MAURITS BRANT

July 2021

Driving Eurol's production scheduling forward with scheduling algorithms: A case study

University Supervisors
Dr. Marco Schutten, Dr. Eduardo Lalla-Ruiz

Company Supervisor
Ir. Aart van Harten

UNIVERSITY
OF TWENTE.

Management summary

Eurol is a producer of lubricants and technical fluids (brake fluid, antifreeze coolant, etc.). At the Eurol production plant, raw materials are stored and used in mixing units to produce blends. After mixing, the blends are stored in storage units (IBCs or tanks) that are emptied by filling lines, which fill a wide variety of containers. Part of the production plant is referred to as the mixing plant, which is responsible for mixing and subsequent storage. Eurol believes they lack insight in the mixing plant's performance but think that resources can be used more efficiently by improving their scheduling strategy. This leads to the main research question:

How can Eurol improve their scheduling strategy of the mixing plant so that the resources of the mixing plant are used more efficiently?

We have identified 2 objectives that can be influenced with considerable effect by scheduling, leading to more efficient use of the mixing plant's resources, namely minimising changeovers and minimise IBC usage by making more efficient use of tanks. Changeovers occur when rinsing is necessary because a mixing unit/tank containing a blend changes product group, e.g., hydraulic oil to gear oil. By means of interviews and data analysis, insight is gained into the process and the performance of the mixing plant. The results show that it may be possible to use fewer IBCs by producing jobs less early. In addition, the analysis shows that the mixing plant is a very variable environment, meaning that the schedules must be revised within the scheduling horizon. Therefore, the scheduling goal is not to find an optimal solution but instead find a good solution quickly. Finally, dedicated tanks may not be as preferable as they seem.

Based on a literature review, we conclude the problem in hand is a single-stage scheduling problem with parallel machines and storage, which is NP-hard. This means that for realistic problem sizes, no optimal solution can be found within a reasonable time. Therefore, we looked at heuristic approaches and scheduling strategies. We found a similar, but not equal, problem with solution approach. The solution approach divides the problem into subproblems as is often done in heuristic approaches. Furthermore, from the solution approach found in literature we derived scheduling rules and strategies.

The problem in hand is divided into 3 subproblems: assigning jobs to mixing units, scheduling of mixing units, assigning jobs to storage units. The subproblems are solved sequentially, we only move to the next subproblem when the previous one is completely solved. The assignment of jobs to mixing units is solved using rules from literature, extended to also take into account rinsing. In short, these rules give preference to the smallest unit that can produce the job. We optimise the scheduling of mixing units applying Just In Time (JIT) and Group Scheduling (GS) strategies. JIT schedules jobs as close to their due date as possible to reduce tank occupancy time. GS groups jobs with the same rinsing group to reduce rinsing. Finally, we assign jobs to storage units following the same principles as the rules for assigning jobs to mixing units. The assignment of jobs to storage units is optimised with a Greedy Randomised Adaptive Search Procedure (GRASP). This procedure selects the next job to assign with a certain probability over multiple iterations generating multiple solutions.

We experiment with 2 of the most impactful subproblems, namely the scheduling of mixing units (subproblem 2) and the assignment of jobs to storage units (subproblem 3). The experiments use data from practice. Based on the results, we recommended parameter settings to be used in case of implementing the algorithm. The solution approach can solve subproblem 2 in less than 1 second. Subproblem 3 is solved in approximately 80 seconds with the recommended settings. The objectives in scheduling mixing units are to minimise changeovers and minimise the early production of a job. Table 1 shows the results of the experiment for scheduling of mixing units.

Table 1 Summary of results: scheduling mixing units

	Earliness improvement	Changeover improvement
Minimum earliness	45.92%	0%
Minimum changeovers	33.79%	17.82%

The objectives in assigning jobs to storage units are to minimise changeovers and the use of IBCs. The results show that the use of IBCs can be reduced by 5.15%, but with an increase in changeovers of 1.14%. However, the reduction in the use of IBCs is more important than the increase in the number of changeovers. Even though cost parameters are not quantified we concluded that reducing IBC use reduces costs more than reducing changeovers. Experiments for both subproblems show positive results. In the scheduling of mixing units, we manage to reduce earliness approximately 40% whilst also reducing the number of changeovers. In the scheduling of tanks, we manage to reduce the number of IBCs used by approximately 5% but increase the number of changeovers 1%. Combining the solutions should therefore also give a positive result, as scheduling closer to the due date of a job reduces overall storage time, allowing tanks to be used for more jobs, leading to less IBC usage. Therefore, we conclude that the scheduling strategy presented in this research can enable the mixing plant to use its resources more efficiently by providing decision support. Furthermore, it can reduce manual scheduling time and can provide insights for plant optimisation.

We recommend Eurol to implement the solution approach to subproblem 1 and 2, initially only for the planner. When all subproblems are implemented for the planner they can also be implemented for the mixing plant to be used during night shifts. Next, Eurol’s data model should be updated, more data is required and should be readily available. Only then should the solution approach to subproblem 3 be implemented. We also recommend Eurol to quantify cost parameters. Finally, we recommend developing KPIs for the mixing plant and automate their calculation. In this way, the efficiency and possible efficiency improvements can be monitored.

The problem described in this research differs in some important aspects from the most similar problem found in literature; that of Kudva, Elkamel, Penky, & Reklaitis (1994). For example, in our objective function we have to take into account the cost of the storage unit (tank or IBC). To the best of our knowledge, this problem is new to literature. The proposed solution approach also differs, but uses some of the same principles as Kudva, Elkamel, Penky, & Reklaitis (1994). For example, we solve each subproblem sequentially, allowing the solution approach to also be implemented sequentially. Also, to the best of our knowledge, we have developed a new heuristic for the mixing unit scheduling problem presented in this research. This problem is also unique because there are only some jobs with release dates. The heuristic has few parameters and is relatively easy to implement, which suits preferences of companies. Furthermore, we have extended the scheduling rules presented by Kudva, Elkamel, Penky, & Reklaitis (1994) for the assignment of jobs to mixing units to the assignment of jobs to tanks.

Preface

To complete my master Industrial Engineering and Management at the University of Twente, I have written this thesis about the research I did at Eurol. I enjoyed writing this thesis, it was interesting and challenging. I would like to take this opportunity to thank everyone who made this possible.

First of all, I would like to thank Eurol for providing me with the opportunity to do my thesis at their company. I would like to thank Ir. Aart van Harten for his guidance during my time at Eurol. Furthermore, I would like to thank all the planners and operators for their help in familiarising me with the processes at Eurol and answering my questions. Last but not least, I would like to thank all colleagues for the great time I had at Eurol.

Secondly, I would like to thank Marco Schutten and Eduardo Lalla-Ruiz. Both of them have shaped me into a better researcher. Their feedback and guidance made me think critically of my own decisions and really improved the quality of this thesis.

Finally, I would like to thank my family and friends. Their support made me strong and a better individual. Without them, I would not have managed to get this far.

I hope you will enjoy reading this thesis and that Eurol and other researchers benefit from it.

Maurits Brant, July 2021

Definitions

BIC	Blend Instruction Card, instruction of how to make a certain blend.
Filling line	Sequential line of machines that fill final packages (i.e., spray cans, jerrycans), screw on lids, adds labels and prepares the final product for transport by for example palletising.
Filling job	A job for a filling line to fill a certain amount of blend in containers.
GS	Group Scheduling is a scheduling strategy, scheduling jobs with equal rinsing groups after one another.
IBC	Intermediate Bulk Container that can contain 1,000 L of a fluid.
Job	A job for a mixing unit to make a certain amount of a blend.
JIT	Just In Time is a scheduling strategy, scheduling a job as close to its due date as possible.
Manifold	A branch where several pipes are reduced to a single pipe.
Mixing unit	Unit in which multiple raw materials are mixed to create a blend.
Rinsing	Cleaning activity applicable to everything that can contain blends. Rinsing is done by pumping raw material through a unit (tank, line, mixing unit) in order not to contaminate the next blend to be contained with remnants of the previous blend.
Rinsing group	Group of blends with similar properties, e.g., hydraulic oil. Units that contained blends of the same rinsing group after one another did not require rinsing.
Scheduling	The process of creating a plan that specifies what is to be produced/stored on/in which unit(s) and when.
Storage unit	A unit (IBC/tank) in which blends can be stored (between mixing and filling).

List of figures

List of tables

Table of contents

1 Introduction

This chapter introduces the company and the research carried out. Section 1.1 introduces Eurol after which Section 1.2 identifies the problem. Thereafter, Section 1.3 explains the research aim and Section 1.4 explains the research design. Lastly, Section 1.5 lists the research deliverables.

1.1 Company introduction

Eurol is a producer of lubricants and technical fluids (brake fluid, antifreeze coolant, etc.) based in Nijverdal, The Netherlands since 1977. Eurol has one mixing and filling plant that allows lubricants and technical fluids to be mixed accurately, reliably, quickly, and flexibly in both small and large volumes (500-50,000 L). Eurol has its own laboratory and R&D centre where testing, continuous development and improvement of their products takes place. The production of lubricants and technical fluids are separate production processes without shared resources. Furthermore, Eurol has 14 buildings for storage. With approximately 250 employees, Eurol is the largest independent producer of lubricants and technical fluids in The Netherlands, serving more than 80 countries. With a full-service approach, Eurol offers a complete range of lubricants and technical fluids. With this approach Eurol serves several markets such as automotive, transport, industrial and agricultural markets. Eurol also proudly supports several teams in the Dakar rally. The quality program ‘Eurol House of Excellence’ contributes to the continuous development of Eurol’s employees and processes. The Eurol promise is central to every employee:

Quality is in our nature

Figure 1.1 gives a very basic overview of Eurol’s lubricant production process and the responsible departments (Mixing, Filling and Planning).

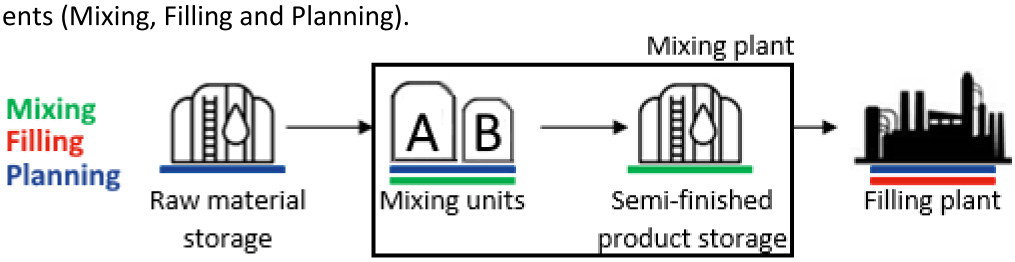

Figure 1.1 Basic description production process

The production starts by pumping raw materials to a mixing unit (mixer) in the mixing plant. A mixer mixes the raw materials into a homogeneous blend. This blend is then pumped into semi-finished product storage units (storage unit from here on always refers to the storage of semi-finished products). Finally, the blend is tapped at a filling line where containers ranging from spray cans and jerrycans to drums containing from 100 ML to 210 L are filled. Finally, labels and lids are applied, and the product is packed ready for dispatch.

We classify the topology of the process as a flow-shop because the sequence of operations is the same for all products (Graham, Lawler, Lenstra, & Kan, 1979). Eurol makes approximately 700 different lubricants packaged in approximately 5,000 different SKUs.

1.2 Problem identification

In recent years Eurol increased their market share and intends to continue to do so aiming for a 10% increase in production output in 2021. Also, Eurol wants to produce larger volumes and reduce sales of small volumes. To be able to do this, the production process must be able to meet this demand. Eurol believes that they lack insight into the current performance of the mixing plant. Even though they feel like they lack insight they think that the mixing plant can be scheduled more efficiently. This is the motivation of this research. Scheduling more efficiently should result in a more efficient use of the mixing plant's resources. This leads to the following core problem:

Mixing plant resources are not used efficiently enough

As a **measure of efficiency**, we propose the minimisation of the number of IBCs to be used in combination with the number of changeovers required (some product sequences do not require changeovers). There are 2 storage media namely, **IBCs** and tanks. Tanks are preferred, we discuss this preference in more detail in Chapter 2. Not all batches can be stored in tanks because of restrictions in the filling plant (e.g., a filling line cannot connect to a tank). All jobs must be scheduled to meet their due date. So, if jobs cannot be filled in a tank(s) (e.g., they are all full) the job must be filled in an IBC(s).

1.3 Research aim

More insight into the current performance of the mixing plant is required to support decisions during the improvement process of the scheduling strategy. The scheduling strategy must be improved to enable the mixing plant and thus Eurol to increase its output. The current scheduling process of the planner is largely manual and does not include the scheduling of storage units. This leaves employees of the mixing plant to schedule storage units. Considering storage units when scheduling manually can lead to an information overload for the planner. Also, when the planner is not available during the night shift employees of the mixing plant can make decisions about changes in the schedule. Due to the possible information overload and night shift decisions we propose to create an algorithm to improve the scheduling strategy. The algorithm should be able to support the planner and employees of the mixing plant in scheduling the mixing plant.

1.4 Research design

Section 1.4.1 demarcates the research scope after which Section 1.4.2 states the research problem and finally, Section 1.4.3 explains the research approach.

1.4.1 Scope

Inefficient use of mixing plant resources can have causes outside of the scheduling strategy of the mixing plant. We do not consider causes outside of the scheduling strategy of the mixing plant, e.g., the design of the plant.

As stated in Section 1.1 Eurol is a manufacturer of lubricants and technical fluids. This research only focuses on the production of lubricants. Lubricants are produced more than technical fluids and the scheduling of lubricant mixing is closer to reaching its limits, i.e., higher utilisation rate.

We do not consider the splitting of jobs (i.e., mixing jobs) unless required. Jobs can consist of multiple filling jobs (same blend different filling line) that can be split. However, filling jobs are combined to jobs by filling planners and these combined jobs can have different due dates (e.g., due tomorrow and 2 weeks). Filling planners are more capable of making these decisions because their horizon is longer. Therefore, we do not let the algorithm split jobs to not mix for a filling job because this can cause small future jobs which can be inefficient, undoing the work of filling planners making efficient jobs.

1.4.2 Research problem

In Section 1.2 we have identified the core problem, the main research question to address the core problem is as follows:

How can Eurol improve their scheduling strategy of the mixing plant so that the resources of the mixing plant are used more efficiently?

This research problem mainly focuses on modelling the problem and designing an algorithm to solve it. We state 5 research questions with their own sub-questions, answering these answers the main research question.

Analysis of the current situation
Our first objective is to map the production process flow, the current way of scheduling and quantify the current performance **(Q1)**. We require the process flow **(Q1.1)** and the current scheduling strategy **(Q1.2)** to determine the performance of the process. To provide an appropriate solution, we need to know the constraints that need to be considered **(Q1.3)**. We determine the performance of the process **(Q1.4)** to identify improvement possibilities and test our proposed scheduling strategy later. Parameter values **(Q1.5)** are important for our solution approach and provide insight into the characteristics of the problem in hand.

Q1 *How is production and the production scheduling currently organised and what is their performance?*
Q1.1 *What does the production process look like?*
Q1.2 *What is the current scheduling strategy of the mixing plant?*
Q1.3 *What constraints need to be considered?*
Q1.4 *What is the bottleneck(s) of the production process and the current performance of the production and scheduling process?*
Q1.5 *What are the values of processing parameters?*

Literature review and analysis
To provide a suitable solution we first classify the problem in hand **(Q2.1)**. After answering question Q2.1 we search for different solving approaches in literature **(Q2.2)**.

Q2 *What is known in literature about similar scheduling problems?*
Q2.1 *How can we classify the problem in hand?*
Q2.2 *What different solution approaches are there for the problem in hand?*

Solution approach
Based on our analysis of the current situation and literature review we select our solution approach **(Q3.1)**. Thereafter we can determine a suitable model of the problem to apply our solution approach to **(Q3.2)**.

Q3 *How can we provide an improved scheduling strategy for the mixing plant?*
Q3.1 *What approach/algorithm should we use to improve the schedule?*
Q3.2 *How should the mixing and subsequent storage schedule be modelled?*

Solution approach evaluation
With our last question we want to analyse the performance of our proposed solution approach which we discuss in Chapter 5.

Q4 *What is the effect of the proposed solution approach on the performance of the schedule?*

Finally, we present our conclusions and recommendations in Chapter 7.

1.4.3 Research approach

Figure 1.2 shows the research approach and an overview of the actions and results of each phase. The research consists of 3 phases, the result of each phase is needed to answer the next sub- question.

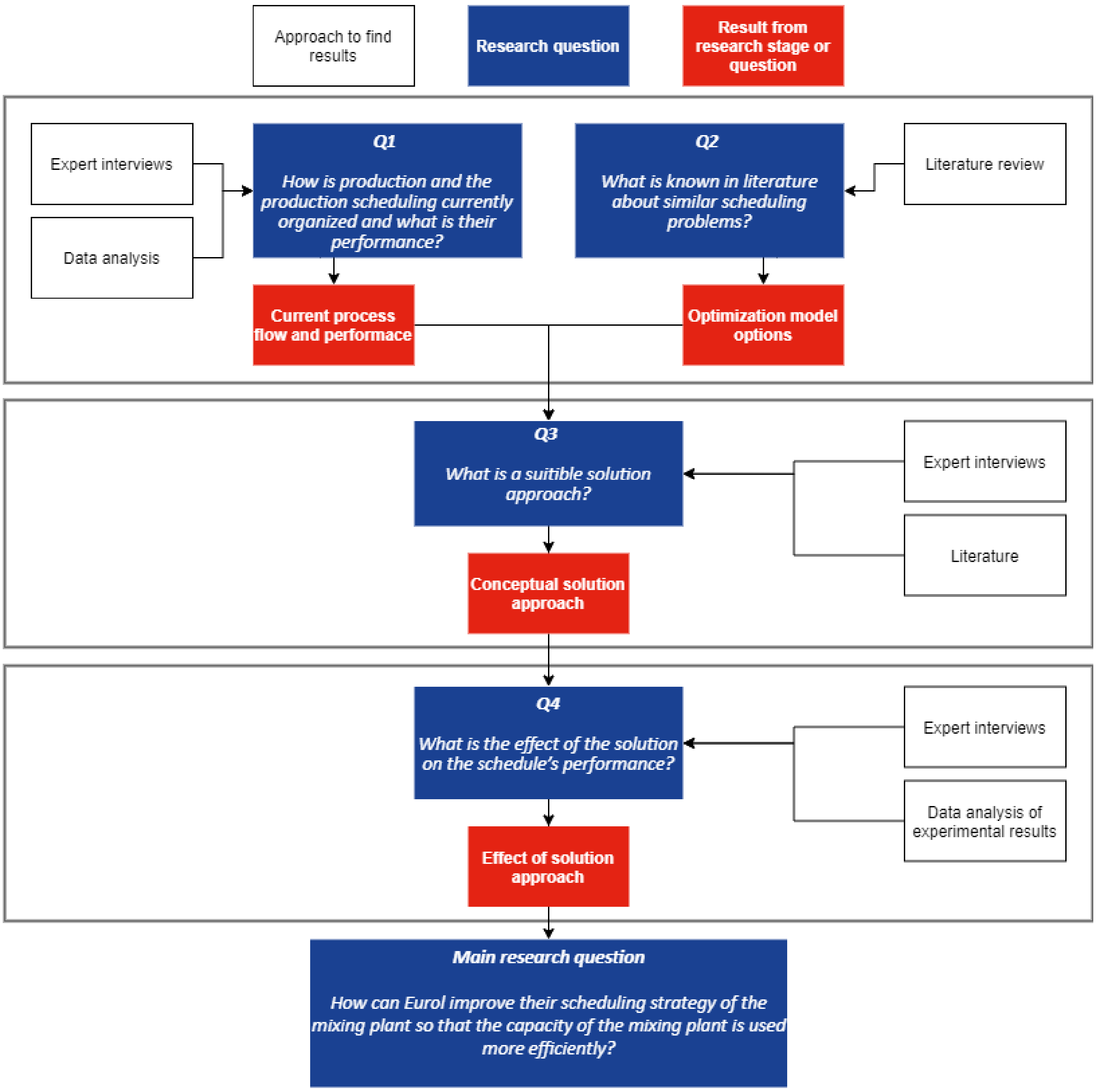

Figure 1.2 Research approach

1.5 Research deliverables

The deliverables of this research are:

- Insight into the current performance of the production and scheduling process of the mixing plant.
- An algorithm designed to support the planner and mixing plant employees in scheduling the production in the mixing plant. The algorithm does not need to have a guarantee of providing an optimal solution. It should however be able to provide a solution quickly, e.g., within 5 minutes.
- An answer to the question if, using the algorithm, mixing plant resources can be used more efficiently.

2 Current situation

This chapter focuses on research question Q1: How is production and the production scheduling currently organised and what is the performance? Section 2.1 explains the production process flow and Section 2.2 the current way of scheduling. Next, Section 2.3 determines the constraints to be considered, Section 2.4 the performance of the process and Section 2.5 determines parameter values to be used when scheduling. Lastly, Section 2.6 summarises and concludes this chapter.

2.1 The production process

This section explains the production process in more detail to answer Q1.1: *What does the production process look like?* Figure 2.1 shows the general flow of the production process and shows which section covers which parts of the production process. Section 2.1.1 covers the storage of raw materials, Section 2.1.2 the mixing process, Section 2.1.3 the storage of semi-finished products and Section 2.1.4 the filling process. Even though raw material storage and filling is out of scope we cover these processes because we think it is important for the reader to better understand the problem in hand. Section 2.1.5 discusses the future of the production process.

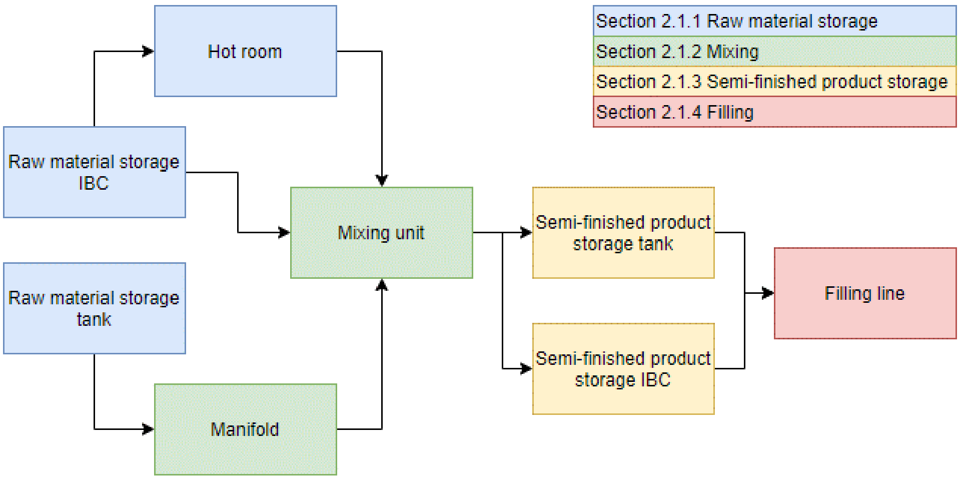

Figure 2.1 Production process flow

As Figure 2.1 shows, the production process starts with raw materials stored in either IBCs or tanks. Raw materials stored in IBCs can be warmed in the ‘hot room’ to reduce their viscosity which can be required to mix raw materials properly or to let the raw material be pumped more easily. Raw materials stored in tanks are pumped through a manifold in the mixing unit. In the mixing unit the raw materials are mixed into a homogeneous blend. After mixing, the blend is stored in tanks and/or IBCs. Finally, the blend is tapped at a filling line where containers ranging from spray cans and jerrycans to drums containing from 100 ML to 210 L are filled. Labels and lids are also applied, and the product is packed ready for dispatch.

2.1.1 Raw material storage

We differentiate raw materials in base oils and additives. Base oils often account for approximately 80% of the final product. Additives are added in small quantities compared to base oils. Additives are often added after base oils, finishing the blend’s contents, and ensuring the blend has the desired properties.

Raw material storage IBC

Additives are often stored in IBCs. When required, the IBC is picked from storage and placed in front of the mixing unit. There, a hose is placed in the IBC to pump the raw material into the mixing unit as shown in Figure 2.2.

Figure 2.2 IBC handling; pumping

Hot room

Some raw materials require a certain temperature before they can be mixed because their viscosity decreases at a higher temperature, making them easier to pump. Some raw materials also need to be heated to mix them properly. Because of these temperature requirements there is a 'hot room'. IBCs can be placed in the hot room where they need to be stored for several hours to several days to reach their desired viscosity level.

Raw material storage tank

Base oils are stored in tanks ranging from approximately 10,000 to 80,000 litres, some of which can warm their contents. When required the raw material is pumped automatically through a manifold into a mixing unit. Some raw materials labelled as additives are also stored in tanks because of heating requirements.

Peculiarities

When receiving raw materials for storage, it may happen that there is not enough capacity in the raw material storage tank. This happens approximately once a week. When this happens a 'premix' can be necessary if the viscosity of the raw material is too high. Premixing is done by filling the mixing unit with the raw material and mixing it with a base oil. It is often mixed with a low viscosity base oil to make it easier to pump later. The premix is then stored in an IBC, if premixing is not needed the raw material is also stored in an IBC. Using the IBC later during mixing can increase the mixing time due to the handling time of an IBC (pick from storage, placing hose, slower pump).

2.1.2 Mixing

Mixing units mix raw materials into a homogeneous blend. There are 4 mixing units dedicated to the production of (lubricant) blends. Because of confidentiality, we do not provide details of the mixing units, e.g., their size.

Manifolds

As mentioned, the mixing units can be filled through a manifold of which there are 2, as shown by Figure 2.3. Manifold 2 has access to 14 raw material storage tanks and manifold 1 has access to the same 14 and 12 additional storage tanks. If MU A or MU B requires raw material from a tank where manifold 2 does not connect to, the pipe shown below the manifolds in Figure 2.3 connecting the 2 manifolds can be used to use manifold 1. A manifold can only serve one mixing unit at a time, and if a manifold is in use the other cannot be in use. When mixing, the mixing department considers the use of the manifolds. In case of conflict, the general rule is to start the job that takes the least amount of manifold usage time first.

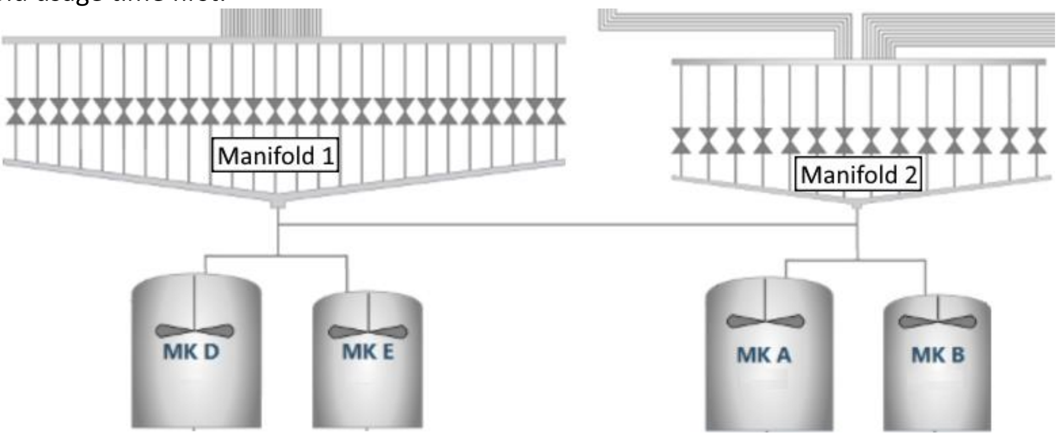

Figure 2.3 Mixing units and manifolds

Mixing process

Figure 2.4 shows a simplified example of a mixing process. As mentioned in Section 1.1, Eurol makes approximately 700 different blends. blends can differ to such an extent that if residues remain in a manifold or mixing unit, contamination may occur. For this reason, the blends are divided into 14 rinsing groups. When changing the rinsing group, it may be necessary to rinse the manifold and mixing unit. We also refer to rinsing as a changeover. Rinsing takes place in batches of 30 kg of which more than one may be needed, depending on the change in rinsing group. The raw material used for rinsing is often a base oil, which base oil is dependent on the change in rinsing group. During rinsing, the raw material used for rinsing is collected in a rinsing IBC. Section 2.3.1 explains rinsing in more detail.

