Universidad de Valladolid

Escuela de Doctorado Universidad de Valladolid

PhD PROGRAMME IN INDUSTRIAL ENGINEERING

DOCTORAL THESIS:

OPTIMIZATION OF MONTHLY CRUDE OIL SCHEDULING IN REFINERIES WITH SHIP ARRIVALS

Submitted by Tomás Jorge García García Verdier in fulfillment of the requirements for
PhD degree by the Universidad de Valladolid

Supervised by:
Dr. Gloria Gutiérrez Rodríguez
Dr. Carlos Alberto Méndez

Financial support was received from the Spanish Government through the projects a-CIDiT (PID2021-123654OB-C31) and InCo4In (PGC2018-099312-B-C31), as well as from the European Social Fund.

Furthermore, this work was supported by the Regional Government of Castilla y León and the EU-FEDER (CLU2017-09, CL-EI-2021-07, UIC233).

Agradecimientos

Agradezco enormemente a mis directores de tesis, Gloria Gutiérrez y Carlos Méndez, por su tiempo, guía y por todo lo que me han enseñado a lo largo de estos años.

A César de Prada, por su dedicación y valiosos consejos, que fueron fundamentales para el desarrollo de la tesis.

A Carlos Palacín, quien me dio una ayuda enorme cuando comenzaba mi doctorado.

A los grandes amigos que conocí gracias al doctorado, quienes me hicieron sentir como si estuviese en familia: Daniel Montes, Alejandra López, Rogelio Rivero, Yury Noguera, Felipe Rivero, Erika Oliveira, Irina Bausa y Sergio Uc.

A mi amigo Demian Presser, quien me motivó a hacer el doctorado.

Agradezco profundamente a Ignacio Grossmann por permitirme realizar una estancia dentro de su grupo de investigación en Carnegie Mellon, una experiencia enriquecedora tanto a nivel personal como profesional.

A los profesionales del equipo de Planificación de Petronor, por las facilidades brindadas para la realización de este trabajo.

A mis padres, Gloria y Marcelo, por estar siempre presentes, por la educación y las oportunidades que me han brindado. Sin ellos, no hubiese sido posible este logro.

A mis hermanos, Joaquín, Ignacio, Federico y Eliana, que siempre me han acompañado y ayudado a lo largo de la vida, tanto en los momentos buenos como en los malos.

A Chiqui, quien ha sido otra madre para mí y que me ha inculcado la importancia del compromiso con el estudio.

Resumen

Objetivo

La tesis se focaliza en la optimización de la programación de operaciones de crudos en una refinería con acceso marítimo, abordando el problema mediante dos enfoques: determinístico y estocástico, y considerando las limitaciones identificadas en la literatura en ambos campos.

Metodología

En el enfoque determinístico, la refinería bajo estudio, ubicada en España, es abastecida con distintos tipos de crudo a través de barcos, y deben realizarse diversas operaciones para satisfacer la demanda especificada. Estas operaciones incluyen la coordinación de arribos de barcos, la gestión de inventarios de crudo y la programación del suministro a las unidades de destilación de crudos (CDUs, por sus siglas en inglés) y unidades de procesamiento ubicadas aguas abajo.

Una característica relevante de este problema es que los tanques pueden almacenar mezclas de crudos, lo cual introduce restricciones que son no lineales y no convexas, derivando en un modelo de programación mixto entero no lineal cuya resolución representa un desafío computacional. Además, el modelo contempla la clasificación de tanques según el tipo de mezcla de crudo almacenado. Cabe destacar, a su vez, que el problema considera un horizonte de programación mensual, lo que da lugar a un modelo de gran escala que resulta imposible de resolver en tiempos compatibles con los requerimientos del personal de la refinería. En particular, el modelo alcanza un tamaño aproximado de 700000 restricciones, de las cuales 165000 son no lineales, junto con 400000 variables continuas y 12000 variables binarias. Es importante señalar que, al intentar resolver el modelo monolítico, no se obtuvo ninguna solución dentro de un tiempo de cómputo de cuatro horas.

Para abordar este problema, se desarrolla un modelo de programación matemática basado en formulación de tiempo continuo utilizando puntos de tiempo globales, definidos como instantes en los que se evalúa el estado de los recursos limitados. El modelo asiste en la toma de decisiones relacionadas con la programación de crudo, como determinar cuándo descargar los barcos,

qué tanques deben recibir el crudo, qué tanques deben alimentar a las CDUs, por cuánto tiempo y en qué cantidades, entre otras decisiones.

Adicionalmente, se propone una estrategia de resolución novedosa para enfrentar la naturaleza no lineal del problema y el tamaño del modelo resultante al considerar un horizonte de planificación mensual. Dicha estrategia combina una técnica novedosa de aproximación lineal por partes utilizando planos para aproximar el producto de dos variables continuas no negativas, junto con una estrategia de descomposición temporal, que consiste en la resolución iterativa de dos modelos: uno agregado y uno detallado.

Como se mencionó anteriormente, el problema también se aborda desde una perspectiva estocástica. La motivación detrás de esta parte del trabajo radica en el hecho de que la fase de programación de operaciones usualmente no contempla eventos imprevistos, los cuales pueden volver inviable el plan generado. Una de las principales fuentes de incertidumbre que puede provocar esta situación es la variabilidad en las fechas de arribo de los barcos que entregan crudo a la refinería. Abordar esta incertidumbre constituye otro de los puntos principales de esta tesis, con el objetivo de obtener cronogramas más robustos. En particular, este trabajo contribuye al desarrollo de modelos estocásticos que incorporan gestión del riesgo.

Con este fin, se desarrolla un modelo de programación estocástica de dos etapas, considerando incertidumbre en las fechas de arribo de los barcos. Si bien el modelo resultante está orientado a aplicaciones de programación a corto plazo y ciertas características del caso de estudio se simplifican respecto al modelo determinístico, mantiene una complejidad suficiente para servir como prueba de concepto, permitiendo analizar los efectos y beneficios de considerar la incertidumbre en este tipo de problemas.

El modelo estocástico se extiende además para incorporar gestión del riesgo mediante la inclusión de la medida Valor en Riesgo Condicional (CVaR, por sus siglas en inglés) como función objetivo.

Resultados

En primer lugar, respecto al enfoque determinístico, se resuelven dos casos de estudio basado en datos reales de la refinería analizada. Ambos modelos se implementan utilizando Pyomo y se resuelve con Gurobi 11.0.0 para problemas MILP, en un equipo con procesador Intel Core i9-13900K a 3.00 GHz y 128 GB de memoria RAM. El primer problema alcanza una solución con un gap relativo inferior al 1% en un tiempo total de 35 minutos, y el segundo problema alcanza una solución con un gap relativo inferior al 1% en un tiempo total de 50 minutos.

Las soluciones obtenidas se analizan a través de diversas gráficas, tales como diagramas de Gantt, evolución de propiedades de las mezclas alimentadas a las CDUs, así como también tablas con información de tiempos de resolución y estadísticas del modelo. Además, cabe destacar que dichos resultados fueron validados por los profesionales del equipo de planificación de la refinería bajo estudio.

En segundo lugar, respecto al enfoque estocástico, se realizan análisis de desempeño utilizando las métricas de Valor Esperado de la Información Perfecta y Valor de la Solución Estocástica, con el fin de evaluar las ventajas potenciales del modelo estocástico frente a su contraparte determinista. Asimismo, se analizan las soluciones obtenidas para distintos niveles de aversión al riesgo. El ejemplo se resuelve utilizando el software GAMS 41.3.0, Gurobi 9.5.2 para modelos MILP y CONOPT 4.29 para modelos NLP, en un ordenador con procesador Intel Core i9-13900K a 3.00 GHz y 128 GB de memoria RAM.

Conclusiones

Por un lado, a partir del análisis de los resultados del enfoque determinístico, se concluye que esta metodología permite obtener soluciones de buena calidad dentro de tiempos computacionales alineados con los requerimientos de los usuarios.

Por otro lado, a partir del valor de la solución estocástica, se concluye que el modelo de programación estocástica de dos etapas ofrece resultados más robustos que los enfoques deterministas, principalmente porque permite corregir las consecuencias de decisiones tomadas en el presente en función de condiciones futuras. Además, la inclusión del CVaR permite penalizar valores extremos que podrían surgir ante la realización de ciertos escenarios, minimizando así el riesgo.

