Solving an inventory-bound multi-item single-machine scheduling problem at Shell Chemicals Europe b.v.

Author:

W.J. Haan

Supervisors:
dr. E. Topan
dr. ir. J.M.J. Schutten
ir. M. Marelli

A thesis submitted to the Faculty of Behavioural, Management and Social Sciences (BMS) and in partial fulfilment of the requirements for the degree of

MSc Industrial Engineering and Management

Faculty of Behavioural, Management and Social Sciences (BMS)

University of Twente

September 22, 2023

ACKNOWLEDGEMENTS

This report marks the end of my Master Industrial Engineering and Management at the University of Twente, and thus my time as a student. It is the result of many months of hard work that would not have been possible without support from people around me. First and foremost I am grateful to Shell for the opportunity to conduct my thesis research during my internship there. I would like to thank Marco Marelli for his support, guidance, and enthusiasm throughout the project, not only helping me finish my education but also significantly helping me develop my personal and professional skills. His involvement played a significant part in making me enthusiastic to continue working for Shell in the future. I would also like to thank all the other Shellers that helped me throughout the project for their time, openness, and trust.

Furthermore I would like to thank my University of Twente supervisors Engin Topan and Marco Schutten for their feedback and guidance throughout the project, helping me shape the research and push the outcomes to a higher level. After every meeting I walked away with a clear sense of direction and improvements to make, as well as helpful suggestions on implementation of these improvements.

Finally, I would like to thank my parents and my two brothers for their never ending support. I would not be where I am today without them.

Wytze Haan,

Rotterdam, September 2023

Disclaimer: This research originally uses confidential information. All values in the report have been altered to maintain that confidentiality and do not represent true values for Shell Chemicals Europe b.v.

MANAGEMENT SUMMARY

We conduct this research at Shell Chemicals Europe, specifically at the production site Energy and Chemicals Park Rotterdam. We focus on optimizing the production, transportation, and sales process for a single production unit located at the very end of a value chain.

Problem Definition

Each month, Shell decides the production volumes for that month based on the expected demand and a larger global optimization. The supply team is then responsible for meeting the production volume. Production is a heavily constrained process due to factors such as limited tank storage, barge transportation, and product quality control. Mistakes in the planning process that lead to demurrage, production downtime or missing orders are extremely costly. Inventory management is key in avoiding these costs; high inventory reduces the risk of on-site demurrage or missing orders, but increases the risk of production downtime or off-site demurrage as well as increasing inventory costs. In this research, we aim to develop a method for the optimization of the entire month's production schedule, including transportation and order fulfillment with the goal of minimizing inventory and transportation costs.

Solution Methodology

We explore options for the optimization approach. Shell must be able to implement the optimization within their current ways of working, meaning that it must be possible to integrate within the software they use. It also has to be easily maintainable and adaptable to changing contexts. Ideally, the optimization method should also allow for extension to other production units. As this is the first time a scientific optimization of the scheduling process takes place, we would also like to use a methodology that can guarantee optimality. We decide on using mathematical programming in the form of a mixed integer linear program (MIP). MIP allows us to fully express the system behavior, is easily maintainable and implementable within Shell's software, and depending on the modelling can be extended to other production units. The downside of a MIP approach is the risk of extremely long runtimes and general difficulty of scaling to larger problem instances.

Modelling

The petrochemical context leads to modelling in the field of process systems engineering, allowing us to more easily deal with continuous production of a liquid chemical in a system where jobs are not explicitly defined. As is done in the wider petrochemical industry, we use the Resource-Task Network model, which allows us to model all constraints in the system at a high level of detail. As there is still a risk of high runtimes, we also explore three simplifications of the fully detailed model. One where we remove waste product from the model, one where we remove the option of moving product around on-site, and one where we rewrite constraints to remove auxiliary variables from the model. In the experimentation, we determine the impact on performance of these simplification.

Results

The full model does indeed run into issues on scalability, and given a runtime of 2 hours, is only able to generate solutions for a 7 day schedule as opposed to the desired 31 days. Extending the runtime does allow for larger schedules, but the runtime increases exponentially. We have the full model performance for a 14 day schedule, taking around 68 hours to generate, and compare this to the schedule made by a human supply planner. We take historic data as input, and find that the model on average has an objective value around 4.2% lower than that of the supple planner. This would correspond to yearly savings of around €60,000. However, we note that the model does not take into account all factors that a supply planner does, mostly to do with interactions in limited resources shared with other production units. For example, a supply planner may adjust production based on maintenance preferences from the operators or the barge scheduling based on limited jetty availability.

When it comes to the simplifications, we find that they scale better than the full model. As a result, they are able to find solutions with lower objective values given a limited runtime. The model including all three simplifications has an average integrality gap of 3% when generating a schedule for 6 days, whereas the full model has an average integrality gap of 6%. Interesting to note is that although the simplifications allow for less flexible scheduling, they still perform well in terms of objective value. This implies that the level of detail in the models may not be required. For example, the model that does not incorporate the waste product instead uses a minimum batch size constraint. This is a heuristic commonly used by supply planners to generate schedules, and does not seem to negatively impact performance on objective function. Although these simplifications improve performance, they do not achieve the desired level of scalability to allow for 31 day schedules to be generated within a reasonable amount of time.

Conclusions and Recommendations

The general conclusion is that the level of detail within the models is too high and negatively impacts scalability. The system appears to be largely feasibility driven, and making simplifications through the use of planning heuristics appears to be a beneficial tool. We recommend reconstructing the model through the use of these heuristics to greatly reduce the number of decision variables and overall complexity. This should allow for the generation of near optimal schedules within reasonable runtimes.

Overall, the full model does construct feasible solutions and accurately reflects the system. Although its use in generating schedules on a regular basis is limited by its runtime, it can be used to evaluate the impact of changes in the system. It should effectively allow us to simulate the impact of increasing tank sizes, adding or removing products and other changes that may be of interest to Shell. By analyzing the impact on the values of the objective function given different parameters, we should be able to conclude if certain investments would pay off in the long term.

We find that a Resource-Task Network allows us to express the behavior of a petrochemical production process, but is unable to be scaled to the desired level. The formulation leads to redundant auxiliary variables that increase the runtime, and we advise a critical approach in the necessity of each variable as well as the overall necessity for Resource-Task Network formulations.

CONTENTS

1 INTRODUCTION

We conduct this research at Shell Chemicals Europe as part of the master graduation assignment of the Industrial Engineering and Management educational master's degree program at the University of Twente. We start this chapter by introducing Shell Chemicals Europe in Section 1.1, then describe the research motivation in Section 1.2, elaborate on the research design in Section 1.3 and finally present the research questions in Section 1.4.

1.1 About Shell Chemicals Europe b.v.

1.1.1 Shell Chemicals Europe b.v.

Shell Chemicals Europe b.v. (SCE) has been supplying business and industry with petrochemicals for over 90 years. They manufacture and supply a wide portfolio of bulk chemicals in the European, Middle Eastern, and African region, where they are then used for a wide range of applications such as making healthcare products, textiles, furniture, cleaning supplies, and much more. SCE is positioned at the end of a long integrated value chain managed by the larger Shell plc, in which crude oil is supplied to refineries such as the Energy and Chemicals Park Rotterdam in Pernis. Here, refinery and cracking operations transform the crude oil into a wide range of products, including petrochemicals. Conversion to the chemical products is one of the very last steps in the entire chain and makes up a relatively small volume of product flowing through the refinery. However, it is vital for SCE to keep their production units running, as different links further up in the chain rely on SCE to use the product they produce. If SCE does not use these products, storage tanks in the chain fill up and cause production to be halted as a result. Depending on the units affected, these costs can go up to millions of Euros per day.

The SCE portfolio consists of 19 different product families, each with unique supply chain challenges and production processes. We conduct this research within one of these product families, which has a total production quantity in the order of magnitude of 100 metric kilotons per year. Within this product family there are a multitude of different liquid products, referred to as 'grades', that are be made from the same general set of raw materials, referred to as 'feedstock'. The product family has several production units, each capable of producing a unique set of grades. Decisions about production quantities and scheduling for these units and their respective grades are made on a monthly basis. Figure 1.1 gives an impression of what a production unit looks like, as well as the tanks that are used to store the finished products.

1.1.2 Production Process

Each month, production quantities are decided while both the exact demand and price of these products is uncertain. An agreement, called the handshake, is made on how much of each grade is to be produced and sold. This handshake is proposed by an optimization based on expected demand, profit margins, as well as inventory levels and production capacity. The supply team then decides how to schedule the production of these products within that month; this

Figure 1.1: Example of petrochemical production unit and storage tanks

entails the day-to-day scheduling. Here, there are many complicating factors that influence the feasibility of the production schedule, which include transportation and customer pick-ups. This project is carried out within the supply team and focuses on this day-to-day scheduling based on the handshake production figures.

The single most crucial aspect of scheduling is the management of inventory. Product is liquid and stored in dedicated tanks. The empty space in a tank, referred to as 'ullage', indicates how much volume can still be deposited into the tank before it is full. If there is not enough ullage in a tank, production must be halted. Stopping production leads to significant costs and problems further upstream in the production process. This stimulates keeping inventory low. On the other hand, inventory must also be kept high enough to have product available to meet an uncertain amount of demand, stipulated by minimum service levels. Not holding enough inventory may cause lost sales, penalties, and loss of market share. A feasible production schedule that maintains optimal inventory levels throughout the month is thus important to Shell Chemicals Europe. They currently rely on historical data and the experience of supply planners to make a schedule that meets demand while minimizing inventory.

Scheduling is complicated by the fact that some of the storage tanks are located at an off-site location, and transporting product there requires a barge. This barge has to be scheduled along with the production, as the supply planner must make sure there is enough product on-site for the barge to pick up, and enough ullage in the off-site tanks so that the barge can deposit all the product upon arrival. So, alongside production scheduling, the transport must be scheduled as well. Having off-site storage also impacts the sales of the products. It means that some orders must be loaded from off-site tanks whereas others from on-site tanks, depending on where product is available. This means that the supply planner must also take into account the orders and forecast demand when making the schedule.

1.2 Research Motivation

Production, transportation, and sales need to be scheduled in line with each other in order for the system to run smoothly. When things do not go to plan, significant costs can be incurred as a result. With regards to production the largest risk is that it needs to be halted due to a lack of ullage in on-site tanks, costing around €100,000 per day. For transportation, the most significant costs are costs of barges and waiting costs. Barges are expensive to use and should be used as effectively as possible. Besides this, if a barge has to wait to (off-)load product due to ullage or product constraints, a penalty cost of around €10,000 per day is incurred. On the sales side, if trucks from a customer come to pick up product, but cannot do so due to low stocks, a penalty cost of around €2,000 is incurred. If it happens often, market share might also be lost, and cost tens of thousands of Euros to gain back. This means there is a large incentive to ensure a schedule that does not cause additional costs to be incurred, while guaranteeing supply to customers as well as an uninterrupted production. All these costs can be managed by keeping the right levels of inventory throughout the month; high enough to be able to meet demand, but low enough to ensure enough ullage for production and off-loading.

There is also the final relevant cost in the form of cost of capital. The grades produced are done so in very large volumes and have a high price per kilogram; requiring significant capital investments to produce. Until the product has been sold, shipped, and paid for, this capital is frozen and cannot be leveraged elsewhere. Shell uses a cost of capital of around 1% per month, which can be seen as the holding cost of inventory. With millions of Euros of product in inventory, this cost runs into the tens of thousands of Euros per month. If the schedule can be made in such a way that the inventory can be brought down even by a few percent, this would provide significant economical savings.

1.2.1 Core Problem

Shell believes the costs associated with production, transportation and sales can be reduced through optimized scheduling. To create a well-functioning schedule, the highest priority lies on creating a schedule that is feasible and meets service levels. A majority of time is spent on finding a schedule that meets these constraints, adapting it to changing situations, and managing the supply chain. This leaves little space and time to explore better schedules that may reduce costs. Overall, there is an excess of complications for the supply planner to take into account, meaning that there is no optimization of the production and transportation schedule with regards to cost. This leads us to our core problem: there is a lack of an integrated scheduling method capable of including production, transportation, and sales. Shell Chemicals Europe is facing an optimization problem where they want to schedule production operations to minimize associated costs while maintaining service levels.

1.3 Research Design

1.3.1 Research Goal

The goal of the research is to find and develop an optimization method that allows Shell Chemicals Europe to minimize the costs between production and customer pick-up while maintaining service levels. Costs consist of inventory costs, in the form of cost of capital, as well as transportation costs for barges, and penalty costs. Shell's focus lies on a reduction of inventory costs, but transportation costs may increase as a result and are thus also part of the optimization. The method must be able to decide what grade to produce at what time, when to move product to off-site tanks, as well as when and from which tank to fulfill customer demand. It should be able to effectively incorporate the complexities in production, transportation, and sales to meet service levels at minimum costs.

1.3.2 Scope

We limit the scope of this research to a single production unit, which we do not specify due to confidentiality reasons. We choose a single production unit due to the fact that there are significant differences between the production units employed by Shell. These differences make it difficult to develop a method that would be able to optimize multiple units within the product family. Factors include continuous vs batch production, batch size constraints, strict sequencing constraints and also unit inter-dependencies. We choose a unit with continuous production that is responsible for one of the major products sold within the family, as it is not dependent on other production units. This means that there is quite a bit of flexibility regarding the scheduling of this unit, and as a result, large estimated potential room for optimization.

Every month, SCE runs a wider optimization to decide on overall production quantities for every month called the handshake. We limit our scope to the in-month scheduling to meet the handshake, and not the handshake itself. This is because the decisions made on a monthly basis are made through an optimization that also takes into account the complete value chain and economic context. Producing significantly more, or less, than is planned will lead to issues for the wider value chain and is generally not considered acceptable.

1.3.3 Approach

With the goal of finding and further developing an optimization method that minimizes logistical costs, a concrete approach can be defined. In Chapter 2, we analyze the current situation. We set up a program of requirements to assess what requirements are relevant for the optimization method, both in terms of output and ability to apply the method in practice. These requirements form a basis to enable decision making during research and development. We also research the production, transportation, and sales processes to create a complete overview of the system. This is required for us to create a conceptual model that is capable of representing the system.

Next, the conceptual model needs to be translated into a mathematical model which can be optimized. This model can be formulated in various ways, and this will also influence what solution method can be used to optimize the model. For this, we conduct literature research in Chapter 3 and select the most appropriate method in Chapter 4. Based on this we construct the mathematical model and implement it into a solving tool to attempt to find an optimal solution. We then validate the results of the tool based on historical production data and assess the performance of the tool in Chapter 5. After that, we identify areas for improvement and further research, as well as make recommendations to Shell with regards to the current process in Chapter 6.

1.4 Research Questions

Based on the goal and approach, the main research question to be answered is;

"How can Shell Chemicals Europe optimize the daily production, transportation, and sales scheduling of a single production unit to minimize inventory and transportation costs while maintaining service levels?"

In order to fully solve this main research question, we structure our investigation into 4 smaller research questions:

RQ 1: ”What does the current situation of the Shell process look like?”

RQ 2: ”What techniques can we use to optimize the Shell process?”

RQ 3: ”How can we apply these modelling techniques to the Shell case and adapt it to improve performance?”

RQ 4: ”How does the model perform and how do the adaptations influence performance?”

We answer research question 1 in the context analysis in Chapter 2. For the second research question, we conduct a literature review in Chapter 3. We then answer the third research question in Chapter 4, and investigate the performance for the fourth research question in Chapter 5. Finally, we present our conclusions and recommendations in Chapter 6.

2 CONTEXT ANALYSIS

This chapter focuses on giving a description of the relevant elements for the Shell system to answer the first research question "What does the current situation of the Shell process look like?". In Section 2.1 we start with a broad overview of the complete process and separate it into 3 main stages: production, transportation, and sales. We then dive deeper with a complete technical description of the process in Section 2.2, again following the 3 main stages of the process: production, transportation, and sales. In Section 2.3 we elaborate on the costs involved in the system, after which we stipulate the requirements for the optimization approach in Section 2.4. We then outline the characteristics of Shell's optimization problem in Section 2.5 and finish by fitting a theoretical demand distribution to Shell's demand data in Section 2.6.

2.1 Current Situation

All activities within this process are integrated and impact each other. The SCE supply team views the entire system as a single production scheduling problem, where other aspects are simply necessary actions to make production feasible and fulfill demand. Based on information from supply planners and process experts, we create a schematic overview in Figure 2.1. The process takes place in 2 different locations, which are referred to as on-site and off-site activities. On-site means all processes occurring on the Shell production site. Off-site indicates external locations, which contain extra tanks for intermediate storage. Once product has been loaded for shipping to a customer, it is considered out of scope for this project.

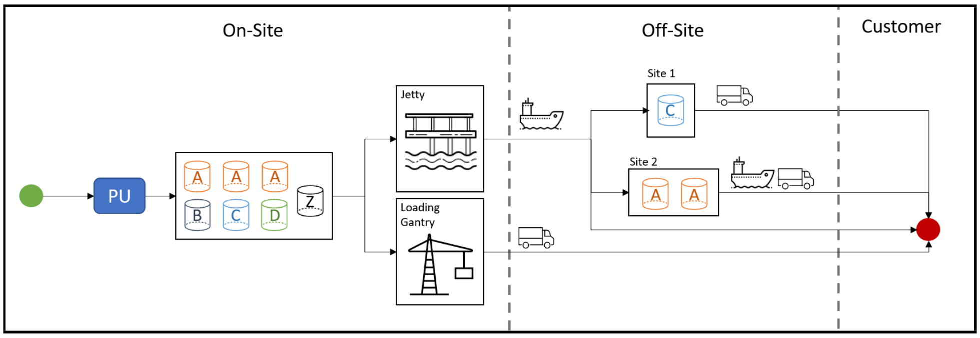

Figure 2.1: Schematic overview of the full process

We split the process into 3 separate stages: production, transportation, and sales. Production entails the manufacturing of the grades and on-site storage, transportation covers transporting product to the off-site storage location via a barge, and sales is the process of loading trucks or barges to ship product to customers.

2.1.1 Production

The process starts on-site at the production unit (PU), which produces 1 of 4 grades; *A, B, C* or *D*. In some cases, waste product *Z* is also generated. The production unit is connected to the storage tanks via pipelines, and operators can switch to which tank the product flows before production. Production is a continuous process, meaning that the product flows directly into the tank and there is no intermediate storage within the production unit itself. As a consequence, if the tanks are full, production must be stopped. On-site storage is thus considered as part of the production process. Each grade is stored in dedicated tanks, where it is possible to move product between tanks of the same grade on-site. Tank grades are generally not changed as it requires costly and time consuming cleaning activities. As can be seen in Figure 2.1, there are 3 tanks for product *A* and a single tank for each *B, C, D*, and waste product *Z*.

2.1.2 Transportation

Due to limited on-site tank capacity, there are also off-site tanks. These are used, in part, to ensure that there is always enough space in the on-site tanks to ensure that production does not need to be halted. Transportation to off-site storage is always done via barges, which need to be arranged via haulers at least a week in advance. Jetties are used for barges to moor while loading product on or off the barge. There are multiple on-site jetties, all of which are heavily used. This means there are limited slots available per month for barges for our product family. If the product is headed for off-site storage, it travels there and unloads the product via another jetty into the dedicated tank. The product then remains in that tank until it is sold. The off-site loading gantry can be considered to have unlimited slots for loading. Once product has been loaded for a sale, the process is considered finished. Product is moved to off-site tanks in case of on-site capacity constraints, meaning that product is also sold directly from on-site tanks and never goes through transportation as described here.

2.1.3 Sales

Shell Chemicals Europe operates within a commodity based industry, where the demand and sales process behaves differently than to specialized goods or fast moving consumer goods. The key difference is that the pricing of products is determined on a monthly basis in negotiation with customers. Demand is heavily influenced by market conditions as well as competitor pricing, where a competitor undercutting will cause a drop in demand for SCE. Similarly, SCE is able to somewhat steer demand through their own pricing strategy. This makes it possible for the sales department to provide a demand forecast each month. Although these can vary significantly between months, the forecast error is thought to be relatively low for the in-month demand.

With regard to the actual demand coming in, customers can place orders for a specific product 1 to 4 weeks in advance, specifying a pick-up date in agreement with Shell. The customer then sends a truck or barge to come pick-up the product and take the agreed upon volume directly from the tank in which it is stored. Shell must then have enough product available at that time and place to be able to fulfill that order. Shell refers to the percentage of orders that are fulfilled on time in a month as the service level. We use the same terminology and definition in this report. It is imperative for Shell that it can maintain its service levels to customers. Important to note is that the 4 grades produced are referred to as reactor grades, meaning that they are the pure product that flows from the production unit. Customers may also require blended grades, which are reactor grades mixed with additional chemicals to meet specific requirements. This blending is done while the product is loaded onto a truck or vessel, and can only be done on-site. So if a customer orders a blended grade, this order will always be fulfilled from on-site tanks. If it is a pure variant, Shell specifies if pick-up is from on-site or off-site tanks. Trucks picking up product do this via the loading gantry, which has limited slots available per day.

2.2 Technical Description

To give a better understanding of the complexities within the scheduling, we present an overview here. This will give an idea of the constraints that are in place within the system and how they influence the feasibility of a production schedule.

2.2.1 Production

Production complexities form constraints that determine the feasibility of the production process. If these complexities are not addressed, it simply is not possible to continue operating and production must be halted.