After rinsing, the base oil is 'automatically' (from a tank) pumped into the mixing unit, after which additives are often added from IBCs. When the blend is finished, a sample is taken and tested in the laboratory (referred to as lab). If the lab confirms the blend meets the specifications, the blend is pumped out to storage. If the lab concludes that the blend does not meet the specifications, an adjustment and/or additional pumping may be necessary. An adjustment involves adding some raw material(s). If an 'automatic raw material' is needed, the manifold is needed again, which can cause a delay because it may be in use and may need to be rinsed. Additional pumping means an extension of the mixing time to make sure the blend is homogeneous.

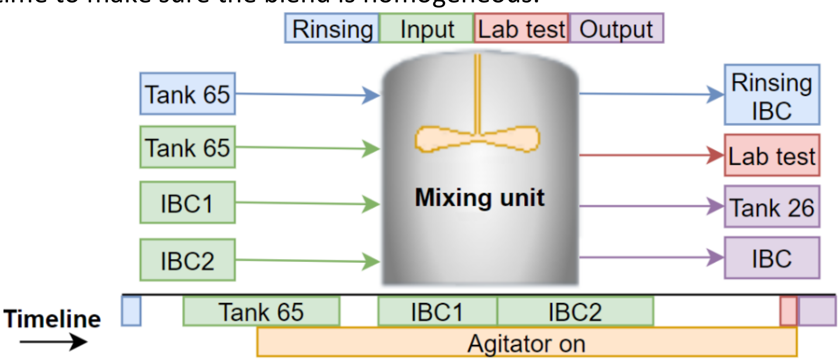

Figure 2.4 Simplified example mixing process

Peculiarities

Customers can order in bulk, which means they can receive their blends in 'bulk', i.e., not packaged. Figure 2.5 shows Eurol's bulk truck which delivers bulk shipments to customers. The bulk truck has seven compartments ranging from 950 to 4,200 litres, each of which may contain different blends. The truck cannot be loaded by storage tanks, IBCs are used for this purpose. One customer also picks-up bulk shipments with its own trucks. These trucks may have a different compartment layout ranging from 1,000 to 30,000 litres. There is an aversion to use IBCs if the bulk load is large (10,000 litres) because of the handling time. However, as these trucks cannot be loaded from storage tanks, they must be filled directly from the mixing unit if IBCs are not used. This may lead to waiting times if the schedule of the mixing unit and truck are not perfectly aligned. Bulk trucks are preferably filled between 08:00 and 17:00.

Figure 2.5 Eurol's bulk truck

There are several rinsing IBCs, one for each base oil used in rinsing. The rinsing IBCs are regularly tested by the lab, after which the lab assigns part of the contents to jobs to empty the rinsing containers. When the contents of a rinsing IBC are added to a mixture the mixing time increases because of handling time.

Filling jobs require more fluid than needed to fill all containers. This is due to variations in the filling process, for example, it can happen that some of the blend is spilled. Blends that remain in storage tanks is drained off in an IBC, so that the tank is again available for the mixing department. The IBC is later added to another similar mixture by the mixing department, which may require a longer mixing time.

As mentioned earlier, some raw materials need to be heated before mixing to reduce their viscosity. When a mixture is ready it can still be warm. Filling lines cannot handle mixtures that are too hot because the low viscosity causes the filling machine to leak and spill. For this reason, mixtures may need to be cooled when they leave the mixing unit via an oil cooler. There is one oil cooler available that resists the flow from the mixing unit and thus reduces the flow.

2.1.3 Semi-finished product storage

Storage unit is an umbrella term for both tanks and IBCs. There are 24 tanks dedicated to the storage of blends of which Figure 2.6 shows 6. Because of confidentiality, we do not provide details of the storage tanks, e.g., their size.

Figure 2.6 6 Blend storage tanks

Input; mixing
Storage units are filled by mixing units, every mixing unit can fill any tank. Tanks are the preferred storage media because of the lost time when filling IBCs. We call this lost time handling time. The handling time is caused by placing of IBC(s) near the mixing unit, as shown in Figure 2.2. In addition, pumping to an IBC is slower because it is pumped through a smaller pump. Each filled IBC requires a label such that it can be added to the stock system and a sample must be brought to the lab. After filling, the IBC(s) must be picked up and moved to storage, which may cause delay. Tanks may need to be rinsed depending on the change in rinsing group, rinsing of tanks if often done during mixing. IBCs may also need to be rinsed, but the rinsing of IBCs does not affect the availability of IBCs because there are enough available.

Output; filling
Storage units provide the filling lines with blends. Not every filling line can connect to every tank, Section 2.3.2 elaborates on these restrictions. Each filling line can be connected to an IBC, but some filling lines lose a lot of time when using IBCs. For example, line 4 is a relatively fast line and has room for only one IBC next to it, the IBC can be empty in minutes. For this reason, IBCs are not preferable for line 4.

We are unable to accurately predict the time at which a tank will be empty. This is due to variation in the filling time. This also causes variation in the start time of filling lines. Section 2.4.2 analyses the variation of the start time. Further difficulty arises because of the speed of the filling line, it increases as the filling job progresses. This is due to finetuning of adjustments in the line after switching bottles and/or blend.

Peculiarities
Some tanks are dedicated to certain blends to prevent excessive rinsing. For example, 2 tanks are dedicated to hydraulic oil. This however does not mean that other tanks cannot be used for hydraulic oil. Another tank is dedicated to one blend that is sold very often, also to prevent excessive rinsing.

2.1.4 Filling

There are 15 filling lines differing in speed, connectivity, and shift schedules. Speed differs because of the amount of automation within the line (e.g., automatic packing robot) and the products made by the line (e.g., 1 L cans, 210 L drums). Connectivity to storage units differ per line, Section 2.3.2 explains in more detail. A filling line can for example connect to 12 tanks (out of 24). The filling department also prefers not to use IBCs on some lines because the limited space available to place IBCs near the line. Also, if the speed of a line is very high it can drain IBCs very quickly requiring a lot of IBCs to be moved at a high rate. Most lines are operated in 2 shifts per day. However, the filling department operates 24/5, meaning there are different combinations of lines running throughout the week.

Peculiarities
If lines require the same blend at the same time, they cannot always drain the same tank. If one line is relatively fast compared to the other, they cannot drain the same tank because the slower line would get air in its pipes. Air in pipes of a filling line causes disruptions such as spillage because there is no continuous stream.

Both mixing units and filling lines experience disruptive events. Disruptive events at filling lines can be spillage when filling, unavailability of material (e.g., caps, labels). A disruptive event can also be that the blend is not available, for which the mixing plant is accountable. This however does not happen often, approximately 99% of the time the blend is available. We do not go into further detail regarding disruptive events of filling lines because this is out of scope.

2.1.5 Future

In 2021 Eurol wants to remove some filling lines and replace them with a faster line, increasing the total theoretical filling output. Also tanks 45 to 49 will be removed and replaced by more storage tanks, increasing the total storage capacity. This reduces the importance of the storage schedule but increase the importance of the mixing unit schedule. Currently, the mixing plant assumes that the storage is the bottleneck in the mixing plant. The mixing plant assumes this bottleneck moves to the mixing units. Therefore, Eurol is also planning on adding a mixing unit.

2.2 Scheduling of the mixing plant

An order generates demand at the filling plant which generates demand at the mixing plant to provide it with blends. Section 2.2.1 explains the current scheduling process of mixing units and Section 2.2.2 explains the current scheduling process of storage.

2.2.1 Scheduling mixing units

On average, there 19 jobs per day (up to approximately 30) for MU A, B and D, which means that there are approximately 180 jobs on average over the 3-day scheduling horizon. The jobs that the mixing planner receives may consist of several filling jobs (possibly from different filling lines) of the same blend. During the day, the schedule is adjusted based on the progress in the mixing and filling plant. Twice a day, new jobs come in, triggering the need to reschedule. Below, in 4 steps, the current scheduling strategy is explained.

<u>Step 1: assign jobs to mixing units</u>
Jobs are assigned to mixing units mostly based on their size. Jobs over 16,000 kg are always assigned to MU D. Jobs under 16,000 kg can be split to MU A, but splitting is not preferred and only applies if MU D is over utilised. Jobs between 8,000 to 5,000 kg are always assigned to MU A and jobs between 850 to 1,700 kg are always assigned to MU B because of the mixing unit size constraints. The remaining jobs between 1,700 to 5,000 kg are split based on planner's judgement, mostly based on rinsing requirements and utilisation rates.

<u>Step 2: sort jobs per mixing unit</u>
The jobs per mixing unit are then sorted, for MU D mostly based by due date, the other mixing units more by rinsing group. MU D is mostly sorted by due date to reduce long-term occupancy of storage tanks. Jobs on the other mixing units are sorted more by rinsing group to reduce rinsing costs. In addition, small(er) jobs less often filled into tanks, so larger jobs can be filled into tanks to ensure higher filling rates. Moreover, small jobs are often for slower lines, so tanks would stay full longer. Rinsing is required when changing rinsing group and is scheduled to take 18 minutes. Section 2.3.1 explains rinsing in more detail.

<u>Step 3: add jobs to schedule</u>
After sorting, jobs are appended to the schedule of the mixing unit which, at the start of a day, already has a schedule for approximately 2 days. Jobs are assigned to one of the 3 days by the planner. Then the start times are assigned automatically to the jobs. The first job gets a start time based on the current time. Jobs thereafter start from the end time of the previous job. If the end time of the last job for that day is before 22:45 then slack is scheduled and the first job for the next day starts at 22:45.

<u>Step 4: finalise</u>
Finally, the schedule is checked for infeasibilities and, if necessary, adjustments are made in consultation with the filling planners and the mixing department. If the schedule is feasible, it is printed out and taken to the mixing department.

The mixing times are calculated based on empirical data from the past 2 years. An estimate is made of the time per kg produced, which is then multiplied by the quantity of kg to be produced. Section 2.5.1 discusses the calculation of mixing times in more detail.

Note that the mixing planner does take into account the use of manifolds or the storage schedule. This is controlled by the mixing department. The mixing department can change the schedule in consultation with the mixing planner. Changes can consist of delaying or advancing job(s) in the schedule, changing the order, or splitting jobs. If the mixing planner is not available, for example during the night shift, the mixing department tries to keep changes to a minimum while maintaining feasibility.

2.2.2 Scheduling storage

If a job is a combination of filling jobs, the filling planners write on the job which part of the job must go in IBCs. This is to prevent a part of a job from remaining in a storage tank for a long time because of different due dates. The rest of the storage schedule is handled on the work floor of mixing department. Initially, there is no communication between the mixing planner and mixing department about the storage. Communication about the storage only takes place when problems arise, for example a large job is moved because there are no storage tanks available. Storing in tanks is preferred over IBCs because tanks have a lower handling time.

The mixing department adheres, not strictly, to guidelines shown in Table 2.1 to decide whether to use an IBC or a tank, e.g., if a job for filling line 1 is above 3,000 litre (>3K L) the mixing department should fill a tank. The mixing schedule indicates which filling line(s) requires the blend so that the mixing department knows which tanks they can use (due to tank to filling line restrictions). When scheduling storage tanks, the filling line schedule is consulted by the mixing department to get an indication when storage tanks are empty.

Table 2.1 Storage guidelines IBC or Tank per filling line, A = Always, N = Never

Filling line nr.	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17
IBC	<3K L	A	A	N	<3K L	A	<4K L	<3K L	<4K L	A	<3K L	<4K L	A	A	A
Tank	>3K L	N	N	A	>3K L	N	>4K L	>3K L	>4K L	N	>3K L	>4K L	N	N	N

2.3 Constraints to be considered

In this section we cover the constraints to be considered. There are 2 constraints namely, rinsing and storage constraints, covered by Section 2.3.1 and Section 2.3.2 respectively.

2.3.1 Rinsing

There are 14 rinsing groups which specify the type of blend, for example gear oil or hydraulic oil. The amount of rinsing batches (30 kg) required may vary from 1 to 4, depending on the change in rinsing group and the amount to be produced. Within rinsing groups, there may be exceptions requiring more rinsing batches. Hydraulic oil is an oil that requires more rinsing because it is easy to contaminate. That is why tank 50 is dedicated to a hydraulic oil. This is also the reason why sometimes large batches are filled in IBCs, to prevent excessive rinsing. Also, MU D mixing 24 tonnes or MU B mixing 5 tonnes makes a difference, a large job is affected less by remnants of a previous batch. The same applies to the storage thereafter. Mixing plant employees use their knowledge to determine the amount of rinsing required, we elaborate upon the reality of rinsing in Section 2.4.1. After rinsing the employee brings a sample of the rinsing fluid to the laboratory to check whether the rinsing fluid is sufficiently 'clean'.

2.3.2 Storage constraints

As mentioned earlier, not every filling line can connect to every tank, Table 2.2 shows these constraints. It may be impossible or there may be an aversion to the use of a tank because a line might only be able to connect to it by manually connecting a hose, possibly with the addition of a pump.

Table 2.2 Storage constraints, green = possible, red = impossible, yellow and grey = separate hose with and without pump

	Filling lines														
Storage tank	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
26															
28															
29															
30															
31															
32															
33															
34															
35															
36															
37															
38															
39															
40															
41															
42															
43															
44															
45															
46															
47															
48															
49															
50															
Total capacity															

2.4 Current performance

In this section we elaborate on the current performance of the mixing plant. Section 2.4.1 elaborates upon the performance of the production process and Section 2.4.2 on the scheduling process.

2.4.1 Production process

In this section we elaborate upon the performance of the production process by multiple criteria namely: disruptions, mixing units, storage tanks and lastly the efficiency measure.

Disruptions

The mixing department monitors the disruptions in their process. A disruptive event is something that interrupts the continuation of some activity or process. Applied to the mixing plant, everything that disturbs the continuation of mixing is a disruption, meaning, e.g., rinsing is a disruption even though it can be foreseen. Disruption monitoring gives an indication of the variability of the process and where this variability comes from. Table 2.3 shows the number of minutes lost due to the top 10 (of 39) disruptions from 25-5-2020 to 19-11-2020. We note that the tracking of disruptions is in the hands of operators in the mixing plant. As a result, the data may deviate from reality, how much deviation this causes is unknown.

A 'raw material unloading' disruption occurs when someone from the mixing plant is needed to unload incoming raw materials. 'No work preparator available' means there was no one to pick up and bring IBCs to the mixing department.

There are 2 disrupting events influenceable by the scheduling strategy, namely rinsing and the waiting for manifolds. Scheduling mixing orders of the same rinsing group after each other reduces the amount of rinsing required. Scheduling such that mixing orders do not start on a mixer within the manifold usage time of another mixer reduces the waiting on manifolds. We now discuss the 'performance', i.e., impact, of both these events per mixing unit.

Table 2.3 Top 10 mixing plant disruptions, colours add perspective to disruption size

Description	Time in minutes	% of total
Loading bulk truck	655	1.65
Loading customer truck	690	1.73
Additional pumping	973	2.45
Raw material unloading	1,075	2.70
No work preparator available	1,405	3.53
Adjustment	3,815	9.59
Waiting for manifold	4,010	10.08
Other	4,595	11.55
Break	6,095	15.32
Rinsing	11,812	29.70
Total top 10	35,125	88.31
Total all disruptions	39,775	100
Total mixing time	443,664	

Disruption; rinsing

Rinsing causes multiple costs, the most prominent being labour cost. Labour cost is caused by the initial rinsing time but also by, for example, rinsing IBC (Figure 2.4) handling and using rinsing IBCs for future orders. We focus on the initial rinsing time lost. Table 2.3 also only shows that cost as time lost in minutes.

To determine the impact of rinsing per mixing unit we determined the scheduled rinsing time and actual rinsing time in minutes, shown by Table 2.4 with data ranging from 1-9-2020 to 30-10-2020. Most notably we see that the actual rinsing time for MU D is much lower than the planned rinsing time, also compared to other mixing units (rows 1 and 2). As mentioned in Section 2.3.1, MU D is affected less by remnants of the previous batch because of the batch size. Employees of the mixing plant know this and apply their knowledge to be more efficient, resulting in a large deviation in scheduled and actual rinsing time for MU D. However, we also see that the actual rinsing time for other mixing units is lower than the scheduled rinsing time. This might be due to incomplete registration of actual rinsing time. Because of this reason we also looked at the amount of rinsing batches used which is better registered (row 3). Every mixing unit uses approximately 3 batches per change in rinsing group, rinsing is scheduled to always take 18 minutes. From there we can calculate the expected rinsing time where we find the same trend namely, MU D is rinsed less often compared to MU A and B which are also rinsed less than scheduled (row 4).

$$Expected\ rinsing\ time = \frac{Rinsing\ batches}{3} * 18$$

If we divide the scheduled rinsing time by the number of orders scheduled, we get the scheduled amount of rinsing time per order, shown by Table 2.4. Table 2.4 shows that this ratio is roughly the same for each mixing unit (rows 5, 6 and 7). If we look at the ratios of the actual and expected amount of rinsing time per order, we see that MU D clearly has less rinsing time per order compared to the other mixing units (rows 8 and 9).

Interestingly, Table 2.4 shows that the scheduled rinsing time per order is approximately equal for every mixing unit (row 7). This, however, does not mean the focus on reducing rinsing is equal for every mixing unit. For example, the average amount of rinsing groups a mixing unit must deal with differs per mixing unit, as shown by Table 2.4 (row 10). So even though MU D is dealing with less rinsing groups per day the scheduled rinsing time per order is approximately equal to MU A and B. This indicates that the focus on rinsing differs per mixing unit and that the focus on rinsing is lowest for MU D in the current scheduling strategy.

Table 2.4 Rinsing analysis 25-5-2020 to 19-11-2020

		MU		
Row	**Description**	**A**	**B**	**D**
1	**Scheduled rinsing time in minutes**	2,790	2,916	2,646
2	**Actual rinsing time in minutes**	1,477	1,938	365
3	**Rinsing batches**	344	350	110
4	**Expected rinsing time in minutes**	2,064	2,100	660
5	**Orders scheduled**	329	322	299
6	**Actual orders mixed**	319	325	278
7	**Scheduled rinsing time per order in minutes**	8.48	9.06	8.85
8	**Actual rinsing time per order in minutes**	4.63	5.96	1.31
9	**Expected rinsing time per order in minutes**	6.47	6.46	2.37
10	**Average rinsing groups per day**	3.42	3.98	2.76

Disruption; waiting for manifold

A manifold can only serve one mixing unit at a time and only 1 manifold can be used at a time, this can cause waiting time. The utilisation rate and the waiting time on manifolds of each mixing unit is shown by Table 2.5 based on data ranging from 25-5-2020 to 30-11-2020. The waiting time follows from the record of disruptions from the mixing department. The last column of Table 2.5 shows the percentage with which the manifolds caused disruptions.

Table 2.5 Manifold analysis (Appendix A Q1.5), colours add perspective within column(s) delineated with a thick border

	Utilisation rate in %		Waiting time in minutes per MU					
Period	**Manifold 1**	**Manifold 2**	**A**	**B**	**D**	**E**	**Total**	**% of all disruptions**
June	26.69	3.99	140	235	70	0	445	6.31
July	23.77	3.85	185	285	113	25	608	9.84
August	24.34	4.19	255	305	90	0	650	10.57
September	32.51	4.63	275	400	155	0	830	12.70
October	35.07	4.95	200	220	170	0	590	7.33
November	38.53	5.92	410	597	210	5	1,222	15.11
25-5– 30-11	28.88	4.39	1,465	2,082	913	30	4,010	10.31

In months of lower production output, such as July and August, the occupancy rate of manifold 1 can be 24%, while in months of high production, such as October and November, the utilisation can be up to 39%. The utilisation rate of manifold 2 is much lower, namely between 4 and 6%. From the table it is clear that a higher manifold utilisation rate causes more waiting time because when the rate increases from June to November the waiting times also increase. This is true, except for June and October for which we have no definitive explanation.

Because Eurol aims to increase production output by 10% the manifolds become more disruptive. However, Table 2.5 also shows that MU A and B must wait the most while MU D is not affected as much. Eurol also aims to produce larger volumes and reduce small volume sales, this should lower the utilisation rate of MU A and B making manifold waiting less critical to the production process.

Mixing units

The filling and utilisation rate are indicators for the production load. The filling rate indicates the size of mixing orders produced on the mixing unit. A high utilisation, e.g., 90%, means that the MU is under a heavy load indicating a possible bottleneck. Table 2.6 shows the results of the analysis.

Table 2.6 Performance mixing units, colours add perspective within column(s) delineated with a thick border

	Filling rate in %				Utilisation rate in %			
Period	**A**	**B**	**D**	**E**	**A**	**B**	**D**	**E**
2018	54.91	30.18	65.11	53.78	73.64	73.46	68.98	1.83
2019	61.91	35.81	69.79	40.07	72.28	68.43	74.12	3.67
2020 (till 23-11)	61.90	34.06	71.43	52.76	64.92	60.81	67.62	4.03

The filling rate of MU D increases every year. Indicating that either Eurol is succeeding in its goal of increasing large volume sales or, filling planners are increasingly combining filling orders into bigger mixing orders. MU A and B follow a similar trend, except for 2020, for which corona could be a factor. The filling rate of MU B however is relatively low indicating that the average mixing order on MU B is approximately 1,700 litres. This indicates that MU B does not often fill tanks (see also Table 2.1). The utilisation rate of a mixing unit increases for MU D except for 2020 which is caused by corona which caused Eurol to sell less of their products. However, the difference between MU A and D is increasing. We can also deduce that MU B is used less over the years. These findings are in line with Eurol's objective to reduce sales of small volume sales and increase large volume sales because MU D is larger than MU B.

Storage tanks
The current use of storage tanks is an important characteristic of the current situation indicating the improvement possibilities. If tanks are used efficiently there is not much room to improve our efficiency measure.

The consensus within the mixing department is that the storage tanks are the bottleneck. From data ranging from 25-5-2020 to 19-11-2020 we extracted the average time a tank is full (buffer), the average empty time, the number of times it was empty, the number of times it was filled, the average litres filled when filled and the total amount of kg it held during that time. Note there is a difference between the number of times a tank was empty and filled because a tank can be filled with the same blend before it was empty. Figure 2.7 shows the use of a tank over time and Table 2.7 the tank performance data. Some tanks are dedicated to a certain blend or rinsing group which is mentioned in the last column of Table 2.7.

Table 2.7 shows that tanks can be full for a day before being used by a filling line which is significant. Also, the time a tank is empty is also significant according to employees at Eurol. These values indicate that there possibly is room for improvement. Because most tanks, on average, are full for 20+ hours and empty for 12+ hours. Meaning that with a better scheduling strategy we can possibly reduce the time a tank is full and empty to use the tanks more efficiently and thus fill more unconstrained litres. Tanks 45 to 49 have a worse performance than other tanks even though they are not dedicated. This is because these tanks can only connect to filling lines 7 and 11 as shown by Table 2.2.

Note that storage units are used to flatten demand peaks of the filling plant but also as a buffer to deal with variations in the mixing and filling process.

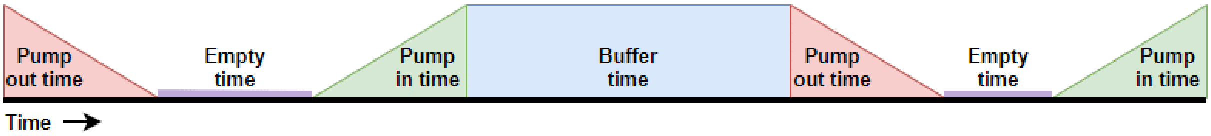

Figure 2.7 Tank usage over time

Table 2.7 Performance storage tanks, colours add perspective within column(s) delineated with a thick border

Name	Size in L	Buffer time HH:MM	Empty time HH:MM	Number of times empty	Number of times filled	Average kg filled	Kg sum	Kg sum/size L	Dedicated
CT 30	5,300	25:24	15:27	62	66	3,746	247,262	46.65	
CT 31	5,300	24:30	18:41	57	63	3,563	224,485	42.36	
CT 33	5,300	26:24	25:53	52	53	3,877	205,461	38.77	
CT 34	5,300	24:54	24:35	53	58	3,693	214,190	40.41	
CT 40	5,300	25:31	30:53	50	54	3,735	201,665	38.05	
CT 41	5,300	27:04	23:22	47	51	3,574	182,278	34.39	
CT 28	10,500	22:58	14:46	62	68	7,186	488,627	46.54	
CT 29	10,500	26:08	12:47	61	67	6,784	454,507	43.29	
CT 32	10,500	22:40	16:32	60	70	7,633	534,320	50.89	
CT 35	10,500	21:17	14:49	65	75	7,386	553,950	52.76	
CT 36	10,500	23:46	16:01	59	68	7,334	498,694	47.49	
CT 37	10,500	22:24	18:08	49	52	7,380	383,738	36.55	
CT 38	10,500	20:15	55:18	32	41	7,143	292,854	27.89	
CT 39	10,500	20:44	50:15	32	41	7,790	319,390	30.42	
NT 45	14,500	27:31	18:39	45	52	6,864	356,946	24.62	
NT 46	14,500	26:17	39:52	41	43	6,886	296,112	20.42	
NT 47	14,500	31:49	32:40	41	45	6,834	307,512	21.21	
NT 48	14,500	31:15	44:21	36	41	7,697	315,585	21.76	
NT 49	14,500	31:49	37:31	34	37	6,430	237,921	16.41	
CT 26	16,400	28:36	14:04	49	61	11,101	677,147	41.29	
NT 42	29,000	14:35	12:55	46	81	19,947	1,615,723	55.71	
NT 43	29,000	21:11	8:52	47	66	19,850	1,310,092	45.18	
CT 50	32,500	25:23	60:54	9	23	10,471	240,836	7.41	X
NT 44	37,000	17:00		0	29	19,466	564,500	15.26	X

Efficiency measure

Part of our efficiency measure relates to the number of litres filled in IBCs when tanks could be filled, i.e., unconstrained. Table 2.8 shows the number of unconstrained litres filled from 1-9-2020 to 19-11-2020.

Table 2.8 Number of unconstrained litres filled

	Unconstrained litres in IBC	Unconstrained litres in IBC over limit (limit Table 2.1)	Percentage of litres unconstrained
MU A	42.16%	17.40%	56.45%
MU D	5.50%	3.00%	60.00%

From Table 2.8 we deduce that there is room for improvement. First, because MU D and especially MU A have room for improvement (column 2). Second and more importantly, the limits determined by Eurol (column 3) are crossed. Indicating that either the limits are unrealistic, or the scheduling in the mixing plant can be improved.

2.4.2 Schedule performance

The rate with which the schedule is in accordance (i.e., how much does reality adhere to the schedule) with reality when the mixing planner is absent indicates how much employees of the mixing plant change the schedule and the level of uncertainty. Accordance over multiple days indicates the amount of re-optimisation in the scheduling strategy of the planner and the level of uncertainty. The amount of accordance also indicates the amount of allowable nervousness when scheduling. Low accordance indicates frequent revisions, i.e., nervousness. Nervousness can have cost associated with it, e.g., wasted setups (Kopanos, Capón-García, Espuna, & Puigjaner, 2008). Allowable nervousness in our case is constrained by the hot room capabilities. Due to the hot room having a limited space, delays could cause the hot room to become full. Also, raw materials can require heating for several hours which can cause unavailability when a 'hot' raw material is required on short notice.

Mixing schedule versus reality

Before showing the results we first explain how we calculate accordance in the next paragraph with help of Figure 2.8. Figure 2.8 shows scheduled mixing orders in red and the reality with which these orders were mixed in blue, divided into blocks of one hour.

We calculate accordance in 2 ways, the first is to divide hours to which the schedule and reality overlap by the scheduled mixing time (scheduled). The second is to divide the overlapping hours by the actual mixing time. Figure 2.8 shows 4 examples (indicated with larger bold numbers) of which their accordance results are shown in Table 2.9.

