Tankpark scheduling optimisation

P.D. Baart

Technische Universiteit Delft

Tankpark scheduling optimisation

by

P.D. Baart

Master Thesis

to obtain the degree of
Master of Science
in Mechanical Engineering

at the Department Transport Engineering ans Logistics of Faculty Mechanical, Maritime and Materials Engineering of Delft University of Technology,
to be defended publicly on Monday May 18th, 2020 at 14:00 PM.

Student number:	4236696
MSc track:	Transport Engineering and Logistics
Report number:	2020.MME.8419

Thesis committee:	Dr. Ir. D. L. Schott,	TU Delft, committee Chair, 3mE
	Dr. Ir. X. Jiang,	TU Delft, committee member, 3mE
	Dr. A.A. Kana,	TU Delft, committee member, 3mE
	MSc. W. van der Tempel,	Sime Darby Oils Zwijndrecht
Date:	May 4, 2020	

Preface

In this thesis a method for scheduling storage and mixing tanks for a refinery is developed. This is done in cooperation with Sime Darby Oils Zwijndrecht, where the tanks are currently scheduled by hand. In this research an existing method was adapted to fit all the constraints of the tank park at Sime Darby Oils Zwijndrecht, This method was verified and validated, proving it to be an improvement on the current scheduling method. Due to calculation power constraints the real life application of this method cannot be done just yet, but once more power is available all tanks can be schedule more effectively, minimising material losses and increasing the effective capacity.

I would like to thank my supervisors, Drs. Jiang and Mr. van der Tempel for their support and advice during this project, my girlfriend, friends and family for motivating me, and especially thank Aytaç Balci, without who I never could have completed this research.

P.D. Baart
Delft, May 4, 2020

Summary

This thesis is done in cooperation with Sime Darby Oils Zwijndrecht (SDOZ). SDOZ is a refinery of vegetable based oils. The goal of this thesis is to optimize the scheduling of the Refined Oil Storage (ROS) of the refinery. This is done because within the company it is suspected that the current way of assigning the tanks, that is reactively by hand, is severely limiting the amount of usable free storage volume within the ROS. By increasing the amount of usable volume within the ROS, the production in the refinery, and therefore its profits, could be increased. This thesis starts out by analysing the exact scheduling problem at SDOZ. This information was then used to compare multiple scheduling optimisation methods from the literature in order to find the best applicable method. This was "A novel network-based continuous-time representation for process scheduling". The report continues by verifying this method, by reproducing example cases included in the literature. During these reproductions some issues were found, which were resolved by making modifications to the method. The next step was to improve the method to better suit the method to be used to schedule the SDOZ ROS. This was done by creating additions, which included the use of loading spots, the use of preloaders, and the need for quality checks. The complete method, including all features of the literature, and the additions made in this thesis, was verified after which it was validated. In the validation the developed method was compared to the current situation at SDOZ. This comparison showed that using the method developed in this thesis can improve the amount of usable free storage volume within the ROS significantly. It is therefore recommended for the company to implement the developed method. In this thesis the first steps needed in order to apply the method to the ROS of SDOZ are taken. In future research this could be used to completely optimise the scheduling of the ROS, something that was due to calculation power limitations, not yet possible in this research.

Contents

1
Introduction

1.1. The context of this research

This research concerns a facility of Sime Darby Plantations. Sime Darby Plantations is an international player in the vegetable based oil industry, focusing mainly on palm oils where it is the second largest manufacturer worldwide. Palm Oil and Palm Kernel oil, often indicated by PO and PK, are made mainly of the fruit of the Elaeis guineensis Jacq., more commonly known as the palm tree. The mesocarp of the fruit is used to make palm oil, and the harder kernel inside the fruit is used for the production of palm kernel oil. The trees are mainly grown in tropical climates and take four years to start producing fruits. After 30 years the amount of fruit produced decreases significantly. This is why, while the trees can get up to 200 years old, most trees are felled at 30. The flowers, of the palm tree grow in so called inflorescence. When these are pollinated they grow into fruit bunches, as shown in Figure 1.1.

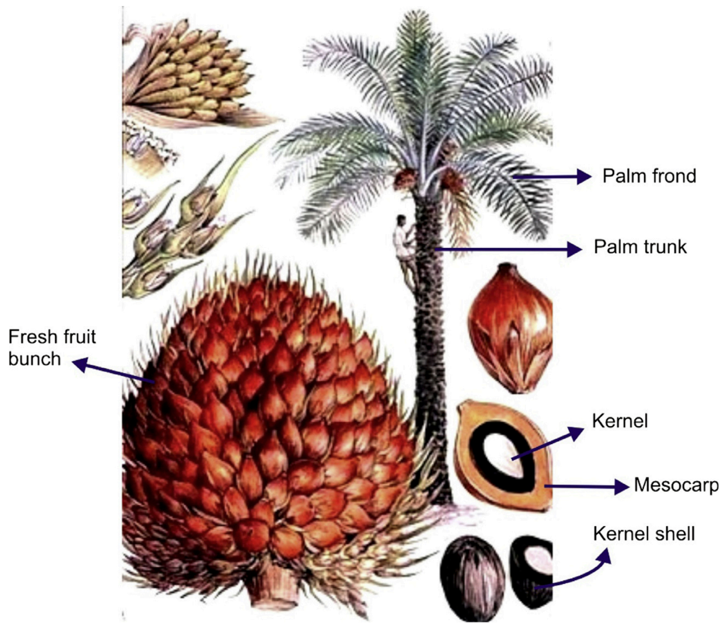


Figure 1.1: The anatomy of the oil palm and its fruit[1]

When ripe the fruit bunches are harvested and transported to mills. Contrary to other vegetable oils, such as sunflower of rapeseed oils, the palm fruit as to be processed as soon as possible since it spoils fairly quickly. After the milling the palm oil and palm kernel oil are transported to refineries to create a wide range of products. [3] [30] [35] [36]

In the last couple of years the usage of palm oil and its derivatives has grown steadily, as can be seen in Figure 1.2. The rise in use of palm oils was accompanied by a rise in concerns about their

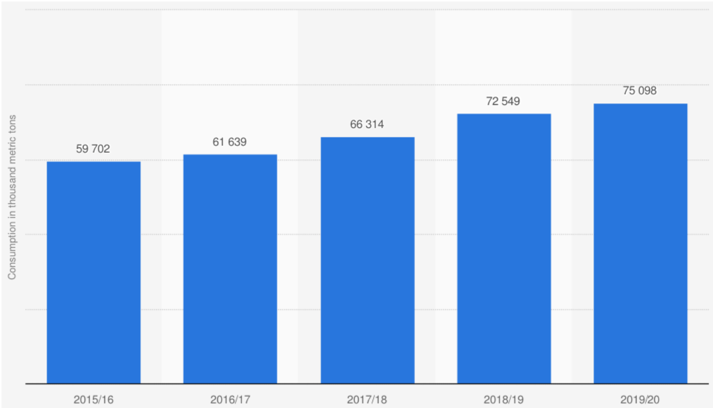


Figure 1.2: Palm oil consumption worldwide from 2015/2016 to 2019/2020 (in 1,000 metric tons)[6]

environmental impact. Since palm trees need a tropical climate to grow, and plantations use a lot of ground, rainforests are often burned down to make place for new plantations. In 2007 most of the decline of the rainforests in Malaysia and Indonesia, the two countries producing over 90% of the worlds palm supply, was linked to palm plantations. [26] This has large effects on biodiversity and the survival of endangered species such as the orangutan and has caused for the public opinion to shift against the use of palm oil. This has gone so far that companies are actively campaigning against the use of palm oils in their ads. An example of this is British supermarket chain Iceland, of which the use of the short film "Rang Tan" made by Greenpeace, was wildly debated and not allowed to air due being too political. [20]. A more recent example could be found in the Netherlands where margarine company The Flower Farm uses the lack of palm oil in its products as its main selling point, using "Eat plants not palm" as its slogan [34]. However, while Palm oil has a bad name it is actually the best crop to supply oils. As Figure 1.3 shows, palm trees produce almost five times as much oil per hectare than any other crop.

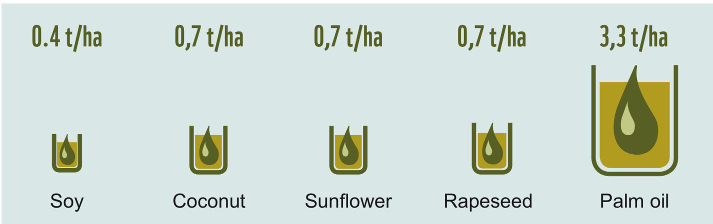


Figure 1.3: Comparison of global oil yields by crop plant: oil yields in tonnes per hectare (t/ha))[27]

The main problem of using palm trees for oil is in the way it is grown. When palm trees are grown in an environmental safe way, it has few downsides. [27] Sime Darby recognises its task as one of the leading palm oil producers in the world to lead the way to a sustainable future and has set its mission to "Making a sustainable future real for everyone". To do this the Roundtable on Sustainable Palm Oil, or RSPO was co-founded by Sime Darby. While the demand for palm oil has risen over the years, the price did not follow, as can be seen in Figure 1.4.

Since sustainable plantations are more expensive to use, Sime Darby has to compensate by reducing its operating costs. To do that Sime Darby started the Play-to-Win program in 2018 in order to increase the profitability, or in some cases bring back profitability, in its sites around the world. It is in this perspective the location in Zwijndrecht, Sime Darby Oils Zwijndrecht, is analysed.

Sime Darby Oils Zwijndrecht, or SDOZ previously known as Sime Darby Unimills, is a processor of vegetable based oils. The company started as a soap-factory in 1912, named Van den Bergh's Zeepfabrieken. In 1927 the company fused with the neighbouring Jurgen's Olie- en veekoekenfabrieken to create the Maatschappij der Vereenigde Oliefabrieken. In 1970 the name was changed to Unimills, which in turn would be changed to Sime Darby Unimills in 2010 when Sime Darby bought the company. Finally the name was changed to Sime Darby Oils Zwijndrecht in 2019. The refinery can make over 400 different products, mainly blends of different types of oils. The greatest part of the output of

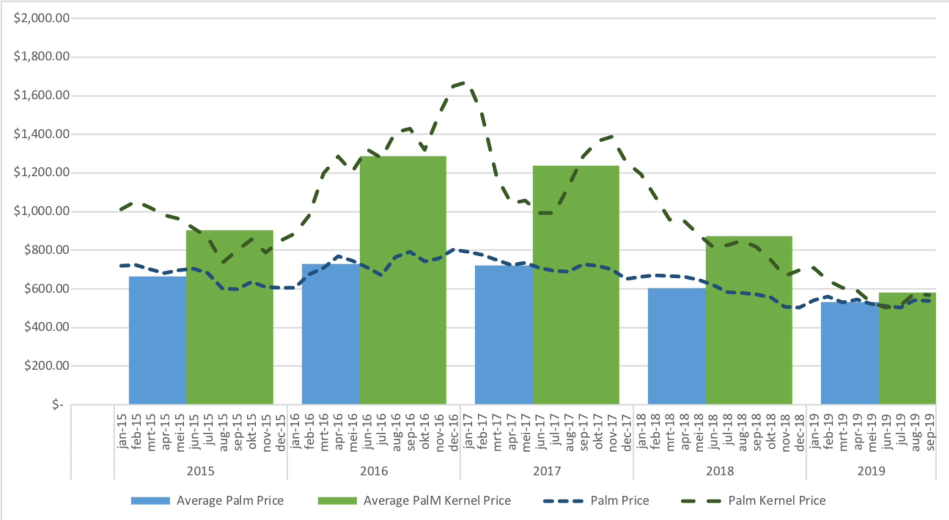


Figure 1.4: Price of Palm oil in U.S. Dollar per metric ton [2][37]

the refinery is destined for the food industry (for example as a basis for margarine) but also candles and animal feed are made with products from SDOZ. Over the last couple of years the profits of the Zwijndrecht refinery dwindled, eventually resulting in the company losing money on operation. To restore the former profitable situation Sime Darby has embarked on a transformation. A preliminary internal research has shown that an increase of 30% of the OEE (Overall Equipment Efficiency) has to be made in order to make the company profitable again. It is thought that a large part of the loss in OEE is caused by infeasible schedules and processes having to wait for resources, pipelines or storage to be available. This project aims to improve the schedules and therefore the OEE of Sime Darby Oils Zwijndrecht.

1.2. Problem definition

The production schedule of SDOZ is made weekly based on the orders of customers. The challenge in making this schedule is that it not only has to account for the different possible batch sizes within the refinery, but also has to take into account the risk of contamination between batches. Since SDOZ is producing materials mainly for foodstuffs, contamination is a serious concern. Furthermore when scheduling the refinery there also has to be accounted for the maintenance, which can require (partial) shutdowns of machines. On top of all this the customers are able to change their orders during the production week, which, along with issues that can occur during production within the refinery, calls for frequent rescheduling. Currently the planning of the refinery is done manually by employees of SDOZ. While these employees have many years of experiences they are human. This means that not only their schedules will not always be optimal, but also a great dependency is created. The complexity of the refinery makes it impossible to schedule all the pipelines between machines and tanks by hand. Therefore the planning of the pipelines is currently omitted. Problems occur when two processes need to use the same pipeline, pump or valve at the same time. When this occurs operators will deviate from the schedule, which can result in delays and rescheduling. To minimize the delays and the extra costs associated, the company asks for a planning tool which can create optimal schedules which account for everything the current schedules do, plus allowing for the use of the pipelines.

1.2.1. Stakeholders

In order to accurately pinpoint where the main problems in the scheduling process are located, a stakeholder research is done. The complete production process, starting with the costumers orders and the purchase of raw materials, and ending with delivery to the customers, includes multiple stakeholders. An overview of this process and its stakeholders can be seen in Figure 1.5.

In this subsection the most important stakeholders are highlighted and their combined requests and demands are listed. The complete takeaways from each stakeholder meeting can be found in Appendix B. The first stakeholder is the planning department. Since the planners are the ones who will be working

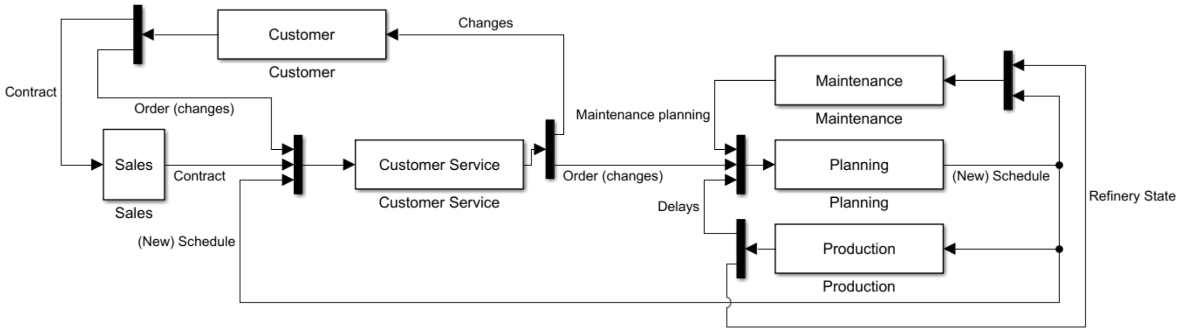


Figure 1.5: An overview of the planning process and its stakeholders

with the finished product it is vital that their priorities are included. Another important stakeholder is the production department. These are the people that will have to execute the created schedules. The next stakeholder is the sales department, which represents the customers needs, weighed against those of the company. The customer service department has direct contact with the customers during the week and communicates changes in the delivery times. This makes the customer service department a stakeholder. The maintenance department, which needs scheduled downtime to work on the production units, is also a stakeholder. Finally the quality assurance department, or QA, also has a stake in the scheduling as this can have possible influences on product quality. After interviewing the different stakeholders the requests and demands shown in Tables 1.1 and 1.2 were found.

Requests	Main Stakeholder(s)
The production schedule should include the pipelines within the refinery	Planning & Production
Rescheduling should be done automatically and faster than the current situation (2 hours)	Planning
The quality of intermediate products should be taken into account, as this has influence on the processing times.	QA & Planning
To be able to confirm orders as fast as possible by having a continuous planning	Customer Service
The scheduling program should have integration into the process control software, changing the role of the process operators to process supervisors	Planning
Materials can not be stored for long periods of time	QA
Maintain FIFO principle for materials	QA
Customers are allowed to change their orders in size and delivery time during the week	Sales & Customer Service
Include all production lines and facilities into the planning	Production

Table 1.1: An overview of the requests of the stakeholders

Demands	Main Stakeholder(s)
Schedules should be feasible	Production
Production downtime should be scheduled for maintenance	Maintenance
Products cannot overstay in the deodorizer lines	QA
The schedule should determine which storage tank is used for both inputs as outputs	Planning & QA

Table 1.2: An overview of the demands the stakeholders

To accurately focus this project, it is important to first find bottlenecks in the current system. During the meetings with the stakeholders, multiple bottlenecks were mentioned. One of the main bottlenecks was the fact that the piping is not included in the scheduling of the refinery. This can cause problems when two or more processes are scheduled to use the same piping. However, to include the piping in the schedule would mean that exact knowledge of the process end times is needed. While the end times of the processes are roughly scheduled, exact times, as needed for a pipeline planning, are not available. This is because the production process is not reliable enough and because the quality of the intermediate products is not accounted for in the scheduling.

Another often mentioned bottleneck by the stakeholders was the limited capacity of the Refined Oil Storage (ROS). According to multiple of the stakeholders the limited capacity means that the ROS is often fully occupied, halting production until space is freed. Others however claim that the problem does not lie with the amount of available storage space, but in the way this is used. In the next section this apparent contradiction is further analysed.

1.2.2. The Refined Oils Storage (ROS)

Currently the tanks in the ROS are not scheduled, which is to say that when the refinery finishes producing a product, the operator decides which tank is going to be used for storage. When loading the material from the tanks to the transport, another operator selects which tank is used for outputting the material. Since there is no schedule for which tanks have to be used, this can result in leftovers occupying tanks, or even spoiled product. The bottleneck at the ROS has been analysed before. This research showed that while, as some of the stakeholders suggested, at most times almost all tanks are in use, the total volume storage is on average 40%, as can be seen in Figure 1.6.

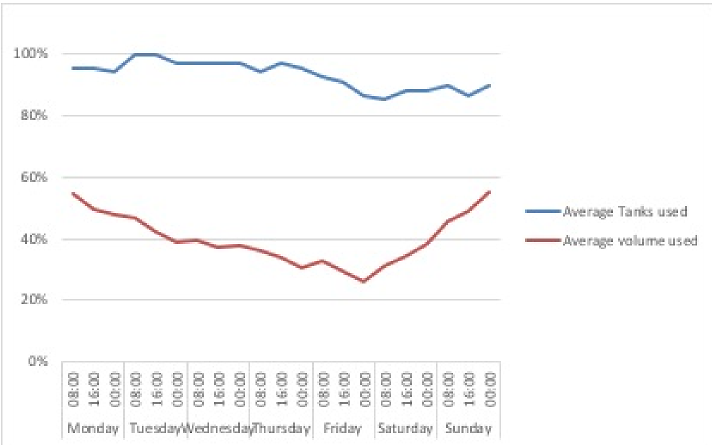


Figure 1.6: Average volume and average percentage of tanks used during the week [38]

This average could be a symptom of storing many different products in small volume, each occupying a single tank. However research also showed that a maximum of 47 unique products are stored in the 67 tanks. This can clearly be seen in Figure 1.7 [38]. It has to be noted that since this analysis was done, more storage tanks were build. The current number of tanks is 85.

Most of the time, products occupy multiple tanks simply because there is not enough room in a single tank, other times tanks are kept occupied due to poor planning. An example of this can be seen in Figure 1.8a, four tanks, each capable of holding 20mT, are occupied holding a total of 52.93mT RP (Rapeseed oil). By better planning which tank is used for loading and unloading, the situation of Figure 1.8b could be created.

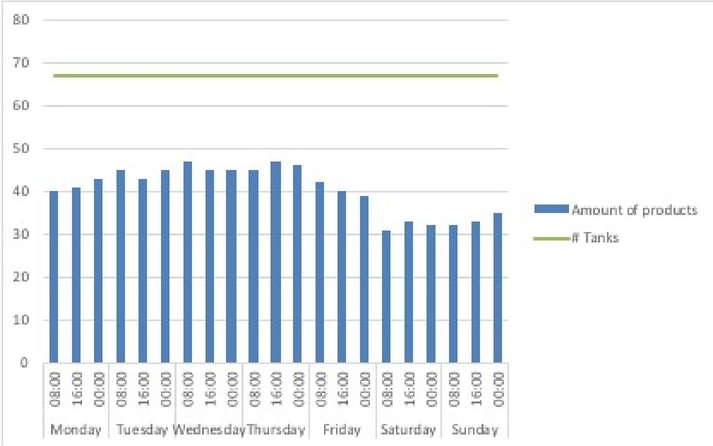


Figure 1.7: Number of unique products vs number of tanks [38]

	Tank nr	Batch nr	Material	Tank level	Max tank level
Group 7C	32	2010 10658053	RP	9.78	20
	51	2010 10658053	RP	19.2	20
	52	2010 10658053	RP	4.8	20
	53	2010 10658053	RP	19.15	20

(a) Current tank planning

	Tank nr	Batch nr	Material	Tank level	Max tank level
Group 7C	32	2010 10658053	RP	14.58	20
	51	2010 10658053	RP	19.2	20
	52			0	20
	53	2010 10658053	RP	19.15	20

(b) Ideal tank planning

Figure 1.8: Distribution of Rapeseed oil over tanks 32, 51, 52 and 53

This would create an extra 20mT of usable storage space. When a quick approximation is made using the script in Appendix C it is found that on average 6 extra tanks can be planned free, as can be seen in Figure 1.9.

Efficiently scheduling these storage tanks should increase the amount of buffer that can be built up, decreasing delays. This buffer should work both ways: in order to account for production delays, the product should be produced long enough before the pick-up date, secondly, in order to account for delayed transport, space should always be kept free to prevent over-filling of the storage. This would result in product being destroyed, and monetary loss.

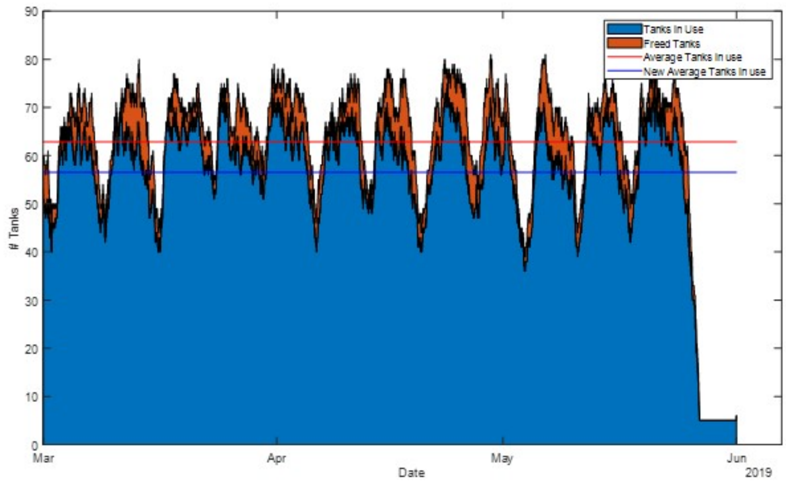


Figure 1.9: Estimate of free-able tanks

1.3. Goal

The goal of this project is to increase the profitability of SDOZ, by creating a model which can optimise the schedule for the storage tanks, pipelines, loading places, and pre-loaders, while adhering to the constraints of the facility. By optimising the schedule for the storage tanks, more effective storage volume will be available, allowing for more production and more profits. Optimising the schedule should also minimise the amount of quality checks needed, as these are costly to perform.

1.4. Research questions

In order to fulfill the goal stated in the previous section, the following research question was formulated.

How can the SDOZ ROS schedule be optimized using a mathematical model?

In answering this question a solution to the scheduling problem of SDOZ should be found. Several subquestions have been formulated in order to analyze the different parts of the main question. These are:

- How is the ROS currently scheduled?
- What are the available modelling methods in the literature?
- Which modelling method is best suited to model the SDOZ ROS?
- Which adaptations are necessary to accurately model the SDOZ ROS?
- How can the optimization model be validated?

By answering each of these subquestions, it should be possible to the answer the main research question.

1.5. Scope

The scope of this project will include the processes starting at the inflow of the ROS, and ending with the transportation of finished materials. This includes tank assignment, deciding which products should be pre-loaded, and the scheduling of the loading places in the loading hall. An overview of the scope can be seen in Figure 1.10.

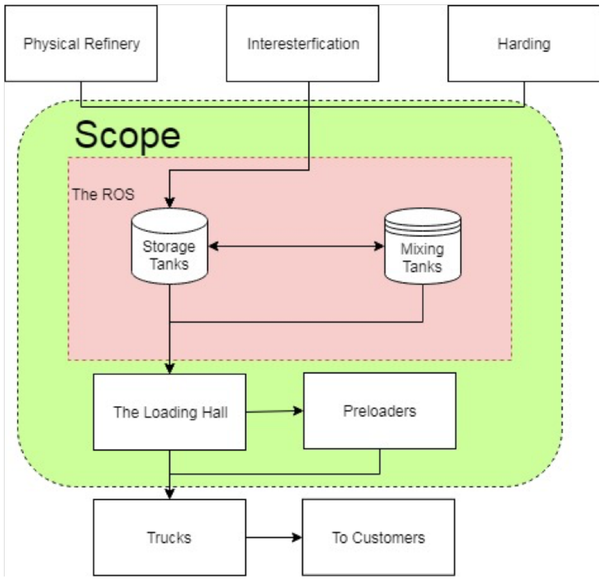


Figure 1.10: Overview of the scope of this project

To accurately scope this project not only physical limitations have to be taken into account but also some other assumptions have to be made. For this project the following assumptions are made:

- Strict schedule adherence
- Transport on time
- First-Time-Right production
- Linear processing and transport times
- Production as needed and on time

The first assumption that was made is that the company will strictly adhere to the schedules made. This includes batch sizes and timings, and also tank assignments of materials. The second assumption is that all transport will arrive on time. While this is naturally not the case, this can partially be done by creating agreements with the customers and transporting companies, and partly by expending the amount of available pre-loading trailers on site. These are both projects which are now being executed by the company. The third assumption is that all products are produced "First-Time-Right", meaning that all quality checks will pass and no product has to be remade. In situations where this is not the case, the scheduling model should be able to reschedule, taking into account processes that have already finished or have already started. The next assumption is that the processing and transport time of all batches is linearly related to their batch sizes. If this is not the case an approximation has to be made. The final assumption is that the refinery will be able to produce as needed and on time. While the schedule should be robust enough to account for small delays, rescheduling will be needed when there are more time-consuming problems. [33]

1.6. Approach and Structure

The approach of this project will be as follows: First an overview of the refinery is created by interviewing employees and observing normal operations. The results of this can be read in section 1.2. After this a literature study will be done to find existing modeling methods and gather information on how they are applied. This will comprise the first part of Chapter 2. In the second part the found information will be applied to the SDOZ refinery in order to find what modeling demands follow out of the system. These demands will then be combined with the information about modeling methods to select a suitable method. This method will then be explained and verified in Chapter 3. After the existing method is verified improvements will be made in order to make the model better fit the situation at SDOZ. This will be done in Chapter 4, where the complete developed method will also be verified. In Chapter 5 The developed method will be validated and the real life application of the method will be discussed. The report will end with its conclusions, the contribution of this thesis to both the academics and the company, the limitations of this research, and recommendations for further research, which can be found in Chapter 6. In Figure 1.11 a overview of the structure of this report is given.