Sequencing

Only 1 product can be produced at a time, and it is not possible to switch between all products immediately after each other. Whenever switching from grade *A* to another grade, or vice versa, cross-contamination of product occurs due to residual product in the pipeline. This means that the first fixed volume of product pumped through it will be contaminated, and this is stored as grade *Z* in a dedicated tank. *Z* can be blended away into product A afterwards, where the grade A may consist of at most 5% product Z. Products *B*, *C*, and *D* belong to the same group; switching between them creates no waste product. However, due to their chemical composition, if these grades are produced without producing *A* in between, they must be produced in alphabetical order. So *BDA* would be a valid production sequence, but *DBA* would not be a valid production sequence.

Ullage

Before starting production, the on-site tanks connected to the PU have to have enough ullage. Production is not possible if there is no space to deposit the grades as they are being produced. Ullage is created when product moves out of the tank, either in an on-site transfer, for a sale, or when transported to off-site storage. An off-site movement, known as a Stock Transfer Order (STO), is a single large volume shipment, whereas sales are much more frequent smaller volume shipments. The benefit of an STO is that it can be planned and controlled by the supply planner, but requires scheduling ahead of time. Due to STOs being large volume to keep freight prices down, the tank needs to contain this large volume and is nearing maximum capacity. This leaves little ullage in the tank the barge arrives. A delayed barge may cause the tank to be filled completely and a lack of ullage as a result, which poses a production risk as a result.

Downtime

There may be unexpected downtime during production or transportation, which can last from a few hours to a few days. In general, a downtime of 1 day per month is used as a rule of thumb within Shell. This is important as enough product needs to be available during downtime to still meet demand. This also means we need to minimize other sources of downtime to be able to maximize production throughout the month.

2.2.2 Transportation

Just because an amount of product is on the balance sheets, this does not mean the product can actually be sold. For example, an amount of product may be in transit from on-site tanks to off-site tanks. This volume of this product is part of the inventory, but cannot be sold as a customer cannot come and pick it up. This is what is referred to as product availability: it entails whether the inventory in a tank is truly able to be picked up by a vehicle.

Testing

Each grade must meet certain specifications with regards to chemical composition. Product that meets requirements is referred to as on-spec, product that does not is called off-spec. Meeting the specifications is a contractual obligation, and Shell tests the product extensively to ensure customers receive on-spec product. After the production run of a grade is completed, a sample is taken from the tank into which it was produced. This sample is tested on-site. Until testing confirms that the product is on-spec, no product may be loaded onto vehicles. The tank is essentially closed once production starts, and the product stored within it is not available to be lifted until production is finished and the grade has been tested. When product is moved off-site in an STO, there is a risk of cross-contamination if the vehicles or pipes were not properly cleaned. As such, testing is also done when a product has been deposited into an off-site tank. This requires sending the sample back to the production site, meaning that testing an off-site tank takes 1-2 days, whereas for on-site tanks it is a matter of a few hours.

Gantry & Jetty Restrictions

Just because a tank's product is tested and on-spec, this does necessarily mean it can be loaded onto a vehicle. Loading occurs via the gantry or jetty, and the ones on-site have a limited capacity available. So even if an immense inventory is present on-site, the gantry and jetty induce a hard cap on the amount of product that is available.

Barge Travelling Time

Barges are slow moving vehicles that take significant amounts of time to travel between the on-site and off-site location. This means that product that is loaded onto a barge will not be available for fulfilling orders for some time, and that there needs to be product either on-site or off-site to cover for this period. This poses a challenge as barges transport large volumes, meaning they leave the tanks on-site quite low after picking up product, and require tanks off-site to be mostly empty such that the product can be deposited on arrival.

2.2.3 Sales

Blends

The unit produces 4 unique grades, also known as reactor grades, which can be sold directly to customers in their pure form. However, some customers require slightly different chemical properties. To achieve this, Shell also offers products known as blends. These grades are made by blending the reactor grade with other products in a specific ratio. This blending is done in the pipelines when the grade is being loaded; blends are never stored in tanks. However, the blending is only possible on-site. Any product stored off-site must thus be sold in its pure form, and demand for blends can only be satisfied from the on-site tanks. For this research, we translate demand for blends back into demand for reactor grades. For example, 1 unit of a blended grade *X1* contains 0.3 units of grade *A*. So if a customer orders 1 unit of *X1*, this is reflected as an on-site demand of 0.3 units of *A*. As blending occurs in the pipelines during the loading process, we do not need to address it explicitly as part of the schedule.

Monthly Demand Volume

In most cases, customers place their orders 2-3 weeks before the desired pick-up date. However, some may be placed as little as a week ahead. As the planning is made for an entire month, not all demand is known yet at the time of making the schedule. Shell forecasts based on historical data and information from the sales team but, due to a very competitive market, the exact monthly volume sold can vary 10-20% from the forecast. The supply planners needs to take this flexibility into account, as it may become desirable to increase or decrease production for certain grades during the month based on how sales progress. Of special importance is being able to fulfil sales in case of higher demand, as otherwise these extra sales will go to competitors and Shell misses out on profits. This is why service levels are in place and must be maintained.

Lifting Pattern

For the orders that have been placed, it is known when the customer is coming to pick-up the product. This means the supply planner knows what volume of each grade needs to be available during each day to meet this demand. However, this is not known for the uncertain quantity of orders that still need to be placed. There is some room to move pick-up dates around, 1-2 days earlier or later is generally possible if the date stays within the same work week. The product availability required is thus not known exactly, making the trade-off between production and product availability difficult.

2.3 System Costs

There are significant costs associated with executing the system, and we give an overview of these costs below.

2.3.1 Holding Costs

Holding costs are the costs associated with holding product in inventory over a certain period of time. For Shell, the main source of these costs are reflected by the capital that is effectively frozen in the value of the inventory. Until the product is sold, that value cannot be used for other investments or to generate profits. This is known as the cost of capital, and for Shell this is around 1% per month. There are also fixed costs for the tanks used to store the product, but these are not taken as part of the optimization due to not being a variable that can be changed. There are technically also some variable costs associated with volume stored in off-site tanks, but these are relatively low compared to the cost of capital and we consider them out of scope. So, the holding costs are taken as 1% of the product value per month.

Demurrage

When a barge is scheduled, it is given a timeslot in which it can load/unload its product. It may happen that when the barge arrives for its timeslot, there is not enough product or ullage available to load or unload. As a result, the barge would have to wait until the product or ullage becomes available. In general, the barge then has to wait a day or more. In this case, Shell has to pay a penalty to the barge owner per day the barge has to wait. This cost is about €10,000 per day.

Transportation Costs

Using barges is not free: Shell pays a fixed cost to the owner per trip, as well as a variable cost depending on the volume of product shipped. These prices vary depending on the barge and time, but are indicated by a fixed cost of €1,000 and variable cost of €10 per metric tons (mt) of product.

Lost Sales

There may be more demand than Shell can supply, in which case customers will not be able to order all the product they want. In this case, Shell loses out on the profits of those sales. These figures are confidential and in this report we use an indication based on publicly available information. If we use the average petrochemical price of April 2023 according to the Platts Global Petrochemical Index (€1201/mt), and a return on investment of 5%, we can estimate a profit of €60/mt. So the cost of losing a sale is estimated at €60/mt.

2.4 Program of Requirements

For the optimization to add value and be applicable in practice, there are certain elements that we must incorporate. This section aims to define the requirements Shell has for an optimization to be of value.

2.4.1 Model Output Feasibility

The most important requirement for the optimization is the feasibility of the output; it must produce a schedule that is feasible and can be carried out in reality. This means that none of the production, storage or transportation constraints can be violated. Besides this, it must meet predetermined service levels, be in line with the agreed upon production level (handshake) and result in month-end inventory levels that ensures stability for the scheduling of following months. The output itself must be a daily production schedule that is generated based on the information available at the start of the production month.

2.4.2 Model Validity

To be able to construct a working model, simplifications need to be made due to reasons such as time constraints or data availability. Some areas, however, cannot be ignored and must be a part of the model. Primarily, the uncertainties in demand and the lifting pattern must be included. This is the most challenging part of the scheduling and answering this is the main value driver for Shell. The model should also address the other complexities described in Section 2.2. Besides this area of validity, the general method of implementation should also allow for it to be extendable to other production units in Shell. This means that modelling choices made should not excessively limit extendability; for example the model should also be extendable to batch-production units without having to construct a fundamentally different model.

2.4.3 Usability

As Shell is looking to use the optimization on a regular basis, the ease of generating the output is also very important. This has to do with the usability of the method; allowing supply planners to effectively create schedules. Primarily, it is important that users are able to adapt parameters to allow optimization when circumstances change. For example, planners may want to indicate planned downtime, changes in storage tanks, or changes in costs. Besides this, generating the schedule should not take a significant portion of time; it should enable supply planners to run multiple scenarios to see how tactical decisions may influence the scheduling or service levels. For example to find out when to best put a tank in maintenance. We maintain a target of around 15 minutes to generate a single production schedule.

Another aspect of usability is the technical knowledge required to use the optimization method and its implementation. Running the optimization should take minimal technical knowledge, and it should allow anyone with a basic understanding of programming and optimization to maintain the implementation. This translates into the implementation having to fit within software currently used by Shell Chemicals Europe, which include solvers capable of tackling mathematical programs, data analysis and visualisation software, as well as enterprise resource planning systems.

To conclude, the optimization must result in a feasible production schedule that takes into account demand uncertainty and lifting patterns. It should allow supply planners to easily run the optimization to create schedules, with the optimization taking less than 15 minutes. Maintenance of the optimization implementation should require minimal technical knowledge, and ideally, the underlying model should be extendable to other Shell production units.

2.5 Optimization Problem Characteristics

When solving an optimization problem, there are several aspects that we need to understand to be able to find an appropriate approach for the problem. We need to know the objective of the optimization, decisions that the optimization needs to make, and finally key characteristics of the system that is optimized.

Objective

With regards to the objective function, it is a cost reduction goal. The goal is to minimize inventory while maintaining service levels. The most significant contributor of costs are the holding costs, followed by penalty costs for leaving barges waiting at docks, transportation costs and finally costs associated with not being able to meet demand. Holding costs serve to minimize inventory, while penalty costs are in place to maintain the service levels. Production costs are not included within the optimization; these costs are not influenced by the daily scheduling of the units and remain constant between completely different schedules.

Decisions

Outcomes of the optimization should yield a combined schedule for production, transportation and sales. This means:

- When to produce a specific product and into which tank
- When to move product between on-site and off-site tanks
- When to meet each order and from what tank

Key Characteristics

There are certain elements in the process that have to be taken into account for the optimization to translate into valid and usable results. According to supply planners, the 2 most significant elements that have to included are maximum inventory constraints and the uncertainty in demand. These are the largest challenges they run into when scheduling, as tanks must never become too full and constrain production. Uncertainty in total sales is the element that makes this scheduling much more challenging; there must be enough product available to meet demand in case it turns out higher than expected, but not so high that tanks are full if demand is lower than expected. As such, uncertainty and inventory capacity constraints are the 2 elements to keep in mind during the literature research.

2.6 Demand Modelling

An important aspect of the optimization in the uncertainty of demand. To properly incorporate this within the optimization, we need to be able to mathematically express the demand and its corresponding uncertainty. Shell forecasts demand on a monthly basis. What is important for us is the uncertainty surrounding this forecast, mostly in the difference between the forecast and the realized demand itself. In this section, we attempt to find a model for demand based on statistical analysis of historical data.

2.6.1 Forecast Error

For the demand of the upcoming month, the exact volume for each grade is unknown. We know that demand changes from month to month and depends on market conditions, making it hard to forecast and no general distribution fitting all demands. However, for our modelling, we are interested in how demand for a specific month is distributed with respect to its forecast. If we can fit a distribution to the forecast error, we have a distribution we can use for demand. Historical forecasts are not saved within Shell, making the data limited. We do have daily historical data from the ERP system containing every sale since 2010. This data stipulated when the order was placed, the pick-up date, as well as the grade and volume. Based on the pick-up date, we can find the total demand per month for each year and grade. We can then find what percentage volume of that demand was ordered before the initial schedule was made. We refer to this to as the percentage of firm demand: the percentage of volume that was ordered before the schedule was made. If a consistent pattern can be found here, we can use this to predict total demand and fit a distribution to it.

First, we plot the firm demand percentage for each month since 2010 in Figure **??** to see if there are any significant changes occurring and select an appropriate date range.

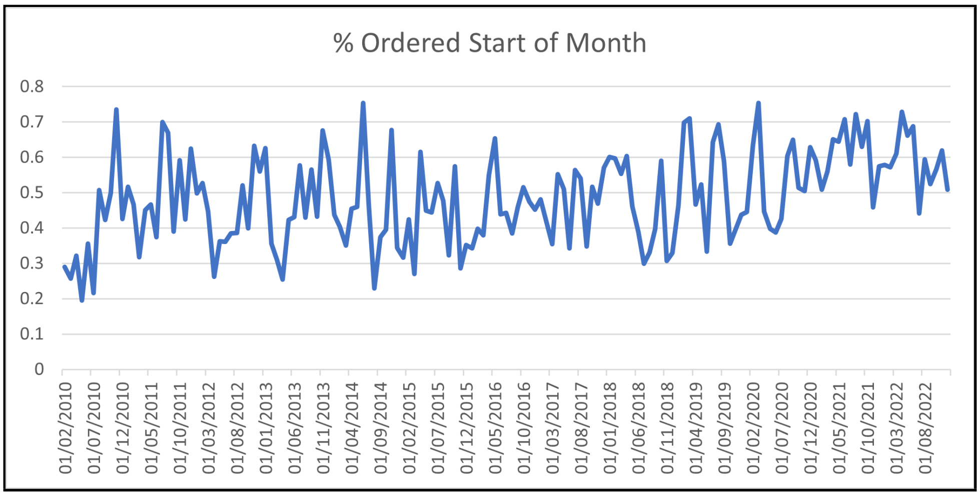

Figure 2.2: Hard demand at the start of each month as a percentage of total demand

Looking at this graph, we see stable behavior until the start of 2019, at which point we see a stable increase. From 2020 on wards this stabilizes again, and volatility also seems much lower. As we see that ordering behavior before 2020 is different from after 2020, we discard all the data from before 2020.

The next step is to see if the firm demand followed any distribution using hypothesis testing. We start of by testing whether the data is normally distributed. We use the Shapiro-Wilk test for this, as Mohd Razali and Bee Wah (2011) find using empirical testing that this is the most powerful test for determining whether a data set is normally distributed. Our null hypothesis for each sample is that they are generated from a normal distribution. For grades *A*, *B*, and *C* we find that, at 95% significance, we can not reject the null hypothesis and thus assume a normal distribution. For grade *D*, at 90% significance, we cannot reject the null hypothesis and thus assume a normal distribution. For grade *D* a lower level of significance is used as we cannot come to the same conclusion at the commonly used 95%. As grade *D* is by far the grade with the smallest sales volume, the added value of extra distribution fitting is limited and Shell is happy enough with 90%. Having all grades fit the same type of distribution also simplifies the modelling in Chapter 4.
With the types of distributions known for each grade's firm demand, we fit parameters to the distribution. We do this via the method of moments, finding the mean and variance equal to the sample mean and variance for each grade. We then use chi-square tests to confirm the fitness of these parameters, with the null hypothesis being that the firm demands are normally distributed with their population sample mean and variance. At 95% significance, we cannot reject the null hypothesis for *A*, *B* or *C*. At 90% significance, we cannot reject the null hypothesis for *D*. We will thus model them as normal distributions with their sample mean and variance as parameters. We can thus use our hard demand at the start of the month as a forecast for total demand, and know that the realizations of total demand will be normally distributed.

2.6.2 Lifting Pattern

Another important aspect of demand is how it is distributed within the month. This is because fulfilling orders is a crucial aspect with regards to feasibility and thus determines the schedule itself. Knowing how demand is distributed throughout the month will allow us to estimate how much product needs to be available at any given time. We do this analysis using time buckets of 1 week, meaning we are looking at the total demand per week. This size was chosen as generally, when a customer places the order, they specify the week in which they want to load and then decide on an exact day in collaboration with Shell to ensure product is available.

So, we look at the percentage of each month's total sales that were loaded during each week of the month, where the first 7 days of the month make up the first week, the second 7 days the second week, etc. Figure 2.3 shows the mean per week for each grade.

From Figure 2.3, we notice that week 5 has a significantly lower demand. This is in line with expectations, as week 5 concerns the 28$^{\text{th}}$-31$^{\text{st}}$ day of the month and is only half a week. For the other 4 weeks, their values are quite close to each other and resemble a uniform distribution. We can use the Chi-Square test to check for uniformity in the first 4 weeks of the month for each grade. Using the Chi-Square test, we obtain χ^2 values ranging between 0.01 and 0.03 depending on the grade. At a confidence level of 95%, we obtain a P-value of 0.35. As such, we fail to reject our null hypothesis that the liftings are uniformly distributed, and assume a uniform distribution for the weekly demand distribution.

Overall, we have answered the first research question: "What does the current situation of the Shell process look like?". We have a broad and technical understanding of the complete

Figure 2.3: Lifting patterns per grade

process, understand the costs involved as well as the requirements Shell has for an optimization approach. We also have modelled the monthly total demand and lifting pattern, allowing us to incorporate that into the optimization as per the requirements from Shell.

3 LITERATURE REVIEW

In this chapter, we aim to answer the second research question: "What techniques can we use to optimize the Shell process?". We start by gaining an overview of optimization problem in Section 3.1. We then look at the methods for solving these problems in Section 3.2. Finally, in Section 3.3, we present an overview of choices that we need to make during the modelling itself.

3.1 Types of Optimization Problems

In this section, we give an overview of commonly found types of optimization problems.

3.1.1 Integrated Production & Distribution Systems

Sarmiento and Nagi (1999) provide an overview of papers using an integrated approach for production and distribution systems, finding evidence that an integrated approach yields significant benefits over a sequential approach. However, most papers discussed had a more complex distribution network, where distribution focused on the routing between multiple nodes. In our research, transportation only includes the shipment of stock to the off-site storage locations with a set constant route; routing is not a variable within the optimization, nor is routing part of our objective function. Gupta and Maranas (2003) focus on distribution networks in the form of multi-site supply chains, more closely resembling the stock transfers in Shell. However, it is found that generally speaking integrated approaches require simplifications that limit the practical applicability of the results. So we instead focus the search strategy on inventory costs.

3.1.2 Inventory Routing

Combining elements from inventory management and the distribution part of the optimization, another modelling approach found in literature is inventory routing. It tackles both the inventory management problem on the supplier side as well as the distribution towards customers. (Cordeau and Laporte, 2014). Unlike integrated production and distribution systems it does not focus on the production side of decision making, only the management of inventory. However, it runs into a similar issue in the context of the Shell case: the delivery of product is out of scope. The optimization stops as soon as product is picked up by the customer. As a result, we exclude inventory routing from further research.

3.1.3 Inventory Management

As inventory costs are a main driver for the need of this project, we also investigate inventory optimization approaches. These fall under the umbrella of inventory management approaches such as the lot sizing problem. Jans and Degraeve (2008) give a review of how the base problem has been successfully solved and extended in other works. An extension that is not found is one that sets the total production level for a time period that must be achieved, which is a requirement for this research. The paper also identifies continuous processes and stochasticity as limitations generally found in lot sizing approaches. Bourland and Yano (1997) state that stochasticity may harm the long term feasibility of production in lot sizing problems. For this research, it is vital that production continues in the months after the schedule is made and so stochasticity must be properly addressed. Besides this, lot sizing approaches have the primary purpose of making the trade-off between production costs and inventory costs (Hezewijk et al. 2022) , whereas production costs are not considered as relevant to this research. Lot sizing and other traditional inventory management techniques are thus excluded as an option for this research.

3.1.4 Scheduling Problem

Another problem commonly found in literature is the production scheduling problem. Ehm and Freitag (2016) integrate production and transport scheduling, demonstrating that the scheduling problem can be applied to both production and transportation as is desired in our research. Lee et al. (1996) successfully use the scheduling problem to address inventory management in a petrochemical context with some similarities to the case of Shell. One challenge of the paper was dealing with computational complexity; the scheduling problem is NP-hard (Lee, 1995). In another petrochemical context Menezes et al. (2018) are able to address this complexity in a way that allows for larger scale problems to be solved. Verderame et al. (2010) give an extensive review of how the scheduling problem has been used to address stochastic demand in production systems, indicating that stochasticity can be incorporated. As a result, the optimization problem is treated as a scheduling problem, whereby transportation is considered one of the activities that needs to be scheduled. This should allow us to address the characteristics of the problem while keeping the complexity (and hence runtime) low. We can now focus our research on different forms of scheduling problems.

Tank Scheduling

A large part of the scheduling activities pertain to the management of inventory in tanks, as the storage in tanks also are what cause many of the constraints in the system to be in place. Tank scheduling is a subset of literature focusing on the use of tanks to store product, either as a buffer in between production stages or as use of inventory to sell from (Broch, 2010). A key decision here is which tanks to use for what storage; this differs from the Shell case as this has dedicated tanks per product that are rarely changed. As a result, we do not consider tank scheduling a suitable modelling approach for our scheduling problem.

Job Shop Scheduling

When inspecting the technical process description, one can see that product undergoes several steps before being picked-up by the customer. This bears resemblance to the job-shop scheduling problem, where jobs undergo multiple processing steps through machines and each processing step requires a machine (Pezzella et al., 2008). In the Shell case, jobs would be the orders from customers coming in, which means that there is significant uncertainty with regards to the jobs that need to be scheduled. Machines conducting the processing steps would be the PU, the jetty, gantry and the testing station.

