account different tanks if there are any. As long as a state is stored, the sum of the amount of storage actions will increase. The sum will only be interrupted and reset when there is at some point no storage of that particular state.

$$S_{st} = BS_{st} \cdot S_{st} \qquad \forall s, t \tag{4.7}$$

$$\sum_{t}^{t+sl_s+1} BS_{st} \leq sl_s \qquad \forall s \in SSL \tag{4.8}$$

Besides the maximum storage time or shelf-life, SDOZR also has to deal with a minimum storage time or quarantine time. This is related to the analysis described earlier in this paragraph. The quarantine time is used to cover the time that is required to perform such an analysis. The oils should be in storage for a minimum amount of time (the quarantine time) so that there is enough time to perform the analysis, something that should be included in the production schedule as well. Something similar has been discussed by Baart (2020), but the intention in this paper was to not use a separate task for performing the required analysis. This because it creates extra states s and tasks i, making the size of the model increase. Unfortunately, the writer of this report did not manage to implement this concept. The first thought was to expedite the prescribed delivery time D_{st} by the quarantine time. However, this does not change the situation since the delivery can still be removed from the model before the analysis has been performed.

4.5. Conclusion

The mathematical model developed in this chapter consists of an objective function, a set of primary constraints and a set of secondary constraints. The objective aims for a maximization of profit based on the value of products, the cost of feedstock and the cost of storage. The constraints that are considered to be primary include the allocation constraint, the material balance, the capacity limitations on the units and the storage limitations in case of either Unlimited Intermediate Storage or Finite Intermediate Storage for dedicated units. To these primary constraints, secondary constraints have been added that emerged from the situation to be described or modelled. This concerns the temporary unavailability of units, limited equipment connectivity and production order. All constraints mentioned so far were derived from the mathematical model discussed by Kondili et al. (1993). In addition to this list, two constraints have been based on the constraints elaborated by Kondili et al. (1993), but were expanded to fit the situation to be described or modelled better. This concerns the batch-size dependent processing times and the shared storage units. The constraint on batch-size dependent processing times elaborated by Kondili et al. (1993) discusses ranges of batch-sizes corresponding to a certain processing time. However, the situation of Sime Darby Oils Zwijndrecht Refinery requires batch-size dependent processing times on single values of batch-sizes. The constraint on shared storage units discussed by Kondili et al. (1993) is elaborated as being a voluntary storage action. However, in the situation of Sime Darby Oils Zwijndrecht Refinery a storage action is mandatory. Both situations have been implemented in the model developed in this chapter. Last, Kondili et al. (1993) did not cover a constraint on maximum shelf-life. Therefore, an extra constraint has been developed in this chapter.

5 Experimental study

This chapter will go more into detail on the scheduling optimization problem of Sime Darby Oils Zwijndrecht Refinery (SDOZR) with respect to the mathematical model developed in Chapter 4. Therefore, a so called 'basic model' will be elaborated in Section 5.1. This includes among others the equipment, the states and tasks, the scheduling horizon, the demand and the processing times. The results of this basic model will be discussed in Section 5.2. The basic model will be validated in Section 5.3, after which some of the findings will be further investigated by means of a sensitivity analysis elaborated in Section 5.4. The results from the experiments will be compared with the results of the basic model.

5.1. Building the scheduling optimization model for Sime Darby Oils Zwijndrecht Refinery

For the mathematical model that describes the situation at SDOZR, the information gathered in the previous chapters has been extensively analysed. Based on this information, the SDOZR scheduling problem can be implemented. Section 5.1.1 will discuss the equipment that will be implemented in the mathematical model for SDOZR. Section 5.1.2 will discuss the states and tasks used to model Bleaching Line (BL) [...] and the Chemical Interesterification (C. I.), followed by a subdivision of the states in a product family which is discussed by Section 5.1.3. The scheduling horizon H considered for the mathematical model and also the demand will be discussed in Section 5.1.4 and 5.1.5 respectively. Section 5.1.6 will discuss the batch-size dependent processing times, whereas Section 5.1.7 will provide additional notes that need to be made to set up the mathematical model. The resulting mathematical model from this section will be named 'basic model' in the remainings of this report.

5.1.1. Equipment

As discussed in Section 1.2, BL [...] and the C. I. will be analysed for the SDOZR problem. Sections 2.1.3 and 2.2.5 elaborated this part of the refinery more extensively and visualised this in Figure X. This figure is also used to define the equipment items to be implemented in the mathematical model for the situation at SDOZR. Simplifications have been applied to this figure to ease the implementation of the problem into the mathematical model and to keep the size of the model as small as possible. The result is shown in Figure X. This figure is a copy of Figure X, but without DEO [...]. The contours of Figure X are faded to the background and blue, white and orange rectangles are placed over it. These rectangles represent the unit of equipment that will be implemented in the mathematical model as a replacement for the real world piece(s) of equipment. The equipment of sub-line [...] has been replaced with a piece of equipment named [...]. This single unit represents the bleaching tank, the filter department and the buffer that the sub-line exists of. The same holds for sub-line [...], for which the replacing piece of equipment is named [...]. Sub-line [...] is replaced by three pieces of equipment. The first piece of equipment is named [...] and represents the bleaching tank. The next piece of equipment, [...], represents the filter department of sub-line [...] and [...] represents the buffer. The reason why the sub-lines are modelled in this way is that the first [...] sub-lines don't have any interaction with the other sub-lines, whereas sub-line [...] shares its filter department with the C. I. cauldrons named [...]. A more detailed description on the sharing of the filter department was given in Section 2.1.3. The last piece of equipment to be implemented is the buffer of the C. I., named [...]. For the implementation in the mathematical model the minimum and maximum capacity of each piece of equipment is required, which are given in Table X.

5.1.2. Five categories for Bleaching Line and Chemical Interesterification

The oils that are processed by BL [...] can be divided in five categories regarding the equipment required for the production of its end product. A short overview of the categories is given below and each category is discussed in one of the paragraphs. For each category an example end product is given, based on a real world end product produced by SDOZR. The decision for choosing these five end products is based on the fact that all five categories discussed in this section should be represented by one of the end products. Second, the three families of products, that have been discussed in Section 2.2.5 and will be further elaborated for the SDOZR mathematical model in Section 5.1.3, should all be represented by the end products as well. At last, the amount of data that was available at SDOZR in order to determine the data required by the mathematical model, such as p_i and p_{is}, was also key in the decision.
Each paragraph describing a category is accompanied with a State-Task-Network (STN) visualising the category. For each real world product that is used as an example it has been assumed that there is only one way to produce it, namely the way described in the paragraphs.

Five categories for BL [...] and C. I.:

1. Only bleaching on BL [...]
2. C. I. without filtering and bleaching on BL [...]
3. C. I. with filtering and bleaching on BL [...]
4. C. I. without filtering and bleaching on BL [...]
5. C. I. with filtering and bleaching on BL [...]

Category 1

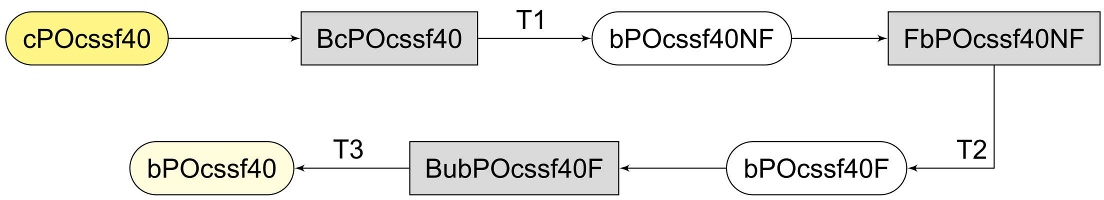

Figure 5.1: The STN representing Category 1 [Own visualisation].

Figure 5.1 shows the STN of the Category 1 - product, that only needs bleaching on BL [...]. The oval shapes represent the types of oil or states and the gray rectangles represent the tasks that have to be performed. The tasks can be linked to the equipment shown in Figure X. Bleaching tasks can be performed on the Bleaching tanks, the filtering tasks are performed on the filtering department and the buffering tasks are performed on the buffers. The yellow state represents the input material. For the mathematical model, this state is delivered from an external resource (R_{st}). The white states represent the intermediate states and the light yellow state represents the end product. These two states cannot be delivered from an external resource. The white states are subjected to a Zero Wait (ZW) storage policy, whereas the light yellow states are subjected to an Unlimited Intermediate Storage (UIS) policy. As can be seen in Figure 3.5 and has been discussed in Section 3.3, the storage policy at SDOZR is actually a Finite Intermediate Storage (FIS) policy with shared storage units. However, since the ROS is not modelled in this mathematical model, the assumption is that there is always sufficient availability of storage. The maximum storage capacity, C_s, is set to a sufficiently large number to simulate the UIS policy. For the ZW storage policy, C_s is set to a value of zero. A FIS policy with shared storage units also applies to the buffers of BL [...] and the C. I.. To be more precise, it concerns a mandatory FIS policy as it has been described in Section 4.4.2. In this particular category, state 'bPOcssf40F' is the 'pre-storing' state, whereas 'bPOcssf40' is the 'after-storing' state on which the voluntary FIS policy programming method can be applied. In reality, the buffering takes place before the oil is deodorized by DEO [...]. The buffering time depends on the characteristics and process restrictions of DEO [...]. Because of this dependent storage time determined by DEO [...], the voluntary FIS policy is not implemented in this mathematical model.

Usually, a STN is provided with both the time required to produce a state, the percentage of the state that enters a task and the percentage of the state that leaves a task. In this and the following figures the percentage of the state that enters or leaves a task is 100%, unless stated otherwise. The processing times required to produce a state are shown in this and the following figures by a T plus ascending numbers. The reason no actual numbers are given has to do with the batch-size dependent processing times as will be discussed in Section 5.1.6.

The names of the tasks mentioned in the rectangles for tasks are structured as follows. The first letter or letters are an abbreviation of the task. This abbreviation is followed by the state the task has an effect on. For example, the first task mentioned in Figure 5.1 is 'BcPOcssf40'. The abbreviation here is 'B' - which stands for Bleaching - and the state the task has an effect on is 'cPOcssf40'. The names of the states are structured in a similar way. Here, the state is followed by an abbreviation. For example, the second state in Figure 5.1 is bPOcssf40NF. The state is 'bPOcssf40' and the abbreviation is 'NF'. This stands for 'Not Filtered', meaning that the state should be filtered by the next task. The next state, 'bPOcssf40F', has the abbreviation 'F' which stands for 'Filtered'. This means that state bPOcssf40 has been filtered. All the abbreviations used in this and the following figures are listed in Table 5.1.

Table 5.1: Abbreviations of the State-Task Networks

Abbreviation	Explanation
B...	Bleaching
F...	Filtering
Bu...	Buffering
CI...	Chemical Interesterification
...R	Raw material of the product mentioned in advance
...NF	Not Filtered
...F	Filtered
...NBu	Not Buffered

Category 2

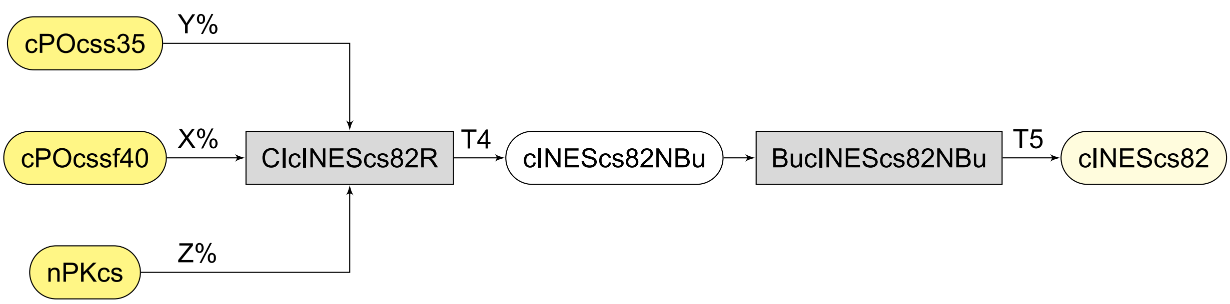

Figure 5.2: The STN representing Category 2 [Own visualisation].

Figure 5.2 shows the STN of the Category 2 - product, that needs C. I. without filtering and is bleached on BL [...]. Since BL [...] is not in the scope of this report, it is assumed that the input state for BL [...] - cINEScs82 - is equal to the output state of the C. I. and the end of the production process for this scope. In this STN, it can be seen that there are multiple input states that are a part of the total input, expressed in percentages, just as it has been discussed in the previous paragraph. Because there are multiple input states, the method of 'Abbreviation - state of task application' described earlier for the first task does not hold. Therefore, the end state is used as the state on which the task is applied, followed by the abbreviation 'R', which stands for raw material.

Category 3

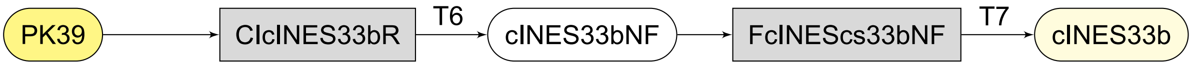

Figure 5.3: The STN representing Category 3 [Own visualisation].

Figure 5.3 shows the STN of the Category 3 - product, that needs C. I. with filtering and is then bleached on BL [...]. In a similar way as with the Category 2 - product, the input state for BL [...] - cINES33b - is equal to the output state of the C. I. and the filters of sub-line [...] of BL [...].

Category 4

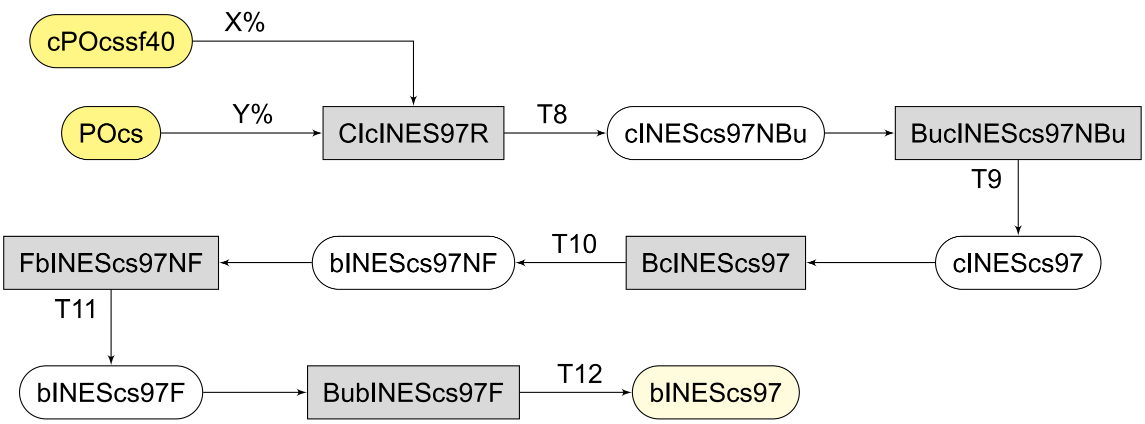

Figure 5.4: The STN representing Category 4 [Own visualisation].

Figure 5.4 shows the STN of the Category 4 - product, that needs C. I. without filtering and is bleached on BL [...].

Category 5

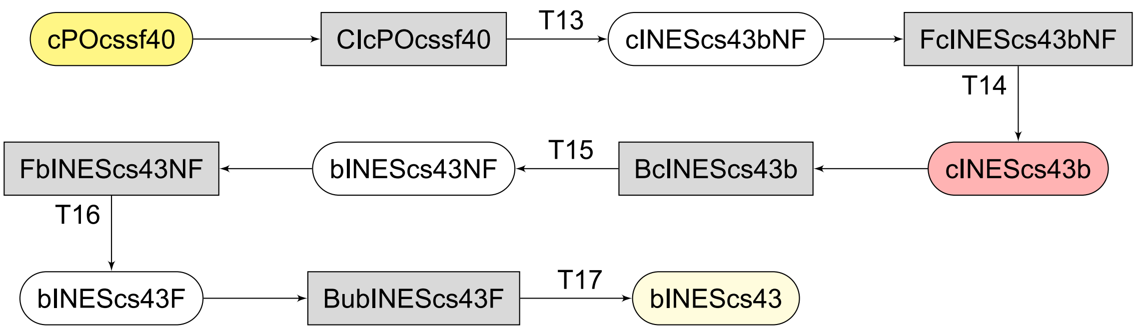

Figure 5.5: The STN representing Category 5 [Own visualisation].

Figure 5.5 shows the STN of the Category 5 - product, that needs C. I. with filtering and is then bleached on BL [...]. The red state in this STN represents an intermediate state that has a FIS policy for dedicated storage units.

Bleaching on sub-line of Bleaching Line

The STN's described in the previous paragraphs that require bleaching on BL [...] are all based on the fact that they will be bleached on the third sub-line of BL [...]. However, as shown by Figure X, there are two more sub-lines on BL [...]. The STN's for these sub-lines can be highly simplified, since there is no interaction with any other sub-line as is the case with sub-line [...]. All separate tasks of sub-line [...] are merged into one task and the production time required for each end product for each single task is summed up. The input states and output states remain the same. The resulting STN is shown in Figure 5.6.

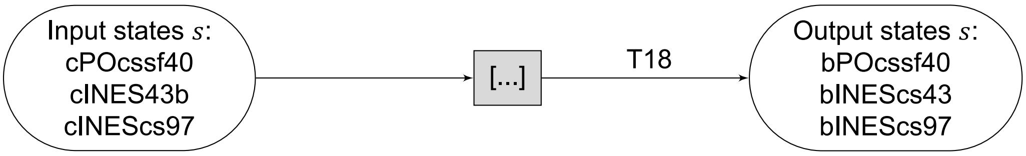

Figure 5.6: The STN representing the production of states on sub-line of Bleaching Line [Own visualisation].

5.1.3. Three families of products

Is has been discussed in Section 2.2.5 already that the products processed at SDOZR can be divided into three groups or families: the lauric products, the non-lauric products without a C12:0 specification and the non-lauric products with a C12:0 specification. The end products discussed in Section 5.1.2 can be classified to one of these families. An overview is given in Table 5.2.
Table 2.1 shows what product of a family can be processed after the product of what other family without a cleaning operation between the processes. Products of Family 2 are not allowed to be processed after products of Family 0 if there is no cleaning operation in between. Therefore, the cleaning time between a task of Family 2 and a task of Family 0 (τ_{jkl}) is set to five time steps. The cleaning time is assumed to be the same for all units. An overview for all processing orders is given in Table 5.3.

Table 5.2: Classification of end products to product families

Lauric (Family 0)	Non-lauric, no C12:0 (Family 1)	Non-lauric, C12:0 (Family 2)
cINEScs82	bPOcssf40	bINEScs97
cINES33b	bINEScs43	

Table 5.3: Cleaning time required when a task of Family l (column) is performed after one of Family k (row) in unit j (τ_{jkl})

	Lauric (0)	Non-lauric, no C12:0 (1)	Non-lauric, C12:0 (2)
Lauric (0)	0	0	5
Non-lauric, no C12:0 (1)	0	0	0
Non-lauric, C12:0 (2)	0	0	0

5.1.4. Scheduling horizon

It has been discussed in Section 2.3 that the schedulers of SDOZR currently create a schedule with a scheduling horizon of [...]. Since the mathematical model introduced in Chapter 4 is a discrete time model, this scheduling horizon should be divided in a number of equal time steps. Based on the information available within SDOZR on the planned shipments it has been decided to set one time step in the mathematical model equal to 30 minutes in the real world. When a scheduling horizon of 10 days is assumed, this results in 480 time steps of 30 minutes. However, the decision has been made to use only half of the scheduling horizon while the model was build. So 5 days resulting in 240 time steps of 30 minutes. For the mathematical model with Objective 4.1, both scheduling horizons have been calculated to be able to make a comparison between the two scenario's on for example objective value and solution time.

5.1.5. Demand

In Section 3.3 the demand pattern of SDOZR problem has been classified as a 'Due date - Multiple product demand' demand pattern. The mathematical model takes this into account by the amount of material in product state s that is due for delivery at time t, D_{st}. The refinery of SDOZR is able to produce a great variety of products, way more than the five products that are now used as an example to represent the five categories as discussed in Section 5.1.2. In order to create a realistic demand for this mathematical model, the data of Bleaching Line [...] and the Chemical Interesterification is used.

The data available for these production lines includes among others the starting time, the finishing time, what oil is produced and the amount of oil that is produced. A random period of 10 consecutive days has been chosen and the finishing times for the products were converted to mach the division of time used for the mathematical model. Since the production lines produce more different sorts of oils than the five oils that are used as an example, the oils to produce were classified in one of the five categories. The oil was then implemented in the demand by the corresponding example product. The total list of input is given in Appendix C. This list has been sorted on time. It can be seen that not everything has been implemented in the final demand for the mathematical model. The list of demand has been randomly arranged and was not sorted on time at first. If the column named 'Implemented' states 'No, IA', it means that it was known in advance that this demand would not be implemented in the final demand, due to the fact that it would cause an infeasible solution. There would simply be not enough time to produce the product. If the column named 'Implemented' states 'No', it means that this demand is not taken into account eventually. This particular demand was listed at the end of the randomly arranged list and at some point, the model would give an infeasible solution. If the column named 'Implemented' states 'Yes', it means that the demand is implemented in the final demand. The list given in Appendix C provides the demand for a scheduling horizon of 10 days, consisting of 480 time steps. This list is used for the demand for the scheduling horizon of 5 days, consisting of 240 time steps. The latter is derived from the first one by removing the demand for any time step that exceeds the 240th time step. Eventually, approximately 54% and 52% of both the number of demands and the volume of demand shown in the list are implemented in the final mathematical model for the 480 time step scenario and the 240 time step scenario respectively.

5.1.6. Batch-size dependent processing times

The principle of batch-size dependent processing times has been discussed in Section 4.4. The batch-sizes at SDOZR can only take certain values. Depending on the equipment used, the batch-sizes can be [...]. As an example, the STN of Category 3 given by Figure 5.3 will be elaborated for the batch-size dependent processing times. Both tasks in this STN have to be divided in four sub-tasks in the same way as it has been elaborated in Figure 4.5. Each task will have its own processing time and its own minimum and maximum capacity V_{ij}^{min} and V_{ij}^{max} of unit j for that task, which will be equal to each other for each sub-task. The resulting STN is shown in Figure 5.7.

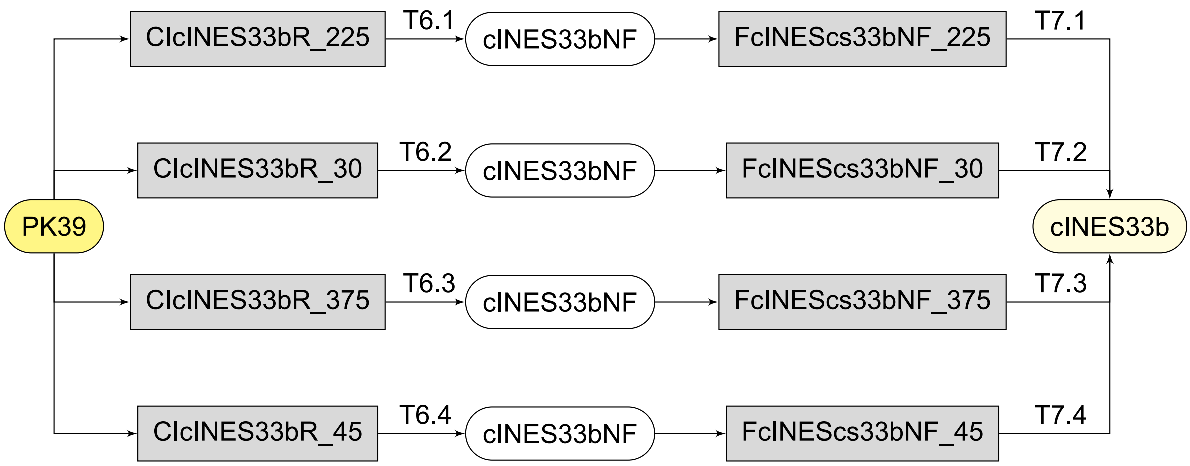

Figure 5.7: The STN representing Category 3 with batch-size dependent processing times [Own visualisation].

All tasks that have been mentioned in the STN's of Figure 5.1 up to and including Figure 5.5 that need to be split in sub-tasks for the sake of the batch-size dependent processing times will have an addition to their name that corresponds to the batch-size that the particular task can process. If a task can process a batch-size of 22,5 T, the addition will be '_225' etc.. In Appendix B, Table B.2 gives the full list of tasks that can be executed according to the mathematical model. This table also includes the corresponding processing times in time steps of all these individual tasks. On top of that, the numbers given in Table X in the column 'Suitable for task' also correspond to the tasks listed in Table B.2. Each unit is able to execute the tasks assigned to it and only one unit is used to perform the task. That is, in reality for example both the bleaching tank and the filtering department are occupied when the filtering

task is executed. This has not been taken into account in this mathematical model.
According to the explanation on batch-size dependent processing times in Section 4.4, the intermediate state in Figure 5.7 should also have been split into sub-states. However, due to the fact that sub-lines [...] are modelled as one piece of equipment, both tasks producing and consuming this state have the same capacity and that this state is subjected to a ZW storage policy, this is not needed.

5.1.7. Additional notes for the mathematical model

The mathematical model discussed so far assumes an immediate material transfer. This does not match the classification given in Section 3.3. However, implementing the pipelines would only make sense if there are restrictions in the pipelines by for example crossing pipelines or pipelines used between multiple equipment units. For this scope, there are no such restrictions. Difficulties regarding piping only occurs between the tank parks and the refinery and the refinery and the ROS. The next best option would be to consider the Time-consuming, No-resources classification. However, data regarding this subject was not available at SDOZR.

For the scheduling horizon considered in this mathematical model it has been assumed that no maintenance needs to be performed and that there will be no non-working periods. On top of that, any resource constraints that are present at SDOZR are not included in this mathematical model. It would have made the model more complex then necessary at this moment.

The maximum shelf-life sl_s for all states that can be stored in this mathematical model has been set to three days, corresponding to 144 time steps.
It is assumed that there is no initial storage, so $S_{st_0} = 0$. In order to create a situation like that, it is also required that no tasks can be performed during the very first time step, so $W_{ijt_0} = 0$. The fictional unit costs C_{st} for the intermediate states are given a value of 0 units per tonne. The end products, so bPOcssf40, cINEScs82, cINES33b, bINEScs97 and bINEScs43, are given a value of 15 units per tonne. It is not desirable to have one of the states remaining in storage at the end of the scheduling horizon. If that situation occurs, the mathematical model will produce more than required according the customer demand. Therefore, the unit cost of the states that can be stored, so bPOcssf40, cINEScs82, cINES33b, bINEScs97, bINEScs43 and also cINEScs43b, will have a value of -15 units per tonne at the final time step. The storage cost C_{st}^S have a value of 5 units per tonne for the intermediate states, whereas the storage costs for the end products equal 15 units per tonne.

Finally, for solving the mathematical scheduling optimization model, both hardware and software tools are used. The hardware tool concerns a HP EliteBook 8570w Workstation. This laptop has an Intel® Core™ i7-3630QM CPU @2.40 GHz processor and 8 GB RAM. The software to be used are Python version 3.7 together with solver Gurobi, version 9.0.0. The choice for these software packages is a commercial one in relation with its capabilities. Python is an open-source software package, meaning that it is available for free. Therefore, it is easily accessible by all different kinds of users that gather in different communities to share their knowledge. Unfortunately, Gurobi is not open-source. However, it is a very powerful mathematical optimization solver for which a lot of documentation is available, accompanied with a great variety of examples.

5.2. Results of the basic model

In Table 5.4, the results of the basic model for both $H = 240$ and $H = 480$ are presented. The basic model has been solved to optimality. In the table, the objective value and the solution time are given. Besides, references have been given to the three decision variables that could be of use when this basic model would be applied in the real world. It concerns the decision variables of the amount raw material delivery, R_{st}, the amount of storage under the UIS policy, S_{st}, and the amount of material which starts undergoing a task in a certain unit, B_{ijt}. The first two decision variables have been visualised by means of a line graph. For each state of material that can be delivered from an external resource or that can be stored respectively, a subplot has been made. On the x-axis, the time periods are plotted, whereas on the y-axis, the corresponding amount of material in tonnes is plotted. The results of B_{ijt} have been visualised by a Ganttchart. Here, the time periods are plotted on the x-axis as well, whereas on the y-axis, the processing units corresponding to Figure X are plotted. Each block in the Ganttchart has an upper half and a lower half. In the lower half, the product family is shown. It makes it easy to see of the processing order-constraint given by Equation 4.6 is not violated. In the upper half of the block, the task number is shown. The number corresponds to the task name described in the table listed next to the Ganttchart and matches the listing in Table B.2. This way of visualising this decision variable makes it possible to see what task is performed during what time periods and by what unit. The description given in this paragraph about the graphs also holds for all the other graphs in Appendix D corresponding to other scenario's that will be elaborated further in Section 5.4.

It has been discussed in Section 5.1.4 that both $H = 240$ and $H = 480$ will be calculated for the basic model. A doubling in the number of time steps resulted in a doubling of the number of constraints and variables. However, the solution time required by the current tools to solve the model for optimality for $H = 480$ is more then 100 times larger compared to $H = 240$. On top of that, the resulting B_{ijt} of the $H = 480$ - scenario showed batch-sizes to be processed in the order of 10^{-12} to 10^{-17}. Also W_{ijt}, which is a binary decision variable, showed numbers in the order of 10^{-13} to 10^{-19}. From all the scenario's that have been calculated for this research, which where 36 in total, this was the only scenario where it happened. No explanation is found why this happens. All other values in the results of these decision variables are numbers that could be expected by the input of the mathematical model. Therefore, these very small numbers are assumed to be zero.

Table 5.4: Results of the model with Objective 4.1

	H	Objective value	Solution time [s]	R_{st}	S_{st}	B_{ijt}
Objective 4.1	240	-9 337,5	197,67	Figure D.1	Figure D.2	Figure X
Objective 4.1	480	-29 062,5	20 238,04	Figure D.3	Figure D.4	Figure X

5.3. Validation of the mathematical model

The adapted method of batch-size dependent processing times for a single value of a batch-size as discussed in Section 4.4.4 has been implemented in the mathematical model as well. Analysing the resulting values of B_{ijt} for both $H = 240$ and $H = 480$ shows that no batches with batch-sizes other than the defined possible batch-sizes are processed by any of the tasks. However, during the calculation of the basic model an infeasible solution appeared when a smaller batch-size than 22,5 T was demanded. The expectation was that when a batch-size smaller than 22,5 T was implemented, the mathematical model would produce the minimum batch-size and keep the surplus in storage. On the other hand, the mathematical model is able to handle a multiple of 22,5 T, for example 67,5 T, without resulting in an infeasible solution.

For the modelling of BL [...] and the C. I., the processes taking place on these production lines in the real world had been split. The splitting has been done in such a way that the bleaching tank is only used for bleaching, the chemical interesterification cauldron is only used for the C. I., the filter is only used for filtering and the buffer is only used for buffering. However, as it has been discussed in the process description in Section 2.2.5, this is not how the process occurs in the real world. In the real world, both the bleaching tank and the filter are occupied during the filtering task and while an oil is filtered, the buffer is also involved to receive the filtered oil. For the total time required for producing the oil, it does not make a difference, since the total time required is divided over the different processes. However, when simultaneity is considered, there might be some friction in the execution of the created schedule. The way of modelling used here does account for some sort of simultaneity, because it is now able to start a new bleaching process once filtering starts. In reality, only a new process in the bleaching tank can be started when the filtration is in the RVF stage.
Sub-lines [...] of BL [...] are now modelled as a single unit and a single task. However, in case the mathematical model might be expanded in the future with the deodorizer, it might be desirable to model the buffers of the two sub-lines as separate units with separate tasks. In that way, restrictions on subtraction from the buffers can be easily implemented and a voluntary FIS policy with shared storage units is also enabled. In the case of simultaneity it might even be desirable to model the two sub-lines in the same way as sub-line [...]. For the modelling of the simultaneity a more strict data collection is needed on the sub-processes on the sub-lines. A data collection that distinguishes the bleaching process, the blackrun, the RVF and the material transfer to the buffers and what equipment is occupied

by these processes.

This mathematical model has the option to give a minimum and maximum capacity to a unit j, V_{ij}^{min} and V_{ij}^{max}. However, this capacity is linked to task i. In the case of the filters, the capacity is actually determined by the amount of state that has to be removed: the bleaching earth. In this mathematical model, the bleaching earth has not been considered. However, it is a very important state to consider, since it determines the time required for filtering. When a batch contains more bleaching earth than the maximum capacity (based on the state) of the filter, the filtering process is extended with a fixed amount of time that is needed to clean the filter. A possibility to model this is to include the bleaching earth as an input state. When it appears that more bleaching earth is added to the mixture to be processed than the filter can filter out at once, automatically a fixed amount of time is added to the filtering time. This has to be determined by the mathematical model.

The demand for the basic model has been set up based on five product categories. The demand formed by other products than the ones representing the five product categories were classified into one of the categories and were implemented into the demand as the representing product. However, the complexity in production at SDOZR is so high, that this way of creating a realistic demand is actually not possible. For example, different products require different amounts of bleaching earth. The more bleaching earth is needed, the longer the filtering process will take and vice versa. Besides, as it has been discussed in section 5.1.5, only approximately 50% of the real world demand was implemented in this mathematical model.

The newly developed method in Section 4.4.6 to give a restriction on maximum shelf-life has been implemented in the mathematical model. The results of both $H = 240$ and $H = 480$ are inconclusive about how this constraint influences the outcome of the mathematical model. The only thing that can be concluded is that the number of consecutive storage actions are not exceeding the maximum shelf-life $sl_s = 144$.
Analysing the Ganttcharts of B_{ijt} for both $H = 240$ and $H = 480$ in the context of the production order shows that tasks processing products of Family 2 do occur behind a task processing a product of Family 0 with no other task in between. However, there is sufficient time in between the tasks in order to clean the equipment. This observation does not make it possible to conclude that the constraint on the production order is not violated. It might be that the demand implemented in the basic model is unintentionally set up in such a way that the basic model is not forced to change its production order to avoid an order where a task processing a product of Family 2 would need to be executed within $\tau_{jkl} = 5$ of a task processing a product of Family 0.
Finally, the solution time required to solve the basic model for $H = 240$ is with approximately three minutes considered to be reasonable. However, the solution time required to solve the basic model for $H = 480$ is with approximately 5,5 hours considered to be long.

5.4. Sensitivity analysis

In Section 5.3 various elements of the basic model and its results have been discussed in the context of the validation of the mathematical model. The aspects discussed in the validation need additional research to clarify the statements made. Some of these statements can be investigated by means of a sensitivity analysis. A sensitivity analysis will therefore be performed on the scheduling horizon, on the objective, the production order and the batch-sizes. The sensitivity analysis on the scheduling horizon will be performed throughout the whole experimental study. This is already shown by the results of the calculation of the basic model in Table 5.4. As stated in Section 5.1.4, a scheduling horizon of 5 days or 240 time steps has been used to build the mathematical model. However, [...] a scheduling horizon of 10 days, resulting in 480 time steps. This doubling in time steps results in a doubling of the number of variables and constraints. Since $H = 480$ requires more then 100 times as much time as $H = 240$ in the case of the basic model, for the sake of time only a few situations have been selected where the full scheduling horizon has been applied. An explanation is provided by the concerning situation.
The sensitivity analysis on the objective will be further elaborated in Section 5.4.1. Section 5.4.2 and 5.4.3 discuss a sensitivity analysis on the production order and batch-sizes respectively. The sensitivity analysis on both production order and batch-size are also combined with the sensitivity analysis concerning the objectives. The combined sensitivity analysis are discussed in the corresponding sections

of both Section 5.4.2 and Section 5.4.3.
All mathematical models in this section have been solved to optimality, unless stated otherwise.

5.4.1. Objective

So far, the objective given by Equation 4.1 has been used. This objective aims to maximize the profit. As it has been observed during the validation discussed in Section 5.3, the time required to solve the mathematical model - and especially the scenario where the scheduling horizon equals $H = 480$ - is considered to be long. In an attempt to improve the solution time required for this scenario, an alternative objective has been formulated. The current objective can also be interpreted as an objective that, because of costs, reduces the number of storage actions. The alternative objective is given by Equation 5.1. This new formulation might also be of an advantage for memory issues compared to the objective given in Equation 4.1 when even longer scheduling horizons or large amounts of states are considered, since variables such as C_{st} and C^S_{st} are no longer needed.

$$Min\left(\sum_{s\in SSL}\sum_{t=0}^{H+1} BS_{st}\right) \tag{5.1}$$

As it has been stated in Section 5.3, the results of the basic model for both scheduling horizons are inconclusive about how the newly developed constraint on the maximum shelf-life influences the outcome of the mathematical model. Pushing the mathematical model to its limits should provide a definite answer to this. Therefore, the demand, D_{st} will be ignored for this scenario and by formulating an alternative objective, the mathematical model is asked to calculate the maximum amount of tonnes end product that could in theory be present at the end of the scheduling horizon. This approach requires an adaption in the material balance given by Equation 5.2, from which the demand D_{st} has been removed. The alternative objective is formulated in Equation 5.3.
To investigate what the influence of the maximum shelf-life is, this objective will be split in two scenario's. One scenario will take into account the maximum shelf-life (SY) and the other scenario will ignore this maximum shelf-life (SN). An investigation of these two scenario's can clarify what the influence of the constraint on maximum shelf-life actually is. For this objective as well, C_{st} and C^S_{st} are no longer needed.

$$S_{st} = S_{s,t-1} + \sum_{i\in\overline{T}_s}\left(\overline{\rho}_{is}\sum_{j\in K_i} B_{ij,t-p_{is}}\right) - \sum_{i\in T_s}\left(\rho_{is}\sum_{j\in K_i} B_{ijt}\right) + R_{st} \qquad \forall s,\ t\neq t_0 \tag{5.2}$$

$$Max\left(\sum_{s\in S_{out}} S_{s,H+1}\right) \tag{5.3}$$

The results of replacing Objective 4.1 with Objective 5.1 and 5.3 respectively in the basic model are shown in Table 5.5. The table has the same set up as Table 5.4. For these scenario's in the sensitivity analysis, both $H = 240$ and $H = 480$ have been calculated.

The objective value of the scenario where Objective 5.1 is implemented with $H = 240$, equals 131. This means that during the whole scheduling horizon, there are 131 storage actions. In other words, at 131 time steps during the scheduling horizon material of a certain state has been stored. Analysis on the number of storage actions in the basic model with Objective 4.1 with $H = 240$ shows that there were 148 storage actions. It shows that an alternative formulation does not necessarily give the same result. This is also seen when the graphs of the decision variables R_{st}, S_{st} and B_{ijt} are compared. Especially comparing the storage graphs of both objectives shows how both objectives approach their optimality. It can be seen that the storage graph for the solution of Objective 5.1 shows higher and smaller spikes in the graph compared to the storage graph for the solution of Objective 4.1. The way of formulating Objective 5.1 makes the mathematical model decide to produce more in one time and deliver to customers from that source. The amounts stored in this scenario are not taken into account and do not give a burden to the solution as it does in the basic model.

Table 5.5: Results of the models with Objective 5.1 or Objective 5.3

	H	Objective value	Solution time [s]	R_{st}	S_{st}	B_{ijt}
Objective 5.1	240	131	265,83	Figure D.5	Figure D.6	Figure X
Objective 5.1	480	-	> 900 000,00	-	-	-
Objective 5.3, SY	240	1 867,5	1 988,82	Figure D.7	Figure D.8	Figure X
Objective 5.3, SY	480	1 867,5	9 191,13	Figure D.9	Figure D.10	Figure X
Objective 5.3, SN	240	2 760,0	2 302,03	Figure D.11	Figure D.12	Figure X
Objective 5.3, SN	480	5 610,0	15 291,14	Figure D.13	Figure D.14	Figure X

The scenario with Objective 5.1 was not solved to optimality. After a calculation time of 904 488 seconds, the run has been interrupted. At that moment, the objective value was 245,10880 with an optimality gap of 25.2%.