Finalmente, cabe destacar que como fruto de esta tesis se publicó un artículo en una revista con factor de impacto JCR, cuatro trabajos sometidos a revisión por pares en congresos internacionales de gran relevancia del área y seis trabajos en congresos nacionales.

Abstract

Objective

The thesis focuses on optimizing crude oil operations scheduling in a refinery with maritime access, addressing the problem through both deterministic and stochastic approaches, and considering the gaps identified in the literature for each.

Methodology

For the deterministic approach, the refinery under study, located in Spain, is supplied with different types of crude oil via vessels, and it must carry out several operations to meet the specified demand. These operations involve coordinating ship arrivals, managing crude oil inventory, and determining the feeding schedule for crude distillation units (CDUs) and downstream processing units.

A relevant feature of this problem is that tanks can store crude oil blends. This behavior introduces nonlinear, nonconvex constraints, resulting in a mixed-integer nonlinear programming model, which is computationally challenging to solve. It is also worth noting that the model considers tank classification based on the type of crude blend stored. Moreover, this problem requires consideration of a monthly scheduling horizon, leading to a large-scale model that cannot be solved within a time frame compatible with the needs of refinery staff. The model reaches an approximate size of 700000 constraints, 165000 of which are nonlinear, along with 400000 continuous variables and 12000 binary variables. It is notable that no feasible solution was obtained when attempting to solve the monolithic model after four hours of computation.

To address this problem, a mathematical programming model is developed based on a continuous-time formulation using global time points, which are time instants at which the status of limited resources are evaluated. The model supports decision-making related to crude oil scheduling, such as determining when to unload vessels, which tanks should receive the crude oil, which tanks should feed the CDUs, and for how long and in what quantities, among other relevant decisions.

In addition, to address the nonlinear nature of the problem and the large model size arising from a one-month planning horizon, a novel solution strategy is proposed. This strategy combines a novel piecewise linear approximation technique using planes to approximate the product of two non-negative continuous variables, and a temporal decomposition approach consisting of the iterative resolution of two models: an aggregate model and a detailed model.

As previously mentioned, the problem is also approached from a stochastic perspective. The main motivation behind this part of the study lies in the fact that the scheduling phase typically does not account for unforeseen events, which may occasionally render the generated schedule infeasible. One of the main sources of uncertainty that can lead to this issue is the variability in the arrival dates of the ships delivering crude oil to the refinery. Addressing this uncertainty represents another key focus of the thesis, with the goal of achieving more robust schedules. Specifically, this work contributes to the development of stochastic models that incorporate risk management.

To this end, a two-stage stochastic programming model is developed, considering uncertainty in the arrival dates of the supplying ships. Although the resulting model is aimed at short-term scheduling applications and certain characteristics of the case study are simplified compared to the deterministic version, it maintains sufficient complexity to serve as proof of concept, allowing us to analyze the effects and benefits of accounting for uncertainty in this type of problem.

The stochastic model is further extended to incorporate risk management by using the Conditional Value-at-Risk (CVaR) measure as the objective function.

Results

First, regarding the deterministic approach, two case studies based on real data from the refinery under study were solved. Both models were implemented using Pyomo and solved with Gurobi 11.0.0 for MILP models on a computer equipped with an Intel Core i9-13900K 3.00 GHz processor and 128 GB of RAM. The first problem was solved with a relative gap of less than 1%, in a total solution time of 35 minutes; the second problem was solved with a relative gap of less than 1%, in a total solution time of 50 minutes.

The resulting solution is analyzed through various visualizations, including Gantt charts, the evolution of property values in feed blends to the CDUs, as well as solution times and model statistics. Additionally, it is worth noting that the results were validated by professionals from the planning team of the refinery under study.

Second, regarding the stochastic approach, performance analyses are conducted using the Expected Value of Perfect Information and the Value of the Stochastic Solution to evaluate the potential advantages of the stochastic model over its deterministic counterpart. Furthermore, the solutions obtained are analyzed for different levels of risk aversion. The example is solved using GAMS 41.3.0 software, Gurobi 9.5.2 for MILP models, and CONOPT 4.29 for NLP models, on a computer with an Intel Core i9-13900K 3.00 GHz processor and 128 GB of RAM.

Conclusions

On the one hand, based on the analysis of the deterministic approach results, we conclude that this methodology makes it possible to obtain high-quality solutions within computational times aligned with user requirements.

On the other hand, based on the value of the stochastic solution, we conclude that the two-stage formulation offers more robust results than deterministic approaches, primarily because it allows us to adjust initial decisions based on future conditions. Furthermore, incorporating the CVaR measure enables the penalization of extreme outcomes that may occur in certain scenarios, thus minimizing risk.

Finally, it is worth noting that this thesis has led to the publication of one article in a JCR-indexed journal, four peer-reviewed papers submitted to major international conferences in the field, and six contributions to national conferences.

Contents

Chapter 1

Introduction

We live in an era where technological advances have facilitated the emergence of new companies in both established and emerging industries. At the same time, increasing awareness of environmental sustainability has imposed new constraints on the production of goods and the delivery of services. In this context, the optimization of resource management along supply chains has become a key issue, especially since many of these resources are scarce or non-renewable.

The oil and gas (O&G) industry is one of the largest sectors in the global energy market. The O&G supply chain is typically divided into three main segments: upstream, midstream, and downstream, each corresponding to a specific stage of the supply chain's operations.

The upstream segment encompasses activities related to the exploration, drilling, and production of crude oil and natural gas. Midstream operations focus on the processing, storage, and transportation of crude oil, natural gas, and natural gas liquids. Finally, the downstream sector is dedicated to refining, marketing, and distributing petroleum products.

Within the downstream segment, oil refineries play a central role. An oil refinery is a complex industrial facility that converts crude oil into a variety of useful petroleum products through refining processes. These processes break down raw crude oil into components such as gasoline, diesel, jet fuel, kerosene, and others. These refined products are essential to modern life, serving as fuels for transportation, heating, and electricity generation, as well as raw materials for chemicals and construction, including road paving.

The structure of a refinery can be divided into three main sections based on its operations. The first section, which is the focus of this thesis, involves the scheduling of crude oil operations. This stage includes the unloading of crude oil (e.g., from ships), managing crude oil inventory in tanks, and scheduling the mixture feed to the crude distillation units (CDUs). The second section pertains to production unit scheduling, which encompasses the fractionation and reaction processes that transform crude oil into various intermediate products. The third section focuses on the scheduling, blending, storage, and distribution of final petroleum products to meet market demands.

In this work, we concentrate specifically on the optimization problem of crude oil operations scheduling in a refinery supplied by ships. This problem involves a complex network of ships, tanks, pipelines, and CDUs [1]. This process optimization is essential to the overall operation, as decisions made at this stage influence and constrain decisions in downstream processes.

Additionally, the scope of this work extends beyond the CDUs, integrating the scheduling of certain downstream units. This approach accounts for volume transfers between units and inventory management in intermediate tanks, with the aim of achieving a more comprehensive and integrated solution.

1.1 Crude oil operations in marine-access refineries

In an oil company, the planning process typically begins at a central office responsible for overseeing a group of refineries. This office develops a production plan for a defined period of time, usually a month, detailing the dates and types of crude oil to be received and specifying the aggregate production targets for various petroleum products at each refinery. Software tools, such as Aspen PIMS, are often used to generate these plans. However, these plans do not take into account time and others constraints, which means they may not represent a feasible solution. For example, certain crude oils may not be available until part way through the planning horizon. After receiving the central plan, each refinery's planning team must create a detailed schedule of crude oil operations, considering time, operational and facility constraints, as well as the refinery's initial conditions to meet the aims of the central plan for the period of time considered. The first and most important step to convert crude oil into the different refinery products takes place in the CDUs. As the amounts and characterisitcs of each of these products changes, the composition of the feed to the CDUs (several dozens componenets) must also be adapted according to the production plan, taking into account the different compositions of the different crude oils to be received.