With regards to the jetty and gantry, these steps are only important in the process as they are resources constraining the flow of product; we are only able to send a limited amount of product through them per day/month. There is no scheduling aspect to this; they function as constraints and not separate decision variables. Similar, the testing station is included in the process description because it determines when a tank becomes available for selling product after production has finished. Testing time is fairly constant, generally taking between 8-12 hours depending on when the sample was sent. It can be assumed to have infinite capacity, as the testing time is rarely influenced by how many other samples are being tested. So again there is no need to include a decision variable for testing, as the scheduling of testing does not have much impact on the entire process. Instead, it is a constraint that determines when a product becomes available. Looking back at the machines we had initially identified, this leaves only the production unit as a machine that jobs need to be scheduled onto.

Job shop scheduling is commonly used in discrete manufacturing contexts, but the Shell case contains liquid chemicals produced in bulk. This bears much more resemblance to the field of process systems engineering, which focuses on modelling the behavior physicochemical systems. (Pistikopoulos et al., 2021). The most significant difference from the manufacturing context for the Shell case is that there are no discrete jobs defined; a continuous production process is in place, which can be scheduled as a single large production job, or as multiple smaller jobs.

For job shop scheduling, decision variables and optimization goal generally focus on the sequencing of the jobs, especially adding value by incorporating set-up times or costs (Allahverdi et al., 2008). Set-up times is not relevant for our case, and sequencing only for the feasibility of the solution. Considering that our system contains only a single machine, has ill-defined jobs, and does not aim to minimize set-up or production costs, we look beyond job shop scheduling for an appropriate modelling approach. An area that is better able to address our system is process systems engineering.

3.1.5 Task Networks

Harjunkoski et al. (2014) provide a broad overview of existing scheduling methodologies applied in the chemical industry based on process systems engineering. They make a distinction between sequential processes and network processes. Sequential processes make discrete products or batches, and these are treated as separate units throughout the entire system. In network processes, products or batches can be combined or separated. This is like the Shell system, where multiple batches of products can be stored in a single tank, and sales orders may be fulfilled from multiple batches so long as the grade is the same. As such, we are dealing with a network problem. According to their overview, these are often represented as a state task network (STN) or a resource task network (RTN).

Both STN and RTN formulate the problem into bipartite graphs with two entities. STN uses states and tasks as the two entities, whereas RTN uses resources and tasks. STN was initially proposed by Kondili et al. (1993) , using states to indicate process materials, such as inventory, and using tasks to indicate the processing steps, such as production. Schilling and Pantelides (1996) iterate on this by introducing RTN, changing states to resources. Resources are physical materials required for processing steps which, unlike states, includes equipment units such as production units or tanks. By explicitly incorporating equipment units we can incorporate their constraining effects on scheduling with lower computational effort. This is relevant for our case, as resources such as tank sizes or gantry heavily influence the feasibility of scheduling; we thus base our modelling on the resource-task network formulation to model the Shell process.

3.2 Solution Methods

In this section, we investigate the different methods available to find solutions to optimization problems.

Knowing that we will be using the resource-task network formulation, we can now research the best methods of solving such a formulation to arrive at a (near) optimal solution. A prevalent approach in the RTN formulation is the use of mathematical programs such as a mixed-integer linear program (Castro et al., 2004). Advantages of using mathematical programming is that many inventory management systems include this as a core functionality, meaning that it can be implemented and maintained relatively easily. Dealing with stochasticity and keeping the problem size tractable can be a challenge, especially when dealing with non-linear programs, but options to address this exist (Grossmann, 2014).

MIP models allow us to express many types of constraints, but risk high runtimes. Even linear systems with few constraints but large solution spaces can quickly run into runtimes of multiple days. Predicting the runtime of MIP models is quite difficult and can vary wildly even for the same model with different inputs (Hutter et al., 2014). A significant challenge of using MILP with large scale instances is dealing with long runtimes, or even being able to find a feasible solution (James and Almada-Lobo, 2011). This is in part due to the number of decision variables, and thus the solution space, growing at ever increasing rates as the problem size increases. Logical implications, which will be required in the Shell case, are another factor that increase the difficulty of arriving at solutions (Codato and Fischetti, 2006).

Another, more modern, approach is reinforcement learning, such as used by Hubbs et al., (2020). The benefits of reinforcement learning are its generality and ability to adapt to new changes in the system, allowing one to incorporate a large amount of complexities while keeping the solution tractable (Wang and Usher, 2005). Its formulation naturally models areas such as beliefs, advanced demand information and learn from new data appearing. It is generally used for complex cases where expressing all behaviors as constraints may be difficult. The downside of reinforcement learning is that it is a very specialized area of optimization, meaning that it will be costly to find and retain people willing/able to maintain the model once developed and implemented for Shell. Besides this, integrating and interfacing a reinforcement learning model with the software currently used by Shell is another large challenge.

Besides mathematical modelling or reinforcement learning, heuristics can also be used to solve the problem to optimality (Avadiappan and Maravelias, 2021). The benefit of a heuristic is that they can be developed specifically for a problem case, potentially leveraging some of its characteristics to reduce runtime or improve the optimality gap. They can be applied in most scenarios, and with a wide body of research present on different heuristics, a strong performing heuristic can generally be found. However, a downside is that the strength of performance is not always guaranteed. Another drawback of these heuristics is that, if they are tailor-made for the problem, they are difficult to extend to other situations. As Shell has many more production units which could be optimized, a generalized and extendable approach is desired.

3.3 Required Choices

Finally, we research what choices we need to make during the modelling and how they influence performance.

3.3.1 Uncertainty

With regards to the requirements, proper integration of uncertainty is a must; it is the factor that currently makes the optimization most difficult. Verderame et al. (2010) provide an extensive overview of methods that have been applied to address uncertainty in the scheduling problem. They identify stochastic programming, chance constraint programming, robust programming and fuzzy programming as approaches that have been used with positive results.

Robust programming guarantees a certain result even in the worst-case realization of the uncertain parameters. This result depends on the optimization, but can be targets such as achieving feasibility, distance to optimality or a threshold objective value (Gabrel et al., 2014). A benefit of robust programming is that it does not require a specific distribution for the uncertainty, while still allowing the adjustments in performance against risk mitigation (Bertsimas and Thiele, 2006). A downside of not requiring a specific distribution is that, instead, the model is made to fit the information at hand (Gorissen et al, 2015). This would mean that extending to model to other units may provide to be more difficult, as the information available could change. Besides this, Shell's focus is truly on optimizing the costs; the model should make the trade-off between costs and risks. With robust programming, this adjustment is done manually.

Chance constraint, or probabilistic, programming differs from robust programming, as this approach allows infeasibilities or constraint violations with pre-specified probabilities (Bertsimas and Sim, 2004). This translates very well into areas such as service levels, for example if Shell wants to meet all demand in 95% of cases. The caveat of this approach is that distributions need to be known or assumed for the uncertain variables and integrated within the constraints. These distributions are often non-linear in nature, making the entire system non-linear and much more complex to solve. Sahinidis (2004) finds that this extra complexity can be addressed via approximation techniques to still arrive at near-optimal solutions. Besides this, hybrid approaches may also offer a solution. Liu et al. (2016) show the possibilities and benefits of a hybrid approach of robust and probabilistic programming for high complexity problems. Zhang (2019) shows how known normal distributions can be converted to deterministic counterparts. Although the relationship between a service level and demand is non-linear, if we keep a fixed service level, then the chance constraint becomes a linear deterministic constraint properly able to capture uncertainty. Using this approach, the model can be kept linear.

Fuzzy sets provide an alternative to using probability distributions when the required information is lacking. It uses partial truths instead of Boolean variables, turning binary variables into continuous variables and allowing deviations with respect to constraints and the objective function, while remaining both tractable and usable (Balasubramanian and Grossman, 2002). However, the nature of our case is crisply defined and the need for fuzzy sets is thus limited.

3.3.2 Planning Horizon

Besides how to model stochasticity, we must also decide on the horizon of the planning to specify for what time period we are optimizing. Finite horizon means the end of the schedule is set and we optimize the schedule until that horizon, but ignoring everything beyond it. Infinite horizon means that the optimization schedules into infinity, using all known information. This increases the complexity of the problem and makes it much more expensive to solve, but increases stability and long-term feasibility of the system (Rawlings and Muske, 1993). A rolling horizon can also be used, in which the schedule is reoptimized with an extended planning horizon. This has lower complexity than infinite horizon and is more stable with regards to long-term feasibility (Zhang et al., 2003) . The long-term feasibility is taken into account because a part of the old schedule is reoptimized within the new planning horizon; this re-optimization should then allow for achieving better results. (Sethi and Sorger, 1991).

Rolling horizon requires a more complex formulation, costing time and resources to implement and maintain than a finite horizon. For Shell, the critical aspect between each month's schedule is the ending inventory to ensure long term stability; we must end the month with enough inventory to construct a schedule capable of supplying the next month's demand. Dong and Maravelias (2021) show that one can develop terminal inventory level constraints in single-stage finite horizon optimization to create a near-optimal and stable long-term schedule.

3.3.3 Optimization Stages

There is also the question of how often during the month the optimization is run. As the production schedule can be adapted during the month, it would be possible to reoptimize the schedule on a weekly basis. Single-stage optimization means optimizing once at the start and not again, whereas multi-stage optimization is an approach which reoptimizes based on the realization of uncertain parameters. Multi-stage optimization is much more expensive to solve due to added complexity, although techniques such as decompositions and algorithms are available to address this and reduce computation time (Ning and You, 2019).

3.3.4 Representation of Time

Another modelling decision to be made is representation of time; chemical scheduling processes can be modelled either using either discrete or continuous representations (Floudas and Lin, 2004). Using discrete time for continous production processes may hurt the validity of the results.(Jia et al., 2003). Pinto et al. (2000) find that the resource constraints in discrete systems are easier to handle, as continuous representations tend to be non-linear, and that the discretized solutions have low integrality gaps. They also find that although the model size does increase due to the discretization, the order of magnitude for execution time stays the same, and that tight formulations with relatively low integrality gaps are another benefit of discrete time models. Continuous time representations do have the benefit of extendability: Karimi and McDonald (1997) show that discrete time events can be accommodated in continuous time representations.

Overall, we have presented the techniques we can use to optimize the Shell process, having gained an overview of techniques, an overview of solution methods as well as modelling choices we need to make. Based on the advantages and disadvantages, we choose for a resource-task network model that we solve using MIP techniques. In Chapter 4, we implement this model.

4 MODEL DESIGN

This chapter aims to answer our third research question: ”How can we apply these modelling techniques to the Shell case and adapt it to improve performance?”. We start the model design by making some general basic modelling decisions in Section 4.1, based on the findings from the literature research. These are decisions that need to be made for any model that is built, and can be generalized to other cases. We follow this by specifying the case specific decisions in Section 4.2, which are made to properly model the Shell system. We know from Chapter 3 that we are using a resource-task network problem formulation. In Section 4.3 we identify the resources and tasks for our resource-task network, which we use to build the network representation in Section 4.4. We then translate this network into a mathematical formulation in Section 4.5. We adapt this model to our specific case in Section 4.6 to include all indices, decision variables and constraints for an accurate reflection of the Shell system. In Section 4.7 we propose several simplifications that can be made with the goal of reducing the complexity of the model and improve runtime.

4.1 General Modelling Decisions

As stipulated in the requirements in Section 2.4, the usability of the solution is imperative. This includes being able to integrate it with software Shell already uses, and having the solution be easy to maintain by people with limited optimization knowledge. We discard reinforcement learning as a result; it is difficult to integrate in the software, and requires strong expertise to maintain or update. Mathematical models are already used within the Shell software and can be implemented by an already existing team; the software contains generic solvers capable of handling complex problems. There is also enough knowledge on this mathematical modelling to make maintenance and updating viable. Heuristics are technically also possible, but require more domain knowledge on the code itself than a mathematical program. As a result, we choose to implement a mathematical model that can be optimized via generic solvers.

The majority of Resource-Task Networks in literature are solved as mixed-integer linear programs (MILP), where the most common downside is the ability to solve larger sized problems within acceptable runtimes. Understanding how we can reduce the complexity of the produced model to decrease runtime, and how these changes influence the quality of the results, are the main focus of the experimentation stage in Chapter 5. In the following subsections we discuss our particular modelling choices.

4.1.1 Uncertainty

From the overview in the literature, it seems that probabilistic programming is most suitable for our research; its allows for incorporation of service levels and Shell has data available that should enable us to model the uncertainty as probability distributions. We can translate chance constraints into their deterministic counterparts to limit the additional complexity, and so long as we are using constant required service levels, the constraints will also remain linear. As a service level, Shell maintains a minimum probability of being able to meet total monthly demand at 95%.

4.1.2 Planning Horizon

The schedule is made at the start of the month and for the entire month. At this stage, Shell has used an LP to determine optimal production quantities for that month as well as expected sales. This optimization also takes into account expected stocks at the end of the month and demand for the next month, ensuring that there will be enough inventory present. As a result, we need not worry about the end-of-horizon effect if we stay within the limits of the planned total production per grade. This means we should be able to utilize a simpler formulation in the form of a finite horizon approach without sacrificing long term stability and feasibility. A finite horizon approach also naturally suits the problem, where each month is treated separately and there is no overlap in the made production schedules. As a result we choose a finite horizon approach.

4.1.3 Optimization Stages

Currently, supply planners do change the production schedule on a weekly basis to account for new information, such as shipping delays, the planner foreseeing specific issues for loading product, or perhaps a larger risk of breakdown in production. This would naturally suit a multi-stage optimization, where we re-optimize the model after a certain time period to account for new information. This would also allow for a rolling horizon approach instead of a finite horizon approach. Inherent within a multi-stage optimization is the use of recourse variables, which allow us to change decision variables based on the outcomes of uncertain parameters. For example, this could be the realization of the demand forecast. However, the most prevalent forms of recourse in the Shell case are fuzzy in nature and thus hard to express mathematically in scenarios. For example, production breakdown risks or expected barge delays are hard to quantify accurately and incorporate into the optimization. This is because this information received verbally or through email, and are often based on qualitative analysis rather than quantitative. This makes it extremely difficult to incorporate within the optimization as a recourse variable in a multi-stage model. Besides this, adaptations in the schedule are also made based on expected issues, meaning the uncertain parameters have not realized yet and we can thus not set our recourse variables.

Changing the schedule also requires significant work for a supply planner, having to contact and check with a multitude of stakeholders who are then also affected by these changes. So, significant changes are only made when new information appears that is expected to endanger the feasibility of the schedule. A multi-stage model with recourse variables would require large amounts of communication and increase the risk of errors, while only being able to incorporate a very limited amount of information required for proper recourse decisions. As a result, we choose to implement a single-stage optimization.

4.1.4 Representation of Time

In our case, the unit to be optimized is a continuous production unit, and all process steps following it also occur on a continuous basis. This means using a discrete time representation would lead to inaccuracies within the model and may threaten the validity of the results if done incorrectly. Time discretization also introduces additional decision variables for each time step, which may negatively impact the runtime. On the other hand, continuous time representation often result in non-linear formulations that negatively impact the runtime and scalability. Discrete time approaches also lead to tight formulations with low integrality gaps, which means strong results are very likely when choosing this approach. A discrete time approach also fits within the desired extendability off the model, as batch-processing units are closer to a discrete production unit. In order to preserve the linearity of the model, we decide to use a discrete time approach.

4.2 Case Specific Modelling Choices

Besides general modelling choices, we also make choice on how we model certain aspects of the Shell case.

4.2.1 Production

Production Rate

We model the production rate as a constant, where it is specified per grade what volume is produced per hour of production. In reality, this is a variable where the production rate can be changed from anywhere between 60-130%. Going above 100% puts mechanical stress on the unit and can only be done temporarily. A lower run rate puts less stress on the unit and is thus used whenever there is space for lower rates in the production schedule. We require a binary decision variable that indicates if we are producing a grade at any given time to ensure we are not producing multiple grades at the same time. Ideally, production rate would be a part of the optimzation. However, it would introduce many more decision variables (one for each time interval), as well as a linearization of constraints to determine production quantities. This would significantly increase the complexity and runtime of the model.

If there is space in the schedule to run at a lower rate, the supply planner can manually adjust this after the optimization. Demand scenarios may occur that require higher production rates than 100% to be feasible. Checking this can be done quite easily by looking at the total required production and total production capacity at the constant run rate. The supply planner can then ensure feasibility by manually increasing the volume produced per hour per grade in the model. Ideally, the supply planner does not need to fine tune parameters in the model. However, doing this for a single parameter (production rate) is possible, and given the significant impact of run rate, it is also desirable to have this be adjustable within the model.

Constant Testing Time

As mentioned in Section 2.2, product needs to be tested after production or transportation. This could be approached as a separate processing step, with a service time based on a theoretical distribution, adding more uncertainty and complexity in the model. Currently, supply planners use 12 hours as a rule of thumb and find this to be a very reliable measure. We thus assume a constant time period of 12 hours required for testing. We can then model this through a constraint on transportation/sales after production or transportation activities.

Waste Product Z

Shell considers product Z a waste product and stores it in its own dedicated tank. We model this as a decision to produce waste product Z and force it between production of A and any other grade. Primarily, this is done to properly track the production quantities of each grade. For example, when switching from A to C, the first few metric tons of product coming out of the production unit are contaminated with residue from A and stored as waste product Z in its dedicated tank. So we are not actually getting the planned quantity of C; although we are technically producing C, the product coming out of the production unit is Z. This mismatch needs to be accounted for during every grade change, complicating the inventory constraints. Instead, we decide to explicitly model producing *Z*; where we switch from producing *A* to producing *Z* and then to producing *C*.

This way the production decisions for each grade contribute directly to the inventory into the tank in which they produce. It also makes it easier to track what quantity of each grade has been produced in total, as it is just the sum of the production decisions for that grade. If we did not explicitly model this, we would instead have to offset for *B/C/D* depending on grade changes. Finally, it means the grade-tank relationship is maintained, where we know what grade is being produced based on which tank it is being produced into.

The downside of this choice is that it introduces extra decision variables into the model (production of *Z*), and thus increases the solution space. As producing *Z* has no benefit and only negatively impacts the objective function, most solution algorithms should quickly recognize *Z* is only required for feasibility. So, although the modelling complexity is increased, we expect it to not significantly increase runtime or the integrality gap.

4.2.2 Transportation

Barge Travel Time

In reality, the arrival times of a barge are uncertain; they may arrive a day earlier or later than planned. This poses a significant challenge in the planning process, but also in the modelling process, as it means that there is also uncertainty in the inventory positions throughout the chain. Including this inventory uncertainty would overly complicate the model, and instead we decide to model a constant barge travel time of 24 hours.

Containers

Trucks transport the product using intermodal containers known as ISO containers, which are standardized containers that can easily be stored or shipped via trucks, trains or ships. Trucks arrive with an empty ISO tank container which is filled at the loading gantry. Technically, it would also be possible for Shell to pre-load an ISO tank container by filling a tank long before a truck arrives and storing that container at a nearby third party storage site. The costs associated with this are generally too high and Shell has stopped doing this for this project's product line. As a result, we consider this option out of scope.

Heel

Moving products in and out of tanks is done via pumps. These pumps are capable of pumping almost all product out of a tank, but generally leave a little bit of product at the very bottom of the tank that they cannot get out; this is known as tank heel. This product is effectively stuck in the tank and cannot only be extracted using a special pump and some manual operations. This is rarely done due to the extra work and expenses required, and as such a constant amount of product is stuck in heel. Although this product is technically in inventory, it cannot be used to meet demand. This is addressed by tracking inventory including and excluding heel. For this project, we use inventory excluding heel and consider the option of using the heel as out of scope.

4.2.3 Sales

Demand Uncertainty

Besides the orders that have already been placed, referred to as hard demand, there are also orders that are expected to come in, known as soft demand. At the start of the month, Shell has a forecast of total demand expected to be sold during that month, which is the sum of hard and soft demand. Soft demand only has a total volume, and it is unknown when exactly it will come in. We know from the data analysis in Chapter 2 that total demand is uniformly distributed throughout the weeks of the month. So we can take our forecast demand and divide it uniformly over the weeks of the month to create a weekly demand forecast. We can then subtract the hard orders from this weekly forecast to arrive at a weekly soft demand to still be fulfilled.

Demand Shaping

The demand that has to be fulfilled within the system is not as straightforward as one may be expect. The products are specialized products that are sold on a monthly basis to regular customers, who can also buy the product at competitors. Every month, negotiations are done on price; if Shell demands too high a price, customers will order less volume. If the Shell price is lower than competitors, demand will be significantly higher. Shell can adjust its negotiating strategy during the month based on what volumes have already been ordered; prices can be increased in case high volumes are ordered, and decreased if low volumes are ordered. So, although demand has a random element to it, Shell can help shape it through their negotiation strategy. This essentially guarantees a minimum and maximum total monthly demand. This is relevant for the soft demand described in the previous section; the model requires orders to be fulfilled. As such, we also need to create synthetic orders to allow the model to fulfil the soft demand and see its impact on feasibility. The model thus receives a set of hard orders, consisting of orders already in the system, and also a set of soft orders which are made based on the maximum possible from the forecast.

Lifting Pattern

Since the price is negotiated, contact with the customer already exists and means that the time of pick-up can also be discussed. This allows the customer to indicate roughly when they need the product, and Shell can then choose when exactly the order is to be fulfilled. We use the rule of thumb that an order has to be fulfilled within the workweek for which it was originally requested to be fulfilled. This is incorporated by creating a time window for each order during which the order has to be fulfilled, which is then used as input for the model. This is automatically initialized as a specific week within the month and can be changed as required. So, we model orders as each having a time interval in between which they must be fulfilled. The time interval is determined by each order's release and due dates, with those dates being input for the model.