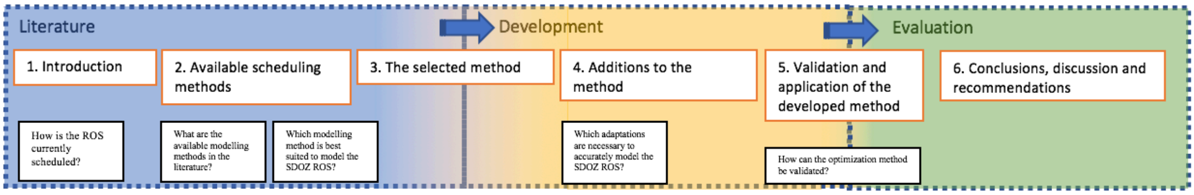


Figure 1.11: The approach of this thesis

1.7. Summary

In this chapter the project, its context, and its research questions were introduced. The chapter started by discussing the company, Sime Darby Oils Zwijndrecht (SDOZ), and its main product, palm oil. After this it was discussed that the OEE of the company needs to be increased and to do this the schedules within the refinery need to be improved. A stakeholder analysis was done, after which it was concluded that the most gain could be had in improving the Refined Oil Storage (ROS), as this is currently not scheduled and not used effectively. Therefore the goal of this project was set to improve the OEE of the ROS by effectively scheduling the tanks. The scope was set to encompass the ROS and the loading hall to which it is connected. Research questions were defined to aim the project, and the approach and structure of this report were explained. In the next chapter the problem will be further characterised and a suitable modelling method will be found.

2

Available scheduling methods

In order to effectively schedule the ROS of SDOZ a optimisation model has to be made. This is done by using one of the many scheduling optimisation modelling methods. The mathematical optimization of production schedules is not a new subject but had been analyzed since as early as 1960[32]. Since then many different authors have suggested as many (or even more) different modelling methods to tackle these problems. [15] gives an overview of some of the available scheduling modeling methods. These methods formulate the scheduling problems as either an Mixed Integer Linear Programming (MILP) or a Mixed Integer Non-Linear Programming (MINLP) problem. Which type is used is partially linked to the way time is handled in the resulting model. Section 2.1 explains the different problem formulations, Section 2.2 will further explain how time is modelled by different methods. While both the (non-)linearity and the time representation have great influence on the performance of the model, they are often not deciding factors but a consequence of the characteristics of the system to be analysed.

While it is not likely to find an existing method that exactly matches the needs of the system to be analysed, modifications can often be made to tailor a method that is close. It has to be noted that some modifications require more work than others.

In order to know which method is the best suitable, the system that is to be scheduled should be characterised. This is done in Section 2.3. The information gathered in this section is then used to characterize the system and make a list of demands for the optimisation method in Section 2.4. This list of demands is then used in Section 2.5 to compare different existing methods to find the one best suited. This method will then shortly be explained in Section 2.6.

2.1. Types of optimisation problems

The problem to be analysed in this project is a scheduling problem, which falls in the category of mixed integer programming problems. As the name suggests these are problems of which the solution is a mix of integers and real numbers. Some of the variables can be a real number, such as the amount of product in a tank: it is possible to have 19.3 tonnes of material in a tank. Other variables can only be integers, or in some cases only be binary: A variable indicating if the tank is in use can only be 1 (tank is in use), or 0 (tank is not in use). A value of 0.5 would not be realistic, as this would indicate that the tank is neither in use nor not in use[1].

Within the mixed integer programming problems a distinction is made between linear problems (MILP) and non-linear problems (MINLP). A standard representation of a MI(N)LP can be found in Equations 2.1

[1]A possible exception would be monitoring alive cats in boxes, but that is a problem for the quantum physicists

$$\min_x f(x) \quad (2.1a)$$

$$s.t. \qquad \text{where:} \quad x = \begin{bmatrix} x_r \\ x_i \end{bmatrix}, \quad x_r \in \mathbb{R}^{n_r}, x_i \in \mathbb{Z}^{n_i}$$

$$h(x) = 0 \quad (2.1b)$$

$$g(x) \leq 0 \quad (2.1c)$$

In these equations function $f(x)$ indicates the objective function of the model. This quantifies how good a solution is. Most models are made as a minimization model, meaning that the solution with the lowest function value of $f(x)$ is the optimal solution. Functions $h(x)$ and $g(x)$ indicate respectively the equality constraints and the inequality constraints. A solution can only be valid if all constraints are satisfied. While an optimisation model can have only one objective function, there is no limit on the amount of constraints used. If functions $f(x)$, $h(x)$, and $g(x)$ are all linear the model is a MILP, otherwise it is a MINLP. Generally Linear systems are easier to solve and therefore preferred. However, since reality is not always linear often simplifications or linearisations have to be made. This can make models less accurate. In other cases the extra constraints and/or variables needed for the linearisation can increase the calculation time. An example of this is the use of switching variables. These are binary variables that are either switched on or switched off based on a continuous variable. For example a binary variable monitoring if a tank is in use (S) based on the amount of material stored in that tank (IS). A non linear model could match these variables by using constraint equation 2.2.

$$IS \times (1 - S) = 0 \qquad (2.2)$$

This ensures that the switch is turned on if there is any material stored in the tank. Linear systems can not use this constraint as variables are multiplied with each other. A linear system would therefore make use of constraint equation 2.3.

$$IS - (M \times S) \leq 0 \qquad (2.3)$$

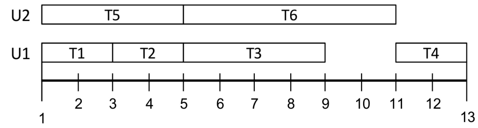
This is a so called big M equation, where a constant M with a large value is used to match the variables. A disadvantage of this method is that if M is to small, the constraint will not work at larger values of IS, but if the value is to high the value of S might be small enough to fall within the integer tolerance. If for example the M was set to 10^4, equation 2.3 would work as long as the tank is filled with less than 10000kg of material. If more material was to be stored the constraint cannot be satisfied, resulting in no feasible solutions. Most systems use a integer tolerance of 10^{-5}. This would mean that if the M was set to 10^7 and 100 kg of material (or even less) was stored, the constraint would be able to satisfy with S being 10^{-5}. A solution to this could be decreasing the integer tolerance to be even tighter, but this dramatically increases calculation times. It is therefore vital that if big-M constraints are to be used, the value of M is chosen correctly. Furthermore, since big-M constraints have great influence on the calculation times, it is recommended to use as few as possible. When a large number of big-M constraints is needed it might even be better to switch to a non linear system altogether. [31] [5]

2.2. Time representations

Apart from the characteristics of the system one other main decision has to be made: that of the time representation within the model. The time representation greatly influences the performance of a model, in terms of quality and in terms of calculation times. Not every time representation is suitable for all types of system characteristics, therefore the decision on which time representation is used, is made after the characterisation of the system is made. There are four main ways of representing time in scheduling models: Discrete-time grid, Continuous-time with single grid, Continuoustime with multiple grids, and the precedence based model. Figure 2.1 shows all four representations.

Discrete-Time

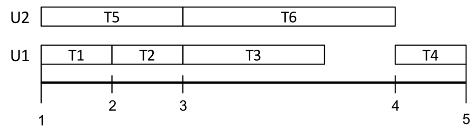
The first type of time-grid representation is global discrete time-grid representation, also known as global time intervals. In this representation the event horizon is split into a number of time-steps with a fixed length. While this length does not have to be uniform it is important that the length determination is done a priori and is not part of the optimization. One of the main advantages of global time-grids is that the synchronization between processing units is automatically done. An advantage of discrete time is ease of handling of constraints as well a good handling of intermediate storage. While this is the most straightforward way of modeling time, a lot of time steps are often needed create an optimal or even feasible solution, resulting in large models.



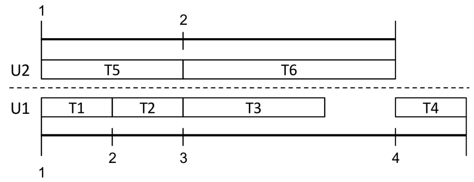
(a) Discrete-time grid

Continuous-Time

Continuous time representation uses less time points than its discrete counterpart but leaves the start and end times free for the optimization to decide. This greatly reduces the amount of binary decision variables compared tot the discrete time representation. The downside of this is that the constraints often become nonlinear and non-convex. Another disadvantage is that the number of time points greatly influences the quality of the result. The ideal number of time points has to be found with an iterative process, which may result in large CPU-times. Instead of global time-grid some optimization models make use of unit specific grids. These can come in both discrete and continuous forms. The main advantage of having multiple grids is the great reduction in the number of time points needed, the main downside is that it is harder to model material balances.



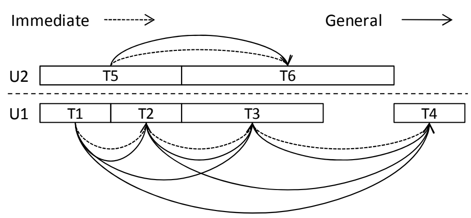
(b) Continuous time: single grid



(c) Continuous time: Multiple grids

Precedence-based

Precedence based models are a sub-type of continuous time models. These models make use of two types sequencing constraints to ensure consecutive tasks / batches are executed in the right order, one for immediate following batches and one for batches that follow later in time. Precedence based models are only applicable to sequential production environments. Batch integrity in a sequential production environment has to stay intact, that is as to say they cannot be merged or split. This is very similar to a discrete manufacturing environment.



(d) Precedence based

Figure 2.1: Different time representations [7]

Characteristic	Discrete time models	Continuous time Models					
Event representation	Global time intervals	Global time points	Unit-specific time events	Time slots*	Unit-specific immediate precedence*	Immediate precedence*	General precedence*
Main decisions	Lot-sizing, allocation, sequencing, timing				Allocation, sequencing, timing		
Key discrete variables	W_{ijt} defines if task *I* starts in unit *j* at the beginning of time interval *t*.	Ws_{in} / Wf_{in} define if task *i* starts/ends at time point *n*. $W_{inn'}$ defines if task *i* starts at time point *n* and ends at time point *n'*.	Ws_{in} /W_{in} / Wf_{in} define if task *i* starts/is active/ends at event point *n*.	W_{ijk} define if unit *j* starts task *i* at the beginning of time slot *k*.	$X_{ii'j}$ defines if batch *i* is processed right before of batch *i'* in unit *j*. XF_{ij} defines if batch *i* starts the processing sequence of unit *j*.	$X_{ii'}$ defines if batch *i* is processed right before of batch *i'*. XF_{ij} / W_{ij} defines if batch *i* starts/is assigned to unit *j*.	$X'_{ii'}$ define if batch *i* is processed before or after of batch *i'*. W_{ij} defines if batch *i* is assigned to unit *j*
Type of process	General network				Sequential		
Material balances	Network flow equations (STN or RTN)	Network flow equations (STN or RTN)	Network flow equations (STN)		Batch-oriented		
Critical modeling issues	Time interval duration, scheduling period (data dependent)	Number of time points (iteratively estimated)	Number of time events (iteratively estimated)	Number of time slots (estimated)	Number of batch tasks sharing units (lot-sizing) and units	Number of batch tasks sharing units (lot-sizing)	Number of batch tasks sharing resources (lot-sizing)
Critical problem features	Variable processing times, sequence-dependent changeovers	Intermediate due dates and raw-material supplies	Intermediate due dates and raw-material supplies	Resource limitations	Inventory, resource limitations	Inventory, resource limitations	Inventory

* Batch-oriented formulations assume that the overall problem is decomposed into the lot-sizing and the short-term scheduling issues. The lot-sizing or "batching" problem is solved first in order to determine the number and size of "batches" to be scheduled.

Table 2.1: General characteristics of current optimization models [25]

The decision which time representation is used is heavily based on the type of system that is to be modelled. As mentioned earlier a system that mixes and/or splits batches cannot use a precedence based time representation. In Table 2.1 an overview of the characteristics of different models with different time representations is given. [8] [9] [10] [15] [25]

2.3. Problem characterisation

In order to know which method is the best suitable, the system that is to be scheduled should be characterised. Different authors have used different ways of characterising systems. Many of these overlap but some are more specific then others. To accurately define the system to be analysed it is opted to use one of the more specific method for classifying systems. [25] In this method 13 key aspects of the system are analysed, as can be seen in Figure 2.2. In the following subsections each of these key aspects will be shortly discussed.

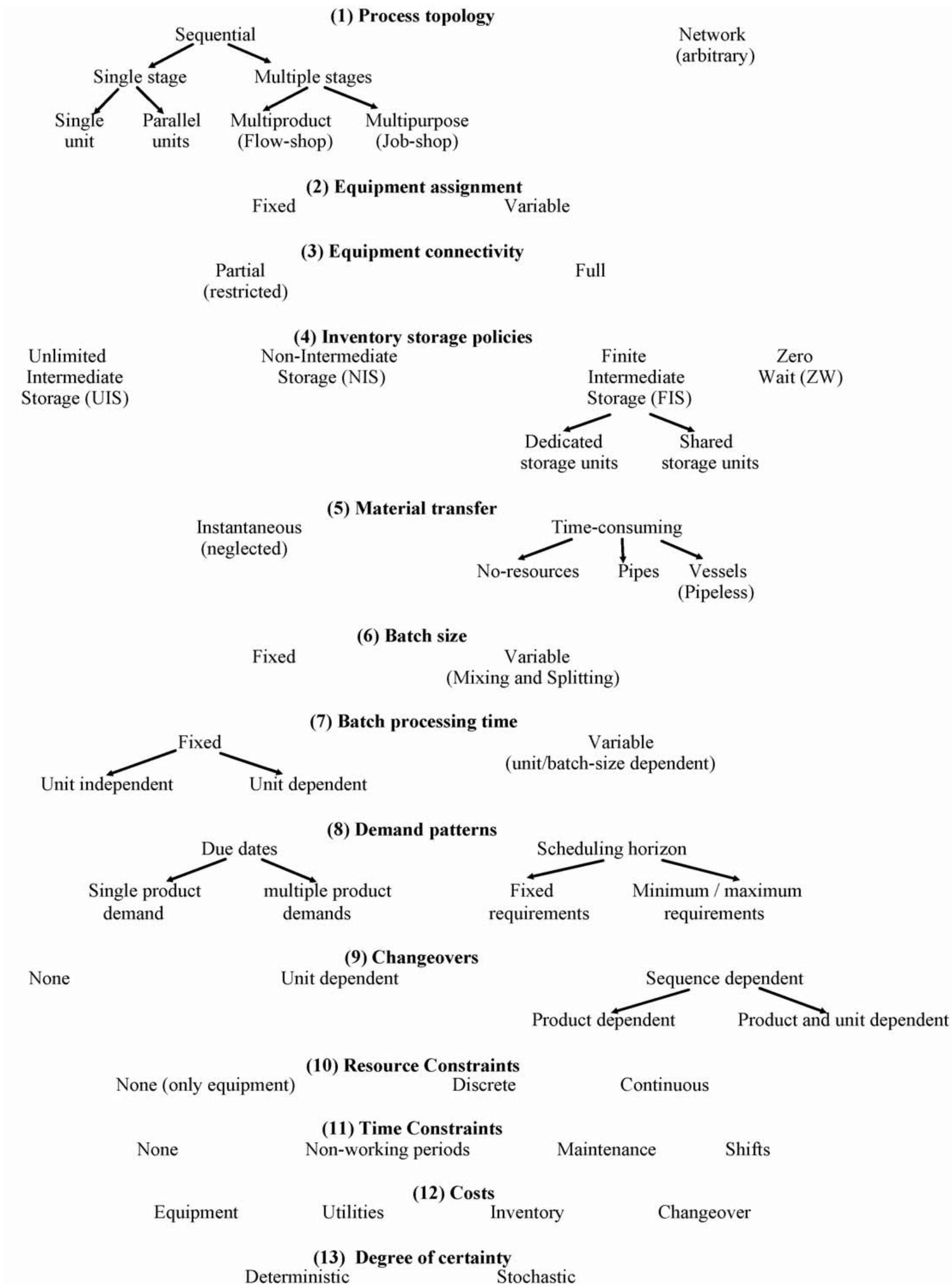
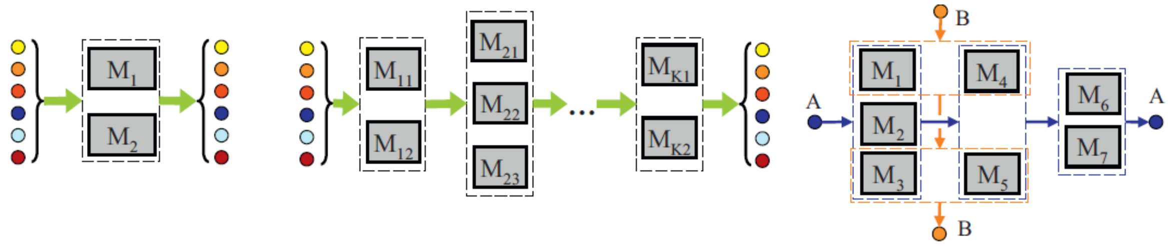


Figure 2.2: A roadmap for scheduling problems of batch plants [25]

Process topology

The first aspect, the process topology depends on how batches are moving through the system. The first split in this is that between a sequential processing environment, in which batch integrity has to stay intact, and a network processing environment, where batches can be merged and split. Within the sequential processing environments a split is made between singe-stage and multiple stage environments. In a single-stage environment all batches have only one process stage. In this process stage the batches can be handled by one and the same processing unit or be divided over a number of parallel units. Each batch has to be assigned to exactly one processing unit. Multiple stage environments can be split into multi-product, also known as flow-shop, environments or multi-purpose, also known as job-shop, environments. If all batches use the same sequence of stages the environment is called multi-product, if sequences differ between batches it is a multipurpose environment. If one or more stages have multiple processing units working parallel, the term flexible is added. An overview of the sequential production environments can be seen in Figure 2.3.

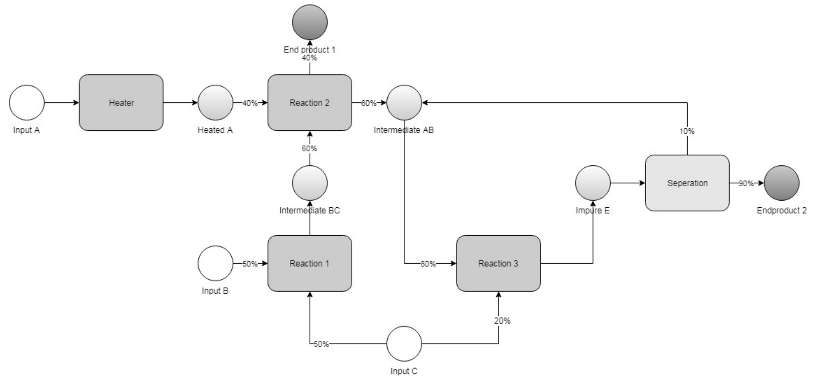


(a) Single stage (b) Multi-product (c) Multi-purpose

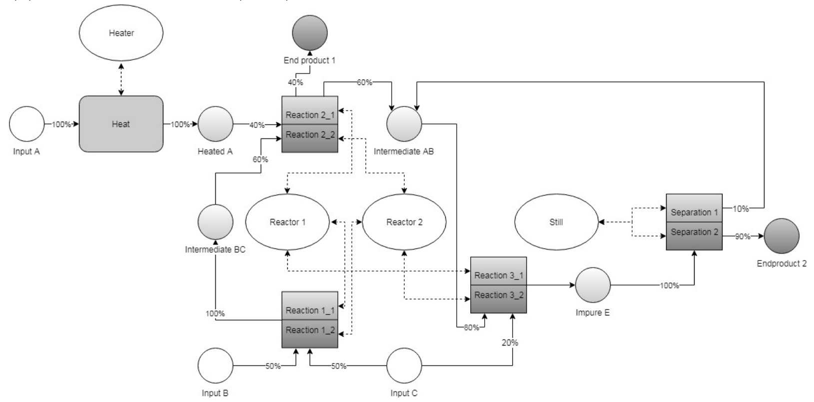
Figure 2.3: Different types of sequential processing environments [15]

In a network processing environment batches can not only move it is possible to split and mix batches. This has a great influence on the way the system can be modelled; In sequential environments the schedule is based on how batches move trough the system. In network environments this cannot be done since batches do not stay intact. The model of a network production environment has therefore to be based on the balances of the materials (and resources). [21] [15] a network modeling environment the distinction is made between State Task Networks, or STN, and Resource Task Networks, RTN. In the State Task Network there are two types of nodes: states, which are compromised by the raw, intermediate and final products, and the tasks, which process one state into another. The Resource Task Network uses resources instead of states. These resources include not only the materials but also the processing units and other resources such as steam. These can be consumed permanently, in the case of materials being processed, of temporary, in case a processing unit is occupied.

A popular example for showing the difference between a RTN and a STN representation can be seen in Figure 2.4. Both subfigures show the same simple plant consisting out two identical reactors which can each perform three reactions a heater and a still. In order to create product 1 the raw feeds of materials A and B have to be combined in reaction 1 into intermediate BC. This then has to be combined with a heated raw feed A, in reaction 2. In this process intermediate AB is created. Adding intermediate AB to raw feed C in reaction 3, creates Impure E. Separating Impure E creates product 2 and a recycled stream of intermediate AB. Figure 2.4a shows only the States (circles) of the materials and the tasks (rectangles) which are performed on them, where Figure 2.4b also shows the reactors, the heater and the still (ovals).



(a) State Task Network (STN) representation



(b) Resource Task Network (RTN) representation

Figure 2.4: STN and RTN representations of an simple processing plant [17] [15] [40] [21] [10] [11] [9] [25] [19] [2]

Equipment assignment

If a process can be only be handled by a single piece of equipment, then the assignment of equipment is fixed. However, when multiple pieces of equipment can execute the same processes assignment is variable and should be decided in the scheduling. This is for example the case when two identical processing units work parallel or when different processing units can both perform the same general task.

Equipment connectivity

The connectivity between processing units (and storage units), can create hard constraints. When not all processing units are interconnected with all storage tanks, the number of feasible solutions for a production schedule decreases significantly.

Inventory storage policies

The next key aspect of a system is how intermediate inventory storage is handled. A model can assume that the intermediate storage is unlimited. While this is of course not possible in practise this is used to decrease the amount of constraints and variables. The second storage policy assumes no intermediate storage. This will force the processing units to start processing a batch directly as it comes available. Another option is to keep processed materials in the processing unit after they have finished processing or already load them in the next processing unit without starting. This is effectively increasing the processing time, and is not always possible. The third storage policy is the most common intermediate storage: Finite intermediate storage. These are units in which intermediate products can be stored until needed in the next step of the process. Finite intermediate storage can be split into two types: Dedicated storage units and shared storage units. As the name suggest dedicated storage units are dedicated to store only one type of (intermediate) material, where shared storage units can be used (after emptying and often cleaning) by different types of materials. The last intermediate storage policy

[2]This is a very popular example

is the zero wait policy where intermediate products have to be processed as soon as they are created. This is for example the case in the production of cast iron: The molten metal has to be cast before it cools down. While some systems only use one type of intermediate storage policy, most systems use multiple, combining different policies for different intermediates.

Material transfer

How material is transferred between different storage and processing units is described by the fifth system aspect. A model can simply ignore the whole matter and assume instantaneous transport, which decreases the problem size, but this can result in unfeasible schedules. There is also the option of account for the time it takes to transport the materials but not the equipment that is used. Some systems like [28] do this by simply elongating the process times and assuming instantaneous transportation. This method does not add any constraints or variables and is therefor the fastest way of modeling this. However, this is only an option when transport times are relatively small compared to processing times, when transport times become relatively large the model quickly becomes infeasible. If the equipment should also be accounted for, for example when two processing units use the same equipment for draining their products, it has to be defined if the transport can be done continuously (by use of piping) or batch-wise (by use of transport vessels).[12]

Batch size

Whether the batch size is fixed before the schedule is made, or during the scheduling process itself, is another key aspect. Fixing the size of the batches before scheduling reduces the model size and calculation time significantly, but can also result in sub-optimal solutions. When mixing and splitting of batches is allowed the batch sizes are always variable.

Batch processing time

Related to the batch size is the processing time for the batches. This can be completely fixed, dependent on which processing unit is used, or variable. In the latter the time can be dependent on units, but also on the batch sizes.

Demand patterns

The way the product demand is handled is the eight aspect of a scheduling system. This can be done by having set due dates on which one or multiple end products should be delivered, but can also be done by having a production demand over a time horizon. In this last case a distinction is made between minimum/maximum amounts to produce and exact quantities.

Changeovers

In some cases contamination between batches of different materials can cause issues. In order to prevent these issues, materials that could contaminate each other may not be run on the same equipment directly after each other. To facilitate the changeover between contaminating materials the equipment needs to be cleaned in between. An alternative is to introduce a third batch, which can not be contaminated by the first, and will not contaminate the second batch, which should be placed between the batches. Sometimes contamination can be directional: A drop of red paint in a can of white paint will have a more severe effect then a drop of white paint in a can of red paint. When this is the case a so called contamination matrix, displaying all relations between materials, has to be made. When models account for changeovers often not only the costs associated with a changeover are considered, but also the time spent, which reduces the production capacity. [15] For example: [39] not only models the costs of the changeover, but also the costs for keeping the system set-up and the reduced capacity. Whether or not changeovers are needed is the ninth aspect of a scheduling system.

Resource Constraints

How the constraints considering resources such as steam and heat, but also employees are handled is seen as the tenth key aspect of a scheduling system. As with the changeovers a model can simply omit the resource constraints. This can be done when there is always a sufficient amount of resources, or no resources are necessary apart from the equipment, this can result in a smaller model without losing feasibility or optimality. When constraints on resources have to be accounted for, a distinction is made between discrete resource constraints, such as the number of employees, and continuous resource constraints such as the volume of steam needed.

Time Constraints

The eleventh key aspect of a scheduling system is how time constraints are handled. Time constraints are used to ensure no production is planned at certain time points. in an Ideal situation there would be none, but most cases have one ore multiple of the following constraints. Non-working periods are often nights and weekends, but also holidays and lunch breaks can be considered. Production stops for maintenance can be planned as a fixed time period in which no batches can be planned, but this is not always necessary. Many scheduling models create maintenance batches to be run on production units, This makes it possible for the maintenance to be scheduled effectively, creating minimal production and/or time loss. Another important constraint is shifts. depending on the system batches should be contained entirely in one shift, ensuring the same employees start and finish the batch. Another constraint that follows from working in shifts is that often no batches can finish or start during the shift handover, since both the "old" and the "new" shift will be preoccupied.

Costs

As with all commercial enterprises the driving force is earning money. To be able to to this costs should be as low as possible compared to the profit. While an optimal schedule in terms of make-time optimization could improve the profit, by enabling more orders to be fulfilled, or reduce costs by reducing the amount of work hours, the opposite effect could also occur. for instance, when a costs is associated with a changeover between products, for example the costs of hiring a cleaning crew, the total costs of a minimal maketime schedule which contains many changeovers, could be higher than its profits. In order to prevent such result a optimization model should account for this type of costs. These costs can include the fixed and variable cost of equipment, utilities, storing materials in an inventory an changeovers.

Degree of certainty

The last key aspect of a scheduling system is the degree of certainty. Where in reality most things like processing times are stochastic, the deviation is often small enough to assume deterministic properties. If the system contains great uncertainties however, a stochastic model has to be used.