Crude oil is often delivered to refineries by ships arriving at nearby marine terminals, which are connected to the refineries via oil pipelines. Marine-access refineries typically follow one of two configurations based on the types of tanks they use: those with both storage and charging tanks (Figure 1.1), and those with only storage tanks (Figure 1.2). Storage tanks are dedicated to receiving and storing crude oil from ships, while charging tanks are used to create blends that meet certain quality specifications before being fed into the distillation units. Alternatively, some refineries use only storage tanks and implement mixing online in the pipelines that feed the CDUs, as

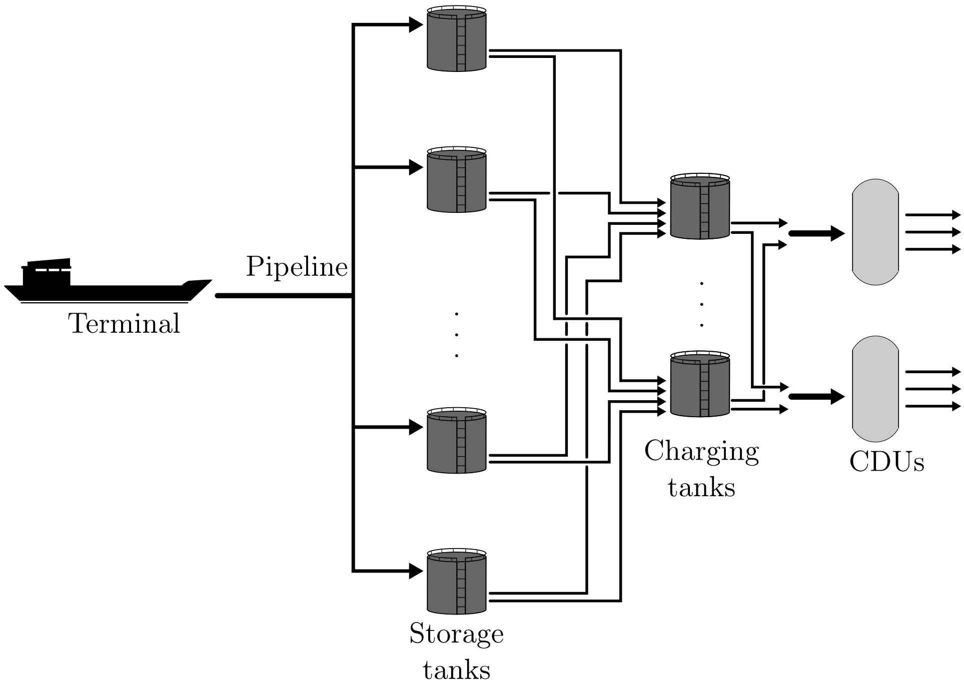

Figure 1.1: Schematic of system with storage and charging tanks

shown in Figure 1.2. This simplified configuration saves space and eliminates immobilized capital, though it requires appropriate control systems to ensure accurate blending.

The types of crude oil tankers contracted, and the frequency of deliveries may vary from refinery to refinery depending on logistical arrangements and operational needs. Tankers are classified according to their crude oil carrying capacity. Medium sized Aframax tankers typically carry between 500000 and 800000 barrels of oil. Suezmax tankers are larger, with capacities ranging from 800000 to 1.3 million barrels, while Very Large Crude Carriers (VLCCs) can carry between 1.3 and 2 million barrels. The largest category, the Ultra Large Crude Carriers (ULCCs), can carry more than 3 million barrels. At the refinery studied in this work, a ship arrives approximately every three days, with the Aframax class vessels, which can carry between 500000 and 800000 barrels, being the most common.

Another important point to note is that not all crude oils are the same; there are different types of crude oil that can vary significantly in properties such as density and yields, among others. Because of these differences, it is necessary to blend different crudes to meet production targets and product quality specifications.

Regarding crude oil operations scheduling, it means allocating the resources involved (vessels, tanks, CDUs) to operations (e.g., loading a tank, feeding a CDU, etc.) and sequencing them over time so that the constraints

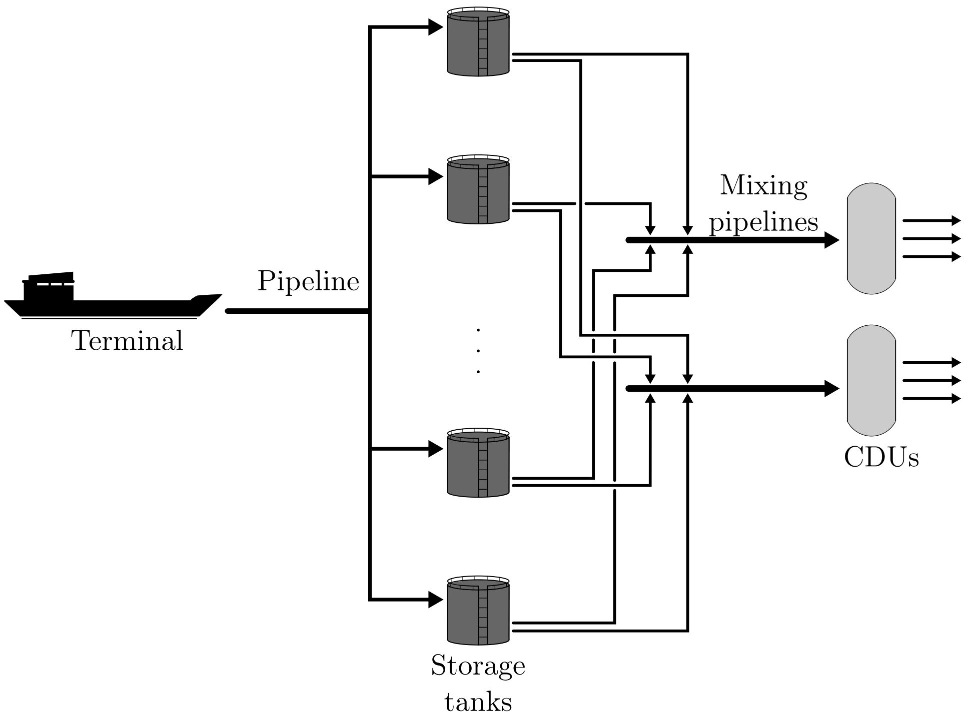

Figure 1.2: Schematic of system with only storage tanks

and aims of the process are satisfied. Optimal scheduling, in addition, will provide the best use of the resources minimizing a cost function.

One point that is interesting to mention is that analogously to batch process operations scheduling problems, one could think of the volumes of crude oil transferred as batches. However, we should note that in the crude operations scheduling problem, these batches are not defined in advance at the beginning of the scheduling horizon. Therefore, this feature increases the complexity of the problem resolution [2].

1.2 Process operation

The optimization of crude oil operations scheduling can be defined as the process of deciding the best way to operate the system based on the management of four macro-operations in a coordinated way, each of which involves a set of interrelated operations.

In the following subsections, these macro-operations and the decisions to be made in each of them are explained in more detail.

1.2.1 Vessel unloading

Vessels are scheduled to arrive at the terminal on the dates specified in contractual agreements. However, these schedules are not always met as arrival dates can be affected by factors such as weather conditions.

When a vessel arrives at the terminal, two scenarios can occur. In the first, the dock is available, allowing the vessel to moor and begin unloading crude oil into the pipeline immediately. In the second, the dock is occupied by another vessel, forcing the arriving vessel to wait until the dock becomes available.

At this point, decisions must be made regarding the timing of unloading to ensure that the production plan is met while simultaneously minimizing vessel demurrage and tardiness costs. Demurrage is defined as the difference between the date a vessel begins unloading and its arrival date, while tardiness is defined as the difference between the completion of unloading and the scheduled departure date, being null if the vessel finishes on or before the stipulated date.

1.2.2 Tank loading

Concurrently with the unloading of ships, we must select the receiving tanks. However, it must be taken into account that there is a maximum number of tanks that can be loaded simultaneously, and therefore, their loading must also be sequenced. For example, if we can load a maximum of three tanks at a time and we decide that four tanks will receive the crude, then at least one of the tanks will receive crude oil once the loading of another tank has been completed.

Tanks are classified not only by their function (i.e., storage tanks or charging tanks), but also by the types of crude oil they store. Tanks are rarely empty and usually contain blends of different crudes, which imposes additional restrictions when loading a tank, as it must be considered that the crude received is compatible with the blend stored. In addition to the tank assignment and order, the volumes transferred to each tank must be calculated.

1.2.3 Tank unloading and crude mixing

At this point, two scenarios may arise depending on the refinery's configuration.

If the refinery has charging tanks, the blends that feed the CDUs are prepared in advance and stored in these tanks. In this case, the decision-making involves determining which charging tanks will supply the CDUs.