The objective value of the scenario where Objective 5.3 is implemented with $H = 240$ is equal to the objective value of the scenario where Objective 5.3 is implemented with $H = 480$. In both scenario's, the restriction on maximum shelf-life has been implemented. This explains why the objective value's of these scenario's are the same. The maximum amount of end product to be produced in these scenario's was scheduled in such a way that the shelf-life limitation was not violated. Since the maximum shelf-life is the same in both situations, both scenario's more or less deal with the same scheduling horizon. It results in a number of tasks to be performed that are all scheduled at the very end of the scheduling horizon. This can be clearly seen when the the Ganttcharts of decision variable B_{ijt} are compared. The first task in the $H = 240$ scenario starts at the 36th time step. The first task in the $H = 480$ scenario starts at the 270th time step. In total, the first scenario executes processes during 204 time steps, whereas the second scenario executes processes during 210 time steps. The first task of the first scenario is a C. I. task of 37,5 T, whereas the first task of the second scenario is a C. I. task of 45 T. This does however not result in a different objective value. Apart from bINEScs97, all four end products have been produced in both scenario's.
A further investigation on the number of storage actions in the context of the maximum shelf-life shows that the maximum shelf-life constraint does influence the results of the mathematical model. However, when going more into detail it appears that the number of consecutive storage actions equals 145 for both $H = 240$ and $H = 480$. The cause might be a mistake when the constraint was implemented in the tools for solving the mathematical model, but it might also be that the formulation of this constraint needs to be improved.

The objective value of the scenario where Objective 5.3 is implemented, where the maximum shelf-life restriction is not taken into account and where $H = 480$ is more then doubled compared to the scenario where $H = 240$. The solution time required with the current tools for $H = 480$ is almost 7 times bigger compared to $H = 240$. Both Ganttcharts for B_{ijt} are packed with tasks to be performed during the scheduling horizon. Compared to the scenario's where the restriction on shelf-life was implemented, only three instead of four of the five possible end products have been produced, namely bPOcssf40, cINEScs82 and cINES33b. All three are tasks that need processing on BL [...] or C. I. only, not both. Overall, these processes need the least amount of time to process. Choosing the processes that require the least amount of time results in a higher production of end products.

5.4.2. Production order

The implementation of a desired production order has been discussed in both Section 4.4.5 and Section 5.1.3. So far, a particular production order where products of Family 2 are not allowed to be processed within 5 time periods of a product of Family 0 has been considered. In these five time periods in between

the processes, the equipment should be cleaned to prevent contamination. Considering the resulting Ganttchart of B_{ijt} of the basic model - in Figure X for $H = 240$ and in Figure X for $H = 480$ - shows that a product of Family 2 is processed after a product of Family 0 with sufficient time in between. However, it is possible that cleaning the equipment does not prevent the products from contaminating. On top of that, there is the doubt described in Section 5.3, stating that the demand implemented in the basic model is unintentionally set up in such a way that the basic model is not forced to change its production order to avoid an order where a task processing a product of Family 2 is executed within $\tau_{jkl} = 5$ of a task processing a product of Family 0. Therefore, two scenario's have been investigated in this context. For both scenario's, only $H = 240$ has been considered. The two scenario's are applied to the mathematical model with Objective 4.1, the mathematical model with Objective 5.1 and the mathematical model with Objective 5.3 - with and without the restriction on maximum shelf-life.
In scenario 1, it is strongly discouraged to process a product of Family 2 after a product of Family 0, even with $\tau_{jkl} = 5$ time steps in between. This is modelled by increasing $\tau_{jkl} = 5$ to half of the scheduling horizon, so $\tau_{jkl} = 120$. An overview is given in Table 5.6. In scenario 2, this processing order is forbidden by increasing the cleaning time to the whole scheduling horizon, so $\tau_{jkl} = 240$. An overview is given in Table 5.7.

Table 5.6: Cleaning time required when a task of Family l (column) is performed after one of Family k (row) in unit j (τ_{jkl}) for the sensitivity analysis in scenario 1.

	Lauric (0)	Non-lauric, no C12:0 (1)	Non-lauric, C12:0 (2)
Lauric (0)	0	0	120
Non-lauric, no C12:0 (1)	0	0	0
Non-lauric, C12:0 (2)	0	0	0

Table 5.7: Cleaning time required when a task of Family l (column) is performed after one of Family k (row) in unit j (τ_{jkl}) for the sensitivity analysis in scenario 2.

	Lauric (0)	Non-lauric, no C12:0 (1)	Non-lauric, C12:0 (2)
Lauric (0)	0	0	240
Non-lauric, no C12:0 (1)	0	0	0
Non-lauric, C12:0 (2)	0	0	0

The results of this sensitivity analysis for all three objectives are given in Table 5.8. As can be seen, the second scenario causes an infeasible solution for both Objective 4.1 and Objective 5.1. This means that the current demand that has been implemented requires a production order where a product of Family 2 is processed after a product of Family 0 in order to fulfill the demand. Because demand is not taken into account in the mathematical model with Objective 5.3, no such order is required since the mathematical model will just avoid this. Therefore, the second scenario gives no infeasible solution there. The other two infeasible solutions show that the application of the constraint on sequence dependent cleaning discussed by Kondili et al. (1993) has been successfully implemented in the mathematical model in order to prevent an undesired production order from occuring.
Comparing the scenario where $\tau_{jkl} = 120$ and Objective 4.1 is implemented with the basic model with the same objective shows that the solution time for this scenario is more then four times larger then the basic model, whereas the objective value is almost 20 times larger. The Ganttchart of B_{ijt} shows a shift of the tasks to earlier in the scheduling horizon compared to the Ganttchart of B_{ijt} of the basic model. By doing this, the mathematical model reserves enough time between for example a buffering task of Family 0 and a buffering task of Family 2 that both need to be processed on buffer [...]. The consequence is that the products produced by the tasks that are expedited need to be stored longer. This can also be clearly seen when the storage graphs are compared. In that light, it is remarkable that Objective 5.1 requires less storage actions compared to the objective sensitivity analysis. However, here the same phenomenon applies as discussed in Section 5.4.1. The storage levels for the first scenario with Objective 5.1 are higher compared to the first scenario with Objective 4.1, but less storage actions are required to fulfill the customer demand.
Comparing the results of applying both scenario's to the mathematical model where Objective 5.3 is implemented and the restriction on shelf-life is taken into account shows that both objective values

Table 5.8: Results of the sensitivity analysis of the model with Objective 4.1, Objective 5.1 or Objective 5.3 concerning production order.

	H	Objective value	Solution time [s]	R_{st}	S_{st}	B_{ijt}
Objective 4.1, $\tau_{jkl} = 120$	240	-178 875,0	711,22	Figure D.15	Figure D.16	Figure X
Objective 4.1, $\tau_{jkl} = 240$	240	Infeasible				
Objective 5.1, $\tau_{jkl} = 120$	240	71	1 525,84	Figure D.17	Figure D.18	Figure X
Objective 5.1, $\tau_{jkl} = 240$	240	Infeasible				
Objective 5.3, $\tau_{jkl} = 120$, SY	240	1 260,0	4 445,11	Figure D.19	Figure D.20	Figure X
Objective 5.3, $\tau_{jkl} = 240$, SY	240	1 065,0	601,85	Figure D.21	Figure D.22	Figure X
Objective 5.3, $\tau_{jkl} = 120$, SN	240	2 152,5	1 065,02	Figure D.23	Figure D.24	Figure X
Objective 5.3, $\tau_{jkl} = 240$, SN	240	1 650,0	610,23	Figure D.25	Figure D.26	Figure X

are lower than the scenario in the objective sensitivity analysis. The solution time for the first scenario is more than two times higher in the first scenario, but more than three times smaller for the second scenario. In the scenario where $\tau_{jkl} = 5$, no task of Family 2 is performed. When $\tau_{jkl} = 120$, still no task of Family 2 is performed, but the Ganttchart of B_{ijt} looks severely different compared to the $\tau_{jkl} = 5$ scenario. The Ganttchart of the first is less packed than the one of the second. The Ganttchart of the second scenario - where $\tau_{jkl} = 240$ - is even less packed. However, here no tasks of Family 0 have been included in the schedule, whereas tasks of Family 2 are.
When comparing the storage graphs of these three options for τ_{jkl}, it can be seen that when $\tau_{jkl} = 5$ all end products but bINEScs97 are created. When $\tau_{jkl} = 120$ this is still the case, although storage levels are not as high. However, when $\tau_{jkl} = 240$ cINEScs82 and cINES33b are no longer produced, whereas bINEScs97 is.

When the restriction on shelf-life is not taken into account, the both scenario's give a lower objective value compared to the scenario where $\tau_{jkl} = 5$. The first scenario takes only half the time to be solved, whereas the second scenario only needs almost a quarter of the time to be solved. Again, when $\tau_{jkl} = 5$, the Ganttchart of B_{ijt} is very packed, whereas the Ganttcharts of B_{ijt} for $\tau_{jkl} = 120$ and $\tau_{jkl} = 240$ are not. In all three scenario's, no tasks of Family 2 are processed.
When $\tau_{jkl} = 5$, end products bPOcssf40, cINEScs82 and cINES33b are produced and they had ever increasing storage levels. However, when $\tau_{jkl} = 120$, the storage levels of the latter two remained the same after some time. From that point on, relatively small amounts of bINEScs43 have been produced. When $\tau_{jkl} = 240$, cINES33b is no longer produced and the amount of bINEScs43 has increased. The storage level remained the same earlier during the scheduling horizon and was lower then when $\tau_{jkl} = 120$.

5.4.3. Batch-size

This experimental study has been developed to investigate the observation described by Section 5.3. It has been discussed in Section 5.1.5 how the demand for the basic mathematical model has been set up. The demand has been set up based on the output data of BL [...] and C. I.. The batch-sizes in this output data were a multiple of 7,5 T starting from [...]. However, the actual customer demand is formed at the loading berths at the ROS, where also smaller batch-sizes than [...] can be demanded by a customer and that are not equal to one of the four batch-sizes. Implementing a batch-size that was

not equal to one of these four batch-sizes resulted in an infeasible solution. However, it was expected that the mathematical model would select a task to produce enough product according to the possible batch-sizes and that the surplus would have been kept in storage. Therefore, this sensitivity analysis has been set up to investigate why this deviating batch-size caused an infeasible solution. For the experiment two hypotheses has been formulated. The first hypothesis considered the maximum shelf-life. One of the possible explanations for the result of the mathematical model was that the limitation on shelf-life might have caused a situation where the surplus would have been in storage for too long. In this light, it might be possible that placing an odd batch-size demand within the shelf-life seen from the end of the scheduling horizon - so when considering $H = 240$, between the 96th and the 240th time step - might give a result from the mathematical model. To compare, a situation is also tested where an odd batch-size demand was placed before 96th time step.
The second hypothesis considered the actual batch-size itself. Since multiples of 7,5 T starting from [...] do not cause an infeasible solution to appear, the question is if it is possible to fulfill a demand that is not a multiple of 7,5 T or that the multiple of 7,5 T is a unintentional hidden restriction in this mathematical model. Therefore, five extra batch-sizes have been introduced in addition to the four mentioned earlier in this sections: 7,5 T, 10 T, 15 T, 20 T and 25 T. 7,5 T and 15 T are a multiple of 7,5 T, whereas 10 T, 20 T and 25 T are not. These five batch-sizes have been combined with the before and after the maximum shelf-life mark in the scheduling horizon already discussed. The batch-sizes have been randomly added either before or after the 96th time step to the existing list of demands that already had been set up for the basic model. A random time step, but the same time step and example product for all the batch-sizes. Table 5.9 gives an overview of the scenario's to be tested for this sensitivity analysis. The scenario's have been applied to the mathematical model with Objective 4.1 and the mathematical model with Objective 5.1. The mathematical model with Objective 5.3 has not been taken into account, since the demand is not included in this mathematical model. All scenario's are only tested for a scheduling horizon of 240 time steps.

Table 5.9: Scenario's for the sensitivity analysis concerning the batch-size

Demand before the 96th time step (< 96)	Batch-size: 7,5 T
	Batch-size: 10 T
	Batch-size: 15 T
	Batch-size: 20 T
	Batch-size: 25 T
Demand after the 96th time step (> 96)	Batch-size: 7,5 T
	Batch-size: 10 T
	Batch-size: 15 T
	Batch-size: 20 T
	Batch-size: 25 T

The results of this sensitivity analysis on the batch-size is given in Table 5.10. As can be seen, all scenario's where the batch-size equals 10 T, 20 T or 25 T result in an infeasible solution. The scenario's where the batch-size was either 7,5 T or 15 T gave a solution for the scenario where this batch-size was before the 96th time step as well as when this batch-size was after the 96th time step. These observations show that the first hypothesis - the maximum shelf-life restriction is of influence - appears to be false. However, the second hypothesis - the demand should be a multiple of 7,5 T - appears to be true.
The addition of the extra batch with varying batch-size over the different scenario's caused a shift in the tasks set up in the Ganttchart of the B_{ijt} decision variable. Also the batch-sizes processed by the mathematical model changed to be able to cope with the extra batch and with the changing batch-size. This holds for both Objective 4.1 and Objective 5.1. In all scenario's, this causes a shift in amount and number of time steps a state is stored compared to the scenario without the extra batch.

Table 5.10: Results of the sensitivity analysis of the model with Objective 4.1 or Objective 5.1 concerning batch-size

	H	Objective value	Solution time [s]	R_{st}	S_{st}	B_{ijt}
Objective 4.1 <96, 7,5 T	240	-225,0	238,72	Figure D.27	Figure D.28	Figure X
Objective 4.1 <96, 10 T	240	Infeasible				
Objective 4.1 <96, 15 T	240	1 312,5	399,54	Figure D.29	Figure D.30	Figure X
Objective 4.1 <96, 20 T	240	Infeasible				
Objective 4.1 <96, 25 T	240	Infeasible				
Objective 4.1 >96, 7,5 T	240	-4 725,0	239,38	Figure D.31	Figure D.32	Figure X
Objective 4.1 >96, 10 T	240	Infeasible				
Objective 4.1 >96, 15 T	240	-7 687,5	397,35	Figure D.33	Figure D.34	Figure X
Objective 4.1 >96, 20 T	240	Infeasible				
Objective 4.1 >96, 25 T	240	Infeasible				
Objective 5.1 <96, 7,5 T	240	71	113,75	Figure D.35	Figure D.36	Figure X
Objective 5.1 <96, 10 T	240	Infeasible				
Objective 5.1 <96, 15 T	240	60	989,94	Figure D.37	Figure D.38	Figure X
Objective 5.1 <96, 20 T	240	Infeasible				
Objective 5.1 <96, 25 T	240	Infeasible				
Objective 5.1 >96, 7,5 T	240	91	452,54	Figure D.39	Figure D.40	Figure X
Objective 5.1 >96, 10 T	240	Infeasible				
Objective 5.1 >96, 15 T	240	91	523,01	Figure D.41	Figure D.42	Figure X
Objective 5.1 >96, 20 T	240	Infeasible				
Objective 5.1 >96, 25 T	240	Infeasible				

5.5. Conclusion

The building of the scheduling optimization model for Sime Darby Oils Zwijndrecht Refinery (SDOZR) is based on the State-Task-Networks that are created for each of the five example end products to be produced. Each end product corresponds to a production category from all possible products to be produced by the refinery. The categories include 'Only bleaching on Bleaching Line [...]', 'Chemical

Interesterification without filtering and bleaching on Bleaching Line [...]', 'Chemical Interesterification with filtering and bleaching on Bleaching Line [...]', 'Chemical Interesterification without filtering and bleaching on Bleaching Line [...]', 'Chemical Interesterification with filtering and bleaching on Bleaching Line [...]'. However, since BL [...] is not in the scope of this report, it is assumed that the input states for BL [...] are equal to the output state of the C. I. and the end of the production processes for this scope. Based on the five State-Task-Networks, a list of states and a list of tasks can be formed. The tasks can be assigned to the pieces of equipment that will be modelled. For the implementation of the pieces of equipment, simplifications have been applied to the processes and some pieces of the real world equipment have been merged into one piece of equipment to be modelled. This resulted in a total of eight units to be implemented in the mathematical model. For the states it should be determined what storage policy needs to be applied to what state and what maximum storage capacity C_s belongs to that. Next, data has been collected to assign the states to one of the three product families, define a scheduling horizon, determine the demand, determine the processing times, determine the unit cost and the cost of storage and determine the conditions for the initial state of the mathematical model. Since the processing times are batch-size dependent, a separate sub-task should be assigned to each batch-size. The list of batch-sizes includes [...]. Concerning the demand, eventually about 50% of the real world demand has been implemented in the mathematical model. Then, the maximum shelf-life and the cleaning time required when a task processing a state of Family 2 is executed after a task processing a state of Family 0 on unit j should be determined. Finally, the hardware and software tools to be used to solve the mathematical model should be defined.

A validation of the basic model showed that the mathematical model was not able to process a batch-size smaller than [...]. On the other hand, a multiple of [...], like for example [...], did not cause any problems during solving. Research by means of a sensitivity analysis showed that the mathematical model is not able to process batch-sizes that are not a multiple of 7,5 T.
Constraints have been implemented on the maximum shelf-life and production order. The results of the basic model were inconclusive on the influence of these constraints. The maximum shelf-life was not exceeded, but did not reach the limit either and also the production order was not violated. However, in both cases it could be that the input of the model did not push it to its limits. Research by means of a sensitivity analysis to both constraints showed that the maximum shelf-life constraint does put a limit on the number of consecutive storage actions. However, the shelf-life is violated by one storage action. The production order constraint has not been violated.
As it has been announced on beforehand, simplifications have been applied to the scheduling optimization problem of SDOZR for the implementation of the units and the corresponding tasks. This way of implementing ignores the simultaneity occurring during production, which might cause friction in the execution of the created schedule. On top of that, the actual filter capacity - which depends on only a fraction of the batch to be processed - is ignored by the mathematical model. The construction of the product demand for the basic model is based on a great number of generalizations that are of influence on among others the processing times. Besides, despite this generalizations, only about 50% of the real world demand was implemented in this mathematical model. Finally, the solution time required to solve the basic model for $H = 240$ is with approximately three minutes considered to be reasonable. However, the solution time required to solve the basic model for $H = 480$ is with approximately 5,5 hours considered to be long.

6

Conclusions, discussion and recommendations

6.1. Conclusions and discussion

The main research question for this research was formulated as: "Is there a way to optimize the current way of scheduling in order to improve the performance part of the OEE formula?" The conclusion is that there is. Even though the mathematical model developed during this research is not (yet) able to replace the scheduling software currently used by Sime Darby Oils Zwijndrecht Refinery (SDOZR), it definitely shows potential. The mathematical model can be seen as a basis for further development to make it suitable to replace the scheduling software currently used. Not only the mathematical model itself, but also the approach that has been used to come to this point. The approach is developed by answering sub-questions that concern the restrictions provided by the current situation at SDOZR, the state-of-the-art in production scheduling solutions, the appropriate scheduling optimization model, the implementation of the scheduling problem of SDOZR in this scheduling optimization model and the functioning of this scheduling optimization model compared to the current situation.

An analysis of the current situation at SDOZR considering the production processes, the equipment and the current way of scheduling showed that the current way of scheduling is a complicated process where most of the handlings are performed by hand. Most of the decisions made for creating a production schedule are based on the knowledge and experience of the schedulers that is not implemented in the scheduling software currently used. The scheduling optimization model to be developed has to deal with equipment connectivity, minimum and maximum batch-size and/or equipment capacity, piping, deterioration or maximum shelf-life and minimum storage or quarantine time. The equipment capacity of the filters is determined by the amount of residue it is able to filter out of the emulsion. Also production order has to be considered because of the risk on contamination between different groups of products.

Based on the state-of-the-art in production scheduling solutions and the restrictions provided by the current situation at SDOZR, an appropriate mathematical model can be selected. From the literature study, fourteen papers have been selected that were considered to be the most promising in describing a basic mathematical model. By means of a problem classification, developed by Méndez et al. (2006a), applied to these fourteen papers and the scheduling optimization problem of SDOZR a comparison could be made between the papers and the requirements of the scheduling optimization problem of SDOZR. Based on that comparison, the mathematical model discussed by Kondili et al. (1993) has been chosen as the basic mathematical model. This mathematical model has been elaborated further by implementing the restrictions provided by the current situation at SDOZR. Novelties to the mathematical model involve a mandatory Finite Intermediate Storage (FIS) policy for shared storage units, batch-size dependent processing times that depend on a single batch-size value instead of a range of values, the application of sequence dependent cleaning to force a certain production order and the newly developed maximum shelf-life.

After the implementation of the SDOZR scheduling optimization problem in the mathematical model and the calculation of this mathematical model, a model validation showed that the mathematical model was not able to process a batch-size smaller than [...]. On the other hand, a multiple of [...], like for example [...], did not cause any problems during solving. Research by means of a sensitivity analysis showed that the mathematical model is not able to process batch-sizes that are not a multiple of 7,5 T. The results of the mathematical model were inconclusive on the influence of the constraints on maximum shelf-life and the production order. The maximum shelf-life was not exceeded, but did not reach the limit either and also the production order was not violated. However, in both cases it could be that the input of the model did not push it to its limits. Research by means of a sensitivity analysis to both constraints showed that the maximum shelf-life constraint does put a limit on the number of consecutive storage actions, but that the shelf-life is violated by one storage action. The production order constraint has not been violated. A simplification on the implementation of the units and the corresponding tasks makes the mathematical model ignore the simultaneity that occurs during production, which might cause friction in the execution of the created schedule. On top of that, the actual filter capacity - which depends on only a fraction of the batch to be processed - is ignored by the mathematical model. The construction of the product demand for the basic model is based on a great number of generalizations that are of influence on among others the processing times. Besides, despite this generalizations, only about 50% of the real world demand was implemented in this mathematical model. Finally, the solution time required to solve the mathematical model for a scheduling horizon of 240 time steps is with approximately three minutes considered to be reasonable. However, the solution time required to solve the basic model for a scheduling horizon of 480 time steps is with approximately 5,5 hours considered to be long.

6.2. Contributions of this research

This research contains different elements that contribute to either the company or the literature. The contributions to literature arise from the development of the mathematical model for the scheduling problem of SDOZR. For this particular situation it was required to implement constraints from the theoretical model from literature slightly different in order to model the situation as close to reality as possible. The development of a method to implement a restriction on maximum shelf-life from scratch is considered to be the major contribution. Only very little is discussed about this topic in literature. The method developed monitors the consecutive storage actions due to which a limitation can be introduced. The little information available on this topic in literature discussed a maximum shelf-life for a product that gradually deteriorates over time, whereas the method applied in this mathematical model is developed for products that deteriorate at one instant. This method can therefore be applied to situations where for example an analysis is only valid for a certain amount of time. When this time has past, even with a few time steps, a new analysis should be performed.

Other contributions that arise from the development of the mathematical model for the scheduling problem of SDOZR concern the introduced distinction between a voluntary and mandatory Finite Intermediate Storage (FIS) policy for shared storage units and the introduced distinction between batch-size dependent processing times for a range of batch-sizes and a batch-size of a single value. The theoretical model only considered the FIS policy for shared storage units as a task that could be performed if it was needed. However, for the scheduling problem of SDOZR it was necessary to implement this constraint as a mandatory one. Further development and especially an extension of the mathematical model with the implementation of the deodorizers requires storage in the buffers before the oil can be processed any further. The mandatory storage might also occur when there are equipment restrictions on transporting the product to the next stage. The batch-size dependent processing times were considered by the theoretical model to apply to a range of batch-sizes, where batches with a batch-size between A kg and B kg have a processing time of p_1, whereas batches with a batch-size between B kg and C kg have a processing time of p_2. The scheduling optimization problem of SDOZR is subjected to batch-size dependent processing times, but the batches can only be of a certain value. That is, batches with a batch-size of A kg have a processing time of p_1, batches with a batch-size of B kg have a processing time of p_2, etc.. The application described in the theoretical model has been adapted to the latter situation. Another minor contribution arising from the development of the mathematical model concerns the application of sequence dependent cleaning in order to force, discourage or forbid a certain processing order.

The contributions to the company arise from the analysis of the situation at SDOZR. This includes an analysis of the scheduling process, an analysis on the refinery layout, the equipment that is to be used and the processes involved. The set up of this analysis can be used to further elaborate the current mathematical model. Besides, the descriptions can help future employees of SDOZR to understand the basic principles of the refinery, even though the description might be schematic at some points. For this mathematical model, only a small part of the total plant has been implemented. This can be considered as a start for the development of a mathematical model that is able to schedule the whole refinery or plant at once. This concerns both the understanding of the processes as well as considering what is needed to develop such a complete model. In particular, it became clear during this research that some restrictions could not be implemented in the mathematical model due to a lack of data or unclear registering of the data. An example is the time needed to pump the oil from one tank to another within Bleaching Line [...] and the way of registering and subdividing the production processing times per stage in the sub-line. That is, for example, a subdivision in time when a bleaching tank is in use for bleaching and when it is in use for the blackrun.

6.3. Recommendations for future research

Recommendations for future research apply to both literature and the company. For the first, the recommendations arise from the development of the mathematical model. Even though the development of the mathematical model already introduced some major contributions to literature, further research could improve the mathematical model even further. The validation of the mathematical model and the sensitivity analysis that have been performed thereafter show that this mathematical model is able to process different batch-sizes, as long as they are a multiple of 7,5 T. During this research no explanation has been found that clarifies this behaviour of the mathematical model. It is therefore recommended to investigate this behaviour of the mathematical model, find an explanation to this and improve the mathematical model to overcome this limitation. The sensitivity analysis also showed that the novel method on a limitation on shelf-life does work, but the results showed that the implemented shelf-life was violated by one time step. The cause of this could be that the formulation was not sufficient or that the implementation in the mathematical model needs to be improved. This should be investigated and solved. Research should also be done to a different or additional way of implementing unit capacity. An example for the particular situation of SDOZR is given by the filtering units. The current way of addressing a maximum capacity to a unit depends on the combination of the unit itself and the task that will be performed in that unit. However, the filtering units need a formulation for the maximum capacity that is determined by the state that has to be removed in this case. For the other states of the emulsion, the filter is just a unit that needs to be passed.

In the particular case of the scheduling problem of SDOZR, the mathematical model could be improved to be a more accurate model in different ways. In the beginning of the implementation of the scheduling problem of SDOZR in the mathematical model, simplifications have been made to the processes and equipment, where for example feedback loops have been ignored. Due to this, simultaneity in the different processes have been ignored. In other words, the mathematical model assumes that a single task is performed in a single unit. In reality however, multiple equipment units are required for one task at some points. It is recommended to investigate how this phenomenon could be implemented or if there are other possibilities to take this phenomenon into account. A contribution of the company is required here to investigate its possibilities in registering processing data. It is recommended to the company to expand its data registration on all the sub-processes on a single processing line with an emphasis on the processing times. That is, to register for example the time required for pumping the material from one piece of equipment to the next, but also make a deviation in the time an unit is occupied for what (part of the) task the unit is occupied for. The availability of this data should make it possible to implement the simultaneity or at least take it into account.

Another simplification that has been made during the implementation of the scheduling problem of SDOZR in the mathematical model is the number of states. All the products that can be produced by the refinery of SDOZR have been divided into five categories that address the five different processes that can be performed. However, with this simplification accuracy is lost on each of the product specifications in terms of for example processing times. The time required for for example bleaching and the time required for filtering - which depends on the amount of bleaching earth added - strongly depends on the oil that will be processed. Implementing all the different products instead of implementing them by means of the product representing the processing category would make the model more accurate. The solution time of the mathematical model for a scheduling horizon of 480 time steps using a particular set of tools was about 5,5 hours and is therefore considered to be slow. By means of a sensitivity analysis it was investigated if a different formulation of the objective would decrease the solution time on this same set of tools. The alternative formulation unfortunately resulted in a longer solution time. This is not desirable when the act of scheduling is highly iterative. In the light of the other recommendations to be implemented, due to which the model size will further increase, it is therefore recommended to investigate different options to improve the solution time. This could include the use of different tools, different objective formulations or different formulations of the constraints. Applying heuristics or meta-heuristics also belongs to the possibilities.

Bibliography

Ariaansz, R. F. (2020). *Hydrogenation in Practice*. https://lipidlibrary.aocs.org/edible-oil-processing/hydrogenation-in-practice. Accessed on October 21, 2020.

Baart, P. D. (2020). *Tankpark Scheduling Optimisation*. Unpublished master's thesis, Delft University of Technology, Delft, The Netherlands.

Beal, R. E. and Eisenhauer, R. A. (Beal, R. E. and Eisenhauer, R. A.) *Method for Removing Metals from Vegetable Oils*. US 3634475 A Jan. 1972.

Birewar, D. B. & Grossmann, I. E. (1990). Simultaneous Production Planning and Scheduling in Multiproduct Batch Plants. *Industrial and Engineering Chemistry Research*, 29, 570-580.

Blom, S. (2018). *Wat is OEE?* https://www.blomconsultancy.nl/wat-is-oee/. Accessed on October 30, 2019.

Cafaro, D. C. & Cerdà, J. (2008). Dynamic scheduling of multiproduct pipelines with multiple delivery due dates. *Computers and Chemical Engineering*, 32, 728-753.

Calliauw, G. (2020). *Dry Fractionation*. https://lipidlibrary.aocs.org/edible-oil-processing/dry-fractionation. Accessed on October 21, 2020.

Castro, M. C., Barbosa-Póvoa, A. P., Matos, H. A., & Novais, A. Q. (2004). Simple continuous-Time formulation for Short-Term Scheduling of Batch and continuous Processes. *Industrial and Engineering Chemistry Research*, 43, 105-118.

Castro, P., Barbosa-Póvoa, A. P. F. D., & Matos, H. (2001). An Improved RTN Continuous-Time Formulation for the Short-term Scheduling of Multipurpose Batch Plants. *Industrial and Engineering Chemistry Research*, 40, 2059-2068.

Claassen, F. (2017, October 6). 'Palmolie niet de grote boosdoener'. *Trouw*. https://www.trouw.nl/opinie/palmolie-niet-de-grote-boosdoener~b46de091/. Accessed on October 17, 2019.

Dawande, M., Dickersbach, J. T., Fowler, J. W., Herrmann, J. W., Kalagnanam, J., Kreipl, S., Kuchta, M., Lee, H. S., Lodree Jr, E. J., MacCarthy, B., Martinez, M., Mason, S. J., McKay, K. N., Mönch, L., Newman, A., Norman, B. A., Pfund, M. E., Pinedo, M., & Reddy, C., … Wiers, V. C. S. (2006). A History of Production Scheduling. In J. W. Herrmann (ed.), *Handbook of Production Scheduling* (pp. 1-22). Springer.

de Vré, K. (2011, January 31). De rijzende ster van palmolie. *Trouw*. https://www.trouw.nl/nieuws/de-rijzende-ster-van-palmolie~b054891d/. Accessed on October 16, 2019.

Editors, B. D. (2018). *Crystallization*. https://biologydictionary.net/crystallization/. Accessed on September 21, 2020.

Entrup, M. L., Günther, H. O., van Beek, P., Grunow, M., & Seiler, T. (2005). Mixed-Integer Linear Programming approaches to shelf-life-integrated planning and scheduling in yoghurt production. *International Journal of Production Research*, 43 (23), 2071-5100.

European Palm Oil Alliance (2017). About Palm Oil. https://issuu.com/epoa/docs/2017_05_19_final_standard_presentat. Accessed on October 16, 2019.

Galhardo, F. and Dayton, C. (2020). *Enzymatic Degumming*. https://lipidlibrary.aocs.org/edible-oil-processing/enzymatic-degumming. Accessed on October 22, 2020.

Gibon, V., De Greyt, W., & Kellens, M. J. (2007). Palm oil refining. *European Journal of Lipid Science and Technology*, 109, 315-335.

Göthe-Lundgren, M., Lundgren, J. T., & Persson, J. A. (2002). An optimization model for refinery production scheduling. *International Journal of Production Economics*, 78, 255-270.

Harjunkoski, I., Maravelias, C. T., Bongers, P., Castro, P. M., Engell, S., Grossmann, I. E., Hooker, J., Méndez, C., Sand, G., & Wassick, J. (2014). Scope for industrial applications of production scheduling models and solution methods. *Computers and Chemical Engineering*, 62, 161-193.

Ierapetritou, M. G. & Floudas, C. A. (1998a). Effective Continuous-Time Formulation for short-Term Scheduling. 1. Multipurpose Batch Processes. *Industrial and Engineering Chemistry Research*, 37, 4341-4359.

Ierapetritou, M. G. & Floudas, C. A. (1998b). Effective continuous-Time Formulation for short-Term Scheduling. 2. Continuous and Semicontinuous Processes. *Industrial and Engineering Chemistry Research*, 37, 4360-4374.

Ierapetritou, M. G., Hené, T. S., & Floudas, C. A. (1999). Effective Continuous-Time Formulation for Short-Term Scheduling. 3. Multiple Intermediate Due Dates. *Industrial and Engineering Chemistry Research*, 38, 3446-3461.

Kondili, E., Pantelides, C. C., & Sargent, R. W. H. (1993). A General Algorithm for Short-Term Scheduling of Batch Operations - I. MILP Formulation. *Computers and Chemical Engineering*, 17 (2), 211-227.

Laning, S. J. (1985). Chemical interesterification of palm, palm kernel and coconut oils. *Journal of the American Oil Chemists' Society*, 62, 400-407.

Lee, H., Pinto, J. M., Grossmann, I. E., & Park, S. (1996). Mixed-Integer Linear Programming Model for Refinery Short-Term Scheduling of Crude Oil Unloading with Inventory Management. *Industrial and Engineering Chemistry Research*, 35, 1630-1641.

Lin, X. & Floudas, C. A. (2001). Design, synthesis and scheduling of multipurpose batch plants via an effective continuous-time formulation. *Computers and Chemical Engineering*, 25, 665-674.

Lin, X., Floudas, C. A., Modi, S., & Juhasz, N. M. (2002). Continuous-Time Optimization Approach for Medium-Range Production Scheduling of a Multiproduct Batch Plant. *Industrial and Engineering Chemistry Research*, 41, 3884-3906.

Magatão, L., Arruda, L. V. R., & Neves Jr., F. (2004). A mixed integer programming approach for scheduling commodities in a pipeline. *Computers and Chemical Engineering*, 28, 171-185.

Maravelias, C. T. (2012). General Framework and Modeling Approach Classification for Chemical Production Scheduling. *American Institute of Chemical Engineers AIChE J*, 58, 1812-1828.

Miller, M. (n. d.). *Oxidation of food grade oils*. `https://www.oilsfats.org.nz/documents/Oxidation%20101.pdf`. Accessed on October 22, 2020.

Mockus, L. & Reklaitis, G. V. (1997). Mathematical programming formulation for scheduling of batch operations based on nonuniform time discretization. *Computers and Chemical Engineering*, 21 (10), 1147-1156.

Mockus, L. & Reklaitis, G. V. (1999). Continuous Time Representation Approach to Batch and Continuous Process Scheduling. 1. MINLP Formulation. *Industrial and Engineering Chemistry Research*, 38, 197-203.

Moro, L. F. L. & Pinto, J. M. (2004). Mixed-Integer Programming approach for Short-Term Crude Oil Scheduling. *Industrial and Engineering Chemistry Research*, 43, 85-94.

MVO (n. d.). *Palmolie*. `https://mvo.nl/palmolie`. Accessed on October 16, 2019.

Méndez, C. A., Cerdà, J., Grossmann, I. E., Harjunkoski, I., & Fahl, M. (2006a). State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Computers and Chemical Engineering*, 30, 913-946.

Méndez, C. A., Grossmann, I. E., Harjunkoski, I., & Kaboré, P. (2006b). A simultaneous optimization approach for off-line blending and scheduling of oil-refinery operations. *Computers and chemical Engineering*, 30, 614-634.

O'Brien, J. J. (1969). *Scheduling Handbook*. Cherry Hill, NJ: McGraw-Hill.

Papageorgiou, L. G. & Pantelides, C. C. (1996). Optimal Campaign Planning/Scheduling of Multipurpose Batch/Semicontinuous Plants. 1. Mathematical Formulation. *Industrial and Engineering Chemistry Research*, 35, 488-509.

Pinto, J. M., Joly, M., & Moro, L. F. L. (2000). Planning and scheduling models for refinery operations. *Computers and Chemical Engineering*, 24, 2259-2276.

Reddy, P. C. P., Karimi, I. A., & Srinivasan, R. (2004). A new continuous-time formulation for scheduling crude oil operations. *Chemical Engineering Science*, 59, 1325-1341.

Roundtable on Sustainable Palm Oil (2019). *About*. `https://rspo.org/about`. Accessed on October 16, 2019.

Roundtable on Sustainable Palm Oil (2020). *Principles and Criteria for the Production of Sustainable Palm Oil 2018*. `https://rspo.org/resources/certification/rspo-principles-criteria-certification`. Accessed on June 24, 2020.

Shah, N., Pantelides, C. C., & Sargent, R. W. H. (1993). Optimal periodic scheduling of multipurpose batch plants. *Annals of Operations Research*, 42, 193-228.

Sime Darby Plantation Berhad (2019a). *Differentiated (Specialty) Products*. `https://www.simedarbyoils.com/products/differentiated-products`. Accessed on October 16, 2019.

Sime Darby Plantation Berhad (2019b). *Consumer Brand Products*. `https://www.simedarbyoils.com/products/consumer-brand-products`. Accessed on October 16, 2019.

Sime Darby Plantation Berhad (2019c). *Refineries*. `https://www.simedarbyoils.com/refineries`. Accessed on October 16, 2019.

Sime Darby Plantation Berhad (2019d). *Commodity (Bulk) Products*. `https://www.simedarbyoils.com/products/commodity-products`. Accessed on October 16, 2019.

Sime Darby Plantation Berhad (n. d.). *Sime Darby*. `http://www.unimills.com/About-us/Sime-Darby`. Accessed on October 16, 2019.

Slingerland, M. (2016). Competitie om natuurlijke hulpbronnen. In K. Giller, G. Spaargaren, P. Oosterveer, B. Defize & R. Buiter (eds.), *Duurzame Derde Wereld* (pp. 21-32). Stichting Biowetenschappen en Maatschappij.

Team Superside (2018). *20+ Free Gantt Chart Templates that are Ready for Your Use | Excel, PPT, Word*. `https://www.superside.com/blog/gantt-charts`. Accessed on November 28, 2019.

The Lean Six Sigma Company (2019). *Overall Equipment Effectiveness (OEE)*. `https://www.theleansixsigmacompany.nl/overall-equipment-effectiveness-oee/`. Accessed on October 30, 2019.

United Nations, Department of Economic and Social Affairs, Population Division (2019). *World Population Prospects 2019*. `https://population.un.org/wpp/DataQuery/`. Accessed on October 16, 2019.

van Kasteren, J. (2013, September 10). 'Productie van voedsel moet efficiënter'. *Trouw*. https://www.trouw.nl/nieuws/productie-van-voedsel-moet-efficienter~b27d23c6/. Accessed on October 16, 2019.

Verleyen, T., Sosinska, U., Ioannidou, S., Verhe, R., Dewettinck, K., Heyghebaert, A., & De Greyt, W. (2002). Influence of the Vegetable Oil Refining Process on Free and Esterified Sterols. *Journal of the American Oil Chemists' Society*, 79 (10), 947-953.