2.4. System demands

To select the best modeling method it is important to know the demands that are set by the system. By using the roadmap in Figure 2.2, the following information is found. Since batches can be mixed and split the system cannot be modelled as sequential and has to be modelled as a network. Since the storage tank and the loading place where products are stored and handled should be a result of the scheduling, the equipment assignment is variable. Not all tanks are connected to all loading places, so the equipment connectivity can be seen as restricted. The storage tanks can be used to store multiple products, and thus have to be modeled as shared finite intermediate storage units. The material transfer will be done by pipes which is a time consuming process. The sizes of the batches that are used are variable, since it might occur that a tank truck is filled from multiple storage tanks. The processing time of each batch is dependent on the size of the batch. The demand pattern is multiple product due dates. To prevent contamination there should be product dependent changeovers considered in the schedule. Depending on the modeling of the pre-loading trailers there are either discrete resource constraints, if the pre-loading trailers are seen as a recourse, or no resource constraints, when they are seen as processing units. Maintenance has to be scheduled, which poses a time constraint. The costs to be considered are changeover costs and inventory costs. while the arrival of trucks can be seen as stochastic, this can be changed to deterministic, when a time slot is given to the customers, and a pre-loader trailer is used when this slot is not met. This policy should reduce the scheduling problem in size and increase performance. An overview of the system demands is given in Table 2.2

System aspect	System demand
Process topology	Network
Equipment assignment	Variable
Equipment connectivity	Partial
Inventory storage policy	Finite Intermediate Storage: Shared Storage units
Material transfer	Time-consuming: Pipes
Batch size	Variable
Batch processing time	batch-size dependent
Demand patterns	Due dates: multiple product demands
changeovers	Sequence dependent: Product dependent
Resource Constraints	None / Discrete [3]
Time constraints	Maintenance
Costs	Equipment changeover and inventory
Degree of certainty	Deterministic

Table 2.2: Properties of the system

2.5. Selecting the method

To find the best suitable optimisation modelling method the information from Table 2.2 was compared to the characteristics of different methods found in literature. The complete comparison can be found in Table 2.3. Each of the found methods was analysed to see if it is able to model each of the different characteristics as described by Table 2.2. If a method was able to model a characteristic without large modifications it was marked by a checkmark (✓). By simply comparing the amount of checkmarks the method that needs the fewest adjustments to model the ROS of SDOZ was found. The method most suitable for the situation at SDOZ was deemed to be that of [12].

[3]depending on how the pre-loading trailers are modelled

Demand / Method	Network topology	Variable Equipment assignment	Partial Equipment connectivity	Shared Finite Intermediate storage	Material transfer via Pipelines	Variable Batch size	Variable Batch processing times	Multiple product demands	Product dependent changeovers	Maintenance time constraints	Utility costs	Inventory costs	Changeover costs
Design, Synthesis and scheduling of multipurpose batch plants via an effective continuous-time formulation [19]	✓	✓				✓	✓				✓	✓	
A novel network-based continuous-time representation for process scheduling [12]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓
An optimization model for refinery production scheduling [14]						✓	✓	✓			✓	✓	✓
A general resource-constrained scheduling framework for multistage batch facilities with sequence-dependent changeovers [23]		✓	✓					✓	✓			✓	
An approximate mathematical framework for resource-constrained multistage batch scheduling [22]		✓	✓					✓	✓				
A CP formulation for scheduling multi product multistage batch plants [41]	✓		✓	✓					✓				
MIP-based decomposition strategies for large-scale scheduling problems in multi product multistage batch plants: A benchmark scheduling problem of the pharmaceutical industry [18]		✓						✓	✓				✓
A continuous-time MILP model for short-term scheduling of make-and-pack production processes [4]	✓	✓	✓	✓		✓		✓	✓			✓	
Batch selection, assignment and sequencing in multi-stage multi-product processes [29]		✓	✓			✓		✓	✓				
Simultaneous Lot Sizing and Scheduling of Multistage Batch Processes Handling Multiple Orders per Product [24]		✓				✓	✓	✓	✓				

Table 2.3: Models found in literature compared to the system demands

2.6. The selected method

The modeling method selected is "A novel network-based continuous-time representation for process scheduling" [12]. This method includes the use of pipelines for transport, can account for contamination and cleaning and can handle previous-horizon tasks, which is needed for rescheduling. The process topology is a network to enable the mixing and splitting of batches, something that is essential for the operating of the ROS tank park. Time is represented in a continuous way. This means that while the amount of timesteps should be set a priori, the length of each timestep will be decided by the optimisation. The selected method is a linear method, resulting in a MILP. The system is thus kept completely linear. While one might expect that big M constraints would be used, as discussed in section 2.1, this is not the case. The switching variables, as discussed in section 2.1 are handled by making use of the maximum values of the continues variables. For example the constraint matching if a tank is in storage mode, $S^S_{m,j,n}$, is matched with the amount of material stored via constraint equation 2.4

$$I^S_{m,j,n} \leq \varsigma^{MAX}_{m,j} S^S_{m,j,n} \qquad \forall m \in M^S_j, j \in J^{SS}, n > 1 \tag{2.4}$$

In this constraint equation m indicates the material in set M^S_j, the set of all materials that can be stored in unit (in this case tank) j. j in turn is part on the set J^{SS}, the set of units which can store multiple materials (tanks in which only one material can be stored do not need a variable indicating that materials are stored). n indicates the timestep. $I^S_{m,j,n}$ is therefore the amount of material m stored in unit j at timestep n. The $\varsigma^{MAX}_{m,j}$, which is used in stead of a big M, is the maximum allowed storage in the tank. Since $I^S_{m,j,n}$ can never be larger then $\varsigma^{MAX}_{m,j}$ the constraint can always be satisfied by making $S^S_{m,j,n}$ equal to 1. The literature of the method provides a set of examples which can be used for verification, these are used in sections 3.2 and 3.3 to verify the method. Further information how the selected method creates its model can be found in section 3.1.

2.7. Summary

This chapter started with explaining different types of scheduling modelling methods. After this the different characteristics of scheduling problems were discussed. To do this a roadmap, shown in Figure 2.2, was used. The information of the first two sections was then applied to the problem resulting into the third section where the modelling demands of the system of SDOZ were explained. The chapter finishes with comparing modelling techniques in the literature to the demands of the system and by choosing the method that matches the closest to the demands of the system, which is "A novel network-based continuous-timerepresentation for process scheduling" [12].

3

A novel network-based continuous-time representation for process scheduling

In this chapter the method selected in the previous chapter will be explained. The method consists of a basic structure and additional features that are implemented on this structure. For both the basic structure as for the advanced method with extra features, examples are given. In Section 3.2 the reproduction of the examples for the basic structure, and issues that occurred during this reproduction, are discussed. In Section 3.3 the same is done for the additional features given in the literature.

3.1. Structure of the method

The selected method, "A novel network-based continuous-time representation for process scheduling", is a MILP based on a continuous time model. This means that, while the amount of time steps is known beforehand, the length and the timing of the time steps is determined by the optimisation. This reduces the amount of time steps needed, and thus calculation time significantly. For the amount of timesteps an estimation can be made. This can for example be done based on the amount of shipments are taking place within the timehorizon. In the model Units are split in groups. In the first part of the method, the basics, the units are split into production units and storage units. Later, groups as cleaning and transport units are added in the additions. Initially only production tasks are considered, later cleaning, transport and quality assurance tasks are added.

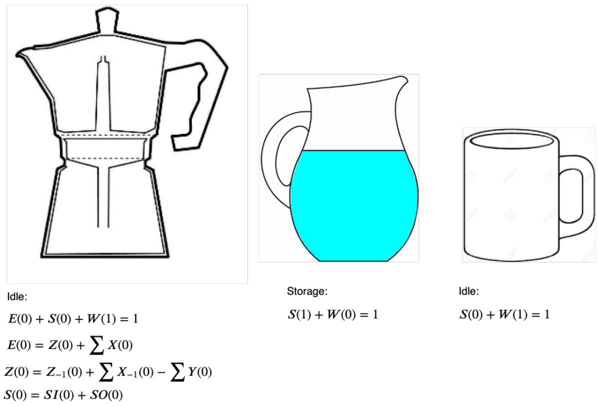
Being a State Task Network, the core of the model revolves around the states of the units. There are, depending on the unit, up to four possible states: the execution state, indicated by variable $E_{j,n}$, the storage state S which is a combination of the input storage state, indicated by variable $S^{I}_{j,n}$, the output storage state, indicated by variable $S^{O}_{j,n}$, and finally the idle state, indicated by variable $W_{j,n}$. Storage units do not have an execution state, since they cannot perform tasks, also since there is no difference between the input and output storage state these are omitted and only $S_{j,n}$ is used. The executing state is activated when a process is either starting (indicated by $X_{i,j,n}$) or being executed (indicated by $Z_{j,n}$) and is deactivated by processes ending (indicated by $Y_{i,j,n}$).

An overview of all the constraints used in the literature can be found in Appendix D. A complete description of these constraints can be found in [12].

An example of how the model works is as follows. The situation examined is a simple coffee making process and considers a water storage device (a jug), a processing unit (a percolator), and a coffee storing device (a mug). Each of these devices has its own state variables. We analyse six time steps the making of a cup of coffee:

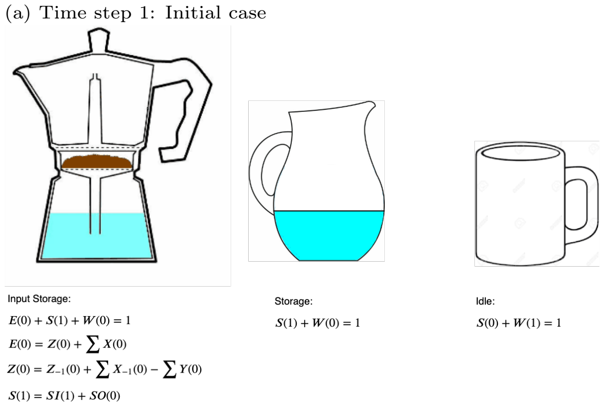
Example: Making coffee

Time step 1: Initial case
The first time step begins with an empty cup, an empty percolator, and a jug of water. See Figure 3.1a. Since neither input nor output materials are stored in the percolator, the storage variables are zero. No processes were already begun and no processes are starting in this timestep, making the execution variable zero as well. This combined results in the idle state being activated. Since the jug and the mug are both storage devices, they do not have an execution state nor have they separate input and output storage's. The jug contains water and is thus in storage state, while the mug is empty and therefore idle.



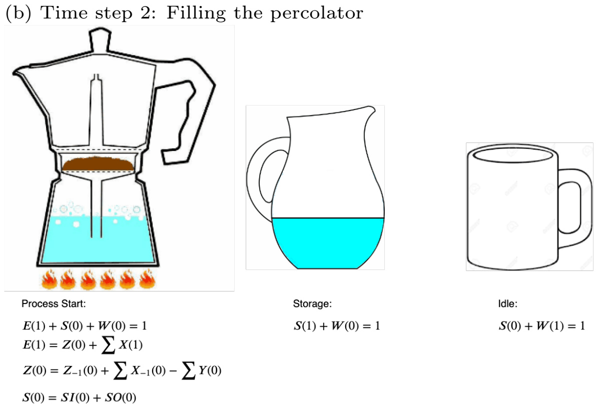
(a) Time step 1: Initial case

Time step 2: Filling the percolator
In order to make a cup of coffee first water and ground coffee beans have to be placed in the percolator. This has been done in the second time step, which can be seen in figure 3.1b. For simplicity the storage of the coffee powder is left out of this example. By adding the water and ground coffee to the percolator the input storage variable is set to one. This causes the Percolator to go into the storage state.



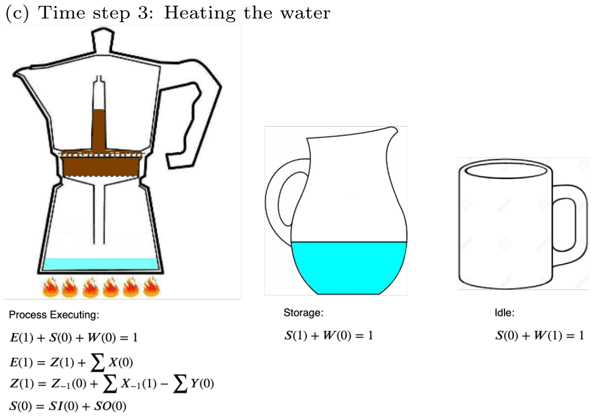
(b) Time step 2: Filling the percolator

Time step 3: Heating the water
The next step is heating the water in the percolator causing it to boil, and force itself trough the coffee powder. This can be seen in figure 3.1c. This is the begin of the process, marked by the starting variable being set to one. This activates the Execution state. Since all input materials are consumed by the process, the input storage variable is back to zero, deactivating the storage state.



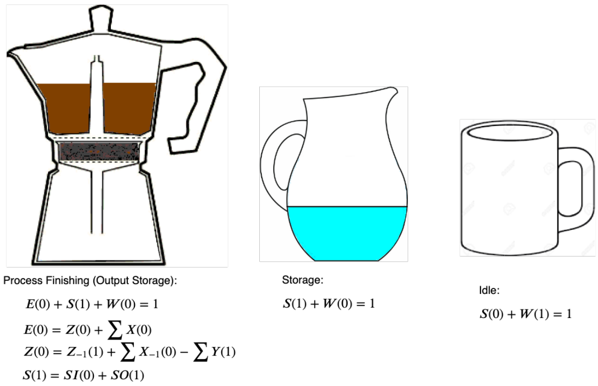
(c) Time step 3: Heating the water

Time step 4: During the process
In some cases a process takes more than one time step. This means that during the process, before it ends, a time step occurs. This is what happens in Figure 3.1d. Since the process is not starting in this time step the starting variable is zero, however in the last time step it has started, and since the processing has not ended yet (stopping variable is zero) the processing variable is set to active. This variable in turn activates the execution variable.



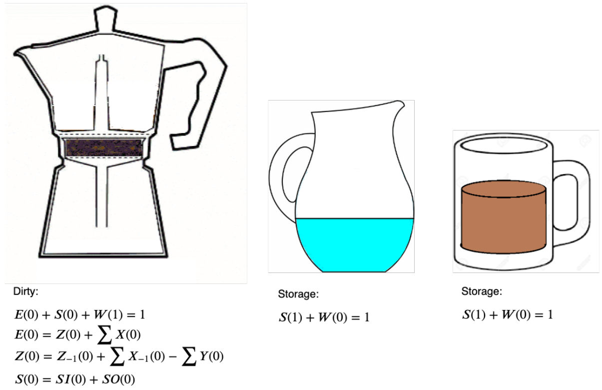
(d) Time step 4: During the process

Time step 5: ending the process
When all the water has boiled the coffee is ready. This situation is shown in Figure 3.1e. The ending variable is set to one, and in combination with the processing variable of the last time step, the processing variable of this timestep is set to zero resulting in the execution state also being zero. The coffee is now in output storage so the output storage variable is set to one, creating the storage state.



(e) Time step 5: ending the process

Time step 6: Pouring the coffee
Since the coffee is now done the mug can be filled. Figure 3.1f shows the final situation.
The mug is filled with coffee so no longer idle but in storage state, the jug still contains some water and therefore in storage state as well, the percolator is empty and thus idle. As the image shows, the used coffee grounds are still in the percolator, prohibiting us from making another cup of coffee. The mode (not to be confused with the state) of the percolator is set to dirty and a cleaning action is needed before it can be used again.



(f) Time step 6: Pouring the coffee

3.1.1. Displaying the results

When the optimisation is done, the found results have to be presented in an clear way. A good way to do this is to make use of a Gantt chart. An example of this can be found in Figure 3.2. In a Gantt chart the X-axis represents the time within the scheduled timehorizon. On the Y-axis the units that make up the system are indicated. Each unit has its own timeline within the chart. Because the timelines are align with eachother, it is simple to find connections between processes. Within the timelines in the Gantt chart not only processes are displayed but also the storage of materials. In this report it was chosen to display input materials at the top of the timeline and output materials on the bottom. Processes take up the full height of the timeline. When a storage tank is analysed, the complete height of the timeline is used for presenting material storage, since there are no processes to display and there is no separate input and output storage. In the blocks on the timelines the material or process name is indicated along with the amount of storage / batch size of the process. The chosen method makes it possible to have processes with a processing time shorter then the timestep the process is scheduled in. In this case the optimisation can decide to start the process late or end it early. This is indicated in the Gantt chart with a darkened block. The colors of the Gantt chart were chosen in such a way that each material is represented by its own color, and processes are colored according to a mix of their input and output material colours. Special processes, such as cleaning processes, were coloured yellow.

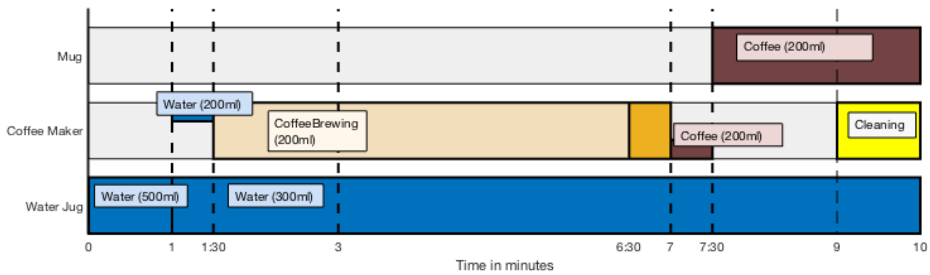


Figure 3.2: Gantt chart of the coffee making example

3.2. The basic method

The literature has split the creation of the method in to two parts, divided over two papers. Since most of the constraints of the basic method were included in the advanced method, but not vice-versa, the complete list of constraints of the advanced model is given in Appendix D. The first part of the model, the basic model, will be discussed in this section, along with the verification of two of the three given examples in the literature. The third example has been reproduced successfully but is omitted in the report. This was done for the sake of brevity. The main difference between the third example and the previous two is the focus on production resources (for example hot steam). Since these kind of resources are not used in the SDOZ ROS, the explanation of this example was deemed less important.

Elements	
Processing Units: R-101, R-102, R-103 Tasks: T1, T2, T3, T4 Storage Units: V-101, V-102, V-103, V-104, V-105, V-106 Materials: RM1, RM2, INT1, INT2, INT3, P1, P2 Utilities: Hot Steam (HS), Cooling Water (CW)	
Logical Connections	
Task/Processing Unit: T1/R-101, T1/R-102, T2/R-101, T2/R-102, T3/R-103, T4/R-103 Task/Utility: T1/HS, T2/CW, T3/HS, T4/CW Material/Storage Unit: RM1/V-101, RM2/V-102, INT1/V-103, INT2/-, INT3/V-104, P1/V-105, P2/V-106	
Stoichiometric Relations	
T1	0.8 RM1 + 0.2 INT1 → INT3
T2	RM2 → 0.3 INT1 + 0.7 INT2
T3	INT3 → P1
T4	0.6 INT2 + 0.4 INT3 → P2
Plant Topology	

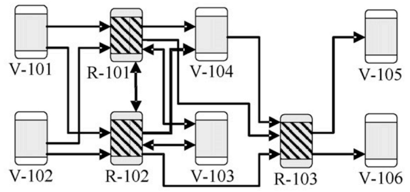


Figure 3.3: The system analysed for example 1 & 2 [12]

The basic model structure assumes that transportation is instantaneous, and that there are no overlapping pipelines causing interference. Furthermore it assumes that material demand is only at the end of the time horizon and that material can be shipped from every container. The constraints governing these assumption are as follows: the instantaneous transport is modelled by variable $F_{m,j,j',n}$. This variable describes the amount of material m transported from unit j to unit j' at timestep n. An example of its use is in constraint 17 of the advanced model, which is shown in Equation 3.1. A similar constraint is used in the basic model. Since the method developed in this thesis is based on the advanced model, it will use the constraint displayed in Equation 3.1

$$I^S_{m,j,n} = I^S_{m,j,n-1} - \sum_{j' \in (J_j \cap (J^S_m \cup J^I_m))} F_{m,j,j',n} + \sum_{j' \in (J_j \cap (J^S_m \cup J^O_m))} F_{m,j',j,n} \leq \varsigma^{MAX}_{m,j} \qquad \forall m \in M^s, j \in J^S_m, n \quad (3.1)$$

Constraint 17 (Equation 3.1) defines $I^S_{m,j,n}$, which is the amount of material m, stored in storage unit j at timestep m. $I^S_{m,j,n}$ is defined as the amount of material stored in the previous timestep ($I^S_{m,j,n-1}$) minus the sum of all transport of material m going out of the storage unit, plus the sum of all the transport of material m coming into the storage unit. The maximum amount of material m stored in unit j is defined with the constant $\varsigma^{MAX}_{m,j}$. As can be seen the transport going out of the unit is defined with $F_{m,j,j',n}$, and material coming into the unit is defined with $F_{m,j',j,n}$. This results in the fact that if material flows from unit A to unit B, unit A will have its output defined by $F_{m,A,B,n}$, and unit B will have its input defined by $F_{m,A,B,n}$. This ensures that the material will be transported instantaneously. There is no constraint prohibiting the simultaneous use of overlapping pipelines in the basic model.

Equation 3.2 shows the constraint governing the shipment of final products. This constraint is not shown in Appendix D, since it is replaced in the advanced model by constraints allowing for flexible shipping times.

$$\sum_{j \in J^S_m} I^S_{m,j,n} \geq d_m \qquad \forall m \in M^s, n = N \quad ^1 \quad (3.2)$$

As Equation 3.2 shows the shipment of material is only defined by the fact that the amount of material m in the system, at the end of the time horizon ($n = N$), should be larger or equal to d_m, the amount of material to be shipped. This is a very straightforward constraint, which assumes that only the amount of material available is important, not the location in which it is stored. Furthermore it only allows for shipments to occur at the end of the timehorizon. In the advanced model the constraints 41 and 42 will be used to ensure that shipments can take place within the timehorizon (see Appendix D).

[1] The literature of the basic model uses k to indicate storage units, for the sake of continuity it was changed here to J^S

3.2.1. Verification of the basic method

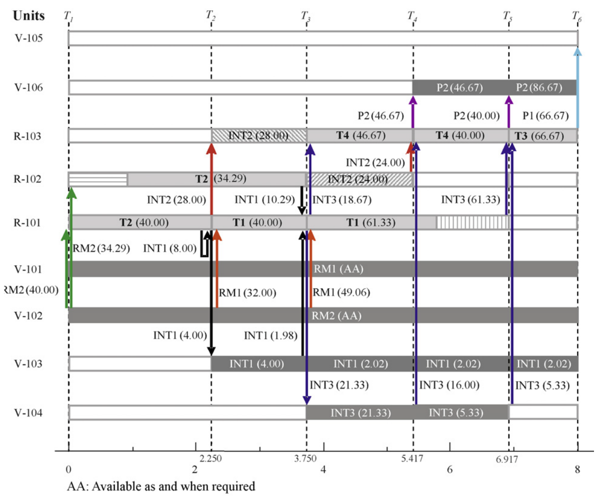
The modelling method gives three examples along with their results. In this section the first two examples were recreated in order to verify the system. The examples analyse the same setup: Three processing units, two for making intermediates and one for making final product, are connected alongside 6 storage tanks, 2 for raw material, 2 for intermediates and two for final products. The system considers 7 materials: two raw materials, three intermediate, of which one has no storage tank, and two final products. An overview of the system, is shown in Figure 3.3. The processing times and utility demands are given in Table 3.2. [12]. Table 3.1 compares the final results of the reproduction with that of the literature. In Appendix E.1 an overview of all intermediate results is shown.

Example	Result literature	Result reproduction	Difference
1	3592.2	3592.2	0 %
2a	7.781	7.781	0 %
2b	11.321	11.321	0 %
2c	13.902	13.856	0.33 %
3	3273.1	3273.1	0 %

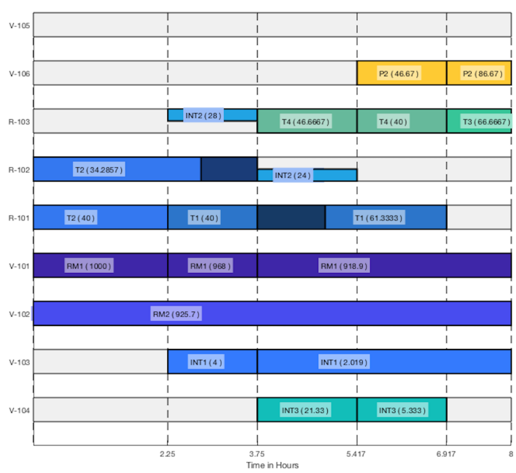
Table 3.1: Final results reproduction examples 1,2 and 3

Example 1

In the first example each product was given a monetary value. The objective of the optimisation was set to produce as much money as possible within a time-limit of 8 hours. To fulfill this objective the optimisation should result in the production of products which have a high value, compared to their input materials. At first the verification of the model failed, but after applying some additions found in Section 3.2.2, the answers given by the literature could be recreated. A side by side comparison between the result in the literature and the recreated model can be found in Figures 3.4 and 3.5. While the cost-function value is identical to that of the literature (3592.2) The Gantt chart is slightly different. In the literature Process 2 in R-102 starts half-way through timestep 1 ($t = 0.943h$) and finishes at the beginning of timestep 3 ($t = 3.75h$). The recreation however Start the process at $t = 0$ and ends it during the second timestep ($t = 2.807h$). This has no influence on the value of cost-function. Another difference between the two results is that an additional timestep at $t = 0$ was needed. This is a result of the additional constraints added in Section 3.2.2. A comparison between the result and calculation time per time step of the literature and the reproduced model, can be seen in Appendix E.1.

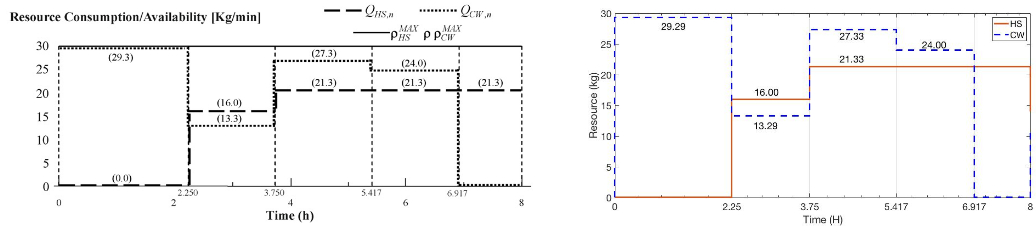


(a) The Gantt chart of Example 1 as given by the literature



(b) The Gantt chart of Example 1 as given by the implemented model

Figure 3.4: A comparison between the Gantt charts in the literature and the output of the implemented model



(a) The resource-usage of Example 1 as given by the literature

(b) The resource-usage of Example 1 as given by the implemented model

Figure 3.5: A comparison between the Gantt charts in the literature and the output of the implemented model

Example 2

The second example is a make-time optimization of the same system used in the first example. Three situations with different demands were analysed. After modifying the system to solve an additional error (see Section 3.2.2) the answers of the method created were almost identical to that of the literature. As mentioned in the previous example, the reproduced model needs an extra timestep to calculate. Remarkable is that, where other in examples the answers of the literature and the reproduced system where identical (apart from the one timestep delay), the answers at 2c differ. The literature found a minimum at Example 2c using 10 timesteps at 13.902, the reproduction found even better solutions when using more timesteps. The best value found was 13.856, using 12 time-steps. The literature however did not calculate scenarios with more than 11 timesteps. The literature used a cpu time limit of 10000 cpu seconds, which was exceeded when calculating the model for 11 timesteps. This meant that the cpu time needed to calculate the model for 12 or more timesteps would be even higher. This fact combined with the fact that the value of 13.902 was found at both 10 and 11 timesteps, led to the conclusion that this was the best possible solution. This was probably a local minimum. Due to advances in computational technology and by using better solving algorithms this research was able to look at significantly higher amount of timesteps. When the model was run with 15 timesteps the value of 13.856 was once again found. While it is possible that this is again a local minimum, the current computational power limits us from examining a model with even more timesteps.