Table 2.9 Example calculations

Example	Scheduled	Actual
1	33%	50%
2	50%	33%
3	60%	100%
4	100%	60%

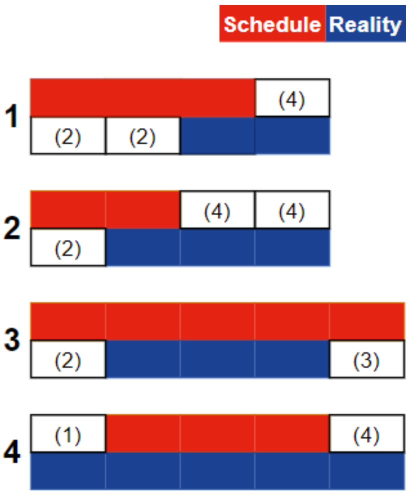

Figure 2.8 Calculation explanation

$$Scheduled = \frac{overlapping\ hours}{Scheduled\ mixing\ time} \qquad Actual = \frac{overlapping\ hours}{Actual\ mixing\ time}$$

A mixing schedule is made definitive at approximately 17:00 every working day. For this reason, day 1 is only 7 hours; from 17:00 to 0:00. Day 2 is 24 hours, from 0:00 to 0:00 and day 3 is also 24 hours. Table 2.10 shows how much of reality is in accordance with the schedule based on data from 1-9-2020 to 23-11-2020.

Table 2.10 Accordance schedule versus reality

Day	Scheduled	Actual
1	47.54%	59.98%
2	25.21%	25.50%
3	17.80%	17.68%

We have also calculated the time with which on average a mixing order is started early or late. There are 4 possibilities indicated in Figure 2.8 by numbers ranging from (1) to (4). An order can be started early **(1)**, started late **(2)**, finished early **(3)** or finished late **(4)**. Table 2.11 shows these results per possibility, showing the average time in HH:MM:SS and a column to the right of that the number of orders that fit the description. The amounts between, for example, (1) and (3) differ due to the inaccuracy of the mixing time.

Table 2.11 Reality early or late (HH:MM:SS)

Day	(1)		(2)		(3)		(4)	
1	02:36:50	130	03:52:30	177	02:56:26	115	04:03:41	192
2	04:16:40	731	04:41:48	487	04:21:05	719	04:59:06	499
3	05:24:02	544	08:05:39	467	05:24:43	540	08:20:22	471

From Table 2.10 and Table 2.11 we can deduce that nervousness is high. Because the accordance is deemed low (Table 2.10). Also, from Table 2.11 we can deduce that orders on day 1 are more often produced late (2) then early (1), while orders from day 2 are more often early (1 compared to 2). Furthermore, interestingly we can deduce from Table 2.11 that the mixing department does not like to wait because orders from day 2 are often produced early. This is also in line with statements made by the mixing department. Mixing early can cause inefficient use of storage tanks due to prolonged occupation.

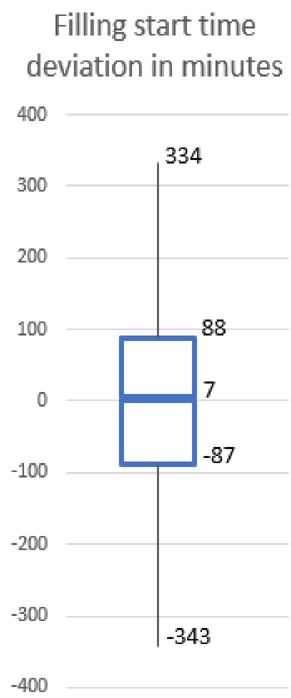

Figure 2.9 Filling line start time deviation in minutes (boxplot)

Filling lines

It is important to know when a tank is empty. This depends on which filling orders a tank serves, their start times and their filling times. If multiple tanks contain the same blend, the filling plant uses the tank that is empty first, i.e., the tank with the lowest amount of blend, is emptied first. The filling time used for scheduling is 75% of the theoretical speed because of disruptions. Disruptions cause filling lines to lag behind the schedule if remained unconsidered, therefore the 75% rule is applied to create a buffer. Furthermore, the theoretical speed is hard to determine because of the increasing speed of a filling line. This causes the actual speed of a filling line to also depend on the size of a filling order. Figure 2.9 shows the variability of the start time of a filling order in minutes. From Figure 2.9 we deduce that the variability is so high that the schedule could become infeasible when left unaccounted for within in the time horizon of 3 days.

2.5 Parameter quantification

In this section we explain how parameters are quantified. Section 2.5.1 explains the quantification of the mixing time and Section 2.5.2 the quantification of the filling time.

2.5.1 Mixing time

As mentioned earlier, the mixing time is calculated based on empirical data from the last 2 years. The mixing time begins approximately when the first litre of a raw material is pumped in a mixing unit and ends when the last litre of the mixture is pumped away. There is a record of disruptions which is kept by the mixing department, disruptions increase the mixing time. We analyse the effectiveness of the mixing time calculation by comparing the calculated mixing time with the actual mixing time of mixing orders from 25-5-2020 to 19-11-2020. Figure 2.11 shows the mixing time deviation in minutes, Figure 2.10 the mixing time in minutes.

The average scheduled mixing time is 128 minutes (2 hours and 8 minutes). 25% of the mixing orders (total 1,217), considering disruptions, exceed the scheduled mixing time between 1 and 29 minutes. Note that we do not know the relationship between volume and variability, mixing time is not an indicator for volume produced. We also do not know the relationship between mixing time and variability, but we presume that a longer mixing time is subject to a higher variability.

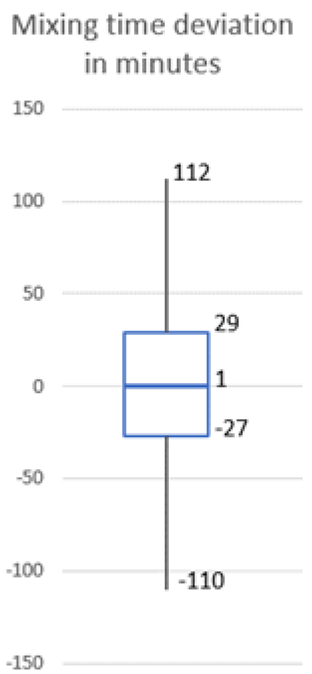

Figure 2.11 Mixing time deviation in minutes (boxplot)

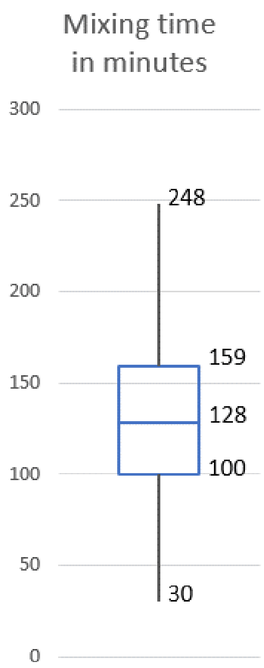

Figure 2.10 Mixing time in minutes (boxplot)

2.5.2 Filling time

Filling and start times of filling orders can be retrieved from the filling schedule. Just like the processing times of mixing the filling times are uncertain, which in turn causes start times to be uncertain. We do not go into further detail regarding filling times as this is out of scope, we only make note of its uncertainty.

2.6 Conclusions

In this chapter, we first explained the production process, the current scheduling strategy, and the constraints applicable to the scheduling of the production process. The production process is complex, and the planner has no decision support. Only the assignment of start times to jobs is automated. There are numerous constraints that must be considered, further complicating the scheduling problem.

Second, we analysed the current performance of the mixing plant and the scheduling process. We conclude there are 2 disruptive events that can be influenced by the scheduling strategy of the mixing plant, namely waiting for manifolds and rinsing. Then we conclude that, compared to rinsing, waiting for manifolds is a minor disruptive event. Especially when considering that MU B and A are most affected. It is positive that it affects mostly the smaller mixing units, as these are less important to the production process now and in the future because Eurol is moving to producing bigger volumes and reducing small volume sales.

After analysing the disruptive events, we conclude there is room for improvement in terms of the storage schedule. We assume that buffer times of tanks can be reduced, as they are full for approximately 20 hours on average. Furthermore, we conclude that the guidelines set up by Eurol to guide the scheduling of storage are not strictly followed. This means that either the guidelines are unrealistic, or the scheduling strategy can be improved. We also conclude there is a significant difference between the schedule and reality. We presume this is mainly due to the variation in the production plant. The mixing times and due dates vary significantly. Since the variance is significant, a schedule with a scheduling horizon of 3 days cannot be implemented in practice. We therefore conclude that this variation must be taken into account. To inform Eurol and verify the data a data presentation was given.

Finally, we have explained how parameters are quantified empirically. We conclude that parameters can deviate significantly from reality. The mixing time can deviate which can cause the mixing plant to deviate from a schedule. The same applies to the filling plant. This can cause tanks to remain full longer or be available earlier. We conclude that the scheduling strategy must take into account the uncertainty of these parameters.

3 Literature review

This chapter focuses on research question Q2: What is known in literature about similar scheduling problems? Q2 has 2 sub-questions, Section 3.1 focusses on the first sub-question: classifying the problem. Section 3.2 discusses different solving approaches used in literature. Finally, Section 3.3 summarises and concludes this chapter.

3.1 Problem classification

In this section we classify the problem in hand. We need the classification when looking for similar problems discussed in literature. We follow a funnel principle in classifying our problem. We start with a more general, but widely used, classification. Then we narrow down to more specific classifications used in literature and finally we classify our problem most clearly, an equal problem is not found in literature.

The production plant of Eurol is a Hybrid Flow Shop (HFS) with intermediate storage. In a flow shop, the orders always go through the shop, i.e., the stages, in the same sequence. An HFS is a flow shop with several parallel machines per stage (mixing units and filling lines) (Rubén & José, 2009). However, we focus only on the mixing plant, which is a single-stage in the production plant (mixing unit stage) with storage afterwards. Fuchigami & Rangel (2018) survey case studies in production scheduling. The problem of a single-stage with parallel machines was found in only 2 studies (4.35% of cases), indicating potential for further research (Fuchigami & Rangel, 2018).

A well-known problem classification for deterministic sequencing and scheduling is the 3-field problem classification α|β|γ of Graham, Lawler, Lenstra, & Kan (1979). Based on this problem classification, we classify our problem as:

$$R \mid res, r_i, d_i, s_{ij} \mid E_{max}$$

First, we have α = R meaning we have unrelated (different sizes) parallel machines. Second, we have β = res, r_i, d_i, s_{ij} meaning there are limited resources (e.g., tanks, mixing units), release dates and due dates may differ per order, and there are sequence dependent setup times. Last, γ = E_{max} means we want to minimise the maximum earliness. We want to minimise the maximum earliness to reduce storage cost, we elaborate in Section 3.2. Note the objective does not fully correspond the objectives of our problem. Because not only do we want to minimise storage cost, but we also want to minimise changeover cost.

Some relevant literature uses the abbreviation SMSP; Single-stage Multi-product Scheduling Problem in a batch plant with parallel units (He & Hui, 2008) (Shi, Yan, & Wu, 2012) (He, Liang, Liu, & Hui, 2017). Other less relevant literature uses the abbreviation PMSP; Parallel Machine Scheduling Problem for an almost similar problem (single-stage parallel machine scheduling problem) (Gedik, Kalathia, Egilmez, & Kirac, 2018).

For a more detailed classification of the problem in hand, we use the classification method of (Méndez, Cerdá, Grossmann, Harjunkoski, & Fahl (2006). Their classification method is designed for the short-term scheduling of batch processes. The completeness of the classification, relevancy to the problem in hand make it applicable. Figure 3.1 shows the classification, in which aspects relevant to the problem in hand are underlined with a red marking. Relevant aspects are discussed in Table 3.1.

The goal of the problem in hand is to find:

- Assignment of batches to units (mixing and storage units); and
- Timing of these batches on units (mixing units);

to minimise IBC use and to minimise changeovers. The goal is relatively similar to that of similar problems except the objective (minimise IBC use) is different. This does mean that the classification method is less applicable. Note that for MU D the objective could be minimising total tardiness to try and produce Just-In-Time (JIT). This reduces the time a tank is full causing the tanks to be used more efficiently and in turn minimising IBC use (Belaid, T'kindt, & Esswein, 2012).

The problem in hand is at a crossroads of different types of scheduling problems because it is a single processing stage followed by a storage stage. No literature has been found on a single processing stage followed by a storage stage. The storage stage of the problem in hand is also at a crossroads because we have a limited amount of storage tanks (Finite Intermediate Storage, FIS). However, we assume that we have an unlimited number of IBCs available causing Unlimited Intermediate Storage (UIS).

He, Liang, Liu, & Hui (2017) state that the SMSP problem in a batch plant with unrelated parallel machines is NP-hard. Because of this, instances of large-size (e.g., >50 orders on 5 machines) are still challenging to solve optimally within 2 hours (He, Liang, Liu, & Hui, 2017). A worst-case scenario for the problem in hand is having to schedule approximately 90 orders across 4 machines. Note that the problem in hand also requires the assignment of storage, which is not the case in the problem of He, Liang, Liu, & Hui (2017).

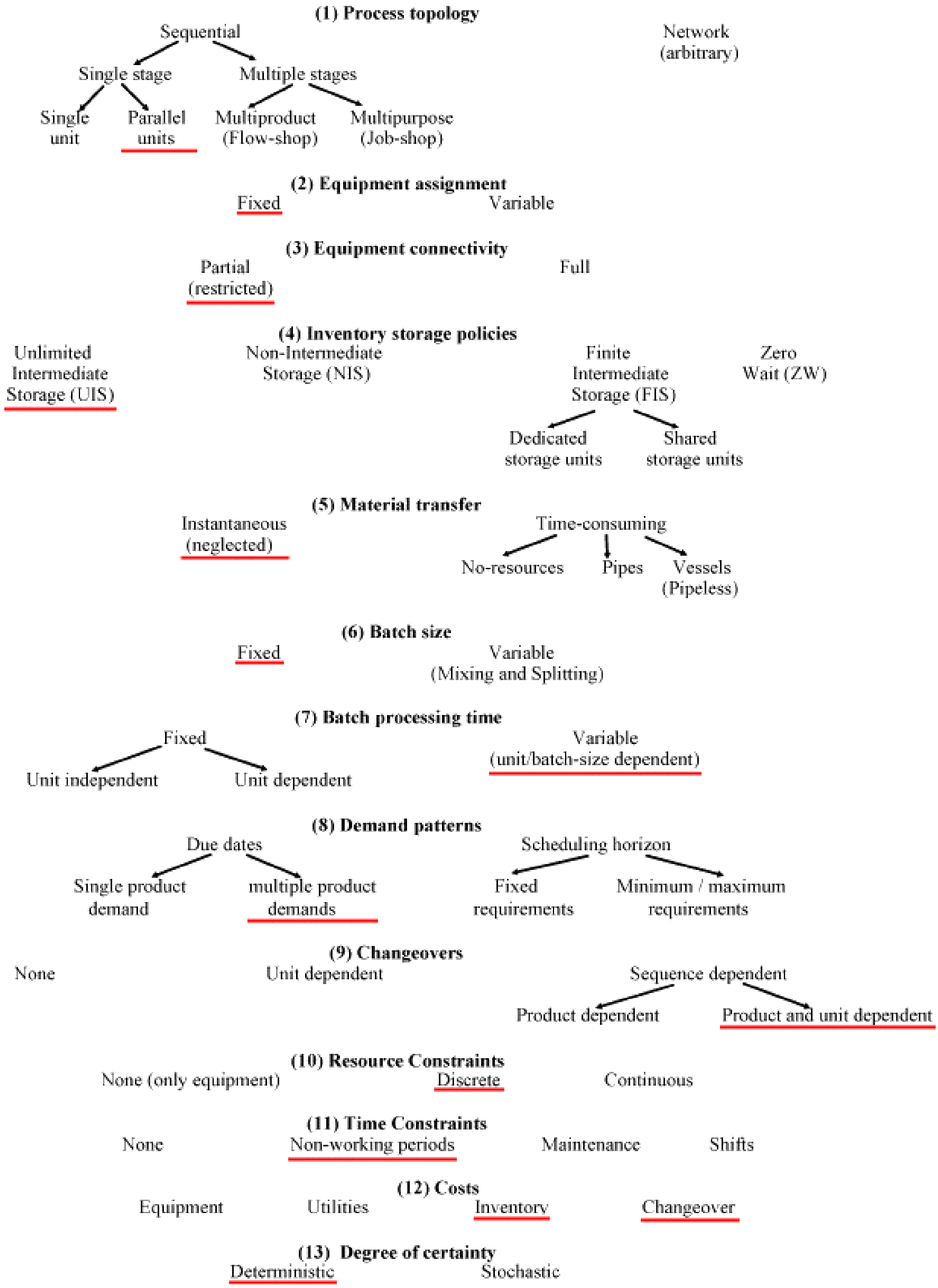

Figure 3.1 Problem classification (Méndez, Cerdá, Grossmann, Harjunkoski, & Fahl, 2006)

Table 3.1 Explanation classification choices

(1) Process topology *Single stage (parallel units)*	We limit our scope to the mixing plant, because of this reason we have a single (processing) stage with parallel units.
(2) Equipment assignment *Variable.*	Equipment assignment, i.e., assignment of orders to mixing units and storage units is variable.
(3) Equipment connectivity *Partial (restricted).*	Even though every mixing unit can connect to every storage unit, storage restrictions from filling lines cause a partial (restricted) connectivity.
(4) Inventory storage policies *Unlimited Intermediate Storage (UIS).*	The mixing plant has a finite number of shared tanks available, in theory, there are an unlimited number of IBCs available.
(5) Material transfers *Instantaneous.*	Material transfer is time-consuming and is done through pipes or hoses. However, we presume it is instantaneous because we have no data regarding time consumption. This is part of the stochasticity of the mixing time.
(6) Batch size *Fixed.*	Jobs are received from the filling planners. They can be split because they can consist of multiple filling orders. However, we assume splitting is not preferable. These combined filling orders can have due dates outside of the mixing schedule horizon; therefore, we do not split mixing orders. We only split mixing orders after mixing to benefit the storage scheduling.
(7) Batch processing time *Variable (batch-size dependent).*	The batch processing time is variable, depending on the batch-size.
(8) Demand patterns *Multiple product demands.*	There are multiple product demands based on due dates in the horizon of the mixing schedule.
(9) Changeovers *Product and unit dependent.*	Changeovers are product and unit dependent (rinsing).
(10) Resource constraints *Discrete.*	Aside from equipment there are labour constraints, sometimes there is no shift.
(11) Time constraints *Non-working periods.*	The goal of the mixing plant is to run 24/5. Weekends are non-working periods, and it can happen that there is no employee available to operate a mixer causing a non-working period for a MU.
(12) Costs *Changeover.*	We have 2 objectives that indirectly reduce cost; inventory and changeovers.
(13) Degree of certainty *Deterministic.*	The degree of certainty is deterministic because we use mixing times and due dates in a deterministic manner. In reality they are stochastic.

3.2 Solution approaches used in literature

In this section, we discuss solution approaches used in the literature for similar problems. We found no literature that discusses our exact problem. Solution approaches are classified into 3 categories: exact algorithms, heuristics, and metaheuristics. Solutions approaches per category are elaborated upon by Section 3.2.1, 0 and 3.2.3 respectively. Then, in Section 3.2.4, uncertainty is discussed.

Fuchigami & Rangel (2018) survey case studies in production scheduling. Case studies have been found from different industries, for example, textiles, chemicals, electronics and food industries. In their survey, they show that there is a large gap between theory and practice regarding scheduling, this is also the motivation for their survey. Figure 3.2 shows all case studies in their survey by year.

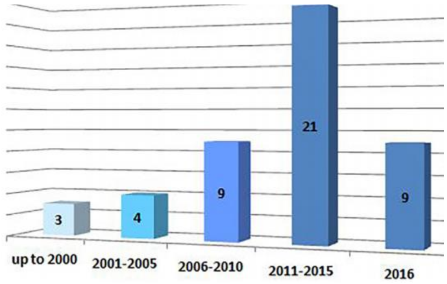

Figure 3.2 Case studies per year(s) (Fuchigami & Rangel, 2018)

There is clearly a growing interest in production scheduling that helps to narrow the gap between theory and practice. However, Fuchigami & Rangel (2018) found no case studies that address uncertainty that is so common in real life problems. Fuchigami & Rangel (2018) mention that case studies do not refer to computer systems present within companies, which seems to indicate incipient use. Increased use of computer systems increases the practical application of research findings. However, implementing computer systems can also be a limiting factor, depending on investment, training, and technical support. According to Fuchigami & Rangel (2018), there is a low representation of chemical industries finding only 3 cases. Rubén & José (2009) also note that literature on chemical engineering scheduling has been neglected.

The problem in hand has a single processing stage with storage. There are 2 popular strategies to improve the cost-effectiveness of a production system namely Just-In-Time (JIT) scheduling and Group Scheduling (GS) (Keshavarz, Savelsbergh, & Salmasi, 2015). JIT scheduling improves cost-effectiveness by reducing in-process inventories. Therefore, focussing on minimising earliness as an optimisation criterion should lead to better warehouse use, i.e., less IBC use (Yazdani, Aleti, Khalili, & Jolai, 2017). GS, which schedules jobs with similar characteristics, i.e., rinsing groups, together to reduce changeovers. The objective of the problem partly arises in the storage stage and is therefore important and must be addressed. No literature was found addressing a similar problem; 1 processing stage followed by a storage stage.

3.2.1 Exact algorithms

Exact algorithms solve problems to optimality. A Mixed Integer Linear Program (MILP) model is often used to represent a problem in the application of exact algorithms. In short, the representation is mathematically formulated by defining the:

- Objective function
- Variables
- Parameters
- Constraints

The problem can then be solved by enumerating over all possible solutions and finally giving the optimal solution as a result. However, enumeration can be computationally expensive in the case of a large solution space, for this reason enumeration is not often used. Software often used for solving MILP problems such as AIMMS make use of, among others, Branch and Bound (B&B) techniques (AIMMS, 2021). The B&B technique use of upper and lower bounds to evaluate solutions, reducing the search space by excluding solutions that cannot reach values above the lower bound.

Shim & Kim (2007) solve a single-stage scheduling problem of *n* independent jobs on *m* unrelated parallel machines with the objective of minimising total tardiness. They use a B&B algorithm combined with heuristics. Heuristics initialise the B&B algorithm by providing an initial upper bound. Scenarios with 2 and 4 machines and 20 jobs were solved in 27.5 and 248.3 seconds respectively. The CPU time requirements increased exponentially with the number of jobs. For example, scenarios with the same numbers of machines and 14 jobs were solved in 0.2 and 0.5 seconds respectively. Notably the percentage gap of the heuristics used for the initial upper bound for 4 machines and 20 orders is 3.59%. Meaning the initial solution made by heuristics was improved 3.59% using B&B techniques. Heuristics used are extensions of the Apparent Tardiness Cost (ATC) and Shortest Processing Time (SPT) dispatching rules which are discussed by Section 3.2.2.

Balasubramanian & Grossmann (2002) propose a B&B algorithm for scheduling flow shops with uncertain processing times. Their objective is to minimise the expected make span and they discuss multiple variations of flow shops. A Unlimited Intermediate Storage (UIS) and a zero-wait, i.e., no intermediate storage, flow shops were compared. Their results show that the computing times required for solving UIS problems are much longer than those for the zero-wait problems. A UIS scenario with 7 and 8 orders in 3 stages was solved in 336 and 7,800 seconds (time limit) respectively. Solving to 95% optimality instead of 100% reduced the CPU time by 80%. The problem solved by Balasubramanian & Grossmann (2002) differs from the problem in hand because of the number of stages and the storage scenario. Nevertheless, the reduced CPU time when solving to 95% optimality is interesting.

Yu & Karimi (2007) test multiple MILP models for scheduling a multistage, multiproduct batch plant with parallel units and no interstage storage. Solving for 7 jobs over 2 stages, both stages having 2 machines. Almost all models reach optimality within 1 minute. They note that slot-based models work best for unrelated parallel machines, but these models require more computation time. As the problem size increases, the computation time grows exponentially. They note that much work remains to be done to solve larger problems.

Mahnam, Moslehi, & Ghomi (2013) address a single machine scheduling problem with unequal release times and due dates, idle time insert, minimising the sum of earliness and tardiness. They note that the problem is shown in literature to be NP-hard in the strong sense, thus no polynomial time algorithm exists to solve the problem. The proposed B&B scheme can solve problems of up to 20 jobs.

More recently He, Liang, Liu, & Hui (2017) adapted the MILP of Liang & Hui (2016) to be more realistic, taking into account unit/order release times. They manage to solve an example to optimality with 50 orders and 4 machines in 3.62 seconds. However, in the example, constraints were used that excludes that some orders end the sequence, start the sequence, are assigned to certain machines and sequence constraints (order *x* cannot follow order *y*). In an example with 50 orders and 5 machines without these constraints, the CPU time reached the limit of 2 hours.

Gedik, Kalathia, Egilmez, & Kirac (2018) solve a PMSP with job sequence and machine dependent setup times. A novel Constraint Programming (CP) algorithm is proposed to solve the problem. They argue that their model outperforms all state-of-art algorithms in solving small instances and is also effective in finding good quality feasible solutions for larger problem instances. The proposed CP algorithm can optimally solve a scenario of 4 machines and 9 jobs in 1,639 seconds, scenarios with more jobs are not given. In the survey of Fuchigami & Rangel (2018), 1 case study was found where CP has been applied production scheduling. No case studies were found applying dynamic programming.

3.2.2 Heuristics

Heuristics are approaches to solve a problem without the guarantee that the solution is optimal. However, the solution can be a good approximation. Heuristic approaches can be divided into 3 groups: dispatching rules, divide-and-conquer based heuristics and tailored heuristics. In HFS scheduling, 50% of the case studies found by Rubén & José (2009) used heuristic approaches and of these, 13% used only dispatch rules. These heuristic approaches usually focus on problems with 2-3 stages.

Divide-and-conquer heuristics
Divide-and-conquer (a.k.a. decomposition) heuristics decompose a problem into simpler subproblems, then solve these in turn, and finally assemble the solutions to solve the decomposed problem. Decomposition techniques are widely used to tackle large-scale problems (Méndez, Cerdá, Grossmann, Harjunkoski, & Fahl, 2006). Decomposition can be used to create subproblems such as, assignment, sequencing and timing problems. The horizon can also be divided into sub-periods; this has been applied to mid-term scheduling problems (Wu & Ierapetritou, 2003). An often-appearing subproblem in more complex scheduling problems is the single machine scheduling problem (Keshavarz, Savelsbergh, & Salmasi, 2015).

Dispatching rules
Haupt (1989) classifies dispatching rules in the following categories:

- Rules based on processing time, e.g., Shortest Processing Time (SPT) and Largest Processing Time (LPT).
- Rules based on due date, e.g., Earliest Due Date (EDD).
- Combined rules, e.g., Apparent Tardiness Cost (**ATC**).
- Rules based neither on processing time, nor on due date e.g. First Come First Served (FCFS).

It is shown by Baker K.R. (1974) that the EDD rule minimizes total tardiness if there is at most one tardy job. Therefore, this simple rule is expected to work well if few jobs are likely to be tardy in the optimal solution. Thus, if due dates are loose, the EDD should be used (Potts & Wassenhove, 1991).

The **ATC** rule creates a priority index of available jobs, selecting the job with the highest priority. To do this, the rule schedules jobs one by one, i.e., each time the machine becomes available, the job with the highest priority is selected. The priority rule can be tailored to the problem in hand.

Dispatching rules are also known as ‘construction’ heuristics. They can be used to create initial solutions which can then be improved. He & Hui (2008) use dispatch rules to create initial solutions which are then used in their genetic algorithm (genetic algorithms see Section 3.2.3). They can also be used to generate upper bounds for B&B techniques (Shim & Kim, 2007). Solutions created with dispatching rules give no guarantee of solution quality or feasibility (Reklaitis, 1995).

Dispatch rules are well suited for complex problems in a dynamic and unpredictable environment. Therefore, they are popular in practical applications of HFS scheduling problems (Rubén & José, 2009).