Voedingscentrum (n. d.). *Transvet*. https://www.voedingscentrum.nl/encyclopedie/transvet.aspx. Accessed on October 17, 2019.

Vorne Industries Inc. (2019). *OEE factors*. https://www.oee.com/oee-factors.html. Accessed on October 30, 2019.

Wenkai, L., Hui, C. W., Hua, B., & Tong, Z. (2002). Scheduling Crudo Oil Unloading, Storage and Processing. *Industrial and Engineering Chemistry Research*, 41, 6723-6734.

Williams, H. P. (2000). Model Building in Mathematical Programming. *Computers and Chemical Engineering*, 24, 2259-2276.

List of Abbreviations

[...]	[...]
BL	Bleaching Line
BO	Soy Bean oil
C. I.	Chemical Interesterification
CDU	Crude Distillation Unit
CL	Centrifuge Line
CN	Coconut oil
CSPO	Certified Sustainable Palm Oil
DEO	Deodorizer
DF	Dry Fractionation
E. I.	Enzymatic Interesterification
FFA	Free Fatty Acids
FIS	Finite Intermediate Storage
MILP	Mixed-Integer Linear Programming
MINLP	Mixed-Integer Nonlinear Programming
NIS	Non-intermediate Storage
NLP	Nonlinear Programming
OEE	Overall Equipment Efficiency
OR	Operations Research
PK	Palm Kernel oil
PKcs	Certified Sustainable Palm Kernel oil
PO	Palm oil
POcs	Certified Sustainable Palm Oil
ROS	Refined Oil Storage
RP	Rapeseed oil
RSPO	Roundtable on Sustainable Palm Oil
RVF	Rest Volume Filtration
SDO	Sime Darby Oils
SDOZR	Sime Darby Oils Zwijndrecht Refinery
SF	Sunflower oil
SN	The restriction on maximum shelf-life is not taken into account
STN	State-Task-Network
SY	The restriction on maximum shelf-life is taken into account
TP [...]	[...]
UIS	Unlimited Intermediate Storage
ZW	Zero Wait

THIS PAGE IS LEFT BLANK INTENTIONALLY.

List of Figures

THIS PAGE IS LEFT BLANK INTENTIONALLY.

List of Tables

THIS PAGE IS LEFT BLANK INTENTIONALLY.

Listings

THIS PAGE IS LEFT BLANK INTENTIONALLY.

A

Classification roadmap for scheduling problems

The content of this overview is based on Méndez et al. (2006a).

- **1. Process topology**
  - Sequential
    - Multiple stages
      - Multipurpose (Job-shop)
      - Multiproduct (Flow-shop)
    - Single Stage
      - Parallel units
      - Single unit
  - Network (arbitrary)
- **2. Equipment assignment**
  - Fixed
  - Variable
- **3. Equipment connectivity**
  - Partial (restricted)
  - Full
- **4. Inventory storage policies**
  - Unlimited Intermediate storage (UIS)
  - Non-Intermediate Storage (NIS)
  - Finite Intermediate Storage (FIS)
    - Shared storage units
    - Dedicated storage units
  - Zero Wait (ZW)
- **5. Material transfer**
  - Instantaneous (neglected)
  - Time-consuming
    - Vessels (Pipeless)
    - Pipes
    - No-resources
- **6. Batch-size**
  - Fixed
  - Variable (Mixing and Splitting)

- **7. Batch processing time**
  - Fixed
    - Unit dependent
    - Unit independent
  - Variable (unit/batch-size dependent)

- **8. Demand patterns**
  - Due dates
    - Multiple product demands
    - Single product demand
  - Scheduling horizon
    - Minimum/maximum requirements
    - Fixed requirements

- **9. Changeovers**
  - None
  - Sequence dependent
    - Product and unit dependent
    - Product dependent
  - Unit dependent

- **10. Resource Constraints**
  - None (only equipment)
  - Discrete
  - Continuous

- **11. Time Constraints**
  - None
  - Non-working periods
  - Maintenance
  - Shifts

- **12. Costs**
  - Equipment
  - Utilities
  - Inventory
  - Changeover

- **13. Degree of certainty**
  - Deterministic
  - Stochastic

B

List of states and list of tasks

Table B.1: List of states

State number	State description	Product family
1.	POcs	
2.	cINEScs97NBu	Product family 2
3.	cINEScs97	Product family 2
4.	bINEScs97NF	Product family 2
5.	bINEScs97F	Product family 2
6.	bINEScs97	Product family 2
7.	cPOcssf40	
8.	cINEScs43bNF	Product family 1
9.	cINEScs43b	Product family 1
10.	bINEScs43NF	Product family 1
11.	bINEScs43F	Product family 1
12.	bINEScs43	Product family 1
13.	cPOcssf35	
14.	nPKcs	
15.	cINEScs82NBu	Product family 0
16.	cINEScs82	Product family 0
17.	bPOcssf40NF	Product family 1
18.	bPOcssf40F	Product family 1
19.	bPOcssf40	Product family 1
20.	PK39	
21.	cINES33bNF	Product family 0
22.	cINEs33b	Product family 0

Table B.2: List of tasks

Task number	Task description	Number of required time steps
1.	ClcINES97R_225	[...]
2.	ClcINES97R_30	[...]
3.	BucINEScs97NBu	[...]

Table B.2: List of tasks

Task number	Task description	Number of required time steps
4.	SL12cINEScs97_225	[...]
5.	SL12cINEScs97_30	[...]
6.	BcINEScs97_225	[...]
7.	BcINEScs97_30	[...]
8.	SL12cINEScs97_375	[...]
9.	SL12cINEScs97_45	[...]
10.	FbINEScs97NF_225	[...]
11.	FbINEScs97NF_30	[...]
12.	BubINEScs97F	[...]
13.	ClcPOcssf40_225	[...]
14.	ClcPOcssf40_30	[...]
15.	ClcPOcssf40_375	[...]
16.	ClcPOcssf40_45	[...]
17.	FcINEScs43bNF_225	[...]
18.	FcINEScs43bNF_30	[...]
19.	FcINEScs43bNF_375	[...]
20.	FcINEScs43bNF_45	[...]
21.	BcINEScs43b_225	[...]
22.	BcINEScs43b_30	[...]
23.	SL12cINEScs43b_225	[...]
24.	SL12cINEScs43b_30	[...]
25.	FbINEScs43NF_225	[...]
26.	FbINEScs43NF_30	[...]
27.	SL12cINEScs43b_375	[...]
28.	SL12cINEScs43b_45	[...]
29.	BubINEScs43F	[...]
30.	ClcINEScs82R_225	[...]
31.	ClcINEScs82R_30	[...]
32.	BucINEScs82NBu	[...]
33.	SL12cPOcssf40NF_225	[...]
34.	SL12cPOcssf40NF_30	[...]
35.	BcPOcssf40_225	[...]
36.	BcPOcssf40_30	[...]
37.	SL12cPOcssf40NF_375	[...]
38.	SL12cPOcssf40NF_45	[...]
39.	FbPOcssf40NF_225	[...]
40.	FbPOcssf40NF_30	[...]
41.	BubPOcssf40F	[...]
42.	ClcINES33bR_225	[...]
43.	ClcINES33bR_30	[...]
44.	ClcINES33bR_375	[...]

Table B.2: List of tasks

Task number	Task description	Number of required time steps
45.	ClcINES33bR_45	[...]
46.	FcINES33bNF_225	[...]
47.	FcINES33bNF_30	[...]
48.	FcINES33bNF_375	[...]
49.	FcINES33bNF_45	[...]

THIS PAGE IS LEFT BLANK INTENTIONALLY.

C
Demand

Table C.1: A realistic demand for SDOZR based on internal documents

Real oil	Example oil	Amount	Delivery day	Delivery time	Time step	Implemented
bPOcsdn	bPOcssf40	[...]	[...]	[...]	[...]	No, IA
bPOcsdn	bPOcssf40	[...]	[...]	[...]	[...]	No, IA
cINEScs48	cINES82	[...]	[...]	[...]	[...]	No, IA
bPOcsdn	bPOcssf40	[...]	[...]	[...]	[...]	No, IA
bPOcs	bPOcssf40	[...]	[...]	[...]	[...]	No, IA
bPOcs45	bPOcssf40	[...]	[...]	[...]	[...]	No
bPOcs45	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bINEScs96	bINEScs43	[...]	[...]	[...]	[...]	Yes
bMAcs1152	bPOcssf40	[...]	[...]	[...]	[...]	No
bMAcs1152	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bRP68	bPOcssf40	[...]	[...]	[...]	[...]	No
bBOip65	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bBOip65	bPOcssf40	[...]	[...]	[...]	[...]	No
bPOcs58	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bPOcs58	bPOcssf40	[...]	[...]	[...]	[...]	No
bPOcs58	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bPOcs58	bPOcssf40	[...]	[...]	[...]	[...]	No
bMAcs1054	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bPOcss35	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bINEScs43	-	[...]	[...]	[...]	[...]	No
bMAcs1152	bPOcssf40	[...]	[...]	[...]	[...]	Yes
cINES1027	cINES82	[...]	[...]	[...]	[...]	Yes
bMAcs1152	bPOcssf40	[...]	[...]	[...]	[...]	No
bMAcs1152	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bPOcsdn	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bBOip37	bPOcssf40	[...]	[...]	[...]	[...]	No
bRP	bPOcssf40	[...]	[...]	[...]	[...]	No

Table C.1: A realistic demand for SDOZR based on internal documents

Real oil	Example oil	Amount	Delivery day	Delivery time	Time step	Implemented
bPOcsdn	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bPOcsdn	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bPOcsdn	bPOcssf40	[...]	[...]	[...]	[...]	No
bRP	bPOcssf40	[...]	[...]	[...]	[...]	No
bPOcs45	bPOcssf40	[...]	[...]	[...]	[...]	No
bPOcsfs42	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bPOcsf42	bPOcssf40	[...]	[...]	[...]	[...]	No
bRP38	bPOcssf40	[...]	[...]	[...]	[...]	No
bINEScs52	bINEScs43	[...]	[...]	[...]	[...]	Yes
bPOcs58	bPOcssf40	[...]	[...]	[...]	[...]	No
bPOcs58	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bPOcsdn	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bPOcsdn	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bPOcsdn	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bINEScs43	-	[...]	[...]	[...]	[...]	No
bMAcs1152	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bPOcs45	bPOcssf40	[...]	[...]	[...]	[...]	No
bPOcsdn	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bMAcs1054	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bPOcsdn	bPOcssf40	[...]	[...]	[...]	[...]	No
bPOcsdn	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bPOcsdn	bPOcssf40	[...]	[...]	[...]	[...]	No
bPOcsdn	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bPOcsf42	bPOcssf40	[...]	[...]	[...]	[...]	No
bINEScs1040	bINEScs97	[...]	[...]	[...]	[...]	Yes
bINEScs64	bINEScs43	[...]	[...]	[...]	[...]	Yes
bPOcssf40	-	[...]	[...]	[...]	[...]	No
INEScs75b	cINES33b	[...]	[...]	[...]	[...]	Yes
bMAcs1054	bPOcssf40	[...]	[...]	[...]	[...]	Yes
INEScs75b	cINES33b	[...]	[...]	[...]	[...]	Yes
bMAcs1054	bPOcssf40	[...]	[...]	[...]	[...]	No
bPOs35	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bMAcs1054	bPOcssf40	[...]	[...]	[...]	[...]	No
bINEScs52	bINEScs43	[...]	[...]	[...]	[...]	Yes
bRP68	bPOcssf40	[...]	[...]	[...]	[...]	No
bPOcsdn	bPOcssf40	[...]	[...]	[...]	[...]	No
bBOip65	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bPOcs58	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bPOcs58	bPOcssf40	[...]	[...]	[...]	[...]	No
bPOcsdn	bPOcssf40	[...]	[...]	[...]	[...]	No
bPOcs58	bPOcssf40	[...]	[...]	[...]	[...]	Yes

Table C.1: A realistic demand for SDOZR based on internal documents

Real oil	Example oil	Amount	Delivery day	Delivery time	Time step	Implemented
bRP68	bPOcssf40	[...]	[...]	[...]	[...]	No
bPOcsss12	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bMAcs1152	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bINEscs97	-	[...]	[...]	[...]	[...]	No
cINEScs30b	cINES33b	[...]	[...]	[...]	[...]	Yes
bBOip37	bPOcssf40	[...]	[...]	[...]	[...]	Yes
cINES1068	cINES82	[...]	[...]	[...]	[...]	Yes
bPOcs45	bPOcssf40	[...]	[...]	[...]	[...]	No
bPOcsf56	bPOcssf40	[...]	[...]	[...]	[...]	Yes
cINEScs48	cINES82	[...]	[...]	[...]	[...]	Yes
bINEScs1040	bINEScs97	[...]	[...]	[...]	[...]	Yes
bPOcssf64	bPOcssf40	[...]	[...]	[...]	[...]	No
cINES35b	cINES33b	[...]	[...]	[...]	[...]	Yes
bPOcsf42	bPOcssf40	[...]	[...]	[...]	[...]	No
bINEScs52	bINEScs43	[...]	[...]	[...]	[...]	Yes
cINEScs44b	cINES33b	[...]	[...]	[...]	[...]	Yes
bPOcsf42	bPOcssf40	[...]	[...]	[...]	[...]	No
bMAcs1152	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bMAcs1152	bPOcssf40	[...]	[...]	[...]	[...]	No
bMAcs1152	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bMAcs1152	bPOcssf40	[...]	[...]	[...]	[...]	No
bPOcs45	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bPOcssf40	-	[...]	[...]	[...]	[...]	Yes
cINES1067b	cINES33b	[...]	[...]	[...]	[...]	Yes
bPOcs58	bPOcssf40	[...]	[...]	[...]	[...]	No
bPOcs58	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bPOcs58	bPOcssf40	[...]	[...]	[...]	[...]	No
bINEScs64	bINEScs43	[...]	[...]	[...]	[...]	Yes
bRP	bPOcssf40	[...]	[...]	[...]	[...]	No
bPOcs	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bINEScs1040	bINEScs97	[...]	[...]	[...]	[...]	Yes
bPOcsf42	bPOcssf40	[...]	[...]	[...]	[...]	No
bMAcs1152	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bINEScs43	-	[...]	[...]	[...]	[...]	No
bMAcs1152	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bMAcs1152	bPOcssf40	[...]	[...]	[...]	[...]	No
bBOip65	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bPOcs58	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bPOcs58	bPOcssf40	[...]	[...]	[...]	[...]	No
bRP38	bPOcssf40	[...]	[...]	[...]	[...]	No
bPOcs45	bPOcssf40	[...]	[...]	[...]	[...]	No

Table C.1: A realistic demand for SDOZR based on internal documents

Real oil	Example oil	Amount	Delivery day	Delivery time	Time step	Implemented
bINEScs43	-	[...]	[...]	[...]	[...]	Yes
bPOcs45	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bMAcs1159	bPOcssf40	[...]	[...]	[...]	[...]	No
bINEScs96	bINEScs43	[...]	[...]	[...]	[...]	No
bPOcsdn	bPOcssf40	[...]	[...]	[...]	[...]	Yes
bPOcsdn	bPOcssf40	[...]	[...]	[...]	[...]	No

D
Results

Objective 4.1: Scheduling horizon of 240 time steps

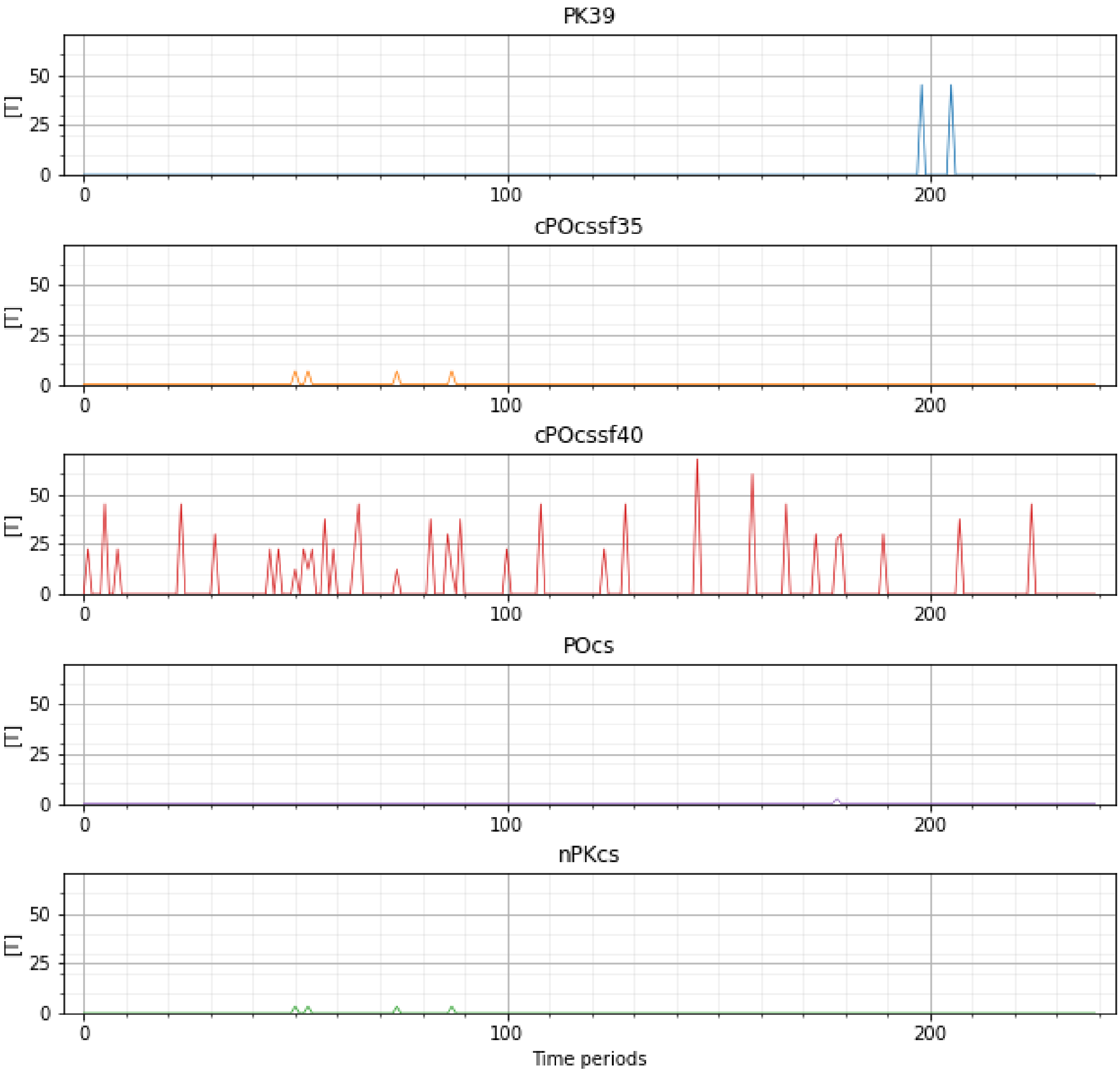

Figure D.1: Line graph of the decision variable R_{st} for the scheduling horizon of 240 time steps and the objective given by Equation 4.1.

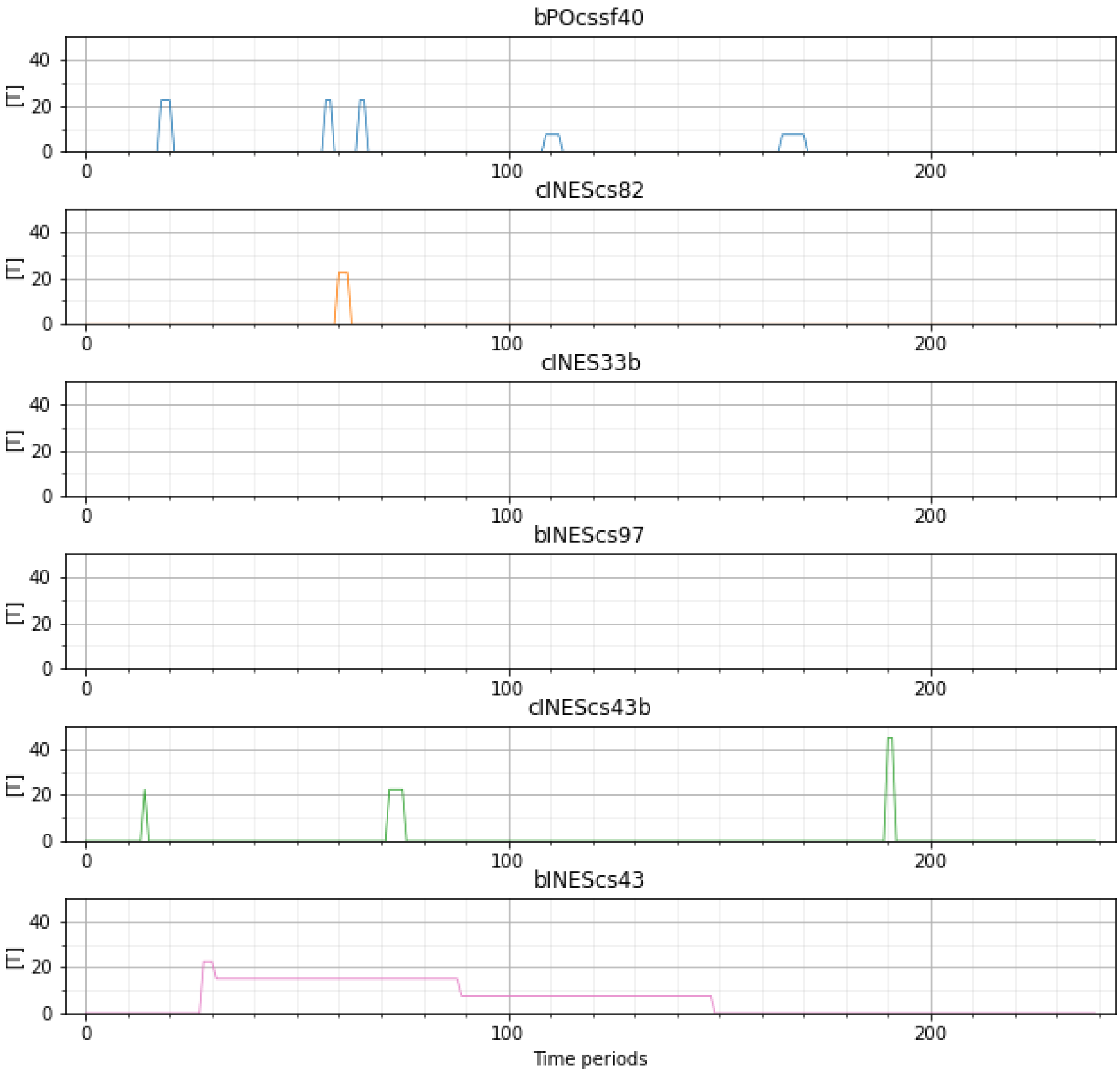

Figure D.2: Line graph of the decision variable S_{st} for the scheduling horizon of 240 time steps and the objective given by Equation 4.1.

Objective 4.1: Scheduling horizon of 480 time steps

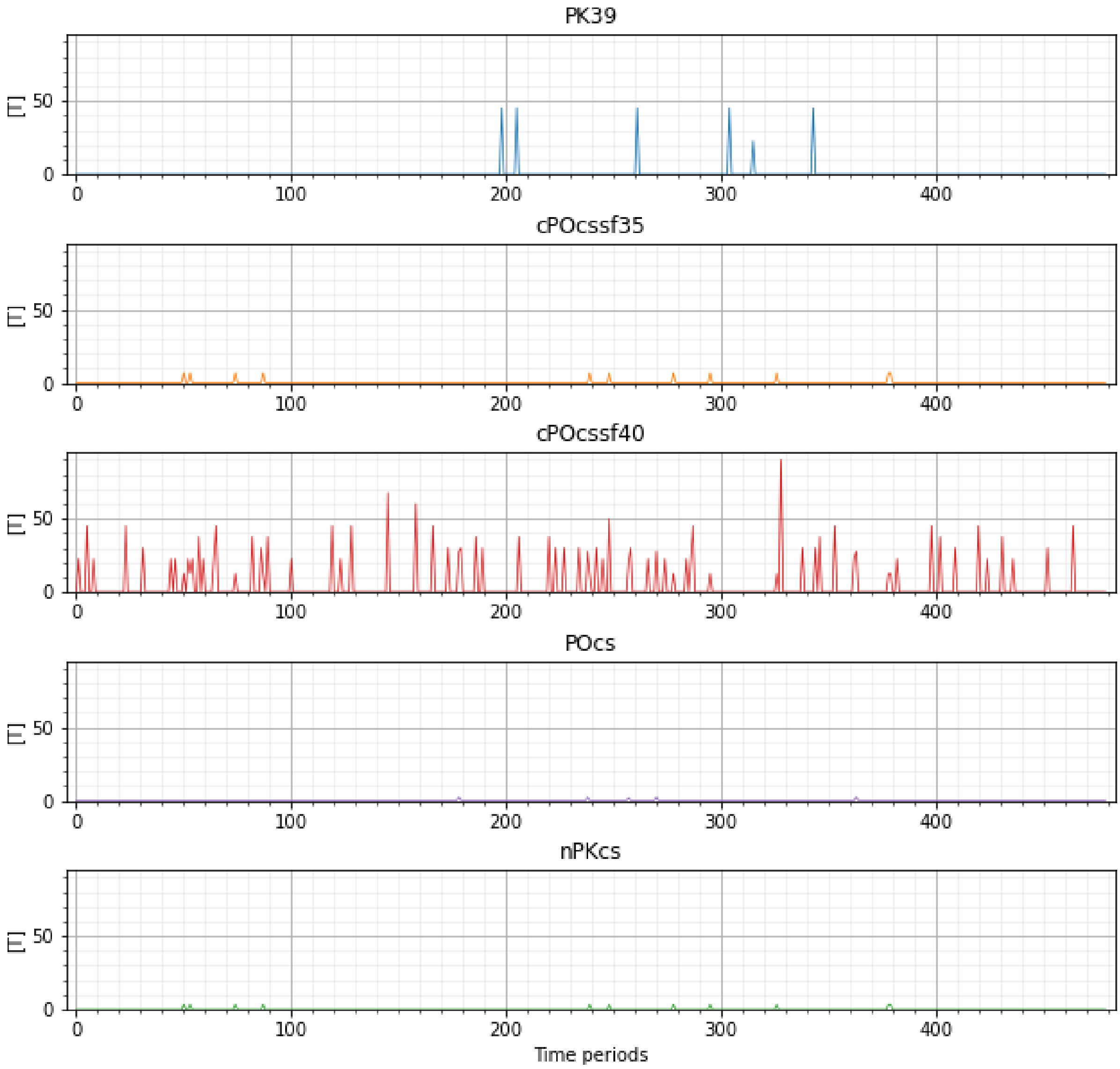

Figure D.3: Line graph of the decision variable R_{st} for the scheduling horizon of 480 time steps and the objective given by Equation 4.1.

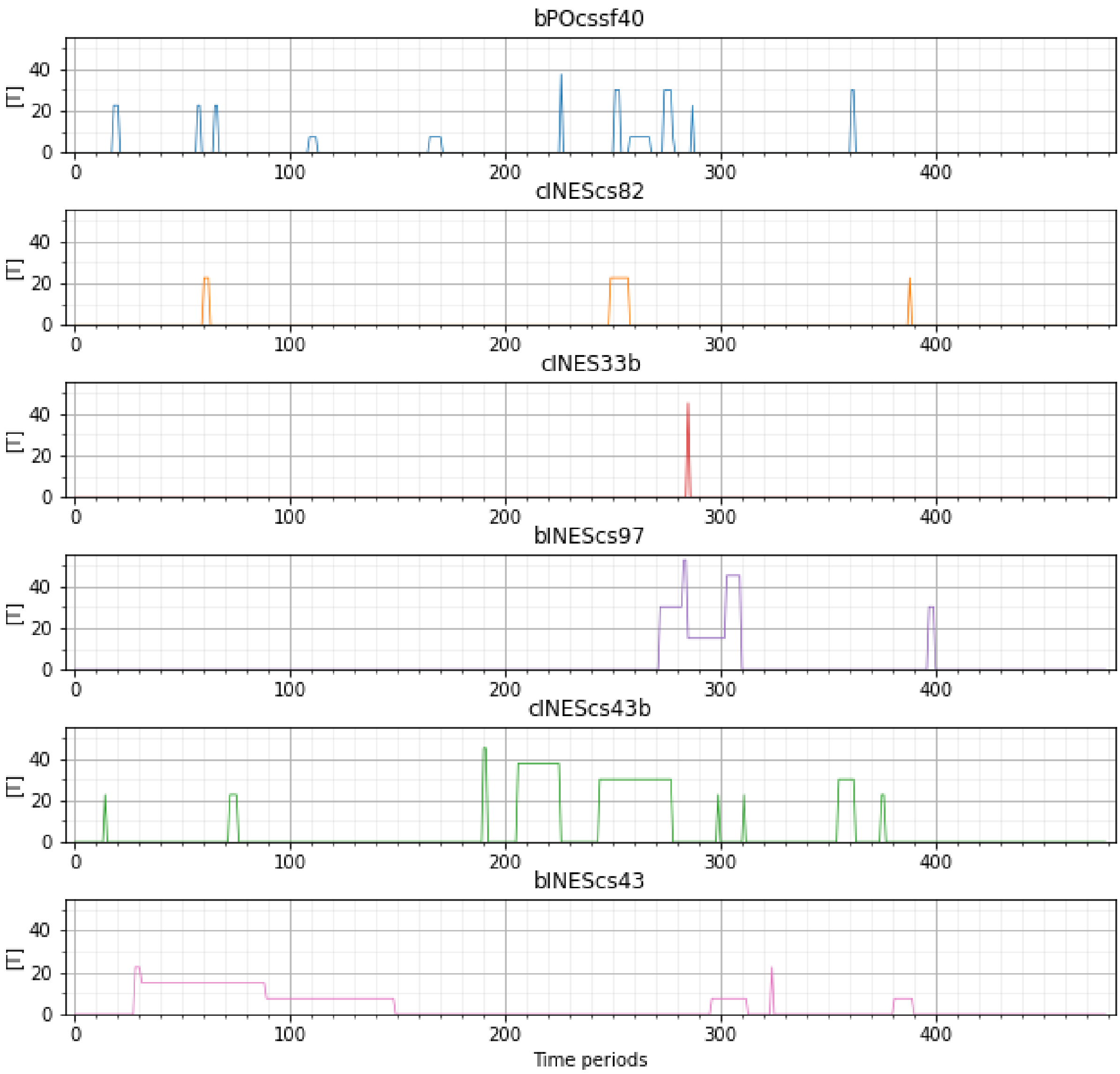

Figure D.4: Line graph of the decision variable S_{st} for the scheduling horizon of 480 time steps and the objective given by Equation 4.1.

Objective 5.1: Scheduling horizon of 240 time steps

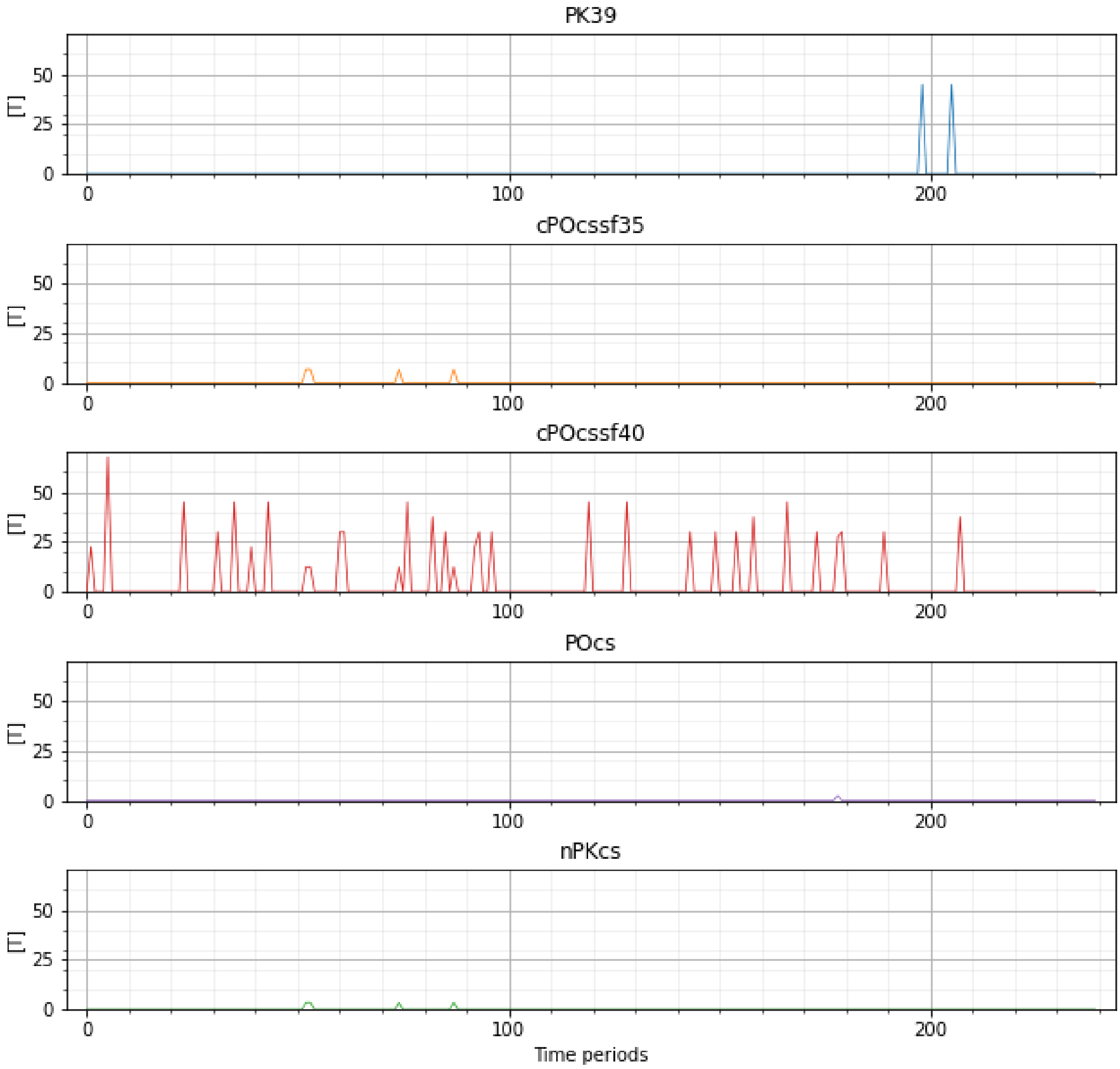

Figure D.5: Line graph of the decision variable R_{st} for the scheduling horizon of 240 time steps and the objective given by Equation 5.1.

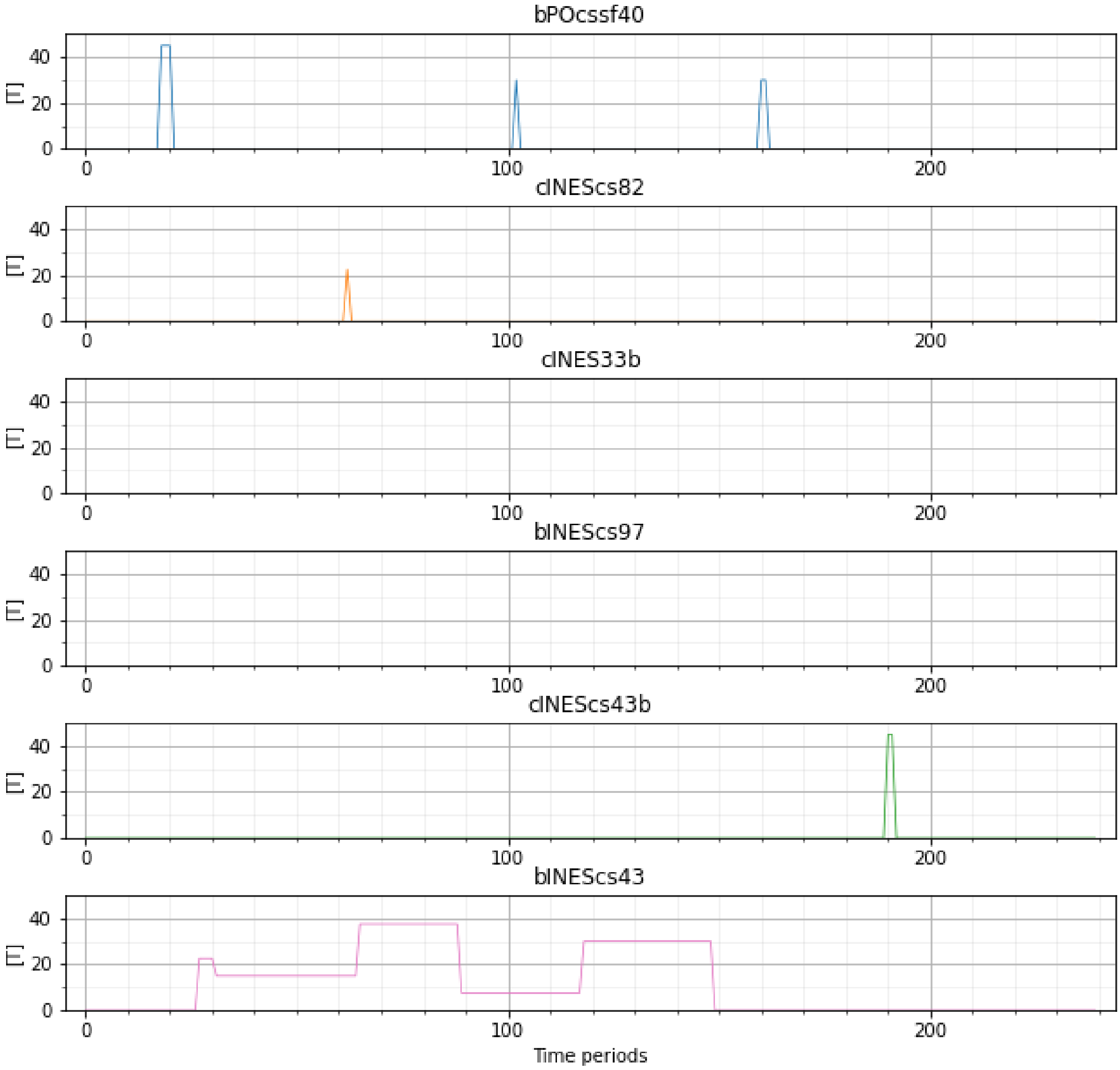

Figure D.6: Line graph of the decision variable S_{st} for the scheduling horizon of 240 time steps and the objective given by Equation 5.1.