Task i	Processing unit j	Constant processing time $a_{i,j}$(h)	Variable processing time $b_{i,j}$(h/kg)	Minimum batch size $B_{i,j}^{MIN}$(kg)	Maximum batch size $B_{i,j}^{MAX}$(kg)	Utility r	Constant utility usage $f_{i,j,r}$ (kg/min)	Variable utility usage $g_{i,j,r}$ (kg/min/kg)	Maximum utility usage ρ_r^{MAX} (kg/min)
T1	R-101	0.5	0.0025	40	80	HS	6	0.25	30
	R-102	0.5	0.04	25	50	HS	4	0.25	30
T2	R-101	0.75	0.0375	40	80	CW	4	0.3	30
	R-102	0.75	0.06	25	50	CW	3	0.3	30
T3	R-103	0.25	0.0125	40	80	HS	8	0.2	30
T4	R-103	0.5	0.025	40	80	CW	4	0.5	30

Table 3.2: Process information Examples 1 & 2

3.2.2. Issues in reproduction

When reproducing the examples given in the literature some errors were encountered. This section will describe those errors and how they were resolved. One of the first errors encountered was that there was no constraint fixing the end of a process to its beginning. While it was prescribed that if a process starts at a certain time, at a later time a process must end, there was no constraint demanding that this should be the same process. In the profit maximisation for example this resulted in the process with the cheapest input materials starting, and the process with the most valuable output materials ending. This fault was resolved by adding an additional variable, variable $C_{i,j,n}$ which monitors if process i is run on unit j at time n. This variable is used two extra constraints. The first constraint, shown in Equation 3.3, sets $C_{i,j,n}$ to be equal to the difference in starts and endings for a given process until a given moment. The second constraint, shown in Equation 3.4, sets the sum of all C over all processes that can be run on a specific unit to be limited by 1. This prevents two processes being active at the same time, and with it the possibility of one process starting and another ending.

$$\sum_{n'=1}^{n'=n-1} (X_{i,j,n'} - Y_{i,j,n'}) = C_{i,j,n} \qquad for\ n < N \ \ i \in I_j \ \ \forall j \tag{3.3}$$

$$\sum_{i \in I_j} C_{i,j,n} \leq 1 \qquad \forall n, j \tag{3.4}$$

The second problem encountered was that of the transport to production units. The literature uses two constraints to monitor the transport in and out of processing units, shown in Equations 3.5 and 3.6[2]. These are the constraints for the input storage ($I^I_{m,j,n}$) and output storage ($I^O_{m,j,n}$) of a material m in unit j at time n. The constraints dictate that the input storage in a unit is equal to the storage in the previous time, plus the transport from connected storage units that can store the material, plus the transport from connected producing units, plus the amount of material used in the batch that started in the timestep. For the output storage this is almost identical except that the beginning of transport out of the unit, and the ending of processes in the unit should be taken into account. What the constraints omit is that the connected processing units should be able to store the material coming into or leaving the unit. This can result in infeasible transports. To prevent this an additional constraint was made, shown in Equation 3.7. This constraint prohibits transport from all processing units except the processing units which can have the selected material in output storage.

$$I^I_{m,j,n} = I^I_{m,j,n-1} + \sum_{j' \in (J^S_j \cap J^S_m)} F_{m,j',j,n} + \sum_{j' \in J^P} F_{m,j',j,n} + \sum_{i \in (I_j \cap I^C_m)} \gamma_{i,m} B^S_{i,j,n} \qquad \forall m, j, n \tag{3.5}$$

$$I^O_{m,j,n} = I^O_{m,j,n-1} - \sum_{j' \in (J^S_j \cap J^S_m)} F_{m,j,j',n} - \sum_{j' \in J^P} F_{m,j,j',n} + \sum_{i \in (I_j \cap I^P_m)} \gamma_{i,m} B^E_{i,j,n} \qquad \forall m, j, n > 1 \tag{3.6}$$

$$F_{m,j' \notin J^{out}_m,j,n} \leq 0 \qquad \forall m, j, n \tag{3.7}$$

Furthermore the constraint in Equation 3.5 does not specifically mention that at $n = 1$ the term $I^I_{m,j,n-1}$ should be replaced with I^I_0, the initial input storage. Constraint 22 applies only to $n > 1$, but the literature omits to include a constraint for $n = 1$, which should include I^O_0, the initial output storage.

A specific error for examples 2b and 2c was encountered. When introducing Examples 2 the literature states the following:

> *In all the cases, parameter H, which now only plays the role of a reasonable upper bound on both slack variables (see Eqs. (4) and (5)) and time intervals (see Eqs. (6)-(8)), was fixed to 8 h.*

This conflicts with the answers given for Examples 2b and 2c, which were both larger than 8 h (respectively 11.321 h and 13.90 h). Increasing the parameter H solves this problem. For the recreation of the examples this was chosen to be 20 h, significantly larger than the highest feasible value found by the literature (14.723h). Finally the literature describes the amount of material m stored in processing unit $j \in J^P$ during time interval n in the nomenclature list as $I^V_{m,k,n}$, but in the constraints uses the variable name $I^S_{m,k,n}$.

[2] The variables in these equations are again modified for continuity

3.3. The advanced method

As mentioned in Section 3.1 the final method consists of several features implemented on the basic method mentioned in Section 3.2. Some of these features were provided in the literature. This is called the advanced model. This section will discuss the features of this advanced model and reproduce the examples provided.

3.3.1. Flexible shipping

The models used in the previous examples could only handle material shipments on the end of the timehorizon. However in reality shipments do not only occur at one point in time but can occur various points during the timehorizon. This is solved by adding shipments to the model. These are modelled by adding a new binary variable to the system, $O_{l,j,n}$. This represents if shipment l is received (or send) by unit j at timestep n. This variable is used in multiple constraints. The first constraint it is used in is constraint 41 shown in Equation 3.8.

$$\sum_{j\in J^s_{m(l)}} \sum_{n\in N} O_{l,j,n} = 1 \qquad l \in L \tag{3.8}$$

In this constraint the set $J^s_{m(l)}$ represents the set of storage units which are able to store the material associated with shipment l. Since some cases it is possible for a shipment to arrive late, or to be send early the timing constraint cannot be made exact. This is accounted for by adding two slack variables $\bar{T}^T$ and $\bar{T}^E$, which can be set to zero to ensure a on-time shipment or constraint as smaller than a margin value for less strict shipments. These are used in constraint 42, shown in equation 3.9.

$$T_n - H(1 - \sum_{j\in J^s_{m(l)}} O_{l,j,n}) \leq \tau_l + \bar{T}^T_l - \bar{T}^E_l \leq T_n + H(1 - \sum_{j\in J^s_{m(l)}} O_{l,j,n}) \qquad l \in L, n \in N \tag{3.9}$$

In this constraint T_n is the time corresponding to timestep n, H is the length of the timehorizon and τ_l is the planned time for shipment l. The constraint shows that if the shipment does not take place in timestep n ($O_{l,j,n} = 0$), the timing of the timestep is free to be determined by other constraints. However if the shipment does take place it is limited to the range indicated by the slack variables $\bar{T}^T$ and $\bar{T}^E$. Lastly the term $+\sum_{l\in L_m} q_l O_{l,j,n}$ is added to the material storage in the storage unit constraint. This ensures that if a shipment is active the amount of material q_l is added to the storage. For outgoing shipments the value of q_l is negative.

3.3.2. Previous horizon and maintenance tasks

Two special kinds of tasks are the Previous Horizon and Maintenance tasks. Unlike other tasks some, if not all, of the variables of these tasks are fixed a priori. For example Previous Horizon tasks, tasks that are already started when the scheduling begins. Since these processes were already running they will have a fixed batch size and a fixed starting point at T=0. This is ensured by constraint 34, shown in equation 3.10.

$$X_{i,j(i),1} = 1, X_{i,j(i),1<n<N} = 0 \qquad i \in I^{PH} \tag{3.10}$$

This constraint ensures the previous horizon tasks, indicated by set I^{PH}, start at their assigned units $j(i)$, in the first timestep and are unable to start at any other time. Furthermore the time needed for the process is reduced, since some of the processing has already been done before the starting point.

Maintenance tasks can be scheduled in two different ways. The first method is to separately make a maintenance schedule. After this has been done the maintenance processes can be loaded into the production scheduler as tasks with a fixed starting and end point, much like the previous horizon tasks. The second method is to set a earliest starting and latest finishing time of a maintenance process, combined with the time duration of the maintenance. This enables the optimisation to fit the maintenance in the schedule in the least interfering way. To ensure both these tactics can be applied constraints 25, 26 and 27 , shown in Equations 3.11, 3.12, and 3.13 were used.

$$\sum_{n<N} X_{i,j(i),n} = 1 \qquad i \in I^{MT} \tag{3.11}$$

$$\tau^{MEB}_i \leq T_n + \bar{T}^{LB}_{j(i),n} + H(1 - X_{i,j(i),n}) \qquad i \in I^{MT}, n < N \tag{3.12}$$

$$\tau_i^{MLE} \leq T_n + \bar{T}_{j(i),n}^{EE} - H(1 - X_{i,j(i),n}) \qquad i \in I^{MT}, n > 1 \tag{3.13}$$

Constraint 25 is very straightforward, it dictates that all tasks of the set I^{MT}, which is to say all maintenance tasks, have to performed exactly once during the time horizon. Constraints 26 and 27 set τ_i^{MEB} and τ_i^{MLE} to be the earliest beginning and ending of the task. When the difference between τ_i^{MEB} and τ_i^{MLE} is equal to the processing time of the maintenance task, the task will be fixed. If the difference is greater the optimisation will decide the timing within the given frame. $\bar{T}_{j(i),n}^{LB}$ and $\bar{T}_{j(i),n}^{EE}$ are used as slack variables. These will be zero when the task is fixed. When a reschedule is needed, due to for example late material shipments or machine breakdowns, Previous Horizon tasks can be used to fix all started processes in place, while rearranging all tasks that have not, in order to update the schedule.

3.3.3. Contamination and cleaning

Since this project concerns a refinery that makes oils for food products, it is very important that an eye is kept on contamination. Contamination can occur in different ways. Some processes can be run only once before the production units involved have to be cleaned. This is for example the case when working with raw milk. Other processes can be repeated but cannot be followed by specific other processes. An example of this is the processing of almond oil, since after processing products containing nuts no allergen free product can be made before cleaning. However, it might be possible to run a batch of a different product, which in itself does not contain any nuts, but is not prohibited from containing traces, first. This batch would then absorb practically all contamination from the production unit, leaving a clean enough environment to run allergen free products. [3] To monitor if a production unit is contaminated and/or needs cleaning modes are introduced. These mode indicate if a unit is clean or contaminated. When a process starts it consumes the mode of the unit it is executed on, and when a process ends it will produce (a different) mode on that unit. By limiting the different modes that can be consumed by a process contamination can be prevented. For example a unit "R-101" can execute three different production processes named T1, T2 and T3. When the unit is clean, all processes can be run. However not every process can be run after every other process: While T1 can be run after itself and after T2, it cannot be run after T3. T2 However can be run after T1 and T3 but not after itself and T3 can be run after itself and T2. Finally cleaning process TC can be run after all processes which produces a completely clean unit able to execute any of the processes. Table 3.3 gives an overview of this situation.

Process	after T1	after T2	after T3	after TC
T1	✓	✓		✓
T2	✓		✓	✓
T3		✓	✓	✓
TC	✓	✓	✓	✓

Table 3.3: Contamination between processes

When all production processes have to be run exactly once all possible orders of execution are given by Table 3.4. As the table shows, there are only two possible sequences where all three processes are run, without having cleaning processes in between. When the amount of times a cleaning process is started is added to the cost function, the optimisation will automatically opt for one of these options.

[3] A small part of the contaminant may remain in the production unit: This is one of the reasons a lot of products have "may contain traces of nuts" on their label.

Option	Order	Cleaning
1	T1 → T2 → T3	
2	T1 → TC → T3 → T2	✓
3	T2 → T1 → TC → T3	✓
4	T2 → T3 → TC → T1	✓
5	T3 → TC → T1 → T2	✓
6	T3 → T2 → T1	

Table 3.4: Possible execution orders to prevent contamination

3.3.4. Pipelines

In a refinery the pipelines connected all the tanks and production units play an important role in the scheduling process. Most connections between tanks and production units share common devices with other connections. These devices could be pumps, valves, or even complete pipe sections. It is important that when transport is scheduled, no other transport is making use of the same devices. Furthermore, since transporting large amounts of material can take up considerable time, the schedule should also account for the transport times. These functions are modelled as follows. All connections within the previous model are replaced by a new set of units J^T the transport units. These units with zero internal storage represent the complete transport between two "normal" units. The input of a transport unit is directly connected to the output of a production of storage unit. Its output is directly connected to the input of the receiving unit. Transport processes are added, which have the same product as input and as output. Internal instantaneous transfer, which exists in processing units for transporting output material to their input, is prohibited for transport units. Transport processes (I^T) are used to transfer material from the input of the transfer unit to their output. With separate constraints the transport devices making up these units are modelled, to prevent two transport processes using the same device at the same time.

3.3.5. Verification of the advanced method

As with the basic method, the advanced method was accompanied by three examples in the literature. These examples showcase the handling of shipments, maintenance and previous horizon tasks, contamination and cleaning, and the use of pipelines. The same as with the basic method, examples were recreated, during which some problems were found and resolved. In this sections the examples of the advanced method are discussed. The setup that was used is the same as with the examples of the basic method. Table 3.5 shows a comparison between the final answers from the literature and the reproduction. In Appendix E a complete overview of the intermediate answers is given.

Example	Result literature	Result reproduction	Difference
4	3253.3	3253.3	0 %
5	3093.0	3093.0	0 %
6	2258.3	2258.3	0 %

Table 3.5: Final results reproduction examples 4, 5 and 6

Example 4

The fourth example (which is the first example of the second paper), looks at the implementation of shipments, maintenance, and previous horizon tasks. The example includes three shipments: a raw material delivery and two final product outgoing shipments. The example also includes two previous horizon tasks and a fixed maintenance task, as can be seen in Table 3.6. Since one of the previous horizon tasks has no outputs it can be assumed that this is actually a second fixed maintenance task. The layout examined is the same as the previous examples and the goal is once again profit maximisation over a time horizon of 8 hours. Due to new tighter constraints added in the advanced model some of the extra constraints made in section 3.2 could be removed. This results in the reproduction needing the same amount of time steps as the example given by the literature. Remarkable however is that when 7 timesteps are used the reproduction gives a answer of 1780.0, which is significantly better than the answer in the literature literature, which was 906.67. Since only the complete answer for 8 timesteps

Previous Horizon Tasks						
Task $i \in I^{PH}$	Unit $j \in J^P$	$a_{i,j}$(h)	$B_{i,j}$(kg)	Utility $r \in R$	$f_{i,j,r}$(kg/min)	$g_{i,j,r}$(kg/min/kg)
PHT1	R-101	0.85	40	CW	4	0.3
PHT2	R-102	1.25	-	-	-	-

Maintenance Tasks						
Task $i \in I^{M}$	Unit $j \in J^P$	$a_{i,j}$(h)	τ_i^{MB}(h)	τ_i^{ME}(h)	Utility $r \in R$	$f_{i,j,r}$(kg/min)
MT1	R-103	0.5	5.0	5.5	-	-

Table 3.6: Previous Horizon and Maintenance processes information Example 4

is given, and this is identical to the answer from the reproduction, it cannot be said why and how the answers differ at 7 timesteps. The complete comparison between the timesteps and their results, can be found in Appendix E.2.

Example 5

The fifth example considers contamination and cleaning. In this situation different production processes cannot be executed after each other without cleaning the processing unit. It considers four cleaning tasks, shown in Table 3.7. The cleaning tasks could be divided into two types. The first type of cleaning task creates an environment which can only be used for one process: For example cleaning process CT1 creates mode $Clean_{T2}$ which is only suitable for executing process T2. The other type can create an environment suitable for multiple processes: for example cleaning process CT5, which creates mode $Clean$ which is suitable for all processes.

Cleaning Task $i \in I^C$	Task Sequence $i' \in I^P \rightarrow i'' \in I^P$	Initial Mode $c^{(i)}$	Final Mode $c^{(f)}$	Processing unit $j \in J^P$	Cleaning unit $j \in J^C$	Cleaning time (h)	Cleaning Costs ($)
CT1	T1 → T2	$Dirty_{T1}$	$Clean_{T2}$	R-101	C-101	0.15	80
CT2	T2 → T1	$Dirty_{T2}$	$Clean_{T1}$	R-101	C-101	0.18	100
CT3	T1 → T2	$Dirty_{T1}$	$Clean_{T2}$	R-102	C-101	0.10	45
CT4	T2 → T1	$Dirty_{T2}$	$Clean_{T1}$	R-102	C-101	0.12	60
CT5	T3 → T4 T4 → T3	$Dirty_{T3}$ $Dirty_{T4}$	$Clean$	R103	C102	0.15	80

Table 3.7: Cleaning processes for Example 5

The results of the reproduction can be found in Appendix E.2. Again some of the intermediate answers differ, but the final answer, and the amount of timesteps needed are identical.

Example 6

The final example given by the literature is example 6. This example considers the pipelines and devices of which the pipelines consist. To reduce complexity Processing Unit R102 was removed from the system. Once again the final answer was identical but the sub-optimal answers differ.

3.3.6. Issues during reproduction

As with the first three examples, the reproduction of the second three examples was met with some difficulties. In this section some of these difficulties are discussed alongside their solutions. Example 4 (example 1 of part 2), using the information given by the literature was impossible to reproduce. When the variable processing time of process T1 at processing unit R102 was set to 0.05 instead of the given 0.04 the solution matches perfectly with the solution given by the literature. The other two examples used the given 0.04. Furthermore to ensure the right batch size and producing time for the Previous Horizon tasks, an additional constraint seen in Equation 3.14 was made.

$$BS_{i,j,n} = B_i^{fixed} \qquad \forall i \in I^{PH}, j \in J_i, n = 1 \tag{3.14}$$

In which B_i^{fixed} is the predetermined batch size for the previous horizon task. This constrains the batch size and therefore the amount of output material. The time needed for the process to finish in the current schedule is not dependent on the batch size, but is determined before the schedule is made. Therefor the variable processing time should be set to 0 and the constant processing time to how much time the process still has left.

In the fifth Example (example 2 of part 2) the reproduction could again not create the given result. After changing the maximum batch size of processing unit R102 from 80 to 60 the exact answer as the literature was found. Whether or not this was also used in Example four is unclear, since the final answer does not contain a batch greater than 40. Example six does not contain processing unit R106. Another issue in Example five was with the assignment of modes. The original constraints did not define that each unit could have only one mode at any given moment. This resulted in the optimisation automatically assigning multiple, to full fill mode consumption constraints. An extra constraint, found in Equation 3.15 solves this by forcing processing units to be either busy or have one and exactly one mode.

$$\sum_{c \in C_j} A_{c,j,n} + E_{j,n} = 1 \qquad \forall j \in J^P, n \tag{3.15}$$

The amount of material in the input storage of a processing unit should be defined as the amount of input storage in the previous timestep, minus the amount of material used in processes, plus the amount of reused output material plus the amount of material delivered in the timestep. In the case of non-instantaneous transfers, this last part should be the batch size of the transfer to the input storage that ends in this timestep. However the literature uses the batch size of transfers starting in the timestep, as can be seen in Equation 3.16.

$$I^I_{m,j,n} = I^I_{m,j,n-1} + \sum_{j' \in (J^S_m \cup J^O_m)} B^S_{i(m,j''),j''(j',j),n} + P_{m,j,n} + \sum_{i \in (I_j \cap I^C_m)} \gamma_{i,m} B^S_{i,j,n} \qquad \forall m, j \in J^I_m, n \tag{3.16}$$

Equation 3.17 shows the modified version of the constraint used in the model.

$$I^I_{m,j,n} = I^I_{m,j,n-1} + \sum_{j' \in (J^S_m \cup J^O_m)} B^E_{i(m,j''),j''(j',j),n} + P_{m,j,n} - \sum_{i \in (I_j \cap I^C_m)} \gamma_{i,m} B^S_{i,j,n} \qquad \forall m, j \in J^I_m, n \tag{3.17}$$

It has to be noted that there is a minus in the last term. This is the result of another modification in which the direction of variable $\gamma_{i,m}$ is transferred from the variable to the constraints. Doing this gives a clearer overview of the processes in the constraints. The reverse of this problem occurred at constraint 49 where the output of material was being defined as the end of the transport processes out of the processing unit. This constraint was replaced by Equation 3.18.

$$I^O_{m,j,n} = I^O_{m,j,n-1} - \sum_{j' \in (J^S_m \cup J^I_m)} B^S_{i(m,j''),j''(j,j'),n} - P_{m,j,n} + \sum_{i \in (I_j \cap I^P_m)} \gamma_{i,m} B^E_{i,j,n} \qquad \forall m, j \in J^O_m, n \tag{3.18}$$

3.4. Summary

In this chapter the modelling method selected in Chapter 2 is discussed. The chapter started with an introduction on the structure of the method including an example about making coffee. The chapter continued by explaining the standard form of the method, along with verification of the examples given in the literature. Since not all examples could be directly reproduced, it is also explained how the modelling method could be adjusted to fix this. After this, the same was done for the advanced method, in which additional features were added to the standard method. In the next chapter improvements were made to better fit the method to the problem at SDOZ, by adding extra features. The method, including the additions will then be verified.

4

Improvements to the method

After the model of the literature was verified in Chapter 3, some Improvements had to be created in order to accurately schedule the ROS. In this chapter the improvements that were made are discussed, and a verification example is given.

4.1. Combining features

While the literature has given examples of the features implemented, the combination of features was not mentioned. Therefore the first thing to do was to combine the features given in the literature into one model. The newly created features in this section were all implemented on this combined model.

4.2. Loading spots and preloading

The ROS at SDOZ makes use of a loading hall to load the outgoing material into trucks. In this loading hall six loading spots are available, each with its own supply pipeline. With these pipelines all storage tanks are connected to one or multiple loading spots. When a material has to be loaded a loading spot that is connected to a tank containing the material has to be selected. Since the selection of which loading spot has to be used influences other shipments, by occupying either pipe segments or the loading spots themselves, it is vital that this selection is included in the optimisation. The loading spots in the loading hall are modelled as shared storage tanks, with no volume, which are set to be the source of the outgoing shipments. Since the capacity of the tanks is zero the material has to be transported to the loading spot exactly at the time of delivery, mimicking the real world situation. This is different compared to the situation where the storage tanks themselves are the source of the outgoing shipments, since now the pipelines to the loading hall are taken into account. As mentioned before, and shown again in Equation 4.1, constraint 41 of the literature indicates that shipments can be done by units in the group $J^S_{m(l)}$. This group includes all tanks that store the material that has to be shipped.

$$\sum_{j \in J^S_{m(l)}} \sum_{n \in N} O_{l,j,n} = 1 \qquad l \in L \tag{4.1}$$

The original constraint therefore allowed shipments directly from tanks, instead of shipments from loading spots. This constraint has been changed to the constraint shown in Equation 4.2, in which $J^L_{m(l)}$ indicates the set of loading spots capable of loading the material of shipment l.

$$\sum_{j \in J^L_{m(l)}} \sum_{n \in N} O_{l,j,n} = 1 \qquad l \in L \tag{4.2}$$

The constraint ensures that each shipment will occur exactly once, at exactly one of the indicated loading spots. Some materials however can be preloaded into so called preloader trucks. These can be stored on site and be swapped out for empty trucks without occupying the loading spot at the time of shipment. This can reduce not only the tanks in use, but can also prevent conflicts when multiple shipments have to be fulfilled at the same time. In order to simulate this shared storage tanks are created with an extra constraint, which dictates that if material is taken out of the tank (due to a shipment), the tank should be completely emptied.

4.3. Quality assurance

When new material is added to a tank, the quality assurance department first has to take a sample, and check its properties, before the material is allowed to be used in a process or pumped further. Since most products made by SDOZ are destined for the food industry this is a very important step. It is therefore vital that the system always schedules time for quality assurance, before the material is shipped. The time this process takes is not dependent on the amount of material stored but on the material itself. In order to correctly model this, for each material that has to be subjected to quality assurance, two materials are modeled: a "Checked" version and an "Unchecked" version. A process with a constant processing time is added for each material, able to convert the unchecked material to checked material. Since this process has to be executed on the material while stored, the storage units involved are now also modeled as production units. When material is added to tanks already in use (and already checked), the quality assurance has to be done again to ensure the quality of the blend. Since only one material can be stored at any given time, the material that has already been checked, should first be marked as unchecked, after which the incoming unchecked material can be added. This is done by adding an additional process with zero processing time transforming checked material into unchecked material, undoing the quality assurance. Some products have to be checked after each transfer. In this case the transfer process is modified to transform checked material into unchecked material.

4.4. Objective function

The objective function or cost function can be seen as the goal of an optimisation model. In a minimisation optimisation problem the solver will try to minimize the result of the objective function by changing the values of the variables in the system. The constraints of the model indicate the bounds of these variables and their internal relations.

The schedule should be optimised over two Key Performance Indicators (KPI's). The first KPI is that the tanks should be filled effectively, meaning that as much space as possible is kept free. Since tanks that are partially filled with product A cannot be used for storage of product B, only the volume of completely empty tanks should be taken into account. In order to monitor this the idle state variable could be used, since this indicates that no material is stored and no quality checks are being performed on a given tank. However, since the length of the timesteps is not known a priori, this would mean that a volume of 10 tonnes being freed for a timestep of 10 hours would be less preferable to a volume of 7.5 tonnes being freed for two timesteps of each 10 minutes. As this is not the case a compensation for the length of the timestep in which the volume is freed has to be made. This is done by using the variable $T^W_{j,n}$ which indicates how much time a unit, in this case a storage tank, is idle. The summation of the idle times multiplied by the volume of the linked tank should be maximized. The second KPI is the amount of assurance checks and cleaning operations, these should be as few as possible, since these are costly. Since both the quality assurance and the cleaning are modelled as processes, the summation of their starting variables can be taken as indication of this KPI and should be minimised. By subtracting the summation of the previous KPI from this summation a minimisation objective function can be made. This results in the objective function shown in Equation 4.3.

$$\mathrm{F} = \sum_{n \in N} \sum_{i \in (I^C \cap I^{QA}} X_{i,j(i),n} - \sum_{n \in N} \sum_{j \in J} T^W_{j,n} \beta^{MAX}_j \tag{4.3}$$

If emphasis on one of the KPI is desired, multiplication factors could be added to the summations.

4.5. Verification of the developed method

In order to test if all the functions behave as planned, a small example has been made. In this example the setup shown in Figure 4.1 is analysed. The setup is based on the components of the real ROS. The setup consists of three tanks and a mixingtank, two loadinghall spots, and one preloader. A virtual processing unit called "steamers" is added to simulate the input from the refinery. While this is marked as a processing unit, there are no processes executed on the steamers. The volume is set to zero ensuring that any shipments arriving will have to be transported to tanks immediately. This is done to simulate the real behaviour at SDOZ where materials cannot overstay in the steamer lines. For the output of the system two loading hall spots are used, which can either ship the material themselves or use a preloader to load materials to be shipped in advance of the actual shipping time.