If only storage tanks are available, we must choose the feed from the stored crude oil to each CDU. This includes selecting and sequencing the participating tanks and calculating the volumes transferred from each tank so that the feed properties in the mixing pipelines fall within the established ranges. While the tanks store blends of crude oils, these blends do not necessarily meet the specifications required to feed the CDUs. The final blends that meet the specifications are obtained in the mixing pipelines.

As previously mentioned, the tanks can store mixtures of crudes. From this characteristic, it must be considered that the concentration of crude oil in the outlet flow of a tank must match the concentration inside the tank.

Besides, other operational constraints must be met. These include a minimum settling time in a tank after receiving a load, to allow possible water to settle, and limits on the maximum number of units that can be fed simultaneously from a single tank.

1.2.4 Crude distillation unit loading

The “crude distillation unit loading" operation is closely related to the “tank unloading" operation, so it is difficult to distinguish which decisions belong to which; in any case, they could be thought of as shared decisions that must be taken considering the constraints and requirements of both. The distillation units are responsible for processing the blends prepared in the mixing pipelines or charging tanks, depending on the refinery, to meet the demand. These units must be fed continuously, i.e., their operation cannot be interrupted. Therefore, it complicates the decision-making process and inventory management. As mentioned above, one of the crucial points in the loading of CDUs is compliance with feed quality specifications, i.e., keeping the concentration of blend properties within defined ranges. In addition, there are operating constraints, such as the maximum number of tanks allowed in parallel and the feed flow rates.

One of the main challenges of the crude oil operations scheduling problem lies in coordinating the decisions taken at the terminal and the crude oil section since their objectives differ. While the former seeks to unload the ships as soon as possible to avoid demurrage and tardiness costs, the latter aims to have the crude oil available at the most convenient times and in the most convenient qualities and quantities to meet the production plan. One way to address this problem is through the development of mathematical programming models.

1.3 State of the art

Scheduling refers to the process of allocating resources to operations, determining their timing, and establishing the sequence in which these operations are executed. In some cases, scheduling problems also involve deciding on the number and size of tasks or operations to be performed.

Mathematical programming is a powerful tool for addressing industrial scheduling optimization problems. This technique relies on mathematical models that represent the processes through variables, constraints that relate these variables, and an objective function. The objective function is usually defined as the minimization of total costs, although other criteria, such as profit maximization, can also be used. The goal is to find the optimal schedule that minimizes or maximizes the objective function while satisfying all constraints to ensure feasibility.

In mathematical programming, models are classified into different categories based on the types of variables involved and the form of the objective function and constraints. First, the simplest one is the Linear Programming (LP) model, where the objective function and constraints are linear and all variables are continuous [3]. Second, the Mixed-Integer Linear Programming (MILP) model, an extension of the LP model that includes discrete variables [4]. Next, there is the Nonlinear Programming (NLP) model. This model encompasses those cases where at least one constraint or the objective function is nonlinear, and the variables are continuous [5]. Finally, extending the NLP model to include discrete variables leads to the Mixed-Integer Nonlinear Programming (MINLP) model, which combines the complexity of nonlinearity and integer decision making [6].

In scheduling problems, before developing a model, we must determine certain characteristics of it, among which two stand out: the time representation and the degree of uncertainty in the parameters. Regarding time representation, we can choose between two primary approaches: discrete-time formulation and continuous-time formulation.

It is worth noting that time representation can also be categorized based on other criteria, such as time-grid-based and sequence-based approaches. However, the focus here will remain on the distinction between discrete and continuous-time formulations.

In discrete-time-based models, the programming horizon is divided into a finite number of intervals with a predefined duration, and events – meaning any change in operations – only take place at the beginning or the end of these intervals. The greater the number of intervals, the more precise the solution will be, but the size of the model will increase and, therefore, will require more computational effort for its resolution.

One alternative to improve the time precision of the solution and to decrease the number of variables involved is the use of models based on continuous-time formulation, where timing decisions are explicitly represented as a set of continuous variables defining the exact times at which the events take place [7].

As previously mentioned, another important decision before developing a mathematical model is choosing between a deterministic or stochastic formulation. In deterministic models, the values of the parameters are assumed to be known, whereas in stochastic models, some of the problem data may be considered uncertain, meaning they can be represented as random variables [8].

1.3.1 Literature review

In the last decades, a wide variety of articles have been published in the area of crude oil operations scheduling, covering the development of both deterministic and stochastic models.

Deterministic models

The paper of Lee et al. [9] involves one of the earlier works to address the optimization of short-term scheduling for crude oil unloading, tank inventory management, and CDU charging. In this article, the authors developed a mixed-integer linear programming (MILP) model that relies on time discretization, in which the bilinear equations arising from mixing operations are replaced with individual component flows to maintain linearity.

In Jia et al. [10], the authors addressed the problem of crude-oil short-term scheduling, which involves optimizing the unloading of crude oil from vessels, its transfer to storage tanks, and the charging of crude distillation units. For this purpose, they developed a novel MILP model based on a continuous-time representation, utilizing the state-task network (STN) representation introduced by Kondili et al. [11].

In the paper Reddy et al. [12], a continuous-time mixed-integer linear programming (MILP) formulation was presented for the short-term scheduling of operations in a refinery that handles crude from very large crude carriers. Moreover, the authors put forward an iterative algorithm to address the crude composition discrepancy. This algorithm entails solving a series of MILPs with gradually reduced size and complexity to achieve a near-optimal solution.

The authors of Furman et al. [13] proposed a mixed-integer nonlinear programming (MINLP) model based on a continuous-time formulation to optimize the scheduling of fluid transfers within tanks and robustly handle the synchronization of time events with material balances. Additionally, they presented a new approach to depict the inflow and outflow from a tank, which holds the potential to reduce the number of time events in continuous-time scheduling formulations. Subsequently, the modeling paradigm was applied to develop charging schedules for refinery crude units.

Mouret et al. [14] developed a continuous-time MINLP formulation to tackle crude-oil scheduling problems. This formulation is based on the representation of a schedule as a sequence of operations and is called the single-operation sequencing (SOS) model. Additionally, they introduced a sequencing rule to address the symmetries that may arise in the model. Finally, a two-step MILP-NLP procedure was implemented to solve the model.

Li et al. [15] developed a novel unit-specific event-based continuous-time MINLP formulation to tackle the crude oil scheduling problem for a marine-access refinery. Furthermore, they introduced a branch-and-bound global optimization algorithm with a piecewise-linear underestimation approach to solve the model.

In Yadav et al. [16], the authors introduced a simplified STN-based formulation to tackle the problem of scheduling crude oil operations using a unit-specific event-based continuous-time representation. The solution strategy proposed by the authors involves relaxing the MINLP model by dropping the nonlinear constraints and solving the resulting MILP model. In case the obtained solution exhibits composition discrepancies, the original MINLP model is also solved to rectify these discrepancies.

Hamisu et al. [17] proposed an enhanced version of the MILP model developed by Lee et al. [9] through the inclusion and modification of a set of constraints that allow for a decrease in operating costs and provide more flexible operation.

The authors of Castro et al. [18] tackled the optimization of scheduling of crude oil blending operations in a refinery. As a solution, they developed an MINLP model based on continuous-time formulation with a single time grid derived from a resource-task network (RTN) superstructure. Subsequently, the MINLP model was solved using a two-step MILP-NLP algorithm, which involves a tight relaxation of the bilinear blending constraints using multiparametric disaggregation.

In Cerdá et al. [19], a MINLP continuous-time approach for scheduling crude oil operations in marine-access refineries was introduced. The developed model is based on global-precedence sequencing variables to establish the order of loading and unloading operations in the storage tanks, and synchronized time slots of variable length are used to model the sequence of feedstock supplied to each CDU.

Zimberg et al. [20] presented a discrete-time MILP model to optimize the reception, blending, and delivery of crude oil from a terminal to a pipeline without considering crude oil processing. The authors proposed replacing the nonlinear equation derived from the blending of crudes in tanks with a set of linear equations that include an adjustment term for composition discrepancies.

The paper of de Assis et al. [21] focused on optimizing operations at a crude oil terminal, specifically, the optimization of crude oil unloading from vessels to storage tanks and transfers from storage tanks to the pipeline that connects the terminal to the refinery. In this work, an MINLP model based on discrete-time formulation was presented, along with an iterative two-step MILP-NLP decomposition algorithm, which involves using piecewise McCormick envelopes to replace bilinear terms and a domain-reduction strategy.