Objective 5.3: Scheduling horizon of 240 time steps

With limitation on shelf-life

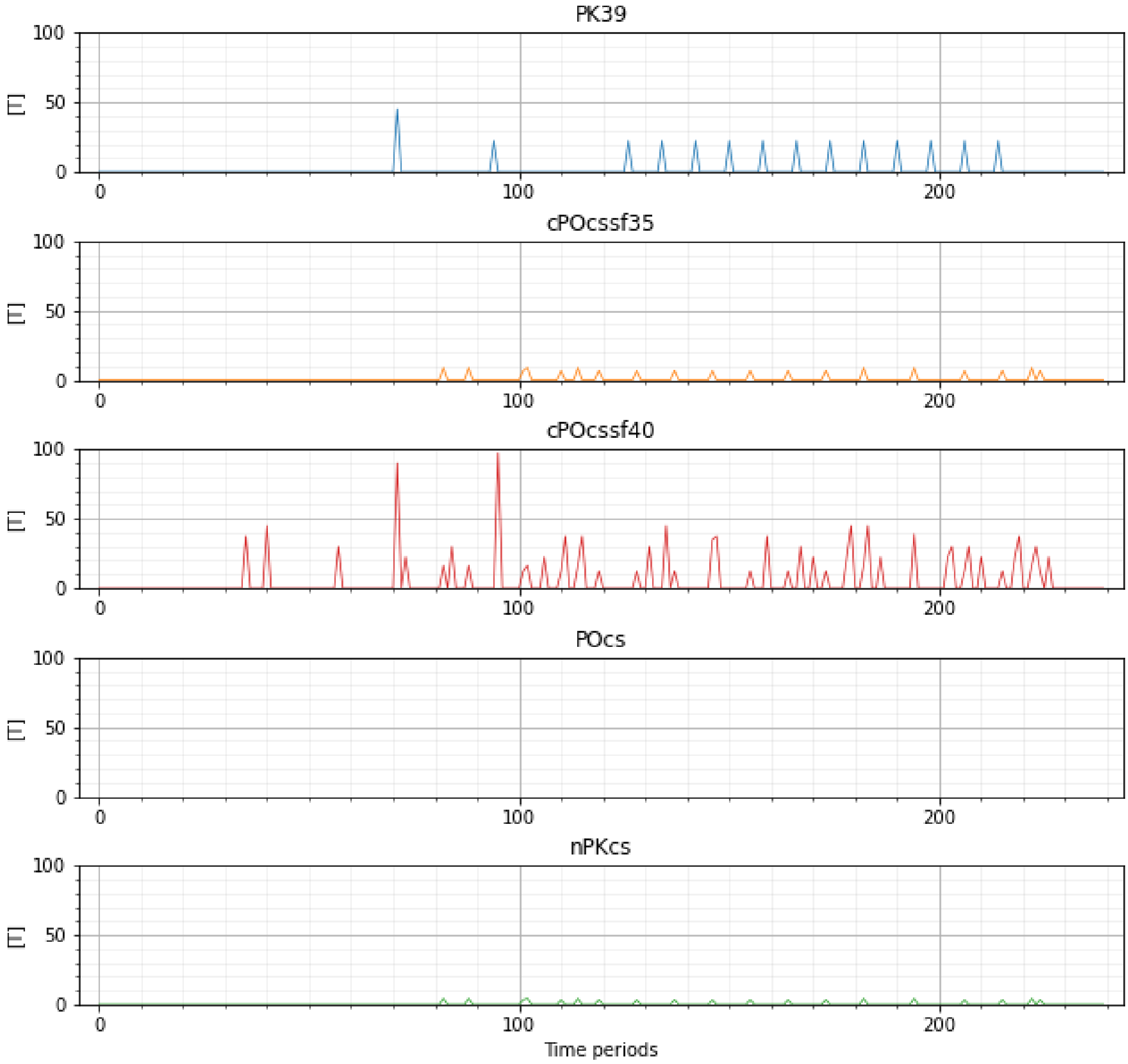

Figure D.7: Line graph of the decision variable R_{st} for the scheduling horizon of 240 time steps, the objective given by Equation 5.3 and with a limitation on shelf-life.

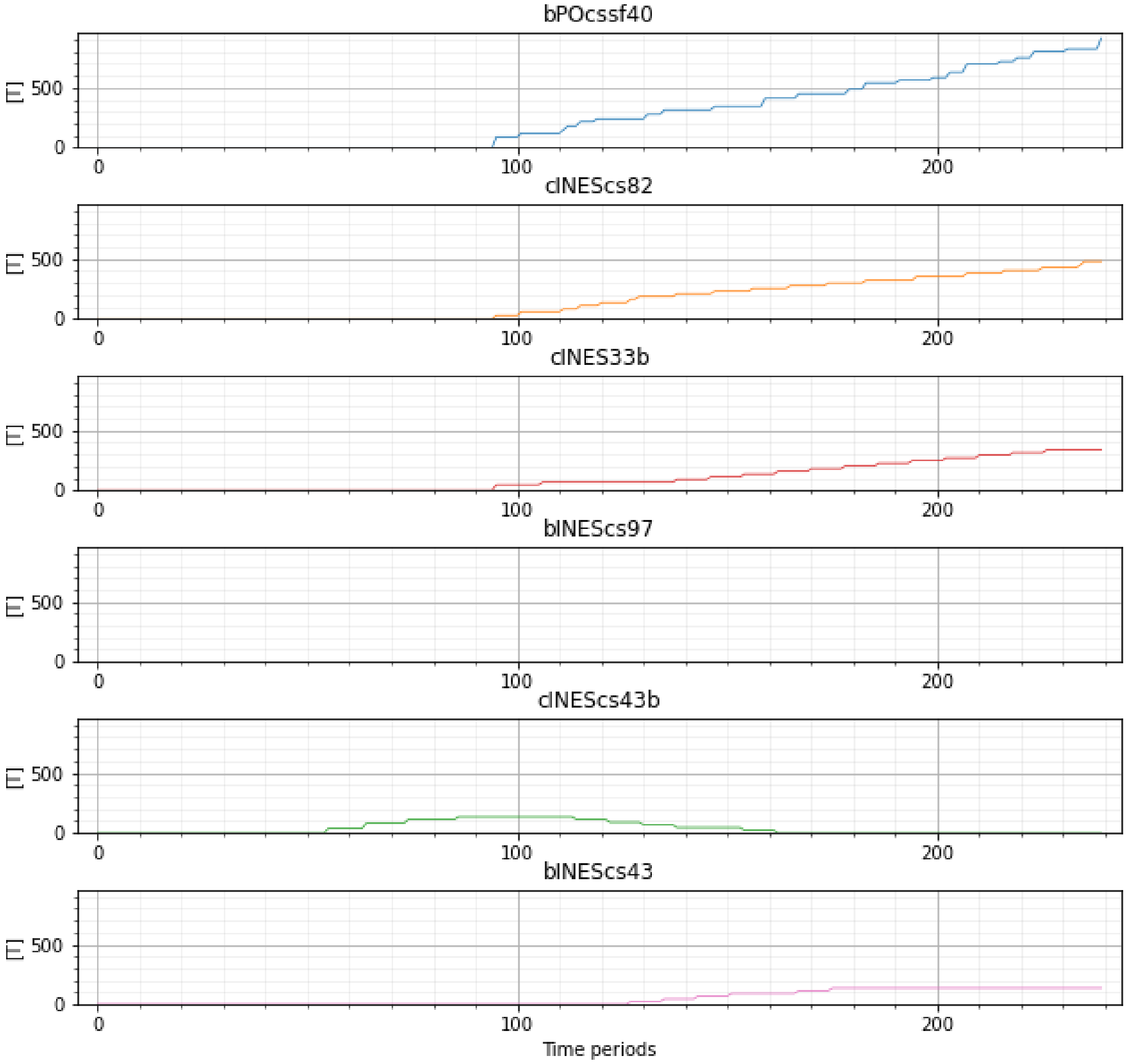

Figure D.8: Line graph of the decision variable S_{st} for the scheduling horizon of 240 time steps, the objective given by Equation 5.3 and with a limitation on shelf-life.

Objective 5.3: Scheduling horizon of 480 time steps

With limitation on shelf-life

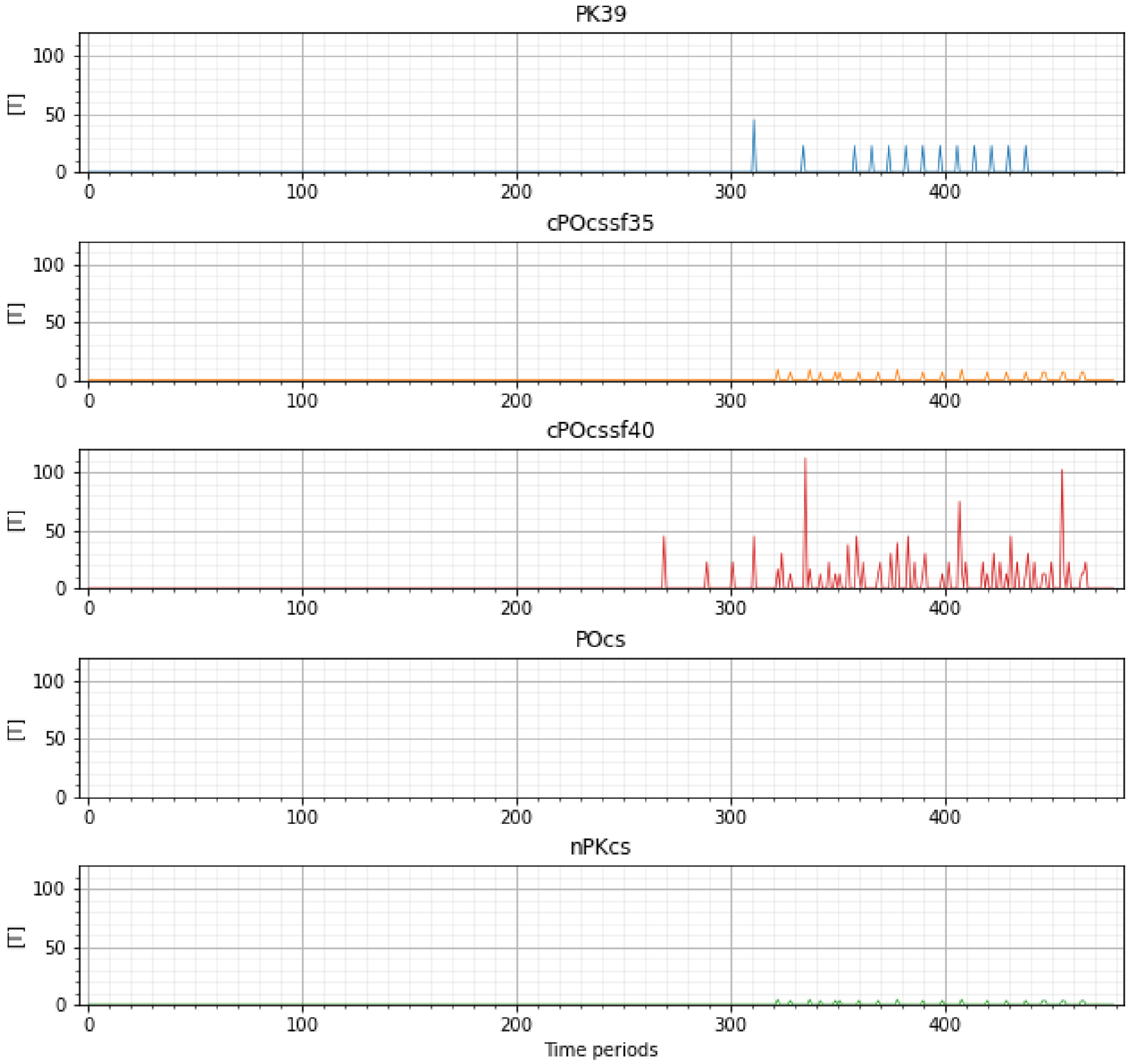

Figure D.9: Line graph of the decision variable R_{st} for the scheduling horizon of 480 time steps, the objective given by Equation 5.3 and with a limitation on shelf-life.

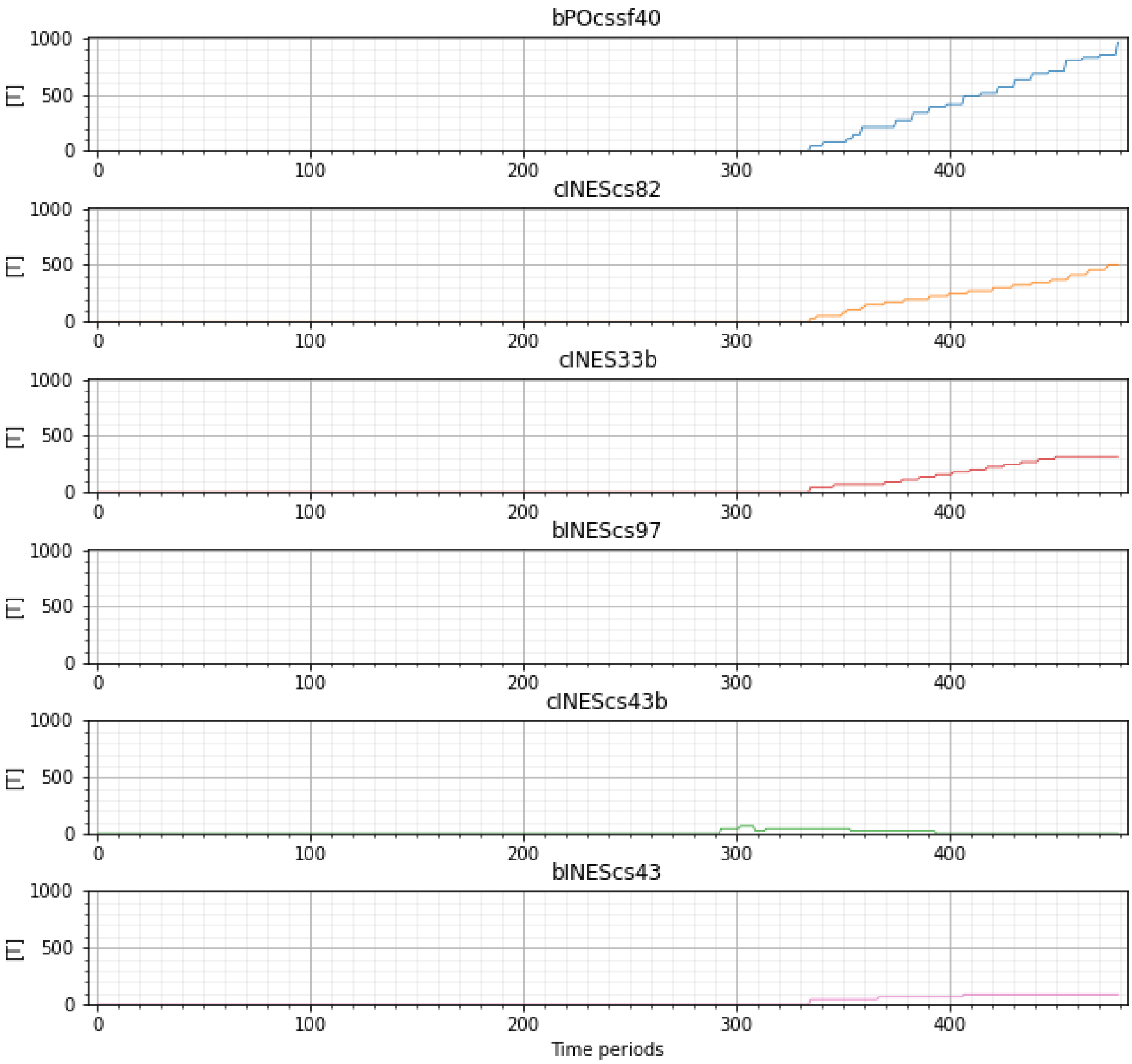

Figure D.10: Line graph of the decision variable S_{st} for the scheduling horizon of 480 time steps, the objective given by Equation 5.3 and with a limitation on shelf-life.

Objective 5.3: Scheduling horizon of 240 time steps

Without limitation on shelf-life

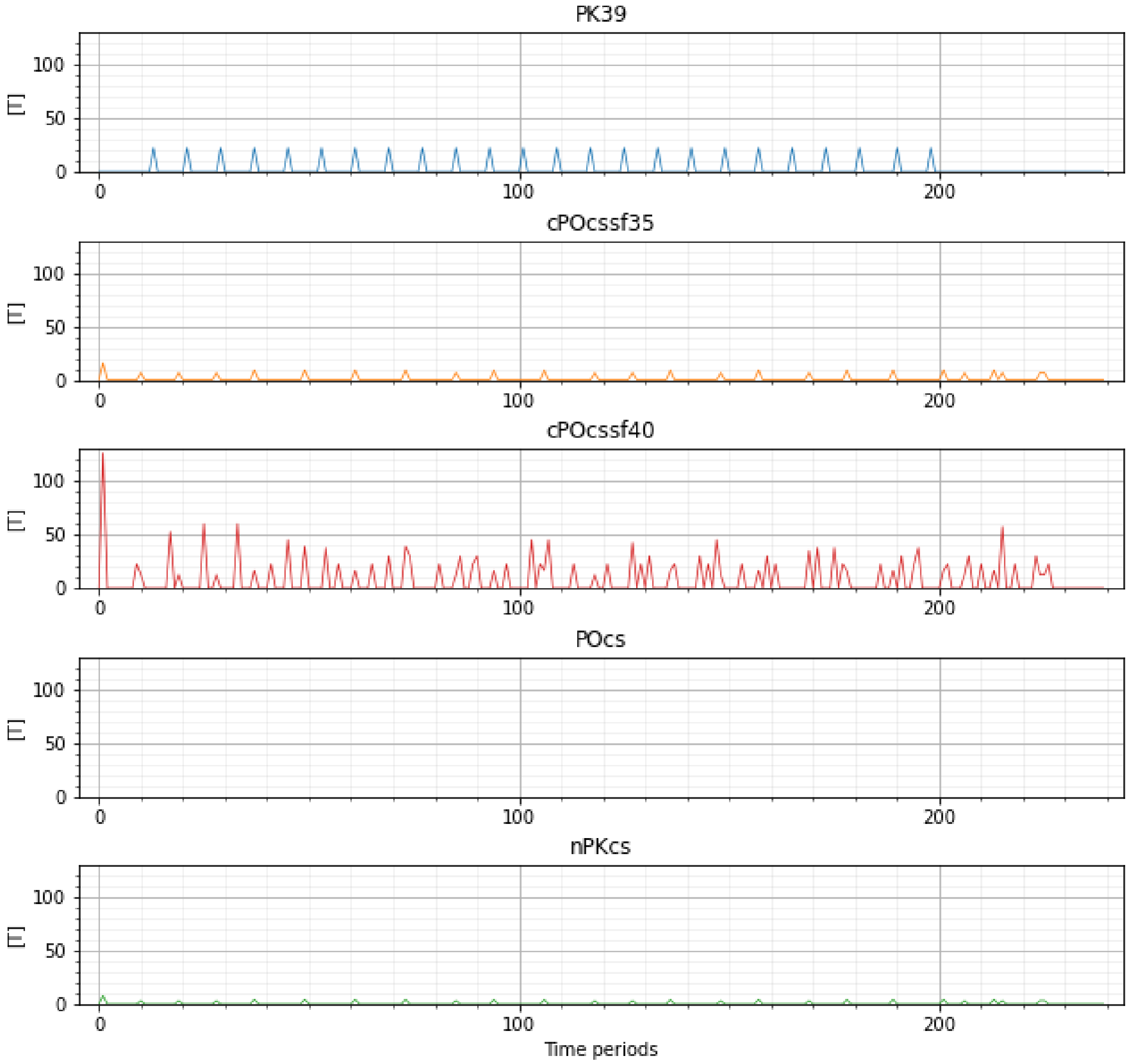

Figure D.11: Line graph of the decision variable R_{st} for the scheduling horizon of 240 time steps, the objective given by Equation 5.3 and without a limitation on shelf-life.

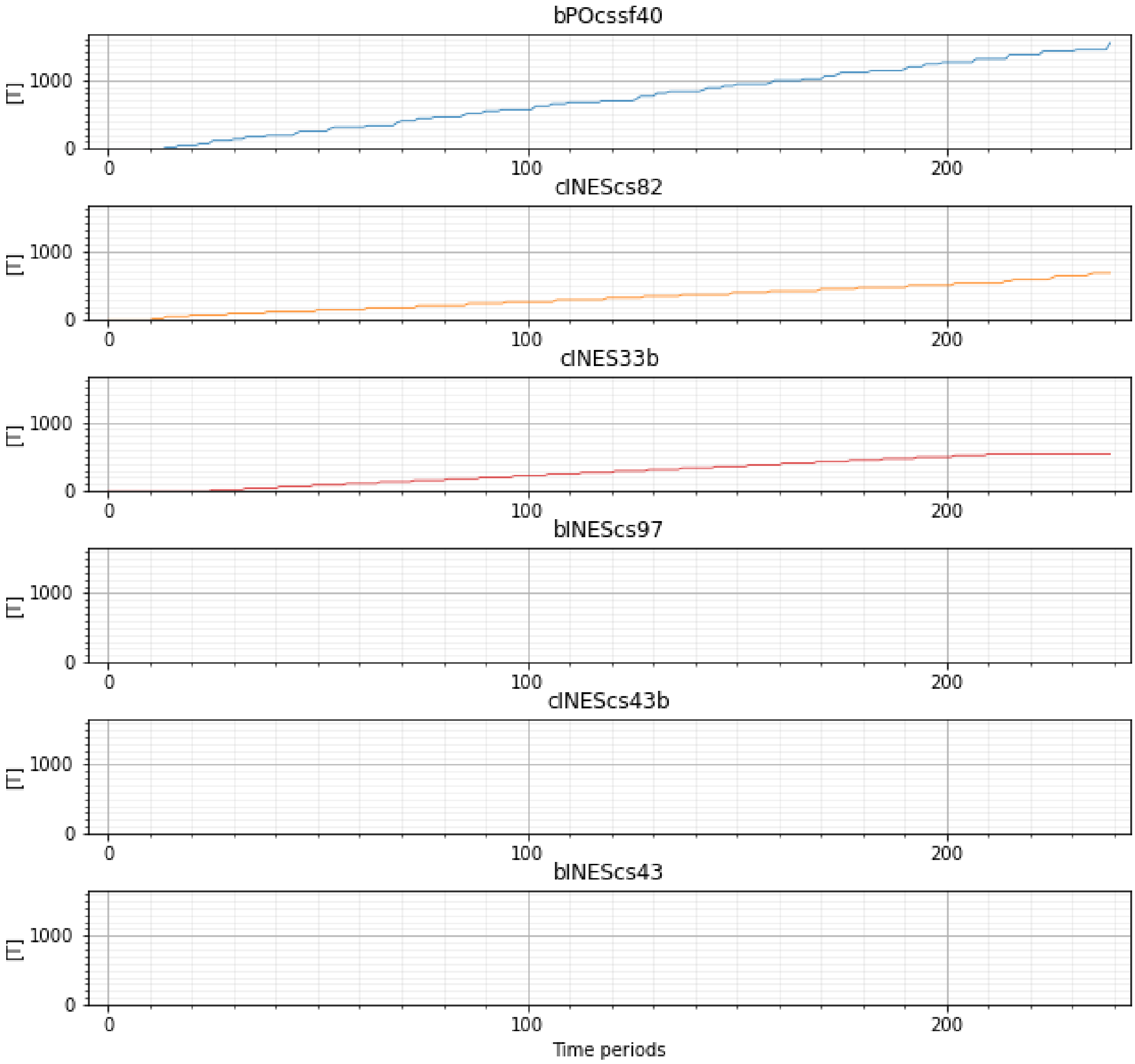

Figure D.12: Line graph of the decision variable S_{st} for the scheduling horizon of 240 time steps, the objective given by Equation 5.3 and without a limitation on shelf-life.

Objective 5.3: Scheduling horizon of 480 time steps

Without limitation on shelf-life

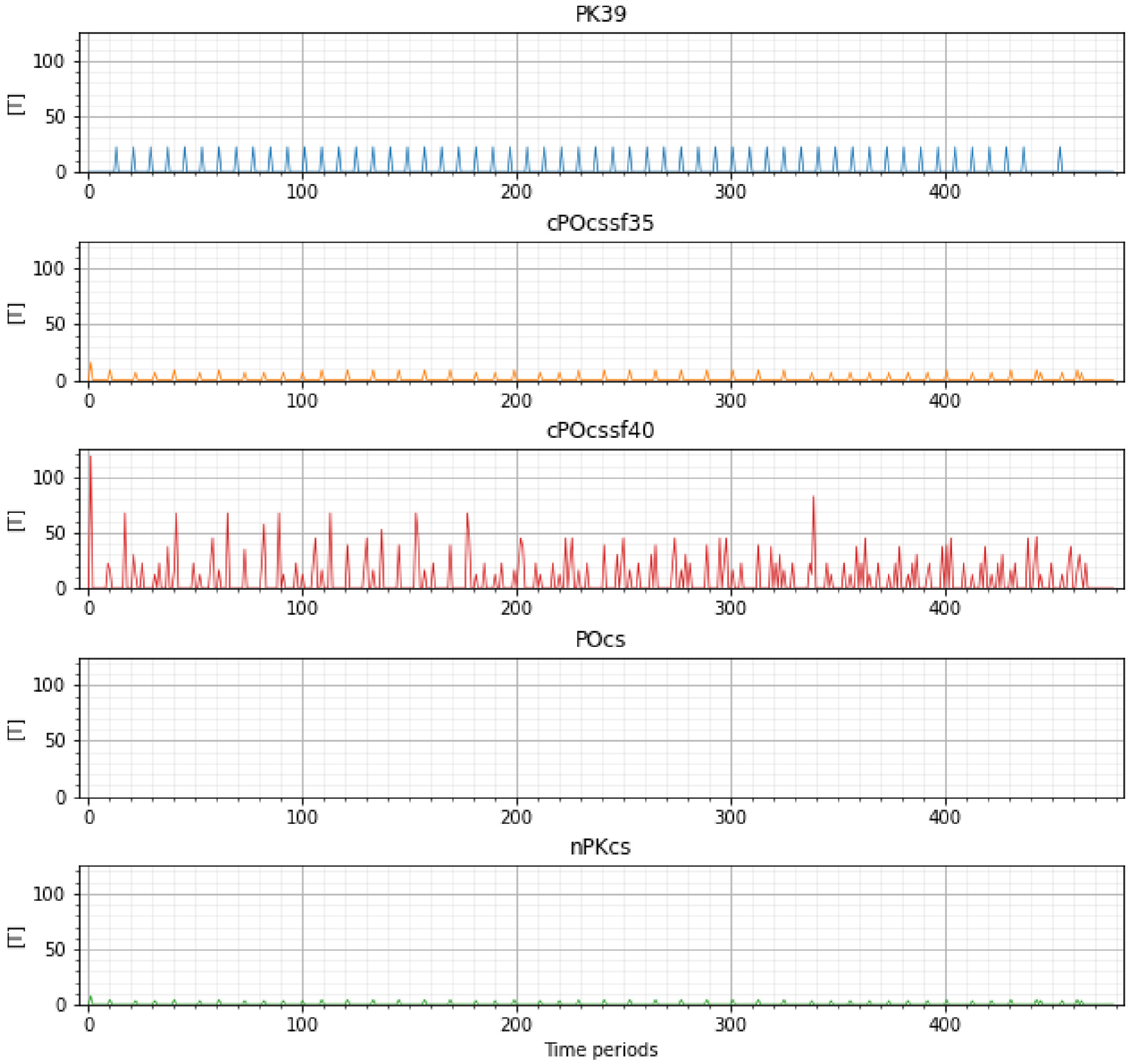

Figure D.13: Line graph of the decision variable R_{st} for the scheduling horizon of 480 time steps, the objective given by Equation 5.3 and without a limitation on shelf-life.

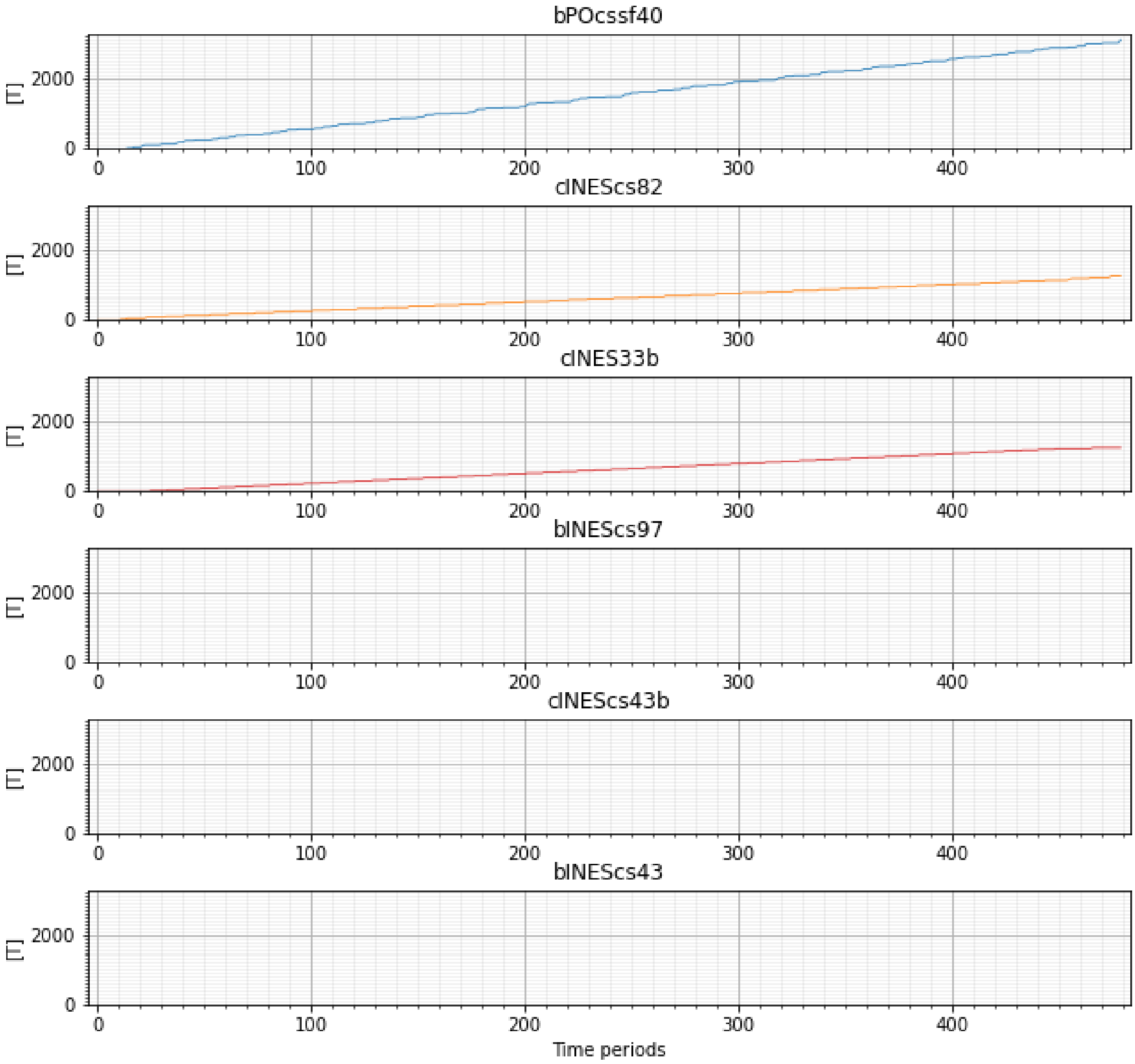

Figure D.14: Line graph of the decision variable S_{st} for the scheduling horizon of 480 time steps, the objective given by Equation 5.3 and without a limitation on shelf-life.

Objective 4.1: Sensitivity analysis on processing order

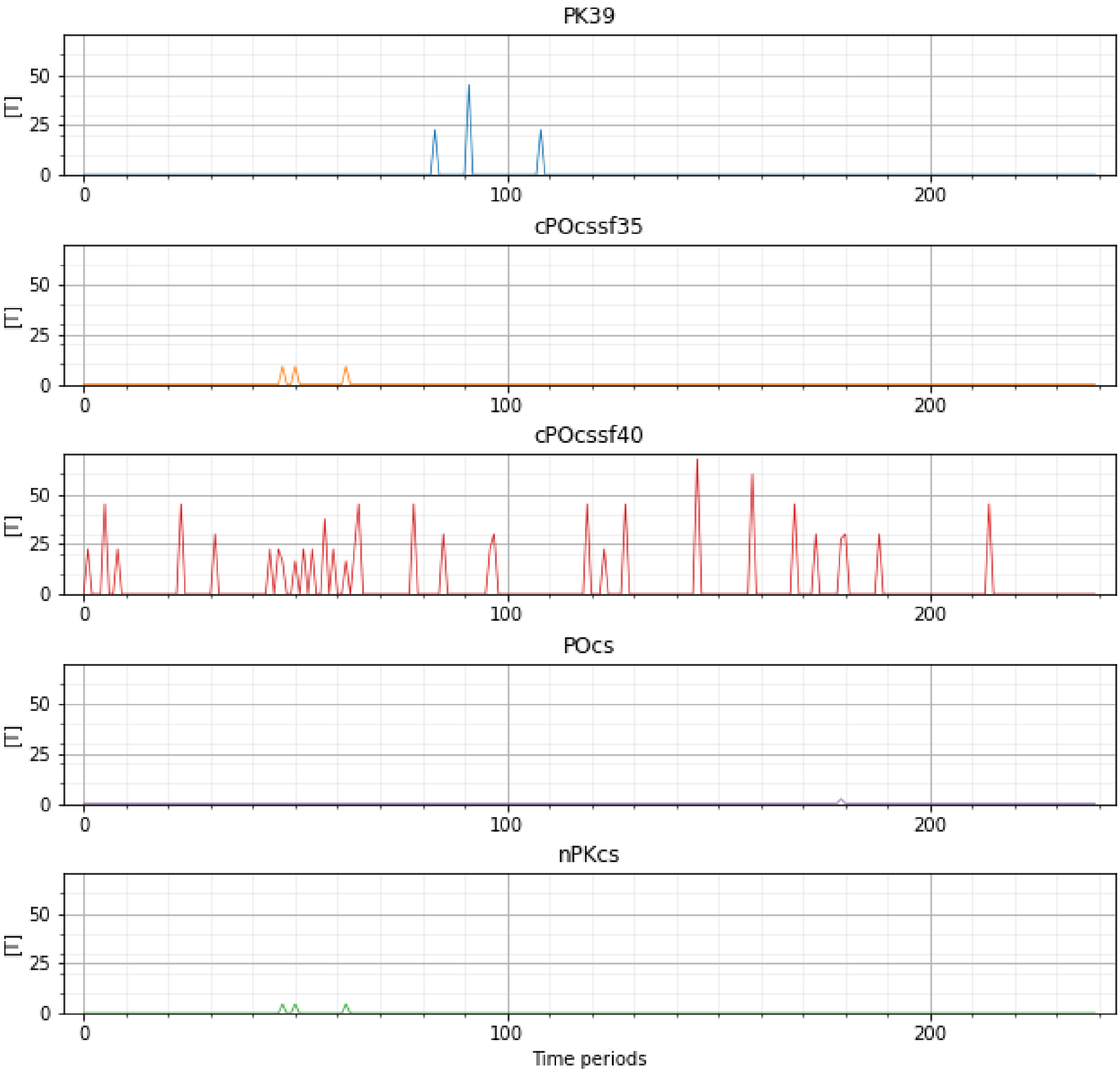

Figure D.15: Line graph of the decision variable R_{st} for the scheduling horizon of 240 time steps, the objective given by Equation 4.1 and a cleaning time τ_{jkl} that equals half of the scheduling horizon.

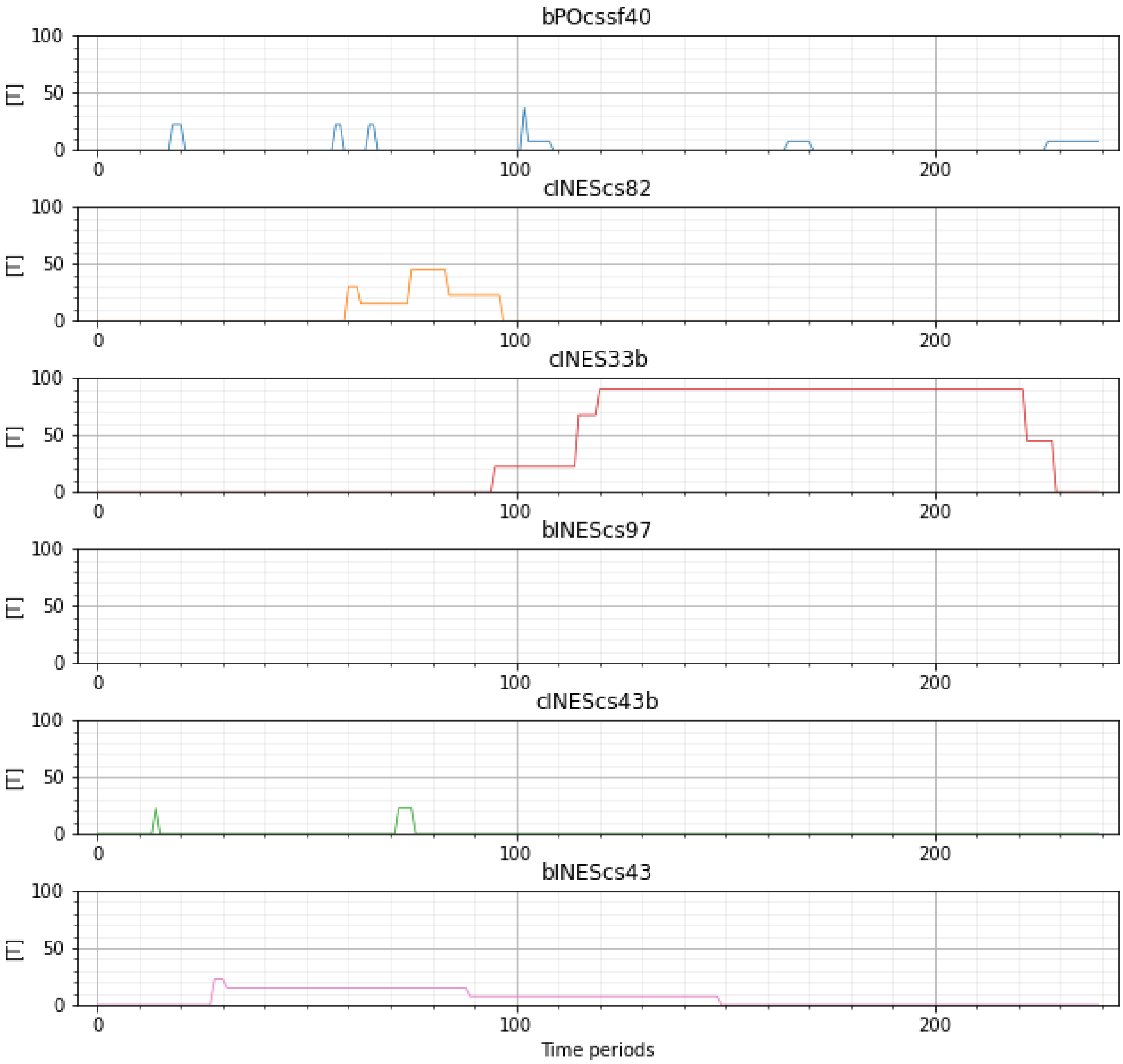

Figure D.16: Line graph of the decision variable S_{st} for the scheduling horizon of 240 time steps, the objective given by Equation 4.1 and a cleaning time τ_{jkl} that equals half of the scheduling horizon.