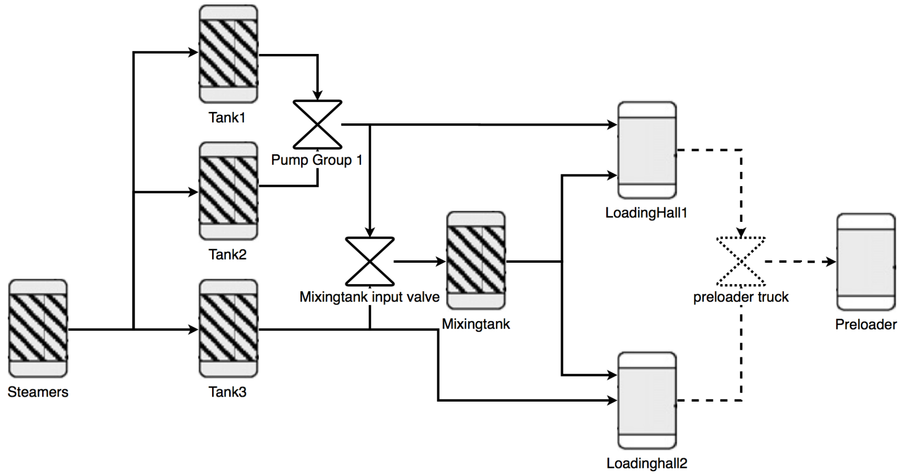


Figure 4.1: Layout of the verification example

The connections between the tanks and the loading spots are mapped in Table 4.1. As can be seen, only transport devices which can be used by multiple connections, such as the pump of group 1, are listed in as transport devices. Other transport devices, such as the pump of Tank 3 have only one input and are therefor not modelled. This reduces the model size while still maintaining accuracy.

Name	From	To	Transport devices
Tr-1	Steamers	Tank1	-
Tr-2	Steamers	Tank2	-
Tr-3	Steamers	Tank3	-
Tr-4	Tank1	LoadingHall1	Pump Group 1
Tr-5	Tank2	LoadingHall1	Pump Group 1
Tr-6	Tank1	Mixingtank	Pump Group 1, Mixingtank input valve
Tr-7	Tank2	Mixingtank	Pump Group 1, Mixingtank input valve
Tr-8	Tank3	LoadingHall2	-
Tr-9	Tank3	Mixingtank	Mixingtank input valve
Tr-10	MixingTank	LoadingHall1	-
Tr-11	MixingTank	LoadingHall2	-
Tr-12	LoadingHall1	Preloader	preloader truck
Tr-13	LoadingHall2	Preloader	preloader truck

Table 4.1: Transport lines used for the verification example

The example considers four materials and four virtual materials which are unchecked versions of the real materials made for the purpose of quality assurance checks. The example also considers two blended materials, M1 and M2, which can be produced in the mixing tank. The blend ratios of these two blends can be found in Table 4.2. In this table the presence of contaminants in materials is also indicated.

In the test case, multiple products are produced by the steamers and multiple products, including one blend, are to be shipped. The exact shipments and their timings can be found in Table 4.3. A negative value for the amount of a shipment is used to indicate material going out of the system. At the start of the time horizon Tank 1 will contain 2 tonnes of material P1, and Tank 3 will be performing quality assurance checks on 3 tonnes of material P3. To further show the capabilities of the created model, a demand for at least 2.5 hours of maintenance between 1h and 5h for tank 2 is added.

Name	Unchecked Variant	Contaminant	Recipe
P1	UncheckedP1	✓	None
P2	UncheckedP2		None
P3	UncheckedP3	✓	None
P4	UncheckedP4	✓	None
M1	None	✓	$\frac{1}{2}P3 + \frac{1}{2}P4$
M2	None	✓	$\frac{1}{3}P3 + \frac{2}{3}P4$

Table 4.2: Materials used for the example

Shipment	Time	Material	Amount	From/to
SH1	1	UncheckedP1	+3	Steamers
SH2	3	UncheckedP4	+5	Steamers
SH3	6	UncheckedP1	+10	Steamers
SH4	6.5	UncheckedP3	+10	Steamers
SH5	7	UncheckedP2	+5	Steamers
SH6	9	P3	-5	LoadingHall 1 or 2
SH7	10	M1	-10	LoadingHall 1,2, or Preloader
SH8	13	P1	-5	LoadingHall 1,2, or Preloader
SH9	15	P2	-5	LoadingHall 1,2, or Preloader

Table 4.3: Shipments used for the example

4.5.1. Expected result

The example case was made in such a way that the result will show all the functionalities of the developed system. For example, by starting tank 1 with a checked batch of material P1, and shipping a new (unchecked) batch of P1 to the system at t=1. It is expected that, in order to storage the material as effectively as possible, an inverse quality check will be performed on tank 1. This way both batches P1 can be added together. Since only checked material can be shipped out of the system, a new quality check then has to be performed on tank 1. It is expected that, in order to free as much tanks as possible, the material will then be transported tot the preloader, awaiting its shipment at t=13. The Gantt chart of the expected result can be found in figure 4.2.

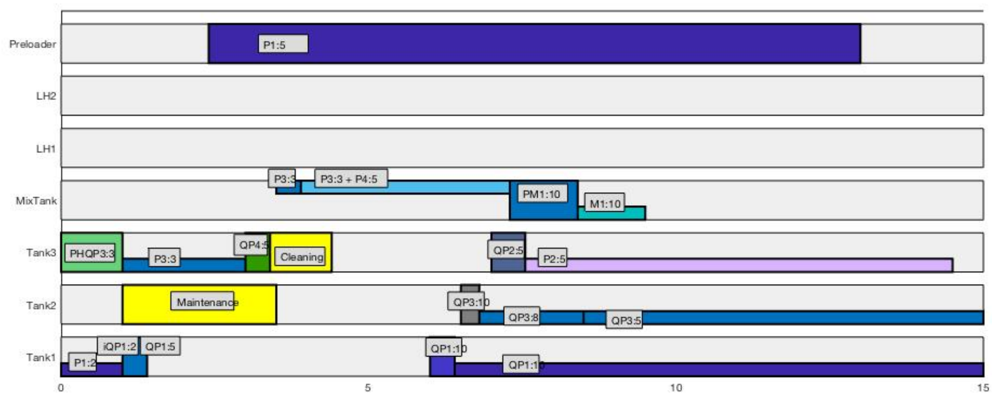


Figure 4.2: Expected result for the verification of the developed method

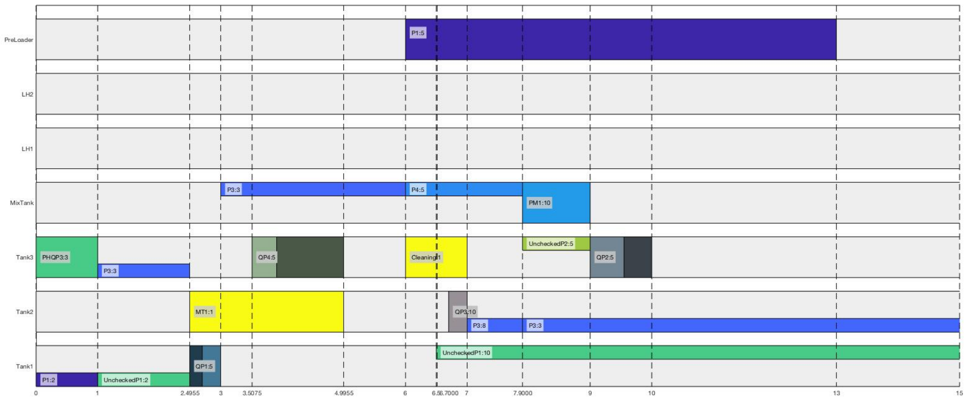


Figure 4.3: Gantt chart of the verification example

4.5.2. Result

In Figure 4.3 the result of this example is shown. The answer found is not identical to the expected result. This is caused by two reasons. The first reason is that, while the system accounts for the late beginnings of the transport processes, this is not visualised in the Gantt chart of the tank. This means that, while the material P2 leaves tank 3 at t=14.4925, the Gantt chart only shows the tank being occupied till t=10. This is inconvenient for reading the chart, but has no influence on the actual result of the optimisation. The second reason the Gantt charts do not exactly match is that this is not the optimal solution to this example. Calculating an optimal solution would take to much CPU-time: The first feasible solution was found after slightly more then 1 hour (4298s). The objective function value was -451.5950 and the optimality gap was 172%. When after 96 hours the time limit was reached, the algorithm had found 22 solutions. The best solution had an objective function value of -468.7451 and the optimality gap was reduced to 13.4 %. It is likely that if the system had run for more time that the best solution would not have improved by much, rather the best bound would have decreased enough to close the optimality gap. A graph depicting the best bound and the best solution over time can be found in Figure 4.4. In this graph each found solution is marked with a cross. Increasing the amount of timesteps could improve the final result (if the optimisation was run until the gap was closed), but since this would take even more cpu time it was chosen to use the current answer.

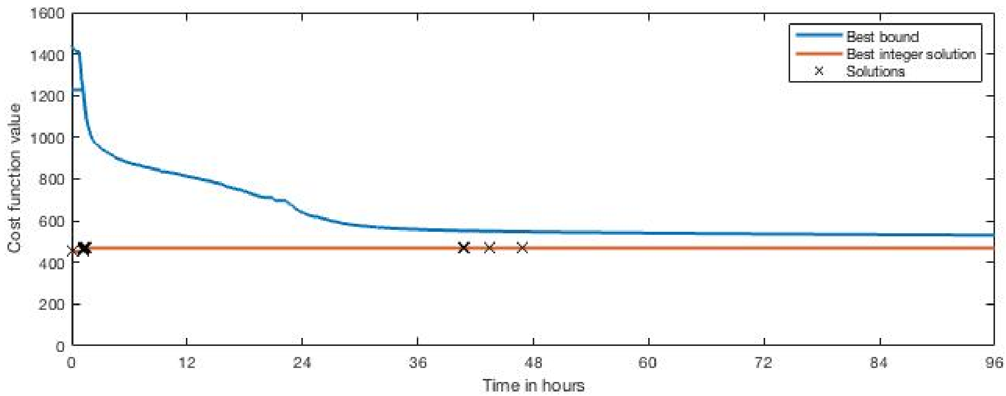


Figure 4.4: Solution progress over time

While the solution shown in Figure 4.3 is not optimal it clearly shows all the implemented functions working together. The previous horizon tasks, maintenance tasks, and cleaning tasks added in the previous section can be seen working at Tanks 2 and 3. The selection of the loading spots can clearly be seen by monitoring the transport routes, in Figure 4.5. This shows that Shipments SH6 and SH7 were transported through LoadingHall1, while shipment SH9 was transported through LoadingHall2.

Shipment SH8 was passed through LoadingHall1 but was preloaded to the preloader and shipped from there, proving that the preloading function works. The quality assurance and reverse quality assurance can be seen in Tank 1, which starts out by having checked material at t=0, converting this to unchecked material in t=1, and, after more unchecked material is added, checking the material at t=2.4955.

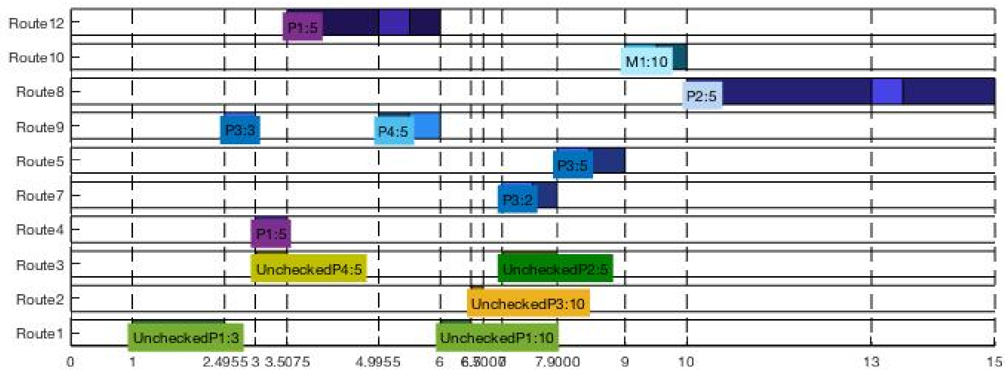


Figure 4.5: Transport in the verification example

4.6. Summary

In this chapter the improvements that were made to the existing method were discussed. These consisted of additions which were made in order to model the SDOZ ROS more accurately. The additions included the use of loading spots, the use of preloaders, and the inclusion of quality assurance checks. The complete method, including all the features described in the literature and the additions made, was then verified using an example. The result of this example showed that all the features and additions work as designed. In the next chapter the validation and application of the developed method will be discussed.

5

Validation and application of the developed method

In this chapter the validation of the developed method will be discussed as well as the possible application of the method to the SDOZ ROS. The chapter will start by explaining the modifications made to the verification example, in order to use it for validation. After this, the results of the current method, which is reactively assigning tanks to materials by hand, are compared with the results of the developed method. When the model is validated the application will be discussed. This will be done by first mapping the ROS and its internal connections. After this the materials and shipments will be analysed. The chapter will end with discussion of a working proof of concept, within the limits computational power available.

5.1. Validation

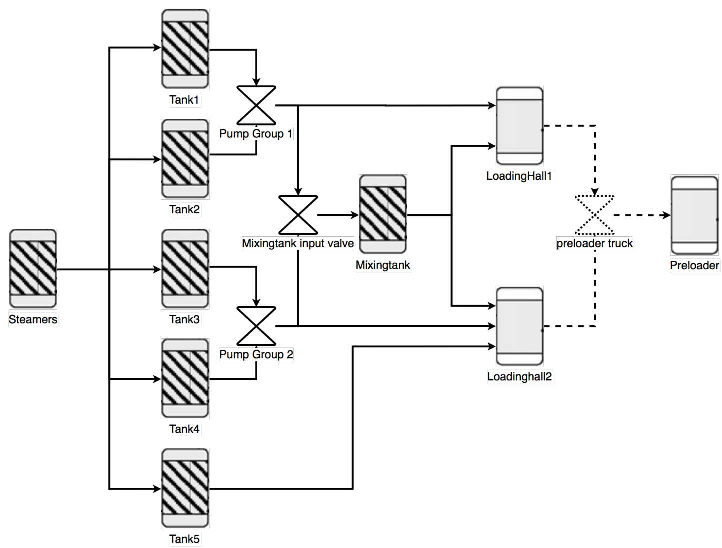


Figure 5.1: Layout of the example for validation

After the created method was verified to be working as designed, it could be compared to the current situation at SDOZ. In the current situation the ROS tanks are not scheduled beforehand, rather when a product arrives from the refinery the tankpark operators will decide on the spot which tank is to be used. This is often based on the availability of pipelines and tanks. In this process empty tanks are preferred over tanks which already store an amount of the same material, in order to reduce the amount of quality assurance checks needed. When a shipment of 20 tonnes is to be loaded, and there are three tanks containing this material, storing respectively 5, 15 and 30 tonnes, it is often opted to only use material of the third tank, rather than emptying the first two. This is mainly because there is virtually no regulation as to which tank is to be used, and switching between tanks requires more work of the operators. As discussed in chapter 1 this is sub-optimal, something that should be prevented using the created method. To compare the developed method with the current situation the example of section 4.5 is expanded. This is done to make sure that multiple solutions, including sub-optimal solutions, are possible. While the shipments and materials of the system remain the same as the verification example, there are two new tanks added: Tank 4 which will make use of the same pump as Tank 3, and Tank 5 which is only connected in loadinghall spot 2. The new layout can be seen in Figure 5.1.

The new connections can be found in Table 5.1. Noticeable is the addition of the pump of group 2. In the verification example it could not create any conflicts and was omitted, but since Tank 4 uses the same pump it has to be included in the new example.

Name	From	To	Transport devices
Tr-1	Steamers	Tank1	-
Tr-2	Steamers	Tank2	-
Tr-3	Steamers	Tank3	-
Tr-4	Steamers	Tank4	-
Tr-5	Steamers	Tank5	-
Tr-6	Tank1	LoadingHall1	Pump Group 1
Tr-7	Tank2	LoadingHall1	Pump Group 1
Tr-8	Tank1	Mixingtank	Pump Group 1, Mixingtank input valve
Tr-9	Tank2	Mixingtank	Pump Group 1, Mixingtank input valve
Tr-10	Tank3	LoadingHall2	Pump Group 2
Tr-11	Tank4	LoadingHall2	Pump Group 2
Tr-12	Tank3	Mixingtank	Pump Group 2, Mixingtank input valve
Tr-13	Tank4	Mixingtank	Pump Group 2, Mixingtank input valve
Tr-14	Tank5	LoadingHall2	-
Tr-15	MixingTank	LoadingHall1	-
Tr-16	MixingTank	LoadingHall2	-
Tr-17	LoadingHall1	Preloader	preloader truck
Tr-18	LoadingHall2	Preloader	preloader truck

Table 5.1: Transport lines used for the example

Figure 5.3 shows a comparison between the Gantt chart created using the current method applied at SDOZ and the Gantt chart made by applying the created method. It can clearly be seen that while the current method needs almost every tank, the new method manages to keep some tanks empty for almost the complete timehorizon. This can also be seen in Figure 5.2 in which the amount of free tanks, and amount of usable volume is plotted for both methods. The total free volume is calculated by multiplying the time a tank is completely empty by the capacity of the tank for each tank. The current method has a total of 878.2125 tonnes×hours free, while the method developed in this thesis results in 1567.965 tonnes×hours free volume. This means that applying the developed method to this example case would result in 178.5% of the current free usable volume. It is expected that when applying the developed method to the SDOZ ROS similar results will be found.

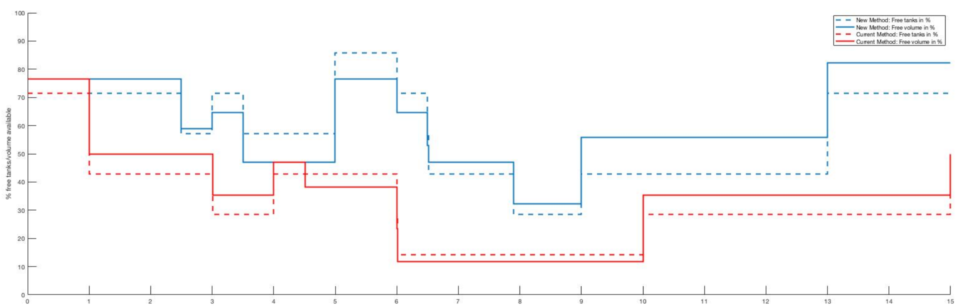
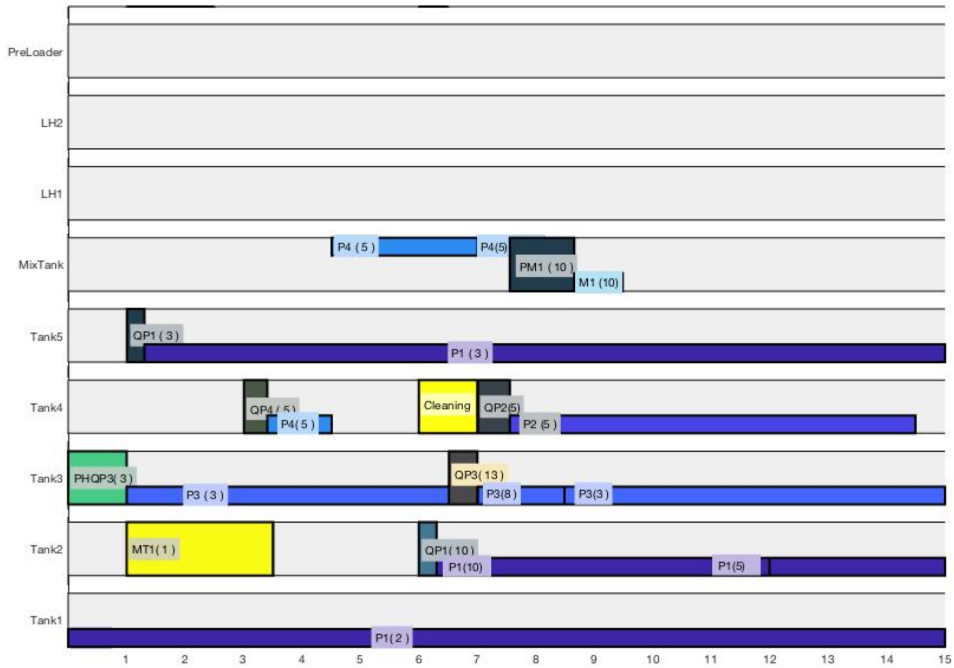
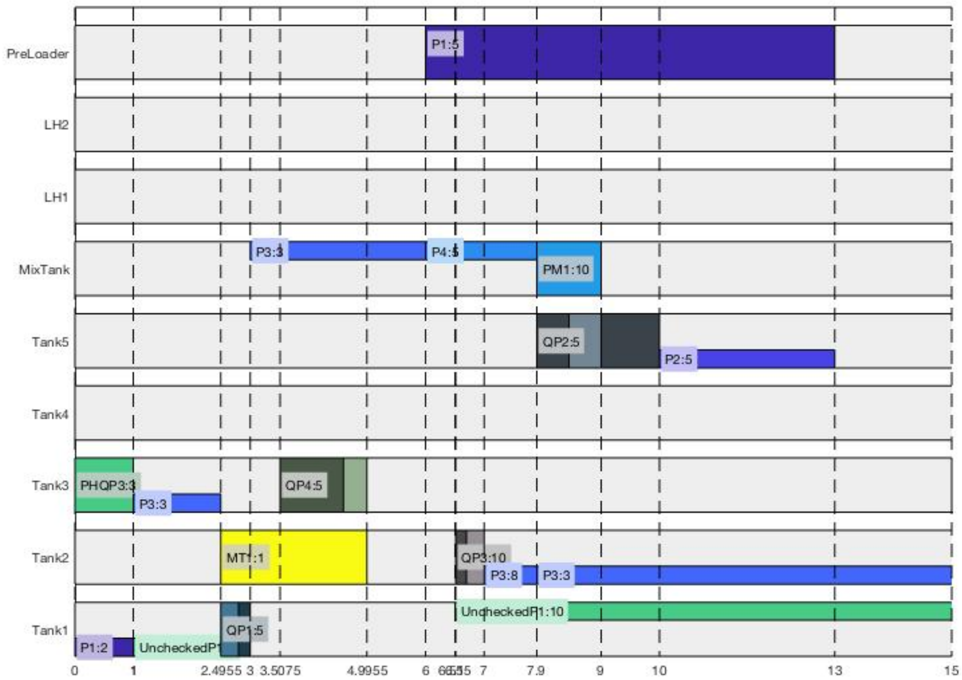


Figure 5.2: Comparison of free tanks and free usable volume between the current situation and the developed method



(a) Schedule made by hand



(b) Schedule made by the developed method

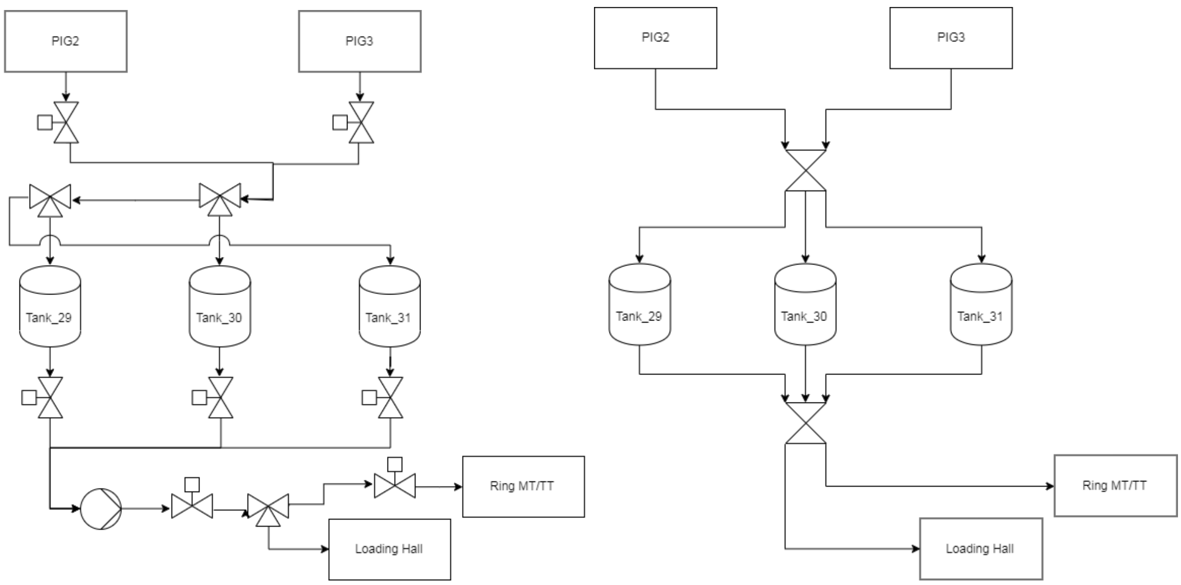
Figure 5.3: Comparison between the Gantt charts made by the created method and by hand.

5.2. Application

After the verification and validation of the method, the application is the next logical step. In the next sections a possible setup that could be used for this will be discussed. In order to apply the created method to the SDOZ ROS, the ROS and its connections first had to be mapped. This is done in section 5.2.1. This information is used to create a model of the complete ROS in Section 5.2.2.

5.2.1. Mapping the ROS

The complete ROS consists of a large number of pipe segments, valves and pumps. If all these transfer devices had to be modelled the system would be extremely large and hard to calculate. While a connection between for example a tank and a loading spot can contain multiple valves and pipe sections, only those were considered that overlapped with another connection. This made for a clearer overview of the connection and a smaller model, which results in an increase in calculation speed without losing accuracy. An example of this can be seen in Figure 5.4.



(a) The real layout (b) The simplified layout

Figure 5.4: Group 4 before and after simplification

The figure shows tank group 4, consisting of three tanks which can be filled from either PIG line 2 or 3. The tanks can be emptied by using the pump of group 4, pumping material to either the loading hall or the ring MT/TT (the ring line supplying the Mixing and Turbo Tanks). In Figure 5.4a all the physical connections and valves in group 4 are displayed. Modeling all of these would create 11 transfer devices (10 valves and 1 pump). But since all inflowing material has to pass the valve above Tank 30, and all out-flowing material has to pass the pump, at each moment only one tank can be filled and only one tank can be emptied. Using this information the simplification in Figure 5.4b was created. In this Figure only two transfer devices or choke-points are needed. This type of simplification was applied to the whole ROS resulting in the schematic presented in Figure 5.5. A larger version of the image can be found in Appendix F.

5.2.2. Modeling the ROS

The simplified ROS was then translated into a list of all possible connections along with the transfer devices included by these connections. This resulted in 1306 unique connections. These connections were combined with information about the tanks and the materials supplied by SDOZ to create the input for the model. To simulate the processing in the refinery itself, each of the four steamers was modeled as a simple tank with zero storage, in which shipments are delivered. Since the storage of the tanks is zero, the material cannot be stored in the steamer, and has to be moved to a tank directly.