4.3 Constructing the Resource-Task Network

We refer back to Figure 2.1 for a schematic overview of the full process. As the name implies, resource-task networks contain two main elements: resources and tasks. Resources are physical materials required for processing steps, which are tracked as they impact the constraints and feasibility of solutions. These can be things such as equipment, materials or machine states. Note that resources within the RTN context has a wider meaning than the traditional word used in English. Resources in RTN are used to represent physical resources, but also more abstract concepts such as states or utilities. For example, material locations could also be considered a resource. Tasks are the processing steps themselves, which consume and/or produce the resources, and are closely related to the activities that need to be scheduled. If material locations were a resource, a transportation task could transform the material location to another material location. In this section, we identify the resources and tasks in the Shell case, and then relate the two in a network diagram.

4.3.1 Resources

We identify resources corresponding to the production unit, tanks, jetty, barges, and the loading gantry.

Production Unit

The first resource we identify from the schematic overview is the production unit; it can only be used to produce a single grade at a time. It is a resource that is consumed by a production task and produced when a production task is finished.

Tanks

The second, and most important, resource we see are the storage tanks; these reflect both the inventory we have on hand for fulfilling orders as well as the ullage for production decisions. Ullage and inventory are the complement of each other; given the tank capacity and inventory, we know the ullage; given the tank capacity and ullage, we know the inventory. We only need to use 1 of these 2 as a tank resource, as using both would introduce redundancy and unnecessary resources into the model. Besides this, there is another aspect to tanks: availability. This is due to required testing of the grade in a tank after production, which effectively means that inventory in that tank cannot be sold or transported. We may have inventory in a tank, but this will not be available until production and testing have been completed for that tank . So, tanks have two separate resources: tank level and tank availability.

Jetty

The next resource is the jetty, which can only have a single barge docked at the same time and has a maximum number of slots available per month. This translates to two resources: jetty availability and monthly jetty slots.

Barge Storage

Barges have a maximum capacity of product they can move, and in that sense behave similar to a tank. Testing is not relevant for products moved onto a barge, but the inventory on a barge is a resource that has to be taking into account to ensure we do not exceed its capacity. Barges also need to be scheduled, meaning that barge availability is also a resource that needs to be incorporated.

Loading Gantry

The final resource is the loading gantry, which is similar to the jetty. It can handle 4 trucks simultaneously, and has a maximum number of trucks per day it can handle. So here there are again two resources: gantry availability and daily truck slots.

4.3.2 Tasks

Tasks consume or produce resources and correspond to the main activities to schedule. We know our main scheduling activities from Section 2.4: production, transportation, and sales.

Production Tasks

When it comes to production the task is quite straightforward; simply producing a grade into a tank. As mentioned in the technical description (Section 2.2), each grade has dedicated tanks. This means that only a single grade is stored in each tank, and that this grade does not change. So, if we know the tank about which a decision is made, we also know what grade is being decided upon. As a result, the task only pertains into which tank is being produced.

Transportation Tasks

There are two options for transportation: moving product between on-site tanks containing the same grade, or moving producing from an on-site tank to an off-site tank. Moving product between on-site tanks is a simple operation and is considered a singular task. Moving product off-site contains a few more steps, where it must first be loaded onto a barge, where it is stored during transport, and then off-loaded into an off-site tank. This means that there are two tasks for off-site transport: loading product from an on-site tank onto a barge, and loading product from a barge to an off-site tank.

Sales Tasks

Sales tasks are again quite straightforward; they entail loading product from a tank onto a truck or barge to be shipped to a customer. When done from on-site tanks this uses the loading gantry resources, while off-site tanks are not restricted by any resources. The core task remains the same however, and we thus use a single task for fulfilling sales orders.

4.4 Resource-Task Network Representation

In Section 4.3 we defined the tasks and resources in a a general manner. Recall that these are generalized tasks and resources. When it comes to the formulation, each different variant of a task is modeled separately; producing into tank 1 is considered a different task to producing into tank 2. As such, we will have multiple separate tasks for each production, transportation, and sales. We create a network diagram to indicate dependencies and interactions between resources and tasks. This also helps us identify additional tasks and resources needed to properly model the behaviour of the system.

In the network diagram, we follow the industry standard and represent resources as circles and tasks as rectangles. The diagram acts as a disjunctive graph; resources can only be connected to tasks and vice versa. Tasks cannot connect to other tasks, and resources cannot connect to other resources. In the diagram, we have color coded the tasks and resources to help differentiate between the grades. We use two types of arrows: closed arrowheads to represent a one-way interactions and open arrowheads to represent two-way interactions. A one-way interaction means that the resource or task impacts the resource or task it points to; for example tank availability resource and sales task. The tank does need to be available for the sales task to occur, but carrying out a sale does not affect the tank availability. A two-way interaction would be the tank level resource and sales task; the tank level needs to be high enough to carry out the sales task, and the tank level is then decreased due to the sale being carried out as volume is sold.

We present the network diagram in 4 figures. Combining it all in a single diagram results in too many crossing paths and effectively makes the diagram unreadable. We start with the basic production process as the base. Each of the following figures also contains this basis, indicated by the dotted line, but differ from there on. These figures do not build on each-other. We have a diagram for the on-site sales tasks, the off-site shipment and sales tasks, and on-site movements. Having separate diagrams means that not all dependencies will be represented in the figures, for example scheduling a barge will influence tank levels, which in turn influences whether we can sell directly from an on-site tank. This is an indirect dependency, caused by tasks being connected to the same resource. All direct dependencies, which are direct connections between tasks and resources, are represented in one of the figures.

We start with the base process in Figure 4.1. We introduce the production tasks, tank level resource, and tank availability resource. We also introduce the previous grade resource, which indicates the grade produced previously. This is needed to be able to track sequencing constraints between *B*, *C*, and *D*, as well as to track when waste product is being produced. We also see that we model producing waste product *Z* as its own resource. Part of the reason for doing this is maintaining the product-to-tank relationship, which allows us to know what grade is being produced based on which tank is being produced into. This significantly reduces the number of decision variables for the production task. This will also allow us to track if enough product *Z* has been produced to prevent cross-contamination after a grade change. We also note the relationship between tank level resource for tank 7, containing *Z*, and production tasks for tanks 1, 2, and 3, containing *A*. This is due to *Z* being added to *A* during production if available. This reduces the tank level for 7 and increases it for 1,2, or 3 depending on which tank is being produced into.

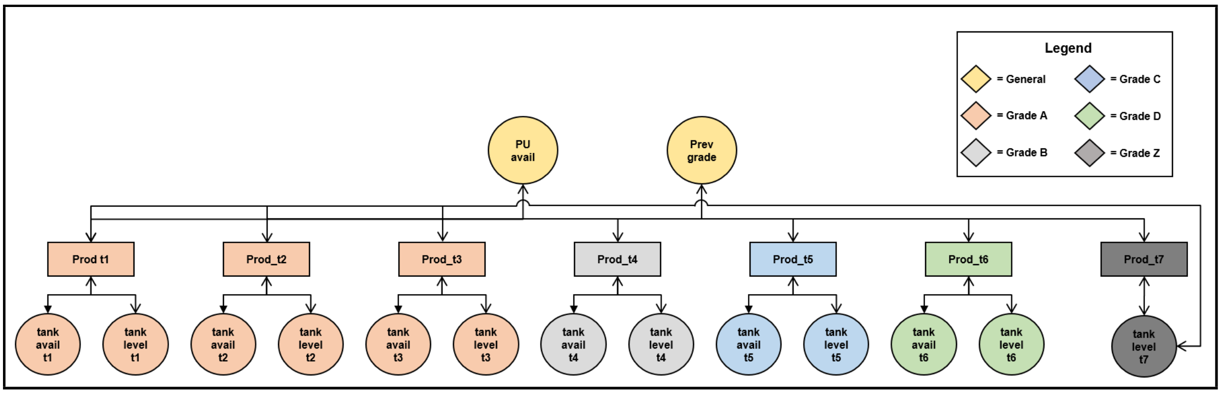

Figure 4.1: Network diagram for the basic production process.

Next we present the on-site sales process in Figure 4.2. This part is relatively straightforward; sales tasks require the tank to be available and contain enough product to be possible. They then also change the tank volume, as it decreases by the amount sold. Sales also require using the gantry to load the trucks, which is limited to 4 trucks simultaneously and has a maximum number of slots per day.

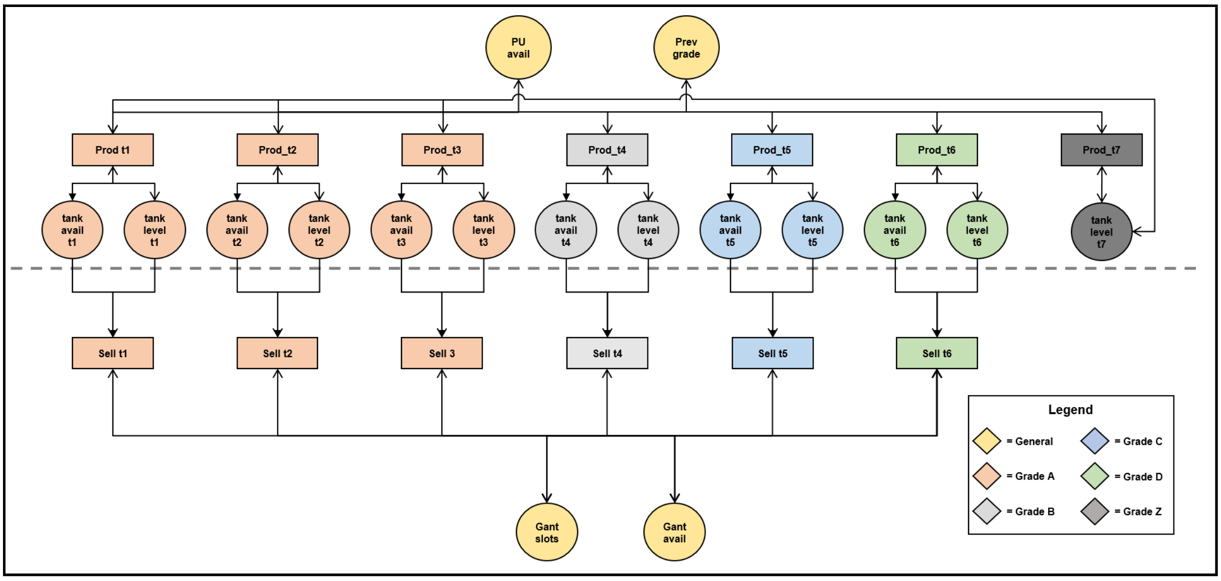

Figure 4.2: Network diagram for the on-site sales process.

We have the off-site shipments and sales in Figure 4.3. We see that moving product onto a barge bears similarities to on-site sales tasks, where they depend on tank availability and levels. They do require a barge to be available and scheduled, which again is limited by jetty availability (only 1 barge available simultaneously), as well as total jetty slots per month that can be used. Moving product onto a barge influences the levels of product on that barge for both grades *A* and *C*, and this also influences how much more product can be loaded as the barge has limited capacity. Off-loading product bears similarities to the on-site production process, where product deposited in tanks needs to be tested and causes the tank to become unavailable.

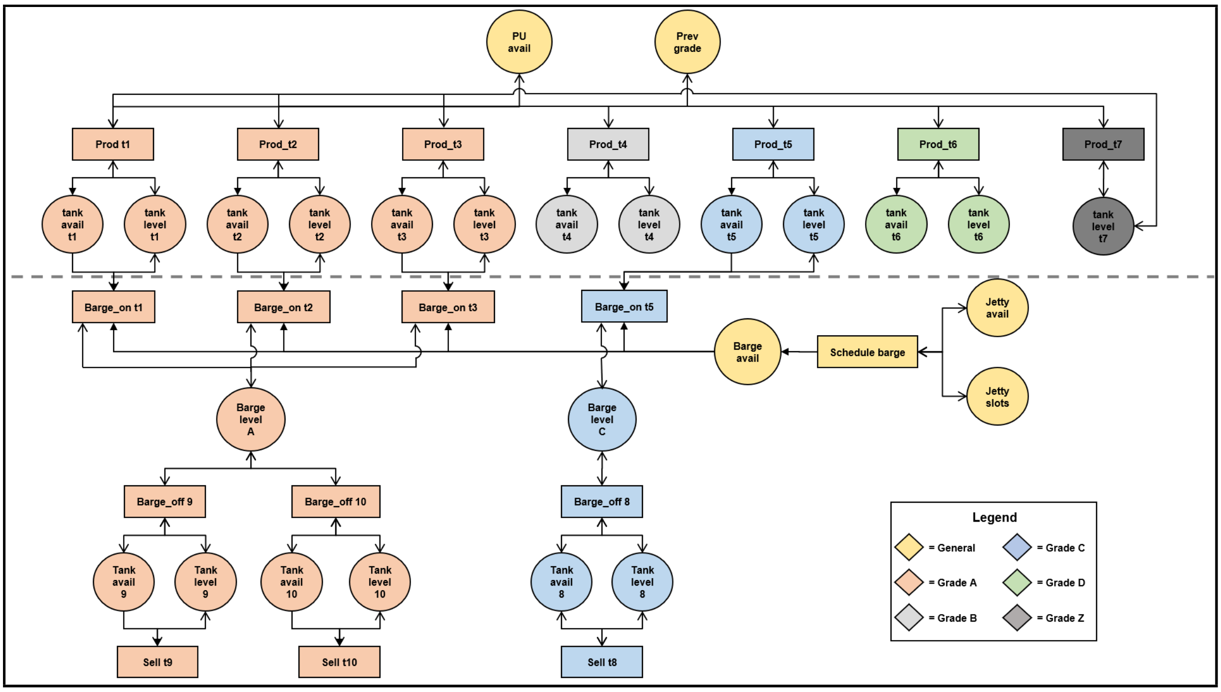

Figure 4.3: Network diagram for the process of barge movements and off-site sales.

Finally, we present the internal on-site movement process in Figure 4.4. This is a relatively simple part of the network. We have mapped tasks off moving product between the same set of two tanks together to keep the overview readable. It requires both tanks to be available, the tank we are moving product from to have enough product for the movement, and the tank into which we are moving the product to have enough ullage.

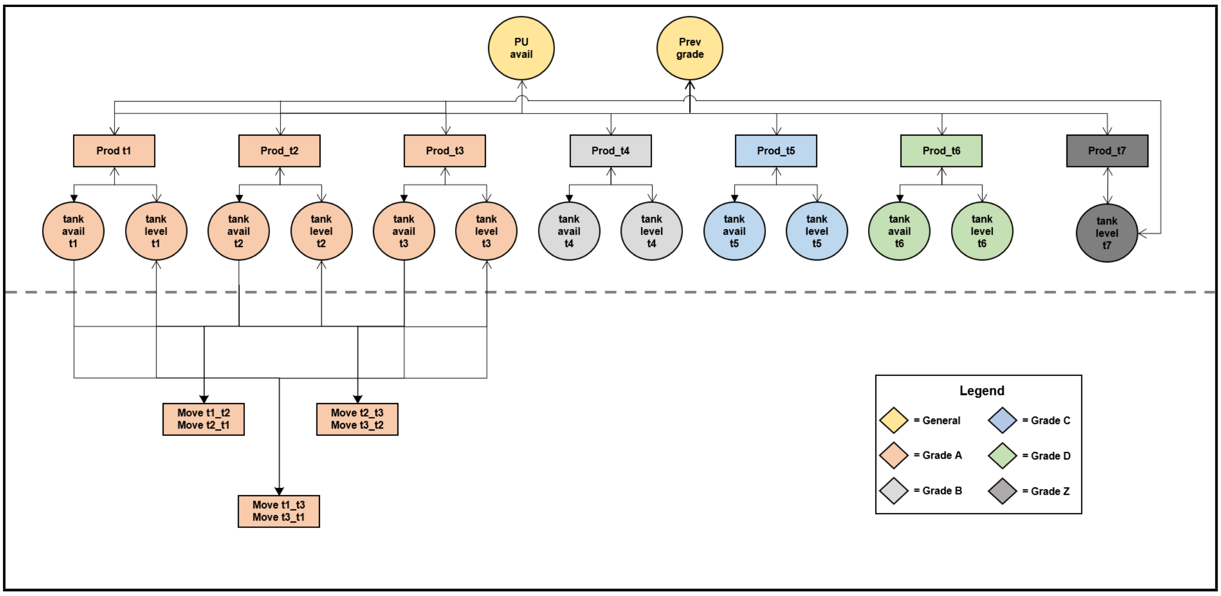

Figure 4.4: Network diagram for the process of moving product around on-site.

The network diagrams show how all tasks are directly or indirectly dependent on the tank levels and tank availability. This indicates how critical the management of tanks is to the performance of the supply chain, and reaffirms our core problem as presented in Figure **??**. We also note how the waste product *Z* influences the possibility of producing other grades. Throughout all four network diagrams, we see that the only method of reducing our waste tank levels is through production of *A*.

4.5 Resource-Task Network Model

In this section, we translate the resource-task network into its corresponding mathematical model. This is a formulation based only on the network's tasks and resources, not yet taking into account all the process details of the Shell system. We adapt and extend this formulation in Section 4.6 to incorporate all these elements, including cost parameters.

4.5.1 Sets and Indices

Hourly time $t \in T = \{0, 1, 2, ..., T^{max}\}$.
Grade $i \in I = \{A, B, C, D, Z\}$.
Tank $k \in K = \{0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9,\ 10\}$.
Barge $b \in B = \{0, 1, 2, ...,\ NrBarges\}$.
Order $o \in O = \{0, 1, 2, ..., NrOrders\}$.

4.5.2 Decision Variables - Tasks

$$X_{k,t} = \begin{cases} 1 & \text{if we produce into onsite tank } k \text{ at time } t \\ 0 & \text{otherwise} \end{cases}$$

$$Z_{o,k,t} = \begin{cases} 1 & \text{if we fulfill order } o \text{ from tank } k \text{ at time } t \\ 0 & \text{otherwise} \end{cases}$$

$$Y_{b,t} = \begin{cases} 1 & \text{if we use barge } b \text{ at time } t \\ 0 & \text{otherwise} \end{cases}$$

$$Y^{on}_{k,b,t} = \begin{cases} 1 & \text{if we pump product from on-site tank } k \text{ onto barge } b \text{ at time } t \\ 0 & \text{otherwise} \end{cases}$$

$$Y^{off}_{b,k,t} = \begin{cases} 1 & \text{if we pump product from barge } b \text{ into off-site tank } k \text{ at time } t \\ 0 & \text{otherwise} \end{cases}$$

$$W_{k,\ k',\ t} = \begin{cases} 1 & \text{if we move product from onsite tank } k \text{ to onsite tank } k' \text{ at time } t \\ 0 & \text{otherwise} \end{cases}$$

4.5.3 Auxiliary Variables - Resources

Tracking resources also requires variables to be introduced into the model. We refer to these as auxiliary variables instead of decision variables. Technically speaking, they are the same as decision variables when solving a model. We make a semantic separation of the two in the model here. This is because it allows us to easily track the need for certain variables; auxiliary variables are theoretically speaking not necessary to arrive at a solution. Auxiliary variables are determined completely by decision variables and parameters. Knowing the values of parameters and decision variables, we can also deduce the values of all auxiliary variables. Their use is to allow us to model complex behaviour, for example by linearizing a non-linear formulation. We introduce the following auxiliary variables:

$$V_t = \begin{cases} 1 & \text{if the production unit is available at time } t \\ 0 & \text{otherwise} \end{cases}$$

$$K_t = \begin{cases} 1 & \text{if the jetty is available at time } t \\ 0 & \text{otherwise} \end{cases}$$

$$S_{b,t} = \begin{cases} 1 & \text{if barge } b \text{ is available at time } t \\ 0 & \text{otherwise} \end{cases}$$

$$L_{i,t} = \begin{cases} 1 & \text{if grade } i \text{ is the grade produced previously at time } t \\ 0 & \text{otherwise} \end{cases}$$

$G_t \in \{0, ..., GantrySlots\} =$ Daily gantry slots left at time t.

$H_t \in \{0, ..., CAP_{Gantry}\} =$ Gantry capacity left for simultaneous trucks at time t.

$J_t \in \{0, ..., JettySlots\} =$ Monthly jetty slots left at time t.

$R_{k,t} \in \{0, ..., CAP_k\} =$ Volume of product in tank k at time t.

$U_{b,t} \in \{0, ..., CAP_b\} =$ Total volume of product on barge b at time t.

4.5.4 Objective Function

The objective function consists of the inventory costs for products in tanks and on barges, measured via auxiliary variables $R_{k,t}$ and $U_{b,t}$. Transport costs for barges are also incurred, which are both a fixed charge if the barge is used (Y_b), as well as a variable cost depending on the volume transported on the barge ($Y^{on}_{k,b,t}$). We incur these costs at the end of the day, so every 24 hours.

$$\min \sum_{\delta=1}^{NrDays} \left(\sum_{k \in K} R_{k,23\delta} + \sum_{b \in B} U_{b,24\delta} \right) + \sum_{b \in B} \left(Y_b + \sum_{t \in T} \sum_{k \in K^{on}} Y^{on}_{k,b,t} \right) \tag{4.1}$$

4.5.5 Constraints

The only constraints formulated by the resource-task network formulation are the actual resource balance constraints for each auxiliary variable. We introduce additional required constraints in the full model in Section 4.6.