Objective 5.1: Sensitivity analysis on processing order

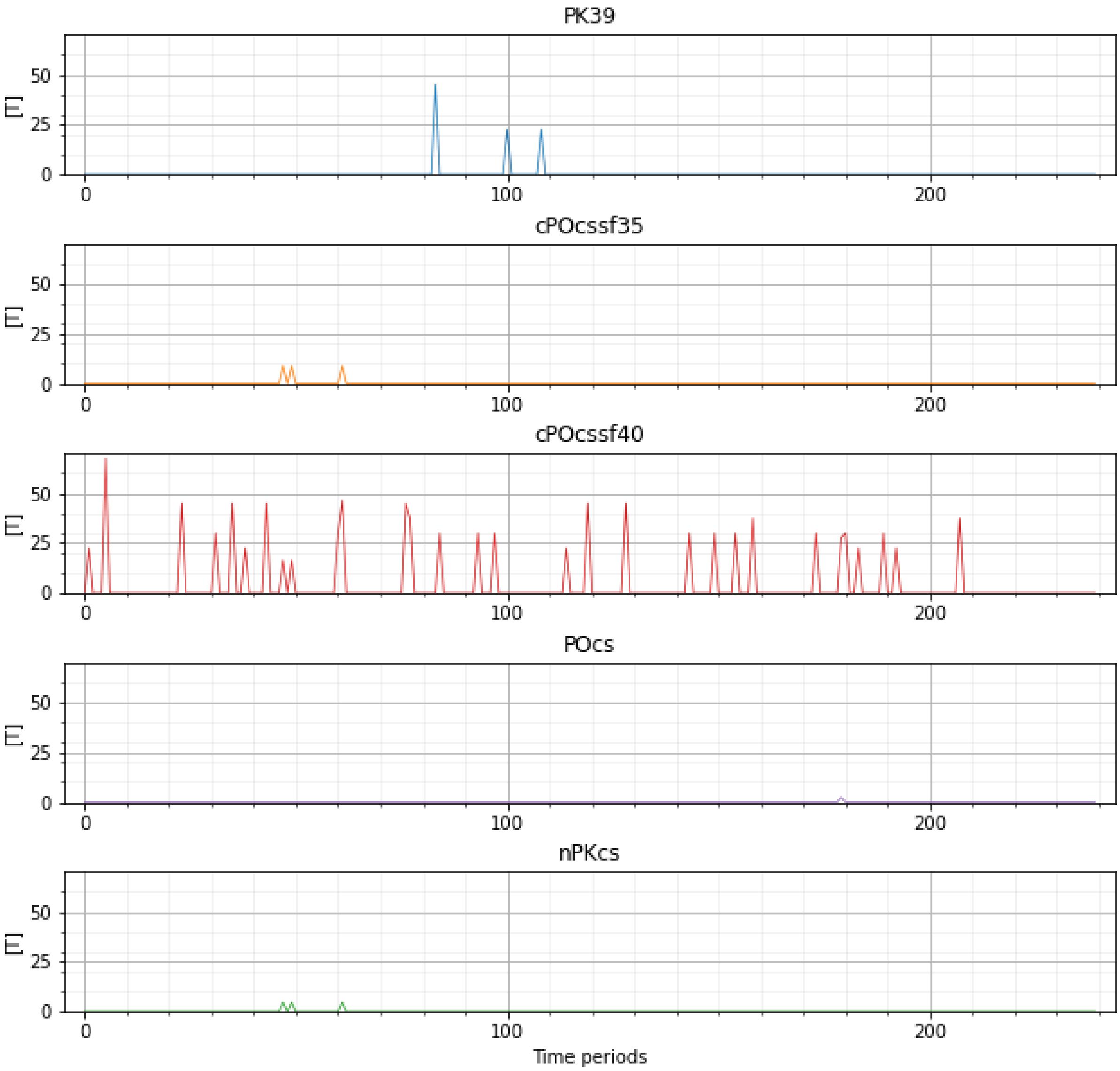

Figure D.17: Line graph of the decision variable R_{st} for the scheduling horizon of 240 time steps, the objective given by Equation 5.1 and a cleaning time τ_{jkl} that equals half of the scheduling horizon.

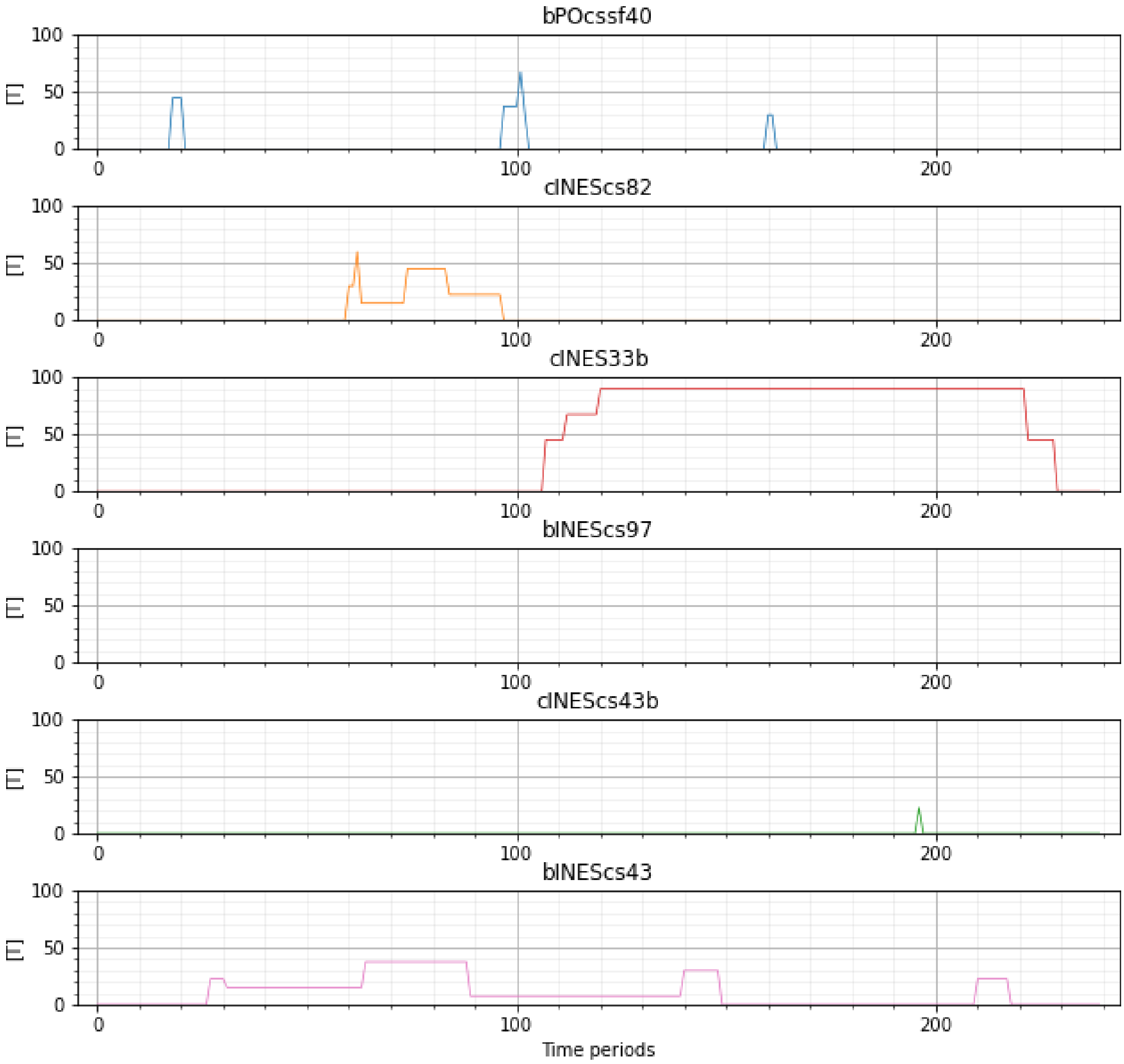

Figure D.18: Line graph of the decision variable S_{st} for the scheduling horizon of 240 time steps, the objective given by Equation 5.1 and a cleaning time τ_{jkl} that equals half of the scheduling horizon.

Objective 5.3: Sensitivity analysis on processing order

With limitation on shelf-life - cleaning time equals half of the scheduling horizon

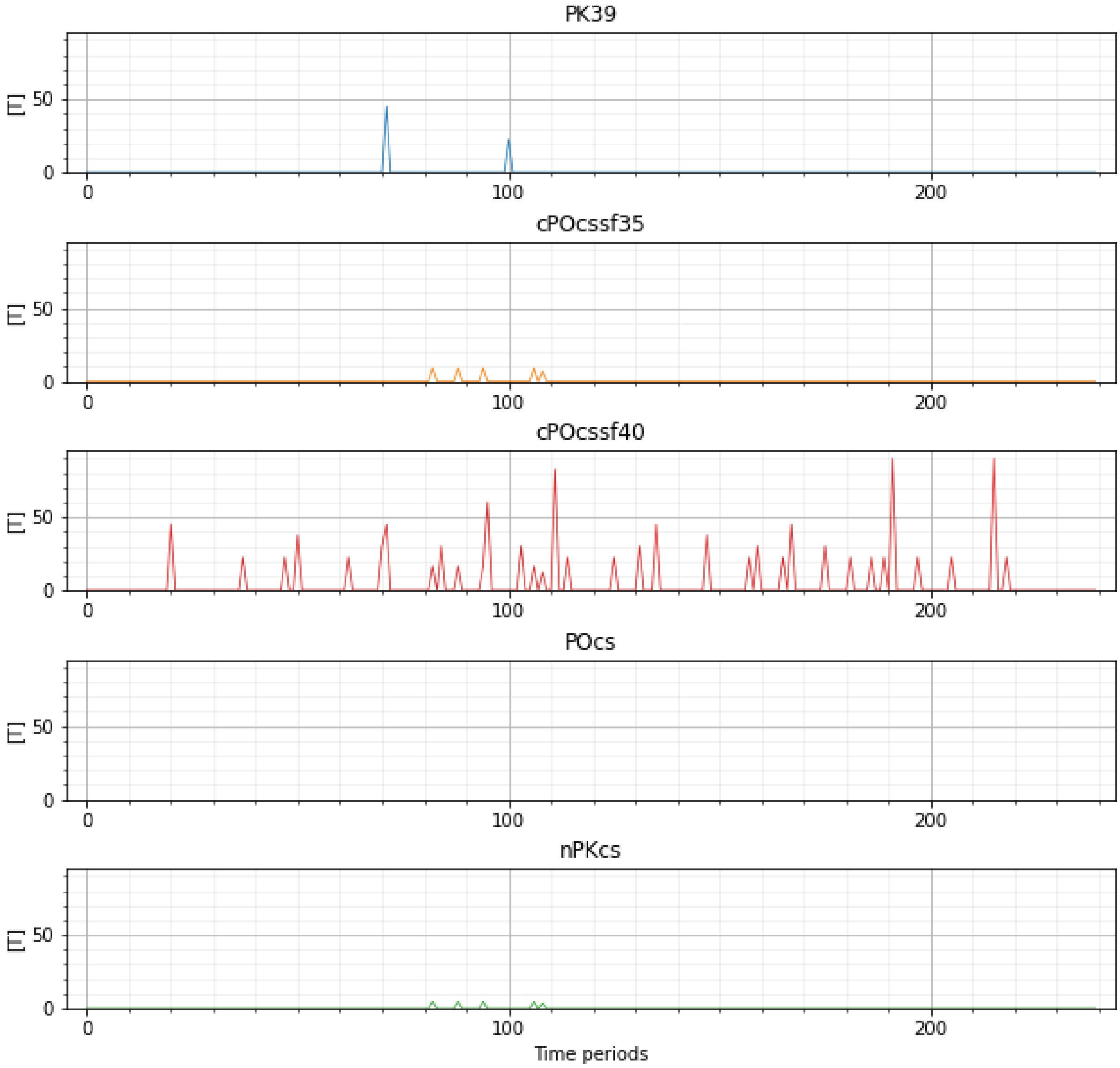

Figure D.19: Line graph of the decision variable R_{st} for the scheduling horizon of 240 time steps, the objective given by Equation 5.3, with a limitation on shelf-life and a cleaning time τ_{jkl} that equals half of the scheduling horizon.

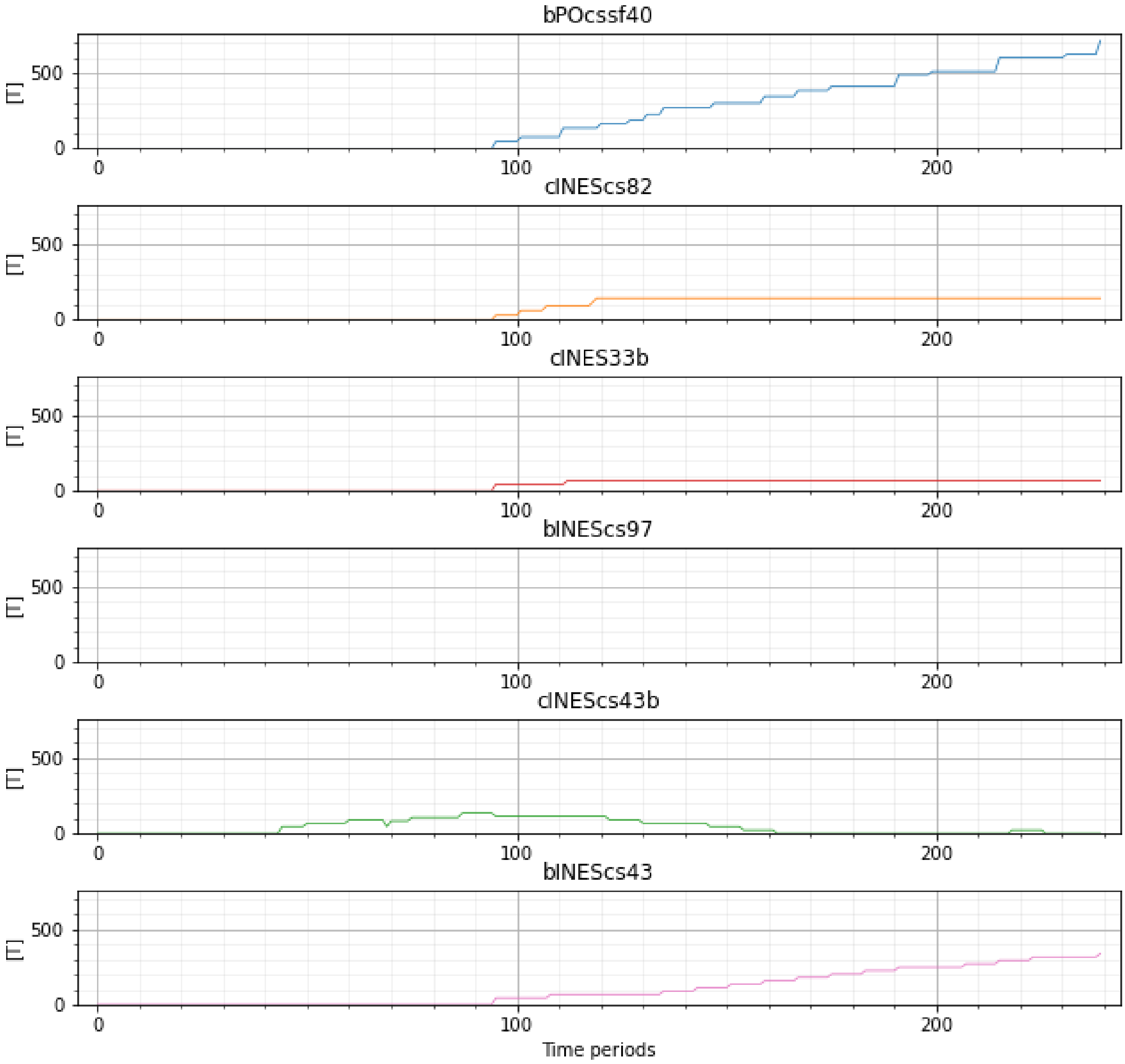

Figure D.20: Line graph of the decision variable S_{st} for the scheduling horizon of 240 time steps, the objective given by Equation 5.3, with a limitation on shelf-life and a cleaning time τ_{jkl} that equals half of the scheduling horizon.

Objective 5.3: Sensitivity analysis on processing order

With limitation on shelf-life - cleaning time equals the whole scheduling horizon

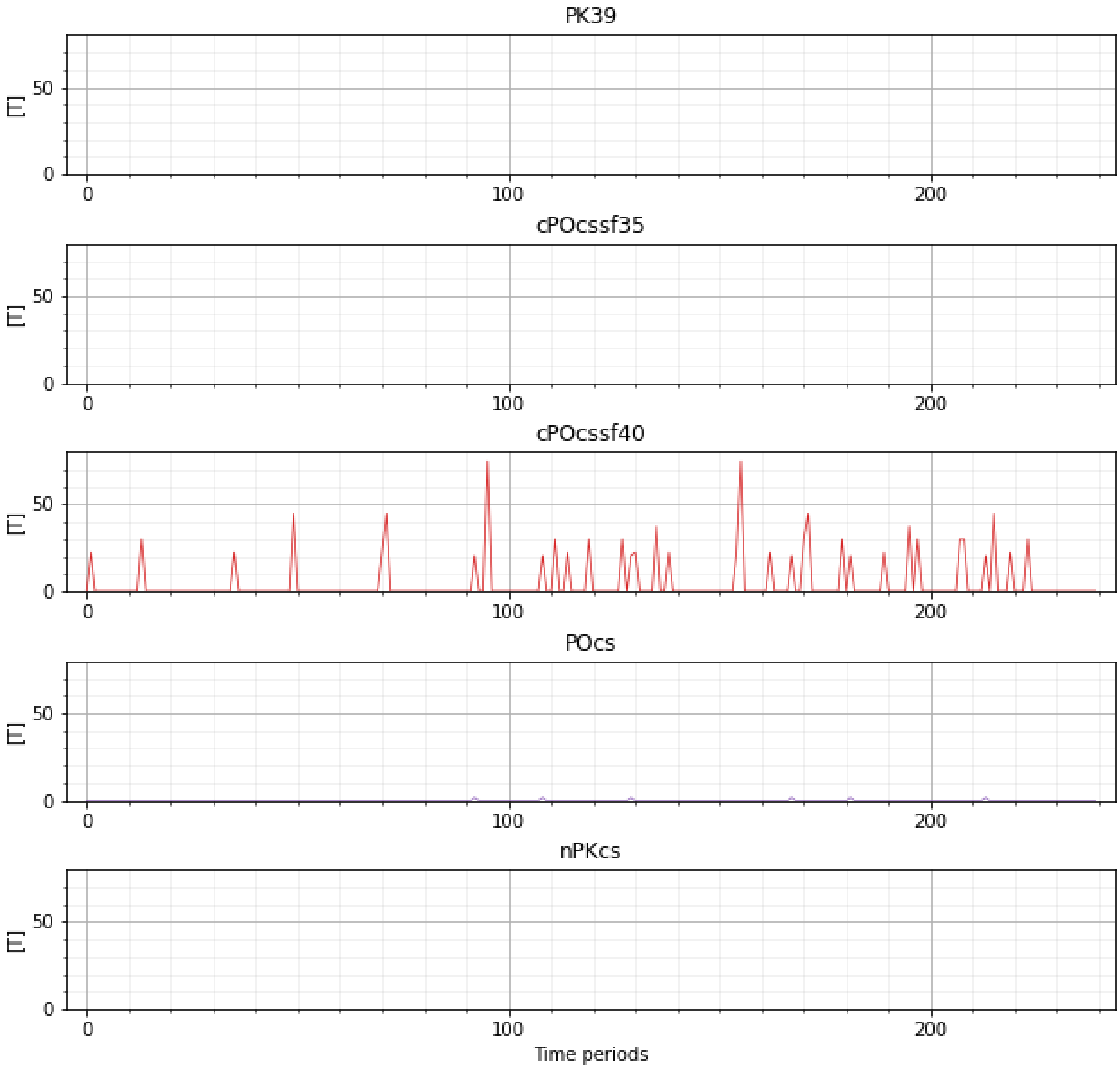

Figure D.21: Line graph of the decision variable R_{st} for the scheduling horizon of 240 time steps, the objective given by Equation 5.3, with a limitation on shelf-life and a cleaning time τ_{jkl} that equals the whole scheduling horizon.

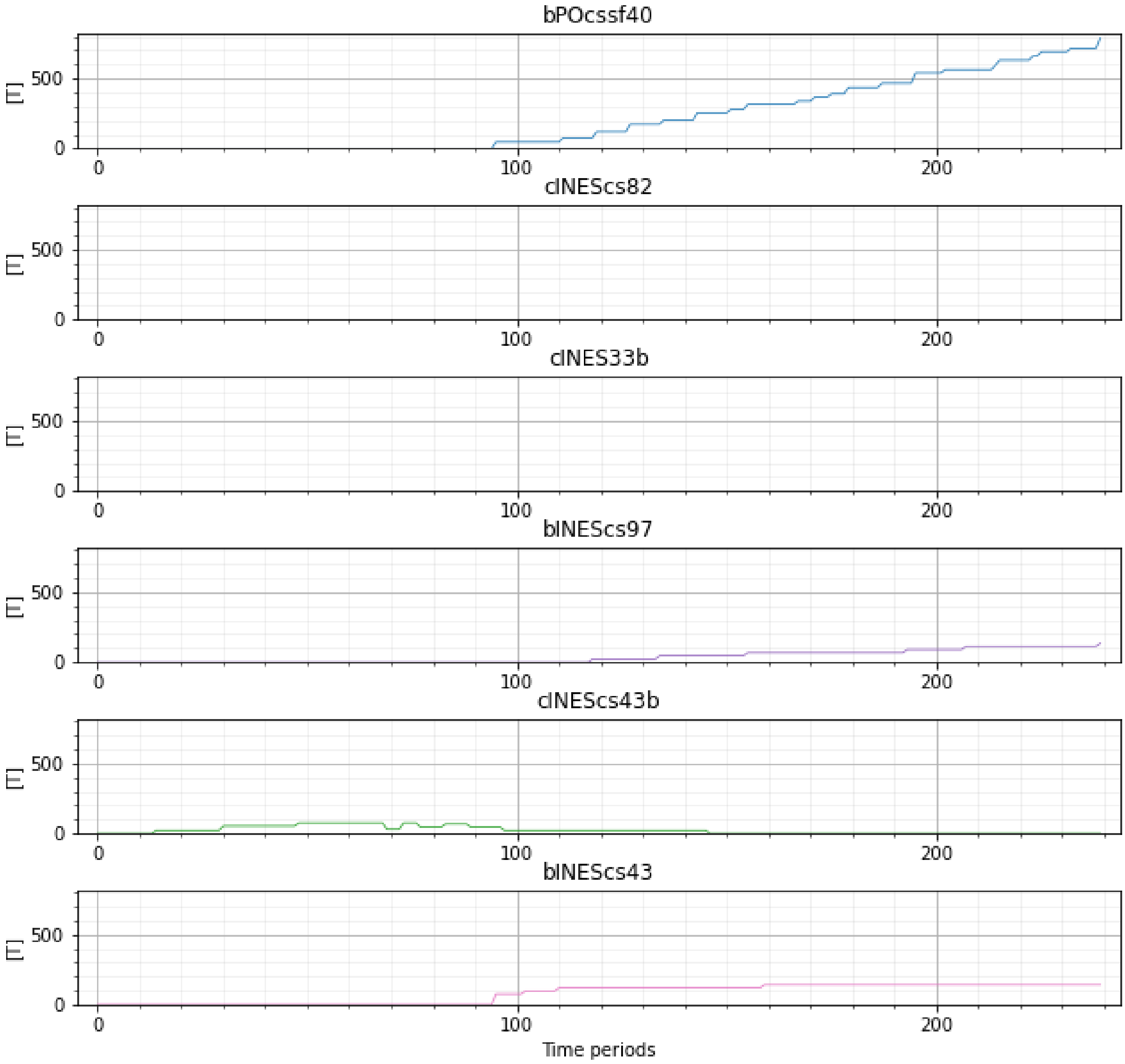

Figure D.22: Line graph of the decision variable S_{st} for the scheduling horizon of 240 time steps, the objective given by Equation 5.3, with a limitation on shelf-life and a cleaning time τ_{jkl} that equals the whole scheduling horizon.

Objective 5.3: Sensitivity analysis on processing order

Without limitation on shelf-life - cleaning time equals half of the scheduling horizon

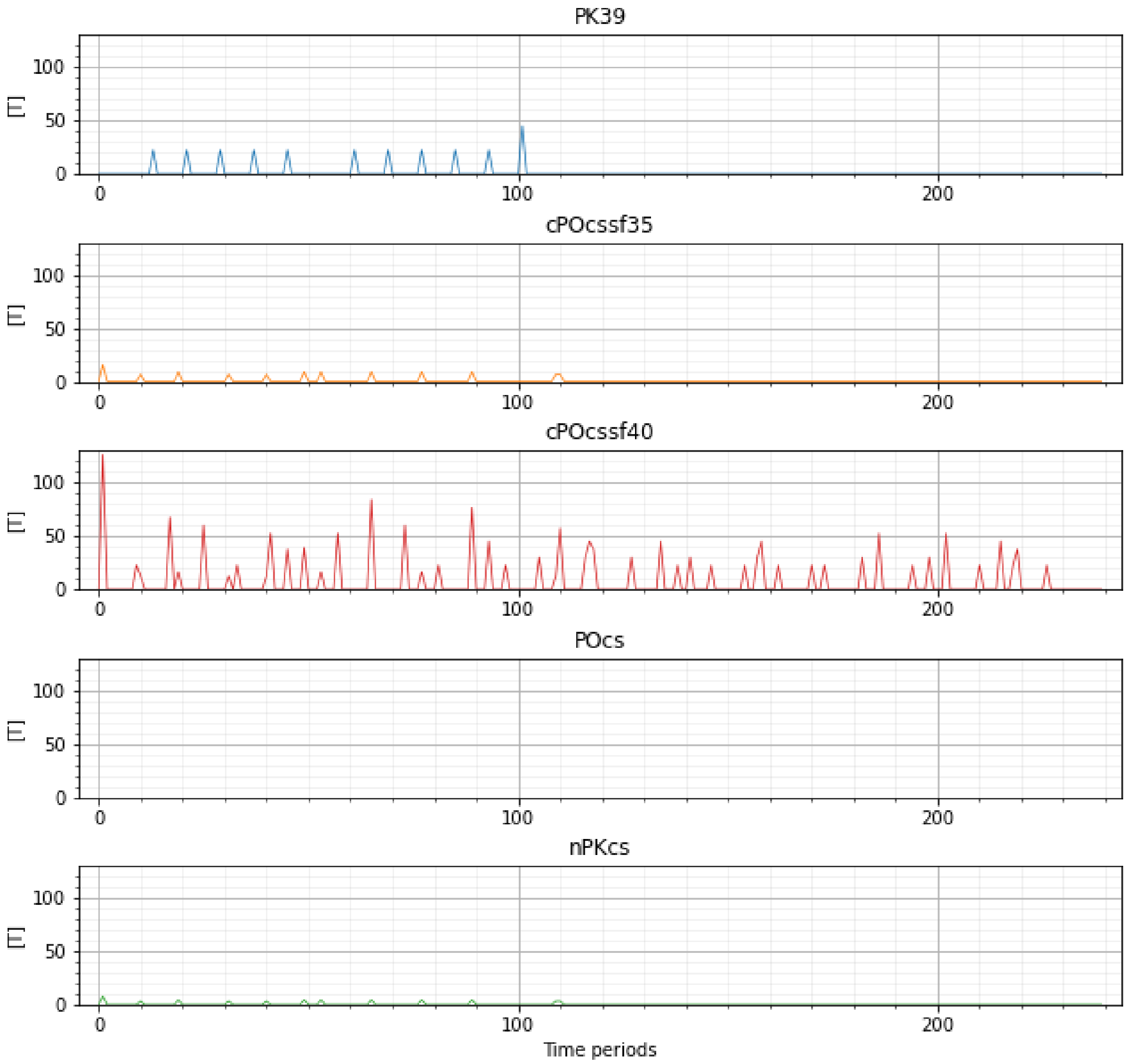

Figure D.23: Line graph of the decision variable R_{st} for the scheduling horizon of 240 time steps, the objective given by Equation 5.3, without a limitation on shelf-life and a cleaning time τ_{jkl} that equals half of the scheduling horizon.

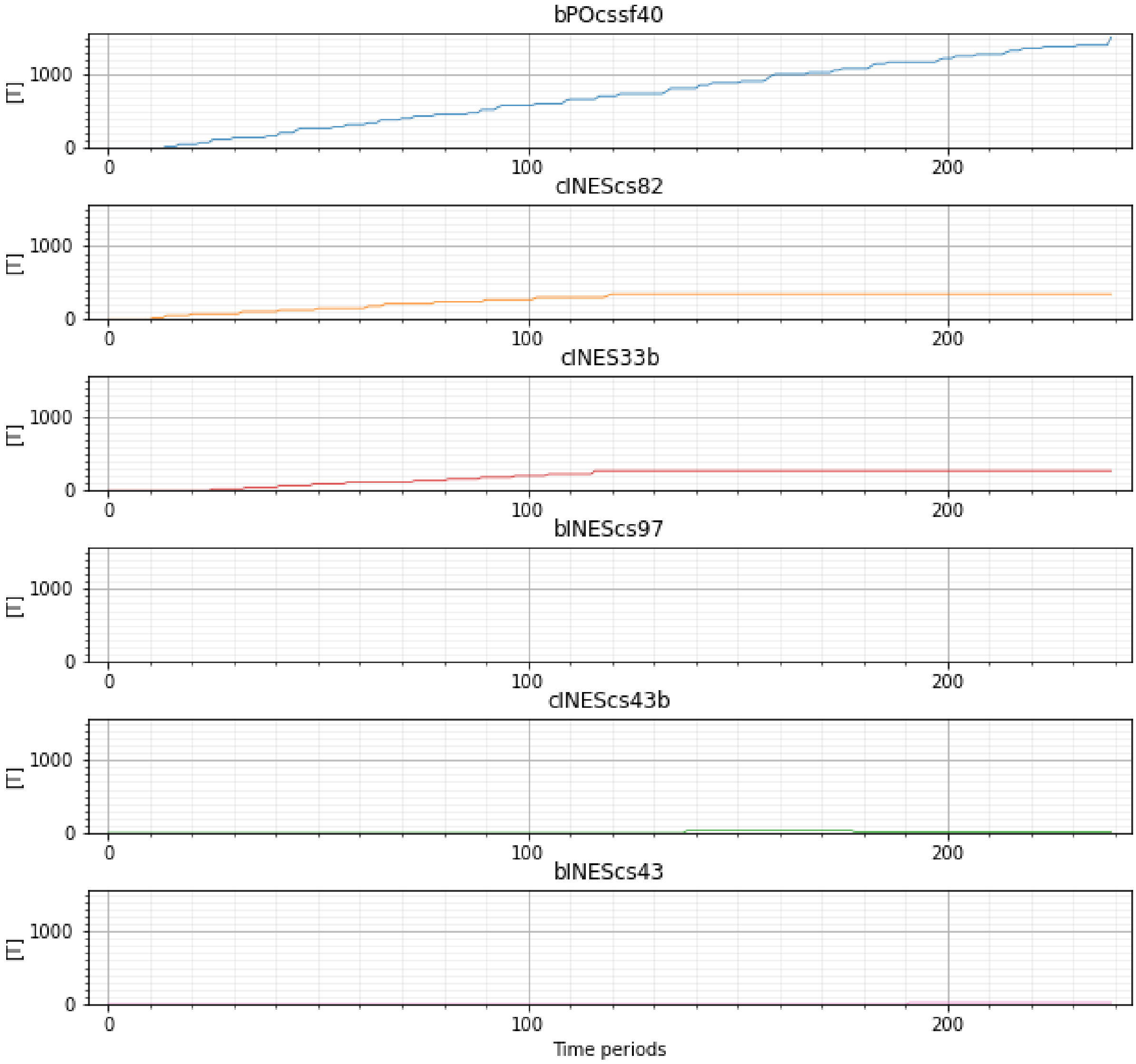

Figure D.24: Line graph of the decision variable S_{st} for the scheduling horizon of 240 time steps, the objective given by Equation 5.3, without a limitation on shelf-life and a cleaning time τ_{jkl} that equals half of the scheduling horizon.

Objective 5.3: Sensitivity analysis on processing order

Without limitation on shelf-life - cleaning time equals the whole scheduling horizon

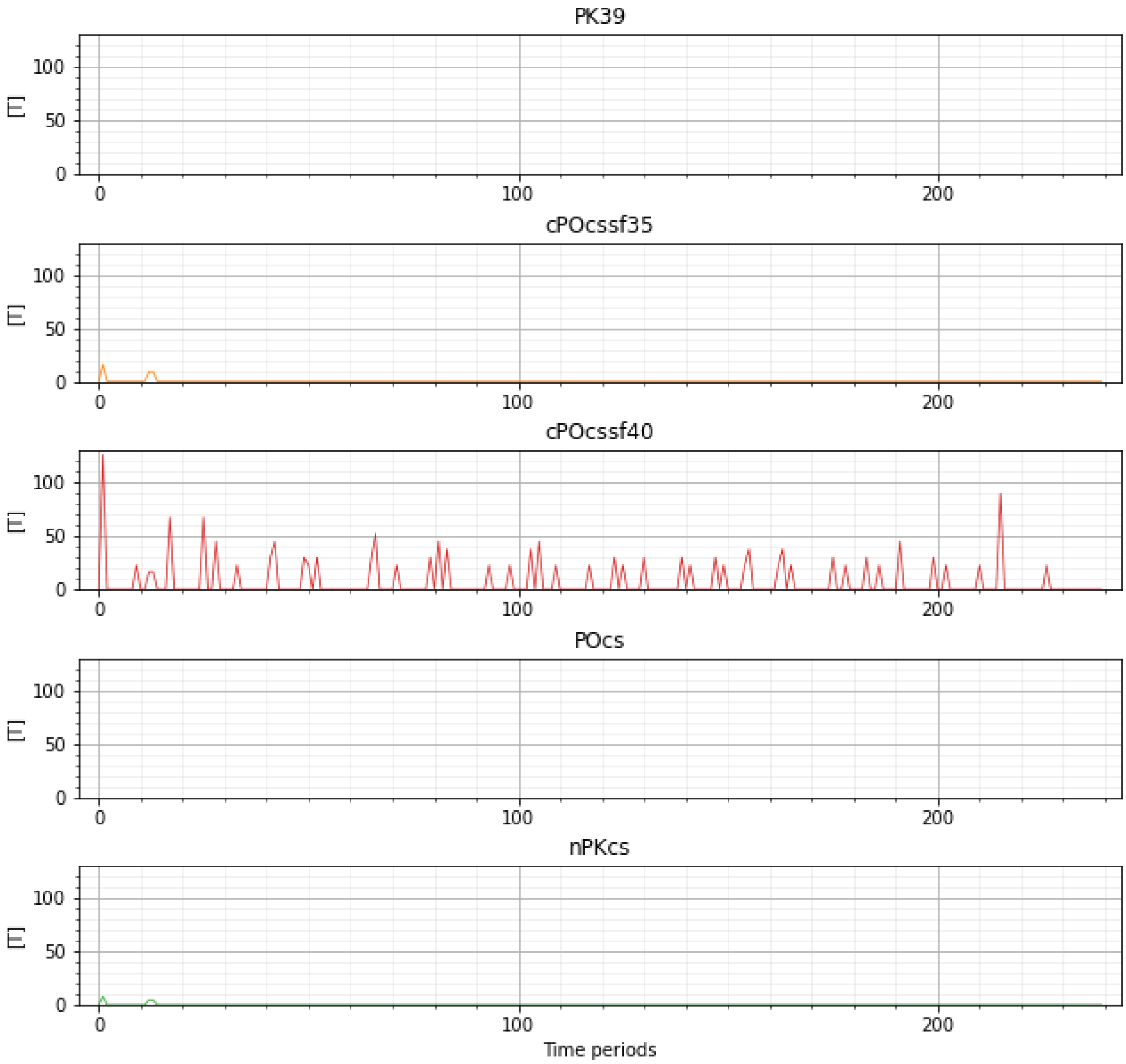

Figure D.25: Line graph of the decision variable R_{st} for the scheduling horizon of 240 time steps, the objective given by Equation 5.3, without a limitation on shelf-life and a cleaning time τ_{jkl} that equals the whole scheduling horizon.

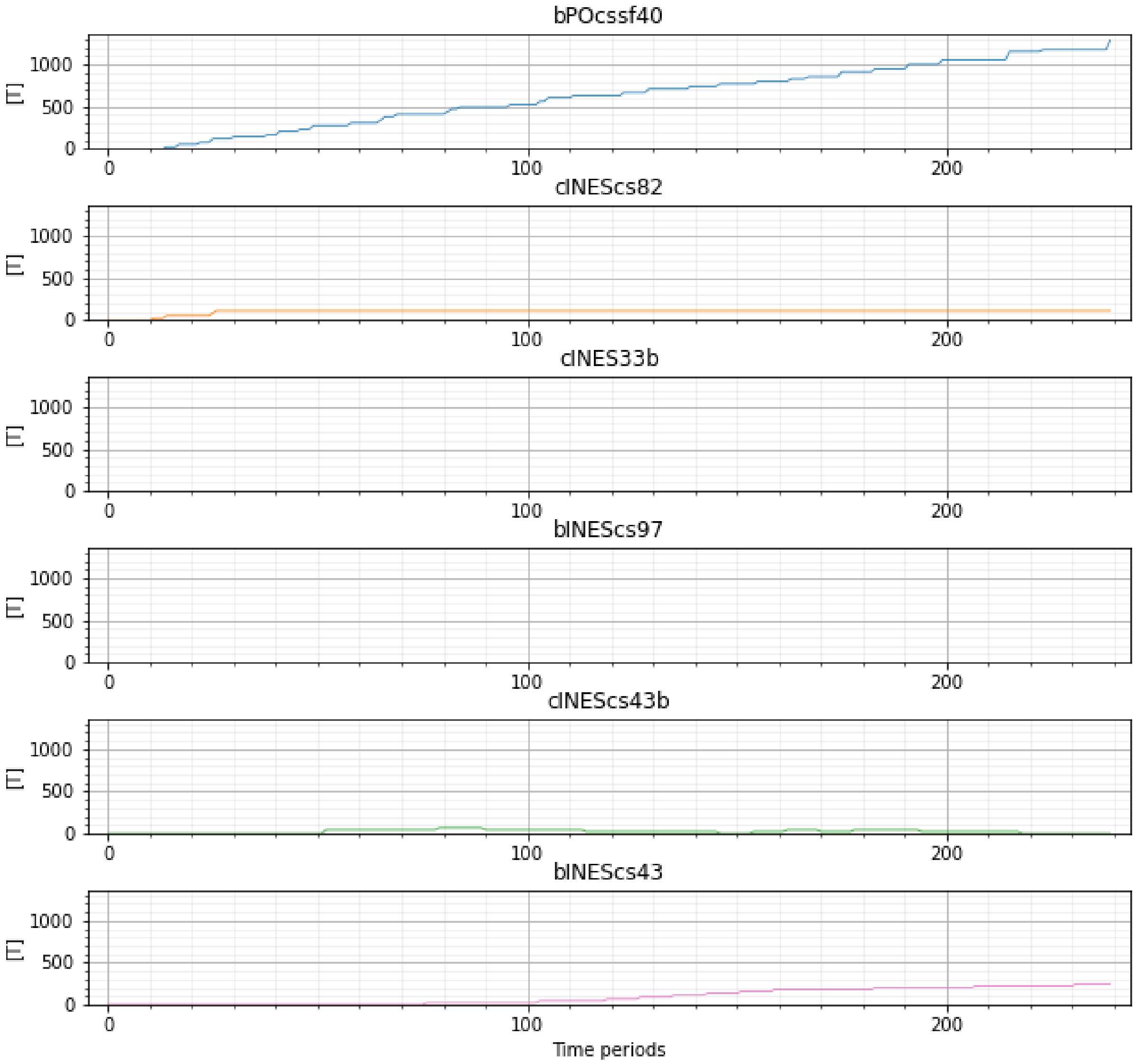

Figure D.26: Line graph of the decision variable S_{st} for the scheduling horizon of 240 time steps, the objective given by Equation 5.3, without a limitation on shelf-life and a cleaning time τ_{jkl} that equals the whole scheduling horizon.