The next step was to model all the different products and blends that can be made by SDOZ. There are 202 different products that can be produced by SDOZ, and there are 425 different blends that are

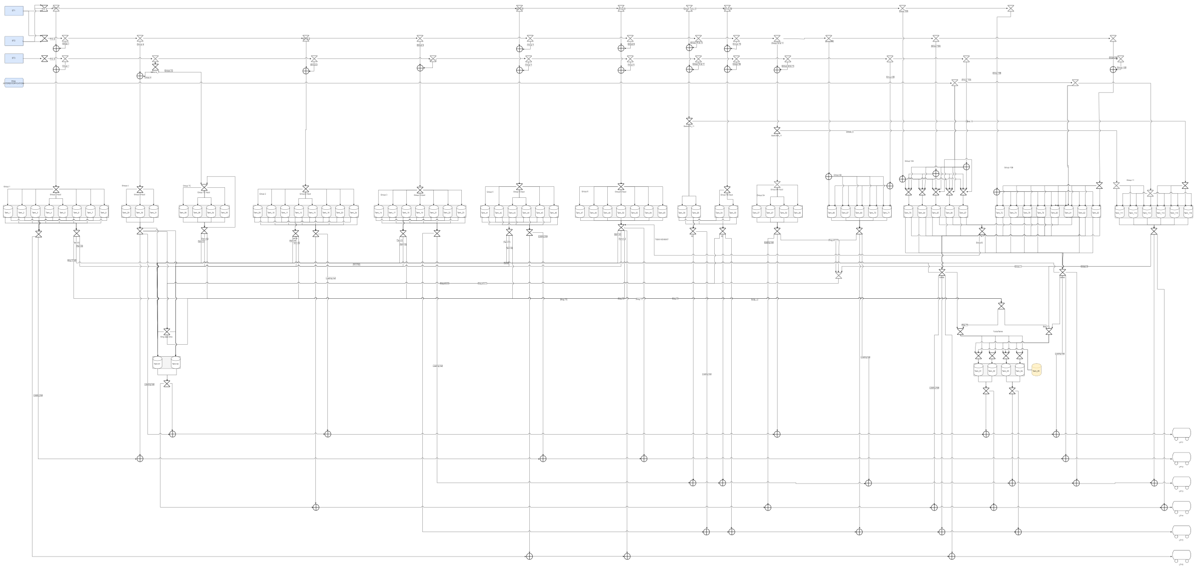


Figure 5.5: The complete ROS in simplified layout

made in the mixing and turbotanks, bringing the total amount of materials in the system to 627.

Next the shipments, both ingoing and outgoing, of an actual production day (16-09-2019) were added to the setup. This included 65 shipments. A summary of the setup can be found in Table 5.2.

Units	97	87 tanks, 4 steamers, 6 loading spots
Connections	1306	
Materials	829	202 products, 202 virtual unchecked products, and 425 blends
Shipments	65	17 ingoing, 48 outgoing

Table 5.2: Summary of the system

While this setup is theoretically possible to calculate, in reality this proved to be significantly too big for the calculation power available. It was therefore opted to only include the materials that were used in the time horizon, and exclude any tanks that were (partially) filled with materials that were neither produced, blended, or shipped within the timehorizon. When this model still proved to be too large to calculate, the decision was made to exclude the pipelines and model instantaneous transfers. Since the pipelines are currently not scheduled this should not cause great differences between the modelled schedule and the current schedule of the tanks.

Unfortunately even after the omission of the pipelines the model was still too big to optimise using the available computing power. To solve this a split was proposed. Due to company policy some of the tanks can store only laurics materials, others only non-laurics materials, and others are able to store both. The system can thus be split into two overlapping systems. One system for laurics oils, containing laurics materials, the dedicated laurics tanks and the shared tanks. The other system for only non-laurics materials, the dedicated non-laurics tanks and the shared tanks aswell. While not optimal it could be possible to first schedule one of the systems, and to use the result of this to mark some of the shared tanks as occupied, in order to schedule the second system.

Using these subsystems it was possible to create a schedule for one day for only the laurics materials, using the dedicated laurics tanks and shared tanks. However, since the model size was still very big, it was opted not to calculate to optimally but to set a calculation time limit of 24 hours. After the optimisation was run an answer with an optimallity gap of 76.6%, was found. In Chapter 4 it could be assumed that the answer, while the optimisation gap was not zero, would not improve significantly over time. The reason for this was that there had not been found any new answers for a significant time, and that the biggest contribution to closing the optimation gap was the decreasing best bound. In this situation this is not the case: as Figure 5.6 shows, there are still solutions being found. Furthermore the decrease in best bound is less than the increase in best solutions, contrary the case in Chapter 4.

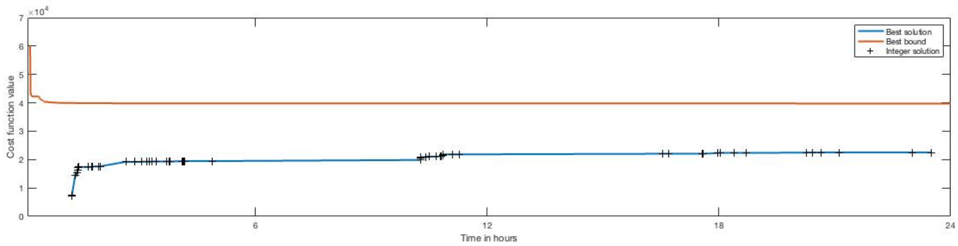


Figure 5.6: Solution progress over time

The best result after this day of calculations was a feasible, yet sub-optimal scheduling of the shared tanks. However, since the shared tanks were not scheduled as good as they possibly could have been, this meant that there were not enough shared tanks free in order to be able to calculate the second system. Switching the order of operations, and scheduling the non-laurics oils first created a similar situation, preventing the laurics oils from being scheduled.

5.3. Summary

In Chapter 4 it was confirmed that the developed method works as designed. The example used for this verification was expanded in this chapter, to ensure that multiple (sub-optimal) solutions were possible. This new example was then scheduled as currently done at SDOZ, and with the developed method. By comparing the two methods using an example it was confirmed that using the method developed in this thesis results in 178% of the usable free storage available compared to using the current method. since the example was made to be similar to the real life situation it is expected that similar increases will be found when the method is applied to the SDOZ ROS. More available free storage means that the total amount of production of the refinery could be increased, leading to more profits. After this result the real-life application of the developed model at the SDOZ ROS was discussed. This started with mapping the ROS and all the connections within. Unfortunately the complete model of the SDOZ ROS turned out to be too large to optimise using the available calculation power. Therefore some reductions were applied in order to create a proof-of-concept version of a schedule. After many reductions, including the omission of the pipelines, a schedule could be made for the laurics products. These product were scheduled in the dedicated laurics tanks and the shared tanks. However, again due to limitations in calculation power, this schedule was not optimal and the result of this could not be used to schedule the non-laurics products. The next chapter will discuss the contributions of this research, both to the academics and to the company, and will further go into the limitations that prevented the complete ROS model from being optimised.

6

Conclusions, discussion and recommendations

6.1. Conclusions

The main conclusion of this report is that is it possible to optimise the scheduling of the SDOZ ROS by using the method developed. As stated in the introduction of this thesis, the current way of assigning tanks at SDOZ is to do so by hand, and not beforehand. Since this is believed to be not effective, the scheduling problem of the SDOZ ROS was analysed and compared to scheduling methods in the literature. This led to the conclusion that "A novel network-based continuous-time representation for process scheduling" [12] was the best suited method. While the method was the best suited, still some improvements were needed, in order to make the method applicable to the SDOZ ROS. The first adaptation needed was the use of loading spots and preloaders. While the method in the literature assumed that material can be shipped from anywhere in the system, in reality this is not the case. In order to ship material it first has to be transported trough a loading spot, to either a preloading trailer or a waiting transporting truck. Loading spots were modelled as special zero volume shared storage units. Preloaders were subjected to an extra constraint, dictating that when a shipment takes place on a preloader, it is completely emptied. The original constraint governing shipments was modified to restrict shipments to loading spots en preloaders only. The next addition made was the need for quality assurance checks. As the products made at SDOZ are mainly destined for the food industry, quality checks are vital. Finally a cost function was designed in such a way that its minimization would result in maximising the amount of usable free volume, while minimising the amount of costly quality checks. The system has been validated by comparing the performance of the current tank assignment method to the developed method, using an example case. In this example case it was proven that the developed method has a significant better performance then the now used method of tank assignment. Using the developed method will therefore increase the amount of available free volume in the ROS, without adding more tanks. However this is not a goal itself. The real goal is to increase profits of the refinery. Increasing the amount of available storage can help this goal in two ways. First by having more available free storage, the production of the refinery can be increased, allowing SDOZ to sell more products and make more profits. Secondly, since more storage is available, product can be made longer beforehand. This prevents that delays within the refinery can cause delays in shipments. These delays could lead to a lower customer satisfaction and even to late fees.

6.2. Contributions of this research

This thesis contains several contributions to both the academics as to the company. The first contribution was the fixes made to the existing method of the literature, described in section 3.2.2 and Section 3.3.6. Without these fixes the method cannot function as described by the literature. The second contribution to the academics was the addition of extra features, as described in chapter 4.

These extra features were made for the specific situation at SDOZ, but are made in such a way they can be applied in general. Mainly the addition of the loading spots and preloaders is believed to be

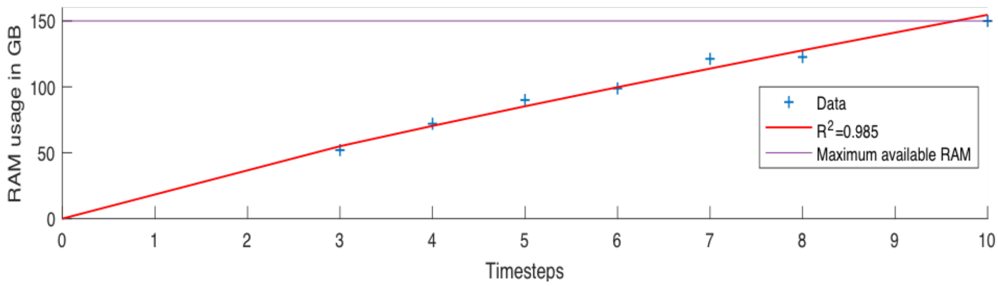


Figure 6.1: Amount of RAM needed vs amount of time steps used

a common situation found in many production facilities. Furthermore, due to the way the method is set-up, it can also be used when the selection of loading spots is not an issue, or when there are no preloaders to be used. This makes the method flexible and applicable to a wider range of situations than the method in the literature, without limiting its use to specific scenarios.

The main contribution to the company is the confirmation that the current mode of operation of the ROS, which is not to schedule the use of tanks beforehand, is highly ineffective. In chapter 5 a comparison was made between the current way of operation and the developed method, using an example case. This resulted in significantly more usable volume being available using the developed method. It is therefore recommended for the company to invest in a scheduling solution for the ROS, rather than to invest in the addition of extra tanks to the tank park. This will increase the amount of usable storage available, allowing for more production and therefore more profits. Secondly, while less influential than the previous contribution, the creation of the complete simplified overview of the ROS (Appendix F) that was created for this thesis, can be used to better understand the connections within the ROS. This information can be used for example to dedicate tanks to specific materials.

6.3. Limitations of this research

As mentioned earlier there were some limitations in this project, these were mainly on the computing side. The first limitation was the amount of Random Access Memory (RAM) available. While there was 150 Gigabytes of RAM available, this was simply not enough. To optimise the complete ROS for one day only, an absolute minimum of 24 timesteps could be used. This would mean that shipments could only be send and arrive on the full hour, which is of course not optimal. However, even with this bear minimum of timesteps not enough RAM was available. To have an indication on how much RAM was needed, the system was run without including the pipelines and including only one shipment, increasing the amount of timesteps and noting the amount of RAM needed. While the omission of the pipelines can have an affect on the feasibility of the schedules generated, the inclusion of the pipelines is the greatest contributor to the size of the model. Since every pipeline has to be modelled as transfer unit, this would mean that the total amount of units would go up from 97 (4 steamers, 87 tanks, and 6 loading spots) to 1403 (4 steamers, 87 tanks, 6 loading spots and 1306 pipelines). The amount of shipments was reduced to one in order to have feasible results at small amounts of timesteps. The decrease in shipments decreases the system somewhat in size, this is however not significant compared to the increase caused by the extra timesteps. Figure 6.1 shows the amount of RAM needed for the first 10 timesteps, after this the limit of 150GB was reached. If this data is extrapolated to the minimum of 24 timesteps needed to model the complete list of shipments, it is found that around 350GB of RAM is needed.

The second limitation was the calculation time needed to find an optimal solution, meaning zero optimality gap. As mentioned in Chapter 4 and again in Chapter 5, the optimality gap is the gap between the best bound and the best feasible solution found. The gap is closed from both sides: on one side the best bound is reduced by cutting infeasible branches of the solution tree, on the other side the gap is closed by finding better feasible solutions. The rate at which the optimality gap is closed, and optimality is reached is highly dependent of the size of the model and the solving algorithm used.

As seen in the results of the examples run in Chapter 3, shown in Appendix E, the time needed to solve the model increases significantly with the amount of time steps used. In this thesis the Gurobi solver was used. [13] The decision to use the Gurobi solver was made after recreating the examples of the literature with the integrated MatLab solver. Solving these rather small examples to optimality costed significant CPU time. To reduce this CPU time the integrated solver was compared with two of the most used solvers, CPLEX made by IBM, and the open-source Gurobi solver [16]. The comparison was done by running the first example of Chapter 3 with all three solvers, increasing the amount of timesteps until the required CPU time was larger than 3600 seconds. The result of this comparison can be found in Figure 6.2.

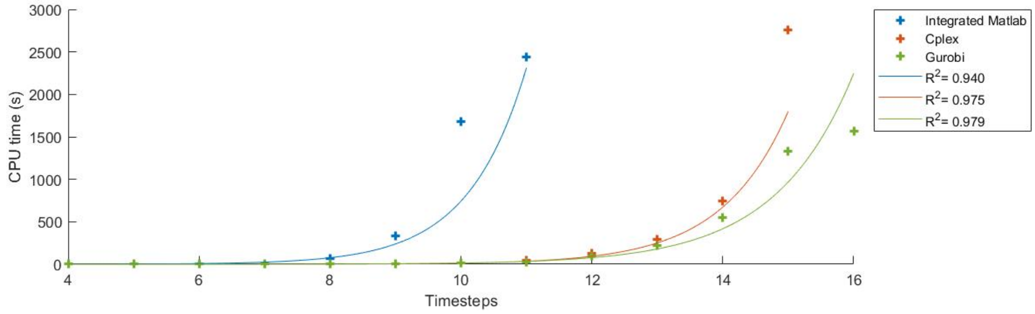


Figure 6.2: Comparison between the amount of time steps used and the CPU time for Matlab, CPLEX and Gurobi

Since there is no feasible answer possible using less than 4 timesteps, the graph starts at 4. For each solver an exponential trend line is shown. It can be seen that, while all solvers show exponential growth when increasing the amount of timesteps, the integrated Matlab solver uses significantly more calculation time. This shows that the choice of solver has a large influence on the CPU-time needed. What is not clearly visible in this graph, but is remarkable, is that in the first timesteps (timesteps 4 through 10) CPLEX was actually faster then Gurobi. However as the model size increased the calculation time of CPLEX grew faster than that of Gurobi. The difference started small at timestep 10 with CPLEX at 18 seconds and Gurobi at 16 seconds (and Matlab at 1680 seconds), but quickly grew larger. For example the model with 15 timesteps took CPLEX 2756 seconds and Gurobi 1327 seconds. A logarithmic plot of the same graph can be found in Figure 6.3, showing clearly that while Gurobi starts slower it is faster in larger models.

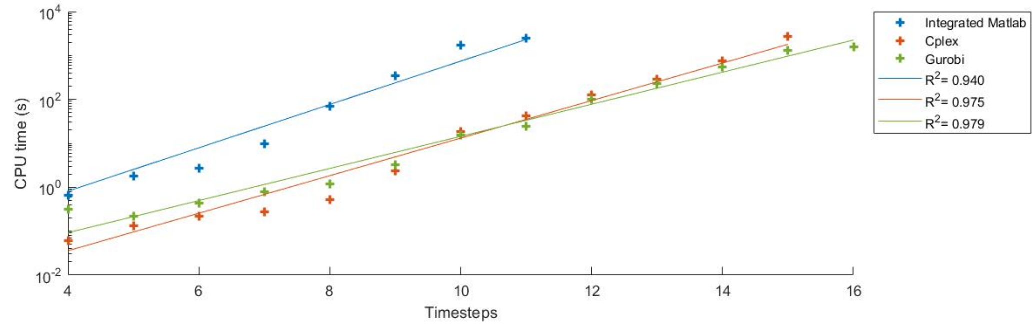


Figure 6.3: Comparison between the amount of time steps used and the CPU time for Matlab, CPLEX and Gurobi on a logarithmic scale

Since only three algorithms were tested, a recommendation for further research is to look into different solvers available in order to find the fastest option to reach the optimal solution.

6.4. Recommendations for future research

As mentioned in the previous section the main limitation of this research was the amount of computational power available. It is therefore recommended to either continue this research with more computational power, or to research better optimisation solver algorithms.

If this succeeds it is recommended to test several layouts for the SDOZ ROS to see if changing the pipelines can improve the performance. If this is not possible a possible research could include a discrete time model. While this might include a sacrifice on the part of optimality, it is possible that this would take less calculation power to calculate.

Since this project focused on planning the ROS of SDOZ, another recommendation is to use the knowledge of this project to create a scheduling methods for the complete facilities at SDOZ. While some obstacles are still to be overcome, such as the unreliable production times in the refinery, it is expected that much gain can be had by scheduling the complete refinery at once.

Another research could focus on these unreliable production times instead. It is known that the uncertainty of the production times is mainly dependent on the quality of input products. By measuring this quality, or by inferring it from the duration of other processes, an estimation of the processing times could be made. Since this would still be an estimation and not a certainty, the method would have to be a stochastic method, contrary to the method of this thesis which was deterministic.

Finally, in the current method it was assumed that all batch sizes were continuous variables. However in some situation this is not the case. For example the Steamer lines at SDOZ can handle batches of 22.5, 30 or 37.5 tonnes. In order to handle this, the constraints governing the batch sizes should be adapted. How these adaptation would work, especially in combination with continuous batch sizes in the ROS, is an interesting subject for further research.

Bibliography

[1] H.B. Aditiya, W.T. Chong, T.M.I. Mahlia, A.H. Sebayang, M.A. Berawi, and Hadi Nur. Second generation bioethanol potential from selected malaysia's biodiversity biomasses: A review. Waste Management, 47:46 – 61, 2016. ISSN 0956-053X. doi: https://doi.org/10.1016/j.wasman.2015.07.031. URL http://www.sciencedirect.com/science/article/pii/S0956053X15300489. Biowaste to Fuel.

[2] Miguel Barrientos. Palm kernel oil monthly price - us dollars per metric ton. https://www.indexmundi.com/commodities/?commodity=palm-kernel-oil&months=60, 2019. Retrieved on 05-11-2019.

[3] Yusof Basiron and Chan Kook Weng. The oil palm and its sustainability. Journal of Oil Palm Research, 16(1), 2004.

[4] Philipp Baumann and Norbert Trautmann. A continuous-time milp model for short-term scheduling of make-and-pack production processes. International Journal of Production Research, 51(6):1707–1727, 2013. doi: 10.1080/00207543.2012.694489. URL https://doi.org/10.1080/00207543.2012.694489.

[5] Pietro Belotti, Pierre Bonami, Matteo Fischetti, Andrea Lodi, Michele Monaci, Amaya Nogales-Gómez, and Domenico Salvagnin. On handling indicator constraints in mixed integer programming. Computational Optimization and Applications, 65(3):545–566, 2016.

[6] Elisa Loeser Bill George and Tani Lee. Oilseeds: World markets and trade. Oilseeds: World Markets and Trade, Oct 2019. Retrieved on 07-11-2019.

[7] Pedro M Castro and Ignacio E Grossmann. From time representation in scheduling to the solution of strip packing problems. Computers & chemical engineering, 44:45–57, 2012.

[8] Pedro M. Castro, Lige Sun, and Iiro Harjunkoski. Resourcetask network formulations for industrial demand side management of a steel plant. Industrial & Engineering Chemistry Research, 52(36):13046–13058, 2013. doi: 10.1021/ie401044q. URL https://doi.org/10.1021/ie401044q.

[9] Pedro M Castro, Ignacio E Grossmann, and Qi Zhang. Expanding scope and computational challenges in process scheduling. Computers & Chemical Engineering, 114:14–42, 2018.

[10] Jaime Cerda. Optimization methods for batch scheduling. Argentina: The Institute of Technological Development for the Chemical Industry, National University of the Littoral, 2012.

[11] Christodoulos A Floudas and Xiaoxia Lin. Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. Computers & Chemical Engineering, 28(11):2109–2129, 2004.

[12] Diego M Giménez, Gabriela P Henning, and Christos T Maravelias. A novel network-based continuous-time representation for process scheduling: Part ii. general framework. Computers & Chemical Engineering, 33(10):1644–1660, 2009.

[13] LLC Gurobi Optimization. Gurobi optimizer reference manual, 2020. URL "http://www.gurobi.com". Retrieved on 05-01-2020.

[14] Maud Göthe-Lundgren, Jan T. Lundgren, and Jan A. Persson. An optimization model for refinery production scheduling. International Journal of Production Economics, 78(3):255 – 270, 2002. ISSN 0925-5273. doi: https://doi.org/10.1016/S0925-5273(00)00162-6. URL http://www.sciencedirect.com/science/article/pii/S0925527300001626.

[15] Iiro Harjunkoski, Christos T Maravelias, Peter Bongers, Pedro M Castro, Sebastian Engell, Ignacio E Grossmann, John Hooker, Carlos Méndez, Guido Sand, and John Wassick. Scope for industrial applications of production scheduling models and solution methods. Computers & Chemical Engineering, 62, 2014.

[16] IBM. Cplex optimizer, 2019. URL https://www.ibm.com/nl-en/analytics/cplex-optimizer. Retrieved on 05-01-2020.

[17] E. Kondili, C.C. Pantelides, and R.W.H. Sargent. A general algorithm for short-term scheduling of batch operationsi. milp formulation. Computers & Chemical Engineering, 17(2):211 – 227, 1993. ISSN 0098-1354. doi: https://doi.org/10.1016/0098-1354(93)80015-F. URL http://www.sciencedirect.com/science/article/pii/009813549380015F. An International Journal of Computer Applications in Chemical Enginnering.

[18] Georgios M. Kopanos, Carlos A. Méndez, and Luis Puigjaner. Mip-based decomposition strategies for large-scale scheduling problems in multiproduct multistage batch plants: A benchmark scheduling problem of the pharmaceutical industry. European Journal of Operational Research, 207(2):644 – 655, 2010. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2010.06.002. URL http://www.sciencedirect.com/science/article/pii/S037722171000408X.

[19] X Lin and C.A Floudas. Design, synthesis and scheduling of multipurpose batch plants via an effective continuous-time formulation. Computers & Chemical Engineering, 25(4):665 – 674, 2001. ISSN 0098-1354. doi: https://doi.org/10.1016/S0098-1354(01)00663-9. URL http://www.sciencedirect.com/science/article/pii/S0098135401006639.

[20] Iliana Magra. U.k. ad highlighting plight of orangutans is deemed too political to air. The New York Times, Nov 2018. Retrieved on 05-11-2019.

[21] Christos T. Maravelias. chemical production scheduling: notation, problem classes, modeling approaches, and solution methods, 2014.

[22] Pablo A. Marchetti and Jaime Cerdá. An approximate mathematical framework for resource-constrained multistage batch scheduling. Chemical Engineering Science, 64(11):2733 – 2748, 2009. ISSN 0009-2509. doi: https://doi.org/10.1016/j.ces.2009.03.002. URL http://www.sciencedirect.com/science/article/pii/S0009250909001730.

[23] Pablo A. Marchetti and Jaime Cerdá. A general resource-constrained scheduling framework for multistage batch facilities with sequence-dependent changeovers. Computers & Chemical Engineering, 33(4):871 – 886, 2009. ISSN 0098-1354. doi: https://doi.org/10.1016/j.compchemeng.2008.12.007. URL http://www.sciencedirect.com/science/article/pii/S0098135408002627.

[24] Pablo A. Marchetti, Carlos A. Méndez, and Jaime Cerdá. Simultaneous lot sizing and scheduling of multistage batch processes handling multiple orders per product. Industrial & Engineering Chemistry Research, 51(16):5762–5780, 2012. doi: 10.1021/ie202275y. URL https://doi.org/10.1021/ie202275y.

[25] Carlos A Méndez, Jaime Cerdá, Ignacio E Grossmann, Iiro Harjunkoski, and Marco Fahl. State-of-the-art review of optimization methods for short-term scheduling of batch processes. Computers & chemical engineering, 30(6-7):913–946, 2006.

[26] Christian Nellemann. The last stand of the orangutan: state of emergency: illegal logging, fire and palm oil in Indonesia's national parks. UNEP/Earthprint, 2007.

[27] Steffen Noleppa. Auf der Ölspur Berechnungen zu einer palmölfreieren Welt. Umweltstiftung WWF - Deutschland, Berlin, 2016. ISBN 978-3-946211-05-1.

[28] Mh Oladeinde, Ai Momodu, and Ca Oladeinde. Flowshop scheduling using a network approach. Nigerian Journal of Technology, 34(2):286, 2015. doi: 10.4314/njt.v34i2.11.

[29] Pradeep Prasad and Christos T. Maravelias. Batch selection, assignment and sequencing in multi-stage multi-product processes. Computers & Chemical Engineering, 32(6):1106 – 1119, 2008. ISSN 0098-1354. doi: https://doi.org/10.1016/j.compchemeng.2007.06.012. URL http://www.sciencedirect.com/science/article/pii/S0098135407001640.

[30] James B. Reeves, John L. Weihrauch, Consumer (U.S.), and Food Economics Institute. Composition of foods : fats and oils : raw, processed, prepared. U.S. Dept. of Agriculture, Science and Education Administration, [Washington, D.C.], 1979.

[31] Paul A. Rubin. Perils of "big m", Jul 2011. URL https://orinanobworld.blogspot.com/2011/07/perils-of-big-m.html. Retrieved on 07-12-2019.

[32] Alan S. Manne. On the job-shop scheduling problem. Operations Research, 8:219–223, 04 1960. doi: 10.1287/opre.8.2.219.

[33] Sharad Sharma. Lean services: Doing transactions right the first time. https://www.isixsigma.com/methodology/lean-methodology/lean-services-doing-transactions-right-first-time/, Jun 2012. Retrieved on 09-09-2019.

[34] Marcel van Wing. Onze missie: The flower farm. https://theflowerfarm.nl/onze-missie, 2019. Retrieved on 06-11-2019.

[35] Sathia Varqa. Essential palm oil statistics 2017. Palm Oil Analytics. Retrieved, 20:2018, 2017.

[36] Willy Verheye. Growth and production of oil palm. In Willy H Verheye, editor, Land use, land cover and soil sciences, Encyclopedia of Life Support Systems (EOLSS). UNESCO-EOLSS Publishers, 2010.

[37] Ian Webster. Inflation rate between 2014-2019: Inflation calculator. http://www.in2013dollars.com/us/inflation/2014?amount=1, 2020.

[38] Marco Weiss. Reduction rework & waste refined oils & fats, feb 2013.

[39] Laurence A. Wolsey. Mip modelling of changeovers in production planning and scheduling problems. European Journal of Operational Research, 99(1):154 – 165, 1997. ISSN 0377-2217. doi: https://doi.org/10.1016/S0377-2217(97)89646-4. URL http://www.sciencedirect.com/science/article/pii/S0377221797896464.