The production unit is available if no production is scheduled on it, an unavailable in case production is scheduled on it:

$$V_t + \sum_{k \in K^{on}} X_{k,t} = 1 \qquad \forall t \in T \tag{4.2}$$

For the daily gantry slots, we can deduce its value from the total slots available during a day and how many sales are scheduled during that day so far.

$$G_t = GantrySlots - \sum_{k \in K_{on}} \sum_{o \in O} \sum_{\delta=0}^{23} Z_{o,k,t+\delta} \qquad \forall t \in \{0, 24, 48, ..., T^{max}\} \tag{4.3}$$

For the gantry capacity, we subtract the number of on-site orders fulfilled at time t from the total capacity.

$$H_t = GantryCap - \sum_{k \in K_{on}} \sum_{o \in O} Z_{o,k,t} \qquad \forall t \in T \tag{4.4}$$

For the monthly jetty slots, we base the constraint on the number of barges scheduled.

$$J_t = JettySlots - \sum_{b \in B} \sum_{\delta=0}^{t} S_{b,\delta} \qquad \forall t \in T \tag{4.5}$$

For the jetty availability, we have a capacity of 1.

$$K_t = 1 - \sum_{b \in B} S_{b,t} \qquad \forall t \in T \tag{4.6}$$

For the barge availability, we know if it is available at time t based on whether it has been scheduled. Each barge has a set parameter for arrival and departure,

$$S_{b,t} = Y_b, \qquad \forall t \in \{ARR_b, ..., DEP_b\}, \forall b \in B \tag{4.7}$$

$$S_{b,t} = 0 \qquad \forall t \in \{0, ..., ARR_b - 1, DEP_b + 1, T^{max}\}, \forall b \in B \tag{4.8}$$

For the last grade produced, we require three separate equations to properly capture all possible scenarios.
First, we define that there must always be a grade that was produced previously:

$$\sum_{i \in I \setminus \{Z\}} L_{i,t} = 1 \quad \forall t \in T \tag{4.9}$$

Second, if a grade was produced into a tank at $t-1$, it must be the grade produced previously at time t:

$$L_{i,t} \geq \sum_{k \in K_i^{on}} X_{k,t-1} \quad \forall i \in I \setminus \{Z\}, \forall t \in T \tag{4.10}$$

Third, and finally, we constrain the previous grade in case no grade was produced at $t-1$. If this is the case, and no other grade was defined as the previous grade produced at $t-1$, then this grade was the previous grade produced at $t-1$ and must still be the previous grade produced at time t.

$$L_{i,t} + \sum_{i' \in I'} \left(L_{i',t-1} + \sum_{k \in K_{i'}^{on}} X_{k,t-1} \right) \geq 1 \quad \forall i \in I,\ \forall t \in T,\ I' = I \setminus \{i, Z\} \tag{4.11}$$

For the barge inventories, we again require multiple constraints. We start with a constraint for the product while product is loaded onto the barge. We use sets LC_b^{on} to indicate the time indices for which the barge is available for loading, and LC_b^{off} to indicate the time indices for which the barge is available for depositing product.

$$B_{b,t} = B_{b,t-1} + f_b \sum_{k \in K^{on}} Y^{on}_{k,b,t} \qquad \forall b \in B, \forall t \in LC^{on}_b \tag{4.12}$$

When off-loading, barge inventory decreases by the product being offloaded:

$$B_{b,t} = B_{b,t-1} + f_b \sum_{k \in K^{off}} Y^{off}_{b,k,t} \qquad \forall b \in B, \forall t \in LC^{off}_b \tag{4.13}$$

When not loading/unloading, barge inventory stays constant:

$$B_{b,t} = B_{b,t-1} \qquad \forall b \in B, \forall t \in T \setminus \{LC^{on}_b,\ LC^{off}_b\} \tag{4.14}$$

For the final set of constraints we have the tank level constraints. We can only define this for off-site tanks and the on-site tanks containing grades *B/C/D*. For the on-site tanks containing *A* and *Z*, we extend the RTN model in Section 4.6.

First we set the initial inventory (INI_k) levels for each tank which are an input into the model:

$$R_{k,0} = INI_k \qquad \forall k \in K, \tag{4.15}$$

We also set the maximum and minimum inventory levels per tank, based on the tank capacity CAP_k.

$$R_{k,t} \geq 0 \qquad \forall t \in T,\ \forall k \in K \tag{4.16}$$

$$R_{k,t} \leq CAP_k \qquad \forall t \in T,\ \forall k \in K \tag{4.17}$$

For on-site tanks of grades *B,C,D*, the inventory is determined by the previous inventory level plus inputs and minus outputs. Inputs are production and product being pumped into it from another tank. Outputs are pumping product into another tank, sales, and stock transfers to off-site tanks. We use parameter p_k as the production rate per hour, f_k as the internal pumping rate per hour, and f_b as the barge loading rate per hour. We do not model flow for sales as they each sale is a small volume and the time taken thus not impactful for the inventory.

$$\begin{aligned} R_{k,t} = R_{k,t-1} + p_k X_{k,t-1} + &\sum_{k' \in\ K^{on}_i k} f_k \left(W_{k,k',t-1} - W_{k',k,t-1} \right) \\ &- \sum_{o \in O^k} Z_{o,k,t-1} - f_b \sum_{b \in B} Y^{on}_{k,b,t} \\ &, \forall t \in T \setminus \{0\},\ \forall k \in K^{on}_i,\ \forall i \in \{B, C, D\} \end{aligned} \tag{4.18}$$

For the off-site tanks, the only inputs are stock transfers from on-site tanks, and the only output is sales:

$$R_{k,t} = R_{k,t-1} - \sum_{o \in O^k} Z_{o,k,t-1} + f_b \sum_{b \in B} Y^{off}_{b,k,t}\ , \qquad \forall t \in T,\ i \in \{A, C\}\ ,\ \forall k \in K^{off}_i \tag{4.19}$$

4.6 Full Model

In this section we adapt and extend the RTN formulation to arrive at the complete model. We do so in two stages. First, we extend the formulation to incorporate stochasticity and properly model waste product *Z*. This introduces extra auxiliary variables into the model. We also introduce many new constraints into the model to properly capture system behaviour in the model and ensure feasible output. We do not present these here, and instead the full model can be found in Appendix A.

4.6.1 Chance Constraints

With regards to the uncertain sales, we use a chance constraint to indicate the max allowed probability of not meeting all demand. This means that the probability of demand for grade *i* exceeding the planned sales of grade *i* must be lower than that grade's service level λ_i.

$$P\left(\sum_{k\in K_i}\sum_{t\in T}\sum_{o\in O} Z_{o,k,t} \leq DEM_i\right) \leq \lambda_i, \qquad \forall i \in I \tag{4.20}$$

As we know that total demand is normally distributed, we can transform this chance constraint into its deterministic counterpart based on the probability density function of the standard normal distribution in similar fashion to Zhang et al. (2019) [33]. So, we replace constraint (4.20) with constraint (4.21).

$$\frac{\sum_{k\in K_i}\sum_{t\in T}\sum_{o\in O}(Z_{o,k,t}) - E\left[DEM_i\right]}{\sqrt{var\left[DEM_i\right]}} \leq \Phi^{-1}(1-\lambda_i), \qquad \forall i \in I \tag{4.21}$$

4.6.2 Waste Product Z

Dealing with waste product *Z* requires quite a bit of extra modelling. This is due to the highly specific set of circumstances that need to be taken into account for both the production and blending of it. In this section we first constrain the model such that production of *Z* occurs when needed, then we constrain its blending into *A*. We then finish the constraints for the tank levels for on-site tanks containing *A* and *Z*.

For production of *Z* are two cases: We have just produced *B/C/D* and now want to produce *A*, for which we use the variable P_t^1, and we have just produced *A* and now want to produce *B/C/D*, for which we use the variable P_t^2. If we did not produce *B/C/D*, we do not need to produce *Z* before producing *A*. This is expressed by P_t^1 being forced to 0:

$$\sum_{i\in\{B,C,D\}} L_{i,t} \geq P_t^1 \qquad \forall t \in T \tag{4.22}$$

Similarly for P_t^2, only with having produced *A*:

$$L_{A,t} \geq P_t^2 \qquad \forall t \in T \tag{4.23}$$

It may also be that we are not producing *A*, and we do not need to produce *Z*. In this case, P_t^1 should be forced to 0.

$$\sum_{k\in K_A^{on}} X_{k,t} \ \geq P_t^1 \qquad \forall t \in T \tag{4.24}$$

Similarly for P_t^2, only with producing *B/C/D*:

$$\sum_{i\in\{B,C,D\}}\sum_{k\in K_i^{on}} X_{k,t} \ \geq P_t^2 \qquad \forall t \in T \tag{4.25}$$

Now we still need to force P_t^1 to 1 in case we produced *B/C/D* previously and want to produce *A*:

$$\sum_{i\in\{B,C,D\}} L_{i,t} + \sum_{k\in K_A^{on}} X_{k,t} - P_t^1 \leq 1 \qquad \forall t \in T \tag{4.26}$$

Similarly for P_t^2, only with producing *B/C/D* and having produced *A* previously:

$$L_{A,t} + \sum_{i \in \{B,C,D\}} \sum_{k \in K_i^{on}} X_{k,t} + P_t^2 \leq 1 \qquad \forall t \in T \tag{4.27}$$

With this set of constraints, we can now recognize when to force production of *Z* before starting to produce a different grade. We force production of *Z* for the required amount of timesteps (t_z) to ensure enough volume has been produced before starting production after a grade change: Force production of *Z* in case we want to produce *A* after having produced *B/C/D*:

$$\sum_{\delta=1}^{t_z} X_{K_z,\ t+\delta} \geq t_z P_t^1 \qquad \forall t \in T \tag{4.28}$$

Force production of *Z* in case we want to produce *B/C/D* after having produced *A*:

$$\sum_{\delta=1}^{t_z} X_{K_z,\ t+\delta} \geq t_z P_t^2 \qquad \forall t \in T \tag{4.29}$$

Having constrained the production of *Z*, we now need to also ensure that it is blended away into *A* afterwards. We introduce an additional variable Q_k, t. This variable becomes 1 if we have *Z* in inventory and are producing *A* into tank k. This variable is required in the resource balance constraints to track inventory levels in each tank. We specify the tank into which it is being blended as this allows us to keep proper track of it in the inventory balance constraints later. First, we constrain it such that we cannot blend *Z* into *A* if there is not enough *Z* ($Q_{k,t} = 0$):

$$r_z^k Q_{k,t} \leq R_{k_z,t} \qquad \forall t \in T,\ \forall k \in K_A^{on} \tag{4.30}$$

Next, we constrain it such that we cannot blend *Z* into *A* if no *A* is being produced into that tank ($Q_{k,t} = 0$):

$$Q_{k,t} \leq X_{k,t} \qquad \forall t \in T,\ \forall k \in K_A^{on} \tag{4.31}$$

Last, we must blend *Z* into *A* if it is being produced and enough *Z* is available ($Q_{k,t} = 1$). We can check if there is enough *Z* by taking $R_{k_z,t} - r_z^k$, if this is positive then there is enough product available; else there is not. By dividing the resulting value by the tank capacity, we scale it to a value between -1 and 1. If this value is positive (ergo there is enough *Z* in the tank) and we are producing *A* ($X_{k,t} = 1$), then we force $Q_{k,t}$ to 1:

$$\frac{R_{k_z,t} - r_z^k}{CAP_k} + X_{k,t} - Q_{k,t} \leq 1 \qquad \forall t \in T,\ \forall k \in K_A^{on} \tag{4.32}$$

With the behaviour of *Z* properly constrained, we can now also introduce the tank level constraints for on-site tanks containing *A* and *Z*.For on-site tanks of grade A, the inventory is determined by the previous inventory level plus inputs and minus outputs. Inputs are production, *Z* being blended in, and product being pumped into it from another tank. Outputs are pumping product into another tank, sales, and stock transfers to off-site tanks:

$$\begin{aligned} R_{k,t} = R_{k,t-1} + p_k X_{k,t-1} + f_z Q_{k,t-1} - \sum_{o \in O^k} Z_{o,k,t-1} \\ + \sum_{k' \in\ K_A^{on} k} f_k \left(W_{k,k',t-1} - W_{k',k,t-1} \right) - f_b \sum_{b \in B} Y_{k,b,t}^{on} \\ , \forall t \in T \backslash \{0\},\ \forall k \in K_A^{on} \end{aligned} \tag{4.33}$$

For the tank containing *Z*, the only input is production, and the only output is blending away into A:

$$R_{k_z,t} = R_{k_z,t-1} + p_k X_{k_z,t-1} - \sum_{k' \in \ K_{on}^A} f_z Q_{k,t-1}, \qquad \forall t \in T \backslash \{0\} \tag{4.34}$$

We note the large number of additional constraints and variables required to model waste product *Z*. It is required in the model as it has a large impact on the feasibility of scheduling and is currently a main driver for scheduling decisions made by supply planners.

4.7 Simplified Model

The full model contains a large number of variables and constraints. Even for small scenarios, we expect the runtime to be very high. As such, we propose several simplifications that may reduce runtime while still resulting in feasible schedules. We will test the impact of these simplifications in Chapter 5.

4.7.1 No waste product

The reason for including the waste product *Z* in the modelling is due to it forming a constraint on production. Waste product is stored in a separate and very small tank that is almost full after 2 grade changes. Although technically the waste product could be disposed of, this is a costly process that is only used in emergencies, and thus considered out of scope for our research. This means the only way of lowering the waste product inventory is by blending *Z* into fresh grade *A* during its production, with a maximum allowed percentage (5%) of *Z* in *A*. By producing enough *A* before switching grades again, we can ensure the waste tank is empty. As a result, the production planners currently use the heuristic of a minimum batch size for grade *A*, where if they produce *A*, they do so for a minimum volume before switching to another grade again. We model this as follows. We introduce the parameter MBS, specifying the minimum batch size in terms of time steps that *A* needs to be produced for after having produced any other grade. We then build the constraint by looking if any other grade is produced previously and whether we are currently producing grade *A*. If both are true, production of *A* is then forced for the next MBS timesteps:

$$\sum_{\delta=0}^{MBS} X_{k,t+\delta} \geq MBS \left(\sum_{i \in I \backslash \{A\}} L_{i,t} + X_{k,t} - 1 \right), \qquad \forall t \in T, \forall k \in K_A^{on} \tag{4.35}$$

When we include constraint (4.35) into the model, we no longer need to model *Z*. As a result, we can remove constraints (4.22)-(4.32). We can also remove the auxiliary variables introduced for these constraints (P_t^1, P_t^2, $Q_{k,t}$). It further reduces the number of decision variables as it reduces the number of grades. This should significantly reduce the complexity and thus runtime of the model.

4.7.2 No pumping on-site

Another simplification that we can make is removing the decision variable $W_{k,\ k',\ t}$. We introduced this variable as it is an option to move product between tanks on-site, adding some flexibility for the scheduler. This flexibility essentially means that it may make an otherwise infeasible schedule feasible. However, the downside is that it introduces an extra $6T^{max}$ variables into the model. If we have an entire month of scheduling, this means an additional 4,320 variables. So, another simplification we test is removing this decision variable.

4.7.3 Reducing Auxiliary Variables

The full model contains a large number of binary variables, which tend to negatively impact runtime. By removing some of these binary variables, we may be able to improve the runtime. However, this does require us to change some constraints. If the new constraints change the problem structure, this may actually make the model more complex and negatively impact runtime.

Some auxiliary variables can be replaced via simple constraints to reduce the number of variables in the model. This is because a constraint is needed to properly define the values of auxiliary variables; if we can instead formulate a constraint of similar structure that is not reliant on the auxiliary variable, this should reduce the runtime. This is best shown in the case of auxiliary variable V_t for PU availability as defined by constraint (4.2). The necessity of PU availability is to ensure that only a single grade is scheduled at a time, which can also be done via constraint (4.36). This makes constraint (4.2) superfluous and we remove it from the model as a result, replacing it with (4.36).

$$\sum_{k \in K^{on}} X_{k,t} \leq 1 \qquad \forall t \in T \tag{4.36}$$

We also do this for the daily gantry slots constraint (4.3), replacing it with constraint (4.37).

$$\sum_{o \in O} \sum_{k \in K^{on}} \sum_{\delta=0}^{23} Z_{o,k,t+\delta} \leq a, \qquad \forall t \in T^{days} \tag{4.37}$$

For the gantry capacity constraint (4.4), we replace it with (4.38).

$$\sum_{k \in K^{on}} \sum_{o \in O} Z_{o,k,t} \leq GantryCap, \qquad \forall t \in T \tag{4.38}$$

For the monthly jetty slots (4.5),we replace it with (4.39).

$$\sum_{b \in B} Y_b \leq j \tag{4.39}$$

For the jetty availability (4.6), we replace it with (4.40)

$$\sum_{k \in K^{on}} Y^{on}_{k,b,t} \leq 1, \qquad \forall b \in B,\ \forall t \in LC^{on}_b \tag{4.40}$$

With these adjustments made to the model, we have removed the variables V_t, G_t, H_t, J_t, and K_t while keeping the model behaviour and validity the same.

Overall, in this chapter we constructed the Resource Task Network. As we expect high runtimes, we suggest three simplifications that may aid in reducing runtime. We note that there are potential simplifications within RTN that should not influence the solution space, such as the removal of auxiliary variables, as well as simplifications that do fundamentally change model behaviour and feasibility such as the removal of the waste product. In Chapter 5 we study the impact of the individual simplifications separately, as well as when all three are combined into a single simplified model, and compare it to the performance of the full model.

5 VALIDATION AND PERFORMANCE

In this chapter we answer the fourth research question: ”How does the model perform and how do the adaptations influence performance?”. We first describe the experimental setup in Section 5.1, then validate model output and performance in Section 5.2. Finally, we present the results in Section 5.3.

5.1 Experimental Setup

In order to arrive at solutions to the mathematical model, the model has to be implemented in a software with an integrated solver that is capable of handling large scale MIP models. The two most commonly options used in academic and commercial context are Gurobi and CPLEX, which can handle complex models with competitive runtimes. In some cases the Gurobi solver has better results, and in other cases the CPLEX solver has better results, but it is difficult to determine based on theory which one is best suited (Hutter et al., 2014) [21]. Besides this, both are implementable in a wide array of programming languages such as C++, Python and Java. Having previously had successful results implementing Gurobi in a python environment, we use this to implement and solve our mathematical model.

For each of the different models, we want to compare their performance in a wide array of scenarios. We are interested in three measures of performance: runtime, objective value, and optimality gap. The objective value is of importance as it ultimately reflects the quality of the solution found; lower objective values mean higher quality solutions found. The optimality gap indicates how close to an optimal solution the solver is able to get, and thus gives us a great indication on model performance when runtimes become too long. The runtime lets us know the time the model has taken to arrive at a solution and is capped at 2 hours per model.

5.1.1 Scenarios

We experiment with different scenarios to be able to compare model performance in different circumstances. The two most important aspects here are the expected demand for a month and the starting inventory levels. We use high demand, which requires the unit to be producing 95% of the time, medium demand which requires the unit to be producing 85% of the time, and low demand with production running 65% of the time. We also run each of these scenarios with three different starting inventory levels, with high tank levels at all being 60% full, medium at 40% full, and low at 20% full. These ranges are determined based on some initial experiments, where going outside the ranges would lead to Gurobi reporting that the model is infeasible. As each combination of starting inventory and demand could occur in reality, we experiment with each possible combination. Ideally, multiple instances with slight variations are run for each combination to increase the reliability of results. However, we find in Section 5.3 that results between scenarios and models are consistent enough to base our conclusions on.

With significantly long runtimes and high model complexities, we cannot experiment with full month schedules. This means that we will be running our experiments with shorter schedules. However, a model performing better on short schedules may not perform better on large schedules; the scalability of a model is also important. To gain insight into the scalability of each model, we run each experiment as several different schedule lengths. The full model, with a time limit of 2 hours, starts struggling to produce any feasible solutions for schedules for 5 days when demand and starting inventory are high, and at 6 days for high or medium demand with high starting inventories. At 7 days, it is not able to find a solution within the work limit of 2 hours for most scenarios. As such, we limit our performance testing and comparisons starting at 2 days and increasing to at most 6 days. We could increase the time limit for which we let the model run, but as we are running 4 different models at 9 different scenarios, each time increase is multiplied by 36. Each hour by which we increase the work limit will increase experimentation time by 36 hours.

5.1.2 Work as a measure of runtime

When looking at the time taken for a model to arrive at a solution, also known as the runtime, there are several issues that hamper the comparability of results. Even using the same hardware, there are many external factors influencing the runtime. This is because factors such as ambient temperature, CPU workloads, cooling performance, and more, will all impact the performance of the hardware and thus the runtime. As a result, the exact same model ran on the same hardware with the same inputs, parameters, and deterministic solving algorithms will take different amounts of time to solve. A method of solving this issue is by using a measure present within the Gurobi ecosystem that is known as 'work'. Work is a unitless measure, roughly reflecting the amount of computation a computer is capable of doing roughly in 1 second on a single thread (although this depends on the hardware). The advantage of work is that, unlike runtime, it is deterministic; running the same model on the same hardware, with the same inputs parameters, will always result in the same measure of work. This means we can always compare runtimes of our models. As such, we use the measure of work as an indicator for runtime instead of directly using the runtime of the model. Work and runtime have a linear relationship.

To give an indication on the necessity of using work instead of runtimes, we run the full model 100 times for a 2-day schedule with the exact same parameters each time. We plot the solution time for each of these runs in Figure 5.1. We find an average solution time of 28.4 seconds with standard deviation of 3.0 seconds, whereas the value for work is 40.23 for every single run with 0.0 deviation. This confirms the importance of using work instead of runtime.