Tailored heuristics
Tailored heuristics often make use of a certain property of the problem addressed. A tailored solution method often uses this essential knowledge to reduce the solution space or to search more efficiently in the solution space. Making use of a certain property of the treated problem is widely used to tackle large-scale problems (Méndez, Cerdá, Grossmann, Harjunkoski, & Fahl, 2006). It makes it possible to generate good solutions in a reasonable time. For example, Méndez, Henning, & Cerdá (2001) used pre-ordering constraints created by an EDD rule to reduce the number of sequencing variables.

Balasubramanian & Grossmann (2002) address the problem of scheduling batches in a flow shop with limited buffers. They propose a tailored heuristic that makes strong use of the structural properties of the problem. They show that their tailored heuristics performs best in terms of computational time and solution quality compared to 3 more general heuristics. Rubén & José (2009) note that the vast majority of non-exact approaches are tailored heuristics specific to the problem, mainly for problems with 2 and 3 stages.

Kudva, Elkamel, Penky, & Reklaitis (1994) address the problem of scheduling batch and semi-continuous plants with due dates, intermediate storage limitations and equipment changeover costs. They successfully tested their heuristic on data from an existing multiproduct plant. Results show the schedules generated by the heuristic are significantly better than those manually generated by operators. Furthermore, statistical analysis shows the heuristic solutions are always less than 8% below optimal indicating a powerful heuristic.

3.2.3 Meta-heuristics

Solutions provided by heuristics, e.g., dispatching rules, can be improved upon by meta-heuristics. Meta-heuristics can be interpreted as a special case of rescheduling, where the initial solution is rescheduled to improve a particular scheduling criterion (Méndez, Cerdá, Grossmann, Harjunkoski, & Fahl, 2006). Fuchigami & Rangel (2018) surveyed case studies in production scheduling. Of the 32 papers that used meta-heuristics, 15 were found to use Genetic Algorithms (GA) and 3 used Simulated Annealing (SA). Neural networks, ant colony, evolutionary algorithms, variable neighbourhood search and particle swam optimisation were all used in 2 papers. We expand upon GA and SA as they are most used in literature. Furthermore, we expand upon GRASP because that is what we use in our solution approach.

Genetic Algorithm

He & Hui (2008) applied a GA to a large-size SMSP problem in a batch plant with parallel units. The GA optimised the order sequence after which the orders were assigned to machines using dispatch rules. Multiple goals were tested among which tardiness. They compared the results of the GA to a MILP and a Random Search (RS) approach. The GA succeeded in solving a case of 50 orders for a 4-machine instance in 4.07 seconds with a tardiness cost of 0.35 (this is a meaningless figure but allows for comparison to other approaches). The MILP and RS approach managed in 2,278 seconds, a 324 tardiness cost and in 0.11 seconds a 148.91 tardiness cost respectively. The GA approach outperforms both the MILP and RS approach. For a larger scenario of 100 orders and 8 machines, the GA solves the problem in 7.03 seconds, however, the optimality is unknown. He & Hui (2006) apply a GA to a large size single-stage batch scheduling problem with parallel machines. The sequence was optimised using the GA, thereafter the orders were assigned to machines by dynamically applying dispatch rules.

Victor, Larisa, & Andrei (2009) apply a GA to a hybrid flow shop with unrelated machines, sequence-dependent setup time, availability constraints and limited buffers, trying to minimise make span. The algorithm was calibrated by “extensive experiments” (Victor, Larisa, & Andrei, 2009), applying the ANOVA technique with a 95% confidence interval. After calibration, experiments were conducted on real-life settings which shows the GA can produce high quality solutions. Ruiz & Concepción (2006) note that many authors separate sequence and assignment decisions in the HFS problem. After a sequence is determined with the GA, jobs are assigned to machines at every stage by priority rules. If a GA assigns both the machine and the sequence, crossover operations could result in an infeasible solution (Randall & & Kurz, 2007). Bean (1994) developed a new chromosome structure named Random Keys (RK). In the random keys structure, the integer part of a number is the order, and the decimal part is the sequence on the machine. By sorting by decimals, the sequence can be found. Shengchao, Jianhui, Ni, & Yan (2018) apply an RKGA procedure for scheduling unrelated parallel batch machines showing promising results. However, in the problem in hand we also need to assign storage units.

GAs are efficient in the exploration of different regions of the search space. However, finding local optima is rather slow (Wang, Löhl, Stobbe, & Engell, 2000). Moreover, a GA has genetic operators and parameters that must be tuned to the problem in hand (Murata, Ishibuchi, & Tanaka, 1996). Not tuning the parameters can lead to premature convergence. Tuning balances and enhances exploration and exploitation abilities of the algorithm (Wang, Zhang, & Zheng, 2006). Performance of GAs can be improved by combining it with problem-specific knowledge, especially for large-scale problems (Wang, Zhang, & Zheng, 2006). Therefore, genetic algorithms are often combined with a local search or add a sequential part following the GA (Wang, Zhang, & Zheng, 2006) (Sun, Zhang, Gao, & Wang, 2010). There does not exist a fixed set of parameters which enable a GA to optimise an arbitrary function (Hart & Belew, 1991).

Simulated Annealing
Janiak, Kozan, Lichtenstein, & Oguz (2007) apply SA to multiple HFS scenarios. SA optimises the order sequence and then tailored heuristics are used to create a schedule. Results of a two-stage HFS for a light, medium, and heavy loaded system show promising results. Parameters were developed experimentally for each system load. The solutions found by the SA algorithm are on average (average of all system loads) 5.86% away from the best-found optimal solution. The best-found optimal solution was often found by a Tabu Search algorithm or a hybrid of the two, however, the computation time of these algorithms however were much higher. The average computation time of the SA algorithm for 100 jobs is 5.06 seconds.

Allaoui & Artiba (2004) apply SA to a hybrid flow shop with maintenance constraints. SA optimises the order sequence, and the First Available Machine (FAM) rule is applied to assign orders to machines. An initial order sequence was constructed with heuristics, because a random solution may not give good performance. The application of SA showed better results than the application of other (non meta) heuristics.

Amine (2018) research multi-objective SA. They note that parameter tuning is a challenge in both single and multiple criteria optimisations. Many works required preliminary experiments for parameter tuning. Moreover, these experiments also require some knowledge of the actual efficiency frontier.

Greedy Randomised Adaptive Search Procedure
González-Neira & Montoya-Torres (2017) apply a GRASP algorithm for the hybrid flow shop scheduling problem. The proposed GRASP obtains satisfactory results in comparison with traditional dispatching rules and can be easily implemented which is preferred by practitioners. Resende & Ribeiro (2010) also underscore the simplicity of implementing a GRASP algorithm. Furthermore, contrary to other metaheuristics such as GAs, which use many parameters in their implementation, a basic GRASP only requires a single parameter.

Rajkumar, Asokan, Anilkumar, & Page (2011) propose a GRASP algorithm for flexible job-shop scheduling with limited resource constraints. The GRASP they propose is compared to the GA of (Du, Li, & Xiong, 2008). GRASP outperforms the GA in every scenario. Kontoghiorghes (2005) note that the GRASP algorithm appears to be competitive with respect to the quality of the produced solutions and efficiency compared to SA and GA. Furthermore, it is easier to implement, and tune compared to other meta-heuristics.

Resende & Ribeiro (2016) note that random construction can be slightly faster than semi-greedy construction. However, being slightly faster does not compensate for the poor quality of random constructed solutions compared to semi-greedy constructed solutions.

3.2.4 Uncertainty

In a recent survey of case studies, none were found to address uncertainty (Fuchigami & Rangel, 2018). There are non-case studies that address uncertainty, but only in a deterministic sense (Fuchigami & Rangel, 2018). Methods to deal with uncertainty are classified into 2 groups: preventive scheduling and reactive scheduling (Li & Ierapetritou, 2008). In preventive scheduling, a robust schedule and/or policies to accommodate changes due to uncertainty are generated before uncertainty occurs. In reactive scheduling, the original schedule is revised, or a new schedule is generated to accommodate changes.

Disruptions or new information can make the current schedule suboptimal or even infeasible, which motivates the need for rescheduling (Gupta & Maravelias, On the design of online production scheduling algorithms., 2019). However, Zhuge & Ierapetritou (2012) emphasise that disruptions and new information are not necessarily unfavourable. They also emphasise that rescheduling should be carried out to take advantage of favourable disruptions and new information rather than dismissing these events and maintaining the current schedule. Karimi & Reklaitis (1985) conducted a variability analysis (variability caused by disruptions) for storage units in a batch processing plant. They emphasise how even small variations can affect all scheduled storage operations.

Gupta, Maravelias, & Wassick (2016) show the importance of taking into account new information as soon as it becomes available. They emphasise that the traditional event-triggered view of rescheduling, i.e., reactive scheduling, has fundamental shortcomings. They therefore motivate to look at periodic rescheduling, which they refer to as 'online scheduling' (Gupta & Maravelias, On deterministic online scheduling: Major considerations, paradoxes and remedies., 2016). Online scheduling reschedules not only based on trigger events, but also periodically to take into account new information. Online scheduling is a special case of periodic rescheduling because the period is variable.

Online scheduling methods can be classified according to the framework as shown by Figure 3.3. Preventive scheduling is part of uncertainty modelling (robust optimisation, stochastic programming). To model uncertainty, a deterministic view is often used (Fuchigami & Rangel, 2018). The computation technology can consist of exact algorithms, heuristics or meta-heuristics. When the computation technology should be used (re-computation trigger) can be based on events, e.g., disruptions. Re-computation can also be periodic, e.g., a re-computation event every hour. Instead of periodic, an online calculated time step can be used which, for example, triggers re-computation based on the utilisation rate of machines. When the utilisation rate is high, i.e., when the machines are used intensively, re-computation can be triggered more often. Finally, the re-computation triggers can be combined in a hybrid method. The re-computation can be restricted based on the amount of nervousness allowed, e.g., in the problem in hand it is restricted by the hot room (Section 2.4.2). When the uncertainty modelling, the computation technology, the re-computation trigger and the allowed changes & constraints are defined, an 'online' scheduling method is defined and problems with uncertainties can be addressed.

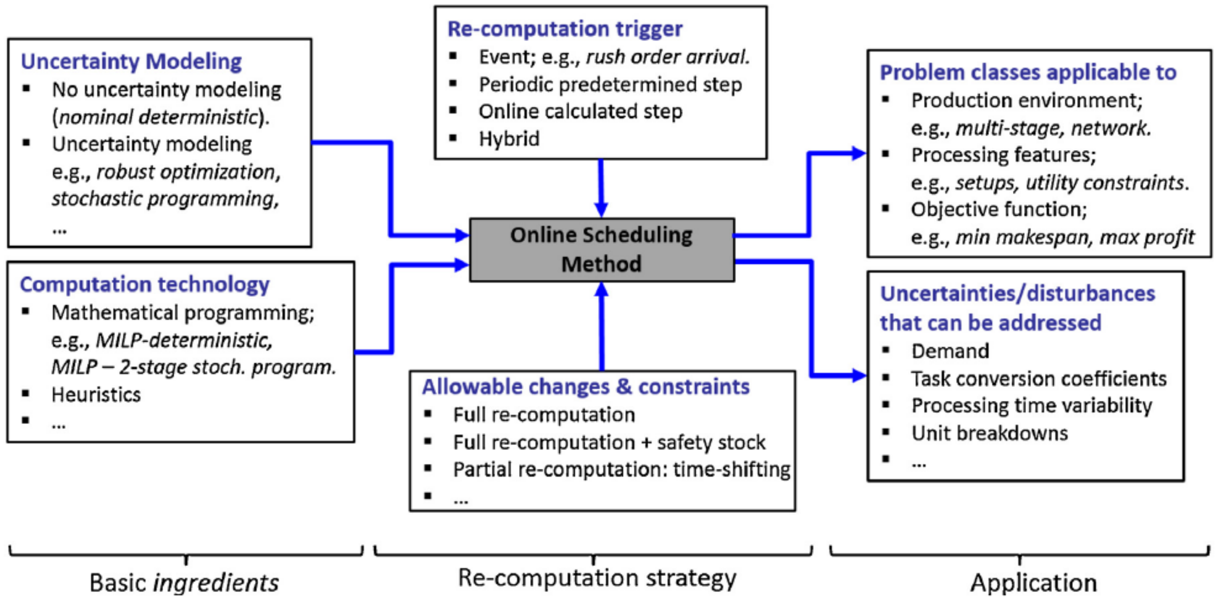

Figure 3.3 Online scheduling framework (Gupta, Maravelias, & Wassick, 2016)

Gupta, Maravelias, & Wassick (2016) note that online scheduling can be implemented using dispatch rules. They cite Sabuncuoglu & Karabuk (1999) who showed for a flexible machine shop environment that dispatch rules can be superior to optimum-seeking approaches when information becomes quickly outdated. Interestingly, scheduling approaches using dispatch rules are not popular in chemical production environments (Gupta, Maravelias, & Wassick, From rescheduling to online scheduling., 2016).

Gupta, Maravelias, & Wassick (2016) found that suboptimalities do not accumulate. For example, a suboptimal solution can be 90-95% optimal. Suboptimalities can be corrected through repetitive revisions required due to uncertainty.

3.3 Conclusion

To conclude this chapter, we summarise the main findings and conclusions in this section. In this chapter we have classified the problem in hand and discussed solution approaches in 3 categories: exact algorithms, heuristics, and meta-heuristics. Thereby answering the question: *What is known in literature about similar scheduling problems?*

No literature addressing the same problem was found, for 2 reasons. The combination of a single stage with storage is unique and the storage scenario is unique. The storage scenario is unique because theoretically we have infinite storage, but we prefer not to use part of the storage capacity: IBCs.

We assume the problem in hand to be NP-hard because SMSP problems are already NP-hard and the problem in hand extends beyond an SMSP problem because of storage. Exact approaches such as B&B, can solve small instances, i.e., small number of jobs and machines, of NP-hard problems in reasonable time to an optimum. However, the computation time of exact approaches increases exponentially as the size of the instance increases. Not solving to optimality can significantly reduce the computation time, but this is unlikely to be sufficient for the problem in hand. Therefore, we conclude that exact algorithms are not applicable to the problem in hand.

Heuristics are often applied to large-scale NP-hard problems where exact algorithms are not applicable due to computation time. A common approach to large-scale problems is divide and conquer, where the problem is divided into simpler subproblems. Kudva, Elkamel, Penky, & Reklaitis (1994) do so for a most similar scheduling problem: scheduling a batch plant with due dates, intermediate storage limitations and changeover costs. They show that their approach yields better schedules than manually created ones, indicating a promising solution approach. Dispatch rules are often used to construct initial solutions. Several categories of dispatch rules have been proposed that are applicable to different objectives, e.g., make span or lateness objectives. Solutions can also be constructed using tailored heuristics. These are often applied to problems with characteristics in which the tailored heuristics can excel.

Meta-heuristics can improve initial solutions. Meta-heuristics that are often applied to most similar problems are GAs and SA. GAs and SA have shown to achieve good results for difficult problems. However, these methods also rely on dispatch rules after dividing the problem. Moreover, these methods require parameter tuning, which may require extensive experimentation and knowledge of the efficiency frontier. GRASP, a single-parameter meta-heuristic allows us to guide the algorithm more easily to a good solution. Moreover, the GRASP algorithm seems to be able to compete with GA and SA. The efficiency frontier of the problem in hand is unknown because the problem in hand has not been solved before.

4 Solution approach

This chapter focuses on research question Q3: How can we provide an improved scheduling strategy for the mixing plant? To answer this question, Section 4.1 first introduces the problem. Thereafter, Section 4.2 introduces the solution approach and explains overarching aspects. Subsequently, Section 4.3 explains phase 1 and Section 4.4 phase 2 of the solution approach. Finally, Section 4.5 summarises and concludes this chapter.

4.1 The problem

In this section we explain the problem. First, Section 4.1.1 outlines the context of the problem within Eurol's production plant. Then, Section 4.1.2 defines the scheduling problem.

4.1.1 Problem context

The mixing plant is part of the production process at Eurol. Figure 4.1 shows part of the production process at Eurol, showing the mixing plant and its in- and output. The inputs for the mixing plant are raw materials stored in raw material storage tanks (RT) or in IBCs. They are pulled to mixing units (MU) in the mixing plant that mix several raw materials into a homogeneous blend. From here on, we refer to the production of a blend as a job. After mixing, the blend is stored in a tank(s) (T) and/or IBC(s). The output of the mixing plant are blends in storage, which are used by filling lines (L) in a pull fashion. Multiple filling lines can require a single blend therefore, a single job can be related to multiple filling jobs.

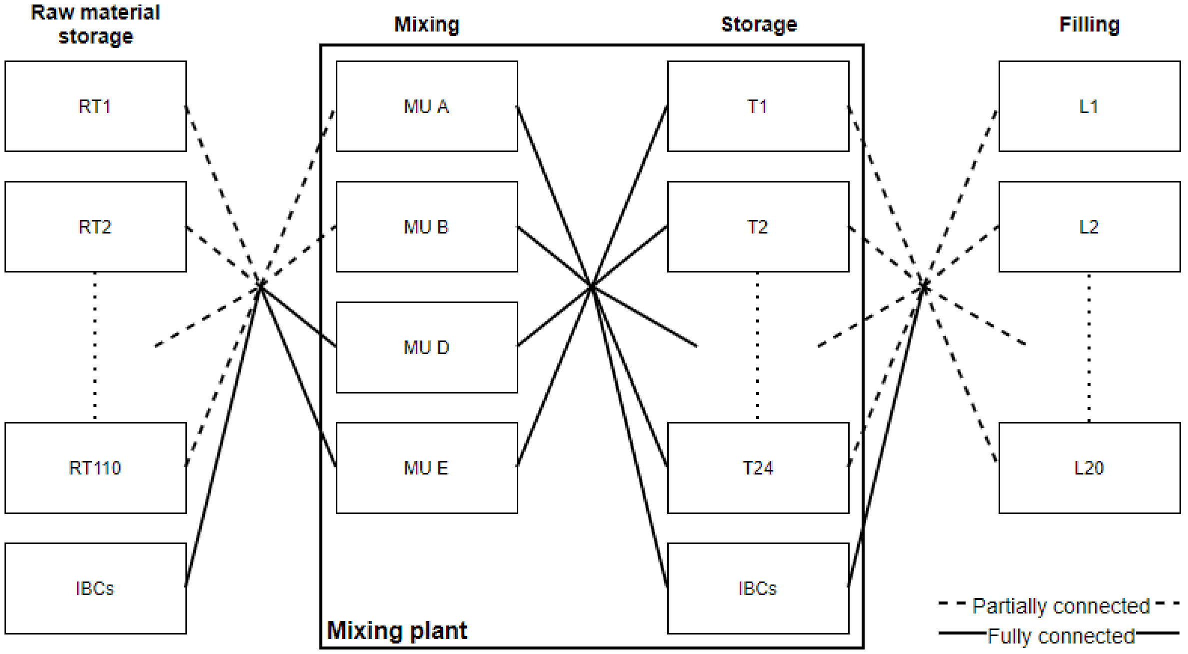

Figure 4.1 Partial schematic overview production plant of Eurol (RT[x] = Raw material Tank [x], T[x] = Tank [x], L[x] = filling line [x])

4.1.2 The scheduling problem

The scheduling problem for the mixing plant can be stated as follows: define a schedule for the mixing plant indicating when and on which mixing unit a blend should be produced (job) and in which storage unit(s) the produced blend is to be stored. Filling lines use the produced blend to produce (filling job) the final product by filling containers.

Table 4.1 shows an example of a schedule for the mixing plant. The example is limited to 1 mixing unit with 2 jobs. The storage units are either tanks or IBCs. If IBCs are to be used the number of litres to be filled in IBCs should be mentioned.

Table 4.1 Example schedule representation

PO ##	Fluid	Rinsing group	kg	Processing time	Start time	End time	Due date	Storage unit(s)
PO1	1	PCMO	2,050	01:40	29-4-2021 22:45	30-4-2021 00:25	3-5-2021 06:30	31, IBC(450L)
RINSING				00:18	30-4-2021 00:25	30-4-2021 00:43		
PO2	2	HDD	8,000	03:30	30-4-2021 00:43	30-4-2021 04:13	3-5-2021 12:30	30, 33

The following parameters apply:

- The scheduling horizon *H* in minutes.
- Mixing unit data: minimum and maximum capacities in kg, input restrictions (manifold), changeover time, and working periods in minutes.
- Tank data: minimum and maximum capacities in litres, which blend it currently contains (and when it will be empty) or contained previously, and the changeover time when changing rinsing group.
- Filling line data: input constraints (tank) (Table 2.2) and guidelines for IBC use (Table 2.1).
- Job data: size in kg, rinsing group, fluid number, density, mixing time in minutes for each mixing unit, release dates, and the earliest due date of all filling jobs which are related to the job.
- Filling job data: to which job it is related, the start time and the filling time in minutes.
- The minimum required buffer time, *b.* The buffer time is the time between the due date of the job and the time the blend is finished. We produce earlier than required as a measure of dealing with variation in the mixing plant, we go into more detail in Section 4.2.1.
- Minimum time between blends assigned to a tank, *bt* (see also Section 4.2.1).
- The maximum time in days a single filling job is allowed to increase the storage time of a blend, *dd*.

The objective is to develop a schedule to produce blends on time and at minimum cost. Costs consist of IBC use and changeover costs. The cost parameters are unknown. The objectives, therefore, are to minimise IBC use and changeovers without tardy jobs. Although the cost parameters are unknown, Eurol prefers minimising IBC use rather than minimising changeovers. Note that the objectives may be conflicting. For example, minimising changeovers may mean using more IBCs.

The following restrictions/constraints apply for scheduling the mixing plant:

1. Tardy jobs are not permitted.
2. The production time required by a mixing unit cannot exceed the scheduling horizon.
3. Pre-emption is not permitted.
4. A job cannot be started before its release date.
5. All jobs must be fully assigned to storage unit(s).
6. Different fluids cannot be stored in the same storage unit at the same time.
7. MU E can only be used for jobs that no other mixing unit can produce.
8. Splitting is only allowed if the job cannot be mixed otherwise.

Based on our analysis of the capacity utilisation of the mixing units (Table 2.6), we show that MU E is preferably not used (7) because it had a utilisation rate of 4% during most of 2020. Therefore, we only use MU E for jobs that can only be produced on MU E. This does not have a big effect on the schedule because MU B can handle all small jobs (MU E and B can only produce small jobs), these jobs are also not stored in tanks due to their size.

Splitting jobs (8) must be avoided because the mixing time approximately doubles. Raw materials must be handled twice. Handling of raw materials twice is costly, not only because of the manpower required, but also because it means the manifold is needed twice and raw materials must remain in the hot room longer. Therefore, we assume that splitting indirectly increase cost. Also, we do not split based on filling jobs, as stated in Section 1.4.1: filling jobs are combined by filling planners to the best of their ability to ensure an efficient combination of filling jobs into jobs. However, sometimes splitting can be required if MU D is full. Avoiding splitting reduces the solution space of the scheduling problem, excluding solutions that are non-optimal.

Furthermore, the following assumptions apply; we elaborate on assumptions 5 and 6 below the summation:

1. Minimising the use of IBCs is a more pressing cost issue than minimising changeovers.
2. The mixing units can remain idle at no cost.
3. Jobs can be paused.
4. Changeover times are negligible compared to the total production time of mixing units.
5. We assume an EDD sequence ensures no tardy jobs.
6. The filling plant always empties the tank with the least number of litres first.

We can assume that an EDD sequence ensures no tardy jobs; if there are no release dates and setup times an EDD sequence ensures no tardy jobs, if there are tardy jobs in this EDD sequence, the schedule is infeasible (Pathumnakul & Egbelu, 2003). Sorting on EDD might cause violation of release dates. Note that we apply a timing procedure before we check for violations, the timing procedure tries to time jobs as close to their due date as possible. We elaborate upon our timing procedure in Section 4.3.2. Because there are few jobs with release dates, we assume that, in case of conflict, we have deviate very little from the EDD sequence to not violate release dates. Furthermore, the EDD sequence is optimal when setup times are negligible (Baker K. R., 1999). We assume setup times are negligible because rinsing time accounts for only 6.4% of the processing time see Table 2.4, Figure 2.10. Note that changeover times are not the main reason for minimising changeovers.

If a blend is assigned to a tank, we fill the tank as much as possible. A job can consist of several filling jobs. If we fill 1 tank with 1 job that consists of 2 filling jobs, we do not try to split to fill 2 tanks: 1 for each filling job. This would constrain an extra tank. Similarly, we do not split if we fill 2 tanks with 1 job that consists of 2 filling jobs. We just fill the first selected tank to its maximum capacity and fill the rest in the other tank. This is because the filling plant always empties the tank with least amount of oil first (6). As a result, the tank is available sooner and only 1 tank is to be drained. Draining is necessary when a filling line finished a job with a tank, for example, 100 kg of fluid may remain that has ensured the filling job could be completed (overcoming spillage, etc.).

4.2 Solution approach introduction: decomposition, algorithm, and uncertainty

We propose a heuristic-based algorithm. Our proposal is an adaption of the heuristic proposed by Kudva, Elkamel, Penky, & Reklaitis (1994). Equal to the approach of Kudva, Elkamel, Penky, & Reklaitis (1994) we decompose our problem into the following subproblems:

1. Assign jobs to mixing units.
2. Schedule mixing units (i.e., sequencing and timing of jobs).
3. Assign blends to storage units.

In the first subproblem we assign jobs to mixing units. In subproblem 2 we schedule mixing units with the objective to minimise the number of changeovers and the earliness, this is also the objective of subproblem 1. Finally, in subproblem 3, we assign jobs to storage units with the objective to minimise IBC use and the number of tank changeovers.

Kudva, Elkamel, Penky, & Reklaitis (1994) solve the scheduling problem by recursively scheduling jobs. We differ from this approach; we propose to fully solve each subproblem before moving to the next subproblem. We do so mainly because we always have IBCs available if tanks are full. To efficiently utilise tanks and reduce computation time we first assign the largest jobs. This requires the completion of the assignment of jobs to and scheduling of MU D (the largest mixing unit). The proposed heuristic uses a two-phase heuristic approach. In the first phase an initial solution is generated aiming to generate a schedule without tardy jobs. In the second phase, the initial solution is optimised to minimise cost without causing tardy jobs. We now briefly explain the phases, and the solving strategy. A detailed explanation of the phases and the heuristics used in them is provided by Section 4.3 and Section 4.4, which explain phases 1 and 2 respectively.

Phase 1 is considered the construction phase. A solution to each subproblem is constructed in phase 1 consecutively. Subproblem 1 is solved by applying an adaptation to the rules proposed by Kudva, Elkamel, Penky, & Reklaitis (1994). For subproblem 2 we rely on the EDD rule to construct mixing unit schedules without tardy jobs. Finally, subproblem 3 is solved by a tailored heuristic.

In phase 1, we generate an initial solution for the scheduling of the mixing units. In phase 2, we optimise the initial solution of phase 1. We refer to phase 2 as the optimisation phase. Figure 4.4 shows how the different subproblems relate to each other during the optimisation. There is no search for alternatives to subproblem 1 because this greatly increases computation time. Subproblem 2 is optimised using GS and JIT scheduling strategies resulting in multiple possible schedules. This is shown by Figure 4.4 as the table titled ‘Combine solutions’ in subproblem 2. GS is used to minimise changeovers. JIT is used to minimise earliness. Minimising earliness should minimise IBC use since there should be less inventory. For each combination of mixing unit schedules, subproblem 3 is solved at least once. Then, for each schedule of subproblem 2, we have a complete mixing plant schedule. Subproblem 3 is then solved multiple times, applying a GRASP strategy to the most promising mixing plant schedule solutions. The table in subproblem 3 as shown by Figure 4.4 in green represents an example of the GRASP solutions for one of the most promising mixing plant schedule solutions.

Uncertainties must be dealt with in the solution approach. The way uncertainties are dealt with can have a great influence on the design of the solution approach. Therefore, we now explain our proposal for dealing with uncertainties in Section 4.2.1.

4.2.1 Uncertainty

There are various causes of uncertainty within the mixing plant. These uncertainties can have a significant influence on the optimality of a schedule and can even cause tardy jobs. Therefore, these uncertainties must be addressed in a solution approach to solve the scheduling problem. We explain our approach for dealing with uncertainties using Figure 4.2 which depicts the online scheduling framework from Gupta, Maravelias, & Wassick (2016). We discuss all the topics represented by the figure in blue to define our method for dealing with uncertainties.