[40] Dan Wu. Unified Frameworks for Optimal Production Planning and Scheduling: Continuous-time Decomposition-based Approaches. PhD thesis, Citeseer, 2005.

[41] Luis J. Zeballos, Juan M. Novas, and Gabriela P. Henning. A cp formulation for scheduling multiproduct multistage batch plants. Computers & Chemical Engineering, 35(12):2973 – 2989, 2011. ISSN 0098-1354. doi: https://doi.org/10.1016/j.compchemeng.2011.01.043. URL http://www.sciencedirect.com/science/article/pii/S0098135411000536.

A

Scientific paper

Tankpark scheduling optimisation

P.D. Baart BSc
University of Technology Delft
Transport Engineering and Logistics

Dr. ir. X. Jiang
University of Technology Delft
Transport Engineering and Logistics

W. van der Tempel MSc
Sime Darby Oils Zwijndrecht

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

May 3, 2020

Abstract

In this paper an improvement on the existing scheduling optimisation method "A novel network-based continuous-time representation for process scheduling" is presented. Existing features were combined, and new features were added. The new features consisted of the use of loading spots and preloaders, and the need for quality assurance checks. Furthermore a new objective function was designed such that the optimisation would create schedules that maximize the amount of usable free volume, while minimizing the amount of costly quality checks. The new method was first verified after which it was validated by comparing the developed method with the current way of working at Sime Darby Oils Zwijndrecht. This comparison shows significant improvements, which leads to the recommendation of implementing the developed method.

I. Introduction

This project was done in an assignment for Sime Darby Oils Zwijndrecht (SDOZ), a processor of vegetable based oils mainly for the foodstuff industry. In order to reduce costs a project was started to optimise the scheduling within the refinery. A stakeholder analysis revealed that the refined oil storage (ROS), was a possible point of improvement. Currently the tanks within the ROS are not scheduled beforehand, rather when a material is produced by the refinery, operators of the tankpark decide on the spot which tank is to be used. An internal research shows that during a standard production week most tanks are in use, while only half of the volume is occupied. It is suggested that by making an optimised schedule for the ROS would increase the amount of usable free volume, which is defined as the combined volume of all empty tanks. By increasing the amount of usable free volume the production and therefore the profits of the refinery could be increased. The goal of the project was thus to increase the profitability of SDOZ, by creating a model which can optimise the schedule for the storage tanks, pipelines, loading places, and pre-loaders, while adhering to the constraints of the facility. By optimising this schedule, more effective storage volume will be available, allowing for more production and more profits. The schedule should also minimise the amount of costly quality checks needed. In order to fulfill this goal, the following research question was formulated:

> How can the SDOZ ROS schedule be optimized using a mathematical model?

In answering this question a solution to the scheduling problem of SDOZ should be found.

The scope of this project starts at the inflow of the ROS, and ends with the transportation of finished materials. This includes tank assignment, preloading, and the scheduling of the loading spots in the loading hall. An overview of the scope can be seen in Figure 1.

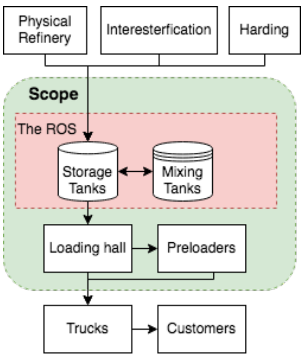


Figure 1: Overview of the scope of this project

To accurately scope this project not only physical limitations have to be taken into account but also some other assumptions had to be made:

- Strict schedule adherence
- Transport on time
- First-Time-Right production
- Linear processing and transport times
- Production as needed and on time

The first assumption is that the schedules created will be strictly adhered. The second is that all transport arrives on time. While this is not always the case, this can be done by creating agreements with the transporting companies, and by the use of pre-loaders. The third assumption is that all products are produced "First-Time-Right", meaning all quality checks succeed. If not, the schedule should be remade, accounting for processes already started and/or finished. It is also assumed that the processing and transport time of batches is linearly related to their size. The final assumption is that the refinery will be able to produce as needed and on time.

II. A novel network-based continuous-time representation for process scheduling

The first step in designing the optimisation method was to search the literature for existing applicable methods. This has been done by describing the system at the SDOZ ROS in thirteen different characteristics, as described by [1]. A comparison between methods found in the literature and these characteristics, led to the selection of "A novel network-based continuous-time representation for process scheduling" [2]. This method includes the use of pipelines for transport, can account for contamination and cleaning and can handle previous-horizon tasks, which are needed for rescheduling. The process topology is a network, allowing for the mixing and splitting of batches, which is essential for the operating of the ROS. Time is represented in a continuous way. This means that while the amount of timesteps should be set a priori, the length of each timestep will be decided by the optimisation. The method is a linear method, resulting in a Mixed Integer Linear Programing problem (MILP).

III. Improvements to the method

In order to better fit the situation at the SDOZ ROS the method had to be expanded. First the features given in the literature had to be combined into one method, as the literature does not mention the combination of features. The newly created features in this section were all implemented on this combined method.

i. Loading spots and preloading

The ROS at SDOZ makes use of a loading hall to load material into trucks. When a material has to be loaded, a loading spot has to be selected. Since this selection influences other shipments, by occupying either pipe segments or the loading spots themselves, it should included in the optimisation. The loading spots are modelled as zero volume shared storage units, meaning material cannot be stored in the loading spot, mimicking the real world situation. The original method allowed for shipments directly from storage tanks. Equation 1 shows the existing constraint that indicates shipments can be done by all units storing the material, indicated by the set $J^S_{m(l)}$.

$$\sum_{j \in J^S_{m(l)}} \sum_{n \in N} O_{l,j,n} = 1 \qquad l \in L \tag{1}$$

This constraint has been changed to the constraint shown in Equation 2, in which $J^L_{m(l)}$ indicates the set of loading spots and preloaders capable of loading the material.

$$\sum_{j \in J^L_{m(l)}} \sum_{n \in N} O_{l,j,n} = 1 \qquad l \in L \tag{2}$$

Some materials can be preloaded into so called preloader trucks. These can be stored on site and be swapped out for empty trucks without occupying the loading spot at the time of shipment. In order to model this, special shared storage units are created using an extra constraint, dictating that the unit can only be completely emptied at once.

ii. Quality assurance

When material is added to a tank, the quality of the material first has to be checked, before it is allowed to be used in a process or pumped further. The processing time of these checks is dependant on the material, not the batch size. This is modelled by creating virtual "unchecked" variants of each material. QA-processes are added, able to convert the unchecked to checked material. Since these processes have to be executed on the material while stored, tanks are now modeled as production units. When material is added to tanks already in use, the material that has already been checked, should first be marked as unchecked, since only one material can be stored at any given time. This is done by an additional process with zero processing time, undoing the quality assurance. Some products have to be checked after each transfer. In this case the transfer process is used to undo the quality assurance.

iii. Objective function

The objective function created for the situation at SDOZ shown in Equation 3.

$$F = \sum_{n \in N} \sum_{i \in (I^C \cap I^{QA})} X_{i,j(i),n} - \sum_{n \in N} \sum_{j \in J} T^W_{j,n} \beta^{MAX}_j \tag{3}$$

Minimising this function will maximise the amount of free available storage while minimising the amount of quality assurance checks.

IV. Verification

In order to test if all the functions behave as planned, a small example has been made. In this example the setup shown in Figure 2 is analysed. The setup is based on the components of the real ROS.

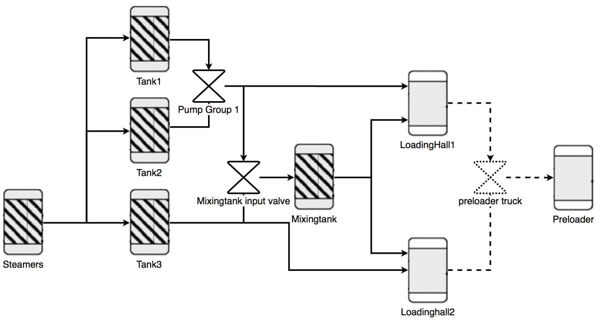


Figure 2: Layout of the verification example

The example considers four materials and four virtual unchecked materials. The example also considers two blended materials, M1 and M2, which can be produced in the mixing tank. In the test case, multiple products are produced by the steamers and multiple products, and one blend, are to be shipped. The shipments and their timings can be found in Table 1.

Shipment	Time	Material	Amount	From/to
SH1	1	UncheckedP1	+3	Steamers
SH2	3	UncheckedP4	+5	Steamers
SH3	6	UncheckedP1	+10	Steamers
SH4	6.5	UncheckedP3	+10	Steamers
SH5	7	UncheckedP2	+5	Steamers
SH6	9	P3	-5	LoadingHall 1 or 2
SH7	10	M1	-10	LoadingHall 1,2, or Preloader
SH8	13	P1	-5	LoadingHall 1,2, or Preloader
SH9	15	P2	-5	LoadingHall 1,2, or Preloader

Table 1: Shipments used for the example

At the start of the time horizon Tank 1 will contain 2 tonnes of P1, and Tank 3 will be performing quality assurance checks on 3 tonnes of P3. Furthermore a demand for at least 2.5 hours of maintenance between 1h and 5h for tank 2 is added.

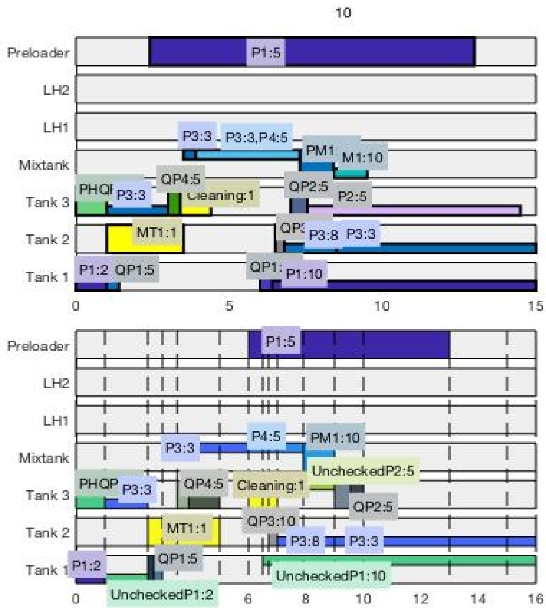


Figure 3: Verification: Expected results (top) vs actual results (bottom)

In Figure 3 the result of this example is shown, compared with the expected result. As can be seen they are not identical. This is because of two reasons. Firstly because, while the system accounts for late beginnings of the transport processes, this is not visualised in the Gantt chart. This is inconvenient for reading the chart, but has no influence on the actual result of the optimisation. The second reason the charts are not identical is that this is not the optimal solution to this example. Calculating an optimal solution would take to much CPU-time: The first feasible solution was found after 4298s. The objective function value was -451.5950 and the optimality gap was 172%. When the time limit was reached

(96 h) the best solution had an objective function value of -468.7451 and the optimality gap was reduced to 13.4 %. It is likely that if the system had run for longer that the best solution would not have improved by much, rather the best bound would have decreased to close the optimality gap.

While the solution shown in Figure 3 is not optimal it clearly shows all the implemented functions working together. The previous horizon tasks, maintenance tasks, and cleaning tasks can be seen working at Tanks 2 and 3. Shipment SH8 was was preloaded to the preloader and shipped from there. The quality assurance and reverse quality assurance can be seen in Tank 1.

V. Validation

After the created method was verified to be working as designed, it could be compared to the current situation at SDOZ. To compare the developed method with the current situation the example of section IV is expanded. This is done to make sure that multiple solutions, including sub-optimal solutions, are possible. The new layout can be seen in Figure 4.

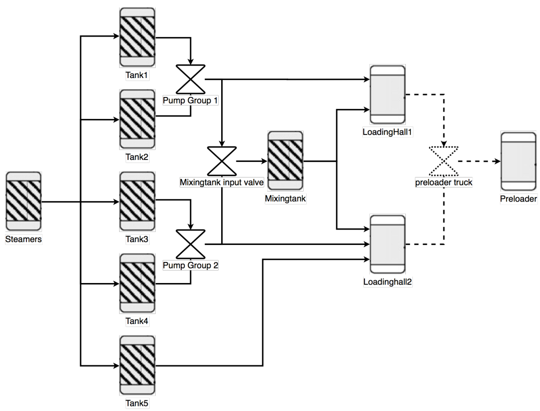


Figure 4: Layout of the example for validation

Figure 5 shows a comparison between the Gantt chart created using the current method applied at SDOZ and the Gantt chart made by applying the created method. It can clearly be seen that while the current method uses almost every tank, the developed method manages to keep some tanks empty for almost the complete timehorizon. The current method has a total of 878.2125 tonnes×hours free, while the method developed results in 1567.965 tonnes×hours free volume. This is an increase of 78.5%

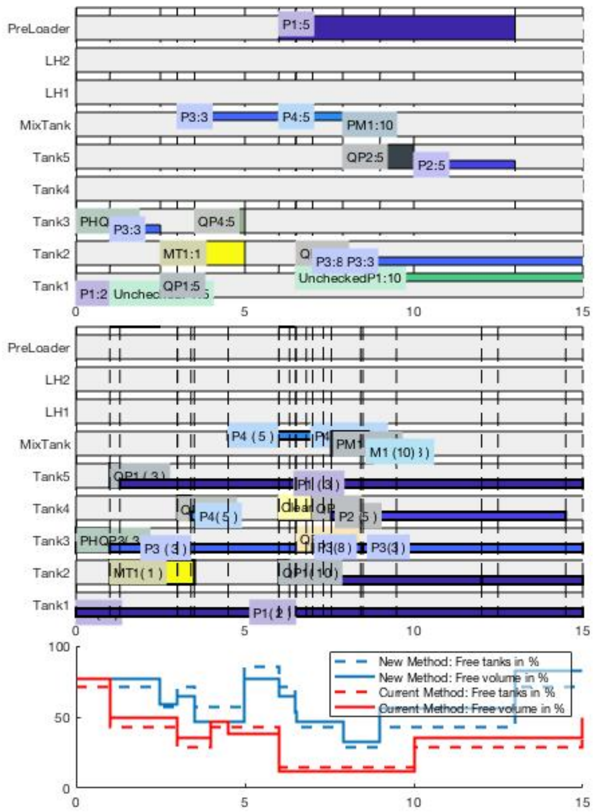


Figure 5: Comparison between the developed method (top) the current situation (middle)

VI. Application

After the verification and validation of the method, the application is the next logical step.

In Table 2 a summary is given for the system that has to be created to schedule a single day of production for the ROS.

Units	97	87 tanks, 4 steamers, 6 loading spots
Connections	1306	
Materials	829	202 products, 202 virtual unchecked products, and 425 blends
Shipments	65	17 ingoing, 48 outgoing

Table 2: Summary of the system

While this setup is theoretically possible to calculate, in reality this proved to be significantly too big for the calculation power available. To create a proof of concept it was chosen to not model the pipelines and create a schedule for only one of the two types of products: The laurics products. All tanks that are dedicated to storing non-laurics products, or were filled with a product that is not delivered nor shipped in the time horizon, were omitted. While not optimal it could be possible to first schedule one of the systems, and to use the result of this to mark some of the shared tanks as occupied, in order to schedule the second system.

Using this subsystem it was possible to create a schedule for one day for only the laurics materials, using the dedicated laurics tanks and shared tanks. However, since the model size was still very big, it was opted not to calculate to optimally but to set a calculation time limit of 24 hours. After the optimisation was run an answer with an optimallity gap of 76.6%, was found. Since when the time limit was reached there were still solutions being found and the decrease in best bound was less than the increase in best solutions, it is likely that the solution found was not optimal. The best result found was a feasible, yet sub-optimal scheduling of the shared tanks. However, due to its sub-optimallity, there were not enough shared tanks free in order to be able to calculate the second material group, the non-laurics materials.

VII. Conclusion

In this paper an improvement to "A novel network-based continuous-time representation for process scheduling" was proposed. [2] It was shown in the verification that the new implemented features such as the use of loading spots and preloaders, as well as the need for quality assurance work. The validation showed that the use of this method could improve the amount of free space within the SDOZ ROS significantly. However, in order to implement this method further research into computational power and solver algorithms is needed.

References

[1] Carlos A Méndez, Jaime Cerdá, Ignacio E Grossmann, Iiro Harjunkoski, and Marco Fahl. State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Computers & chemical engineering*, 30(6-7):913–946, 2006.

[2] Diego M Giménez, Gabriela P Henning, and Christos T Maravelias. A novel network-based continuous-time representation for process scheduling: Part ii. general framework. *Computers & Chemical Engineering*, 33(10):1644–1660, 2009.

B
Stakeholder meetings

B.1. Planning department

The planning department is the department that makes the schedules and is thus the most important stakeholder. If a scheduling tool is designed the planners are the people who will use it, so their requests and demands are very important.

B.1.1. General information

In the eyes of the planners the production capacity is currently not limited by the planning. According to the planners there are simply not enough orders to run the Sime Darby Oils refinery at full capacity. This poses a problem for planning the deodorizer lines. Since the start-up time and costs of the deodorizers are high, it is decided to keep them running 24/7. A result of this is that when not enough orders are available the production of intermediate products, which can be done better and/or cheaper in other installations, is planned on the deodorizer lines in order to make use of the otherwise wasted operation time. The planners also noted that scheduled downtime for maintenance is not a problem, due to the lack of orders. During the day, changes made by the production department are discussed with the planning department before execution. During the nighttime operations these changes are completely in the hand of the production department. If management should decide to (partly) close the refinery in the weekends, the planners do foresee scheduling problems if the amount of orders, and the distribution of orders trough the week stays the same. An issue within the planning department is that a lot of constraints of the refinery are not documented but kept in the minds of the planners. This can cause great problems when one, or both planners, are unavailable, for example due to health problems.

B.1.2. Requests and demands

The planning department sees that each day up to two hours of the planners time is spent updating the schedule to account for changes in customers orders and delays. A request of the department is to have this automated to save time. Another request of the planning department is to have the quality of the intermediates taken into account in the schedule, as this has an influence on the cycle time and the amount of product consumed. The last request of the department is to have an integrated planning system that controls the piping and can start batches on the processing units. The role of the operator should become one of an observer. A demand of the planning department is the system should document all constraints that are currently not documented but kept mentally by the planners.

B.1.3. Bottlenecks

The bottlenecks that the planning department sees in the scheduling are directly linked to their requests and demand for the project. The first bottleneck is the human factor. The first reason for this is that there always a chance that human errors will result in a sub-optimal, or even an unfeasible schedule. Secondly, since not all constraints are mapped in the system, but a large portion of them is kept in the heads of the planners, the dependency is large. This is directly linked to the second bottleneck:The current scheduling program, named Fygir. Not all constraints are documented in this program, allowing for in feasible schedules. In the current situation this is countered by having the processing operator make changes in the schedule as he or she sees fit. This reduces the quality of the schedule and should be prevented in the future. The program also does not have an option to create schedules automatically. It also lacks integration to nor the process control software nor the laboratory software that keeps track of quality.

B.2. Production

The production department is the user of the schedule. Since the production operators are the ones that need to execute the schedule it is very important that their requests and demands are heard. When materials used for production are stored in more than one storage tank, the production department has to make the decision which tank is used. Since it is easier to produce within specifications by using a higher grade material, this often results in that the highest quality material is used, leaving the lower grade material in the storage. Since materials degrade over time this can result in a negative feedback loop, after which material can become unusable and has to be re-refined. This can be prevented by applying the First In First Out (FIFO) principle. In order to be able to do this the operators need to have knowledge over how long each batch is in storage, information they do not currently posses.There is a lot of variation within process times, this can for example be caused by catalyst being reused, or filters being full. The variation in process times makes it hard to plan for pipelines, since it can not be determined when exactly processes will be finished. A request from production is therefore to have more dedicated pipelines. This should prevent delays. Another problem encountered by the production is that the amount of resources are limited. For example: in the harding plant there are six kettles but only enough resources for heating one kettle and cooling one kettle at a time.

B.2.1. Requests and Demands

The foremost demand of the production department is to have feasible schedules. While this might seem trivial this is not always the case, since currently the planners do not account for all constraints. As long as no new dedicated pipelines are available the production department requests that the scheduling of the pipelines is taken into account as good as possible. There is also a request for the schedule to include the limited resources, such as heating and cooling. The production department requests more information about the amount of material stored. This information can then be used to determine from or to which tank material is transferred. Making these decisions in the scheduling phase could also solve this problem.

B.2.2. Bottlenecks

The production department sees a bottleneck in the pipelines. According to the production department this cannot be solved solely by planning, but extra dedicated pipelines should be installed to improve production. The second bottleneck seen by the production department is the ROS, if not enough output storage is available production could be halted. Another bottleneck is found in the production resources, such as heating, since not enough resources are available some processes can not be run simultaneously, reducing production capability.

B.3. Quality assurance

The Quality assurance department, also called the QA department for short, is there to check if all produced products are not only within government norms but also in the right specifications that the customers. In order to do this not only the end products are checked but also the intermediates and the raw materials. since these checks cost time, they should be accounted for in the schedule. The quality itself might also be dependent on the production schedule, more specifically on the amount of time product is stored. These two fact make the Quality Assurance department a stakeholder in the production scheduling.

B.3.1. General information

Raw material shipments are kept separate from each other in storage in Rotterdam. In the Raw material storage in Zwijndrecht it can occur that multiple shipments are combined. Every five days the raw storage is checked for quality. The catalyst used in the harding is reused but degrades in quality. This is counteracted by using more catalyst, but may increase production times. It often occurs that products are not made to specification and have to be reprocessed. This causes a lot of delays. In intermediate storage and ROS, different batches (of the same product) may be combined in the same tank, but will still be labeled as individual batches. When this happens the tank is "locked" until quality checks are performed. The quality of the end-products in the ROS is checked every three days. Not all products go to the ROS, some products are loaded directly from the deodorizers.

B.3.2. Requests and demands

In order to ensure product quality the finished product may not "overstay" in the deodorizers but has to be transferred immediately after processing. This is a firm demand. A request is that the Bill Of Material (BOM), or the "recipe" for each product is decided by the planning. This will mean that if product A can be made out of either intermediate B or C, the schedule will indicate which intermediate has to be used. This ensures levels of B and C can be monitored. The second request is to keep batches as separated as possible. This prevents the need rechecking already checked batches. The third request is to minimize storage times for all intermediates. While most intermediates can be stored for some time, the quality degrades over time, which sometimes creates the need for extra production steps. The fourth request follows on the second and the third, by having a preference for the FIFO (First In, First Out) principle of handling batches. This reduces storage time for batches.

B.3.3. Bottlenecks

In the eyes of the Quality assurance department the following bottlenecks are found: The pipelines between the raw storage and refineries are the main bottleneck. These are "shared" pipelines which have to be planned. The distribution of pickup times (transport to customers) is heavily concentrated in the beginning of the week, an more equal distribution should increase performance.

Contrary to what other departments reported, the ROS is not seen as a bottleneck by the Quality assurance department. Research done by the Quality assurance department shows that in an average production week only 50% of the ROS is used. Research also showed that when the ROS was planned manually, accounting for all restrictions, an usage of 93% could be reached. While this will probably not always be possible, room for improvements is proved. The key in improving the ROS performance is determined to be in the loading hall and its operators, by planning the transport trucks correctly, and planning which tank is used for loading which product.

B.4. Sales

Depending on how efficient the schedules are the Sales department can sell more or less products to customers. An important part in the sales process is the reliability and flexibility to the customer. This is influenced by the scheduling as better schedules result in more on time planning which means more reliable delivery times, and the possibility for rescheduling creates flexibility to change customers orders when needed.

B.4.1. General information

As far as the sales department is concerned, the reason for the company to exist is the flexibility to the customers. The company is to small to contend with the large bulk oil companies, but is also not in the market for the exact precision materials. The clients of the company want relatively small batches, and most importantly the option to make changes during the week. the sales department fear that if the company will have strict limits on on amount of changes or even charge on changes, customers will leave. This is partly since the company itself often experiences delays (i.e. due to breakdowns) and had to communicate changes to the customer. A wish from the sales department is that orders can be confirmed to customers earlier. The current situation is that all customers will get an order confirmation on Fridays, after the planning is made, even if the customer in question has placed its order on Monday.

B.4.2. Requests and demands

Agreeing with the Quality assurance department, the sales department thinks that the selection of which tank of materials is used for production, should not be done by the production department, but by the planning department. Another point is the blending of mixed oils. In order to create an oil blend within the desired specifications from a customer, a range of blending ratio can be used. Currently the ratio is fixed, however there is an initiative the make the ratio variable. Depending on how often this ratio is changed this may pose a problem for scheduling, since more/less materials are used than previously thought, resulting in shortages in either materials or storage. This can be solved by either fixing the ratios for the duration of a planning or working with buffer values. The latter will depend on how much storage is available and how much the ratio is expected to change.

B.4.3. Bottlenecks

The Sales department sees a lot of room of improvement in the customer service department. If the customers communicate their orders earlier in the week, the first iteration of the planning can be made faster. the top twenty of customers is good for 80% of all orders. Another improvement is how the orders are communicated from the customers to the planners. The current situation is that the information is exported from the customers SAP, and faxed to the company. here a customer service employee will manually type this information into the SAP of the company. Sometime the information is emailed, but not in a way that can be directly imported into SAP (yet). automating this information transfer would save a significant amount of time. The sales department sees the availability of tanks as the greatest bottleneck within the refinery. Both the ROS and the intermediate storage lacks sufficient capacity storage for efficient scheduling. The sales department thinks that most initiatives in the organization, such as closing down in weekends, are not applicable until the performance and the availability of tanks is on par.

B.5. Customer service

The customer service is the link between the customers and the planners. changes in the schedule, whether caused by the production or by the customers, are communicated through the customer service.

B.5.1. General information

After a sales-contract is signed customers are able to order an amount of materials each week, with a set maximum per month. When an order is placed a end time is set. In the case that the customer uses their own transport (FCA) this end time will be the time the truck is loaded. When the transport is done by (or better: in the name of) SDO Zwijndrecht (CIP) the agreed end time will be the time of delivery. Orders are placed in SAP, when placing an order the customer service cannot see if the agreed end time is possible, or that the maximum loading capacity of the loading hall is reached for that time. The confirmation of the order, including information about the planned end time, is available after the production schedule is created. Since the production schedule is made every Thursday for the upcoming Sunday till Saturday, orders placed on a Friday will not be confirmed until the end of the next week. Some customers have complained that the order confirmation takes to long. The deadline to place orders is Thursday mornings on 10 am, after which the SAP order list is loaded to Fygir, the scheduling program. When this is done an order check is performed to make sure the transfer was complete and successful. When the planners have created a schedule this is then communicated to the customer service department. This is often around 3 p.m.. The customer service department then contacts customers whose orders could not met schedule at the agreed end time, and suggest end time as presented in the schedule. Most of the times customers accept these changes, however some customers cancel their order completely or ask for an other end time. In this last case the information is send back to the planners and an attempt is made to make a new schedule which satisfies the customer.