5.1.3 Dealing with cases where no solution is found

There are 4 experimental instances where the model is not able to find a feasible solution within the given work limit. For each case, we have the model report the reason for not being able to find a solution, as this allows us to check if it is due to infeasibility or simply exceeding the runtime. We find 0 reports of infeasibility, but do find several cases where no feasible solution is found within the runtime limit. It is difficult to numerically compare the models when only a few of them find solutions, so we remove the 4 scenarios where 1 or more of the models cannot find a solution within the set time limit from the general data analysis in Section 5.3, but look at these results separately to try and gain insights in Subsection 5.3.4.

5.2 Model Output Validation

For small instances with 3 days to optimize for, runtime is still relatively low and allows us to run the full model with multiple different settings and scenarios to validate the functionality of

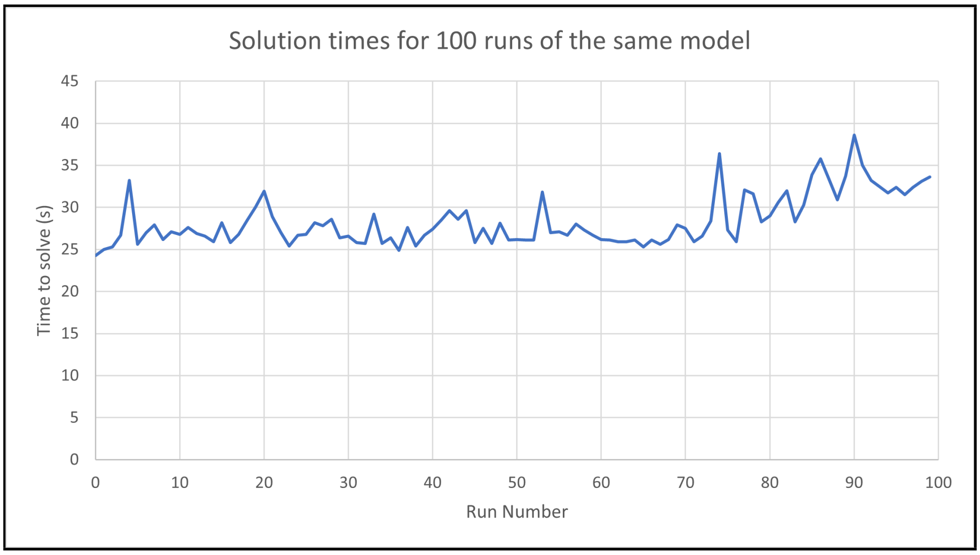

Figure 5.1: Runtimes of solving the exact same model 100 times in a row

the model. As the goal is validation of output, scenarios are varied based on input from supply planners to cover a range of possibilities.

In each of the cases, the solutions provided by the model are checked manually together with the supply planner to ensure that all constraints are met and the schedules are feasible in reality at the time the schedule is made. Having ran scenarios with high demand and respectively high, medium, and low starting inventories, we conclude the model provides feasible results. We do note that there are several aspects outside of scope that impact the optimality of scheduling. For example, the jetty is a resource used by many other product families, and the actual scheduling of the barge must thus also be done in accordance with them. It may happen that another barge gets priority and the production schedule must thus be changed as a result. Supply planners anticipate on some of these events and will thus make a slightly different schedule compared to our optimization.

We then move on to validating the output of the simplified models, which is something we do by using the solutions they yield and inputting the values for all decision variables as constraints into the full model. For the model without on-site product movements and reduced auxiliary variables this presents no problems, as all other aspects of the model match and we find the solutions to be feasible in the full model. For the models without waste product Z, there are a few complications. This is mostly due to the fact that in the full model, there is an explicit step for producing Z, which must take place between production of A and any other grade. In the model without Z, this step does not exist and instead we switch directly between A and any other grade. This also means we cannot directly feed the output from this simplification into the full model.

We know the schedule should still be feasible as there is a minimum batch size in place to ensure the management of waste product Z in terms of storage capacity. One dynamic this does not address accurately is the storage and blending of Z. In reality, Z is stored in a separate tank until it is blended into freshly-produced A. When we remove Z from the model, this process is also skipped. As a result, Z is not stored in a tank and instead effectively flows directly into tanks of A. This means the waste product spends less time in the system before being blended into A and sold. The models without waste product thus have higher volumes of A available at earlier stages, meaning that these volumes can also be sold earlier and lower average inventory. This means that it may be possible to create schedules that are infeasible in reality by selling *A* that would otherwise not yet be available. We validate whether this occurs by manual inspection of the solutions, where we check to see if the total volume of *A* drops below the volume of *Z* produced during a grade change. This is because the volume of extra grade *A* available due to the simplification is between 0 and the volume of *Z* produced during a grade change. Any time the total volume of *A* is within this range thus means it is possible, but not guaranteed, that technically unavailable product is sold. We note that this occurs only a few times for each scenario tested, and that there is space within the schedule to make the sale at a slightly later day with a manual correction to arrive at a feasible schedule. So as long as a supply planner is aware of this, output from this model is close enough to feasibility to still be used.

5.3 Results

In this section, we analyse the results of the different scenarios. We discuss the performance of the models regarding objective values, integrality gaps, and runtimes respectively.

5.3.1 Objective Value

When we look at the objective values of the different models in Figure 5.2, we notice the models achieve almost identical results. The full model, no pumping model and no auxiliary variables model find the exact same objective values for 2, 3, and 4 day schedules. Similarly, for the no waste and simplified model, we also observe the exact same objective values for 2, 3, and 4 day schedules, but note that these are lower than the other models. This is expected due to differences in the no waste model as discussed in Section 5.2, which also applies to the simplified model. We see that this difference in performance stays relatively constant and does not vary depending on the length of the schedule. This implies that the number of infeasibilities due to not modelling the waste explicitly are also constant and become relatively less impactful as the schedule is constructed for longer time periods.

When the schedule reaches lengths of 5 and 6 days, we start seeing a minor divergences in model performance. We note that the no pumping model finds an average solution 0.06% lower than the full model, and the reduced auxiliary model 0.02% lower than the full model. As these differences seem extremely small, we take a further look into the integrality gap achieved to gain more insight.

5.3.2 Integrality Gap

When looking at Figure 5.3, we confirm the identical model behaviour for 2, 3, and 4 day schedules as they all have integrality gaps of 0.0, which means the optimal solution is found in each instance. This also solidifies the validity of the models with no pumping and reduced auxiliary variables, as they find the exact same objective values for the optimal solutions as the full model, implying they have the same feasible region as the full model. It also confirms the infeasibility in the model without waste product, as it finds a lower objective value solution than the full model with both reporting an integrality gap of 0. This implies a larger solution space for the no waste model than that of the full model, where the optimal solution of the no waste model cannot lie in the solution space of the full model. It must thus be proposing solutions that are infeasible in reality.

When looking at the 5 and 6 day schedules, we see a linear increase in the integrality gap moving from 4 days through to 6 days, indicating the difficulty the models have with dealing with

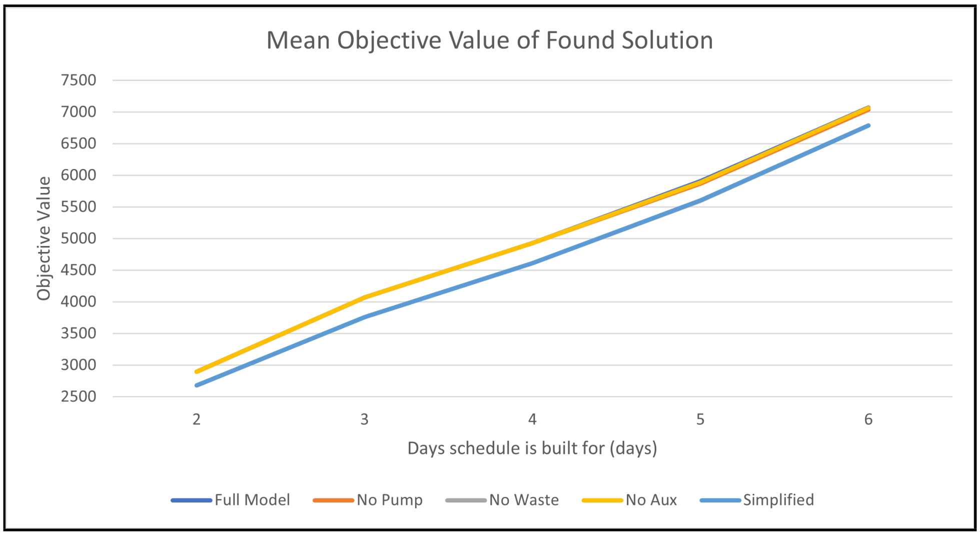

Figure 5.2: Average objective values found by each model depending on the number of days the schedule is run for

larger scale problems. We note that the no waste and simplified model have significantly lower integrality gaps, and that these gaps also increase at a lower rate. The full model and reduced auxiliary model have the highest integrality gaps, indicating that these two models do not scale well. The no pumping model performs in line with the simplified model for a 5 day schedule, but the integrality gap increases rapidly afterwards. This implies the removal of the pumping variable is a only a partial remedy to help the size of the problems we are able to solve, but does not structurally affect scalability.

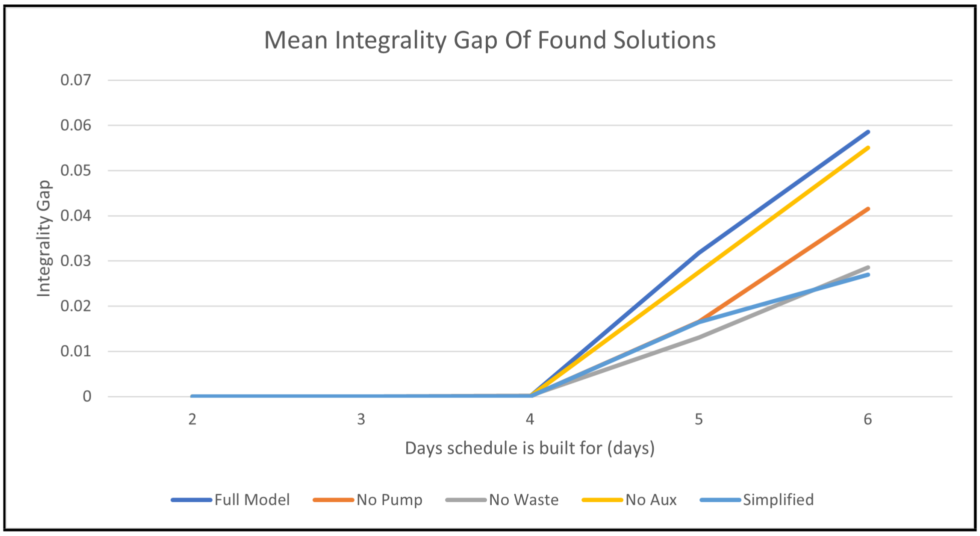

Figure 5.3: Average integrality gaps found by each model depending on the number of days the schedule is run for

5.3.3 Work

Work is used as an indication for the runtime of the model and we plot the average work required to arrive at the solution in Figure 5.4. The amount of work increases rapidly when generating schedules for 5 or 6 days, underscoring the difficulty of the models when dealing with larger scale problem instances. This is especially the case for the no waste model, which performs significantly worse than all other models for the 5 or 6 day schedules. One would expect this then also to influence the performance of the fully simplified model due to both using the no waste simplifications, but the fully simplified model performs in line with other models.

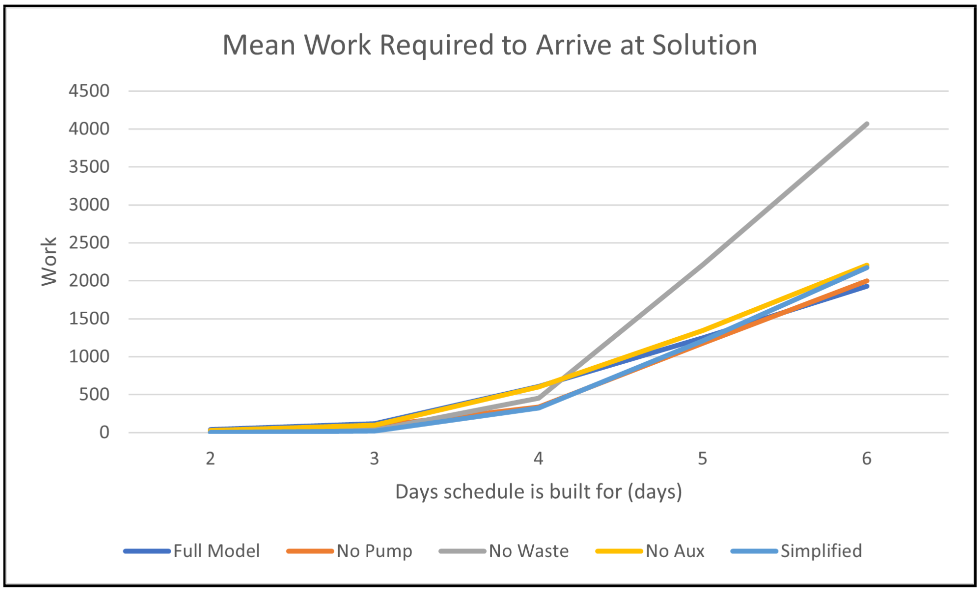

Figure 5.4: Average work required to find its best solution for each model depending on the number of days the schedule is run for.

To better differentiate the model performances, we plot the same data but without the no waste model in Figure 5.5. Interesting is that we see a lower average work for the no pumping and fully simplified models for small schedules of 2, 3, or 4 days. This is something we expect, as the simplifications are made with the goal of reducing the runtime. However, when the schedules become longer and the problem size increases, we see that the full model has the lowest average runtime. This would imply better scalability. We do know that the average integrality gap for the full model is also higher, which contradicts the better scalability. The lower work is most likely due to how the Gurobi solver terminates its optimizations. It does this by carrying out 'termination checks', which are snippets of code that check if any of the termination criteria have been reached. Part of this check is seeing how much work has been done since last improving the incumbent best solution. The higher integrality gap of the full model implies the solver struggles to improve on its solutions, which may lead Gurobi to terminate its optimization at an earlier stage than the other models in which it is still able to find improvements. This would explain the lower average work due to earlier termination as well as the higher integrality gap. From the models that include changes compared to the full model, we note that the no pumping model structurally has the lowest average work required to arrive at its final solution. The model with fewer auxiliary variables and the fully simplified model have higher amounts of work required. Overall the behaviour and performance is quite similar for all models when it comes to the work required, with the exception of the no waste model which requires significantly more work.

5.3.4 Unsolved Scenarios

As mentioned in Section 5.1.3, there are several scenarios for which some of the models cannot find a feasible solution within the given work limit. We give an overview of this in Table 5.1. In general, models are unable to find a feasible solution in cases with high demand and

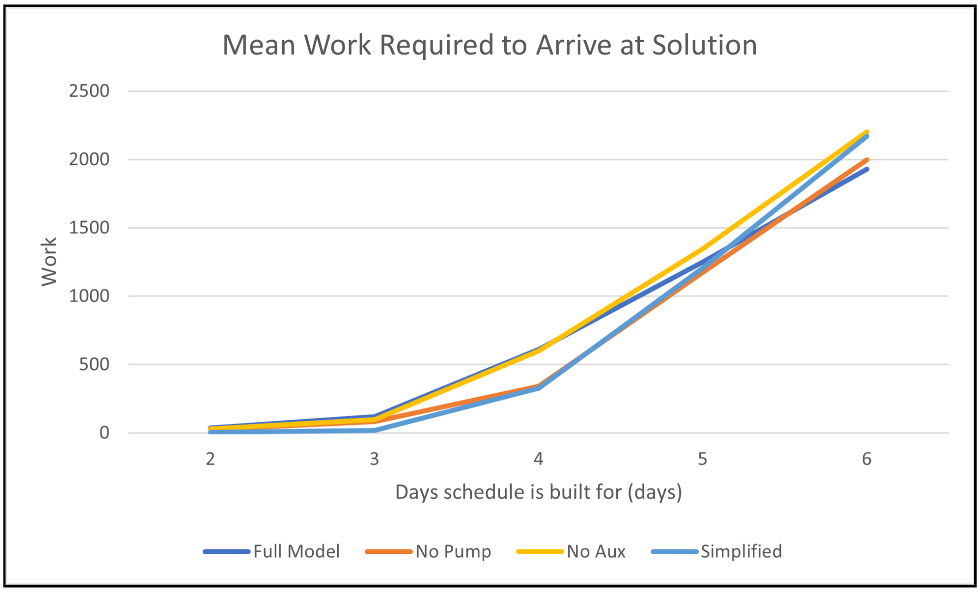

Figure 5.5: Average work required to find its best solution for each model depending on the number of days the schedule is run for, excluding the no waste model.

low starting inventories. High demand scenarios increase the problem size, as they increase the number of orders and thus also the number of decision variables corresponding to the fulfillment of these orders. As such, it follows that high demand scenarios are more challenging to solve than lower demand scenarios. Starting inventory does not influence the problem size, as it does not impact the number of decision variables. It does, however, impact the solution space due to the limited storage capacity of tanks and enough product needing to be available to meet demand. It seems that this smaller solution space, combined with a larger problem size, is what results in the models being unable to find a feasible solution. We note that there are feasible solutions, which we confirmed by letting the models run for extended periods of time and Gurobi not reporting out infeasibilities. These reports of infeasibilities do occur when we run at lower starting inventories, or when increasing the starting inventories above high (60% of tank capacity).

Nr Days	Demand	Starting Inventory	Models unable to solve scenario
5	High	Low	Full model, no auxiliary
6	Medium	Low	Full model, no pumping, no waste, no auxiliary, simplified
6	High	Low	No waste, no auxiliary, simplified
6	High	Medium	Full model

Table 5.1: Overview of models unable to find a feasible solution for a given scenario

From the initial data analysis on integrality gap, it appears that the full model and model with fewer auxiliary variables are the most difficult to solve. We see this reflected in Figure 5.6, which clearly shows that these models have the most number of scenarios they are unable to solve. We also see this reflected in Table 5.1, where these two models appear frequently. Interesting to note, however, is that the full model is able to solve the high demand and low starting inventory scenario, whereas the no waste, no fewer auxiliary and simplified models are not. In this scenario the full model and the no pumping model are able to find a single feasible solution.

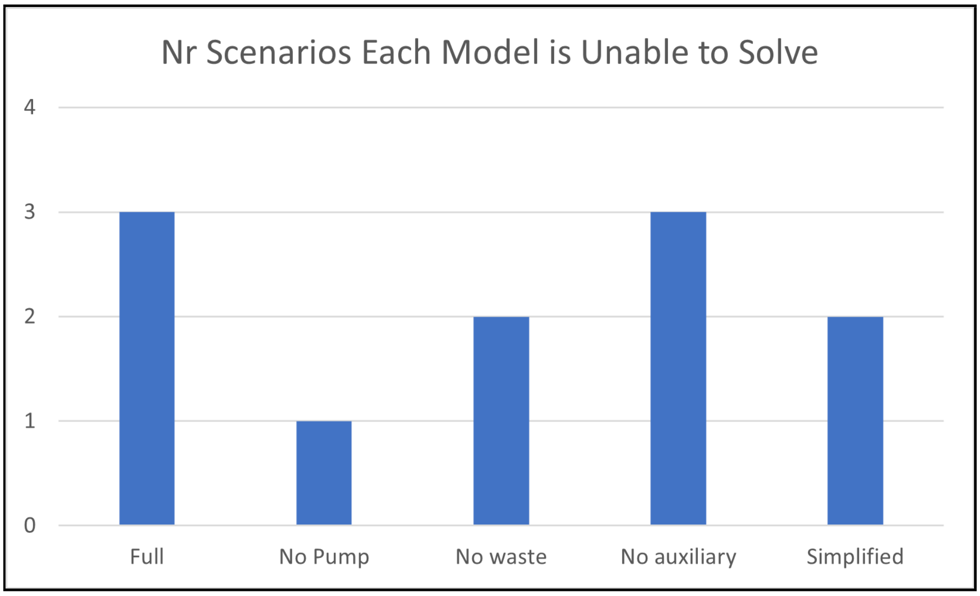

Figure 5.6: Number of scenarios in which each model is unable to find a solution.

These solutions are also feasible for the other models, and it is uncertain why the other models are not able to find this solution. When we increase the work limit, the other models are eventually able to find feasible solutions.

Here we also not that when other are unable to solve a scenario, we generally see that the models that are able to solve that scenario only ever find less than 3 feasible solutions within the time limit.

5.4 Comparison to current situation

In order to compare the performance of the model to the performance of current supply planners, we use historical data and schedules. We want to do a comparison for as large a schedule as possible, but are limited by the runtimes of the models, and choose a 2-week schedule as a result. This allows us to generate a feasible schedule in around 68 hours. We collect data for the monthly schedules, starting inventories, and demands from December 2022 through to May 2023. We cannot access historic schedules with the required levels of details more than 3 months back due to limitations in the software used for supply planning. It should be noted that petrochemical demand during this period was much lower than usual, which affected the way in which we choose the historic scenarios to have the model solve for comparison to the schedules made by human supply planners. We choose the 2-week period from the data range with the highest demand for this period, 1 with close to average demand for this period, and 1 that is roughly in between these two. This is done to compensate for the already overall lower demand in all data samples.