Objective 4.1: Sensitivity analysis on batch-size

Batch-size: 7.5 T, $t < 96^{th}$ time step

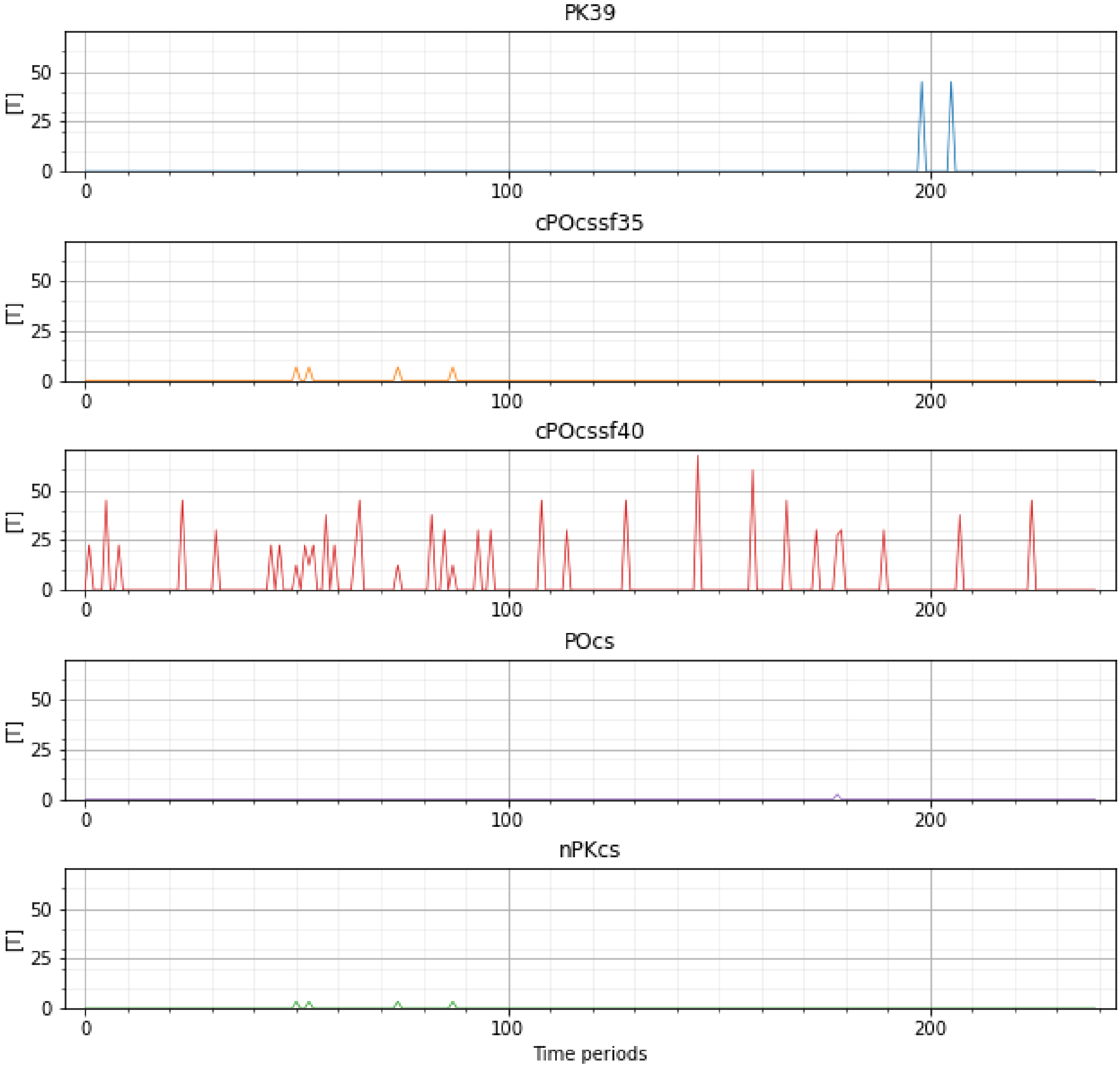

Figure D.27: Line graph of the decision variable R_{st} for the scheduling horizon of 240 time steps, the objective given by Equation 4.1 and an implemented batch-size of 7.5 T at $t < 96^{th}$ time step.

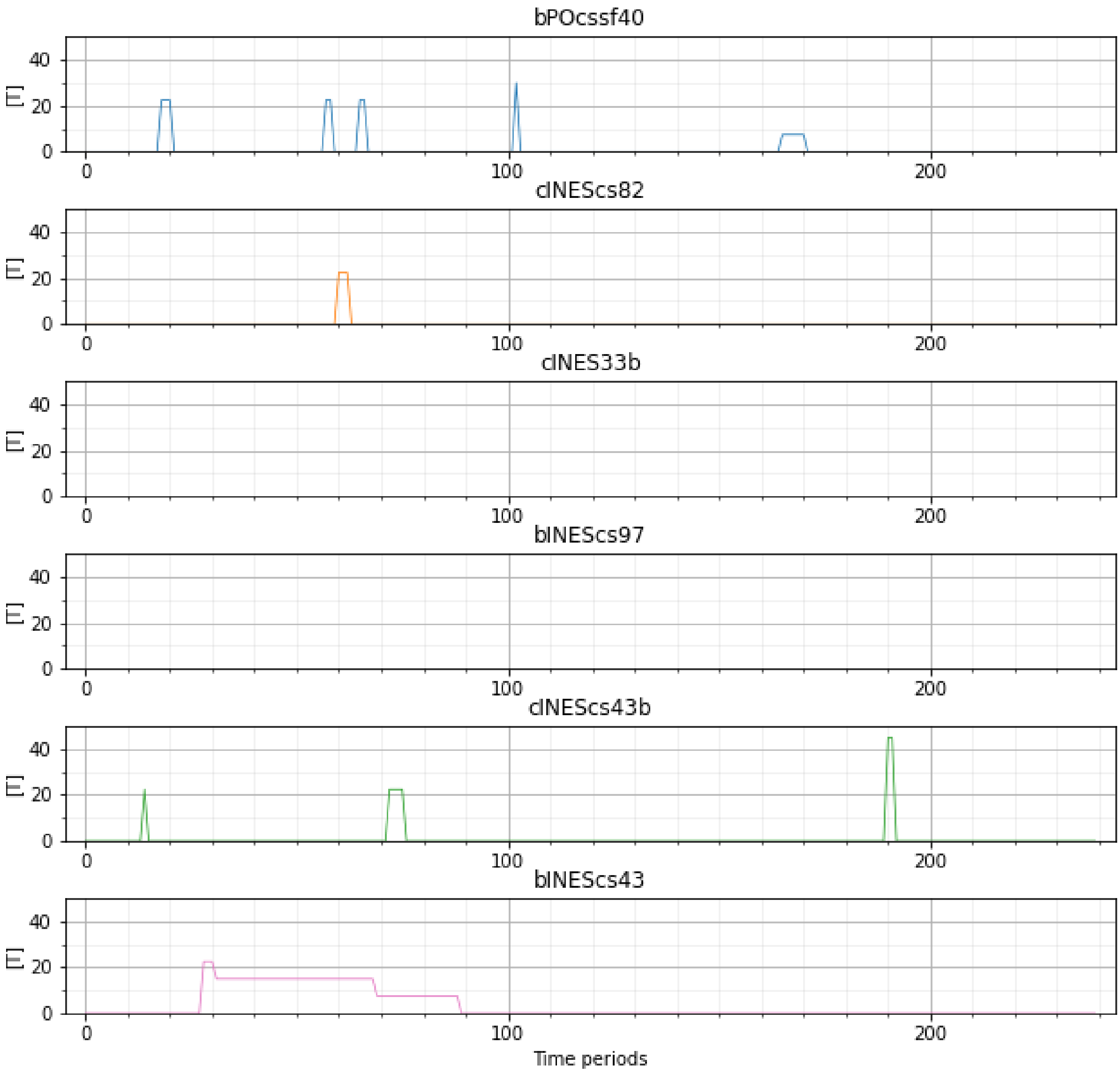

Figure D.28: Line graph of the decision variable S_{st} for the scheduling horizon of 240 time steps, the objective given by Equation 4.1 and an implemented batch-size of 7.5 T at $t < 96^{th}$ time step.

Objective 4.1: Sensitivity analysis on batch-size

Batch-size: 15 T, $t < 96^{th}$ time step

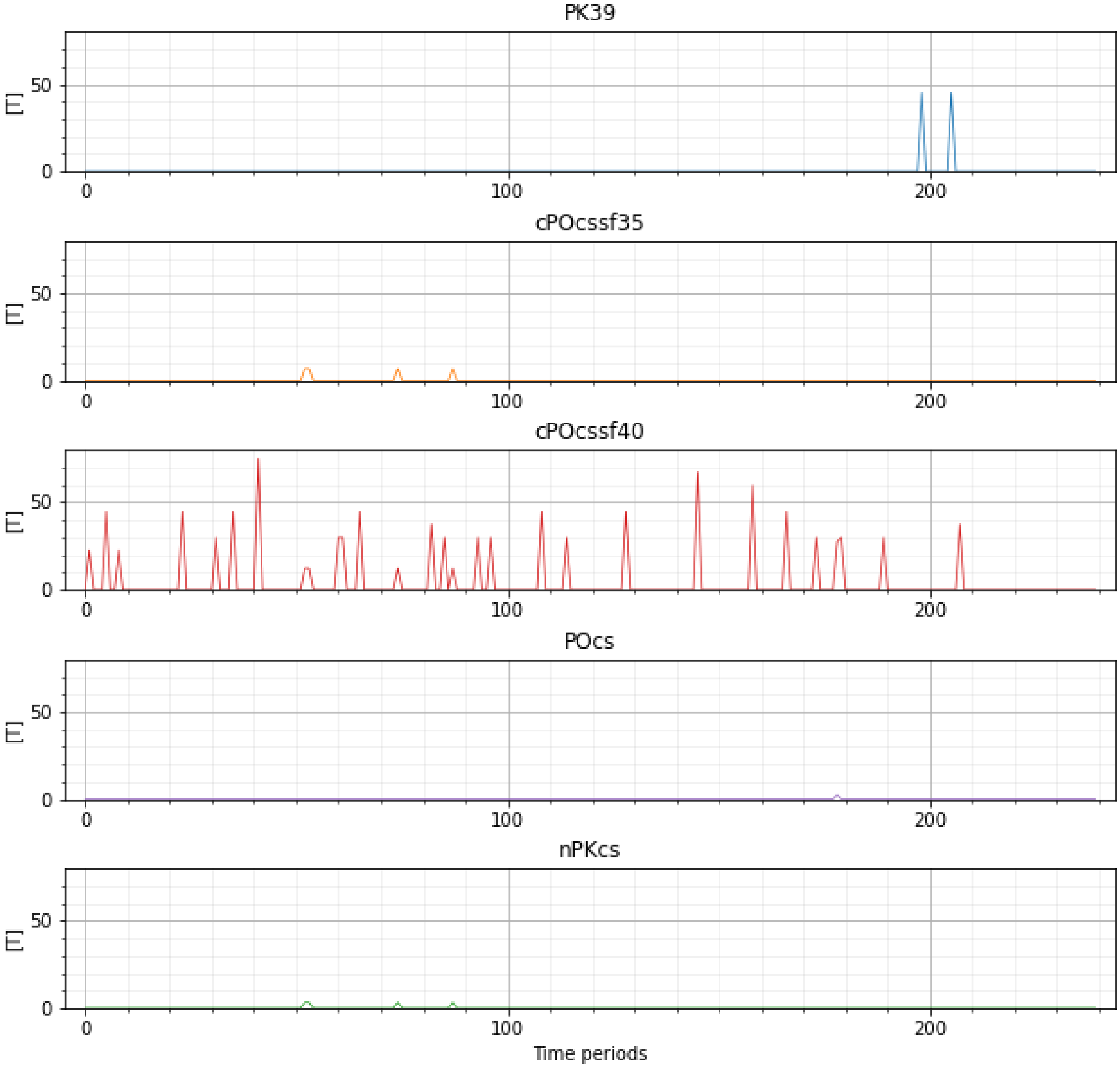

Figure D.29: Line graph of the decision variable R_{st} for the scheduling horizon of 240 time steps, the objective given by Equation 4.1 and an implemented batch-size of 15 T at $t < 96^{th}$ time step.

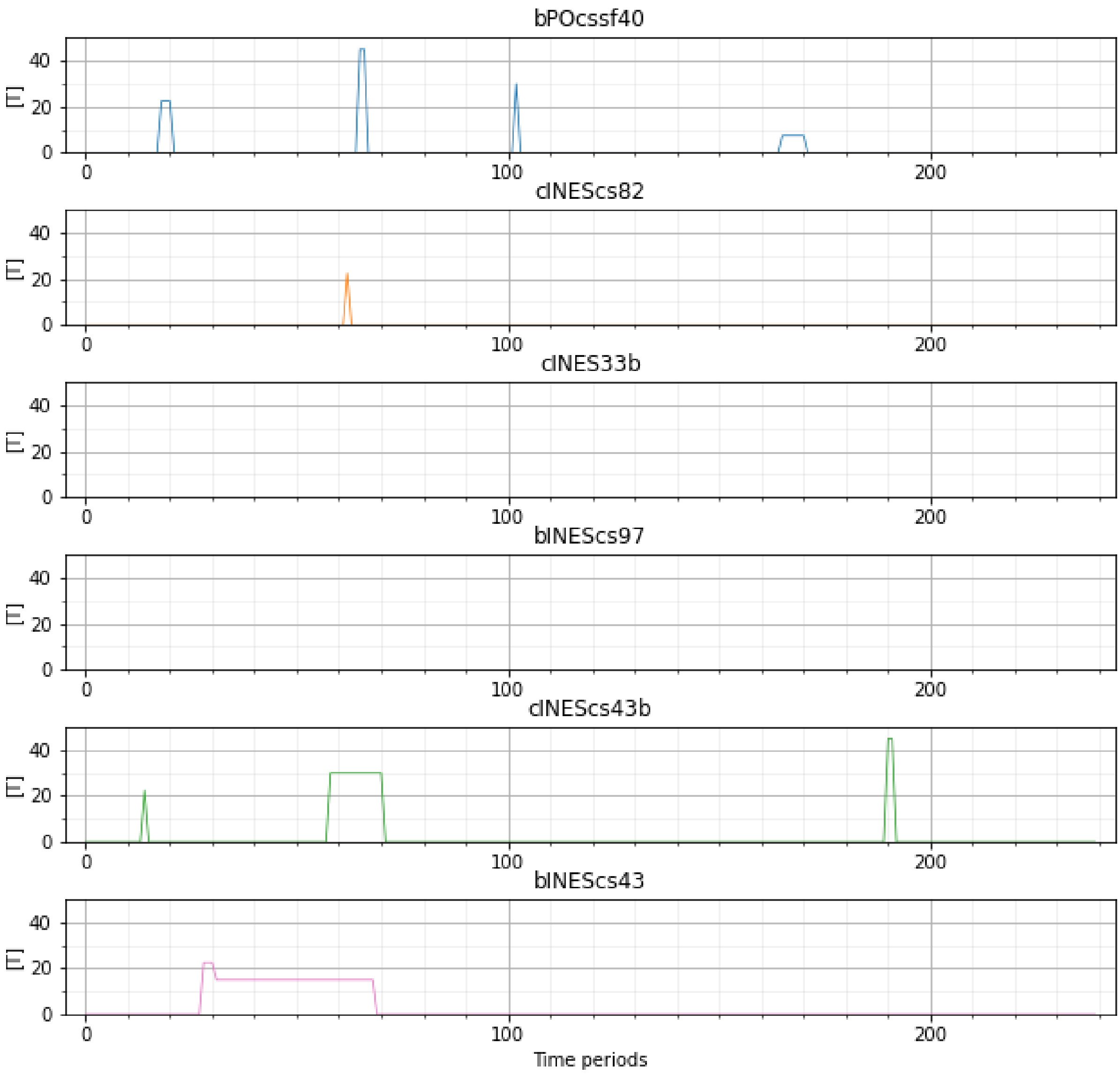

Figure D.30: Line graph of the decision variable S_{st} for the scheduling horizon of 240 time steps, the objective given by Equation 4.1 and an implemented batch-size of 15 T at $t < 96^{th}$ time step.

Objective 4.1: Sensitivity analysis on batch-size

Batch-size: 7.5 T, $t > 96^{th}$ time step

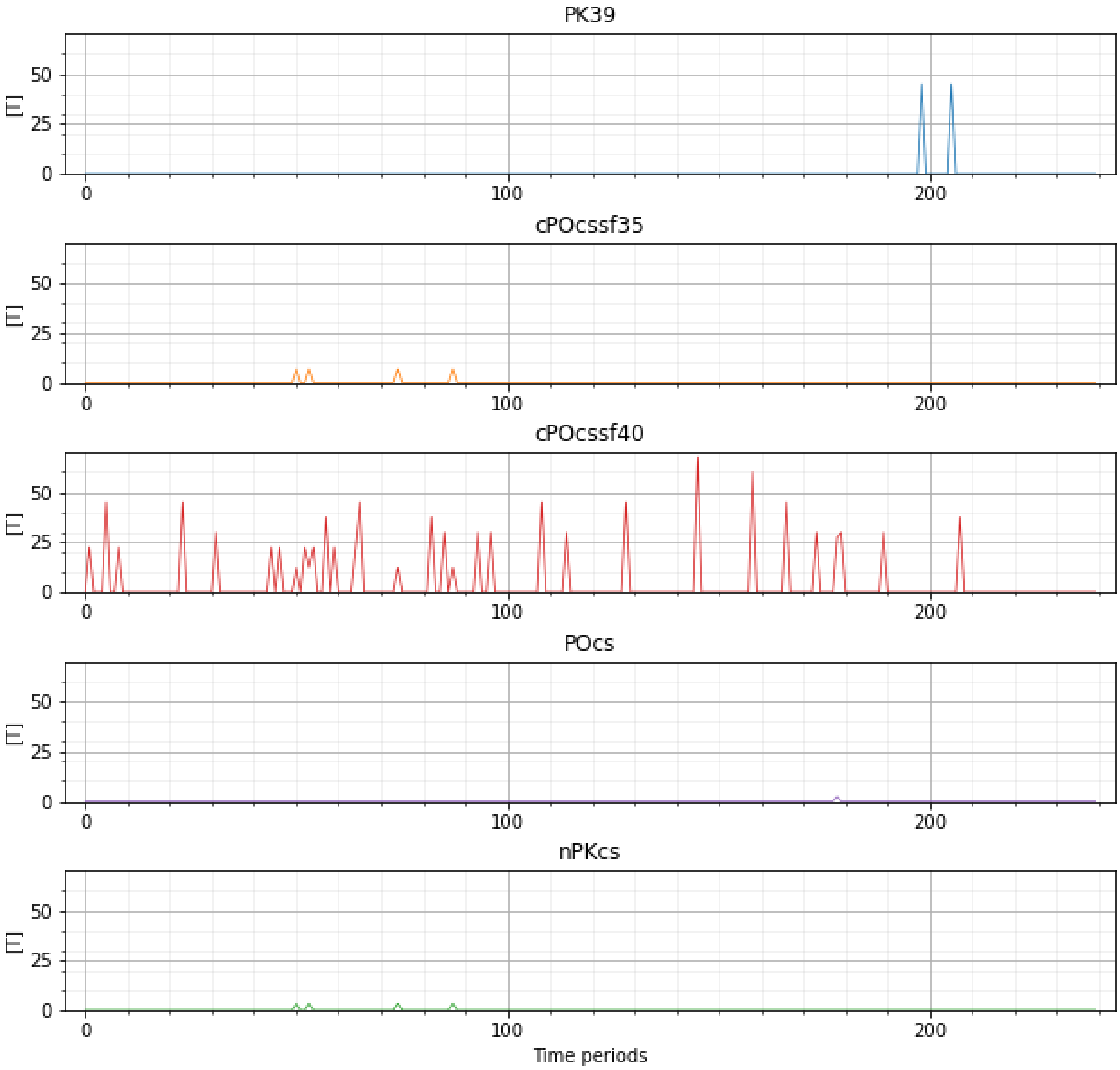

Figure D.31: Line graph of the decision variable R_{st} for the scheduling horizon of 240 time steps, the objective given by Equation 4.1 and an implemented batch-size of 7.5 T at $t > 96^{th}$ time step.

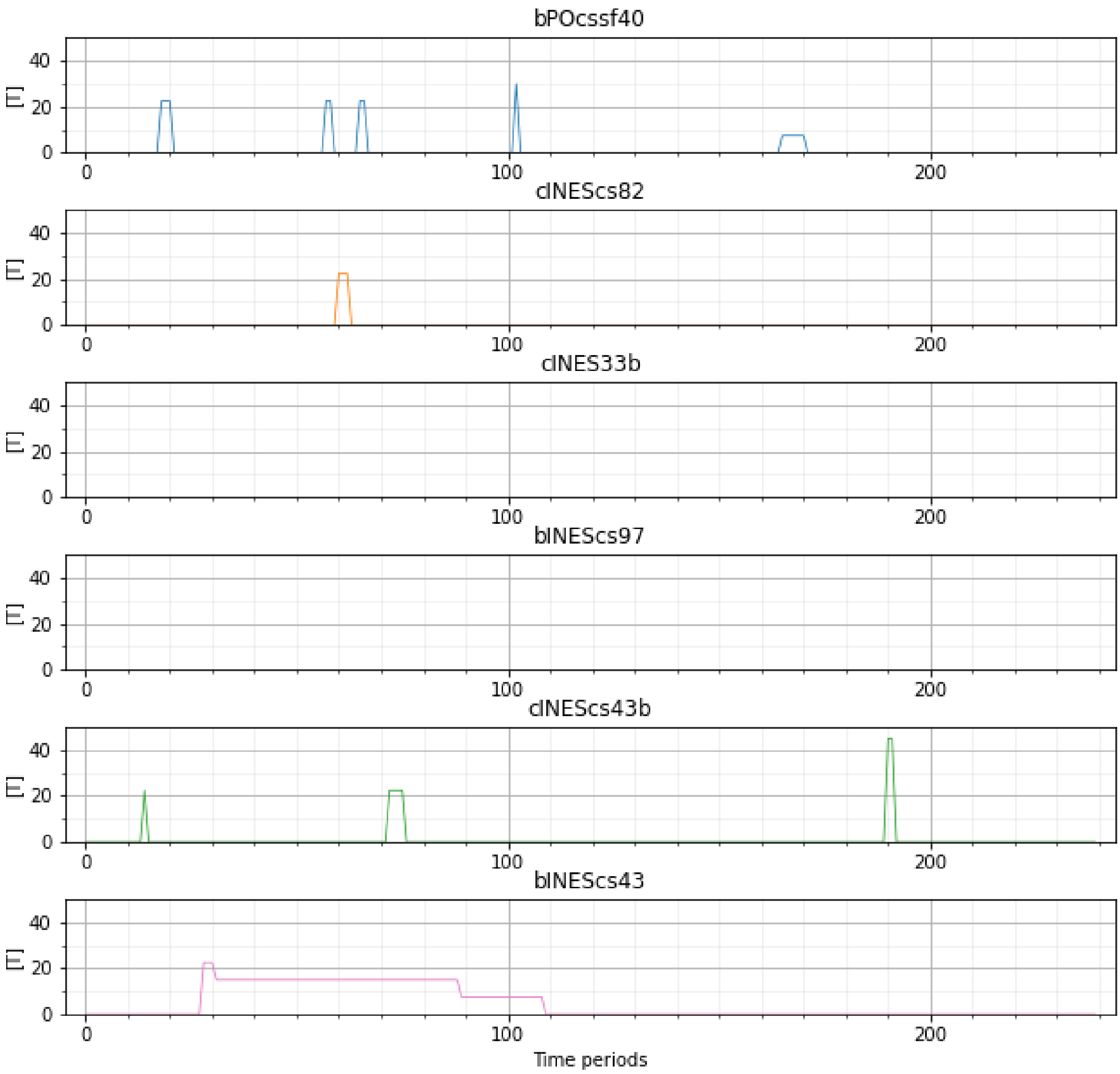

Figure D.32: Line graph of the decision variable S_{st} for the scheduling horizon of 240 time steps, the objective given by Equation 4.1 and an implemented batch-size of 7.5 T at $t > 96^{th}$ time step.

Objective 4.1: Sensitivity analysis on batch-size

Batch-size: 15 T, $t > 96^{th}$ time step

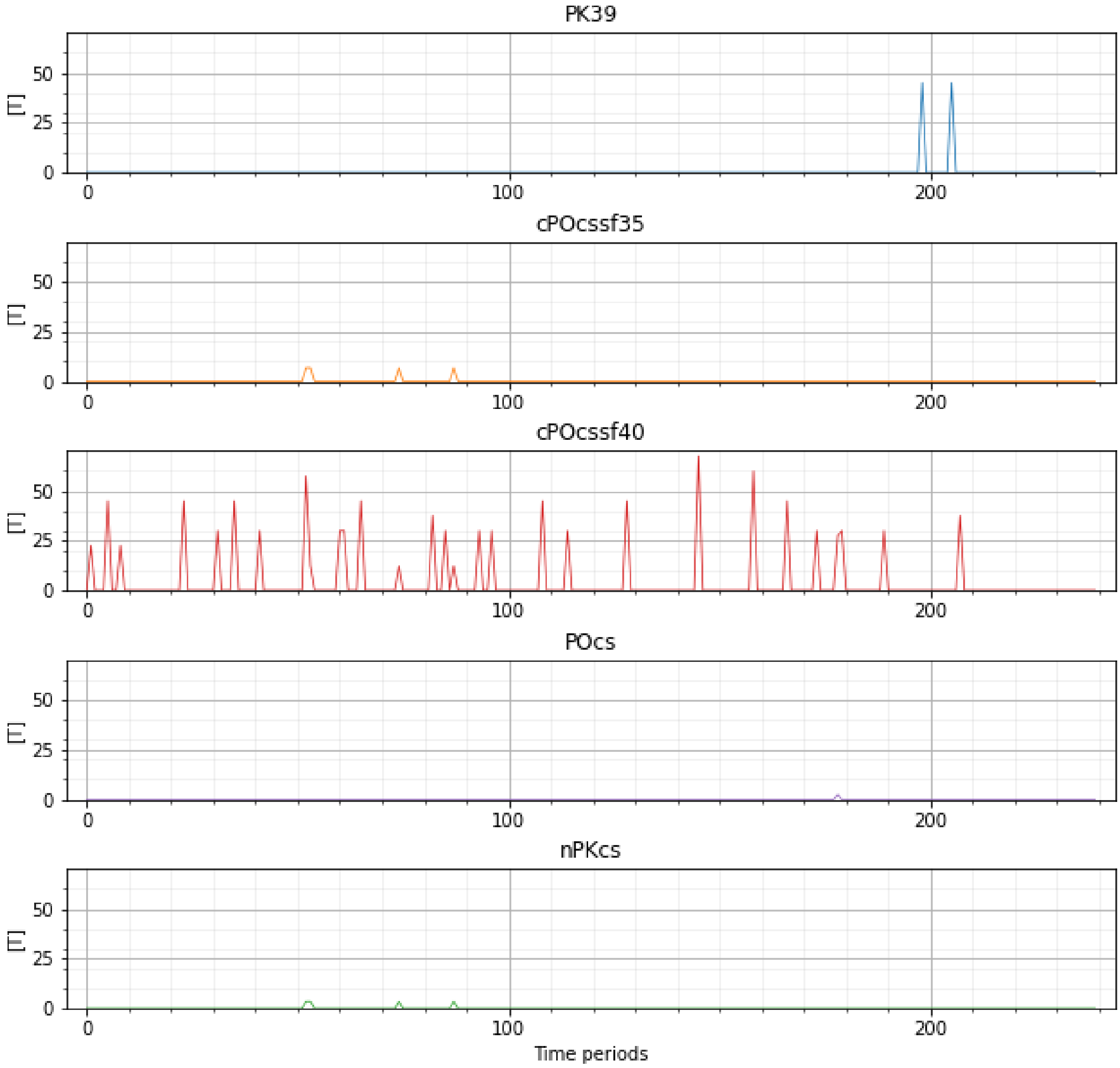

Figure D.33: Line graph of the decision variable R_{st} for the scheduling horizon of 240 time steps, the objective given by Equation 4.1 and an implemented batch-size of 15 T at $t > 96^{th}$ time step.

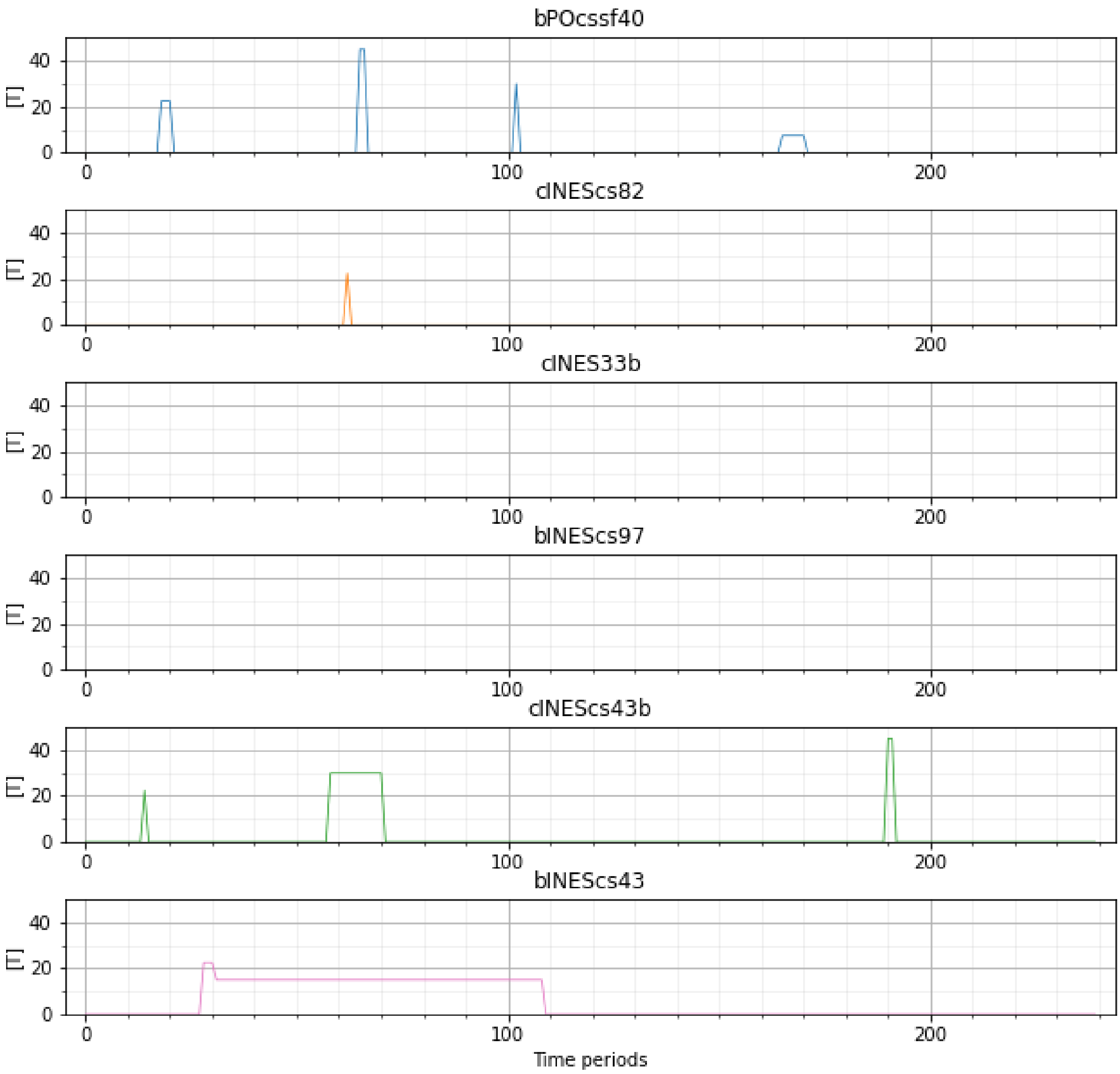

Figure D.34: Line graph of the decision variable S_{st} for the scheduling horizon of 240 time steps, the objective given by Equation 4.1 and an implemented batch-size of 15 T at $t > 96^{th}$ time step.

Objective 5.1: Sensitivity analysis on batch-size

Batch-size: 7.5 T, $t < 96^{th}$ time step

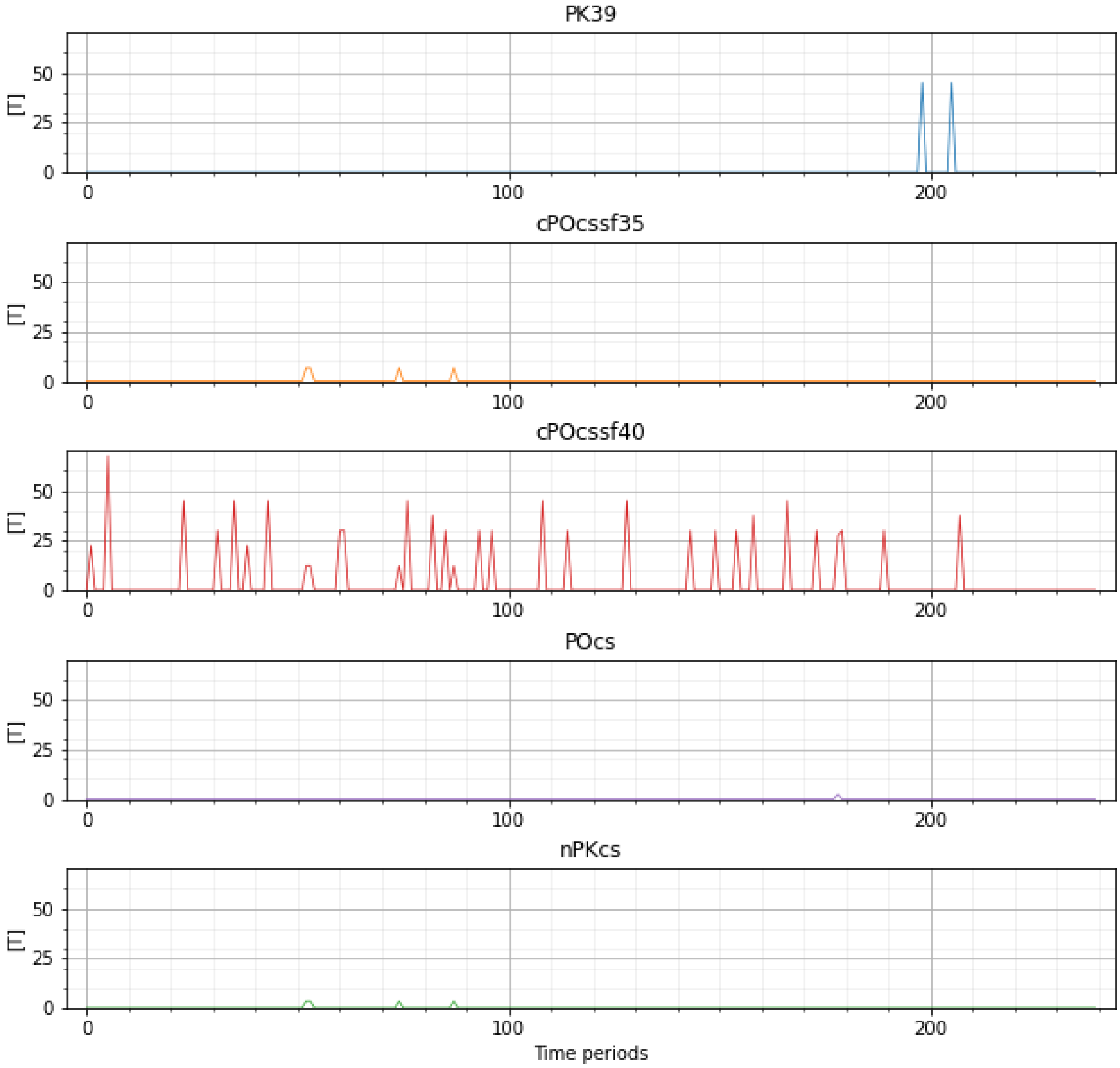

Figure D.35: Line graph of the decision variable R_{st} for the scheduling horizon of 240 time steps, the objective given by Equation 5.1 and an implemented batch-size of 7.5 T at $t < 96^{th}$ time step.

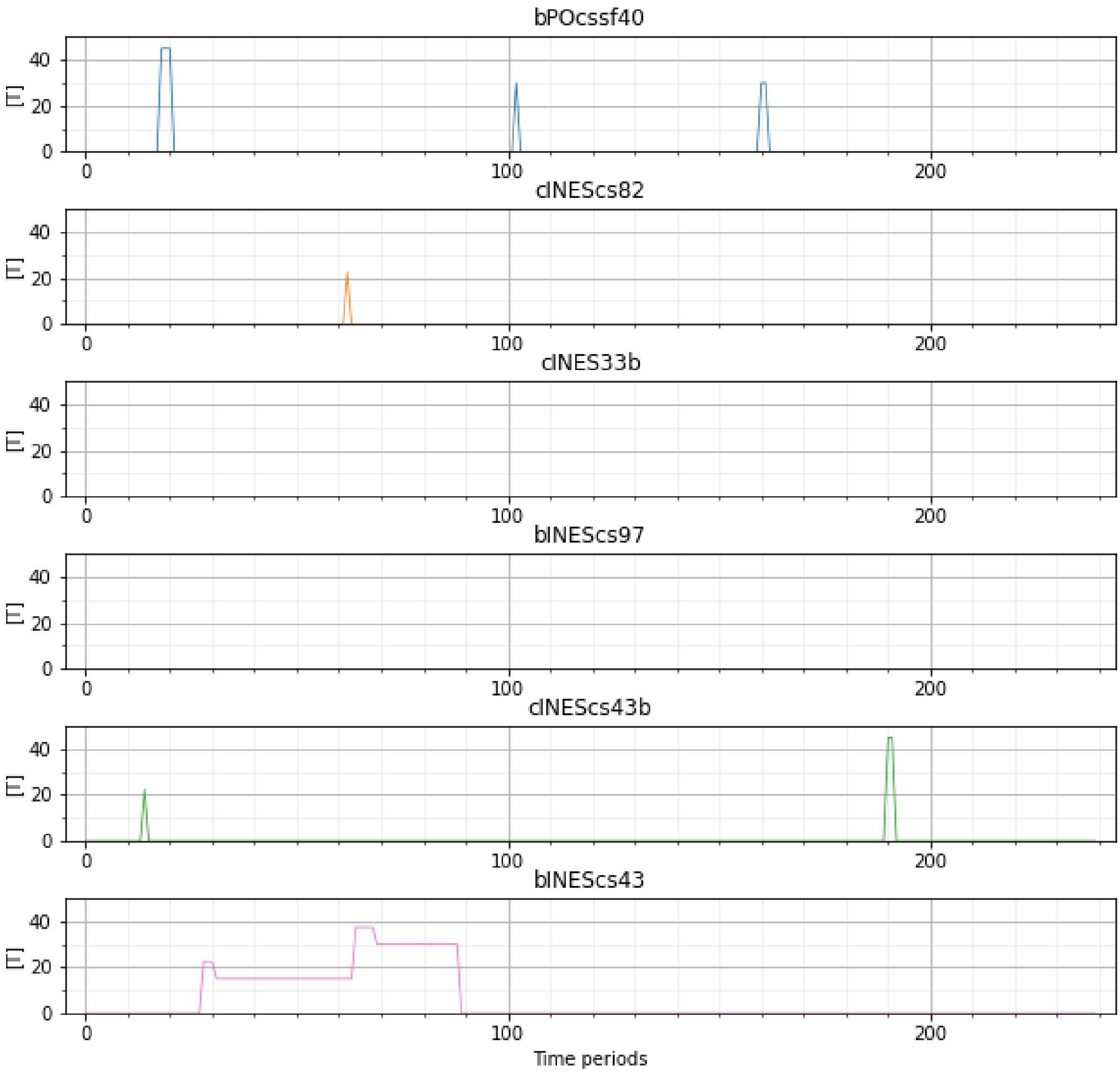

Figure D.36: Line graph of the decision variable S_{st} for the scheduling horizon of 240 time steps, the objective given by Equation 5.1 and an implemented batch-size of 7.5 T at $t < 96^{th}$ time step.