B.5.2. Requests and demands

The request of the Customer service department is to improve their selected KPI's. The KPI's the customer service look to improve are called: on time, in full and in plan. These are defined as follows. On time means that the product is either filled or delivered, depending on the type of transportation contract, as indicated by the schedule. In full is defined as delivering the complete amount of ordered product. This indicator is made because not always all orders can be filled completely. This can be because production errors but also in the case of CIP, that there was not enough capacity in the transport truck. In plan looks at the original agreed end time, compared to the scheduled end time. This means that is a schedule is update quickly enough all "on time"-errors will be transformed into "on plan"-errors. Another request is to shorten the wait for order confirmation to the customers, to improve customer satisfaction. Customer service also notes that the current production is done "to order"but if more storage(tanks) were available the refinery could switch to production "to-stock". If this change is made there would be a better buffer to cope with productions stops, resulting in a better customer experience.

B.5.3. Bottlenecks

The largest hurdle to overcome by the customer service department is the way orders are communicated from the customers to SDO Zwijndrecht. Since there is no template customers send their information in a variety of different ways, ranging from emails containing full information, to emails with only a contract number, to exports from the customers order software, to even faxes, which have to be entered manually. standardizing the ordering process will streamline the customer service significantly.

B.6. Maintenance department

In order to keep the refinery running, preventive maintenance has to be executed on (almost) all production units. While this maintenance is being executed no other processes can be run on the equipment. Since this has to be accounted for in the production schedule, the maintenance department is a stakeholder.

B.6.1. General information

The maintenance department needs planned downtime on production lines in order to do preventive maintenance. The main problem in this is the variable end times of processes, and the frequent rescheduling of the production schedule. The variable end times of the refinery production are mainly caused by the quality of the intermediate product used. When an intermediate of a lower quality is used more so called Fuller's earth is needed. Since this Fuller's earth has to be filtered out again, and filters only have a set capacity, more filtration steps are needed causing a discrete increase in production times. This makes it very hard to make a reliable maintenance schedule. In the future the maintenance planning can possibly be linked to the production schedule and even be created simultaneously, but this is a long way of. For now the maintenance planning needs either fixed maintenance slots or a longer frozen horizon in the production planning.

B.6.2. Requests and demands

A request from the maintenance department is to have as few reschedules as possible, allowing for an effective maintenance planning. A second request is that there is made a better estimation of the production end-rimes. Another request is to integrate the maintenance planning in the production planing.

B.6.3. Bottlenecks

According to maintenance the bottleneck in the harding which is caused by limited heat resources will soon be resolved. this is going to be done by using heat from other process steps to heat water which in turn will heat the kettles, instead of the steam that is now used. A bottleneck as seen by the maintenance department is the planning of the transportation to the customers, including pre-loading, and transport arranged by the customers themselves.

C

TankAnalysis.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% TankAnalysis.m                          %%
%%                                         %%
%% Estimates how many tanks could be freed, %%
%% by combining volumes                    %%
%%                      1-6-2019 P.D.Baart %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all
close all
tic
f=waitbar(0,'Starting');
Lijst=[22500;22500;22500;22500;22500;22500;22500;22500;22500;22500;...
    22500;22500;0;22500;21000;22500;22500;22500;22500;22500;10000;...
    10000;10000;10000;8400;8400;8400;8400;24000;24000;24000;20000;...
    0;0;0;0;0;0;0;0;30500;30500;30500;30500;30500;30500;24000;24000;...
    24000;24000;20000;20000;20000;90000;90000;154000;43000;43000;0;...
    0;20000;29000;42000;42000;42000;100000;100000;100000;0;54000;...
    47000;150000;0;150000;0;150000;0;100000;50000;150000;150000;...
    150000;150000;150000;150000;150000;150000;0;0;0;44000;44000;...
    44000;44000;0;161000;43000;43000;0;0;0;0;0;0;0;0;0;0;0;0;...
    60000;60000;60000;60000;60000;60000];
%Bron=readtable("RawData2.csv");
load('bron.mat')
%Bron.Time_Stamp=datetime(Bron.Time_Stamp,"InputFormat",'dd-MM-yyy HH:mm:
    ss');
%w=datetime(w,"InputFormat",'dd-MM-yyy HH:mm:ss');
[q,w]=findgroups(Bron.Time_Stamp);
VrijTePlannenVolume=zeros(1,length(w));
VrijTePlannenTanks=zeros(1,length(w));
Ruw=cell(1,length(w));
Verwerkt=cell(1,length(w));
disp("start")
toc

for i=1:length(w)
waitbar(i/length(w),f,'Calculating')
A=Bron(ismember(Bron.Time_Stamp,w(i)),:);
A.Model_Name=erase(string(A.Resource),"ROS");
```

```
A.PCS_Quantity=str2double(string(A.PCS_Quantity));
A=A(ismember(A.Status,'Available'),:);
% A(ismember(A.Status,'Loading'),:)=[];
% A(ismember(A.Status,'Unloading'),:)=[];
A(ismember(A.PCS_Product_Code,""),:)=[];
 A(ismember(A.PCS_Product_Code,'0'),:)=[];
for j=1:length(A.Model_Name)
    A.Max_Quantity(j)=Lijst(str2double(A.Model_Name(j)));
end

 if numel(A)<24
      continue
 end
B=varfun(@sum,A,"inputvariables",["PCS_Quantity","Max_Quantity"],'
    GroupingVariables','PCS_Product_Code');
if isempty(B)
      continue
 end

 B.Properties.RowNames=B.PCS_Product_Code;
 B.PCS_Product_Code=[];
C=varfun(@min,A,"inputvariables","Max_Quantity",'GroupingVariables','
    PCS_Product_Code');
D=varfun(@mini,A,"inputvariables","Max_Quantity",'GroupingVariables','
    PCS_Product_Code');
B.kleinstetank=C{:,3};
B.eennakleinstetank=D{:,3};
[leegbaar,extraruimte]=ruimte(B.sum_Max_Quantity,B.kleinstetank,B.
    eennakleinstetank,B.sum_PCS_Quantity);
B.leegbaar=leegbaar';
B.extraruimte=extraruimte';
VrijTePlannenVolume(i)=sum(B.extraruimte);
VrijTePlannenTanks(i)=sum(B.leegbaar);
B{'Total',:}=sum(B{:,:},'omitnan');
if max(B.leegbaar(1:end-1)) < 2
     B=removevars(B,"eennakleinstetank");
end
Ruw{i}=sortrows(A,'PCS_Product_Code');
Verwerkt{i}=B;
end

figure(1)
subplot(2,1,1)
area(w(1:length(VrijTePlannenVolume)),VrijTePlannenVolume)
xlabel ("Time")
ylabel("Free space in kg")
subplot(2,1,2)
area(w(1:length(VrijTePlannenVolume)),VrijTePlannenTanks)
xlabel( "Time")
ylabel( "# Free tanks")
toc
```

D

Defined constraints in the method

Nomenclature

Sets/Indices

$\mathbf{N}/n, n'$	global time points/intervals
$\mathbf{I}/i, i'$	tasks
$\mathbf{I}^P/\mathbf{I}^C/\mathbf{I}^M/\mathbf{I}^T/\mathbf{I}^{PH}$	processing/cleaning/maintenance/transfer/ previous horizon tasks
$\mathbf{I}^{CZW}/\mathbf{I}^{PZW}$	processing tasks $i \in \mathbf{I}^P$ that consume/produce an unstable material
$\mathbf{I}^{TZW}$	transfer tasks $i \in \mathbf{I}^T$ associated with an unstable material
$\mathbf{J}/j, j'$	units
$\mathbf{J}^P/\mathbf{J}^S/\mathbf{J}^C/\mathbf{J}^T$	processing/storage/cleaning/transfer units
$\mathbf{J}^{DS}/\mathbf{J}^{SS}$	dedicated/shared storage units
$\mathbf{M}/m$	materials
$\mathbf{M}^D/\mathbf{M}^P$	sold/purchased materials
$\mathbf{M}^V$	intermediate materials with commercial value
$\mathbf{M}^{ZW}$	unstable materials (they must be handled under a zero-wait policy)
$\mathbf{M}^{NIS}$	materials for which no storage unit is available
$\mathbf{M}^S$	materials that can be stored in storage units
$\mathbf{L}/l$	material shipments
$\mathbf{R}/r$	non-unary resources
$\mathbf{C}/c$	unit modes
$\mathbf{V}/v$	transfer devices
$\mathbf{I}_m^C/\mathbf{I}_m^P$	processing tasks $i \in \mathbf{I}^P$ that consume/produce material m
$\mathbf{I}_j$	tasks associated with unit j
$\mathbf{I}_r$	tasks that require non-unary resource r
$\mathbf{I}_c^P$	tasks that produce mode c when end
$\mathbf{J}_j$	units connected to unit j
$\mathbf{J}_i$	units associated with execution of task i
$\mathbf{J}_m^S$	storage units that can store material m
$\mathbf{J}_m^I/\mathbf{J}_m^O$	processing units $j \in \mathbf{J}^P$ that can store material m as an input/output material
$\mathbf{J}_v^T$	transfer units $j \in \mathbf{J}^T$ in which transfer device v is a component
$i(m, j)$	transfer task $i \in \mathbf{I}^T$ associated with the transfer of material m through transfer unit $j \in \mathbf{J}^T$
$j(i)$	unit j on which task i is scheduled to take place
$j(j', j'')$	transfer unit $j \in \mathbf{J}^T$ that connects the outlet of unit $j' \in (\mathbf{J}^P \cup \mathbf{J}^S)$ with the inlet of unit $j'' \in (\mathbf{J}^P \cup \mathbf{J}^S)$
$\mathbf{M}_j^S$	materials that can be stored in unit $j \in \mathbf{J}^{SS}$
$\mathbf{M}_j^I/\mathbf{M}_j^O$	materials that can be stored in processing unit $j \in \mathbf{J}^P$ as input/output materials
$m(l)$	material corresponding to shipment l
$m(c)$	material whose storage during a time interval produces mode c
$\mathbf{C}_j$	modes defined for unit $j \in (\mathbf{J}^P \cup \mathbf{J}^{SS})$
$\mathbf{C}_i^C/\mathbf{C}_m^C$	modes that can be consumed by the beginning of task i/storage of material m

Parameters

$A_{c,j}^0$	1 if mode $c \in \mathbf{C}_j$ is initially available in unit $j' \in (\mathbf{J}^P \cup \mathbf{J}^{SS})$ at the beginning of the horizon; otherwise, it must be equal to zero.
H	time horizon
$a_{i,j}/b_{i,j}$	fixed/variable duration of task i in unit j
$B_{i,j}^0$	batch size of previous horizon processing task $i \in (\mathbf{I}^P \cap \mathbf{I}^{PH})$ executed in unit $j \in \mathbf{J}^P$
$f_{i,j,r}/g_{i,j,r}$	fixed/variable amount of resource r required by task i in unit j
$I_{m,j}^0$	initial amount of material m in storage unit $j \in \mathbf{J}^S$
q_l	amount associated with shipment l
$\beta_j^{MAX}/\beta_J^{MIN}$	maximum/minimum capacity for unit j
$\gamma_{i,m}$	mass fraction for the consumption (−)/production (+) of material m by processing task $i \in \mathbf{I}^P$
π_m	price (value) of material m
ρ_r^{MAX}	maximum availability of resource r
ς_j^{MAX}	maximum storage capacity for storage unit $j \in \mathbf{J}^S$
σ_i	cleaning cost associated with cleaning task $i \in \mathbf{I}^C$
τ_l	time corresponding to shipment l
τ_i^{MB}/τ_i^{ME}	fixed start/end time of maintenance task $i \in \mathbf{I}^M$
$\tau_i^{MEB}/\tau_i^{MLE}$	earliest start time/latest end time for maintenance task $i \in \mathbf{I}^M$

Binary variables

$O_{l,j,n}$	1 if shipment l occurs to/from storage unit $j \in \mathbf{J}^S$ at time point n
$S_{j,n}^I/S_{j,n}^O$	1 if in processing unit $j \in \mathbf{J}^P$ input/output materials are stored during time interval n
$S_{m,j,n}^S$	1 if material $m \in \mathbf{M}_j^S$ is stored in shared storage unit $j \in \mathbf{J}^{SS}$ during time interval n
$X_{i,j,n}/Y_{i,j,n}$	1 if task $i \in \mathbf{I}_j$ formally starts/ends in unit j at T_n

Continuous (non-negative) variables

$A_{c,j,n}$	1 if mode $c \in \mathbf{C}_j$ is available in unit $j \in (\mathbf{J}^P \cup \mathbf{J}^{SS})$ at time point n
$E_{j,n}$	1 if unit j is formally performing/receiving a task during time interval n
$K_{c,j,n}^C/K_{c,j,n}^P$	1 if mode $c \in \mathbf{C}_j$ is consumed/produced in unit $j \in (\mathbf{J}^P \cup \mathbf{J}^{SS})$ at time point n
$S_{j,n}$	1 if unit $j \in (\mathbf{J}^P \cup \mathbf{J}^S)$ is storing materials during time interval n
$W_{j,n}$	1 if unit j is available during time interval n
$Z_{j,n}$	1 if unit j is performing/receiving at time point n a task started in a previous time point
$B_{i,j,n}^S/B_{i,j,n}^E$	batch size of task i which formally starts/ends at T_n in unit j
$B_{i,j,n}^P$	batch size of task i being executed at T_n in/by unit j
$F_{m,j',j,n}$	instantaneous transfer of material m from unit $j' \in (\mathbf{J}^P \cup \mathbf{J}^S)$ to $j \in ((\mathbf{J}^P \cup \mathbf{J}^S) \cap \mathbf{J}_{j'})$ at time point n
$\mathrm{I}_{m,j,n}^I/\mathrm{I}_{m,j,n}^O$	amount of input/output material m stored in processing unit $j \in \mathbf{J}^P$ during time interval n
$I_{m,j,n}^S$	amount of material m stored in storage unit $j \in \mathbf{J}^S$ during time interval n
$P_{m,j,n}$	amount of output material m in unit $j \in \mathbf{J}^P$ that becomes an input material for the same unit at T_n
$Q_{r,n}$	total amount of non-unary resource r required during time interval n
T_n	time corresponding to global point n
$\bar{T}_l^E/\bar{T}_l^T$	earliness/tardiness for shipment l
$\bar{T}_{j,n}^{LB}/\bar{T}_{j,n}^{EE}$	number of time units the actual beginning/end of a task is delayed/anticipated in unit j with respect to its formal beginning/end at T_n
$\bar{T}_{j,n}^S/\bar{T}_{j,n}^W$	length of the storage/idle interval n taking place in unit j from T_n to T_{n+1}

1.
$$E_{j,n} + S_{j,n} + W_{j,n} = 1 \qquad \forall j \in J^P, n < N$$

2.
$$E_{j,n} = Z_{j,n} + \sum_{i \in I_j} X_{i,j,n} \qquad \forall j \in J^P, n < N$$

3.
$$Z_{j,n} = Z_{j,n-1} + \sum_{i \in I_j} X_{i,j,n-1} - \sum_{i \in I_j} Y_{i,j,n} \qquad \forall j \in J^P, n > 1$$

4.
$$S_{j,n} = S^I_{j,n} + S^O_{j,n} \qquad \forall j \in J^P, n < N$$

5.
$$\bar{T}^{LB}_{j,n} \leq H \sum_{i \in I_j \setminus I^{CZW}} X_{i,j,n} \qquad \forall j \in J^P, n < N$$

6.
$$\bar{T}^{EE}_{j,n} \leq H \sum_{i \in I_j \setminus I^{PZW}} Y_{i,j,n} \qquad \forall j \in J^P, n > 1$$

7.
$$\bar{T}^{S}_{j,n} \leq H(S_{j,n}) \qquad \forall j \in J^P, n < N$$

8.
$$\bar{T}^{W}_{j,n} \leq H(W_{j,n}) \qquad \forall j \in J^P, n < N$$

9.
$$T_{n+1} = T_n - H(1 - S_{j,n} - W_{j,n}) \leq \bar{T}^S_{j,n} + \bar{T}^W_{j,n} \leq T_{n+1} = T_n \qquad \forall j \in J^P, n < N$$

10.
$$T_n \geq \sum_{1<n' \leq n} \sum_{i \in I_j} a_{i,j} Y_{i,j,n'} + b_{i,j} B^E_{i,j,n'} + \sum_{1<n' \leq n} \bar{T}^{EE}_{j,n'} + \sum_{n'<n} \bar{T}^{LB}_{j,n'} + \bar{T}^S_{j,n'} + \bar{T}^W_{j,n'} \qquad \forall j \in J^P, n > 1$$

11.
$$H - T_n \geq \sum_{n \leq n' < N} \sum_{i \in I_j} a_{i,j} X_{i,j,n'} + b_{i,j} B^S_{i,j,n'} + \sum_{n' > n} \bar{T}^{EE}_{j,n'} + \sum_{n \leq n' < N} \bar{T}^{LB}_{j,n'} + \bar{T}^S_{j,n'} + \bar{T}^W_{j,n'} \qquad \forall j \in J^P, n < N$$

12.
$$\sum_{n>1} \bar{T}^{EE}_{j,n'} + \sum_{n<N} \bar{T}^{LB}_{j,n'} + \bar{T}^S_{j,n'} + \bar{T}^W_{j,n'} + \sum_{n>1} \sum_{i \in I_j} a_{i,j} X_{i,j,n'} + b_{i,j} B^S_{i,j,n'} = H \qquad \forall j \in J^P$$

13.
$$\beta^{MIN}_j Y_{i,j,n} \leq B^E_{i,j,n} \leq \beta^{MAX}_j Y_{i,j,n} \qquad \forall i \in I^P, j \in J_i, n > 1$$

14.
$$B^S_{i,j,n} \leq \beta^{MAX}_j X_{i,j,n} \qquad \forall i \in I^P, j \in J_i, n < N$$

15.
$$\sum_{i \in I_J} B^P_{i,j,n} \leq \beta^{MAX}_j Z_{j,n} \qquad \forall j \in J^P, n$$

16.
$$B^S_{i,j,n} + B^P_{i,j,n} = B^P_{i,j,n+1} + B^E_{i,j,n+1} \qquad \forall i \in I^P, j \in J_i, n < N$$

17.

$$I^S_{m,j,n} = I^S_{m,j,n-1} - \sum_{j' \in (J_j \cap (J^S_m \cup J^I_m))} F_{m,j,j',n} + \sum_{j' \in (J_j \cap (J^S_m \cup J^O_m))} F_{m,j',j,n} \leq \varsigma^{MAX}_{m,j} \qquad \forall m \in M^s, j \in J^S_m, n$$

18.

$$\sum_{m \in M^S_j} S^S_{m,j,n} \leq 1 \qquad \forall j \in J^{SS}, n > 1$$

19.

$$I^S_{m,j,n} \leq \varsigma^{MAX}_{m,j} S^S_{m,j,n} \qquad \forall m \in M^S_j, j \in J^{SS}, n > 1$$

20.

$$I^I_{m,j,n} = I^I_{m,j,n-1} + \sum_{j' \in (J_j \cap (J^S_m \cup J^O_m))} F_{m,j',j,n} + \sum_{i \in (I_j \cap I^C_m)} \gamma_{i,m} B^S_{i,j,n} \qquad \forall m \in M, j \in J^I_m, n$$

21.

$$I^O_{m,j,n} = I^O_{m,j,n-1} + \sum_{i \in (I_j \cap I^P_m)} \gamma_{i,m} B^E_{i,j,n} - \sum_{j' \in (J_j \cap (J^S_m \cup J^I_m))} F_{m,j,j',n} \qquad \forall m \in M, j \in J^O_m, n > 1$$

22.

$$\sum_{m \in M^I_j} I^I_{m,j,n} \leq \beta^{MAX}_j S^I_{j,n} \qquad j \in J^P, n < N$$

23.

$$\sum_{m \in M^O_j} I^O_{m,j,n} \leq \beta^{MAX}_j S^O_{j,n} \qquad j \in J^P, n < N$$

24.

$$Q_{r,n} = Q_{r,n-1} + \sum_{i \in I_r} \sum_{j \in J_i} f_{i,j,r}(X_{i,j,n} - Y_{i,j,n}) + g_{i,j,r}(B^S_{i,j,n} - B^E_{i,j,n}) \leq \rho^{MAX}_r \qquad \forall r \in R, n$$

25.

$$\sum_{n<N} X_{i,j)i),n} = 1 \qquad i \in I^{MT}$$

26.

$$\tau^{MEB}_i \leq T_n + \bar{T}^{LB}_{j(i),n} + H(1 - X_{i,j(i),n}) \qquad i \in I^{MT}, n < N$$

27.

$$\tau^{MLE}_i \leq T_n - \bar{T}^{EE}_{j(i),n} - H(1 - X_{i,j(i),n}) \qquad i \in I^{MT}, n > 1$$

28.

$$\sum_{j \in ((J^P \cup J^{SS}) \cap J_i)} X_{i,j,n} = \sum_{j \in (J^C \cap J_i} X_{i,j,n} \qquad \forall i \in I^C, n < N$$

29.

$$\sum_{j \in ((J^P \cup J^{SS}) \cap J_i)} Y_{i,j,n} = \sum_{j \in (J^C \cap J_i} Y_{i,j,n} \qquad \forall i \in I^C, n > 1$$

30.

$$\sum_{j \in J^T_v} E_{j,n} \leq 1 \qquad v \in V, n < N$$

31.

$$\sum_{j \in J_j} E_{j''(j',j),n} \leq S^I_{j,n} \qquad \forall j \in J^P, n < N$$

32.
$$\sum_{j \in J_j} E_{j''(j,j'),n} \leq S^{O}_{j,n} \qquad \forall j \in J^P, n < N$$

33.
$$\sum_{j \in J_j} E_{j''(j',j),n} + \sum_{j \in J_j} E_{j''(j,j'),n} \leq S^{j,n} \qquad \forall j \in J^S, n < N$$

34.
$$X_{i,j(i),1} = 1, X_{i,j(i),1<n<N} = 0 \qquad i \in I^{PH}$$

35.
$$A_{c,j,n} = A_{c,j,n-1} - K^{C}_{c,j,n} + K^{P}_{c,j,n} \qquad \forall j \in (J^P \cup J^{SS}), c \in C_j, n$$

36.
$$\sum_{c \in (C_j \cap C^C_i)} K^{C}_{c,j,n} \geq X_{i,j,n} \qquad \forall i \in I_j, j \in J^P, n < N$$

37.
$$\sum_{c \in C_j} K^{C}_{c,j,n} \leq 1 \qquad \forall j \in J^P, n$$

38.
$$K^{P}_{c,j,n} = \sum_{i \in (I_j \cap I^P_c)} Y_{i,j,n} \qquad \forall j \in J^P, n$$

39.
$$\sum_{c \in C^C_m} K^{C}_{c,j,n} \geq S^{S}_{m,j,n} \qquad \forall j \in J^{SS}, m \in M^S_j, n < N$$

40.
$$K^{P}_{c,j,n} = S^{S}_{m(c),j,n-1} + \sum_{i \in (I_j \cap I^P_c)} Y_{i,j,n} \qquad \forall j \in J^{SS}, n$$

41.
$$\sum_{j \in J^S_{m(l)}} \sum_{n \in N} O_{l,j,n} = 1 \qquad l \in L$$

42.
$$T_n - H(1 - \sum_{j \in J^S_{m(l)}} O_{l,j,n}) \leq \tau_l + \bar{T}^T_l - \bar{T}^E_l \leq T_n - H(1 - \sum_{j \in J^S_{m(l)}} O_{l,j,n}) \qquad l \in L, n \in N$$

43.
$$S_{j,n} = \sum_{m \in M^S_j} S^{S}_{m,j,n} \qquad \forall j \in J^{SS}, n < N$$

44.
$$\bar{T}^{LB}_{j,n} \leq H \sum_{i \in I_j, i \notin (I^{CZW} \cup I^{TZW} \cup I^{PH})} X_{i,j,n} \qquad \forall j, n < N$$

45.
$$\bar{T}^{EE}_{j,n} \leq H \sum_{i \in I_j, i \notin (I^{PZW} \cup I^{TZW}} Y_{i,j,n} \qquad \forall j, n > 1$$

46.

$$I^S_{m,j,n} = I^S_{m,j,n-1} - \sum_{j' \in (J^S_m \cup J^I_m)} B^S_{i(m,j''),j''(j,j'),n} + \sum_{j' \in (J^S_m \cup J^O_m)} B^E_{i(m,j''),j''(j',j),n} + \sum_{l \in L_m} q_l O_{l,j,n} \leq \varsigma^{MAX}_{m,j} \quad \forall M^S, j \in J^S_m, n$$

47.

$$I^S_{m,j,n} \leq \varsigma^{MAX}_{m,j} S_{j,n} \qquad \forall m \in M^S, j \in J^S_m, n < N$$

48.

$$I^I_{m,j,n} = I^I_{m,j,n-1} - \sum_{j' \in (J^S_m \cup J^O_m)} B^S_{i(m,j''),j''(j',j),n} + P_{m,j,n} + \sum_{i \in (I_j \cap I^C_m)} \gamma_{i,m} B^S_{i,j,n} \qquad \forall m, j \in J^I_m, n$$

49.

$$I^O_{m,j,n} = I^O_{m,j,n-1} + \sum_{i \in (I_j \cap I^P_m)} \gamma_{i,m} B^E_{i,j,n} - \sum_{j' \in (J^S_m \cup J^I_m)} B^E_{i(m,j''),j''(j,j'),n} - P_{m,j,n} \qquad \forall m, j \in J^O_m, n$$

E Reproduction Results

Timesteps	Answer literature	CPU time literature (s)	Answer reproduction	CPU time reproduction (s)
Example 1				
3	No Solution		No solution	
4	1420.7	0.05	No solution	
5	2730.8	0.38	1420.7	0.219
6	3592.2	0.87	2730.8	0.25
7	3592.2	9.86	3592.2	0.265
Example 2a				
5	No Solution		No solution	
6	7.800	1.06	No solution	
7	7.781	8.86	7.800	0.38
8	7.781	84.5	7.781	1.17
Example 2b				
6	No solution		No solution	
7	11.488	3.08	No solution	
8	11.417	67.6	11.488	1.19
9	11.321	1238	11.329	6.17
10	11.321	>10,000[1]	11.321	32.28
Example 2c				
8	No solution		No solution	
9	14.723	58.2	16.154	3.30
10	13.902	1493	13.939	36.86
11	13.902	>10,000[1]	13.860	366
12			13.856	2733
13			13.856	18,510
Example 3				
3	No solution		No solution	
4	2042.7	0.07	No solution	
5	3168.8	0.28	2042.7	
6	3273.1	2.53	3168.8	
7	3273.1	27.1	3273.1	

Table E.1: Comparison in answers an CPU-times per timesteps used

[1] cut-off

Timesteps	Answer literature	CPU time literature (s)	Answer reproduction	CPU time reproduction (s)
Example 4				
5	No Solution		No solution	
6	906.67	1.08	906.67	0.58
7	906.67	11.9	1780.8	2.72
8	3253.3	24.2	3253.3	8.84
9	3253.3	240	3253.3	23.64
10	3253.3	1969	3253.3	60.98
Example 5				
3	No Solution		No solution	
4	1420.7	0.13	1762.9	0.156
5	2730.8	0.30	2730.8	0.281
6	2730.8	1.55	2879.0	0.859
7	2730.8	16.4	3045.9	3.359
8	3093.0	47.0	3093.0	8.625
9	3093.0	189	3093.0	41.125
10	3093.0	1089	3093.0	126.157
Example 6				
9	No solution		No solution	
10	1225.7	7.59	1095.7	
11	1225.7	16.9	1095.7	
12	2258.3	31.4	2258.3	
13	2258.3	110	2258.3	
14	2258.3	416	2258.3	
15	2258.3	1238	2258.3	

Table E.2: Comparison in answers an CPU-times per timesteps used

F

The complete ROS in simplified layout