In de Assis et al. [22], the authors tackled the Operational Management of Crude Oil Supply (OMCOS), which involves optimizing the schedule of vessel trips and crude oil operations at a terminal in an integrated manner. As a solution, they proposed a discrete-time MINLP formulation that was solved through an iterative MILP-NLP decomposition approach.

In another paper, de Asis et al. [23] introduced an MILP clustering formulation, whose solution serves as a preliminary step before solving the OMCOS MINLP formulation. Based on the clustering solution, bounds on crude properties inside tanks can be inferred, enabling the linearization of bilinear terms in blending constraints and resulting in an MILP approximation. Subsequently, they applied an MILP-NLP decomposition strategy to achieve a solution for the MINLP model.

Although not developed in the context of crude oil operations, the work in Lagzi et al. [24] presents a comparative study between discrete-time formulations, with both uniform and nonuniform discretization, and continuous-time formulations for short-term scheduling of multipurpose facilities. For this purpose, the authors extend the discrete-time formulation of Patil et al. [25] by incorporating the flexible discretization approach proposed by Velez and Maravelias [26]. The study, which includes features such as single-purpose machines, discrete batches, and multitasking, offers alternative modeling strategies that could be adapted or extended to crude oil scheduling problems.

Furthermore, for scheduling problems in batch environments, methodologies have been proposed to integrate long-term planning with short-term scheduling over extended horizons. In Menon et al. [27], the authors propose an iterative integration framework for linking operational planning and scheduling decisions in large-scale multijob batch plants. A key contribution of this work is its ability to handle plants without fixed products or recipes, where job specifications vary according to client demands. The integration combines a long-term planning model and a short-term scheduling model through a rolling horizon approach.

Despite being applied to a different industrial context, such frameworks illustrate how extended time horizons can be addressed in batch scheduling problems through the integration of planning and scheduling models.

However, in crude oil operations scheduling, most of these papers address the problem using short scheduling horizons or simplified refinery configurations, unlike those encountered in real-world scenarios. Only a few studies address the monthly crude scheduling problem using relatively detailed models, as seen in Zhang et al. [28].

Considering this monthly horizon significantly increases the problem size and makes it much harder to solve, especially within a short time frame, so that it can be used in practice by the personnel of the operation programming departments. As a consequence, the development of strategies for decomposing and fast solving the mathematical programming problem appears as a central research topic. At the same time, incorporating all significant elements of the industrial site under consideration also becomes an important aspect for the applicability of the developments, which is not always present in the literature. As mentioned earlier, in addition to classifying tanks according to their function (i.e., storage or charging tanks), tanks can also be classified according to the types of crude oil they store. When included, only a limited number of crude classes are usually addressed (typically, heavy and light crude), and tanks are permanently assigned to a specific grade. In the same way, the only processing units considered in most papers are CDUs, but other important units, such as vacuum distillation or visbreaking, also play a role and influence the operation of the CDUs.

Stochastic programming models

All the works mentioned above tackled the problem through a deterministic approach. However, it is very important to take into account unplanned events for generating practical and useful schedules, so stochastic programming models have also been developed to address the problem of crude oil operations scheduling under uncertainty, among which the following stand out.

In Wang et al. [29], a discrete-time two-stage robust model was proposed to address the crude oil operations scheduling problem considering uncertainty in vessel arrival times and product demand.

Cao et al. [30] presented stochastic chance-constrained mixed-integer nonlinear programming (SCC-MINLP) models based on discrete-time formulation to solve the integrated problem of crude oil short-term scheduling, blending, and storage management under uncertainty in crude distillation unit demands.

In Li et al. [31], the authors addressed the crude oil scheduling problem for a marine-access refinery under demand uncertainty. To achieve this, they utilized the unit-specific event-based continuous-time formulation presented in Li et al. [15] and applied the theory of robust optimization framework to formulate the robust counterpart optimization model.

Oliveira et al. [32] proposed a two-stage stochastic MILP model, based on discrete-time formulation, that simultaneously defines the scheduling of oil pumping through a pipeline and the sequencing of ships berthing at a terminal at the lowest possible cost.

As in the case of deterministic models, it is worth highlighting research developed for other scheduling areas, distinct from crude oil operations scheduling, where novel stochastic programming approaches have been proposed. One such example is found in Menon et al. [33], where the authors present a discrete-time two-stage stochastic programming model for scheduling batch processes under type II endogenous uncertainty, in which the timing of uncertainty realization depends on model decisions. A key feature of this approach is its ability to enforce non-anticipativity implicitly, without requiring auxiliary binary variables or explicit non-anticipativity constraints.

All these papers aimed to minimize the expected value of a certain cost function, but none of them took into account the problem of risk. That is, the minimization of the probability that in some scenarios the cost function may have a very bad value. At the same time, it is worth noting that most of these authors presented stochastic models using a discrete-time representation, and only one used unit-specific continuous-time formulation.

Machine learning-based scheduling

Although not addressed in the current thesis, recent works have started to explore the use of machine learning techniques for process scheduling, especially under uncertainty. These approaches represent an emerging research direction that could complement traditional optimization-based methods. For example, in Rangel et al. [34], the authors propose a framework for training deep reinforcement learning (DRL) hybrid agents to perform online scheduling in state-task networks (STNs) under aleatoric and epistemic uncertainty. The main features of the agent are its parametrized action space, which allows to complement an action, and the use of RNNs to approach the problem as a partially observable Markov decision process (POMDP), thereby incorporating temporal information from past observations to improve decision-making under uncertainty.

1.3.2 Summary of the state of the art

The state of the art in the optimization of crude oil operations scheduling has advanced significantly over the past decades in the development of deterministic and stochastic models. However, as mentioned before, there are still important gaps that require further investigation. Below, we provide an overview of both approaches, highlighting their advances and existing research gaps.

Advances in deterministic and stochastic models

In deterministic models, extensive research has been conducted using both discrete-time and continuous-time formulations for short-term scheduling. Notable advances include the development of models that integrate operations from crude unloading from ships to the feeding of distillation units, considering both charging and refinery tanks. While most attention has been given to cases involving charging tanks, some models have addressed scenarios limited to refinery tanks.

On the stochastic side, progress includes the development of stochastic programming models based on both continuous-time and discrete-time representations, with the latter receiving more attention. These models consider uncertainty in parameters such as demand and ship arrival dates, focusing on minimizing the expected value of the objective function.

Research gaps in deterministic and stochastic models

Despite these advances, several gaps remain in the literature. Deterministic models still face challenges in scaling up to monthly planning horizons, so the development of efficient solution strategies for such MINLP models is an open area of research. Moreover, there is a lack of models that include downstream units such as vacuum distillation or visbreaking, which are essential for obtaining a comprehensive solution. Another important gap is the absence of models that classify tanks according to the type of crude blend stored, which could affect scheduling decisions.

Regarding stochastic modeling, models based on continuous-time formulations remain underexplored compared to their discrete-time counterparts. Additionally, existing models do not include risk management, which is important for obtaining more robust solutions.

1.4 Purpose and scope of the thesis

The thesis focuses on optimizing crude oil operations scheduling in a refinery with maritime access, addressing the problem through both deterministic and stochastic approaches, and considering the gaps identified in the literature for each.

For the deterministic approach, the refinery under study, located in Spain, is supplied with different types of crude oil via vessels, and it must carry out several operations to meet the specified demand. These operations involve coordinating ship arrivals, managing crude oil inventory, and determining the feeding schedule for crude distillation units (CDUs) and downstream processing units.

A relevant feature of this problem is that tanks can store crude oil blends. Since calculating the outlet volume for each crude oil type involves the product of the total outlet volume and the concentration of that crude oil type, both of which are also variables, this introduces nonlinear, nonconvex constraints, resulting in a mixed-integer nonlinear programming (MINLP) problem, which is computationally challenging to solve.

In addition, two key features must be taken into account in this problem. First, tank classification based on the type of crude blend stored must be considered. Second, the problem requires consideration of a monthly scheduling horizon, which leads to a large-scale model that cannot be solved within a time frame compatible with the needs of refinery staff.