UNIVERSITY OF TWENTE.

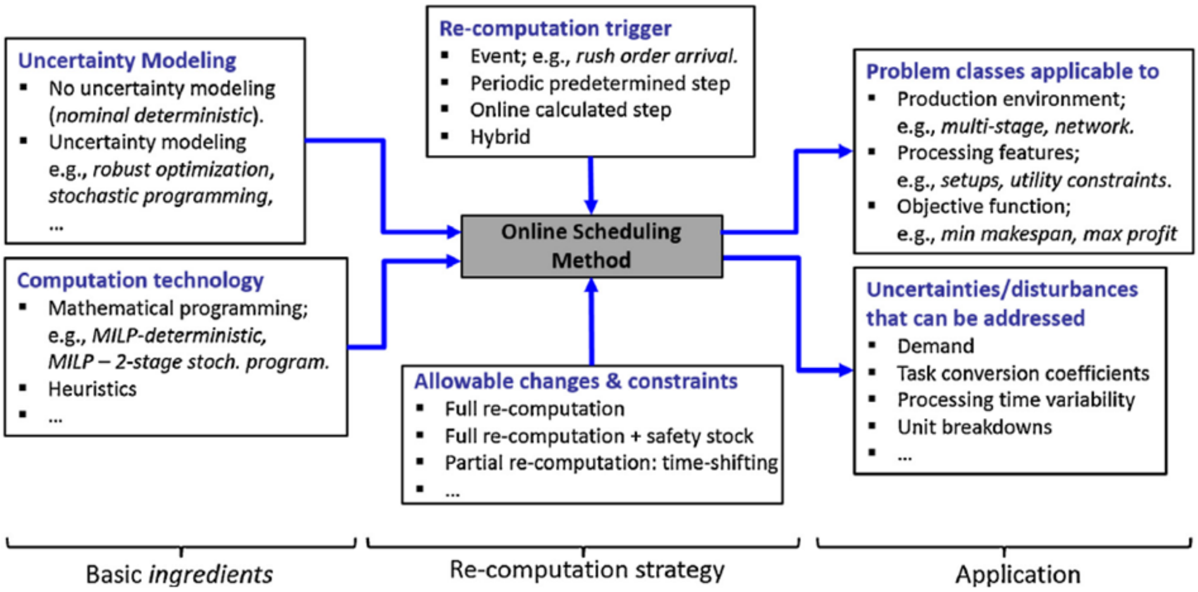

Figure 4.2 Online scheduling framework (Gupta, Maravelias, & Wassick, 2016)

Uncertainty modelling

We propose to apply deterministic uncertainty modelling, which is also applied in literature. Fuchigami & Rangel (2018) found no case-studies that address uncertainty in a non-deterministic manner. To account for uncertainty in, e.g., processing times and due dates, we apply a robustness measure: producing earlier than required. We call this the buffer time (*b*). Figure 4.3 shows all components of the storage time of a blend. Furthermore, to ensure robustness when assigning blends to storage we ensure there is a minimum time between blends assigned to a tank (*bt*), as shown by Figure 4.3. We elaborate upon our robustness measure and the preferred settings in Section 5.2.

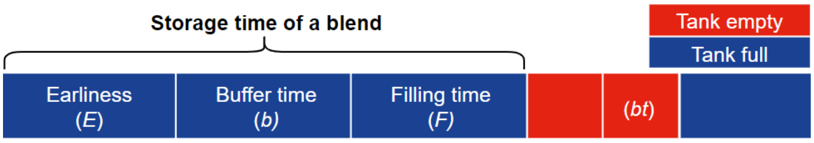

Figure 4.3 Components of storage time

Computation technology

To find solutions quickly, we propose the use of heuristics. A solution could also be provided quickly by applying scenario scheduling, where solutions are created for scenarios that may occur due to uncertainties. However, this can become computationally challenging (Gupta, Maravelias, & Wassick, From rescheduling to online scheduling., 2016) and a solution for the scenario in practice may not be available. Ensuring that the scenario that occurs is generated can quickly become difficult, especially with the advent of rush orders.

Re-computation trigger

We propose a hybrid re-computation trigger, re-computing based on events and periodically. Events are usually “negative”, such as delays or a rush order coming in. Periodic rescheduling is advised to account for new information that may be positive (Gupta & Maravelias, On deterministic online scheduling: Major considerations, paradoxes and remedies., 2016). Instead of periodic re-computation, an online calculated step can be applied, e.g., re-computing more often when units are heavily used can be applied. The detection of events and scheduling periodically is up to the planner. In case of events, the planner is notified and if new information comes in, the schedule can be re-computed (periodically). An online calculated step is therefore not applicable.

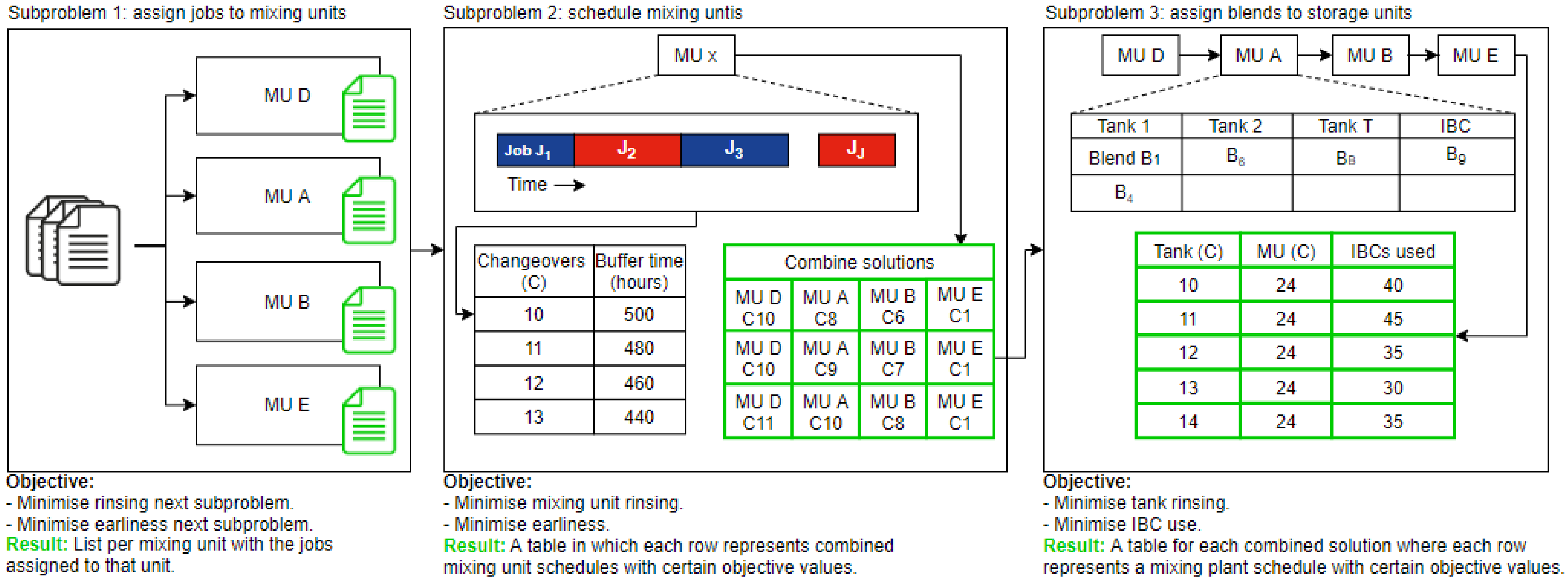

Figure 4.4 Overview results and objectives per subproblem

Allowable changes & constraints

We propose a full re-computation of the schedule. The disadvantages of a full re-computation are the computational burden, the loss of manual optimisation steps and nervousness in the schedule. Since we apply heuristics, the computational burden is small. Manual optimisation, i.e., manual changes in the mixing plant schedule might no longer be applicable. If they are, they can be copied from the old schedule. We assume that nervousness is allowed because the execution of jobs in the current situation may deviate significantly from the schedule (Table 2.11). The advantages of full re-computation are that there is no need to choose what to keep from the old schedule. Moreover, full re-computation might be preferred because small variations can affect all scheduled storage operations (Karimi & Reklaitis, 1985).

Problem classes applicable to

The proposed method is specific the problem in hand.

Uncertainties/disturbances that can be addressed

All uncertainties/disturbances in the problem in hand can be addressed by the described approach:

- Filling line ahead/behind schedule (causing uncertain due dates)
- Processing time variability causing mixing units to be ahead or behind on schedule.
- Rush order arrival
- Release date variability advanced/delayed/assigned

4.3 Phase 1: constructing an initial solution

In this section, we explain our approach to phase 1 of the heuristic: constructing an initial solution. To explain the solution approach for each subproblem, we follow the same structure. First, we explain the general solution approach outline. Then we explain each step of the solution approach to the subproblem in more detail. We explain the solution approach for subproblem 1 and 2 with help of a logic flowchart and an example.

4.3.1 Subproblem 1: assign jobs to mixing units

In subproblem 1 we assign jobs to mixing units. The aim is to assign jobs to mixing units in such a way that rinsing and earliness can be minimised in the following subproblem. If there is only one unit that can produce the blend, the decision is trivial. When there are several units that can produce the blend, the problem becomes less trivial. A decision is made based on information and intuition about the specific problem, using a suitable unit choice algorithm Kudva, Elkamel, Penky, & Reklaitis (1994).

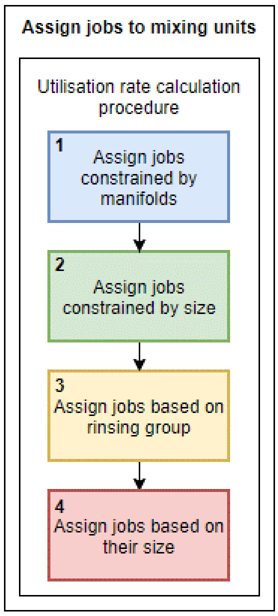

Figure 4.5 Solution approach overview subproblem 1

<u>General solution approach outline</u>

Figure 4.5 shows the steps we propose to solve the subproblem. The solution approach is a tailored heuristic based on general rules. We have divided the heuristic into 4 steps that are executed consecutively. The utilisation rate calculation procedure is carried out after each assigned job and is thus an overarching procedure. The utilisation rate is calculated to ensure we do not assign too many jobs to a mixing unit. If too many jobs are assigned to a mixing unit, it cannot finish within the scheduling horizon. The 4 steps used to solve this subproblem, use an adaptation of the rules proposed by Kudva, Elkamel, Penky, & Reklaitis (1994) to assign jobs. These rules are:

1. Exclude mixing units that cannot produce the job.
2. Prefer the mixing unit with the total batch size closest to the amount needed to be processed.

We first assign jobs that are constrained by manifolds to mixing units. Then we assign jobs constrained by size. Next, we assign jobs based on rinsing groups, this is where rules proposed by Kudva, Elkamel, Penky, & Reklaitis (1994) do not apply. Finally, the remaining jobs are assigned based on their size, analogous to rule 2.

We now explain the solution approach in more detail. Figure 4.6 shows the logic flowchart of the heuristic to solve this subproblem. The steps of the heuristic can be identified using the colour scheme as shown by Figure 4.5. In the figure we follow an example path through the heuristic, highlighted in blue. Actions that assign jobs in our example path are also given a number in the top left corner of the action, e.g., **A1** (action 1). The example path is referred to throughout the explanation. The result of the example path is shown by Table 4.2, where the outer right column indicates the action that assigned the task. We first explain the utilisation rate calculation procedure, after which we explain the other steps consecutively following the same order as the heuristic.

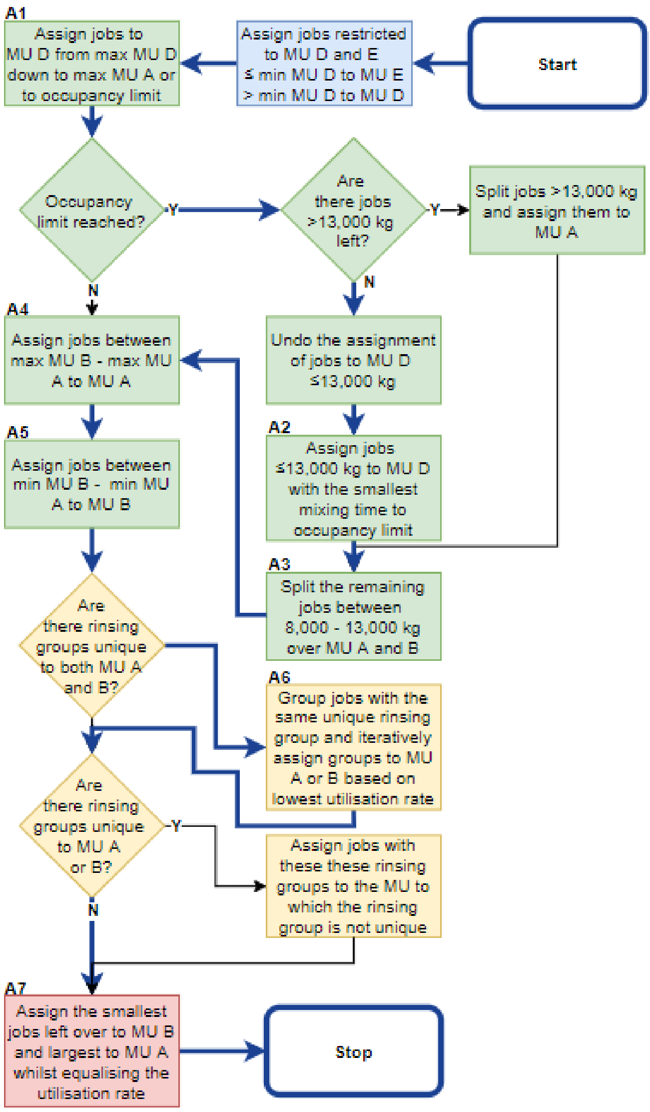

Figure 4.6 Logic flowchart: assign jobs to mixing units, max (maximum) and min (minimum) refer to the capacity of the mixing unit

Table 4.2 Example path subproblem 1

Rinsing group	Size (kg)	Mixing time H:MM	Due date	MU	A
HDD	18,000	3:00	16-02 16:45	D	1
- Multiple orders assigned to D (S1) -					
GEAR OIL	14,000	5:00	16-02 15:00	D	1
PCMO	9,000	1:30	16-02 23:30	D	2
PCMO	12,000	2:00	19-02 19:00	D	2
PCMO	8,000	3:30	18-02 01:00	A	3
PCMO	4,000	3:45	18-02 01:00	B	3
HYDRO	8,000	2:30	16-02 21:00	A	4
- Multiple orders assigned to A (S4) -					
HDD	5,000	1:30	19-02 11:00	A	4
PCMO	1,000	2:00	17-02 08:00	B	5
- Multiple orders assigned to B (S5) -					
HDD	1,700	1:30	16-02 21:30	B	5
GEAR OIL	4,000	2:30	17-02 13:00	B	6
HDD	4,000	1:30	19-02 18:30	A	7
HDD	3,000	1:30	19-02 17:00	B	7

Utilisation rate calculation procedure
We propose the following formula to calculate the utilisation rate:

$$[1] \qquad utilisation\ rate = \frac{sum\ of\ mixing\ times\ MU + required\ changeovers * 18}{scheduling\ horizon\ - non\ working\ periods} * 100\%$$

The numerator is the time the mixing unit needs to produce the assigned jobs. In the denominator we have the available time. The numerator consists of the mixing time plus the required number of changeovers, multiplied by the required time per changeover, which is 18 minutes. For the required number of changeovers, we propose a worst-case scenario: the number of changeovers required when sorting assigned jobs on EDD. This is the worst-case because from there on we only reduce the number of changeovers required. In phase 1 subproblem 2, we generate an initial solution to schedule mixing units, applying EDD sorting. In phase 2 subproblem 2, we optimise this initial solution. Therefore, this is the worst-case scenario. There are few jobs with release dates (Section 4.1.2). The EDD sequence may be infeasible due to the violation of release dates, even after applying the timing procedure which we explain later. Therefore, we may have to deviate slightly (because there are few jobs with release dates) from the EDD sequence. Deviating from the EDD sequence may lead to more changeovers. However, since there are few jobs with release dates and we aim to minimise changeovers in subproblem 2, we assume the utilisation is not exceeded due to jobs with release dates.

Step 1: assign jobs constrained by manifolds
There are few jobs with manifold restrictions, and we prefer not to use MU E. Manifold restrictions can constrain jobs to MU D and E. This is because the manifold that MU D and E connect to has access to all raw material tanks, whereas the manifold of MU A and B does not (Section 2.1.2). MU D has a minimum capacity of 4,000 kg. Therefore, we must assign jobs smaller than 4,000 kg constrained by manifolds to MU E.

Step 2: assign jobs constrained by size
We assume that splitting is to be avoided as much as possible and we must adhere to minimum and maximum capacities of mixing units. Figure 4.7 shows the jobs we have restricted based on these assumptions in blue. We assign jobs from 24,000 (maximum capacity MU D) down to 8,000 kg (maximum capacity MU A) to MU D **(A1)**. Thereby approximately 30% of all jobs are assigned to MU D, outlined in red by Figure 4.7. Large jobs need on average a longer mixing time because of the pumping time needed in and out of the mixing unit. Therefore, jobs of less than 8,000 kg are not assigned to MU D because the utilisation rate of MU D is probably already the highest if the remaining jobs are evenly spread among the other mixing units based on the mixing time. This is already likely and becomes even more likely in the future as Eurol is moves to producing larger batches (Section 1.2). Moreover, assigning jobs less than 8,000 kg to MU A instead of MU D is analogous to Rule 2 as proposed by Kudva, Elkamel, Penky, & Reklaitis (1994). Rule 2 gives preference to the processing unit whose maximum capacity is closest to the quantity to be processed.

In addition, splitting is sometimes necessary because MU D would exceed its utilisation limit if all jobs between 8,000 to 24,000 kg are assigned to MU D (Section 2.2.1). When the utilisation limit of MU D is reached, a check is made to see if there are any remaining jobs left larger than 13,000 kg. Remaining jobs larger than 13,000 kg must be split to MU A because the limits of MU A and B are 8,000 and 5,000 kg respectively. If they were not split to MU A, we would have to split the job even further, which is not preferable.

If there are only jobs left smaller than 13,000 kg, we undo the assignment of jobs to MU D below 13,000 kg. Since we prefer not to split jobs, we assign jobs between 8,000 and 13,000 kg with the smallest mixing time to MU D **(A2)**. We assume that in this way most of the jobs between 8,000 and 13,000 kg are assigned to MU D, resulting in fewer split jobs. The jobs between 8,000 to 13,000 kg that need to be split are assigned to MU A and B **(A3)**. We split over MU A and B because this does not constrain the earliness to increase by the mixing time of the job. We assign 8,000 kg to MU A and the rest to MU B. We assign the largest part to MU A because we prefer large jobs as this can reduce the amount of rinsing required (Section 2.4.1). We now assign jobs of 5,000 (maximum capacity MU B) to 8,000 kg to MU A **(A4)**, as shown in yellow by Figure 4.7. We assign jobs between 850 (minimum capacity MU B) and 1,700 kg (minimum capacity MU A) to MU B **(A5)**, as shown in green by Figure 4.7. We have now assigned all jobs shown in blue in Figure 4.7.

<u>Step 3: assign jobs based on rinsing group</u>

Jobs that now remain are between 1,700 and 5,000 kg, shown in grey by Figure 4.7. These can be assigned to both MU A and B at no direct cost. All jobs (in grey) with the same unique rinsing group to both MU A and B are grouped together and assigned to the mixing unit with the lowest utilisation rate at that moment **(A6)**. If a job (in grey) is unique to, e.g., MU A, we assign it to MU B and vice versa. This should reduce the number of changeovers required.

<u>Step 4: assign jobs based on their size</u>

Finally, the jobs (in grey) remaining after step 3 are assigned to MU A and B according to their size, aiming for equal utilisation rates. The largest jobs are assigned to MU A and the smallest jobs to MU B, analogous to rule 2 of Kudva, Elkamel, Penky, & Reklaitis (1994).

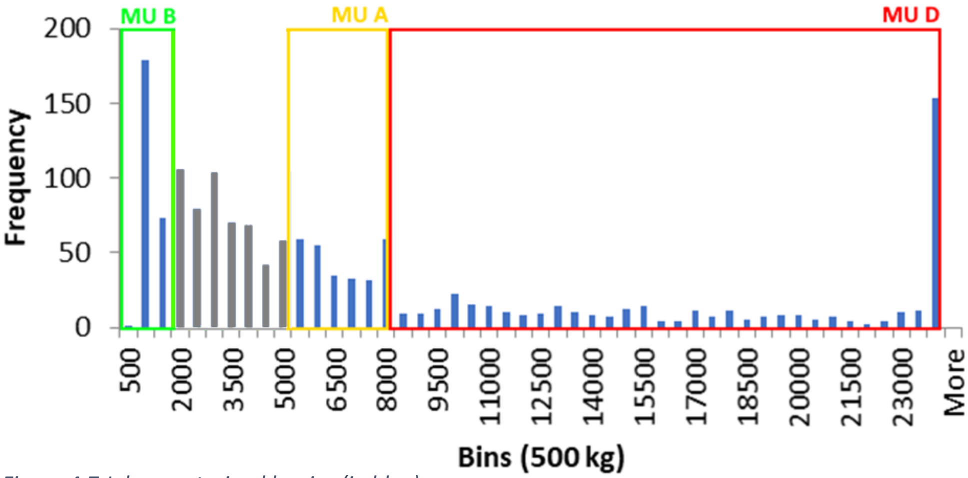

Figure 4.7 Jobs constrained by size (in blue)

4.3.2 Subproblem 2: schedule mixing units

In the previous subproblem we assigned jobs to mixing units. In subproblem 2 we schedule mixing units. The objective in this phase is to create an initial solution where there are no tardy jobs. In phase 2, we focus on the minimising changeovers and earliness. We deviate from the solution approach as proposed by Kudva, Elkamel, Penky, & Reklaitis (1994) because they allow jobs to be discarded. Therefore, their approach is assumed inefficient for the problem in hand because it can result in many infeasible schedules.

<u>General solution approach outline</u>

Figure 4.8 shows the steps which we propose to solve the subproblem. The solution approach is a tailored heuristic that uses dispatch rules. We have divided the heuristic into 2 steps which are executed consecutively. We assume that there are few jobs with release dates and that changeover times are negligible. Therefore, sorting on EDD should generate a schedule without tardy jobs, as explained in Section 4.1.2. If this schedule is not feasible (without considering release dates) the scheduling problem cannot be solved. After sorting on EDD we move jobs with release dates as close to their due dates as possible. Finally, we assign start times to all jobs. We now explain the solution approach in more detail.

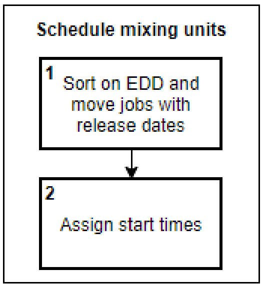

Figure 4.8 Solution approach overview subproblem 2

<u>Step 1: sort on EDD and move jobs with release dates</u>

We sort the job list (result from subproblem 1) by EDD. Then we move all jobs with release dates as close as possible to their due date. As close as possible means we postpone these jobs to the first opportunity where they can be produced without causing mixing unit idle time. This is behind the job being processed, where the earliness of the job with a release date is minimal. Figure 4.9 illustrates how we move jobs with a release date. In Figure 4.9 J1 is the job with a release date. If J1 is scheduled before J3 and behind J2, idle time would occur between J1 and J3, potentially causing an infeasible schedule. Therefore, the algorithm moves J1 before J2.

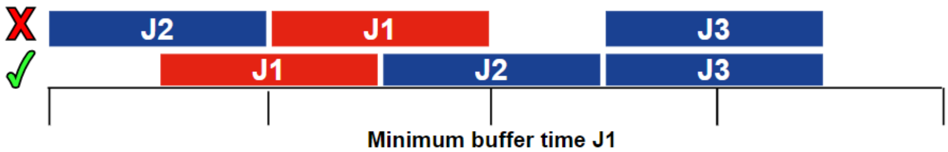

Figure 4.9 Placement of job with earliness (J1)

<u>Step 2: assign start times</u>

In this step, we assign start times to each job in a given sequence. We do this recursively in a backwards fashion, i.e., starting with the last job in the sequence. The last job is assigned a start time with the following formula:

$$\text{[2]} \quad start\ time\ last\ job = \min\{end\ scheduling\ horizon, due\ date\} - required\ buffer\ time - processing\ time - non\ working\ period\ if\ applicable$$

The following formula is used to assign start times to the remaining jobs (if applicable):

$$\text{[3]} \quad start\ time = \min\{due\ date - required\ buffer\ time\ (b), start\ time\ previous\ job\} - changover\ time\ if\ applicable - non\ working\ period\ if\ applicable - processing\ time$$

4.3.3 Subproblem 3: assign blends to storage

In the previous 2 subproblems, we assigned jobs to and scheduled mixing units. In subproblem 3 we assign blends to storage. The objective in this phase is to create an initial solution where IBC use and changeovers are minimised. We deviate from the solution approach as proposed by Kudva, Elkamel, Penky, & Reklaitis (1994) because we always have storage space available. Moreover, our storage scenario differs because in our inventory cost function the cost does depends on the storage unit used.

<u>General solution approach outline</u>

Figure 4.10 shows the steps we propose to solve the subproblem. The solution approach is a tailored heuristic. We have divided the heuristic into 2 steps that are executed consecutively for each blend. First, we apply a pre-processing step. If there are filling jobs restricted to IBCs, we assign this part of the blend to IBC(s). Then we divide the blend into families. A family consists of filling jobs with the same tank constraints (not every filling line can connect to every tank). Next, in step 2, we try to find an appropriate storage tank for each family. We prefer the smallest available tank which can fully store the family. If there are several tanks with the same size available, we prefer the one that does not require a changeover. If there are still several tanks left, we prefer the tank with smallest available time slot. If there is no tank available, we assign the blend to IBC(s). We start with the blend with the smallest storage time. We expect to be able to fill more litres into tanks if we place the blends in a sequence based on storage time.

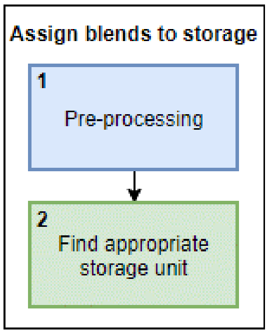

Figure 4.10 Solution approach overview subproblem 3

We now explain the solution approach in more detail. Figure 4.11 shows the logic flowchart of the heuristic to solve this subproblem. The steps of the heuristic can be identified using the colour scheme as shown by Figure 4.10. In the figure, we follow an example path that a blend follows through the logic flowchart to be assigned. The example path is highlighted in blue. In our explanation, we refer to actions with a number that can be found in the top left corner of the action. The result of the example path is shown by Table 4.3. The outer right column in the table shows why the blend was not assigned to the tank in that row and by which action. Each row in the table is a tank to which the job can be assigned. The tank to which the blend family is eventually assigned is shown in bold.

<u>Step 1: pre-processing</u>

Blends for filling jobs can be produced, e.g., 2 weeks in advance. We do not want to store filling jobs for such a long period in a tank. Therefore, filling jobs that are due more than *dd* (maximum due date difference) days after the due date of the (mixing) job are assigned to IBC(s). In addition, fillings jobs which require IBCs are also assigned to IBCs. Next, we split the blend into families of filling jobs with the same tank restrictions. We then execute step 2 for each family.

<u>Step 2: find appropriate storage unit</u>

In our example, we have a <u>10,000 L</u> blend, which consists of a <u>single filling job</u>, rinsing group <u>HDD</u> and a required storage time of <u>20 hours</u>. First, we check if there are tanks available during the required storage time, which in our example results in Table 4.3. Then, the tank(s) to which the family cannot connect are removed **(A1)**. Next, we check whether there is a tank available that already contains same fluid, and if so, we fill this tank. If not, we remove the tank(s) that are not equal to the smallest available tank that can fully contain the blend family **(A2)**. This avoids using unnecessarily larger tanks that are better suited for larger families. If there are tanks that can fully contain the blend family, we remove tank(s) that cannot fully contain the blend family **(A3)**. This ensures that we do not unnecessarily occupy 2 tanks.