We then have the full model solve the selected 2-week scheduling problems given their demands and production quantities. We then manually calculate the objective function of the human made schedules, as the data contained inventory levels as well as transfers to off-site storage via barge. Overall, we find that in each of the scenarios, our model yields an objective value that is on average 4.2% lower. It achieves this by taking a slightly more aggressive approach on production scheduling, delaying it as much as possible and shipping less product via barges. Instead, the model tries to fulfill as much demand as possible from the on-site production tanks. This makes sense, as using barges costs money and increases the objective value. Besides this, it also means that the product is in inventory for a longer period of time and again increases costs and the objective value. The supply planners tend to avoid such an aggressive on-site sales strategy, as it can quickly lead to congestion or other logistic issues on-site, and will thus prefer shipping more product to the off-site tanks at a slightly higher cost. This also gives them a bit more flexibility in the form of ullage in tanks, which can come in handy in case of unexpected issues such as machine failure. As a result, supply planners would be hesitant to use the proposed schedules as is and instead adapt them to be slightly more conservative and also costly.

6 CONCLUSIONS AND RECOMMENDATIONS

The original research question we set out to answer is: “How can Shell Chemicals Europe optimize the daily production, transportation, and sales scheduling of a single production unit to minimize inventory and transportation costs while maintaining service levels?”. We draw our conclusions in Section 6.1 and discuss these in Section 6.2. We then make several recommendations in Section 6.3 and propose areas for further research in Section 6.4.

6.1 Conclusions

6.1.1 Model Performance

A main concern we discuss during the modelling phase in Chapter 4 is the runtime of the model, and we attempt several simplifications to try and remedy this issue. We see in Chapter 5 that we are able to generate feasible and optimal schedules only for small sized problems using the full model based on the Resource-Task Network formulation. Scalability is the main issue when it comes to performance, as the runtimes become exceedingly high very quickly as we increase the problem size by creating schedules for longer time periods. We are able to improve the runtime of the models by making some simplifications, such as removing the variable for pumping product around on-site as well as by removing the explicit modelling of waste product. However, we are not able to generate schedules for a full month with a runtime anywhere close to the desired 15 minutes.

We see that these simplifications do not significantly harm the optimality of the results. The no pumping model solution space is a subset of the full model’s solution space, but we see that when both report an integrality gap of 0, their objective values are equal. A proper comparison of the no waste model and full model is a bit more difficult due to the validity of the no waste model as described in Section 5.2. Overall, the solution space of the no waste model is smaller than that of the full model, but contains some solutions that are not part of the full model’s solution space and technically not feasible. The no waste model solution space is smaller due it containing a minimum batch size constraint the full models doesn’t, restricting behaviour. However, making the no waste schedules feasible is a relatively simple operation that only requires a few orders of grade *A* to be fulfilled slightly later. So even when reducing the size of the solution space through simplifications in the model, we are generally not hurting model performance in terms of objective function or integrality gap. This indicates we can rely heavily on these simplifications and could potentially reformulate the model around this. Overall, we can conclude that despite making simplifications that reduce runtime, we are not able to get it anywhere close to the desired 15 minutes to generate a 30 day schedule. We can see that the simplifications do not significantly influence the scalability of the model as runtime increases rapidly as the schedules are built for more days. As such, we conclude that a MIP model formulated as a Resource-Task Network does not seem to be the desired approach for Shell’s desired optimization.

Our models use a large number of binary variables, which is known to hurt runtime and scalability. We do this as many of the behaviours we need to model contain a binary element within them, such as the testing constraint or use of barges. We also use a relatively small time scale of hourly time steps, which again is chosen such that the system is modeled accurately and we do not unnecessarily reduce the solution space. However, we generally find that simplifications and reduced accuracy improve the runtime without hurting the objective value and thus quality of the solutions. This implies that feasibility drives finding solutions more so than optimality, which is a conclusion that resonates with the Shell supply planners as they generally describe the system as being heavily constrained. As such, using a modelling approach that focuses on finding feasible solutions over optimal solutions is considered a valid approach worth taking.

6.1.2 Auxiliary Variables

One of the simplifications tested is the removal of several auxiliary variables from the full model. We remove these variables to reduce the number of decision variables without affecting the solution space or problem structure overall. We can see that this slightly improves model performance regarding objective value and integrality gap, however the impact is limited when compared to the other simplifications. Removing unnecessary variables from the model is something that solvers such as Gurobi already do during the pre-solve phase, but the improved performance of the reduced auxiliary model shows that this pre-solving phase is not able to achieve the same results as proper modelling decisions. We show that reformulating constraints with fewer decision variables will improve runtime, given that we do not change the structure of the constraints.

6.2 Discussion

6.2.1 Improvement over Current Situation

We see that our model does slightly improve on the scheduling currently done by supply planners. However, given the current performance of the models regarding their runtime and difficulty in arriving at solutions for large scale problems, we recommend further development of the models before using and implementing them within Shell. The models also do not take into account many aspects that supply planners do, for example regarding the jetty scheduling or preferences on production from the operators. Combining this with the slight improvement over human planning of around 4%, it seems the added value of the optimization is limited. As this improvement is achieved with the full model capable of exploring all feasible schedules, it would appear that there is simply limited opportunity for improvement within the scheduling. One explanation for this is the fact that the entire system is heavily-constrained due to Shell continuously evaluating performance and cutting costs where possible. Over time, this leads to a system that contains limited room for error or sub-optimal scheduling. For example, tank space is simply limited, and thus does not allow supply planners to maintain excessive amounts of inventory. It thus remains questionable whether any optimization approach will be able to find better solutions and more savings for Shell.

6.2.2 Integration of Production, Transportation, and Sales

In the literature review of Chapter 3, we find that papers indicate stronger results when integrating multiple stages such as production, transportation, and sales. We also do this integration for the Shell system as it is required for validity of the schedules. However, it also complicates the model by increasing many dependencies between variables, as well as introducing many additional decision variables. We know from the improved performance of the no waste model that reducing the number of decision variables will help the scalability of our optimization. If there is a way to separate the decision variables and solve the scheduling separately, this may improve the scalability.

6.2.3 Resource-Task Network Formulation

While most literature in Chapter 3 on implementation of RTN claimed applicability in large scale problems, we do not find this in our research. This could be because we define our model as a NP-hard combinatorial optimization problem. This is not inherent to RTN formulations, of which many implementations use continuous variables. If we are able to move from a combinatorial optimization problem towards a more continuous formulation, we may be able to significantly reduce the runtime of the model.

By using the Resource-Task Network formulation, we note that we are able to properly capture all system behaviour and generate feasible schedules. However, it also leads to us defining an excess of auxiliary variables, and model performance is improved when removing the unnecessary variables. We create these auxiliary variables based on the identification of tasks and resources within the system, which is a general step found in literature. One step that is not identified is this step of critical inspection on the need for all of these auxiliary variables, and seeing which ones could be replaced by other simple linear constraints. So long as this does not introduce non-linearity or create otherwise complex constraints, we do not find a downside to taking this step. A further benefit of this simplification, over the other simplifications we make in this research, is that it does not harm the generalizability of the model.

6.2.4 Propagation

Theoretically, it would be possible to use the 5 day schedules and propagate them 6 times to generate 30 day schedules. We can solve a small 5 part day for the scheduling problem, then use the results from that solution to give input for the next 5 days in terms of inventory levels. There are several difficulties when implementing this, which have to do with continuity across the different partial problems. Primarily, we need to pre-define our production quantities for each partial problem. This requires us to already make a macro schedule for production, and the model then only solves the micro aspects of the scheduling itself. However, the macro scheduling is expected to influence optimality much more than the micro scheduling, and would thus ideally require another model to generate this. Second to this, feasibility across partial schedules becomes difficult to guarantee. For example, there are limited monthly slots at the jetty for barges. Early partial problems may use the barges to generate local optimal solutions but lead to infeasible production schedules in subsequent partial problems due to the jetty slots having been used up. This is also why we want to generate 30 day schedules in the first place, as we need to take into account all actions across a month to ensure feasibility. As such, propagation of multiple smaller schedules to generate a large schedule is not a suitable solution.

6.3 Recommendations

Although we find limited improvement when compared to the current human scheduling in terms of direct financial impact, another aspect to consider is the ability to find feasible schedules. Supply planners spend significant portions of time on creating and changing the schedule throughout the month, as finding a feasible planning can be a tough challenge. They also need to be able to communicate how potential changes in demand will impact the schedule, for example if the sales department finds and additional short term opportunity and wants to know if there is enough supply to accommodate this. As such, the tool does offer value by being able to check

As such, developing a model that is able to generate high quality feasible schedules (or report infeasibilities) is still of value if the runtime is relatively low. We already see that the fully simplified model performs better than the original full model, and does not compromise much on the quality with regards to objective value. This indicates that we can leverage aspects such as the heuristic on minimum batch size for grade *A* without heavily losing out on optimality. We would recommend investigating to see if the entire model could be reformulated around these heuristics. For example, we could assume a set production sequence of *A, B, C, A, C, D*. This is something the supply planner frequently does. We could then alter the production decision variables from a binary variable to a continuous variable denoting the quantity of each batch. This would remove a significant amount of binary variables from the model. It would also mean that the model is no longer generalizable, and instead needs to be developed per production unit and require a lot of domain knowledge.

6.4 Future Research

Runtime of our full RTN model can be reduced using simplifications. However, two of the three simplifications we use are case specific and can not be generalized. Being able to extend the model for other production units is a desirable property and one of the strengths of RTN formulations. Further research can be focused on finding simplifications that reduce runtime but do not harm extendability of the model. For example, we could increase the size of the time steps from 1 hour to 2 hours. This introduces a trade-off; larger time steps reduce the number of decision variables, but also limit the number of solutions that can be explored. This is because with 1 hour time steps we can produce for 1 hour and not produce the next. If the time step is 2 hours, we have to produce the full 2 hours or not at all. We are thus artificially reducing the solution space, which may mean optimal solutions are not explored. Another possible simplification that can be explored is aggregating the 3 tanks for grade *A* into a single tank.

Another area for further research would be how to separately model different aspects of an integrated system. As mentioned in Section 6.2.2, separating the aspects of production, transportation, and sales should reduce the complexity of the model. The challenge for our system is that these aspects influence the feasibility of the overall schedule and do need to be taken into account. Finding a method of incorporating aspects of feasibility for each aspect, without explicitly modelling these aspects, would help in the creation of smaller separate models.

As feasibility appears to be a large challenge within this system, it may also be of interest for Shell to investigate the benefits of changing their tank footprint. Fixed costs of renting and maintaining the storage tanks was not taken into account within this research, but do play a significant role when it comes to the cost of the system. It may be that slightly increasing the tank footprint allows for lower cost scheduling. Further developing our model should allow for impact comparison of tactical decisions such as these, and may allow Shell to find long term savings. By optimizing schedules under a range of different parameters, we can effectively simulate the expected impact of changes made to the setup of the entire system. This allows for evaluating the value of renting additional (or fewer) tanks, adding an extra production grade, or adding the ability to blend products off-site.

We ultimately conclude that Resource-Task Networks are not the right approach for the Shell case due to scalability issues. Future research into alternative approaches may lead to stronger results. As mentioned in Section 6.3, heuristics are definitely an area of interest, both in model formulation as well as a solution method. Another interesting approach may be reinforcement learning, which is capable of dealing with large problems in complex environments. Shell has enough data and structure to train the model, and the clear costs and penalties will allow for rewarding and punishing model decisions. A challenge for reinforcement learning is dealing with a large action space, which means the model has too many possible actions it can take. Having integer variables such as in our model will play an advantage here, as it limits the size of the action space. Implementation and maintenance of the model will be much more challenging and costly, and considering the scheduling improvements achieved by our model, may not be worth it.

REFERENCES

[1] Ana Maria Sarmiento and Rakesh Nagi. A review of integrated analysis of production-distribution systems. *IIE Transactions (Institute of Industrial Engineers)*, 31:1061–1074, 1999.

[2] Anshuman Gupta and Costas D. Maranas. Managing demand uncertainty in supply chain planning. *Computers and Chemical Engineering*, 27:1219–1227, 9 2003.

[3] Leandro C. Coelho, Jean François Cordeau, and Gilbert Laporte. Thirty years of inventory routing. *Transportation Science*, 48:1–19, 2014.

[4] Raf Jans and Zeger Degraeve. Modeling industrial lot sizing problems: A review. *International Journal of Production Research*, 46:1619–1643, 3 2008.

[5] Karla E. Bourland and Candace A. Yano. A comparison of solution approaches for the fixed-sequence economic lot scheduling problem. *IIE Transactions (Institute of Industrial Engineers)*, 29:103–108, 1997.

[6] Lotte van Hezewijk, Nico Dellaert, Tom Van Woensel, and Noud Gademann. Using the proximal policy optimisation algorithm for solving the stochastic capacitated lot sizing problem. *International Journal of Production Research*, 2022.

[7] Jens Ehm and Michael Freitag. The benefit of integrating production and transport scheduling. *Procedia CIRP*, 41:585–590, 2016.

[8] Heeman Lee, Jose M Pinto, Ignacio E Grossmann, and Sunwon Park. Mixed-integer linear programming model for refinery short-term scheduling of crude oil unloading with inventory management, 1996.

[9] Chung-Yee Lee. Machine scheduling with an availability constraint, 1995.

[10] Brenno C. Menezes, Jeffrey D. Kelly, and Ignacio E. Grossmann. Logistics optimization for dispositions and depooling of distillates in oil-refineries: closing the production scheduling and distribution gap. *Computer Aided Chemical Engineering*, 43:1135–1140, 1 2018.

[11] Peter M. Verderame, Josephine A. Elia, Jie Li, and Christodoulos A. Floudas. Planning and scheduling under uncertainty: A review across multiple sectors. *Industrial and Engineering Chemistry Research*, 49:3993–4017, 5 2010.

[12] R P J Broch. Multiple approaches to solve tank scheduling problems, 2010.

[13] F. Pezzella, G. Morganti, and G. Ciaschetti. A genetic algorithm for the flexible job-shop scheduling problem. *Computers and Operations Research*, 35:3202–3212, 10 2008.

[14] E. N. Pistikopoulos, Ana Barbosa-Povoa, Jay H. Lee, Ruth Misener, Alexander Mitsos, G. V. Reklaitis, V. Venkatasubramanian, Fengqi You, and Rafiqul Gani. Process systems engineering – the generation next? *Computers and Chemical Engineering*, 147, 4 2021.

[15] Ali Allahverdi, C. T. Ng, T. C.E. Cheng, and Mikhail Y. Kovalyov. A survey of scheduling problems with setup times or costs. *European Journal of Operational Research*, 187:985–1032, 6 2008.

[16] Iiro Harjunkoski, Christos T. Maravelias, Peter Bongers, Pedro M. Castro, Sebastian Engell, Ignacio E. Grossmann, John Hooker, Carlos Méndez, Guido Sand, and John Wassick. Scope for industrial applications of production scheduling models and solution methods. *Computers and Chemical Engineering*, 62:161–193, 3 2014.

[17] E Kondili, C C Pantelides, and R W H Sargent. A general algorithm for short-term scheduling of batch operations-i. milp formulation, 1993.

[18] G Schilling and C C Pantelides. A simple continuous-time process scheduling formulation and a novel solution algorithm, 1996.

[19] Pedro M. Castro, Ana P. Barbosa-Póvoa, Henrique A. Matos, and Augusto Q. Novais. Simple continuous-time formulation for short-term scheduling of batch and continuous processes. *Industrial and Engineering Chemistry Research*, 43:105–118, 1 2004.

[20] Ignacio E. Grossmann. Challenges in the application of mathematical programming in the enterprise-wide optimization of process industries. *Theoretical Foundations of Chemical Engineering*, 48:555–573, 10 2014.

[21] Frank Hutter, Lin Xu, Holger H. Hoos, and Kevin Leyton-Brown. Algorithm runtime prediction: Methods & evaluation. *Artificial Intelligence*, 206:79–111, 2014.

[22] Ross J.W. James and Bernardo Almada-Lobo. Single and parallel machine capacitated lotsizing and scheduling: New iterative mip-based neighborhood search heuristics. *Computers and Operations Research*, 38:1816–1825, 12 2011.

[23] Gianni Codato and Matteo Fischetti. Combinatorial benders' cuts for mixed-integer linear programming. *Operations Research*, 54:756–766, 7 2006.

[24] Christian D. Hubbs, Can Li, Nikolaos V. Sahinidis, Ignacio E. Grossmann, and John M. Wassick. A deep reinforcement learning approach for chemical production scheduling. *Computers and Chemical Engineering*, 141, 10 2020.

[25] Yi Chi Wang and John M. Usher. Application of reinforcement learning for agent-based production scheduling. *Engineering Applications of Artificial Intelligence*, 18:73–82, 2 2005.

[26] Venkatachalam Avadiappan and Christos T. Maravelias. State estimation in online batch production scheduling: concepts, definitions, algorithms and optimization models. *Computers and Chemical Engineering*, 146, 3 2021.

[27] Virginie Gabrel, Cécile Murat, and Aurélie Thiele. Recent advances in robust optimization: An overview. *European Journal of Operational Research*, 235:471–483, 6 2014.

[28] Dimitris Bertsimas and Aurélie Thiele. A robust optimization approach to inventory theory. *Operations Research*, 54:150–168, 1 2006.

[29] Bram L. Gorissen, Ihsan Yanikoğlu, and Dick den Hertog. A practical guide to robust optimization. *Omega (United Kingdom)*, 53:124–137, 6 2015.

[30] Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Operations Research*, 52:35–53, 1 2004.

[31] Nikolaos V. Sahinidis. Optimization under uncertainty: State-of-the-art and opportunities. *Computers and Chemical Engineering*, 28:971–983, 6 2004.

[32] Guodong Liu, Yan Xu, and Kevin Tomsovic. Bidding strategy for microgrid in day-ahead market based on hybrid stochastic/robust optimization. *IEEE Transactions on Smart Grid*, 7:227–237, 1 2016.

[33] Tao Zhang, Yue Wang, Xin Jin, and Shan Lu. Integration of production planning and scheduling based on rtn representation under uncertainties. *Algorithms*, 12, 6 2019.

[34] J Balasubramanian and I E Grossmann. Scheduling optimization under uncertainty: an alternative approach, 2002.

[35] James B. Rawlings and Kenneth R. Muske. The stability of constrained receding horizon control. *IEEE Transactions on Automatic Control*, 38:1512–1516, 1993.

[36] Chuqian Zhang, Jiyin Liu, Yat wah Wan, Katta G. Murty, and Richard J Linn. Storage space allocation in container terminals. *Transportation Research Part B*, 37:883–903, 7 2003.

[37] Suresh Sethi and Gerhard Sorger. A theory of rolling horizon decision making *, 1991.

[38] Yachao Dong and Christos T. Maravelias. Terminal inventory level constraints for online production scheduling. *European Journal of Operational Research*, 295:102–117, 11 2021.

[39] Chao Ning and Fengqi You. Optimization under uncertainty in the era of big data and deep learning: When machine learning meets mathematical programming. *Computers and Chemical Engineering*, 125:434–448, 6 2019.

[40] C A Floudas and X Lin. Continuous-time versus discrete-time approaches for scheduling of chemical processes: A review. *Computers and Chemical Engineering*, 28:2109–2129, 10 2004.

[41] Zhenya Jia, Marianthi Ierapetritou, and Jeffrey D. Kelly. Refinery short-term scheduling using continuous time formulation: Crude-oil operations. *Industrial and Engineering Chemistry Research*, 42:3085–3097, 6 2003.

[42] J M Pinto, M Joly, and L F L Moro. Planning and scheduling models for refinery operations, 2000.

[43] Iftekhar A Karimi and Conor M Mcdonald. Planning and scheduling of parallel semicontinuous processes. 2. short-term scheduling, 1997.