To tackle the deterministic problem, a mathematical programming model is developed based on a continuous-time formulation using global time points. Global time points refer to time instants at which the status of limited resources are evaluated. The model supports decision-making related to crude oil scheduling, such as determining when to unload vessels, which tanks should receive the crude oil, which tanks should feed the CDUs, and for how long and in what quantities, among other relevant decisions.

In addition, to address the nonlinear nature of the problem and the large model size arising from a one-month planning horizon, a novel solution strategy is proposed. This strategy combines a novel piecewise linear approximation technique using planes to approximate the product of two non-negative continuous variables, and a temporal decomposition approach consisting of the iterative resolution of two models: an aggregate model and a detailed model.

As previously mentioned, the problem is also approached from a stochastic perspective. The main motivation behind this part of the study lies in the fact that the scheduling phase typically does not account for unforeseen events, which may occasionally render the generated schedule infeasible. One of the main sources of uncertainty that can lead to this issue is the variability in the arrival dates of the ships delivering crude oil to the refinery. Addressing this uncertainty represents another key focus of the thesis, with the goal of achieving more robust schedules. Specifically, this work contributes to the development of stochastic models that incorporate risk management.

To this end, a two-stage stochastic programming model is developed, considering uncertainty in the arrival dates of the supplying ships. Although the resulting model is aimed at short-term scheduling applications and certain characteristics of the case study are simplified compared to the deterministic version, it maintains sufficient complexity to serve as proof of concept, allowing us to analyze the effects and benefits of accounting for uncertainty in this type of problem.

Performance analyses are conducted using the Expected Value of Perfect Information (EVPI) and the Value of the Stochastic Solution (VSS) to evaluate the potential advantages of the stochastic model over its deterministic counterpart. Moreover, the stochastic model is further extended to incorporate risk management by using the Conditional Value-at-Risk (CVaR) measure as the objective function, and solutions obtained for varying levels of risk aversion are analyzed.

1.5 Thesis structure

The thesis is structured as follows.

Chapter 2 provides a comprehensive description of the refinery, focusing on its objectives, topology, and process characteristics.

Chapter 3 presents the formulation of the mathematical programming model for optimizing crude oil operations scheduling. This includes an explanation of the continuous-time formulation used, the assumptions made, and a detailed description of the notation, logical and operational constraints, and the objective function.

Chapter 4 introduces a strategy based on piecewise linear approximation using planes to address the nonlinear, nonconvex constraints arising from blending crude oils in tanks. Additionally, a temporal decomposition method is developed to obtain feasible and high-quality solutions for a one-month scheduling horizon.

In Chapter 5, the performance of the model developed in Chapter 3 is analyzed together with the solution strategy proposed in Chapter 4. To this end, two case studies are solved using real data from the refinery under study. The resulting solutions are analyzed through various visualizations, including Gantt charts, the evolution of property values in the feed blends to the CDUs, as well as solution times and model statistics.

In Chapter 6, a two-stage stochastic programming model is developed to account for uncertainty in the arrival dates of ships, using a discrete set of scenarios with varying arrival times. The concepts of Expected Value of Perfect Information (EVPI) and Value of the Stochastic Solution (VSS) are introduced, enabling the evaluation of solutions obtained from the two-stage stochastic programming model against those derived from deterministic models. Furthermore, the model is extended to incorporate risk management by employing the Conditional Value-at-Risk (CVaR) measure as the objective function.

Chapter 7 focuses on evaluating the performance of the stochastic optimization approaches developed in Chapter 6. In this context, the EVPI and VSS measures are computed to assess the impact of incorporating uncertainty into the model. Furthermore, comparisons are made between the results of the risk-neutral approach and those obtained using the risk management approach at different confidence levels.

Finally, Chapter 8 draws conclusions based on the findings of the thesis and proposes potential future research directions that could build on the work presented here.

Chapter 2

Refinery description

This chapter provides a comprehensive description of the refinery under study, focusing on its objectives, topology, and process characteristics.

2.1 Centralized planning and refinery-level scheduling

At the organizational level, a central head office oversees several refineries, including the refinery studied in this work. The head office is responsible for defining each refinery's monthly production plan, specifying crude oil supply dates, and setting target demand at each facility.

While each refinery receives an individualized production plan, this planning process is not conducted in isolation. Instead, the head office develops plans by considering all refineries in an aggregated manner to establish synergies.

However, this overarching planning approach, which aims to optimize the overall system performance without accounting for time constraints, may result in challenging plans for certain refineries. This is mainly because the generated plan establishes targets, but assumes immediate availability of all crudes, including those to be delivered later in the month, and does not consider the specific operational and facility constraints of each refinery.

Upon receiving their respective production plans, each refinery's operations programming unit develops the detailed monthly crude oil operations schedule and communicates it to the head office.

2.2 Refinery topology

We consider a system that consists of three main areas: the marine terminal, the storing section, and the crude oil processing section. These areas are depicted in Figure 2.1 and described in detail below.

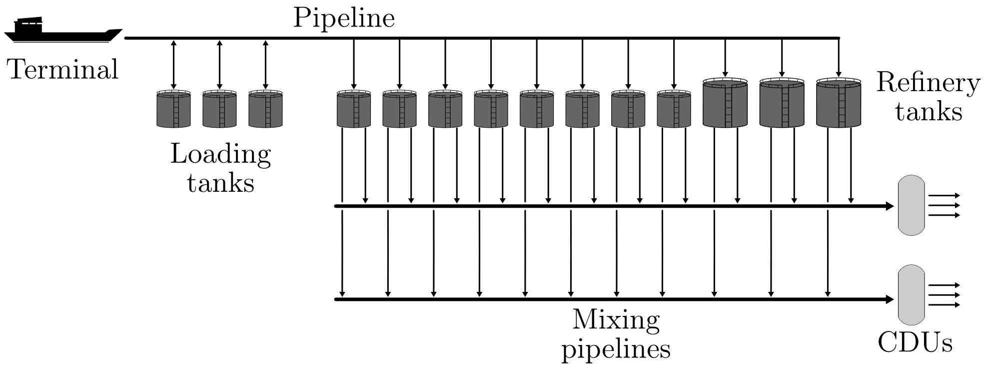

Figure 2.1: Schematic of refinery under study

2.2.1 Marine terminal

The terminal is the facility where the unloading operation of the crude oil transported by the ships begins, which means that it plays the role of a link between the ships and the refinery, where the supplied crude is stored in tanks and then processed. Here we analyze the case of a single-dock terminal, so only one ship can be unloaded at a time.

This configuration may result in some vessels waiting to begin unloading, while others may complete unloading later than their scheduled departure dates. Both scenarios lead to operational costs, demurrage costs in the former case and tardiness costs in the latter.

2.2.2 Storing section

The storing section is connected to the marine terminal by a pipeline and, as its name indicates, is composed of tanks that store the crude oil coming from the terminal.

Typically, refineries involve two types of tanks: storage and charging tanks. In our case, only storage tanks are available. Since the storage capacity is limited and there are many types of crude oil based on their composition, the tanks are not exclusively dedicated to a single type of crude oil; in other words, it is possible to store blends of crude oils.

Although only storage tanks are used in this case, they can be divided into two types: loading tanks and refinery tanks. The difference between one type and the other is that the loading tanks cannot feed the CDUs since they are not physically connected to them, they can only store crudes and transfer them to the refinery tanks. Discharge tanks are located halfway between the port and the refinery tank area. In the refinery under study, there are three loading tanks and eleven refinery tanks.

Tank and crude oil classification

There are seven grades used to classify both tanks and crude oils: TASF, TPES, TM10, TMMF, TMBF, TBIA, and TLGR. Tank classification is based on rules that evaluate the tank's composition (i.e., the types and volumes of crude oils it contains). Each tank is assigned only one grade at a time, though this classification may vary over the planning horizon. The rules are described below:

- A tank is classified as TASF if it contains at least 65% TASF crude oil.
- A tank is classified as TPES if it contains between 40% and 65% of a mixture of TPES, TASF, and TM10 crude oils, or if it contains between 65% and 90% TM10 crude oil.
- A tank is classified as TMMF if it contains at least 40% TMMF crude oil or at least 10% of a mixture of TPES and TASF crude oils.
- A tank is classified as TMBF if it contains at least 65% TMBF crude oil and no more than 5% of a mixture of TASF, TPES, and TMMF crude oils.
- A tank is classified as TBIA if it contains at least 95% TBIA crude oil and no more than 5% of a mixture of TASF, TPES, and TMMF crude oils.
- A tank is classified as TLGR if it contains at least 65% TLGR crude oil and no more than 5% of a mixture of TASF, TPES, and TMMF crude oils.
- A tank is classified as TM10 if it contains at least 90% TM10 crude oil.