Then we check if there are any tanks available that do not need to be rinsed. If there are, we remove the tanks that do need to be rinsed **(A4)**. Of the remaining tanks, we choose the one with the smallest available time slot **(T4)** **(A5)**. This leaves as much flexibility as possible left for other blend families which allows us to squeeze out as much idle time as possible (Papadimitriou & Kanellakis, 1980). Now that we have assigned the family to a tank it may be that the whole family did not fit in the tank. Therefore, we sum the IBC limit of all filling jobs. If there is more than this summation left, we go back and check again if there are available tanks. If there is less than the minimum left, we fill the remainder in IBCs. Then we check if there are any blends left to be assigned to storage and if so, we go back to select the blend with the smallest storage time. If not, the algorithm stops.

Table 4.3 Example path subproblem 3, bold is chosen tank

Tank (T)	Rinsing group	Size (L)	Available time slot (hours)	A
1	HDD	5,300	30	3
2	GEAR OIL	10,500	30	4
3	HDD	10,500	30	5
4	**HDD**	**10,500**	**25**	
5	HDD	14,500	30	1
6	HDD	29,000	30	2

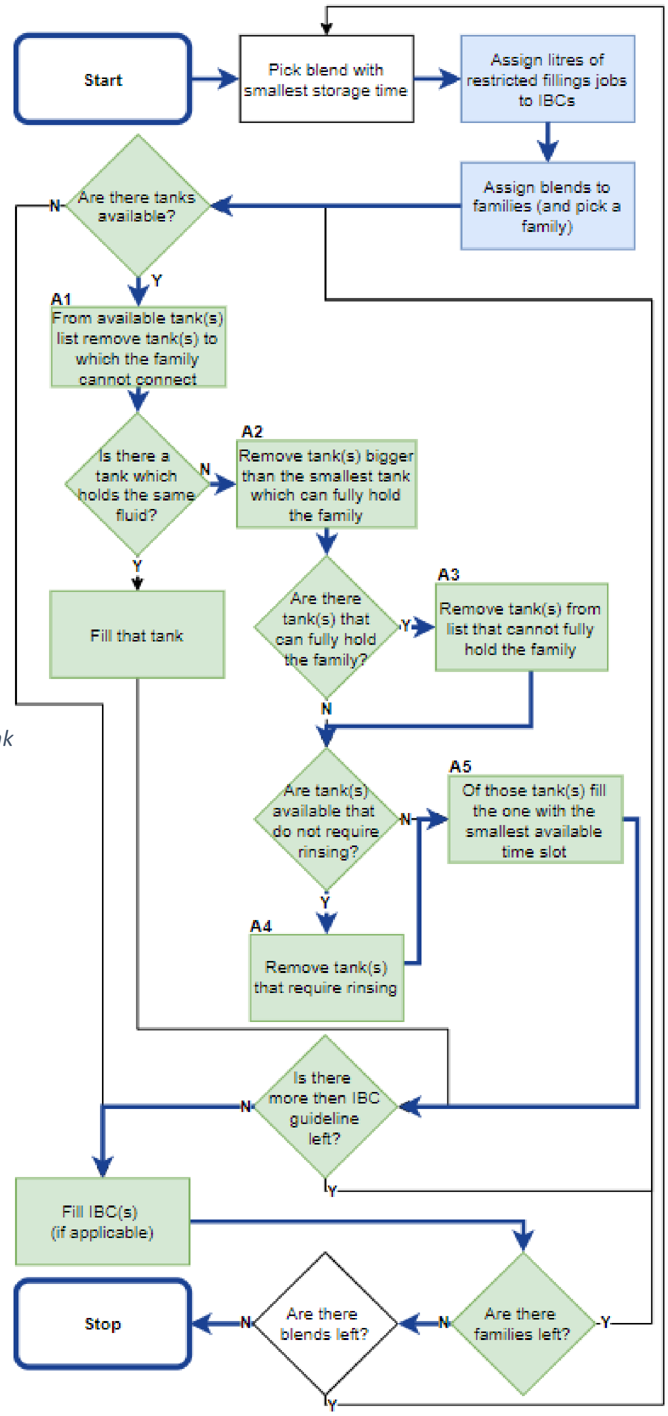

Figure 4.11 Logic flowchart: assign blends to storage

4.4 Phase 2: optimisation

In this section, we explain our approach to phase 2 of the heuristic: optimisation. Section 4.2. explained the general procedure in phase 2 with help of Figure 4.4. Before explaining in more detail how we propose to optimise, Section 4.1.1 first presents the scheduling strategies for optimisation. Next, Section 4.4.2 explains the optimisation of subproblem 2 and Section 4.4.3 the optimisation of subproblem 3.

4.4.1 Scheduling strategies

First, we illustrate and explain our definition of a group. Thereafter, we illustrate and explain the scheduling strategies (GS and JIT) by means of toy problems. In these toy problems, we schedule/assign only a few jobs/blends on a limited number of mixing units/tanks. GS groups jobs with equal rinsing groups to reduce rinsing. JIT schedules jobs as close to their due date as possible. The first toy problem illustrates the relation between GS and JIT scheduling when scheduling a mixing unit. The second toy problem illustrates the impact of GS and JIT scheduling of mixing units on the assignment of blends to storage. In the toy problems, different schedules/assignments are possible, but these do not contribute to the goal (illustrating the strategies), nor do they affect the objective results, so they are disregarded.

Definition of a group

The sequence from subproblem 1 can be viewed as a sequence of rinsing groups (also called family or batch in literature), where each group contains a subsequence of jobs with the same rinsing group. We define a group as a job or jobs surrounded by groups with a different rinsing group. This definition is illustrated by Figure 4.12, where colours define the rinsing groups. From here on we refer to the sequence as the succession of groups and a subsequence as the succession of jobs within a group.

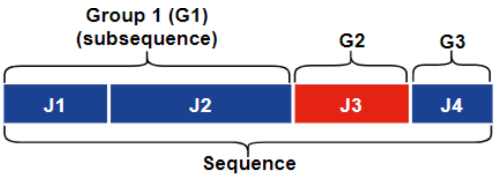

Figure 4.12 Definition of groups

Toy problem 1; mixing unit schedule, GS vs JIT

Table 4.4. shows the job information for this example. Suppose we have 3 jobs with a mixing time of 2 hours that are from 2 different rinsing groups. All jobs have different due dates and a minimum buffer time (*b*) of 2 hours, i.e., we want to finish production 2 hours before the due date. Figure 4.13 shows the schedules resulting from the different scheduling strategies. We can schedule the jobs to minimise earliness (JIT). In the JIT schedule, we need 2 changeovers. If we were to group the jobs based on their rinsing group (GS), we would need 1 changeover, as shown by Figure 4.13. But by doing this we have increased the earliness, as shown by Table 4.4. We also need to start producing earlier, as shown by Figure 4.13.

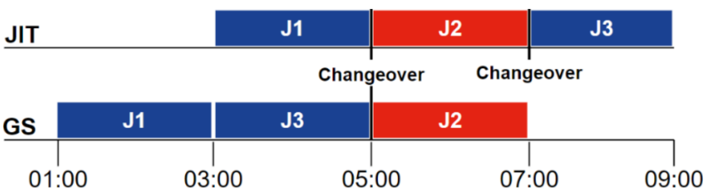

Figure 4.13 Mixing unit schedules toy problem 1

Table 4.4 Job information

Job (J)	Rinsing group	Due date	Earliness JIT	Earliness GS
1	Gear oil	07:00	00:00	02:00
2	Hydro	09:00	00:00	00:00
3	Gear oil	11:00	00:00	04:00

Toy problem 2; impact GS on storage schedule
After scheduling the mixing units, we assign the blends to storage. In this toy problem we use the same jobs as in the previous toy problem. Suppose we have 2 tanks available for assignments over the whole horizon. Each job is related to a single filling job of 1 hour. Figure 4.14 shows the assignments resulting from the different scheduling strategies. If we use the JIT schedule, we can assign every blend to a tank. However, using the GS schedule, we cannot assign every blend to a tank because there is no tank for the blend produced by job 2 from 07:00 to 10:00, as shown by Figure 4.14. Therefore, we must assign job 2 to IBCs. Thus, GS causes a blend to be assigned to IBCs.

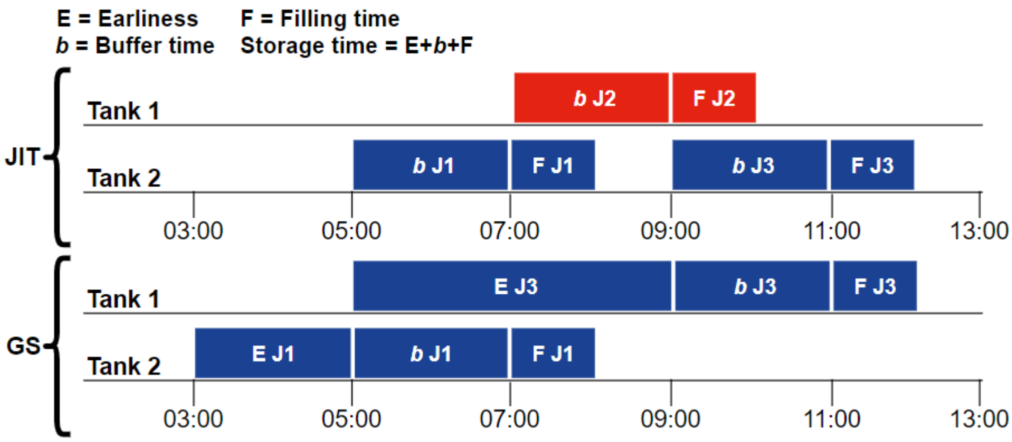

Figure 4.14 Tank schedules toy problem 2

4.4.2 Optimisation subproblem 2: schedule mixing units

In phase 1, we generated an initial solution for the scheduling of the mixing units. In phase 2, we optimise the initial solution of phase 1. We apply theorem 1 of Monma & Potts (1989). Their theorem is twofold: first, for the maximum lateness problem without tardy jobs, there is an optimal schedule where the jobs within a subsequence are ordered by EDD. Second, for minimising earliness, there is an optimal schedule where the jobs within a subsequence are sorted according to the LPT rule.

<u>General solution approach outline</u>
The initial solution from phase 1 sorts the jobs according to the EDD rule to ensure that there are no tardy jobs. To minimise earliness, i.e., schedule JIT, we sort as many subsequences as possible according to the LPT rule. Then we merge subsequences of the same rinsing group, which we call as GS. We apply JIT (step 1) first and then GS (step 2) because, even though the objective parameters are unknown, we assume minimising IBC use is more beneficial than minimising changeovers. Moreover, because we first apply JIT and then GS, we reduce the number of possible schedules, thus reducing the computation time.

<u>Step 1: JIT scheduling</u>
In the initial sequence, every subsequence is sorted by EDD. For each subsequence, we create a counterpart sorted according to the LPT rule. We now want to test which LPT subsequences we want/can use in our sequence of groups. We do this iteratively. We test every subsequence by swapping an EDD subsequence with its LPT counterpart to create a sequence. In each iteration one subsequence is swapped for its LPT counterpart, if feasible. Figure 4.15 shows an example result of the first iteration, note that SPT G6 is removed because it is infeasible. After creating a new sequence (because we swapped a subsequence), we assign start times (step 2 subproblem 2 phase 1) to create a schedule. For each swap that produces a feasible schedule, we remember the reduction in earliness that it caused. An infeasible schedule may be due to the violations of due or release dates. In each iteration, we keep the LPT subsequence that reduces earliness the most. We stop when there are no LPT subsequences left to swap or every swap produces an infeasible schedule.

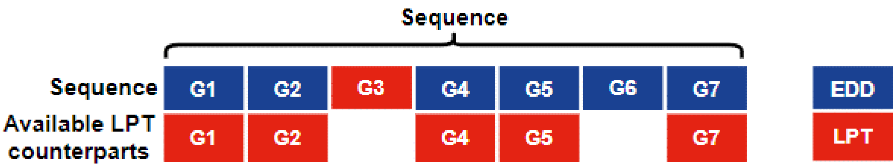

Figure 4.15 Example result of first JIT iteration (swapping EDD for LPT)

<u>Step 2: Group Scheduling (GS)</u>

After focusing on JIT to minimise the use of IBCs, we now focus on GS to minimise changeovers. There are 2 ways in which we can prevent a changeover. Firstly, we can advance a subsequence so that it merges with another subsequence with the same rinsing group. Secondly, we can advance a subsequence that is between 2 subsequences of the same rinsing group so these groups to merge. We apply GS iteratively in a similar way to JIT scheduling. Each iterations merges 2 subsequences, if feasible. Figure 4.16 shows an example result after 2 iterations, thus starting with 7 groups because there are 5 groups left. After a merge we sort the newly formed (larger) group according to LPT. Then we assign start times (step 2 subproblem 2 phase 1) to create a schedule for the original sorting's of the subsequences and the new LPT subsequence. If the merge is feasible, we remember the increase in earliness that it causes. In each iteration, we keep the merger that increases earliness the least. We stop when there are no more mergers left that do not cause an infeasible schedule or when changeovers are already minimal. In Figure 4.16, the newly formed group G1 did not have a feasible LPT subsequence, so the original sorting's of the subsequences was kept, unlike G2.

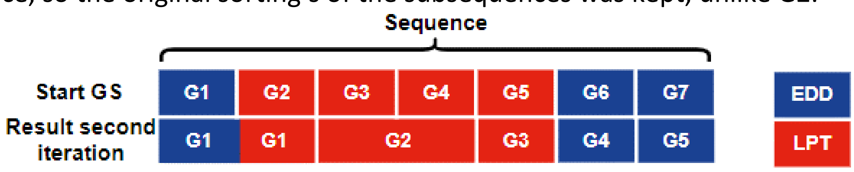

Figure 4.16 Example result of second GS iteration

4.4.3 Optimisation subproblem 3: assign blends to storage

In phase 1, we generated an initial solution by first picking the blend with the smallest storage time. We do not optimise this initial solution, but instead generate alternative solutions by randomising the picking of blends. This means that we no longer pick the blend with the smallest storage time first. This is similar to the way Kudva, Elkamel, Penky, & Reklaitis (1994) propose to optimise, but we only apply it to the last step. The optimisation of subproblem 3 is stopped when either the time limit or a predefined number of solutions have been generated.

4.5 Conclusion

To conclude this chapter, we summarise the main findings and conclusions in this section. In this chapter we explained our solution approach for the problem in hand. Thereby answering the question: *How do we find an improved scheduling strategy for the mixing plant?*

The problem in hand is modelled as a two-phase algorithm. In the first phase the algorithm generates an initial solution. In the second phase the algorithm employs the JIT and GS strategies and GRASP to improve upon the initial solution. For each subproblem, objectives are set to which cost parameters can be assigned, as yet these cost parameters are unknown. The solution approach is most similar to that of Kudva, Elkamel, Penky, & Reklaitis (1994). However, we deviate from their approach because our storage objectives and constraints differ. Our problem division is similar to that of Kudva, Elkamel, Penky, & Reklaitis (1994), dividing the problem into 3 subproblems: assign jobs to mixing units, schedule mixing units, assign jobs to storage. Their rules for assigning jobs to mixing units are applied and extended to take rinsing into account. To the best of our knowledge, we have developed a new heuristic for scheduling mixing units. The principle of the rules of Kudva, Elkamel, Penky, & Reklaitis (1994) for assigning jobs to mixing units is also applied to assigning jobs to tanks.

5 Solution evaluation

This chapter focuses on research question Q4: What is the effect of the proposed solution approach on the performance of the schedule? In this chapter, we evaluate solutions generated by the algorithm based on real life data to answer this question. Section 5.1 first describes the instances we use to test the performance of different parts of the algorithm. Then Section 5.2 describes the experiments for subproblem 2, scheduling mixing units, and subproblem 3, assigning jobs to storage units. Finally, Section 5.3 summarises and concludes this chapter.

5.1 Descriptions of test instances

To determine the effect of the algorithm on the performance of the mixing plant we use data from practice provided by Eurol. Another possibility is to generate data. We choose not to do this so that our results are not dependent on the method by which the data is generated. The algorithm reads data from Excel files, headings in each file are shown by Appendix A: Data in-and outputs. We do not evaluate subproblem 1, assigning jobs to mixing units. The reason is that this subproblem is heavily restricted and therefore has a small solution space and therefore a small influence on the final solutions compared to the other 2 subproblems. Thus, we do not need its evaluation to conclude whether the algorithm can improve the efficiency of the mixing plant.

1. Instances subproblem 2.

There are 14 instances to test subproblem 2, scheduling of mixing units. Each instance represents 3 days from a week between September up to December 2020 of MU D. Data is retrieved from schedules as made by the mixing planner, but what actually happened is retrieved from the Warehouse Management System (WMS)

2. Instances subproblem 3.

In subproblem 3, we assign jobs to tanks. There are 35 instances representing overlapping schedules from September up to 20 October 2020. An instance contains all jobs of all mixing units as scheduled by the mixing planner with a horizon of 3 days. The data in addition to which jobs are in the instance (e.g., completion time, due dates, filling times) are retrieved from the WMS. Therefore, the instances are fully in line with reality. We test the instances sequentially and individually. To ensure that the schedule in the individual case remains feasible/in line with reality, we need an additional constraint. Because not only can some tanks be unavailable at the start of the schedule, but also all tanks are unavailable at some point in the future due to jobs beyond the scheduling horizon. This is illustrated by Figure 5.1. Adding this constraint does not require changes to the algorithm.

Figure 5.1 Additional constraint to tank data

5.2 Experimentation

The algorithm is implemented in Python because it is fast, free and easy to use due to its built-in libraries. The Pandas library is used to read input data from Excel files. The Pandas library uses C++, which is even faster than Python, so we use Pandas functions as much as possible. We run the experiments on a Windows machine with 32 GB of memory and an I5 8600 CPU with 6 cores at 3.1 GHz.

5.2.1 Subproblem 2: scheduling mixing units

To solve subproblem 2, the algorithm applies JIT and GS scheduling strategies to schedule the mixing units. In this section, we test the algorithmic solution to subproblem 2 to see if it can lead to a more efficient use of mixing units. A more efficient use of mixing units is part of a more efficient use of mixing plant resources. Solutions are generated with different parameter settings to see what the best settings are. To do this, we first explain how we measure the performance of a solution. Then we analyse in detail a solution from the algorithm to validate it to ensure it is representative for reality, meaning it can be used in practice. Finally, we show the test results.

Performance indicators

The objective in solving subproblem 2 is to minimise the earliness and the number of changeovers. These objectives are performance indicators. The algorithm applies the conflicting JIT and GS scheduling strategies. We first maximise JIT and then we apply GS. With each GS iteration, a new solution is generated. To clearly demonstrate this effect, a schedule of MU A with many jobs (35) is chosen. Figure 5.2 shows the average earliness over 20 iterations of GS at a minimum buffer time of 2 hours. From here on we refer to the solution at iteration 12 as the Minimum Earliness (ME) solution. In addition to the ME solution, we also have the Maximum GS (MGS) solution at iteration 20. In this solution, the number changeovers is minimised by maximising GS. Finally, the 5th performance indicator is the number of Jobs Late (JL) in a schedule (taking into account mixing and due date deviations).

1. Total Minimum Earliness ME-E
2. Total Minimum Earliness Changeovers ME-C
3. Total Earliness Maximum GS MGS-E
4. Total Changeovers Maximum GS MGS-C
5. Number of Jobs Late JL

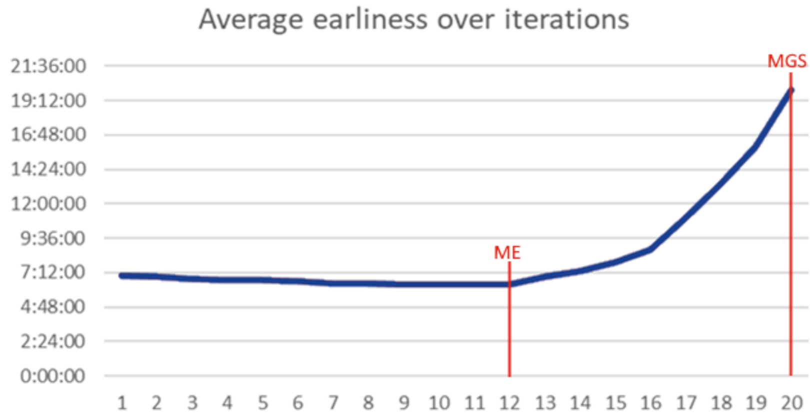

Figure 5.2 Average earliness over GS iterations

Validation

We want to know whether a solution provided by the algorithm can be valid, i.e., representative of reality. To validate the solution provided by the algorithm, we compare a manual solution (generated by a planner as explained in Section 2.2), what happened in reality, and a solution from the algorithm for MU D. The solution provided by the algorithm is the Minimum Earliness (ME) solution. The manual solution requires 10 changeovers, in reality 13 changeovers took place, the algorithm's solution requires 11 changeovers. The minimum buffer time (*b*) for the algorithm is 6 hours. Table 5.1 shows the manual, reality, and algorithm's schedule. The first column for each type of schedule shows the job order, the initial being determined by the manual solution. The 2nd column shows the end time of mixing by displaying the day in bold and then the hour and minute mixing finished. The due date in the next column has the same format. Finally, for each solution, the 4th column shows the number of hours and minutes a job's earliness. For the solution of the algorithm, we also show the actual earliness (only taking into account the deviation from the due date) in the same format as the earliness. The last row shows the total earliness, excluding negative values.

Jobs 1 and 15 are bulk jobs (trucks from customers) and thus do not need to meet the minimum buffer time. Furthermore, the start time of the schedule is 17:01 (time at which the schedule was manually created), so the algorithm cannot produce job 1 on time as it cannot produce before 17:01. Note that jobs 2 and 8 have equal due dates, this is a single filling job. The planner and mixing plant seem to know this and act accordingly, the algorithm cannot. To properly address this problem the due date of the second job should be delayed accordingly. Note job 10 has no actual earliness date, this is because this blend is made on stock. If the algorithm's solution had been implemented, 2 jobs would have been late. Job 2 would have been 6 hour and 9 minutes late. Then again, the due date of the job was advanced by 15 hours, meaning the buffer time must be 15 hours to ensure this job would be on time. Job 12 would have been 2 hours late, its due date advanced by 8 hours and 19 minutes.

Figure 5.3 shows the actual earliness of the manual, real and algorithmic solution, in the same colour scheme as in Table 5.1, and the mixing time and due date deviation. The actual earliness is calculated based on the theoretical mixing time and on the actual mixing time, i.e., accounting only for due date deviation and accounting for both due date and mixing time deviation. Negative values are excluded in the data with regards to actual earliness. Jobs should not be produced late, which does not occur in the algorithmic solution, but implementing the solution would cause jobs to be produced late. However, we conclude that this does not mean the algorithm's solution is not valid. Based on the given data input, the algorithmic solution is representative of reality, meaning it is feasible and can be used in practice. To ensure that we schedule jobs on time, the due dates should be updated more frequently. It is not advisable to produce all jobs at least 15 hours in advance, as this adversely affects the tank schedule. Another option is we accept some jobs are late.

Finally, the planner also received solutions from the algorithm for mixing unit schedules where he was currently working on. The solutions seemed valid to the planner and influenced the planner's decisions, thus providing decision support.

Table 5.1 Mixing units schedules; manual, reality, and algorithmic solution

	Manual			Reality				Algorithm				
Order	End time	Due date	Earli-ness	Order	End time	Due date	Earli-ness	Order	End time	Due date	Earli-ness	Actual Earliness
	Day HH:MM		HH:MM		Day HH:MM		HH:MM		Day HH:MM		HH:MM	
1	**3** 17:48	**3** 16:45	-01:03	1	**3** 17:05	**3** 16:45	-00:20	1	**3** 19:05	**3** 16:45	-02:20	
2	**3** 20:57	**4** 12:31	15:34	2	**3** 21:12	**4** 12:31	15:18	5	**4** 00:31	**4** 11:39	11:08	22:55
3	**4** 00:51	**5** 02:05	25:14	3	**3** 23:58	**5** 02:05	26:06	2	**4** 03:40	**4** 12:31	08:51	-06:09
4	**4** 04:52	**5** 04:17	23:25	5	**4** 01:14	**4** 11:39	10:24	8	**4** 06:31	**4** 12:31	06:00	13:48
5	**4** 06:59	**4** 11:39	04:40	6	**4** 03:30	**5** 00:33	21:02	9	**4** 11:00	**5** 08:03	21:03	12:03
6	**4** 10:32	**5** 00:33	14:01	11	**4** 05:04	**5** 08:20	27:15	6	**4** 14:33	**5** 00:33	10:00	44:23
7	**4** 13:14	**5** 09:28	20:14	4	**4** 14:39	**5** 04:17	13:37	3	**4** 16:39	**5** 02:05	09:26	09:11
8	**4** 16:23	**4** 12:31	-03:52	7	**4** 16:29	**5** 09:28	16:58	4	**4** 20:40	**5** 04:17	07:37	07:25
9	**4** 18:55	**5** 08:03	13:08	8	**4** 18:38	**4** 12:31	-06:07	11	**4** 23:32	**5** 08:20	08:48	03:32
10	**4** 22:09	**5** 13:06	14:57	9	**4** 23:03	**5** 08:03	08:59	7	**5** 02:32	**5** 09:28	06:56	07:14
11	**5** 01:01	**5** 08:20	07:19	12	**4** 23:52	**5** 14:47	14:54	10	**5** 06:04	**5** 13:06	07:02	
12	**5** 03:26	**5** 14:47	11:21	13	**5** 01:36	**5** 20:08	18:31	12	**5** 08:29	**5** 14:47	06:18	-02:01
13	**5** 06:26	**5** 20:08	13:42	14	**5** 03:29	**6** 07:02	27:32	13	**5** 12:38	**5** 20:08	07:30	05:25
14	**5** 08:29	**6** 07:02	22:33	10	**5** 12:30	**5** 13:06	00:36	15	**5** 16:45	**5** 16:45	00:00	
15	**5** 12:18	**5** 16:45	04:27	15	**5** 14:57	**5** 16:45	01:48	14	**5** 21:09	**6** 07:02	09:53	01:51
16	**5** 18:15	**6** 08:48	14:33	16	**6** 00:15	**6** 08:48	08:32	16	**6** 02:48	**6** 08:48	06:00	15:58
		Total earliness	205:08			Total earliness	211:32			Total earliness	126:32	

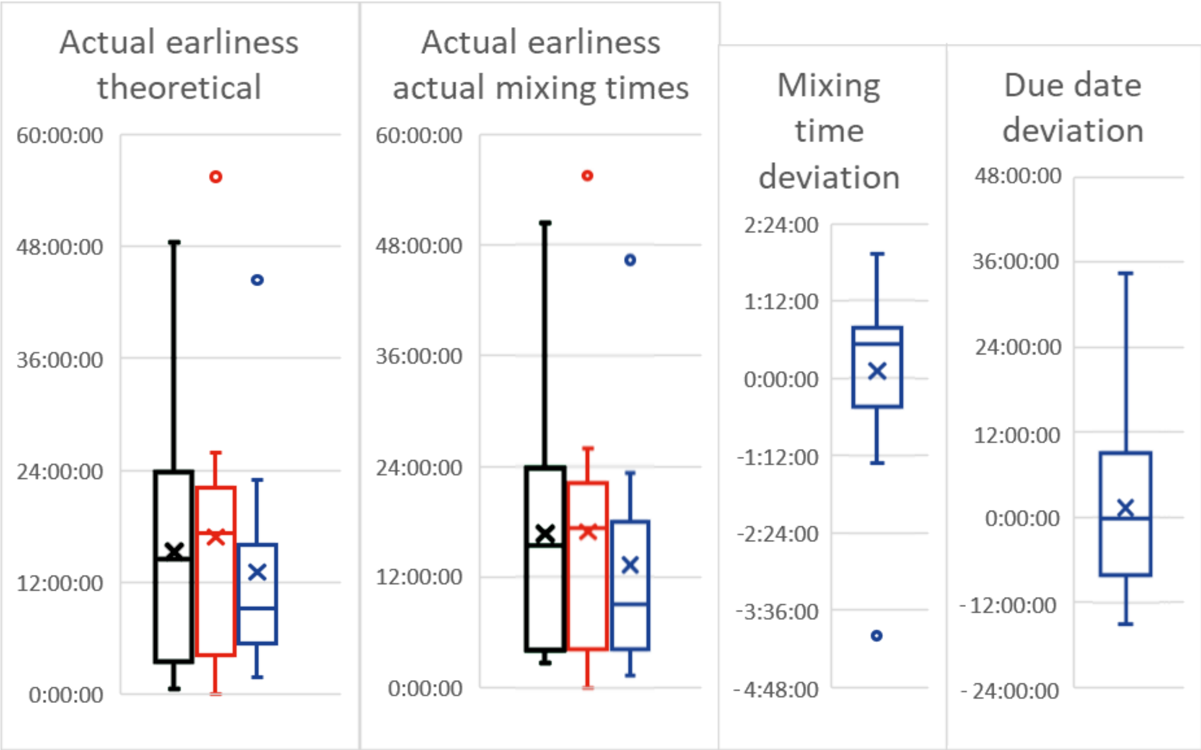

Figure 5.3 Actual earliness per solution type plus mixing time and due date deviation

Results

Now we have established that the algorithmic solutions are representative of reality, we proceed to test all the instances. All schedules are generated by the algorithm in less than 1 second. The algorithm has 1 parameter, namely the minimum buffer time (*b*). To find the best setting, the results are shown for different settings. Table 5.2 shows the results, the earliness figures (ME-E, MGS-E) are shown in hours. The results are sorted by total mixing time, which indicates the difficulty of the instance. The minimum buffer time in reality is nonnegative. Bulk jobs are excluded from the Jobs Late (JL) figure.