A FULL MODEL

A.1 Sets/Indices

$K = Set\ of\ all\ tanks$
$K^{on} \subset K = Set\ of\ all\ onsite\ tanks$
$K^{off} \subset K = Set\ of\ all\ offsite\ tanks$
$K_i^{on} \subset K = Set\ of\ all\ onsite\ tanks\ containing\ grade\ i$
$K_i^{off} \subset K = Set\ of\ all\ offsite\ tanks\ containing\ grade\ i$
$T = Set\ of\ all\ time\ indices\ from\ t = 0\ to\ t^{max}$
$I = Set\ of\ all\ grades$
$B = Set\ of\ barges\ that\ can\ be\ scheduled$
$O = Set\ of\ orders\ to\ fulfill$
$O_{soft} \subset O = Set\ of\ orders\ that\ are\ uncertain$
$O_{hard} \subset O = Set\ of\ orders\ that\ are\ certain$
$O^k \subset O = set\ of\ orders\ that\ can\ be\ fulfilled\ from\ tank\ k$
$LC_b^{on} \subset T = Set\ of\ times\ in\ period\ during\ which\ a\ barge\ can\ be\ loaded$
$LC_b^{off} \subset T = Set\ of\ times\ in\ period\ during\ which\ product\ can\ be\ offloaded\ from\ barge$
$T^{days} \subset T = Set\ of\ times\ that\ mark\ the\ start\ of\ a\ new\ day$
$T^{weekends} \subset T = Set\ of\ times\ that\ mark\ the\ start\ of\ a\ weekend$

A.2 Parameters

$NrDays = number\ of\ days\ to\ create\ a\ schedule\ for$
$t^{max} = end\ of\ time\ horizon$
$t_{co} = time\ required\ to\ changeover\ between\ two\ grades$
$t_{barge} = time\ it\ takes\ for\ a\ barge\ to\ travel\ between\ the\ sites$
$t_{test} = time\ it\ takes\ to\ test\ whether\ grade\ meets\ specifications$
$t_z = time\ it\ takes\ to\ produce\ the\ required\ volume\ of\ Z\ between\ grade\ switches$
$v_Z = volume\ of\ waste\ product\ created\ from\ switching\ grades$
$f_Z = fraction\ of\ waste\ product\ that\ can\ be\ blended\ back\ into\ grade\ A$
$p_k = volume\ of\ a\ grade\ produced\ in\ an\ hour\ into\ tank\ k$
$HSK_i^{max} = max\ HSK\ volume\ that\ may\ be\ produced$
$HSK_i^{min} = min\ HSK\ volume\ that\ must\ be\ produced$
$r_i^k = volume\ of\ grade\ i\ that\ can\ be\ pumped\ between\ onsite\ tanks\ in\ 1\ hour$
$r^b = volume\ of\ any\ product\ that\ can\ be\ pumped\ onto\ a\ barge\ in\ 1\ hour$
$a = number\ of\ orders\ that\ can\ be\ loaded\ via\ the\ gantry\ per\ day$
$j = number\ of\ barges\ that\ can\ be\ loaded\ via\ the\ jetty\ per\ month$
$c_{fix} = fixed\ cost\ of\ scheduling\ a\ barge$
$c_{var} = variable\ cost\ of\ scheduling\ a\ barge\ per\ tonne\ of\ product$
$c_{dem} = demurrage\ cost\ of\ barge\ per\ day$
$c_{inv} = cost\ of\ holding\ one\ tonne\ of\ product\ in\ inventory\ for\ one\ day$

$c_{ord} = cost\ of\ missing\ an\ order$
$g_k = grade\ stored\ in\ tank\ k$
$k_z = tank\ that\ holds\ waste\ product\ z$
$CAP_k = max\ capacity\ of\ tank\ k$
$INI_k = starting\ volume\ product\ in\ tank\ k$
$u_b = max\ capacity\ of\ barge\ b$
$TW_o^{start} = Time\ at\ which\ an\ order\ becomes\ available\ for\ completion$
$TW_o^{end} = due\ date\ of\ order$
$s_o = whether\ order\ o\ can\ be\ fulfilled\ from\ onsite\ tanks\ only\ or\ not$
$M_1,\ M_2 = Big\ M\ values\ used\ for\ constraints$
$DEM = Realization\ of\ soft\ demand\ at\ end\ of\ month$
$h_k = daily\ holding\ cost\ of\ a\ single\ unit\ of\ inventory\ stored\ in\ tank\ k$

A.3 Decision Variables

$X_{k,t} \in \{0,1\} = Whether\ we\ produce\ into\ onsite\ tank\ k\ at\ time\ t.$
$Y_b\ \in 0,1 = Whether\ we\ use\ barge\ b\ or\ not$
$Y^{on}_{k,b,t}\ \in \{0,1\} = Whether\ we\ pump\ product\ from\ onsite\ tank\ k\ onto\ barge\ b\ at\ time\ t$
$Y^{off}_{b,k,t}\ \in \{0,1\} = Whether\ we\ pump\ product\ from\ barge\ b\ onto\ offsitetank\ k\ at\ time\ t$
$Z_{o,k,t} \in \{0,1\} = Whether\ we\ fulfill\ order\ o\ from\ tank\ k\ at\ time\ t$
$W_{k,\ k',\ t} \in \{0,1\} = Whether\ we\ move\ product\ from\ onsite\ tank\ k\ to\ onsite\ tank\ k' at\ time\ t$

A.4 Auxiliary Variables

$R_{k,t} = Volume\ of\ product\ in\ tank\ k\ at\ time\ t$
$U_{b,t} = Volume\ of\ product\ on\ barge\ b\ at\ time\ t$
$L_{i,t} \in \{0,1\} = Whether\ grade\ i\ was\ the\ previous\ grade\ produced\ at\ time\ t,\ i \in \{A,B,C,D\}$
$P_t^1 \in \{0,1\} = Whether\ we\ need\ to\ have\ finished\ producing\ Z\ at\ time\ t\ for\ producing\ A$
$P_t^2 \in \{0,1\} = Whether\ we\ need\ to\ have\ finished\ producing\ Z\ at\ time\ t\ for\ producing\ B/C/D$
$Q_{k,t} \in \{0,1\} = Whether\ we\ have\ Z\ available\ for\ blending\ if\ we\ are\ producing\ A\ into\ tank\ k$

A.5 Objective Function

The objective function includes costs for inventory and transportation. First we include the holding costs of inventory, which is paid over the inventory remaining in tanks and barges at the end of the day, the expression for this is:

$$\sum_{\delta=0}^{NrDays} \left(\sum_{k\in K} h_k R_{k,24\delta} + \sum_{b\in B} h_k U_{b,24\delta} \right) \tag{A.1}$$

Second, we also include the costs of using barges, which has a minimum fixed tariff paid regardless of the volume shipped, and an additional variable cost depending on the volume shipped. It is expressed by:

$$\sum_{b\in B} \left(c_{fix} Y_b + \sum_{t\in T} \sum_{k\in K^{on}} c_{var} Y^{on}_{k,b,t} \right) \tag{A.2}$$

Combining these expressions yields the following objective function:

$$\min \sum_{\delta=0}^{NrDays} \left(\sum_{k\in K} h_k R_{k,24\delta} + \sum_{b\in B} h_k U_{b,24\delta} \right) + \sum_{b\in B} \left(c_{fix} Y_b + \sum_{t\in T} \sum_{k\in K^{on}} c_{var} Y^{on}_{k,b,t} \right) \tag{A.3}$$

A.6 Constraints

Production Constraints

As mentioned in the problem description, at the start of the month a separate optimization takes place to determine the total volume of each grade that should be produced. Forcing exactly this volume restricts the feasible solution space to be very small, so we instead use a minimum and maximum total production volume per grade to allow for some flexibility:

$$p_k \sum_{k\in K_i^{on}} \sum_{t\in T} X_{k,t} \leq HSK_i^{max} \quad \forall i \in I,\ i \neq Z \tag{A.4}$$

$$p_i \sum_{k\in K_i^{on}} \sum_{t\in T} X_{k,t} \geq HSK_i^{min} \quad \forall i \in I,\ i \neq Z \tag{A.5}$$

A hard constraint that cannot be violated is the fact that only a single grade can be produced at any given time (no simultaneous production of multiple grades):

$$\sum_{k\in K^{on}} X_{k,t} \leq 1 \quad \forall t \in T \tag{A.6}$$

Next, we formulate a constraint that specifies the changeover time required when changing the grade that is produced. This is formulated such that if a certain grade is produced, no other grade may be produced for the next hours until the changeover time has passed. This forms many partially overlapping constraints, which ensure a strong formulation; if one constraint in the solution is violated, the solution will still be close to being feasible.

$$\sum_{k\in K_i^{on}} X_{k,\,t} + \sum_{k'\in K'} \sum_{\delta=0}^{t_{co}} X_{k',\,t+\delta} \leq 1 \quad \forall t \in T,\ \forall i \in I,\ K' = K^{on} \setminus \{K_i^{on}\} \tag{A.7}$$

A more complicated matter is constraining production to force production of waste product when switching away from, or back to, producing grade *A*. The difficulty of enforcing this behavior lies in the fact that it is only required in a very specific set of circumstances to occur. There are two scenarios: We have produced *B/C/D* previously, and are now going to produce *A*, or we have produced A previously and are now going to produce *B/C/D*. Simply put, if a grade was produced at $t-1$, it must be the grade produced previously:

$$L_{i,t} \geq \sum_{k\in K_i^{on}} X_{k,t-1} \quad \forall i \in I\setminus\{Z\},\ \forall t \in T \tag{A.8}$$

However, in case no grade was produced at $t-1$ (e.g. during changeover or production of grade *Z*) we need another constraint for the previous grade produced. So first, we define that there must always be a grade that was produced previously:

$$\sum_{i\in I/Z} L_{i,t} = 1 \quad \forall t \in T \tag{A.9}$$

We can then use the process of elimination to deduce if a grade was produced previously. If no other grade was produced at $t-1$, and no other grade was defined as the previous grade produced at $t-1$, then this grade must have been the previous grade produced at $t-1$ and is still the previous grade produced at t.

$$L_{i,t} + \sum_{i' \in I'} \left(L_{i',t-1} + \sum_{k \in K_{i'}^{on}} X_{k,t-1} \right) \geq 1 \quad \forall i \in I,\ \forall t \in T,\ I' = I \backslash \{i, Z\} \tag{A.10}$$

Now that we have defined the previous grade produced, we can use this to determine whether grade *Z* has to have been produced before being able to produce a certain grade. There are two cases: We have just produced *B/C/D* and now want to produce *A*, for which we use the variable P_t^1, and we have just produced *A* and now want to produce *B/C/D*, for which we use the variable P_t^2. If we did not produce *B/C/D*, we do not need to produce *Z* before producing *A*. This is expressed by P_t^1 being forced to 0:

$$\sum_{i \in \{B,C,D\}} L_{i,t} \geq P_t^1 \quad \forall t \in T \tag{A.11}$$

Similarly for P_t^2, only with having produced *A*:

$$L_{A,t} \geq P_t^2 \quad \forall t \in T \tag{A.12}$$

It may also be that we are not producing *A*, and we do not need to produce *Z*. In this case, P_t^1 should be forced to 0.

$$\sum_{k \in K_A^{on}} X_{k,t} \ \geq P_t^1 \quad \forall t \in T \tag{A.13}$$

Similarly for P_t^2, only with producing *B/C/D*:

$$\sum_{i \in \{B,C,D\}} \sum_{k \in K_i^{on}} X_{k,t} \ \geq P_t^2 \quad \forall t \in T \tag{A.14}$$

Now we still need to force P_t^1 to 1 in case we produced *B/C/D* previously and want to produce *A*:

$$\sum_{i \in \{B,C,D\}} L_{i,t} + \sum_{k \in K_A^{on}} X_{k,t} - P_t^1 \leq 1 \quad \forall t \in T \tag{A.15}$$

Similarly for P_t^2, only with producing *B/C/D* and having produced *A* previously:

$$L_{A,t} + \sum_{i \in \{B,C,D\}} \sum_{k \in K_i^{on}} X_{k,t} + P_t^2 \leq 1 \quad \forall t \in T \tag{A.16}$$

With this set of constraints, we can now recognize when to force production of *Z* before starting to produce a different grade. We force production of *Z* for the required amount of timesteps (t_z) to ensure enough volume has been produced before starting production after a grade change: Force production of *Z* in case we want to produce A after having produced *B/C/D*:

$$\sum_{\delta=1}^{t_z} X_{K_z,t+\delta} \geq t_z P_t^1 \quad \forall t \in T \tag{A.17}$$

Force production of *Z* in case we want to produce *B/C/D* after having produced *A*:

$$\sum_{\delta=1}^{t_z} X_{K_z,t+\delta} \geq t_z P_t^2 \quad \forall t \in T \tag{A.18}$$

Having constrained the production of *Z*, we now need to also ensure that it is blended away into A afterwards. We introduce an additional variable Q_k, t. This variable becomes 1 if we have *Z* in inventory and are producing *A* into tank k. This variable is required in the resource balance constraints to track inventory levels in each tank. We specify the tank into which it is being blended as this allows us to keep proper track of it in the inventory balance constraints later. First, we constrain it such that we cannot blend *Z* into *A* if there is not enough *Z* ($Q_{k,t} = 0$):

$$r_z^k Q_{k,t} \leq R_{k_z,t} \quad \forall t \in T, \ \forall k \in K_A^{on} \tag{A.19}$$

Next, we constraint it such that we cannot blend Z into A if no A is being produced into that tank ($Q_{k,t} = 0$):

$$Q_{k,t} \leq X_{k,t} \quad \forall t \in T, \ \forall k \in K_A^{on} \tag{A.20}$$

Last, we must blend Z into A if it is being produced and enough Z is available ($Q_{k,t} = 1$). We can check if there is enough Z by taking $R_{k_z,t} - r_z^k$, if this is positive then there is enough product available; else there is not. By dividing the resulting value by the tank capacity, we scale it to a value between -1 and 1. If this value is positive (ergo there is enough Z in the tank) and we are producing A ($X_{k,t} = 1$), then we force $Q_{k,t}$ to 1:

$$\frac{R_{k_z,t} - r_z^k}{CAP_k} + X_{k,t} - Q_{k,t} \leq 1 \qquad \forall t \in T, \ \forall k \in K_A^{on} \tag{A.21}$$

To then finish the last production constraints, there are also some sequencing constraints in place. We can use the earlier defined previous grade produced variable $L_{i,t}$ to set sequencing constraints: we cannot produce C directly after D, B directly after D, or C directly after B:

$$L_{D,t} + \sum_{k \in K_C^{on}} X_{k,t} \leq 1 \qquad \forall t \in T \tag{A.22}$$

$$L_{D,t} + \sum_{k \in K_B^{on}} X_{k,t} \leq 1 \qquad \forall t \in T \tag{A.23}$$

$$L_{C,t} + \sum_{k \in K_B^{on}} X_{k,t} \leq 1 \qquad \forall t \in T \tag{A.24}$$

A.6.1 Inventory Constraints

First we set the initial inventory levels for each tank which are an input into the model:

$$R_{k,0} = INI_k \qquad \forall k \in K, \tag{A.25}$$

We also set the maximum and minimum inventory levels per tank, based on the tank capacity CAP_k.

$$R_{k,t} \geq 0 \quad \forall t \in T, \ \forall k \in K \tag{A.26}$$

$$R_{k,t} \leq CAP_k \quad \forall t \in T, \ \forall k \in K \tag{A.27}$$

For on-site tanks of grade A, the inventory is determined by the previous inventory level plus inputs and minus outputs. Inputs are production, Z being blended in, and product being pumped into it from another tank. Outputs are pumping product into another tank, sales, and stock transfers to off-site tanks:

$$R_{k,t} = R_{k,t-1} + p_k X_{k,t-1} + f_z Q_{k,t-1} - \sum_{o \in O^k} Z_{o,k,t-1}$$
$$+ \sum_{k' \in \ K_A^{on} k} f_k \left(W_{k,k',t-1} - W_{k',k,t-1} \right) - f_b \sum_{b \in B} Y_{k,b,t}^{on}$$
$$, \forall t \in T \backslash \{0\}, \ \forall k \in K_A^{on} \quad \text{(A.28)}$$

For tanks containing *B/C/D*, the inventory level is almost the same, but does not include any *Z* being blended into it:

$$R_{k,t} = R_{k,t-1} + p_k X_{k,t-1} - \sum_{o \in O^k} Z_{o,k,t-1}$$
$$+ \sum_{k' \in \ K_i^{on} k} f_k \left(W_{k,k',t-1} - W_{k',k,t-1} \right) - f_b \sum_{b \in B} Y_{k,b,t}^{on}$$
$$, \forall t \in T \backslash \{0\}, \ \forall k \in K_i^{on}, \ \forall i \in \{B, C, D\} \quad \text{(A.29)}$$

For the tank containing Z, the only input is production, and the only output is blending away into A:

$$R_{k_z,t} = R_{k_z,t-1} + p_k X_{k_z,t-1} - \sum_{k' \in \ K_{on}^A} f_z Q_{k,t-1}, \qquad \forall t \in T \backslash \{0\} \quad \text{(A.30)}$$

For the off-site tanks, the only inputs are stock transfers from on-site tanks, and the only output is sales:

$$R_{k,t} = R_{k,t-1} - \sum_{o \in O^k} Z_{o,k,t-1} + f_b \sum_{b \in B} Y_{b,k,t}^{off} , \qquad \forall t \in T, \ i \in \{A, C\}, \ \forall k \in K_i^{off} \quad \text{(A.31)}$$

We have a separate variable for keeping track of inventory in barges. This has to be included as otherwise the model will maximize barge transportation, as barges would temporarily lower the tank inventory. First, the inventory is initialized for every barge:

$$B_{b,0} = 0, \qquad \forall b \in B \quad \text{(A.32)}$$

During loading, barge inventory increases by the product being loaded onto it:

$$B_{b,t} = B_{b,t-1} + f_b \sum_{k \in K^{on}} Y_{k,b,t}^{on} , \qquad \forall b \in B, \forall t \in LC_b^{on} \quad \text{(A.33)}$$

When off-loading, barge inventory decreases by the product being offloaded:

$$B_{b,t} = B_{b,t-1} + f_b \sum_{k \in K^{off}} Y_{b,k,t}^{off} , \qquad \forall b \in B, \forall t \in LC_b^{off} \quad \text{(A.34)}$$

When not loading/unloading, barge inventory stays constant:

$$B_{b,t} = B_{b,t-1}, \qquad \forall b \in B, \forall t \in T \ \backslash \{LC_b^{on}, ..., \ LC_b^{off}\} \quad \text{(A.35)}$$

A.6.2 Transportation

For barges, we specify that we may only load product onto a barge if it has been scheduled to be used:

$$\sum_{t\in T}\sum_{k\in K^{off}} Y^{on}_{k,b,t} \leq M_1 Y_b\ , \qquad \forall b \in B \tag{A.36}$$

In a similar fashion, we constrain the maximum volume of product that can be transported by a barge:

$$f_b \sum_{t\in T}\sum_{k\in K^{on}} Y^{on}_{k,b,t} \leq u_b Y_b\ , \qquad \forall b \in B \tag{A.37}$$

Barge transportation is a zero-sum game: everything that is loaded onto the barge from on-site tanks must be off-loaded onto off-site tanks at a later stage:

$$\sum_{t\in T}\sum_{k\in K_i^{on}} Y^{on}_{k,b,t} = \sum_{t\in T}\sum_{k\in K_i^{off}} Y^{off}_{b,k,t}, \qquad \forall i \in I \tag{A.38}$$

Due to the pipeline connections to the jetty (on-site), we can only pump from a single tank onto a barge at a time:

$$\sum_{k\in K^{on}} Y^{on}_{k,b,t} \leq 1, \qquad \forall b \in B,\ \forall t \in LC_b^{on} \tag{A.39}$$

Due to the pipeline connections from the jetty (off-site), we can only pump from a barge into a single tank at a time:

$$\sum_{k\in K^{off}} Y^{off}_{b,k,t} \leq 1, \qquad \forall b \in B,\ \forall t \in LC_b^{off} \tag{A.40}$$

Due to there being a limited number of slots available on the gantry, we can only load a certain amount of trucks per day:

$$\sum_{o\in O}\sum_{k\in K^{on}}\sum_{\delta=0}^{23} Z_{o,k,t+\delta} \leq a, \qquad \forall t \in T^{days} \tag{A.41}$$

In a similar fashion, there are a limited number of slots available on the jetty per month:

$$\sum_{b\in B} Y_b \leq j \tag{A.42}$$

For on-site tanks, if we have produced into a certain tank, then the tank must first be tested before anything can be done with the product. So we constrain product movements and orders from that tank for test time periods after production:

$$\sum_{\delta=0}^{t_{test}} \left(\sum_{b\in B} Y^{on}_{k,b,t+\delta} + \sum_{o\in O} Z_{o,k,t+\delta} + \sum_{k'\in K_i^{on}} \left(W_{k,\ k',t} + W_{k',\ k\ ,t}\right) \right) \leq M_2\left(1 - X_{k,t}\right), \qquad \forall t \in T, \forall k \in K^{on} \tag{A.43}$$

For off-site tanks, if we have moved product into a tank there, the product may have become contaminated in the barge or pipelines during transport. As such, we cannot make any sales from that tank until it has been tested:

$$\sum_{\delta=0}^{t_{test}}\sum_{o\in O} Z_{o,k,t+\delta} \leq M_2\left(1 - Y^{off}_{b,k,t}\right), \qquad \forall t \in T, \forall k \in K^{off},\ \forall b \in B \tag{A.44}$$

A.6.3 Sales

First and foremost, all committed sales (hard demand) have to be fulfilled:

$$\sum_{t=TW_o^{start}}^{TW_o^{end}} \sum_{k \in K_i} Z_{o,k,t} = 1, \qquad \forall o \in O^{hard} \tag{A.45}$$

With regards to the uncertain sales, we use a chance constraint to indicate the max allowed probability of not meeting all demand:

$$P\left(\sum_{k \in K_i} \sum_{t \in T} \sum_{o \in O} X_{o,k,t} \leq DEM_i\right) \leq \lambda_i, \qquad \forall i \in I \tag{A.46}$$

As we know that total demand is normally distributed, we can transform this chance constraint into its deterministic counterpart based on the probability density function of the normal distribution in similar fashion to Zhang et al. (2019) [33]. This yields the following constraint:

$$\frac{\sum_{k \in K_i} \sum_{t \in T} \sum_{o \in O} (X_{o,k,t}) - E[DEM_i]}{\sqrt{var[DEM_i]}} \leq \Phi^{-1}(1 - \lambda_i) \tag{A.47}$$

Each order also has a time window, outside of which it cannot be fulfilled:

$$\sum_{t=0}^{TW_o^{start}} \sum_{k \in K_i} Z_{o,k,t} = 0, \qquad \forall o \in O \tag{A.48}$$

$$\sum_{t=TW_o^{end}}^{T} \sum_{k \in K_i} Z_{o,k,t} = 0, \qquad \forall o \in O \tag{A.49}$$

If an order is on-site only, it cannot be fulfilled from off-site tanks:

$$\sum_{k \in K^{off}} Z_{o,k,t} \leq 1 - s_o, \qquad \forall o \in O \tag{A.50}$$

On-site, we can only fulfill 2 orders simultaneously due to loading gantry capacity:

$$\sum_{k \in K^{on}} \sum_{o \in O} Z_{o,k,t} \leq 2, \qquad \forall t \in T \tag{A.51}$$

Orders also cannot be fulfilled during weekends:

$$\sum_{\delta=0}^{48} \sum_{k \in K} \sum_{o \in O} Z_{o,k,t} = 0, \qquad \forall t \in T^{weekends} \tag{A.52}$$