A key consideration in handling crude oil is its unloading from ships, as the grade of the receiving tanks must be taken into account. Crude oil can be unloaded into tanks of different grades; however, a priority scale links crude oil grades to corresponding receiving tank grades.

As an example, Table 2.1 illustrates the priority scale for two types of crudes, C1 and C2, where a value of 7 represents the most preferred tank grade, and a value of 0 indicates that unloading into that tank grade is not permitted. In the case of C1 crude, it should ideally be discharged into a TBIA tank. If that option is unavailable, the next preference is a TMBF tank, and so on. For C2 crude, the preferred tank is TMBF, and if this is not possible, the next option would be a TLGR tank, and so on.

Table 2.1: Priority scale.

CRUDE	TASF	TPES	TMMF	TMBF	TBIA	TLGR	TM10
C1	0	0	5	6	7	6	0
C2	0	0	5	7	0	6	0

2.2.3 Crude oil processing section

Crude oil is composed of a mixture of hydrocarbons, and the petroleum refining process aims to separate this crude oil into useful fractions and convert some of the hydrocarbons into higher value products.

Typically, crude oil is first distilled in a crude distillation unit (CDU), also known as an atmospheric distillation unit. As oil is fed into the crude distillation unit, the first step is desalting, which is achieved by heating the crude to a temperature of 100-150 °C, allowing salts to be removed in the desalter.

After desalting, the crude oil is further heated to temperatures of up to 400°C as it is fed into the atmospheric distillation tower. This tower separates the crude oil into different products or fractions based on their boiling points. At temperatures above 400°C, the oil would thermally crack or break apart, disrupting the distillation process.

The products obtained and the corresponding temperature ranges (expressed in °C) at which each product is recovered are shown below:

- Gases (<30)
- Light gasoline (30-85)
- Heavy gasoline (85-175)
- Kerosene (175-250)
- Diesel or light gas oil (250-330)
- Heavy gas oil (330-355)
- Atmospheric gas oil (355-385)
- Atmospheric residue (>385)

Atmospheric residue consists of the heaviest hydrocarbons that do not vaporize. These are drawn from the bottom of the atmospheric tower.

Continuing the description of Figure 2.1, the refinery tanks are connected to the crude distillation unit area through a piping system (mixing pipelines) where final crude blends are created to achieve the flows and properties required by each CDU. Ensuring that these blends meet feed quality specifications is critical; failure to do so can result in processing inefficiencies, product quality degradation, or CDU disruptions.

Although estimating the properties of each crude type is beyond the scope of this work, as they are provided as input data for determining the blend properties, this is an area of significant importance. Current research is exploring innovative methods to predict crude oil properties more rapidly, cost-effectively, and with reduced environmental impact. Among these, artificial neural networks (ANN) have shown promise for efficiently estimating crude oil properties, thus enhancing the speed and accuracy of blend planning processes [35].

In the problem at hand, the following properties are considered when optimizing the scheduling of crude oil operations.

- Total acid number (TAN): is the quantity of potassium hydroxide in milligrams, which is required to neutralize the acids in one gram of oil.
- Atmospheric residue yield (RA): is the fraction, typically expressed by weight, of heavy hydrocarbons produced from input of crude oil to the atmospheric distillation unit.
- Cetane index (CTI): is a measure of how long it takes diesel fuel to ignite after injection. A higher cetane index indicates that the fuel will combust more quickly in the engine.
- Specific gravity (SPG): is a dimensionless quantity defined as the ratio of the density of a given substance (in this case, crude oil) to the density of water at 4°C.

Moreover, three modes of operation or types of crude processes are carried out in the refinery: standard, asphaltic, and low-sulfur fuel oil. For each process, there are specific recipes that indicate the grades of tanks permitted in preparing the feed blends.

The refinery operates two crude distillation units that process the crude blends to meet the established demand for each process, but they are not identical. Figure 2.2 illustrates the processes allowed in each CDU, the corresponding recipes for each process and the types of tanks allowed in each recipe.

For CDU 1, two processes can be performed: standard and asphaltic.

- Standard process: This process follows the M1 recipe, which allows for TASF, TPES, TMMF, TLGR, TMBF, or TBIA tanks.
- Asphaltic process: There are two possible recipes for this process.
  - Recipe M2: This recipe requires TASF, TM10, TMMF, TLGR, TMBF or TBIA grade tanks.
  - Recipe M3: This alternative recipe requires the use of TASF, TMMF, TLGR, TMBF, or TBIA tanks.

For CDU 2, two processes can be run: standard and low sulfur (LS) fuel.

- Standard process: The standard process in CDU 2 uses the same M1 recipe as in CDU 1.
- LS fuel process: This process follows recipe M4, which requires TBIA or TLGR grade tanks.

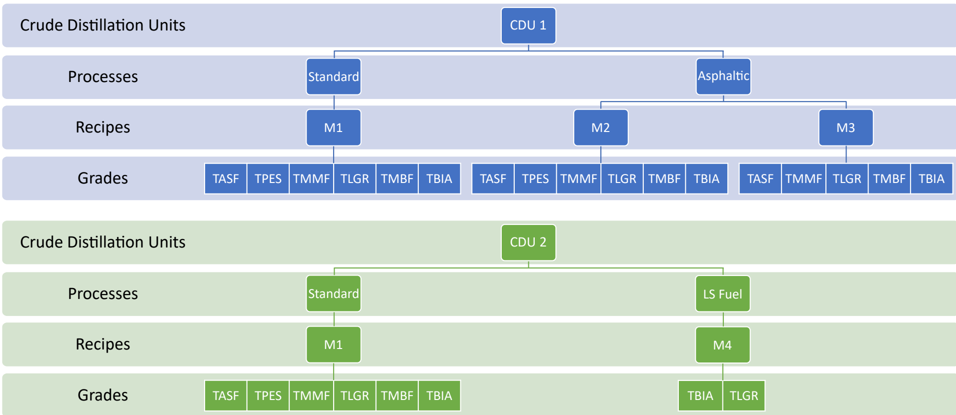

Figure 2.2: Processes and recipes

Regarding the asphaltic process, it is important to note that only one is conducted each month, with start and end dates known at the beginning of the scheduling horizon. These fixed dates must be considered when defining the schedule. For the remaining processes, campaign dates are not fixed and are determined as part of the optimization process.

2.2.4 Downstream processing units

As previously mentioned, the system includes two crude distillation units, specifically atmospheric distillation units, referred to as CDU 1 and CDU 2. Also, additional processing units located downstream of these CDUs are considered in this study:

- Vacuum distillation unit (V3): This unit receives atmospheric residue from CDU 1 and is used either to produce asphalt or to produce residue that serves as feedstock for CK6 to perform the standard process.
- Visbreaker (VB3): This unit processes atmospheric residue from CDU 2 and is used to produce low-sulfur (LS) fuel or to produce feedstock for CK6 to run the standard process.
- Coker (CK6): This unit receives residue from either V3 or VB3 to perform the standard process.

It is worth noting that V3 and VB3 units are also referred to as intermediate units, a nomenclature that is frequently used in subsequent chapters.

This area also includes dedicated tanks for storing residues. These tanks are used to segregate any excess from the residue streams, allowing for later use as feedstock in the V3, VB3, or CK6 units if the output yields from the crude processes do not meet the target feed rates for these units. Specifically, there are two tanks for CDU 1 residue: one for asphalt production and one for the standard process. For CDU 2, there are also two tanks: one for low-sulfur fuel production and one for the standard process. These four tanks are called intermediate tanks. In addition, there are two tanks upstream of CK6: one for V3 residue and the other for VB3 residue. These two tanks are referred to as final tanks.

In this work, the planning of units located downstream of the atmospheric crude distillation units is also considered. This involves taking into account the yields of the crude mixtures, with a focus on the volumes of residues produced. In this way, it is possible to manage the feeding of these downstream units and to manage the inventory levels in the dedicated residue storage tanks. It should be noted that a detailed analysis of each downstream unit is beyond the scope of this work.