In instances 1, 3, 5, 6, 8, and 13 with *b* 6, ME-E is equal to MGS-E, meaning we cannot apply JIT and GS. This is due to the instance being infeasible, often because the first job is infeasible, which is also the case in Table 5.1. It may be that the first job is in progress at the time. Therefore, we conclude that the algorithm must be given a time that it cannot exceed (the planner should then possibly also remove the first job from the instance). In the case of a minimum buffer time of 6 hours, the ME solution improves the earliness in 12 out of 14 instances, in 5 out of 14 instances the number of changeovers required is also decreased. In case of maximum GS, the earliness decreased in 9 out of 14 instances, reducing the number of changeovers in 11 out of 14 instances.

The improvements increase when the buffer time is reduced. However, a buffer time of 6 hours is preferred, because decreasing the buffer time below 6 hours increases the number of jobs that are late. With a 6-hour buffer time there are no jobs late in the first 24 hours, with the other 2 settings there is 1 job late in the first 24 hours (not counting infeasible instances). This also shows that rescheduling is required within the scheduling horizon since most instances have late jobs after 24 hours.

Results show that, on average, the algorithm can schedule the mixing units with fewer changeovers and can schedule jobs less early. Some instances are infeasible, so the algorithm is unable to improve because the optimisation phase is not started.

Table 5.2 Results solution evaluation: scheduling mixing units

					Algorithm														
	Reality				***b* 2 hours**					***b* 3 hours**					***b* 6 hours**				
Order	**Total mixing Time** HH:MM	**E**	**C**	**Min. *b*** HH:MM	**ME-E**	**ME-C**	**MGS-E**	**MGS-C**	**JL**	**ME-E**	**ME-C**	**MGS-E**	**MGS-C**	**JL**	**ME-E**	**ME-C**	**MGS-E**	**MGS-C**	**JL**
1	57:50	284	5	08:59	60	4	60	4	6	77	4	77	4	6	119	4	119	4	5
2	54:43	263	7	10:23	91	6	157	5	0	111	6	177	5	0	165	6	213	5	0
3	52:04	145	7	00:04	113	10	113	10	1	128	10	128	10	1	173	10	173	10	1
4	51:07	252	7	04:27	117	6	256	3	1	133	6	272	3	1	162	6	169	5	3
5	49:40	119	7	02:54	117	5	117	5	1	115	6	115	6	2	152	6	152	6	1
6	48:43	204	5	01:55	79	4	79	4	5	98	4	98	4	4	130	5	130	5	3
7	48:28	379	10	06:18	74	10	274	8	2	89	10	290	8	2	123	10	313	8	1
8	46:33	190	8	00:00	83	8	106	7	2	87	9	87	9	2	112	9	112	9	2
9	45:36	200	11	04:40	75	11	229	5	2	91	11	202	6	2	126	11	205	7	1
10	44:09	396	5	15:53	56	5	97	4	8	72	5	112	4	4	96	5	135	4	2
11	42:57	405	6	08:36	54	9	101	5	1	71	9	117	5	1	111	9	153	5	1
12	42:00	287	6	13:21	66	8	105	5	1	80	8	119	5	1	103	8	113	5	1
13	41:31	225	7	03:31	143	3	143	3	1	158	3	158	3	1	192	4	192	4	1
14	38:38	111	10	03:53	48	8	53	6	0	62	8	67	7	0	107	8	112	6	0
	Total	**3,460**	**101**	**-**	**1,176**	**97**	**1,890**	**74**	**30**	**1,372**	**99**	**2,019**	**79**	**27**	**1,871**	**101**	**2,291**	**83**	**22**

5.2.2 Subproblem 3: assigning jobs to storage units

To solve subproblem 3, for each job the algorithm looks for an available tank with the smallest capacity that can completely contain the job. If there are multiple available tanks with the same capacity, the algorithm tries to prevent rinsing. In this section we test the algorithmic solution to subproblem 3 to see if it can lead to a more efficient use of the mixing plant's resources and with which parameters. First, we explain the performance indicators and then we analyse a solution from the algorithm in detail to make sure that it is representative of reality. Finally, we show the test results.

Performance indicators
The objective in solving subproblem 3 is to minimise the number of IBCs (<u>IBCs</u>) used and to minimise the number of tank changeovers (<u>C</u>). The highest priority is to minimise the number of IBCs used. Therefore, the results shown are the results for a minimum use of IBCs. For the optimisation, the algorithm applies a GRASP strategy. To see the effect of the optimisation and the difficulty of the problem for the algorithm, we show the percentage of solutions using less IBCs than reality (<u>%IMP</u>). Furthermore, we show the number of Unique Solutions (<u>US</u>) found by the algorithm to get an indication of the preferred number of optimisation iterations. If for example few new unique solutions are found after 100 iterations optimisation can stop such that a good solution can be presented quickly.

Validation
We want to know whether a solution provided by the algorithm can be valid, i.e., is representative of reality, meaning it can be used in practice. To validate the solution, we compare reality with a solution from the algorithm. The minimum observed time between 2 jobs on the same tank is in reality 28 minutes. Therefore, the algorithm must adhere to a minimum buffer time (*bt*) of 30 minutes. The *bt* ensures robustness of the solution. However, the algorithm can provide solutions quickly and there are no dependencies on the tank schedule, unlike mixing unit schedules where the hot room depends upon. Therefore, if up to date data is available, a *bt* of 0 can be sufficient. To show the impact of the minimum buffer time we also generate a solution with a minimum buffer time of 0 minutes. The minimum IBC solutions after 100 iterations are shown by Table 5.3, the calculation takes approximately 81 seconds per *bt*.

Table 5.3 Validation example results

Solution type	IBCs	C	%IMP	US
Reality	50	14	-	-
***bt* 30**	44	14	9%	25
***bt* 0**	37	19	34%	31

Figure 5.4 shows the reality and *bt 30* solution in more detail. The figure shows in green the end time of the same job in both solutions. In reality, the tanks are emptied one by one, the algorithm does not interpolate when a tank will be empty and therefore the time that each tank is empty is equal to the end time of the filling. This does not mean the solution is not valid, but that tanks unnecessarily constrained. In reality, tank 41 is empty at 7:30 and tank 38 at 10:47. The algorithm thus unnecessarily constraints the tanks by approximately 3 hours for this job. Overcoming this issue adds further complexity (e.g., due to shifts) and is therefore not implemented therefore, it is a future research opportunity. The solution of the algorithm does not violate any constraint.

Finally, the mixing plant also received solutions from the algorithm for active schedules. The solutions seemed valid but did not influence the decisions because data was not up to date, mainly due to the mixing plant deviating significantly from the mixing unit schedules. Therefore, the solution provided by the algorithm could not be implemented. We conclude that the algorithm is representative of reality. Although the solution could not be implemented in an active schedule, it still seemed valid to experts, taking into account the fact that the data is not up to date.

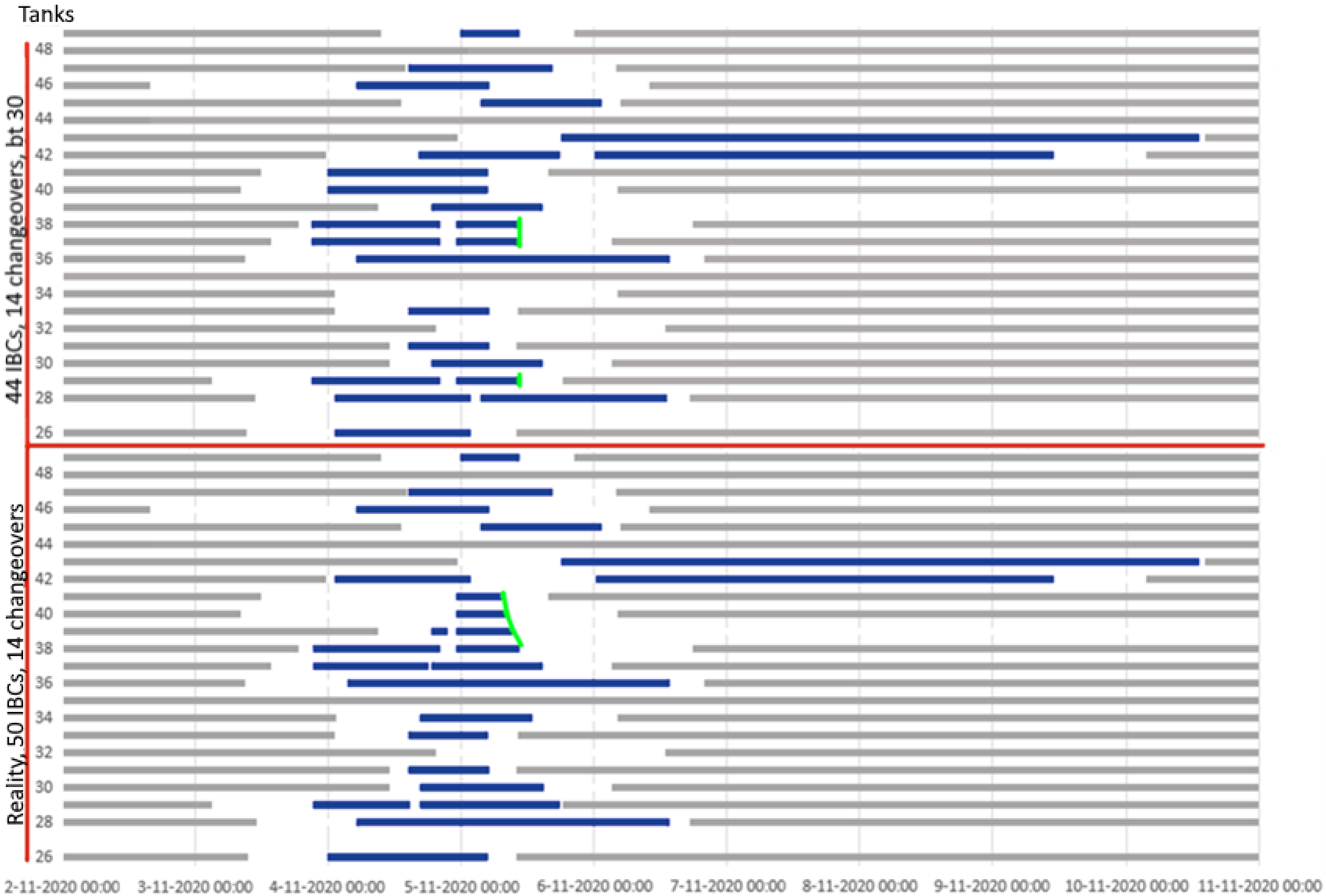

Figure 5.4 Gantt chart manual and algorithm solution to the tank scheduling problem

Results sequential evaluation

We evaluate all 35 instances by scheduling sequentially. Note that the instances overlap, therefore double jobs were removed using a macro thus creating 1 big instance. Based on the results, we can determine the setting for the maximum due date difference (*dd*) in days. This is the time between the completion time of the job and filling job. Based on the sequential evaluation we can determine the best setting for this parameter, because *dd* in the individual evaluation does not show the impact on 'future' instances as in the sequential evaluation (due to the additional constraint in individual evaluation).

In reality, over the whole horizon (35 instances), 1,844 IBCs and 264 tank changeovers were needed. Table 5.4 shows the results of the algorithm with different *dd* (in days) settings. We do not optimise because the number of possible solutions is huge, computationally expensive, and it may be that for one setting one very good solution is found, creating an unfair comparison. This means that we assign jobs based on the storage time, the job with the smallest storage time is assigned first.

Table 5.4 Results solution approach evaluation: assigning jobs to storage units (sequential)

dd 4		dd 5		dd 6		dd 7		dd 8	
IBCs	C	IBCs	C	IBCs	C	IBCs	C	IBCs	C
1,975	273	1,769	269	1,749	267	1,804	254	1,876	257

Based on the results, we conclude that a maximum due date difference (*dd*) of 6 is best, which means that filling jobs with a *dd* larger than 6 days are directly assigned to IBCs. This ensures that tanks are not occupied for more than 6 days.

The algorithm can help optimise the plant by providing insights in the impact of changes to the plant. In Section 2.1.5 we mentioned Eurol wants to remove tanks 45 to 49 and replace them with more storage tanks. Eurol is also making changes to filling lines, e.g., adding a faster line. We do not know what the effect of a faster line will be. However, we can get an impression of what happens when we replace tanks. We only changed the tank input data, removed tanks 45 to 49 (5x 14,500 L) and added their replacements (10x 8530 L). Sequential evaluation with *dd* 6 results in 1,603 IBCs with 303 changeovers. The number of IBCs used is reduced but the number of changeovers increases. The increase in changeovers is probably due to the fact that more blends are filled into tanks, and that the replacement tanks are smaller (8,530 litres instead of 14,500), so that more tanks may be needed for a single job. However, the greater number of changeovers may be a lesser problem because these tanks can be more easily rinsed. To optimise the plant, the cost parameters must therefore be quantified.

Results individual evaluation

Of the 35 test instances, 15 are selected for individual evaluation. Note these instances overlap. These 15 are representative of different load scenarios. Based on the results, we can determine a recommended number of iterations. We also generate solutions without the additional constraint (Figure 5.1) to demonstrate the effect of the constraint. We consider this important because the constraint makes the solution space of the algorithm smaller, which is a disadvantage. Table 5.5 shows the results sorted on the timeline, i.e., the first row is in September, the last in October. The last row is a summation of the rows above it.

In some instances, we cannot improve upon reality with the additional constraints. This is mainly due to first assigning jobs from MU D and then assigning jobs from MU A. Therefore, the algorithm cannot find every possible solution which may be better in some cases. However, increasing the solution space by also considering scheduling MU A first may lead to the algorithm needing more iterations to find a good solution. This increases the required computation time, which must remain below 5 minutes.

Without the additional constraint, the number of IBCs used cannot be reduced in 2 out of 15 instances. However, this does not translate directly to an improvement overall as the instance thereafter may be negatively affected. For example, the improvement in the sequential evaluation is lower than the sum of improvements of only 15 out of 35 instances.

The number of unique solutions remains relatively the same when the additional constraint is removed. Moreover, the best solution does become better after 500 iterations compared to 100 iterations. The number of unique solutions can differ significantly. In some scenarios, the best solution that the algorithm can find in very few iterations. Thus, in some scenarios, very few iterations may be sufficient. However, since this is not always the case, we recommend using 100 iterations, which gives a computation time of approximately 80 seconds. Based on the percentage improved solutions, 10 iterations should be sufficient to give a good enough answer to “what if” questions. This allows the algorithm to answer “what if” questions in approximately 8 seconds.

5.3 Conclusion

To conclude this chapter, we summarise the main findings and conclusions in this section. In this chapter we have presented the impact of the solution on subproblem 2, scheduling mixing units and 3, assigning jobs to storage units. Thereby answering the question: *What is the effect of the solution on the performance of the schedule?* We only experimented with 2 of 3 subproblems as these have the largest impact on the use of resources within the mixing plant because subproblem 1 has a small solution space and therefore not much can be improved.

We conclude that solutions provided by the algorithm for subproblem 2 are valid. When applying the solution to subproblem 2 we recommend using a minimum buffer time of 6 hours (*b*6). This should allow feasible schedules if the schedule is feasible before optimisation. In case we want to minimise earliness, which is the main objective for MU D, the algorithm can reduce the earliness by 33,79% with the same number of changeovers required using the recommended parameter settings. We conclude the algorithm can provide decision support to the planner.

For subproblem 3 we also conclude the solutions provided by the algorithm are valid. The recommended settings when applying the solution to subproblem 3 are a maximum due date difference of 6 and optimising for 100 iterations. Completely changing the tank schedule does not affect any operations in the production plant. Therefore, we recommend not to use the minimum buffer time function of the algorithm in the final implementation, i.e., set it to 0. Optimisation for 100 iterations takes approximately 80 seconds which is the most time-consuming aspect of the algorithm (subproblem 1 and 2 combined takes approximately 10 seconds). Thus, the algorithm adheres to the maximum calculation time requirement of 5 minutes. We have shown that the algorithm can reduce the number of IBCs used in the sequential evaluation while also reducing the number of changeovers. Furthermore, in the individual evaluation (more restricted) we have shown the algorithm can compete with the solutions from reality.

Because of the positive results for 2 of the most impactful subproblems we can conclude beyond reasonable doubt that the algorithm can have a positive effect on using resources more efficiently. However, before this effect can be realised the algorithm must be implemented which requires Eurol to update their data model.

Table 5.5 Results solution approach evalation: assigning jobs to storage units (individual)

			Algorithm													
			With additional constraint										No additional constraint			
	Reality		Initial		100 iterations				500 iterations				100 iterations			
	IBCs	C	IBCs	C	IBCs	C	%IMP	US	IBCs	C	%IMP	US	IBCs	C	%IMP	US
	71	16	59	13	56	16	100%	37	56	16	100%	52	56	12	100%	31
	78	22	80	17	70	20	48%	37	70	20	55%	58	68	18	100%	16
	90	17	100	17	90	19	0%	54	90	19	0%	111	85	12	59%	56
	108	26	149	22	113	27	0%	84	113	27	0%	249	92	25	11%	85
	123	22	174	18	116	20	7%	85	116	20	4%	239	107	18	30%	78
	123	15	154	11	144	15	0%	77	144	15	0%	193	103	14	35%	77
	112	13	120	15	98	20	54%	38	98	20	47%	65	96	13	66%	57
	158	18	167	19	144	24	12%	89	144	24	18%	287	134	24	59%	88
	113	20	125	19	120	18	0%	64	120	18	0%	172	114	20	0%	55
	117	14	122	16	108	18	27%	39	108	18	31%	70	103	14	81%	51
	118	13	143	12	138	15	0%	2	138	15	0%	2	109	10	100%	23
	104	13	96	11	96	11	100%	3	96	11	100%	3	96	6	100%	8
	110	10	118	10	118	10	0%	4	118	10	0%	4	118	5	0%	6
	72	9	68	4	68	4	100%	1	68	4	100%	1	67	4	100%	3
	112	7	83	9	82	11	100%	7	82	11	100%	7	69	8	100%	11
Total	**1,609**	**235**	**1,758**	**213**	**1,564**	**247**	**-**	**621**	**1,564**	**247**	**-**	**1,513**	**1,430**	**203**	**-**	**645**

6 Solution implementation

To use the resources of the mixing plant more efficiently with an algorithm it must be implemented effectively. In this chapter Section 6.1 discuss what is needed before the algorithm can be implemented and Section 6.2 the recommended approach. Finally, Section 6.3 discusses how the algorithm can be used and Section 6.4 evaluates the view of the company regarding implementation.

6.1 Requirements

One problem we face is the large amount of data required. Moreover, frequent updates of this data are necessary to achieve the best results. The data is currently incomplete, e.g., there is no data regarding the hot room and jobs and filling jobs are not properly linked. Providing the algorithm with data is currently a manual process. To implement the algorithm effectively, the data must be complete and automatically provided to the algorithm. With this study, we provide Eurol with data model requirements for scheduling the mixing plant, as shown by Appendix A: Data in-and outputs.

An interface must be created such that the planner can interact with the algorithm in a user-friendly way. The interface should allow the planner to ask “what if” questions for better decision support. Soft constraints, such as not mixing for customer trucks when it rains, cannot be taken into account at present. Moreover, the current data model requires some human interpretation because data is often not complete. Also, there can be peculiarities (as explained in Section 2.1, explaining the production process) that can require changes that theoretically worsen the solution but in practice improve it or even make it feasible. Thus, there is still a need for a mixing planner.

6.2 Recommendations

Because the algorithm solves the subproblems one by one, algorithmic solutions can be implemented one by one. This turns out to be a great advantage since providing data to the algorithm is currently a manual process. Providing data for subproblem 2 takes approximately 5 minutes per mixing unit, while providing data for subproblem 3 takes approximately 2.5 hours. Therefore, we recommend to first implement the algorithmic solution to subproblem 2. Then concentrate on automatically providing data to the algorithm and updating the data model itself. Only then should the algorithmic solution to subproblem 3 be implemented, as it will most likely not be effective without up-to-date data. As mentioned by (Karimi & Reklaitis, 1985), small variations can affect all scheduled storage operations. Therefore, a solution to subproblem 3 provided by the algorithm can quickly become infeasible without up-to-date data. When data is provided automatically to the algorithm, we recommend letting the algorithm calculate solutions for subproblem 3 continuously, as there is no reason to freeze the tank schedule.

We recommend first implementing the algorithmic solution with an interface for the planner. Thereafter, it can be expanded so that employees of the mixing plant can also get decision support, for example during night shifts when the planner is not available. When implementing the algorithm, we recommended that the results are made public within Eurol. Publicising results can make departments to work together more efficiently. For example, the logistic department contacts the mixing plant every morning to ask how many IBCs they can expect to receive that day.

Nah, Lau & Kuang (2001) advise there to be a project ‘champion’, this member should have business, technical and leadership competencies. Therefore, we recommend that there is 1 person dedicated to implementing the solution approach, i.e., the ‘champion’. The champion’s technical competencies must include knowledge of the solution approach and the data model. The champion must continuously resolve conflicts, manage resistance and, be available to transfer his knowledge of the solution approach to others.

6.3 Using the algorithm

To further optimise the schedule, manual adjustments can be made to the solution(s) of subproblems. Moreover, these manual adjustments can be fixed and provided to the algorithm as input for a subproblem. Manual adjustments can make the algorithm act as a simulator for answer "what if" questions the planner has. Input for the algorithm, such as available tanks or IBC guidelines, can also be modified. In doing so the algorithm can act as a simulator to optimise the production plant. Furthermore, when solving a subproblem the planner can use the parameters *b, bt* and *dd* to tweak the solution(s).

Manual adjustments can be made to the solution of subproblem 1 in phase 1. Adjustments can be made if the planner thinks it can be done better or because of undocumented constraints. Undocumented constraints may be, e.g., if a blend is to be produced of more than 1,700 kg but requires agitating at 1,500 kg, it cannot be produced on MU A and must be produced on MU B. This is because some raw materials have to mixed first, resulting in mixing less than 1,700 kg, which MU A cannot do. Undocumented constraints rarely occur in subproblem 1. Manually made adjustments can be given as input to the algorithm to generate a solution for subproblem 2.

Adjustments can also be made to the solution of subproblem 2 after phase 2. As with adjustments in subproblem 1, adjustments can be made if the planners thinks it can be done better or because of undocumented constraints. Undocumented constraints can be soft constraints such as not mixing for customer trucks when it rains. Manually made adjustments can be given as input to the algorithm to generate a solution for subproblem 3.

In subproblem 3, adjustments can be made before any solutions are generated. Blends can be assigned manually to one or more storage units, after which the algorithm assigns the remaining blends. adjustments can be made if the planners thinks it can be done better. There are undocumented constraints, such as not assigning a blend less than approximately 12,000 litres to a large tank, e.g., tank 44 of 37,000 litres, due to rinsing. These constraints can easily be added to the algorithm when conclusions regarding these constraints are drawn.

6.4 Company survey

In order to evaluate the company's view on the implementation, we conduct a survey. The Unified Theory of Acceptance and Use of Technology (UTAUT) method of Vekatesh, G. Morris, B. Davis, D. Davis (2003) is used to structure the survey. This method divides the survey into 5 parts:

- **Performance Expectancy (PE):** The degree to which an individual believes that using the solution approach will help him or her to attain gains in job performance.
- **Effort Expectancy (EE):** The degree of ease associated with the use of the solution approach.
- **Social Influence (SI):** The degree to which an individual perceives that important others believe he or she should use the solution approach.
- **Facilitating Conditions (FC):** The degree to which an individual believes that an organisational and technical infrastructure exists to support using the solution approach.
- **Behavioural Intention (BI):** The degree to which an individual has the intention to use the solution approach.

There are 3 participants for the survey: a planner, the manager of the planning department and the manager of the mixing plant. The method identifies 4 variables to better understand the input of the participants, namely gender, age, experience and voluntariness of use. All participants are of the same gender, have almost the same age and, work at Eurol for more than 10 years. Since gender, age and experience are almost equal, we assume the impact of these variables is negligible. It is not known whether using the solution approach will become mandatory. We assume that participants do not feel obliged to use the solution approach because no conclusions have yet been drawn. Moreover, according to Vekatesh, G. Morris, B. Davis, D. Davis (2003) the effect of the voluntariness variable (in case it is mandatory) also decreases with experience. All participants have significant experience at Eurol. If it is not mandatory, there is no effect. Therefore, we conclude that the 4 variables are the same for all participants and thus have no effect on the results.

The participants are not equally knowledgeable of the solution approach and this research. Therefore, the participants are first informed by means of a presentation. The presentation lasts approximately half an hour and is aimed at providing the information needed to answer the questions of the survey. Even after the presentation, the participants are not equally knowledgeable. The manager of the planning department, A. van Harten, is well informed and was not present at the presentation. The manager of the mixing plant is least informed because he (of all participants) was the least involved in the research.

The survey consists of 17 statements and 3 open questions. A five-level Likert scale is used to indicate the extent to which participants agree with a statement. There are 5 levels of agreement: strongly disagree, disagree, neutral, agree or strongly agree. If a participant strongly disagrees the score is 1, if the participant strongly agrees the score is 5, other levels score between 1 and 5. Appendix B: Company surveyelaborates upon the survey, showing the questions and all answers. Table 6.1 shows the mean score and the standard deviation per statement. Below the table, we discuss the results per type of statement. Thereafter, we discuss the open questions.

Table 6.1 mean score and standard deviation per statement

Statement [type . total]	Mean	Standard deviation
PE 1.01	4,33	0,58
PE 1.02	4,33	1,15
PE 1.03	3,33	0,58
EE 2.04	3,00	1,00
EE 2.05	3,00	1,00
EE 2.06	4,00	1,00
SI 3.07	3,33	0,58
SI 3.08	3,33	1,15
FC 4.09	3,67	0,58
FC 4.10	2,00	0,00
FC 4.11	2,33	1,15
FC 4.12	2,33	1,53
FC 4.13	4,00	0,00
BI 5.14	4,33	0,58
BI 5.15	3,33	0,58
BI 5.16	4,33	0,58
BI 5.17	4,00	1,00

Performance expectancy

The performance expectancy is relatively high with a low standard deviation. PE statements 1 and 2 refer to the use of the algorithm for scheduling, statement 3 refers to the use of the algorithm for plant design. In summary, it is expected that the resources of the mixing plant can be used more efficiently by using the algorithm. The use of the algorithm for plant design can be an interesting topic for further research.