Objective 5.1: Sensitivity analysis on batch-size

Batch-size: 15 T, $t < 96^{th}$ time step

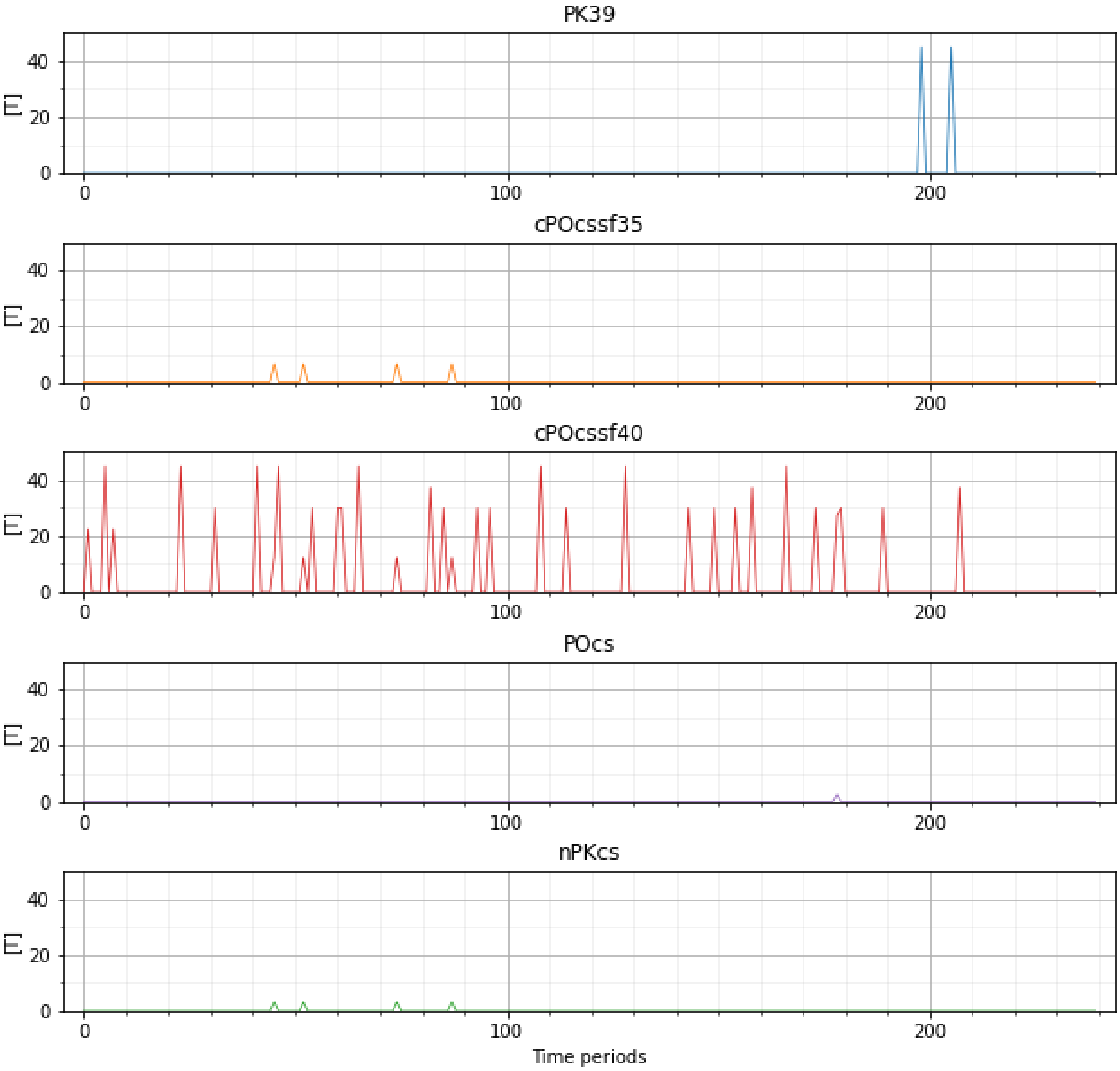

Figure D.37: Line graph of the decision variable R_{st} for the scheduling horizon of 240 time steps, the objective given by Equation 5.1 and an implemented batch-size of 15 T at $t < 96^{th}$ time step.

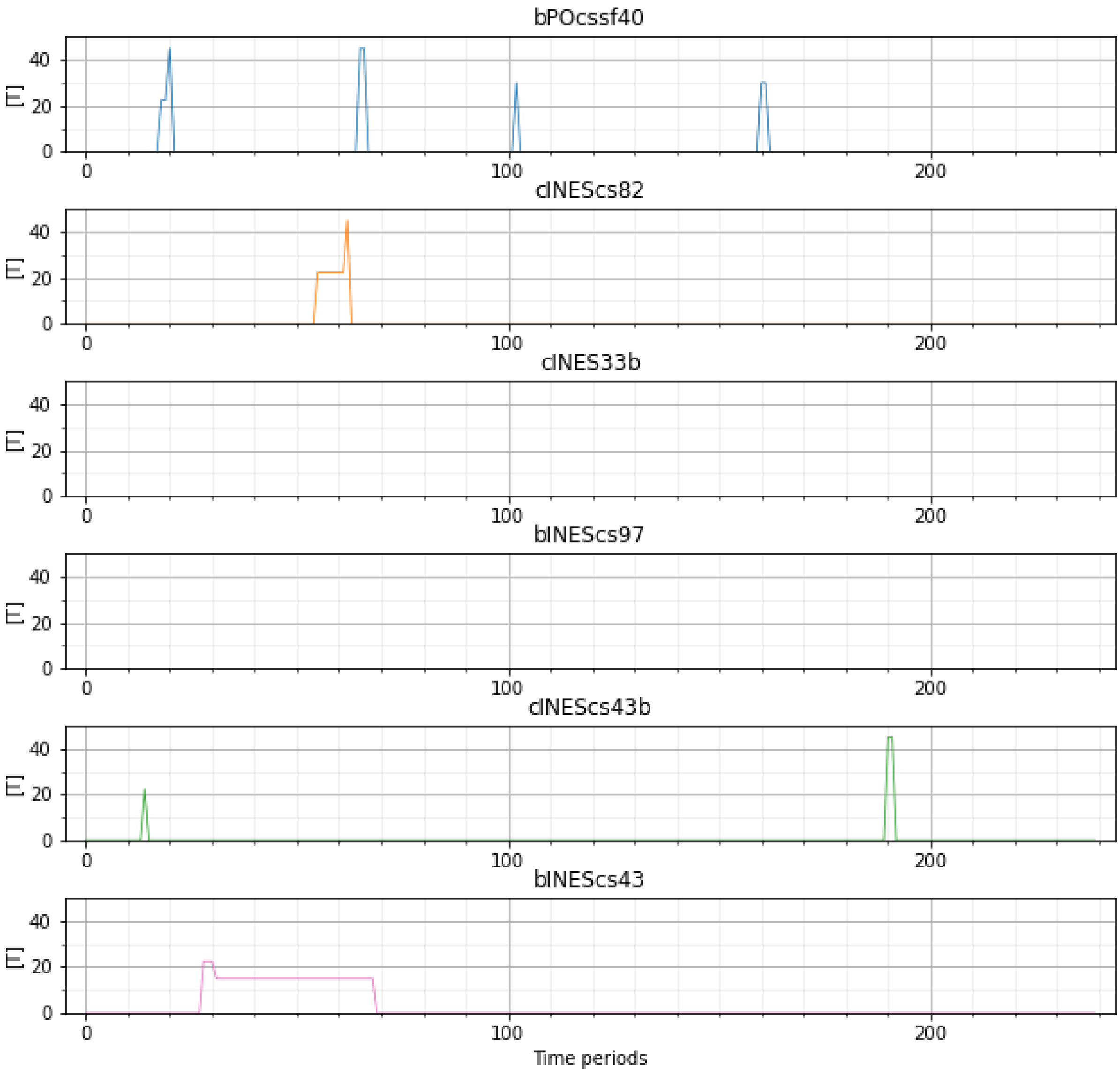

Figure D.38: Line graph of the decision variable S_{st} for the scheduling horizon of 240 time steps, the objective given by Equation 5.1 and an implemented batch-size of 15 T at $t < 96^{th}$ time step.

Objective 5.1: Sensitivity analysis on batch-size

Batch-size: 7.5 T, $t > 96^{th}$ time step

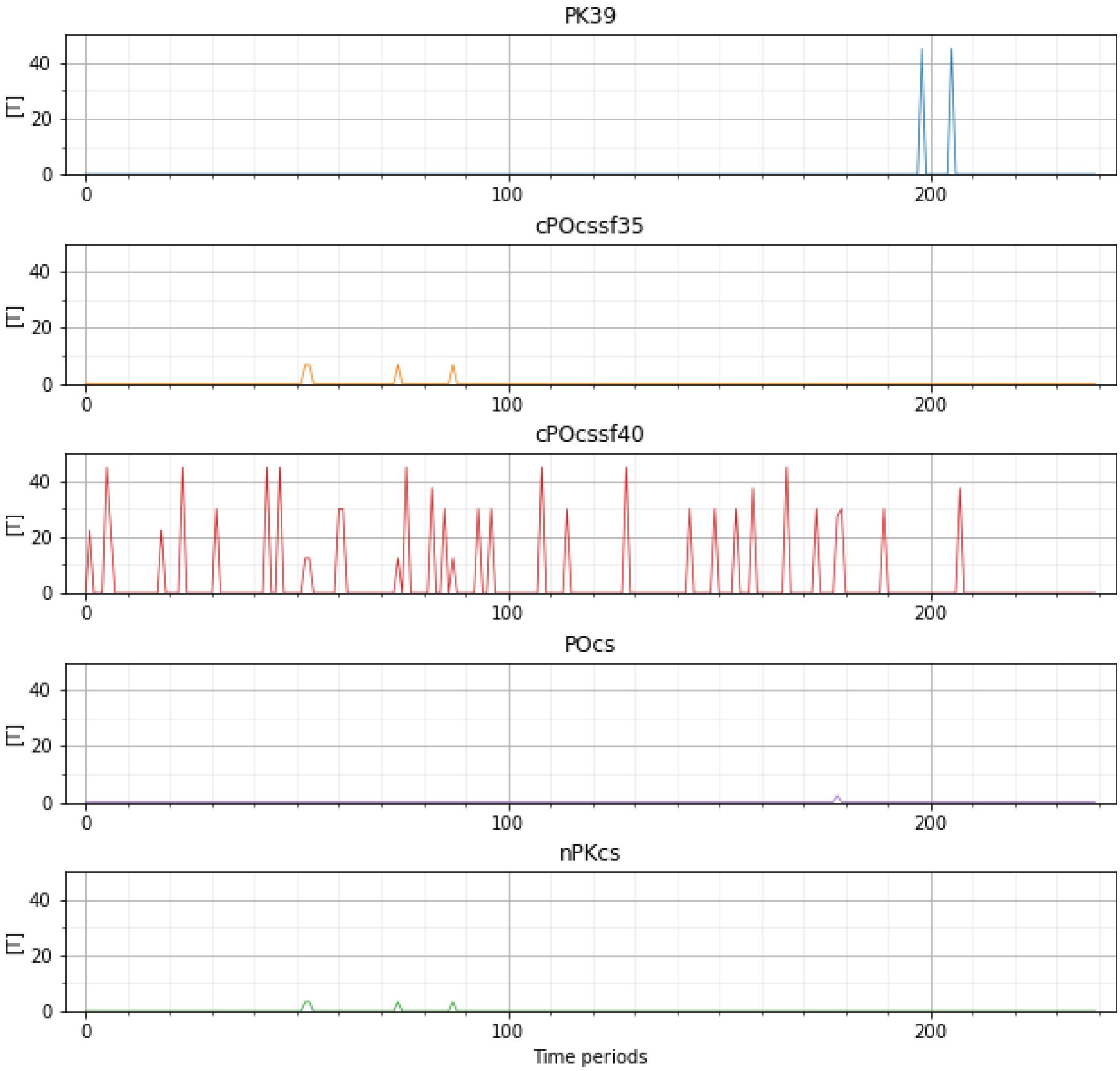

Figure D.39: Line graph of the decision variable R_{st} for the scheduling horizon of 240 time steps, the objective given by Equation 5.1 and an implemented batch-size of 7.5 T at $t > 96^{th}$ time step.

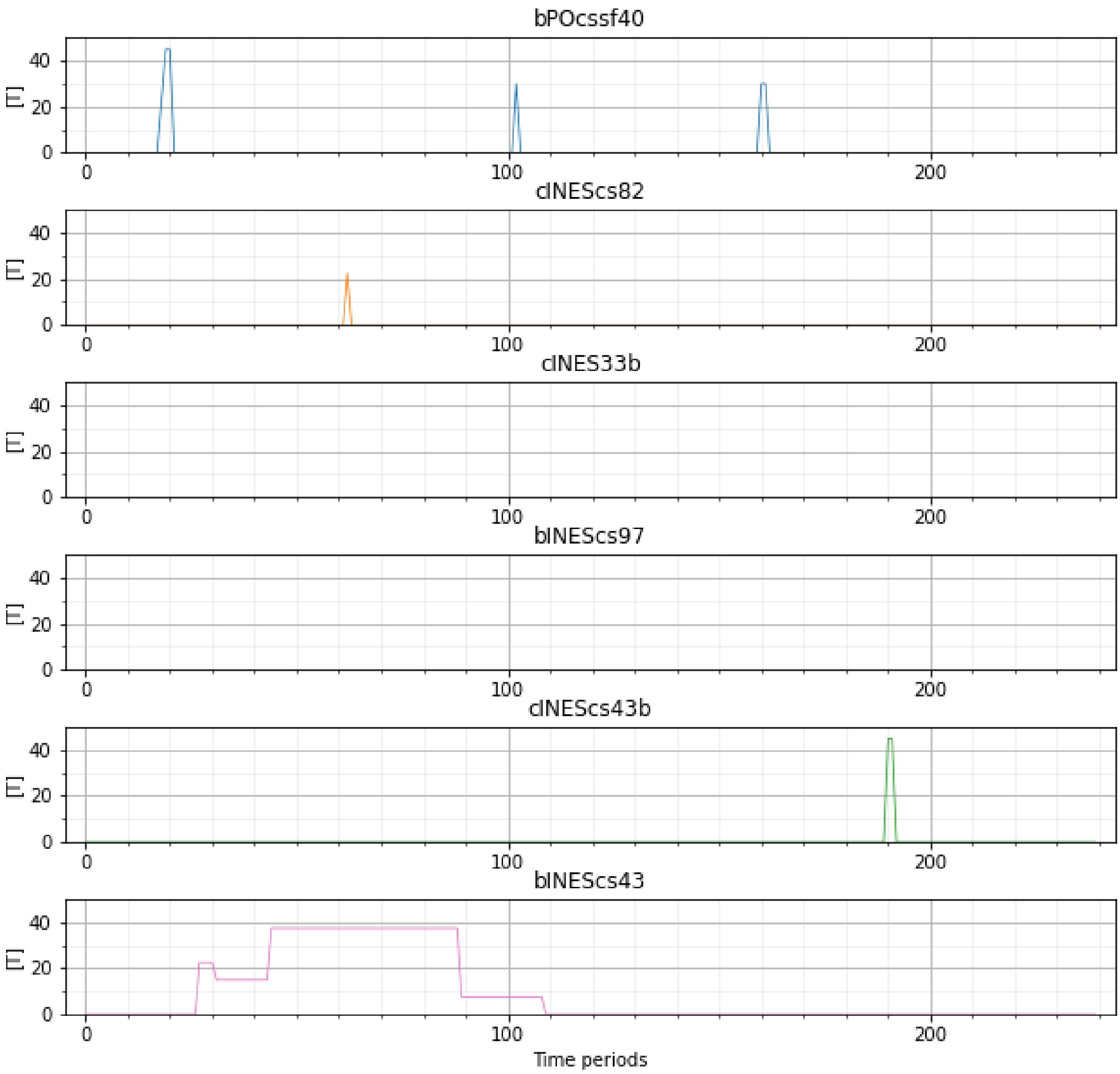

Figure D.40: Line graph of the decision variable S_{st} for the scheduling horizon of 240 time steps, the objective given by Equation 5.1 and an implemented batch-size of 7.5 T at $t > 96^{th}$ time step.

Objective 5.1: Sensitivity analysis on batch-size

Batch-size: 15 T, $t > 96^{th}$ time step

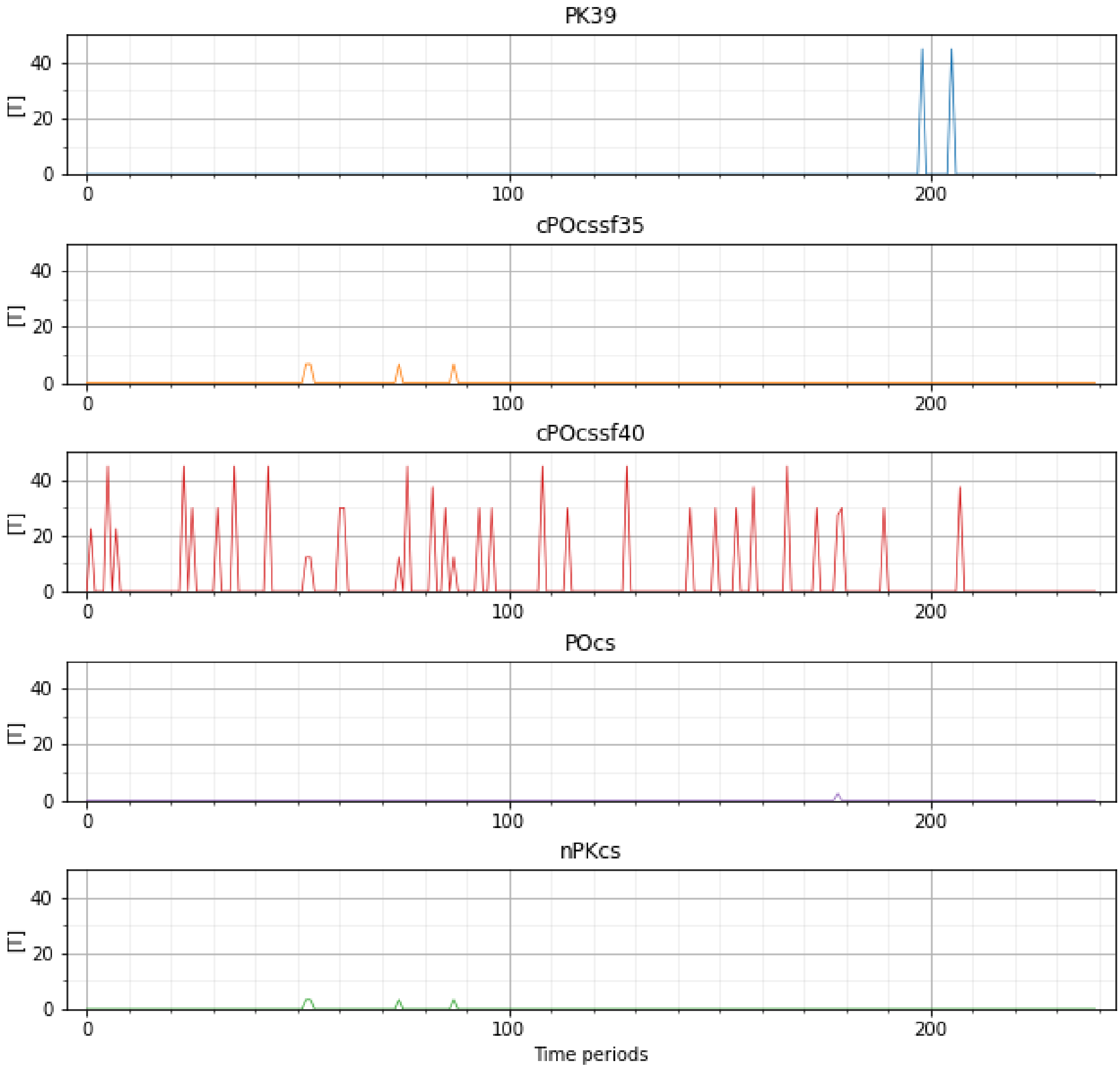

Figure D.41: Line graph of the decision variable R_{st} for the scheduling horizon of 240 time steps, the objective given by Equation 5.1 and an implemented batch-size of 15 T at $t > 96^{th}$ time step.

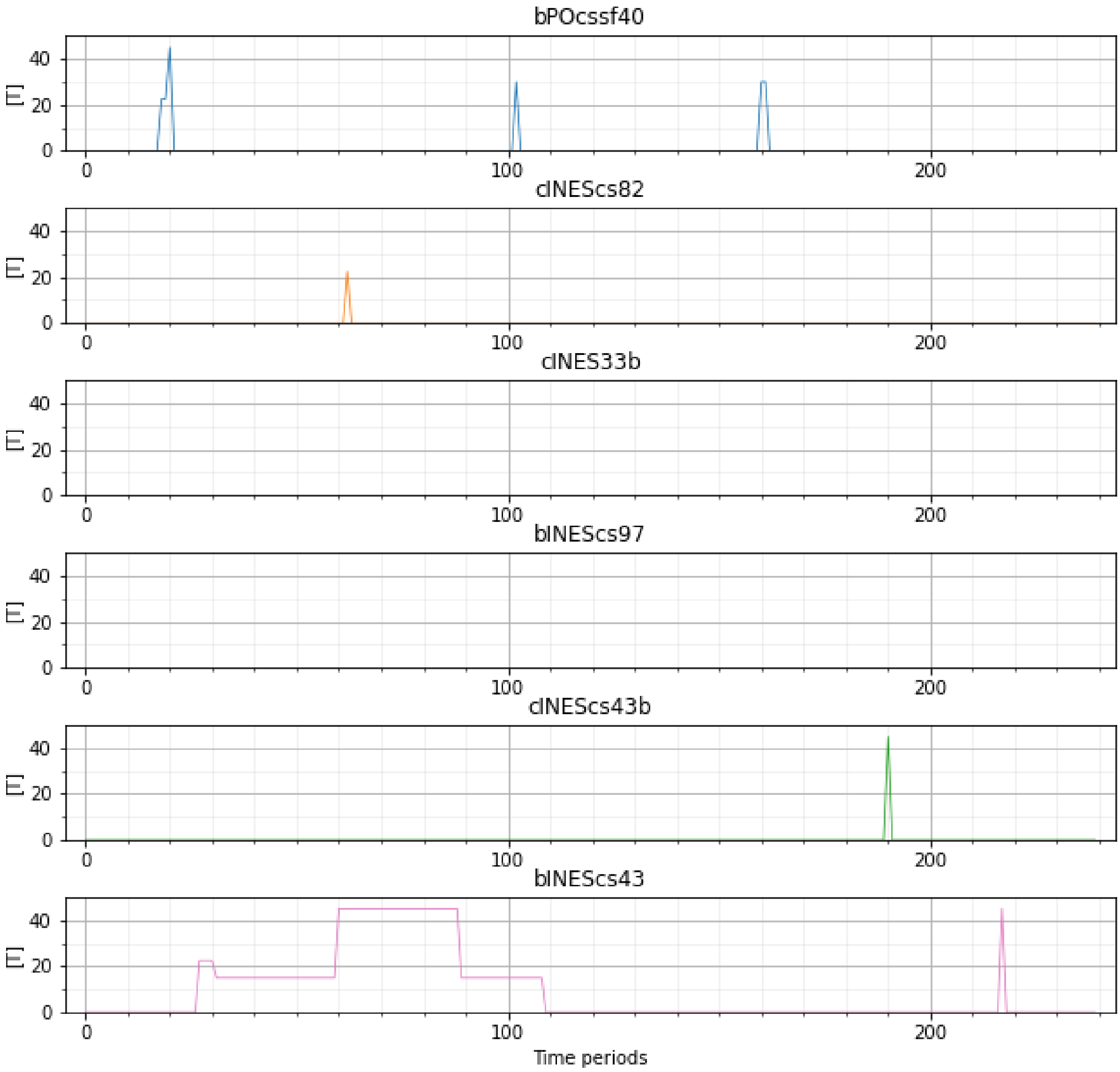

Figure D.42: Line graph of the decision variable S_{st} for the scheduling horizon of 240 time steps, the objective given by Equation 5.1 and an implemented batch-size of 15 T at $t > 96^{th}$ time step.

THIS PAGE IS LEFT BLANK INTENTIONALLY.

E
Mathematical model in Python-Gurobi

```
# -*- coding: utf-8 -*-

"""
Created on Tue Oct 27 2020 16:38:25

This Python script is made to implement in the research paper. The model has
been set up in such a way, that the different situations that have been
calculated are merged in one script. Due to this, this script will give errors
once it is run. It is only to use as an illustration how the mathematical
model was implemented in Python.

@author: Kirsten Lekkerkerk
"""

#%% IMPORTING PACKAGES

import pandas as pd
import numpy as np

from gurobipy import *

#%% LOADING THE INPUT

File_Path = r' ... Obj Paper INPUT.xlsx'
File_Path_Demand = r' ... DEMAND.xlsx'

data = pd.read_excel(File_Path, sheet_name = None, encoding = "utf-8",
                     header = None)
Demand = pd.read_excel(File_Path_Demand, sheet_name = None,
                     encoding = "utf-8", header = None)

del(File_Path, File_Path_Demand)

#%% LOAD EXCEL SHEETS WITH INPUT DATA

TASKS = data['Tasks (i)']
UNITS = data['Units (j)']
STATES = data['States (s)']
TIME_PERIODS = data['Times (t)']

rho_is = data['Proportion of input (rho_is)']
rho__is = data['Proportion of output (rho__is)']
p_is = data['Processing time (p_is)']
p_i = data['Processing time (p_i)']
V_ijmax = data['V_ijmax']
V_ijmin = data['V_ijmin']
C_s = data['Storage capacity (C_s)']
S_s0 = data['Initial storage (S_s0)']

#%  The demand to be implemented when a scheduling horizon of either 240 or
```

```
#   480 time steps is considered.
D_st = Demand['240_ACTSTD']
D_st = Demand['480_ACTSTD']

C_st = data['Unit cost (C_st)']
CS_st = data['Storage cost (CS_st)']

I_j = data['I_j']
I_jk = data['I_jk']
tau_jkl = data['tau_jkl']
K_i = data['K_i']
T_s = data['Task requiring s (T_s)']
T__s = data['Task producing s (T__s)']

del(data, Demand)

#%% LOAD TASKS, UNITS AND STATES

#%  TASKS
TASKS2 = TASKS[0].dropna().unique().tolist()
TASKS = tuple(TASKS2)
LENGTH_TASKS = len(TASKS)

#%  UNITS
UNITS = UNITS[0].dropna().unique().tolist()
UNITS = tuple(UNITS)
LENGTH_UNITS = len(UNITS)

#%  STATES
STATES = STATES[0].dropna().unique().tolist()
STATES = tuple(STATES)
LENGTH_STATES = len(STATES)

#%  States that can be stored and have a maximum shelf life
SSL = ['bINEScs97', 'cINEScs43b', 'bINEScs43', 'cINEScs82', 'bPOcssf40',
       'cINES33b']

#%  Input states
S_in = ['nPKcs', 'POcs', 'cPOcssf40', 'cPOcssf35', 'PK39']

#%  Output states
S_out = ['bINEScs97', 'cINEScs82', 'bPOcssf40', 'cINES33b', 'bINEScs43']

#% Time Periods or scheduling horizon
TIME_PERIODS = TIME_PERIODS[0].dropna().unique().tolist()
TIME_PERIODS = [round(x) for x in TIME_PERIODS]
TIME_PERIODS = tuple(TIME_PERIODS)
LENGTH_TIME_PERIODS = len(TIME_PERIODS)

#%% LOAD VARIABLES

#% rho_is
rho_is.columns = rho_is.iloc[0]
rho_is.set_index('Task', inplace = True)
rho_is = rho_is.drop([0], axis = 1)
rho_is = rho_is.drop([0, 'Task'], axis = 0)
rho_is = rho_is.stack().to_dict()

#% rho__is
rho__is.columns = rho__is.iloc[0]
rho__is.set_index('Task', inplace = True)
rho__is = rho__is.drop([0], axis = 1)
rho__is = rho__is.drop([0, 'Task'], axis = 0)
rho__is = rho__is.stack().to_dict()

#% p_is
p_is.columns = p_is.iloc[0]
p_is = p_is.iloc[:-3]
p_is.set_index('Task', inplace = True)
p_is = p_is.drop([0], axis =1)
p_is = p_is.drop([0, 'Task'], axis = 0)
```

```
p_is = p_is.stack().to_dict()

#% p_i
p_i.set_index(0, drop = True, inplace = True)
p_i = p_i.drop([0], axis = 0)
p_i = p_i.to_dict()[1]

#% V_ijmax
V_ijmax.columns = V_ijmax.iloc[0]
V_ijmax.set_index('Task', inplace = True)
V_ijmax = V_ijmax.drop([0, 'Task'], axis = 0)
V_ijmax = V_ijmax.stack().to_dict()

#% V_ijmin
V_ijmin.columns = V_ijmin.iloc[0]
V_ijmin.set_index('Task', inplace = True)
V_ijmin = V_ijmin.drop([0, 'Task'], axis = 0)
V_ijmin = V_ijmin.stack().to_dict()

#% C_s
C_s.set_index(0, drop = True, inplace = True)
C_s = C_s.drop([0], axis = 0)
C_s = C_s.to_dict()[1]

#% S_s0
S_s0.set_index(0, drop = True, inplace = True)
S_s0 = S_s0.drop([0], axis = 0)
S_s0 = S_s0.to_dict()[1]

#% D_st
D_st = D_st.drop([0], axis = 0)
D_st = D_st.drop([1], axis = 0)
D_st.set_index(0, drop = True, inplace = True)
D_st = D_st.drop([0], axis = 0)
D_st[1] = D_st[1].astype(float)
D_st = D_st.stack().to_dict()

#% C_st
C_st = C_st.drop(0, axis = 0)
C_st.set_index(0, drop = True, inplace = True)
C_st = C_st.drop([0], axis = 0)
C_st[1] = C_st[1].astype(float)
C_st = C_st.stack().to_dict()

#% CS_st
CS_st = CS_st.drop(0, axis = 0)
CS_st.set_index(0, drop = True, inplace = True)
CS_st = CS_st.drop([0], axis = 0)
CS_st[1] = CS_st[1].astype(float)
CS_st = CS_st.stack().to_dict()

#% sl_s
sl_s = {SSL[0]: 144,
        SSL[1]: 144,
        SSL[2]: 144,
        SSL[3]: 144,
        SSL[4]: 144,
        SSL[5]: 144}

#%% LOAD SETS

#% I_j
I_j = I_j.transpose()
I_j.set_index(0, drop = True, inplace = True)
I_j['LIST'] = I_j.values.tolist()
I_j = I_j['LIST'].to_dict()
I_j = {k: [elem for elem in v if elem is not np.nan] for k, v in I_j.items()}

#% I_jk
I_jk = I_jk.transpose()
I_jk.set_index(0, drop = True, inplace = True)
```

```
I_jk.columns = I_jk.iloc[0]
I_jk = I_jk.drop(['Family'], axis = 0)

for j in UNITS:
    I_jk.loc[j] = I_jk.loc[j].str.split(', ').tolist()

for column in list(range(0, 3)):
    I_jk.loc[I_jk[column].isnull(), [column]] =
        I_jk.loc[I_jk[column].isnull(), column].apply(lambda x: [])

del column

I_jk = I_jk.stack().to_dict()

#% tau_jkl
Header = tau_jkl.loc[0]
tau_jkl = tau_jkl[1:]
tau_jkl.columns = Header
del(Header)
tau_jkl = tau_jkl.set_index(['Unit', 'k'])

tau_jkl = {r+(k,):v for r, kv in tau_jkl.iterrows()
                    for k, v in kv.to_dict().items()}

#% K_i
K_i.set_index(0, drop = True, inplace = True)
K_i['LIST'] = K_i.values.tolist()
K_i = K_i.drop([0], axis = 0)
K_i = K_i['LIST'].to_dict()
K_i = {k: [elem for elem in v if elem is not np.nan]
                for k, v in K_i.items()}

#% T_s
T_s.set_index(0, drop = True, inplace = True)
T_s['LIST'] = T_s.values.tolist()
T_s = T_s.drop([0], axis = 0)
T_s = T_s['LIST'].to_dict()
T_s = {k: [elem for elem in v if elem is not np.nan] for k, v in T_s.items()}

#% T__s
T__s.set_index(0, drop = True, inplace = True)
T__s['LIST'] = T__s.values.tolist()
T__s = T__s.drop([0], axis = 0)
T__s = T__s['LIST'].to_dict()
T__s = {k: [elem for elem in v if elem is not np.nan] for k, v in T__s.items()}

#Number of disjoint families for cleaning task regarding to contamination
NF_j = {'00X00': list(range(0, 3)),    #Start of processing line
        '00X01': list(range(0, 3)),    #Start of processing line
        '00X02': list(range(0, 3)),    #Start of processing line
        '00X03': list(range(0, 3)),    #Belongs to multiple lines
        '00X04': list(range(0, 3)),    #Belongs to multiple lines
        '00X05': list(range(0, 3)),    #Belongs to multiple lines,
                                       #00X05 always has the same order as
                                       #00X01.
        '00X06': list(range(0, 3)),    #Start of processing line
        '00X07': list(range(0, 3))}    #Start of processing line

#%% DEFINE MODEL

MWE=Model('MWE')    #Minimum Working Example

#%% Define decision variables

#   Amount of material which starts undergoing task i in unit j at the
#   beginning of time period t.
B_ijt = MWE.addVars(TASKS, UNITS, TIME_PERIODS, lb=0, ub=GRB.INFINITY,
                          vtype=GRB.CONTINUOUS, name="B_ijt")

#   1 if unit j starts processing task i at the beginning of time period t;
#   0 otherwise.
```

```
W_ijt = MWE.addVars(TASKS, UNITS, TIME_PERIODS, vtype=GRB.BINARY,
                    name=”W_ijt”)

#   Amount of material stored in state s, at the beginning of time period t.
S_st = MWE.addVars(STATES, TIME_PERIODS, lb=0, vtype=GRB.CONTINUOUS,
                   name=”S_st”)

#   1 if state s is stored at the beginning of time period t.
BS_st = MWE.addVars(STATES, TIME_PERIODS, vtype=GRB.BINARY, name=”BS_st”)

#   Amount of material of feed state s received from external sources at
#   time t.
R_st = MWE.addVars(STATES, TIME_PERIODS, lb=0, vtype=GRB.CONTINUOUS,
                   name=”R_st”)

MWE.update()

#%% Add constraints
#%  Constr. 4.2, the allocation constraint

M = 100000

for j in UNITS:
    for t in TIME_PERIODS:
        for i in I_j[j]:
            if t > TIME_PERIODS[-p_i[i]]:
                MWE.addConstr(W_ijt[i, j, t] == 0)
            else:
                MWE.addConstr(quicksum(W_ijt[ia, j, ta] for ia in I_j[j]
                    for ta in [m for m in range(t, t+p_i[i])])-1
                <=M*(1-W_ijt[i, j, t]))

del(i, j, t)

#%  Constraint 4.3a part 1. Capacity limitations of the units (minimum)
MWE.addConstrs(W_ijt[i, j, t] * V_ijmin[i, j] <= B_ijt[i, j, t]
    for i in TASKS for j in K_i[i] for t in TIME_PERIODS)

#%  Constraint 4.3a part 2. Capacity limitations of the units (maximum)
MWE.addConstrs(B_ijt[i,j,t] <= W_ijt[i,j,t]*V_ijmax[i,j]
    for i in TASKS for j in K_i[i] for t in TIME_PERIODS)

#%  Constraint 4.3b part 1. Capacity limitations on storage (minimum)
MWE.addConstrs(0 <= S_st[s, t] for s in STATES for t in TIME_PERIODS)

#%  Constraint 4.3b part 2. Capacity limitations on storage (maximum)
MWE.addConstrs(S_st[s, t] <= C_s[s] for s in STATES for t in TIME_PERIODS)

#%  Constraint 4.4 part 1. Material balance: initial storage level
MWE.addConstrs(S_st[s, TIME_PERIODS[0]] == S_s0[s] for s in STATES)

#%  Constraint 4.4 part 2. Material balance
MWE.addConstrs(S_st[s, t] == S_st[s, TIME_PERIODS[TIME_PERIODS.index(t)-1]]\
      + (quicksum(rho__is[i, s] *
         quicksum(B_ijt[i, j, TIME_PERIODS[TIME_PERIODS.index(t)-p_is[i, s]]]
         for j in K_i[i]) for i in T__s[s]))
      - (quicksum(rho_is[i, s] * quicksum(B_ijt[i, j, t]
         for j in K_i[i]) for i in T_s[s]))
      + R_st[s, t]
      - D_st[s, t]
      for s in STATES for t in TIME_PERIODS if t != TIME_PERIODS[0])

#%  For R_st, only the feed states can be deliverd from external sources
#   at time t.
FEED = list(STATES)

FEED = [x for x in FEED if x not in S_in]

FEED = tuple(FEED)

MWE.addConstrs(R_st[s, t] == 0 for s in STATES for t in TIME_PERIODS)
```

```python

#%  It is not desirable to start a task at the first time period of the
#   scheduling horizon, similar to Constraint 4.5.
MWE.addConstrs(W_ijt[i, j, TIME_PERIODS[0]] == 0 for i in TASKS for j in UNITS)

del(FEED)

#%  Constraint 4.6. Sequence dependent cleaning - time reservation. Here used
#   for contamination/production order
for j in UNITS:
    for t in TIME_PERIODS:
        for k in NF_j[j]:
            for l in NF_j[j]:
                if l != k:
                    for i in I_jk[j, k]:
                        if t > TIME_PERIODS[-p_i[i]-tau_jkl[j, k, l]]:
                            MWE.addConstr(W_ijt[i, j, t] == 0)
                        else:
                          MWE.addConstr(quicksum(W_ijt[ia, j, ta]
                          for ia in I_jk[j, l]
                          for ta in
                          [m for m in
                             range(t + p_i[i], t + p_i[i] + tau_jkl[j, k, l])])
                          <= M * (1 - W_ijt[i, j, t]))

#%   Constraint 4.7. BS_st equals 1 when state s is stored at the beginning
#    of time t.
MWE.addConstrs(S_st[s, t] == BS_st[s, t]*S_st[s, t]
        for s in STATES for t in TIME_PERIODS)

#%   Constraint 4.8. Maximum shelf life
for s in SSL:
    for t in TIME_PERIODS:
        if TIME_PERIODS[-1] - t >= sl_s[s]+2:
            MWE.addConstr(quicksum(BS_st[s, ta]
                for ta in [m for m in range(t, t+sl_s[s]+2)]) <= sl_s[s])

del(M)
del(i, j, t, k, l)

#%% OBJECTIVE FUNCTION

Horizon = TIME_PERIODS[1:-1]

obj4.1 = quicksum(C_st[s, TIME_PERIODS[-1]]*S_st[s, TIME_PERIODS[-1]]\
                    for s in STATES)\
       +quicksum(C_st[s,t]*D_st[s,t] for s in STATES for t in Horizon)\
       -quicksum(C_st[s, TIME_PERIODS[0]]*S_st[s, TIME_PERIODS[0]]\
                    for s in STATES)\
       -quicksum(C_st[s, t]*R_st[s, t] for s in STATES for t in Horizon)\
       -quicksum(CS_st[s, t]*S_st[s, t] for s in STATES for t in Horizon)

obj5.1 = quicksum(BS_st[s, t] for t in TIME_PERIODS for s in SSL)

obj5.3 = quicksum(S_st[s, TIME_PERIODS[-1]] for s in S_out)

del (TASKS, STATES, UNITS, TIME_PERIODS)
del (LENGTH_TASKS, LENGTH_STATES, LENGTH_UNITS, LENGTH_TIME_PERIODS)
del (SSL, S_in, S_out, sl_s)
del (V_ijmax, V_ijmin, C_s)
del (K_i, I_j, I_jk, NF_j, tau_jkl)
del (CS_st, C_st)
del (D_st)
del (S_s0)
del (T__s, T_s)
del (p_i, p_is)
del (rho__is, rho_is)
del (Horizon)

#%% MODEL OPTIMIZATION
#   One of these objectives is to be choosen to optimize
```

```
MWE.setObjective(obj4.1, GRB.MAXIMIZE)
MWE.setObjective(obj5.1, GRB.MINIMIZE)
MWE.setObjective(obj5.3, GRB.MAXIMIZE)

MWE.optimize()
```

Listing E.1: Mathematical model in Python-Gurobi

THIS PAGE IS LEFT BLANK INTENTIONALLY.