Effort expectancy

The scores for the effort expectancy are relatively low, meaning that the expected effort to use the algorithm is relatively high. Based on the statements and their scores, we conclude that before the algorithm can be used, it needs an interface to create a user-friendly experience. The interface must also allow for easy use of the parameters (*dd, bt, b*). Finally, statement 3 refers to the time it takes to use the algorithm. The participants strongly agree that it currently takes too much time to use the algorithm. Therefore, we conclude that, in addition to the interface, the data model of Eurol must be updated. Moreover, the data should be automatically fed to the algorithm.

Social influence

The mean score is just above neutral, meaning that the participants do not agree that the use of the algorithm is socially supported. Interestingly, the manager of the mixing plant gave the lowest scores to both SI statements. We conclude that it is important to create KPIs in order to better track efficiency. This can help to create support for the use of the algorithm because, we expect that when the algorithm is used, efficiency increases. Being able to show this improvement should help create support.

Facilitating conditions
Statements 10 to 13 score negatively. These statements are about resources, knowledge and whether there is someone available within the organisation for questions. This means that the participants believe that further investment is needed to implement the solution approach. Interestingly, the manger of the mixing plant believes there is someone available to ask questions (statement 12). Statements 9 and 13 score positively, these refer to the implementation recommendations and whether the solution approach fits in the intended way of working of the planning department. We conclude that there should a project 'champion', as mentioned in the recommendations.

Behavioural intention
BI scores relatively high. The statements refer to whether the use of the algorithm is a good idea, whether the scheduling process proposed by this research is logical and efficient and, whether the participants intend to use the recommendations and conclusions of this research. Because the score is relatively high, it means that the participants believe that using the algorithm is a good idea and that they intend to use the recommendations and conclusions of this research.

Open questions
We draw the following conclusions from the open questions:

- Participants believe that using the algorithm can and has provided new insights.
- Participants had hoped that implementation would have been further along. However, participants mention that data is not sufficiently available to implement the algorithm.

Also, the manager of the mixing plant indicated that he believes to be insufficiently knowledgeable to properly score the statements and answer the questions. The scores of the manager of the mixing plant are lower than the scores of the other participants in all but 2 of the statements. So, there is no 'unfair advantage' in the mean scores, the opposite could be true.

7 Conclusions and recommendations

In this chapter, we conclude the thesis. First, we draw our conclusions in Section 7.1. Then we give our recommendations to Eurol, discuss the limitations of this research and its scientific contribution in Section 7.2. Finally, Section 7.3 presents our suggestions for further research. These suggestions can be used by future researchers for similar problems, but also by Eurol.

7.1 Conclusions

We want to answer the following research question with our research:

How can Eurol improve their scheduling strategy of the mixing plant so that the resources of the mixing plant are used more efficiently?

The mixing plant is a part of the Eurol production plant that is responsible for the production of blends. The blends are produced in mixing units and then temporarily stored in storage units (tanks or IBCs). Filling lines empty the storage units to fill a large variety of containers. Currently, scheduling is a largely manual process. An analysis of the process has shown that the process is subject to many peculiarities, variation and deviations. Based on a detailed data analysis of the mixing plant, we confirm the process is highly variable. We also conclude that dedicated tanks are not as preferable as they seem. Furthermore, we assume that there is room for improvement, among other things based on the buffer time of tanks, which is more than 20 hours on average. The buffer time of tanks is key to using the mixing plant resources more efficiently. After all, more efficient use of resources means that blends are stored in tanks instead of IBCs. Although cost parameters are not quantified we conclude this is the highest priority. The second priority is the reduction of changeovers caused by rinsing. Rinsing is necessary when something containing a blend changes product group, e.g., hydraulic oil to gear oil.

Based on our literature review we classified our problem and found scheduling strategies. The scheduling problem of the mixing plant is identified as a single-stage scheduling problem with parallel machines and storage. The scheduling problem is NP-hard. This means that for realistic problem sizes, no exact solution can be found within a reasonable time. Therefore, we reviewed heuristic approaches. These approaches often divide the problem into subproblems.

We also divide our problem. First, jobs are assigned to machines (mixing units), then the mixing units are scheduled, and storage is assigned. Rules for assigning jobs to mixing units have been found in literature. Scheduling strategies for mixing units include Just In Time (JIT) scheduling and Group Scheduling (GS). JIT scheduling leads to more efficient use of tanks by reducing buffer time by scheduling job as close to their due date as possible. GS reduces the number of changeovers by grouping jobs of similar product groups. GS and JIT can be conflicting scheduling strategies. The rules found for assigning jobs to mixing units are extended to assigning jobs to storage units.

We solve each problem completely before moving to the next subproblem. This has the advantage that each subproblem can be optimised and implemented separately. Rules from literature are used to assign jobs to mixing units, with the extension of taking into account rinsing if possible. We schedule the mixing units by first trying to find a solution that is feasible, i.e., no late jobs. If a feasible solution is found, we apply the JIT strategy and then the GS strategy. When applying the GS strategy, we also apply the JIT strategy within the product groups. The assignment of jobs to storage units is done according to the same principles as for the assignment of jobs to mixing units. We experimented with the 2 most impactful subproblems, scheduling mixing units and assigning jobs to storage, using data from practice. Recommended parameters settings were found with these experiments.

Table 7.1 summarises the results of the scheduling of the mixing units, applying the recommended settings. Applying the solution to the storage problem results in 5.15% less IBCs used and 1.14% more changeovers (based on the non optimised sequential evaluation) over approximately 1.5 month. The reduction in the number of IBCs used is more important than the increase in the number of changeovers.

Table 7.1 Summary of results: scheduling mixing units

	Earliness improvement	**Changeover improvement**
Minimum earliness	45.92%	0%
Minimum changeovers	33.79%	17.82%

Experiments for both subproblems show positive results. Combining the solutions should therefore also give a positive result, as scheduling closer to the due date of a job reduces overall storage time, allowing tanks to be used for more jobs, leading to less IBC usage. Therefore, we conclude that the scheduling strategy presented in this research can enable the mixing plant to use its resources more efficiently by providing decision support. Moreover, it can reduce manual scheduling time and can provide insights for plant optimisation.

7.2 Recommendations, limitations and scientific contributions

Based on our data analysis, we recommended Eurol to reconsider to use dedicated tanks. There is now only 1 dedicated tank left. Furthermore, we recommend Eurol to update its data model. Updating the data model consists of updating the structure, e.g., correctly linking jobs to filling jobs, and adding data, e.g., hot room data, disruptions, tank occupancy rate linked to filling jobs. Also, we recommend Eurol to create KPIs for the mixing plant and to automate the calculation of these KPIs. This enables Eurol to monitor the efficiency of the mixing plant in case of plant changes, application of the algorithm and modification of the parameters. We also recommend that Eurol quantify cost parameters so that other parameters can be updated, such as the IBC guidelines. This also allows the algorithm to support the optimisation of the plant. See Section 6.2 for recommendations regarding implementation.

The solution approach also has some assumptions/limitations. We assume that raw materials are always available because there is no data regarding the hot room and the hot room is not the bottleneck according to experts. Our data analysis shows that the hot room is able to provide earlier than expected. However, we cannot be sure that the hot room is able to provide earlier. The planner must take action accordingly if a job cannot be started due to hot raw materials not being available. If the algorithm is implemented with a user-friendly interface, it can provide decision support. We recommend that Eurol research the hot room so that either the assumption can be made without any adverse consequences or preventive measures can be taken, such as freezing the production order of the first part of the schedule. Finally, cost parameters are unknown. Therefore, we cannot guarantee which solution is best (changeovers versus IBCs used).

In our solution approach we apply the same problem division as Kudva, Elkamel, Penky, & Reklaitis (1994). We deviate from their solution approach by not scheduling jobs sequentially, i.e., we solve each subproblem completely before moving to the next subproblem instead of solving the subproblems per job. This, for example, makes the solution approach easier to implement and the solutions to subproblems easier to optimise by the planner, making decision support more effective as "what if" questions can be answered for each subproblem. Furthermore, our problem differs from that of Kudva, Elkamel, Penky, & Reklaitis (1994) in the objective function. Important in our objective function is the cost of the storage unit. This addition to the objective function creates, to the best of our knowledge, a problem that is new to literature. In our literature review we found that literature related to case studies in chemical engineering is neglected. Therefore, our research is an addition to a neglected topic. Finally, to the best of our knowledge, we have developed a new heuristic for the mixing unit scheduling problem presented in this research. This problem is also unique because there are some jobs with release dates. The heuristic has few parameters and is relatively easy to implement, which suits preferences of companies. Furthermore, we have extended the scheduling rules presented by Kudva, Elkamel, Penky, & Reklaitis (1994) for the assignment of jobs to mixing units to the assignment of jobs to tanks.

7.3 Suggestions for further research

The algorithm can be used to optimise the schedule of the mixing plant but can also be used for optimising the mixing plant itself. Further research can be carried out into the possibilities of optimising the plant and generating results for these possibilities.

Frequent data updates are required to apply the algorithm and exploit its full potential. This requires further research into Eurol's data model. Literature may be able to provide a generalised data model. If literature is not able to provide such a model, then the aim of further research could be to create a generalised data model based on a case study at Eurol. The full implementation of the algorithm may also be a topic for further research.

The hot room can be a topic for further research. Numerous questions arise regarding the hot room such as: how long must fluids remain in the hot room, in which containers can additives best be kept in the hot room, what is the impact of a lower temperature on the mixing time, how long does it take for fluids to cool down after having been in the hot room and what if they are isolated? All these questions and more can be topics for further research to shed light on the hot room.

The mixing time of new blends cannot be determined empirically. The first time a blend is made, the mixing time is estimated by experts. After the first mixing, the mixing time is calculated empirically. However, when it is only made once, the empirical calculation is not accurate. Therefore, it may be useful to add a safety margin to the mixing time of new blends. This safety margin can be a topic of further research, estimating the required margin and its effect on the schedule.

The filling plant at Eurol always empties the tank with the least number of litres first. It is not known whether this is always the best strategy. Further research can cover the impact of other strategies and whether these can be beneficial in some scenarios.

Quantifying cost parameters can be a topic of further research. This should allow for a better comparison of different solutions. Furthermore, it should lead to a better understanding of parameters such as IBC guidelines. Cost parameters are also important when using the algorithm for plant optimisation.

For example, since filing line 4 is very fast, it may have a greater chance of being further ahead on schedule. Therefore, it may be advantageous to increase the minimum buffer time of a fast line compared to a slow line. It may be that this can also be solved through frequent data updates if implementation reached this stage, this is preferable.

We assume that splitting is never advantageous. However, if the earliness for MU D is very high, it may be advantageous to split a job to MU A. further research can determine if and when splitting is advantageous. For this, quantified cost parameters are needed.

The IBC guidelines, i.e., the minimum litres before tanks are allowed to be filled per line, can presumably be optimised. The guidelines ensure that the cost of rinsing at the filling lines remains efficient. Because it may be more efficient to fill, e.g., 3 IBCs, than to fill a tank where it and its pipes must be rinsed. Currently, the guidelines are based on gut feelings of experts. In addition, the guidelines can be extended to also depend on the tank to be filled, as the length of the pipes differs for each combination of tank, mixing unit and filling line. For example, tank 44 is a large tank at a relatively long distance from the mixing units. Therefore, the mixing plant never fills anything below 5,000 litre in tank 44 (note tank 44 is no longer dedicated). A safety margin is also added to jobs to take into account rinsing, thus more is produced than necessary. Jobs that are assigned to IBCs may suffice with a lower safety margin.

Filling line 10 is capable of filling IBCs and there is a pipe connecting MU D directly to line 10. If MU D has to fill IBCs because there are no tanks available or customers order IBCs, it is possible that line 10 fills these IBCs when it is not working on a filling job. This may reduce the cost of IBC use at MU D and can be a topic for further research.

To increase the computation speed of the algorithm, it can run in parallel. Each CPU core can calculate a subproblem. Thereby solutions for each mixing unit at different buffer times can be generated in parallel. Subsequently, tank schedules can be calculated in parallel for each of the mixing units. Implementing parallel computing can be a topic of further research.

Bibliography

AIMMS. (2021, 1 25). *CPLEX documentation*. Opgehaald van documentation.aimms: https://documentation.aimms.com/platform/solvers/cplex.html

Amine, K. (2018). *Multiobjective simulated annealing: principles and algorithm variants.* Rabat: Hindawi.

Baker, K. R. (1974). *Introduction to sequencing and scheduling.* New York: Wiley.

Baker, K. R. (1999). *Heuristic Procedures for Scheduling Job Families with Setups and Due Dates.* Hanover.

Balasubramanian, J., & Grossmann, I. E. (2002). *A novel branch and bound algorithm for scheduling flowshop plants with uncertain processing times.* Elsevier.

Bean, J. C. (1994). *Genetic algorithms and random keys for sequencing and optimization.* ORSA journal on computing.

Belaid, R., T'kindt, V., & Esswein, C. (2012). *Scheduling batches in flowshop with limited buffers in the shampoo industry.* Elsevier.

Du, X., Li, Z., & Xiong, W. (2008). *Flexible Job Shop scheduling problem solving based on genetic algorithm with model constraints.* IEEE.

Fuchigami, H. Y., & Rangel, S. (2018). *A survey of case studies in production scheduling: Analysis and perspectives.* Elsevier.

Gedik, R., Kalathia, D., Egilmez, G., & Kirac, E. (2018). *A constraint programming approach for solving unrelated parallel machine scheduling problem.* Elsevier.

González-Neira, E. M., & Montoya-Torres, J. R. (2017). *A GRASP meta-heuristic for the hybrid flowshop scheduling problem.* Bogotá: Taylor & Francis Group.

Graham, R. L., Lawler, E. L., Lenstra, J. K., & Kan, A. R. (1979). *Optimization and approximation in deterministic sequencing and scheduling: a survey.* Elsevier.

Gupta, D., & Maravelias, C. T. (2016). *On deterministic online scheduling: Major considerations, paradoxes and remedies.* Elsevier.

Gupta, D., & Maravelias, C. T. (2019). *On the design of online production scheduling algorithms.* Elsevier.

Gupta, D., Maravelias, C. T., & Wassick, J. M. (2016). *From rescheduling to online scheduling.* Elsevier.

H., A., & Artiba, A. (2004). *Integrating simulation and optimization to schedule a hybrid flow shop with maintenance constraints.* Elsevier.

Hart, W. E., & Belew, R. K. (1991). *Optimising an Arbitrary Function is Hard for the Genetic Algorithm.* ICGA.

Haupt, R. (1989). *A survey of priority rule-based scheduling.* Springer.

He, Y., & & Hui, C. W. (2006). *Dynamic rule-based genetic algorithm for large-size single-stage batch scheduling.* Elsevier.

He, Y., & Hui, C. W. (2008). *A rule-based genetic algorithm for the scheduling of single-stage multi-product batch plants with parallel units.* Elsevier.

He, Y., Liang, Y., Liu, Z., & Hui, C. W. (2017). *Improved exact and meta-heuristic methods for minimizing makespan of large-size SMSP.* Elsevier.

Janiak, A., Kozan, E., Lichtenstein, M., & Oguz, C. (2007). *Metaheuristic approach to the hybrid flow shop scheduling problem with a cost-related criterion.* Hong Kong: Elsevier.

Karimi, I. A., & Reklaitis, G. V. (1985). *Deterministic variability analysis for intermediate storage in noncontinuous processes.* Wiley Online Library.

Keshavarz, T., Savelsbergh, M., & Salmasi, N. (2015). *A branch-and-bound algorithm for the single machine sequence-dependent group scheduling problem with earliness and tardiness penalties.* Elsevier.

Kontoghiorghes, E. J. (2005). *Handbook of parallel computing and statistics.* CRC Press.

Kopanos, G. M., Capón-García, E., Espuna, A., & Puigjaner, L. (2008). *Costs for rescheduling actions: a critical issue for reducing the gap between scheduling theory and practice.* ACS Publications.

Kudva, G., Elkamel, A., Penky, J., & Reklaitis, G. (1994). *Heuristic algorithm for scheduling batch and semi-continuous plants with production deadlines, intermediate storage limitations and equipment changeovers.* Elsevier.

Li, Z., & Ierapetritou, M. (2008). *Process scheduling under uncertainty: Review and challenges.* Elsevier.

Liang, Y., & Hui, C. W. (2016). *Simultaneous subtour elimination model for single-stage multiproduct parallel batch scheduling with sequence dependent changeovers.* Elsevier.

Mahnam, M., Moslehi, G., & Ghomi, S. M. (2013). *Single machine scheduling with unequal release times and idle insert for minimizing the sum of maximum earliness and tardiness.* Elsevier.

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

Méndez, C. A., Henning, G. P., & Cerdá, J. (2001). *An MILP continuous-time approach to short-term scheduling of resource-constrained multistage flowshop batch facilities.* Elsevier.

Monma, C. L., & Potts, C. N. (1989). *On the Complexity of Scheduling with Batch Setup Times.* INFORMS.

Murata, T., Ishibuchi, H., & Tanaka, H. (1996). *Genetic algorithms for flowshop scheduling problems.* Elsevier.

Nah, F. F.-H., Lau, J. L.-S., & Kuang, J. (2001). *Critical factors for successful implementation of enterprise systems.* Texas: Emerald.

Papadimitriou, C. H., & Kanellakis, P. C. (1980). *Flowshop Scheduling with Limited Temporary Storage.* Massachusetts: Journal of the Association for Computing Machinery.

Pathumnakul, S., & Egbelu, P. J. (2003). *Algorithm for minimizing weighted earliness penalty in single-machine problem.* Elsevier.

Potts, C. N., & Wassenhove, L. N. (1991). *Single Machine Tardiness Sequencing Heuristics.* London: Taylor & Francis.

Rajkumar, M., Asokan, P., Anilkumar, N., & Page, T. (2011). *A GRASP algorithm for flexible job-shop scheduling problem with limited resource constraints.* Taylor & Francis.

Randall, P. A., & & Kurz, M. E. (2007). *Effectiveness of Adaptive Crossover Procedures for a Genetic Algorithm to Schedule Unrelated Parallel Machines with Setups.* International Journal of Operational Research.

Reklaitis, G. V. (1995). *Scheduling approaches for the batch process industries.* Elsevier.

Resende, M. G., & Ribeiro, C. C. (2010). *Greedy randomized adaptive search procedures: Advances, hybridizations, and applications.* Boston: Springer.

Resende, M. G., & Ribeiro, C. C. (2016). *Optimization by GRASP.* Springer.

Rubén, R., & José, A. V.-R. (2009). *The hybrid flow shop scheduling problem.* Valencia: Elsevier.

Ruiz, R., & Concepción, M. (2006). *A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility.* Elsevier.

Sabuncuoglu, I., & Karabuk, S. (1999). *Rescheduling frequency in an FMS with uncertain processing times and unreliable machines.* Elsevier.

Shengchao, Z., Jianhui, X., Ni, D., & Yan, P. (2018). *A random-keys genetic algorithm for scheduling unrelated parallel batch processing machines with different capacities and arbitrary job sizes.* Elsevier.

Shi, B., Yan, L. X., & Wu, W. (2012). *Rule-based scheduling of single-stage multiproduct batch plants with parallel units.* ACS Publications.

Shim, S. O., & Kim, Y. D. (2007). *Minimizing total tardiness in an unrelated parallel-machine scheduling problem.* Tanfonline.

Sun, Y., Zhang, C., Gao, L., & Wang, X. (2010). *Multi-objective optimization algorithms for flow shop scheduling problem: a review and prospects.* London: Springer.

Victor, Y., Larisa, B., & Andrei, T. (2009). *Hybrid flowshop with unrelated machines, sequence-dependent setup time, availability constraints and limited buffers.* Elsevier.

Viswanath, V., Michael, G. M., Davis, G. B., & Fred, D. D. (2003). *User Acceptance of Information Technology: Toward a Unified View.* Minnesota: University of Minnesota.

Wang, K., Löhl, T., Stobbe, M., & Engell, S. (2000). *A genetic algorithm for online-scheduling of a multiproduct polymer batch plant.* Elsevier.

Wang, L., Zhang, L., & Zheng, D.-Z. (2006). *An effective hybrid genetic algorithm for flow shop scheduling with limited buffers.* Beijing: Elsevier.

Wu, D., & Ierapetritou, M. G. (2003). *Decomposition approaches for the efficient solution of short-term scheduling problems.* Elsevier.

Yazdani, M., Aleti, A., Khalili, S. M., & Jolai, F. (2017). *Optimizing the sum of maximum earliness and tardiness of the job shop scheduling problem.* Elsevier.

Yu, L., & Karimi, I. (2007). *Scheduling multistage batch plants with parallel units and no interstage storage.* Singapore: Elsevier.

Zhuge, J., & Ierapetritou, M. G. (2012). *Integration of scheduling and control with closed loop implementation.* ACS Publications.

Appendices

Appendix A: Data in-and outputs

This appendix shows for all files handled by the algorithm all headers relating to the required input data and the output data for each subproblem.

Input subproblem 1:
The input file for subproblem 1 is a single sheet containing all to be scheduled jobs with the following headings:

PO	Fluid	Rinsing group	Kilogram	Mixing time MU A	Mixing time MU B	Mixing time MU D	Due date	Release date	Density

Output subproblem 1:
The output file for subproblem 1 contains sheets for each mixing unit, each with the following headings:

PO	Fluid	Rinsing group	Litres	Mixing time	Due date	Release date

Input subproblem 2:
The input file for subproblem 2 contains sheets for each mixing unit, all with the following headings:

PO	Fluid	Rinsing group	Litres	Mixing time	Due Date	Release date	Minimum buffer time

Output subproblem 2:
The output file for subproblem 2 contains sheets for each mixing unit, all with the following headings:

Group	PO	Fluid	Rinsing group	Litres	Mixing time	Due Date	Release date	Earliness

Input subproblem 3:
The job input file for subproblem 3 contains sheets for each mixing unit, all with the following headings (note a job can contain at most 20 filling jobs):

PO	Completion time	Fluid	Rinsing group	Litres	Filling line filling PO 1	Litres filling PO 1	Filling time filling PO 1	Filling line filling PO 20	Litres filling PO 20	Filling time filling PO 20	Due date difference

Note the due date difference parameter can be changed for every job, that is why it is in the job input file.

The parameter input file for subproblem 3 contains 2 sheets, sheet 1 has the following headers:

Filling line	Family	Minimum IBCs

Sheet 1 indicates which line can connect to which tank family and the minimum number of IBCs per line, relating to the IBC guidelines.

Sheet 2 contains all tank information. The availability of a tank is indicated by a start time it is available and an end time. Each row indicates an available time slot for a tank, which means that there can be several rows for each tank. Sheet 2 has the following headings:

Tank	Minimum capacity	Maximum capacity	Dedicated	Fluid	Rinsing group	Start time	End time	Family	PO

Output subproblem 3:

The output file for subproblem 3 is a single sheet with the following headings:

Tank	Minimum capacity	Maximum capacity	Dedicated	Fluid	Rinsing group	Start time	End time	Family	PO	Litres

Jobs assigned to IBCs are assigned to tank -1 and family IBC.

Appendix B: Company survey

This appendix first shows all 17 statements by Table B.1 and the open questions. Thereafter, the participants' scores are shown by Table B.2 and the answers to open questions paraphrased are shown by Table B.3.

Table B.1 Survey statements

Statement [type.total]	English (EN)	Dutch (NL)
PE 1.01	The algorithm can provide good support when generating a schedule.	Het algoritme kan goede ondersteuning bieden bij het maken van een planning.
PE 1.02	By using the algorithm, the scheduling of the mixing plant can be improved and thus the mixing plant can produce more efficiently.	Door het algoritme te gebruiken kan de planning van de mengerij worden verbeterd en dus de mengerij efficiënter produceren.
PE 1.03	The algorithm can help in the design of the production facility/plant.	Het algoritme kan helpen bij het ontwerpen van de productie-installatie/fabriek.
EE 2.04	The solutions (schedules) of the algorithm are easy to use in practice.	De oplossingen (planningen) van het algoritme zijn makkelijk te gebruiken in praktijk.
EE 2.05	The options (parameters: dd, bt, b) offered by the algorithm are easy to use.	De opties (parameters: dd, bt, b) die het algoritme bied zijn gemakkelijk mee om te gaan.
EE 2.06	Using the algorithm takes a lot of time (e.g., due to data input).	Het gebruiken van het algoritme kost veel tijd (bijv. door data input).
SI 3.07	People who influence my behaviour think it is good to use the algorithm.	Mensen die mijn gedrag beïnvloeden denken dat het goed is het algoritme te gebruiken.
SI 3.08	I think that the application of the algorithm can expect support within the company.	Ik denk dat het toepassen van het algoritme binnen de organisatie steun kan verwachten.
FC 4.09	The algorithm fits well with how the planning department works or wants to work.	Het algoritme past goed bij hoe het bedrijfsbureau werkt of wil werken.
FC 4.10	I have all the resources needed to use the algorithm.	Ik heb alle middelen die nodig zijn om het algoritme te gebruiken.
FC 4.11	I have all the knowledge needed to use the algorithm.	Ik heb alle kennis die nodig is om het algoritme te gebruiken.
FC 4.12	a specific person or group is available for assistance with the use of the algorithm.	Een specifiek persoon of groep is bereikbaar voor hulp m.b.t. het gebruik van het algoritme.
FC 4.13	I intend to use the implementation recommendations because they for a good basis for an implementation plan/strategy.	Ik heb de intentie gebruik te maken van de implementatie aanbevelingen omdat deze een goede basis vormen voor een implementatie plan/strategie.
BI 5.14	I think that applying the algorithm is a good idea.	Ik denk dat het toepassen van het algoritme een goed idee is.
BI 5.15	I am looking forward to using the algorithm.	Ik kijk er naar uit om het algoritme te gebruiken.

BI 5.16	The proposed schedules process is logical and efficient in achieving the goal (increasing mixing plant efficiency).	Het voorgestelde planningsproces is logisch en efficiënt in het bereiken van het doel (verhogen efficiëntie mengerij).
BI 5.17	I intend to make use of the conclusions and recommendations of this study.	Ik heb de intentie gebruik te maken van de conclusies en aanbevelingen van dit onderzoek.

Open questions (EN, *NL)*:

1. Did the research lead to new insights in existing processes or for future process improvements?
   Heeft het onderzoek tot nieuwe inzichten geleid in bestaande processen of voor toekomstige procesverbeteringen?
2. Do you have any expectations or questions from/for the research that have not been met/answered by the research, if so which (please note the scope of the research)?
   Heb je verwachtingen of vragen van/voor het onderzoek die niet vervuld/beantwoord zijn door het onderzoek, zo ja welke (let op de scope van het onderzoek)?
3. Other remarks
   Overige opmerkingen

Table B.2 Scores of the participants to the statements

Statement [type . total]	Manager Planning	Planner	Manager Mixing Plant
PE 1.01	4	5	4
PE 1.02	5	5	3
PE 1.03	3	4	3
EE 2.04	4	3	2
EE 2.05	4	3	2
EE 2.06	5	3	4
SI 3.07	3	4	3
SI 3.08	4	4	2
FC 4.09	4	4	3
FC 4.10	2	2	2
FC 4.11	3	3	1
FC 4.12	2	1	4
FC 4.13	4	4	4
BI 5.14	5	4	4
BI 5.15	4	3	3
BI 5.16	4	5	4
BI 5.17	5	3	4

Table B.3 Paraphrased answers of the participants to open questions

Question	Manager planning	Planner	Manager mixing plant
1	Yes. I had expected a larger focus on the mixing part. However, it was shown that most is to be gained in the assignment of tanks.	Yes. I think that the algorithm can improve the scheduling process when data is available.	This is difficult to answer as I do not know the research in detail. I do think it leads to new insights

			regarding tank scheduling.
2	I had hoped (besides it being fair) that implementation would have been further along (but looking back I think that this is due to us not having the data)	I had hoped that implementation would have been further along. However, due to data not being available this is impossible.	I had hoped that rinsing would have been considered in more detail, e.g., manifold rinsing.
3	Implementation is a nice goal for 2022.		I believe to have too little knowledge of the research to properly answer the questions.