F

Scientific Paper

Scheduling optimization in a refinery for vegetable oils

K. Lekkerkerk BSc
Delft University of Technology
Transport Engineering and Logistics

Dr. F. Schulte
Delft University of Technology
Transport Engineering and Logistics

Drs. J. Vernooy
Sime Darby Oils Zwijndrecht Refinery

Dr. Ir. D. L. Schott
Delft University of Technology
Transport Engineering and Logistics

November 20, 2020

Abstract

In this paper, a mathematical model has been developed to determine the timing of the tasks that can be performed on a variety of units to produce a variety of end states and to monitor the flow of the material through the network. The mathematical model is based on the work of Kondili et al. (1993) and is extended with the ability to monitor and restrict the shelf-life of a state, the ability to implement a mandatory storage action with shared storage and the ability to implement batch-size dependent processing times for batch-sizes that can only take a certain value.

I. Introduction

In the years to come, the world population will grow. It is the expectation of the United Nations that the population will grow from over 7.6 billion in 2018 to almost 9.8 billion people in 2050 (United Nations, Department of Economic and Social Affairs, Population Division, 2019). A big concern is how to feed all these people. Besides the growth of the world population, also the growth in welfare contributes to the demand for a higher food production (van Kasteren, 2013). Companies that process vegetable oils for among others food and care products experience this growth in world population already or in the near future by an increasing demand for their products. At some point these companies may have trouble with continuing meeting the demand of their customers and need to expand their capacity. For some companies a change of scheduling tactics might be sufficient to unlock at least a part of the newly required capacity.

In literature, a lot of scheduling optimization problems regarding oil products have been discussed, but are particularly focused on crude industrial oil products. For example, Lee et al. (1996) specifies a mixed-integer linear programming (MILP) optimization model for short-term crude oil unloading, tank inventory management and a Crude Distillation Unit (CDU) charging schedule. The example discussed consists of one docking station, one CDU and several storage and charging tanks. Before the oil is charged into a CDU, it is mixed for the right composition in the charging tanks. Moro and Pinto (2004) discuss a problem of crude oil inventory management where the crude oil is delivered by a pipeline and the transfers from the pipeline to the crude tanks, the settling time, interface separation between the different types of oil and the charging of the CDU's are considered. Magatão et al. (2004) discusses a problem where a long bidirectional pipeline connects a harbor to an inland refinery. In the example discussed, the length of the pipeline is defined to be almost 100 km long. The pipeline transfers a limited set of products, where some orders of transfer are not recommended based on product specifications. To overcome this limitation, a plug (small volume of product) can be used to avoid specific interfaces. It does, however, increase the operational cost. A more complex pipeline scheduling system is described by Cafaro and Cerdà (2008). Here, a unidirectional multi-product pipeline connects a single origin to multiple distribution terminals, where the products are stored for further distribution. The amount of products transferred by the pipeline varies and the distance over which the product is transferred depends on its destination. The scheduling problem has to update the sequence and volumes of new product batches to be pumped in the pipeline dynamically throughout a multi-period rolling horizon. Research has also been done to crude oil scheduling in situations that could be considered further downstream than has been discussed earlier. For example in Göthe-Lundgren et al. (2002), where a scheduling problem is described for an oil refinery company that has one CDU and two hydro-treatment units. In this problem, inventory is also taken into account. In order to produce the required amounts of the products the refinery is able to produce, the CDU can run in 10 modes, and the hydro-treatment units can run in 10-15 modes. Changing modes is expensive, so long sequences and few changeovers are preferred. However, longer sequences require more storage capacity, resulting in higher storage costs. It is up to the scheduling optimization model to find a balance between these two factors.

All scheduling problems discussed by literature mentioned above can be classified as batch processes. Literature discusses these batch processes in a much more general way as well. For example, Maravelias (2012) discusses a more general framework and modelling approach in the field of chemical production scheduling. The framework and approach is based on the general classification for scheduling problems of batch processes presented in Méndez et al. (2006a). This general classification has often been used to create a well structured mathematical model for batch processes. A general algorithm used as basis by many researchers is the algorithm developed by Kondili et al. (1993). It is a general algorithm for short-term scheduling of batch operations and consists of a MILP model. The model is based on a network representation defined as a State-Task-Network (STN), newly presented in this work. Based on the work of both Kondili et al. (1993) and Méndez et al. (2006a) researchers expand the possibilities of scheduling optimization by going more into detail on specific aspects of the classification as for example the flow shop scheduling problem discussed by Birewar and Grossmann (1990) and Lin et al. (2002) or the multiple product demand scheduling problem discussed by Ierapetritou et al. (1999).

II. Problem definition

When the challenges are taken into account which companies processing vegetable oils encounter during their production processes, the current scheduling optimization solutions will not be sufficient. Examples of restrictions companies encounter apart from the restrictions given in the classification of Kondili et al. (1993) are a mandatory storage action for a product in a shared storage unit, processes with batch-size dependent processing times where the batch-size can only take a particular value and a maximum storage time or shelf-life. The first two restrictions are briefly addressed by Kondili et al. (1993), but are discussed as a voluntary storage action for a product in a shared storage unit and as a process with batch-size dependent processing times where each processing time covers a range of batch-sizes, respectively. Entrup et al. (2005) introduces a scheduling approach where customers

pay a price for a unit of yoghurt - a perishable product - related to the time it is in storage. So, the shorter the unit of yoghurt is in storage, the more the customers pay since it will have an higher freshness. However, vegetable oils are less perishable than yoghurt. Therefore, it might be that a company manages a maximum shelf-life, without decreasing the selling price once a product remains in storage for a longer time.
Therefore, this paper will develop a mathematical model taking the three points discussed above into account. The mathematical model is based on the mathematical model provided by Kondili et al. (1993). In the next section, the mathematical model will be elaborated, followed by an illustrative example in Section IV. The results will be given in Section V and Section VI will conclude the paper with the conclusion and recommendations.

III. Mathematical formulation

In mathematical modelling, there are three fundamental or primary constraints. The primary constraints concern the allocation constraint, the capacity limitations on units and storage and the material balance constraint. Apart from the primary constraints, mathematical models might also have additional or secondary constraints that represent among others requirements or limitations that emerge from the situation to be described or modelled. One could think of temporary unavailability of units, a mandatory shared storage policy, limited equipment connectivity, batch-size dependent processing times, production order due to possible contamination and a maximum shelf-life. The nomenclature is given in Appendix A.

The objective defined by Kondili et al. (1993) is able to accommodate various performance measures such as value of products, cost of feedstock, cost of storage and cost of utilities resulting in a profit function to be maximized. Since this mathematical model does not take into account utility cost, only the first three elements of the profit function will be considered for the objective of this mathematical model.

$$Profit = Value\,of\,products - Cost\,of\,feedstock - Cost\,of\,storage$$

$$Value\,of\,products = \sum_s \left(C_{s,H+1} S_{s,H+1} + \sum_{t=1}^{H} C_{st} D_{st} \right)$$

$$Cost\,of\,feedstock = \sum_s \left(C_{s0} S_{s0} + \sum_{t=1}^{H} C_{st} R_{st} \right)$$

$$Cost\,of\,storage = \sum_s \sum_{t=1}^{H} C_{st}^{S} S_{st}$$

III.I. Primary constraints

Allocation constraint

Any item of equipment, or unit, can only start and perform at most one task at the same time. The next task to be performed in the same unit can only start after the current task has been finished. The constraint is only binding if $W_{ijt} = 1$, forcing all other $W_{i'jt'}$ to be zero. M is considered to be a very large number.

$$\left(\sum_{i' \in I_j} \sum_{t'=t}^{t+p_i-1} W_{i'jt'} \right) - 1 \leq M \left(1 - W_{ijt}\right) \qquad \forall j,\, t,\, i \in I_j$$

Unit capacities

The amount of material that can be processed at time t depends on the combination of task i and unit j. The maximum and minimum capacity of each combination of task i and unit j, V_{ij}^{max} and V_{ij}^{min} respectively, is known in advance. The constraint forces the batch-size B_{ijt} to be zero if $W_{ijt} = 0$.

$$W_{ijt} V_{ij}^{min} \leq B_{ijt} \leq W_{ijt} V_{ij}^{max} \qquad \forall i,\, t,\, j \in K_i$$

Maximum storage capacity

The amount of material stored in a state s is not allowed to exceed the maximum storage capacity C_s. This constraint only takes into account the situation where the storage for each state s is dedicated to that particular state.

$$0 \leq S_{st} \leq C_s \qquad \forall s,\, t$$

Material balance

The storage level of state s at time t (S_{st}) equals the storage level of the same state s at time $t-1$ (S_{st-1}), added to the amount of state s produced at time t, subtracted with the amount of state s

used at time t, added to the amount of state s delivered from an external supplier (R_{st}) at time t, subtracted with the amount of state s that was delivered to an external customer (D_{st}) at time t. The initial storage level of state s at time $t = t_0$, S_{st_0}, is known in advance.

$$S_{st} = S_{s,t-1} + \sum_{i \in \overline{T}_s} \left(\overline{\rho}_{is} \sum_{j \in K_i} B_{ij,t-p_{is}} \right) - \sum_{i \in T_s} \left(\rho_{is} \sum_{j \in K_i} B_{ijt} \right) + R_{st} - D_{st} \quad \forall s,\ t \neq t_0$$

III.II. Secondary constraints

Temporary unavailability of units

Due to for example maintenance or non-working periods, it might be possible that a unit is not available for some time during the time horizon. A way to deal with this is to assign the value zero to an appropriate subset of W_{ijt} variables.
If equipment item j is unavailable between times t_1 and t_2 when $t_2 > t_1$, then it is not able to start processing any task at the start of the intervals t_1 to $t_2 - 1$. Furthermore, the item must already be idle at time t_1, which implies that it must not have started any task i at any time after $t_1 - p_i + 1$.

$$W_{ijt} = 0 \qquad \forall i \in I_j,\ t = t_1 - p_i + 1, \dots, t_2 - 1$$

Mandatory shared storage units

Kondili et al. (1993) describes a shared storage policy for which the storing of a state is a voluntary storage action. This means that it is not necessary to store the state before it is used for another task. However, there might be situations where a state is to be stored mandatory before it can be used as an input state for another task. An example could be a buffer tank standing between two units which should be used to store the intermediate task, since the second unit needs a smaller, fixed amount of the intermediate state at once than the output of the first unit delivers. In that case, a separate storing task should be created with different in- and output states, just as a normal task. The voluntary shared storage policy can then be applied to the output state.

Limited equipment connectivity

In most real world situations, it is not possible or desirable to be able to perform all tasks of a single family on all units. Besides, there might be processing lines which give physical limitations on the connectivities. For example, assume that units A1 and B1 can both perform a task i_1 and that A2 and B2 can perform a task i_2. When there is no limited equipment connectivity, the output states of both A1 and B1 can be used as input states for both A2 and B2. However, when there is a restriction on the equipment connectivity, it might be that B2 is only able to process the input state of B1. A mathematical model deals with this restriction by splitting task i_1 and i_2 in tasks i_{11}, i_{12}, i_{21} and i_{22} respectively and by splitting state s_2 in state s_{21} and s_{22}. The values of variables p_{is}, ρ_{is} and $\overline{\rho}_{is}$ link for example task i_{11} and state s_1 and task i_{12} and state s_{22} to each other in the same way they do for the original task i and state s. The STN showing the described principle is shown in the figure below.

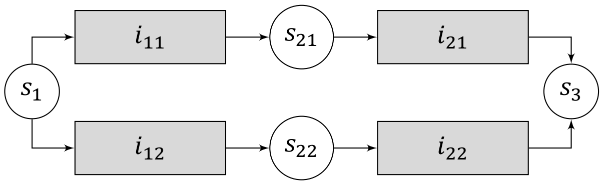

Figure F.1: A simplified STN representing limited equipment connectivity (Kondili et al., 1993).

Batch-size dependent processing times

In the definition of batch-size dependent processing times given by Kondili et al. (1993) states that each processing time covers a range of batch-sizes. Mathematically, this can be written as:

For batchsize B ∈ $[0, B^1]$: Processing time p^1
For batchsize B ∈ $(B^1, B^2]$: Processing time p^2
For batchsize B ∈ $(B^2, B^3]$: Processing time p^3
For batchsize B ∈ $(B^3, B^4]$: Processing time p^4

Each of the instances written above can be seen as a separate task. Just as with the limited equipment connectivity described in the previous paragraph, the tasks perform the same action, but have different batch-sizes - and thus different minimum and maximum capacities V_{ij}^{min} and V_{ij}^{max} and different processing times.
In the situation where the batch-size can only take certain values, for example 5, 10, 15 and 20 kg and

no other value in between, the minimum and maximum capacities V_{ij}^{min} and V_{ij}^{max} will be set equal to each other for each task. Its application will be the same as the situation described above. Mathematically, this can be written as:

For batchsize B $\in [B^1, B^1]$: Processing time p^1
For batchsize B $\in [B^2, B^2]$: Processing time p^2
For batchsize B $\in [B^3, B^3]$: Processing time p^3
For batchsize B $\in [B^4, B^4]$: Processing time p^4

Production order

Kondili et al. (1993) discusses a constraint that deals with equipment cleaning. The cleaning of the equipment might be needed in some, but not all, situations. In a dye manufacturing plant, little or no cleaning is required when a dark paint is produced after a light paint. However, when a light paint needs to be produced after a dark paint, extensive cleaning is needed. This concept is described as sequence dependent cleaning. In mathematical modelling, sequence dependent cleaning can be approached in two ways. For the first method, an actual cleaning task is created, for which the model should decide when the task needs to be performed. For the second method, only the time required for the cleaning task is taken into account. For both methods, I_j is split into NF_j disjoint families I_j^k, $k = 1, \ldots, NF_j$. Then a (virtual) cleaning task i_{jkl} is created with a processing time of τ_{jkl}, where $l = 1, \ldots, NF_j$, $l \neq k$.
For this research, the second method is assumed to be sufficient. More information on the first method can be found in Kondili et al. (1993). For the second method, it is sufficient to add a constraint stating that when unit j has processed a task of family k, no task of family l is started within τ_{jkl} time units after the task of family k in unit j has finished.

$$\sum_{i' \in I_j^{(l)}} \sum_{t'=t+p_i}^{t+p_i+\tau_{jkl}-1} W_{i'jt'} \leq M\left(1 - W_{ijt}\right) \qquad \forall j, i \in I_j^{(k)}, t$$

The concept described here for sequence dependent cleaning can also be considered to be a method to force a certain production order. The application of the constraint works the same. Namely, for a product of family l that is processed after a product of family k, the value of 'cleaning time' τ_{jkl} can be made sufficiently large to force the model not to create a production order where a product of family l follows a product of family k.

Maximum storage time

Two additional constraints are needed to model the maximum storage time or shelf-life. First, the storage time should be tracked. This is done by adding a decision variable to the model, BS_{st}, which is equal to 1 when $S_{st} > 0$ as described by the first constraint written down below. It should be noted that this constraint does not force BS_{st} to be 0 when $S_{st} = 0$. Second, the number of time periods t should not exceed the maximum shelf-life of the product or state s, sl_s. This is covered by the constraint shown by the second constraint written down below. This equation states that in the range of time periods from t up until $t + sl_s + 1$ the number of BS_{st} is not allowed to exceed the maximum shelf-life sl_s. Because the time range is one time period larger than the shelf-life itself, at least one of the BS_{st} in this range of time periods is forced to be zero, forcing the mathematical model not to exceed the shelf-life of a state s. This way of monitoring does not take into account different tanks if there are any. As long as a state is stored, the sum of the amount of storage actions will increase. The sum will only be interrupted and reset when there is at some point no storage of that particular state.

$$S_{st} = BS_{st} \cdot S_{st} \qquad \forall s, t$$

$$\sum_{t}^{t+sl_s+1} BS_{st} \leq sl_s \qquad \forall s \in SSL$$

IV. Illustrative example

An example is given to illustrate the mathematical model. The size of the model given here is bigger than the illustrative example discussed by Kondili et al. (1993). A scheduling horizon of 10 days is considered, divided in 480 time steps of 30 minutes. The illustrative example exists of 8 units, which are listed in Table X. Bleaching tank [...], filter unit [...] and buffer [...] form one processing line. However, filter unit [...] is shared with the processing line of the chemical interesterification cauldrons. The chemical interesterification cauldrons operate in parallel and both have buffer [...] as a buffer. Bleaching units [...] and [...] represent a full processing line that both consist of a bleaching tank, a filer unit and a buffer as well. However, since their equipment is not used by other processing lines, they are merged.

Each unit has a minimum and maximum capacity and a number of possible batch-sizes. Besides, each unit is able to execute a number of tasks. The description of each task number is given in the Appendix B. The illustrative example can produce five states as an output from five different states of input which can also be combined. A list of states is given in Appendix C. In Table F.1, information is added on the states. The states can be classified in three families of products. This are Family 0, Family 1 and Family 2. It is not allowed for a product of Family 2 to be processed within 5 time steps after a product of Family 0 when there is no state processed in between from Family 1.

Table F.1: Information on states (see Appendix C)

Raw Material delivery (input states)	1, 7, 13, 14 and 20
No storage capacity	2, 3, 4, 5, 8, 10, 11, 15, 17, 18 and 21
Unlimited storage capacity	6, 9, 12, 16, 19 and 22
Output states	6, 12, 16, 19 and 22

For each of the five states that can be produced as an output, a STN is presented. The numbers given in the rectangles correspond to a task, whereas the numbers given in a circle or oval correspond to a state. Usually, the STN's are accompanied with a processing time p_{is}, required by a task to produce an (intermediate) state. However, the STN's presented here include multiple tasks in the task-box due to the batch-size dependent processing times. Separate tasks are required to account for this requirement and go hand in hand with different processing times. Therefore, the processing time for each task is given in Appendix B.

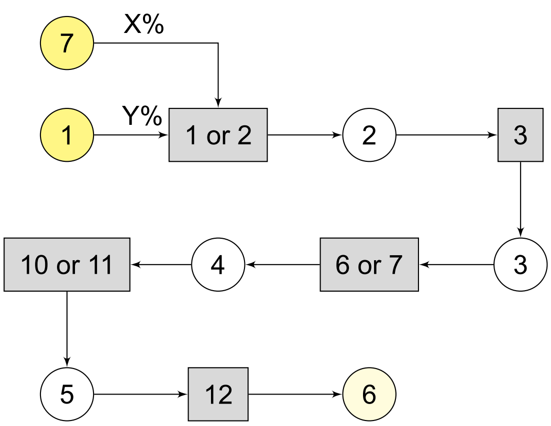

Figure F.2: The STN for the production of state 6.

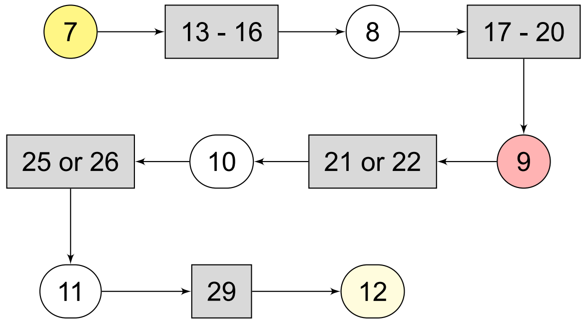

Figure F.3: The STN for the production of state 12.

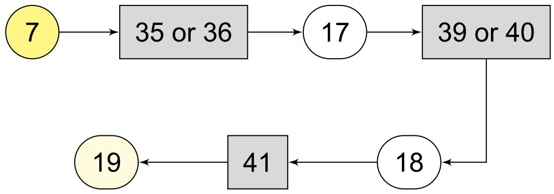

Figure F.4: The STN for the production of state 19.

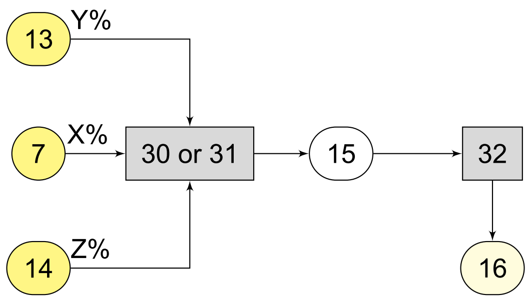

Figure F.5: The STN for the production of state 16.

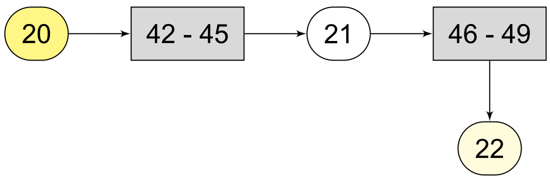

Figure F.6: The STN for the production of state 22.

The fictional running cost of keeping in storage a state is 5 units per tonne for all states but the end states. For these states, it is 15 units per tonne to store them. The unit price of 15 units per tonne is given to the end states. All other states receive a unit price of 0 units per tonne. Since it is not desirable to have one of the states remaining in storage at the end of the scheduling horizon, the unit cost of the states that can be stored will have a value of -15 units per tonne at the final time step.

With the information provided above, the scheduling problem can be formulated in a similar way as Kondili et al. (1993) did for their scheduling problem:

Given:
The State-Task-Network for the production of five different states, all the information required for these STN's, the time horizon of interest and the demand for the five states to be produced with multiple due dates throughout that time horizon of interest.

Determine:
The timing of the tasks that can be performed in each unit and the flow of the material through the network.

So as to *optimize* the objective function.
Assumptions that are made for the purposes of this paper:

- Instantaneous material transfer between units
- The data used is deterministic and fixed over the time horizon
- No resource constraints
- No time constraints

V. Results and validation

The mathematical model is solved on a HP EliteBook 8570w with an Intel® Core™ i7-3630QM CPU @2.40 GHz processor and 8 GB of RAM. For the solving, the mathematical programming language of Python, version 3.7 combined with the Gurobi solver, version 9.0.0, has been used. The Optimal schedule derived from the mathematical model by these tools is shown in Figure X in Appendix D. In this figure, the units introduced in Table X are listed at the y-axis. The time periods are shown on the x-axis. All bars are divided in two parts. The upper part shows the task number, whereas the lower part shows the number of the family of products. In Figure F.7 - also in the appendix, the storage level per state that can be stored is shown.
The maximum profit of this problem is calculated as -29.062,5 units. The mathematical formulation of this problem results in 10.560 constraints, 209.280 continuous variables and 198.720 integer, binary variables. It took 20.238,04 seconds to solve the mathematical model.

The first impression when checking the Ganttchart of B_{ijt} shown in Figure X is that the (newly) implemented constraints - the mandatory storage action for a product in a shared storage unit, the processes with batch-size dependent processing times where the batch-size can only take a particular value and a maximum storage time or shelf-life - are not violated according to this figure. No tasks processing a product of product Family 2 is scheduled within 5 time steps after a task that processes a product of Family 0. Also no other batch-sizes were started then the ones defined in Table X. Checking the line graph of S_{st} in Figure F.7 shows that - according to this figure - the maximum shelf-life is also not violated. However, no violation of the processing order and the maximum shelf-life might also be due to the fact that the mathematical model is not pushed to its limits. It might be that the demand used for this paper was set up in such a way that a change of order was not needed to prevent a product of Family 2 being processed within five time steps after a product of Family 0. This demand might also be the reason that the shelf-life did not reach its limits, because the demand was not high enough to push the mathematical model to its limits. By means of a sensitivity analysis it has been investigated if the newly developed restriction on maximum shelf-life that is implemented has an influence on the results of the mathematical model. Therefor, the mathematical model has been pushed to its limits by asking the mathematical model to produce as much as it possibly can during the scheduling horizon. The results showed that the implementation of the maximum shelf-life indeed showed a limitation on the time that the mathematical model was producing in order not to violate the maximum shelf-life restriction. However, even though the mathematical model was not producing any product right from the start of the scheduling horizon, it did violate the maximum shelf-life limitation by one time step.

To investigate the production order, a sensitivity analysis has been performed as well. This has been done by increasing the time that needs to be in between a task processing a product of Family 2 that is executed after a task processing a product of Family 0. This value has been increased from 5 time steps to the length of the scheduling horizon. This equals a bid on a certain processing order. The result of this situation showed an infeasible solution. When the value was increased from 5 to half of the scheduling horizon - which equals a discourage on a certain processing order, the mathematical gave a solution. In this solution, a task processing a product of Family 2 was not scheduled after a task processing a product of Family 0 when there was no task processing a product of Family 1 in between.

The validation of the mathematical model also showed that the mathematical model would give an infeasible solution when a batch-size smaller than 22,5 T was implemented. It was expected that when the demand would be less than 22,5 T, the mathematical model would process a batch of 22,5 T and keep the surplus in storage after delivery. A sensitivity analysis showed that the mathematical model is only able to handle demands that are a multiple of 7,5 T. The maximum shelf-life restriction had no influence on this issue.

VI. Conclusions and recommendations

In this paper, a mathematical model has been developed based on the mathematical model developed by Kondili et al. (1993). A major contribution to this work as the development of a method to implement a restriction on maximum shelf-life for products who suffer from instant deterioration instead of gradual deterioration. This method works since it does give a limitation on the consecutive storage actions, but the behaviour of this method should be investigated in more detail.
Contributions to this work are also covered by adapting existing methods to apply them to a more specific situation. This concerns the constraint on a mandatory intermediate storage and a batch-size dependent processing time where the batch-size can only take a particular value. Finally, the application of a sequence dependent cleaning task in order to force a required processing order has also been successfully implemented in the mathematical model.

A limitation of this mathematical model can be found in the inability to handle demands with a batch-size that are not a multiple of 7,5 T. It is recommended to investigate this behaviour of the mathematical model and improve the mathematical model to overcome this limitation.
The sensitivity analysis showed that the newly developed method on a limitation on shelf-life works, but the results also showed that the implemented shelf-life was violated by one step. Since this newly developed methodology shows promising results, it is recommended to further investigate this behaviour and improve the method.

References

Birewar, D. B. & Grossmann, I. E. (1990). Simultaneous Production Planning and Scheduling in Multi-product Batch Plants. *Industrial and Engineering Chemistry Research*, 29, 570-580.

Cafaro, D. C. & Cerdà, J. (2008). Dynamic scheduling of multiproduct pipelines with multiple delivery due dates. *Computers and Chemical Engineering*, 32, 728-753.

Entrup, M. L., Günther, H. O., van Beek, P., Grunow, M., & Seiler, T. (2005). Mixed-Integer Linear Programming approaches to shelf-life-integrated planning and scheduling in yoghurt production. *International Journal of Production Research*, 43 (23), 2071-5100.

Göthe-Lundgren, M., Lundgren, J. T., & Persson, J. A. (2002). An optimization model for refinery production scheduling. *International Journal of Production Economics*, 78, 255-270.

Ierapetritou, M. G., Hené, T. S., & Floudas, C. A. (1999). Effective Continuous-Time Formulation for Short-Term Scheduling. 3. Multiple Intermediate Due Dates. *Industrial and Engineering Chemistry Research*, 38, 3446-3461.

van Kasteren, J. (2013, September 10). 'Productie van voedsel moet efficiënter'. *Trouw.* `https://www.trouw.nl/nieuws/productie-van-voedsel-moet-efficienter~b27d23c6/`. Accessed on October 16, 2019.

Kondili, E., Pantelides, C. C., & Sargent, R. W. H. (1993). A General Algorithm for Short-Term Scheduling of Batch Operations - I. MILP Formulation. *Computers and Chemical Engineering*, 17 (2), 211-227.

Lee, H., Pinto, J. M., Grossmann, I. E., & Park, S. (1996). Mixed-Integer Linear Programming Model for Refinery Short-Term Scheduling of Crude Oil Unloading with Inventory Management. *Industrial and Engineering Chemistry Research*, 35, 1630-1641.

Lin, X., Floudas, C. A., Modi, S., & Juhasz, N. M. (2002). Continuous-Time Optimization Approach for Medium-Range Production Scheduling of a Multiproduct Batch Plant. *Industrial and Engineering Chemistry Research*, 41, 3884-3906.

Magatão, L., Arruda, L. V. R., & Neves Jr., F. (2004). A mixed integer programming approach for scheduling commodities in a pipeline. *Computers and Chemical Engineering*, 28, 171-185.

Maravelias, C. T. (2012). General Framework and Modeling Approach Classification for Chemical Production Scheduling. *American Institute of Chemical Engineers AIChE J*, 58, 1812-1828.

Méndez, C. A., Cerdà, J., Grossmann, I. E., Harjunkoski, I., & Fahl, M. (2006a). State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Computers and*

Chemical Engineering, 30, 913-946.

Moro, L. F. L. & Pinto, J. M. (2004). Mixed-Integer Programming approach for Short-Term Crude Oil Scheduling. *Industrial and Engineering Chemistry Research*, 43, 85-94.

United Nations, Department of Economic and Social Affairs, Population Division (2019). *World Population Prospects 2019*. `https://population.un.org/wpp/DataQuery/`. Accessed on October 16, 2019.

Appendices

A. Nomenclature

Parameters

i	Standard subscript for processing tasks; $i \in I_j$.
j	Standard subscript for equipment units; $j \in K_i$.
k	Standard subscript for a family of tasks; $k \in NF_j$.
l	Standard subscript for a family of tasks; $l \in NF_j$, $l \neq k$.
t	Standard subscript for absolute time. Relative to the start of the horizon; $t \in H$.
s	Standard subscript for states.
H	Number of time intervals. The length of the time interval is taken to be the highest common factor of the processing times involved in the problem.

Decision variables

B_{ijt}	Amount of material which starts undergoing task i in unit j at the beginning of time period t.
BS_{st}	Equals 1 if state s is stored at the beginning of time t.
R_{st}	Amount of material of feed state s received from external sources at time t.
S_{st}	Amount of material stored in state s, at the beginning of time period t.
W_{ijt}	Equals 1 if unit j starts processing task i at the beginning of time period t.

Sets

I_j	Set of processing tasks that can be performed by unit j.
$I_j^{(k)}$	Set of processing tasks which can be performed by unit j and belong to family k.
K_i	Set of units capable of performing task i.
NF_j	Number of disjoint families of tasks on unit j.
S_{in}	Set of input states, the states that are required to start a task i.
S_{out}	Set of output states, the states that can be sold.
S_i	Set of input states of task i.
$\overline{S}_i$	Set of output states of task i.
T_s	Set of tasks requiring material from state s.
$\overline{T}_s$	Set of tasks producing material in state s.
SSL	Set of states that can be stored and are subjected to a shelf-life of sl_s time periods.

Variables

C_s	Maximum storage capacity dedicated to state s.
C_{st}	Unit cost or price of material in state s at time t.
C_{st}^S	Running cost of keeping in storage a unit of material of state s at time t.
D_{st}	Amount of material in product state s due for delivery at time t.
p_i	Processing time of task i.
p_{is}	Processing time for the output of task i to state $s \in \overline{S}_i$.
S_{st_0}	Amount of material stored in each state s, at the beginning of time period $t = t_0$.
sl_s	shelf-life time for state s.
V_{ij}^{max}/ V_{ij}^{min}	Maximum/minimum capacity of unit j when used for performing task i.
ρ_{is}	Proportion of input of task i from state $s \in S_i$.
$\overline{\rho}_{is}$	Proportion of output of task i in state $s \in \overline{S}_i$.
τ_{jkl}	Cleaning time required when a task of family l is performed after one of family k in unit j.

B. List of tasks

1. ClclNES97R_225 - [...]

2. ClcINES97R_30 - [...]
3. BucINEScs97NBu - [...]
4. SL12cINEScs97_225 - [...]
5. SL12cINEScs97_30 - [...]
6. BcINEScs97_225 - [...]
7. BcINEScs97_30 - [...]
8. SL12cINEScs97_375 - [...]
9. SL12cINEScs97_45 - [...]
10. FbINEScs97NF_225 - [...]
11. FbINEScs97NF_30 - [...]
12. BubINEScs97F - [...]
13. ClcPOcssf40_225 - [...]
14. ClcPOcssf40_30 - [...]
15. ClcPOcssf40_375 - [...]
16. ClcPOcssf40_45 - [...]
17. FcINEScs43bNF_225 - [...]
18. FcINEScs43bNF_30 - [...]
19. FcINEScs43bNF_375 - [...]
20. FcINEScs43bNF_45 - [...]
21. BcINEScs43b_225 - [...]
22. BcINEScs43b_30 - [...]
23. SL12cINEScs43b_225 - [...]
24. SL12cINEScs43b_30 - [...]
25. FbINEScs43NF_225 - [...]
26. FbINEScs43NF_30 - [...]
27. SL12cINEScs43b_375 - [...]
28. SL12cINEScs43b_45 - [...]
29. BubINEScs43F - [...]
30. ClcINEScs82R_225 - [...]
31. ClcINEScs82R_30 - [...]
32. BucINEScs82NBu - [...]
33. SL12cPOcssf40NF_225 - [...]
34. SL12cPOcssf40NF_30 - [...]
35. BcPOcssf40_225 - [...]
36. BcPOcssf40_30 - [...]
37. SL12cPOcssf40NF_375 - [...]
38. SL12cPOcssf40NF_45 - [...]
39. FbPOcssf40NF_225 - [...]
40. FbPOcssf40NF_30 - [...]
41. BubPOcssf40F - [...]
42. ClcINES33bR_225 - [...]
43. ClcINES33bR_30 - [...]
44. ClcINES33bR_375 - [...]
45. ClcINES33bR_45 - [...]
46. FcINES33bNF_225 - [...]
47. FcINES33bNF_30 - [...]
48. FcINES33bNF_375 - [...]
49. FcINES33bNF_45 - [...]

C. List of states

1. POcs
2. cINEScs97NBu, Product family 2
3. cINEScs97, Product family 2
4. bINEScs97NF, Product family 2
5. bINEScs97F, Product family 2
6. bINEScs97, Product family 2
7. cPOcssf40
8. cINEScs43bNF, Product family 1
9. cINEScs43b, Product family 1
10. bINEScs43NF, Product family 1
11. bINEScs43F, Product family 1
12. bINEScs43, Product family 1
13. cPOcssf35
14. nPKcs
15. cINEScs82NBu, Product family 0
16. cINEScs82, Product family 0
17. bPOcssf40NF, Product family 1
18. bPOcssf40F, Product family 1
19. bPOcssf40, Product family 1
20. PK39
21. cINES33bNF, Product family 0
22. cINEs33b, Product family 0

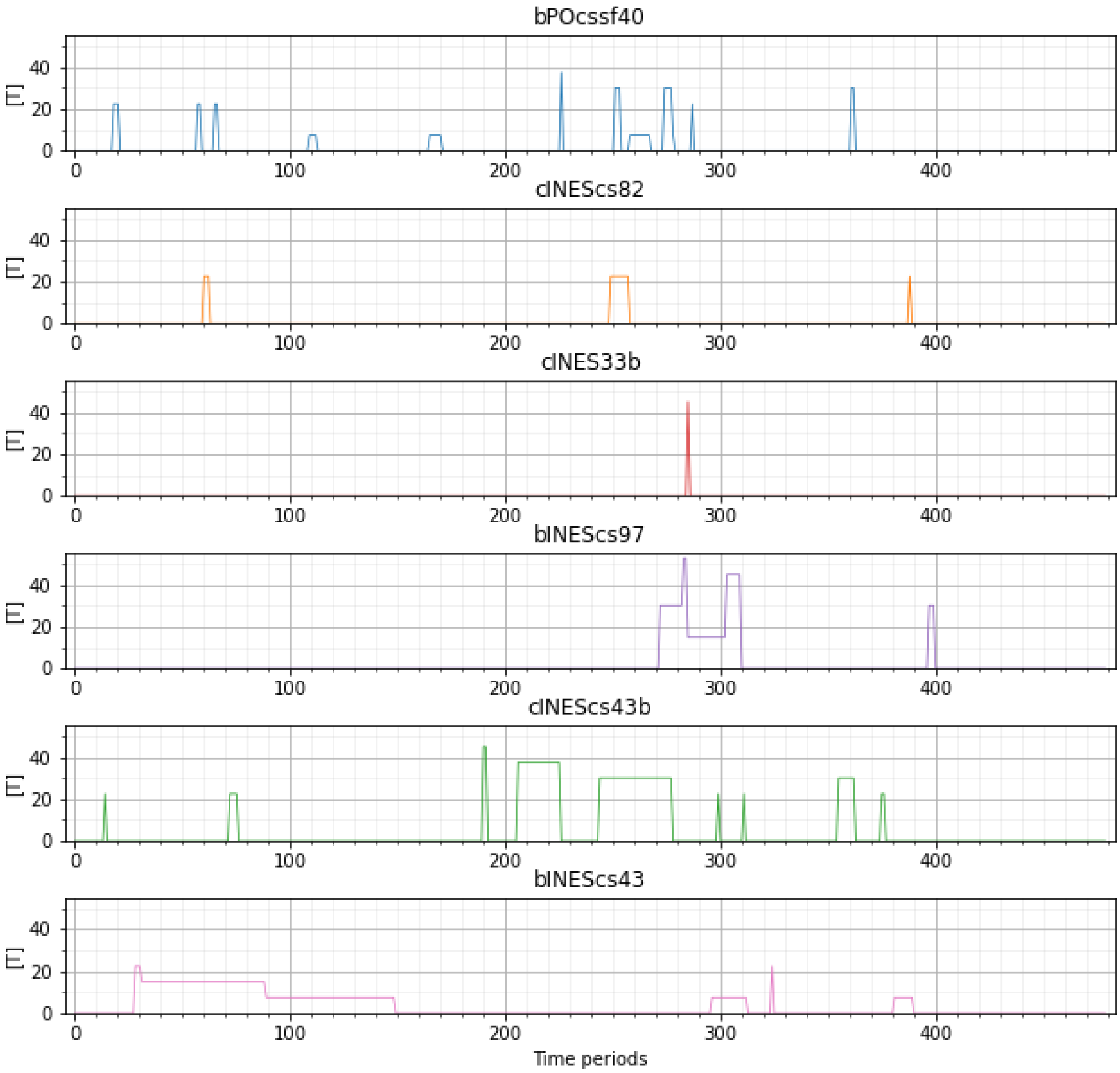

Figure F.7: Storage level per state that can be stored, with state names.