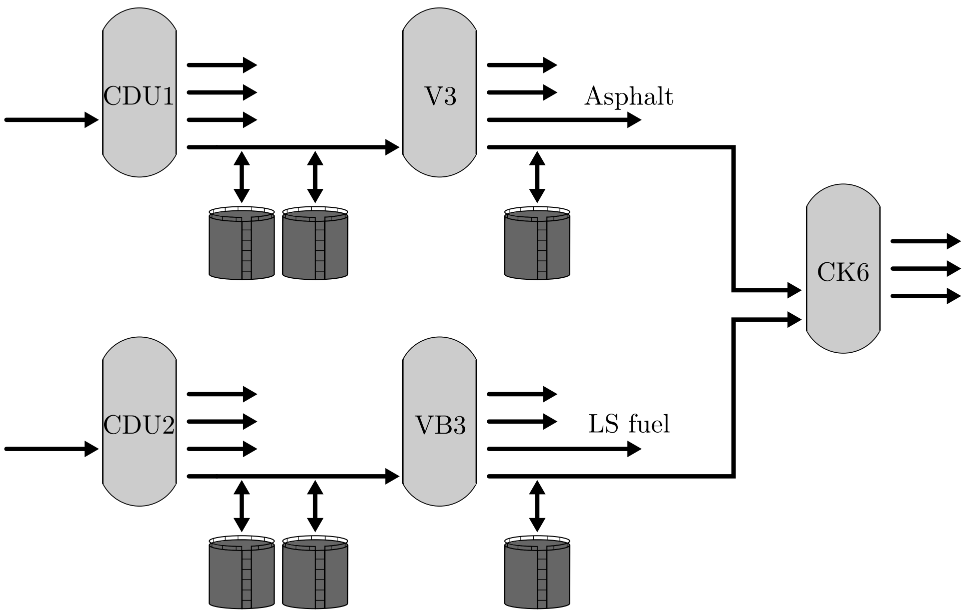

Figure 2.3: Schematic of downstream processing units

Chapter 3

Deterministic model

In this work, we make use of mathematical programming to address the problem of crude oil operations scheduling. This problem presents a large number of logical and operational constraints that are not simple to model. Therefore, the challenge lies in developing a model with sufficient complexity to faithfully represent the process but, at the same time, robust and able to be solved in a time according to the users' needs.

Before the development of the model, we must select the time representation to be used. We can choose between two basic approaches: discrete-time formulation and continuous-time formulation.

3.1 Time representation

In discrete-time models, the scheduling horizon is divided into fixed-length intervals or slots, with events (any change in operations) occurring only at the beginning or end of each interval. This approach facilitates the formulation of the model, in particular regarding the synchronization of events among different resources, but it has the disadvantage that the size of the model, its computational efficiency, and the accuracy of the solutions obtained strongly depend on the number of time intervals defined. Increasing the number of intervals improves solution precision but also enlarges the model, requiring more computational effort to solve.

Otherwise, if a smaller number of intervals is defined, the precision of the solution will decrease, and there is a risk of obtaining suboptimal or even infeasible solutions since, for a fixed horizon, a small number of slots would imply that no actions would be taken for an extended period. As a result of this, infeasibility can arise for various reasons. For example, tanks could be emptied (interrupting the feed to units), and certain properties in the blends might not be met, among others. It is also possible that the number of intervals may not be sufficient to execute the feasible solution that involves the least possible number of operations over the entire horizon.

One alternative to improve the time precision of the solution and to decrease the number of variables involved is the use of models based on continuous-time formulation, where timing decisions are explicitly represented as a set of continuous variables defining the exact times at which the events take place [7].

Continuous-time formulations can be represented in different ways, including using global time points or unit-specific time events for network processes, as can be seen in [7]. With global time points, also called MOS-SST in [36], a common time grid is used for all resources. The intervals of time (slots) represent the time between two consecutive events that take place at any resource of the process. In contrast, the unit-specific time events approach (or MOS representation in [36]) defines a unique time grid for each resource, allowing for different operations to start at different times for the same event point.

In our study, we have chosen the option characterized by global time points as a compromise between the ease of synchronization and the complexity of the corresponding continuous-time model. This continuous-time formulation is performed as follows:

- The scheduling horizon is divided into consecutive variable-length slots, synchronized across all the resources (vessels, tanks, and crude distillation units).
- Three mutually exclusive states are defined for the tanks: loading, unloading, and idle.
- A new slot is activated whenever a resource changes its state.
- Even so, a resource can maintain its state during consecutive time slots.

To make it more understandable, Figure 3.1 depicts an example involving two tanks and three slots whose start and end times are represented by the variables is_s and ts_s, respectively. Initially, both tanks are idle during slot 1, whose start time is $is_{s1} = 0$. After some time, tank 1 begins loading, and slot 1 ends at that moment, starting slot 2 with $is_{s2} = ts_{s1}$, which is the end time of slot 1. It is important to note that in slot 2, the state of tank 2 (“Idle") has not changed, while the state of tank 1 has changed to “Loading". Then, after some time, tank 2 starts unloading, which ends time slot 2 and begins a new slot 3, whose start time is $is_{s3} = ts_{s2}$. From this moment on, the new state of tank 2 is “Unloading", while the state of tank 1 remains “Loading". Notice that the slot number imposes a precedence of events over time and that, in the proposed formulation, not only the start time is the same for operations belonging to the same slot, but also the duration of the operations.

Figure 3.2 shows the same example using a discrete-time formulation. In this case, the horizon is divided into seven intervals of fixed length. The first event (start loading tank 1) occurs at the beginning of the third interval, that is, at time point t_2; the second event (start unloading tank 2) occurs at the beginning of the sixth interval, at time point t_5. It should be noted that if a longer interval length were defined, the number of intervals would decrease, but so would the precision of the solution.

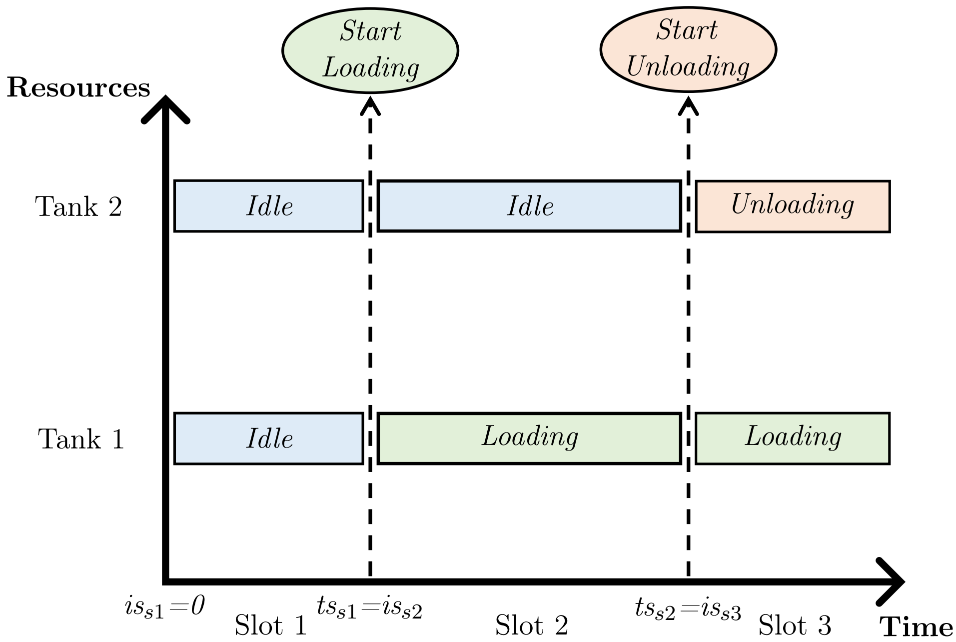

Figure 3.1: Continuous-time representation (global time points).

3.2 Model assumptions

In the development of the mathematical programming model, we must consider the specific features of the problem under study, corresponding to the system depicted in Figure 2.1, and make certain simplifications. The following assumptions have been taken into account when formulating the model:

1. There is only one pipeline connecting the terminal with the refinery, and only one vessel can unload at a time.
2. A vessel that has started unloading crude can leave the terminal once it is completely emptied.
3. Unloading of vessels and loading tanks cannot be carried out simultaneously as both use the same pipeline.
4. Each vessel carries a single type of crude oil and it is considered that the pipeline has a negligible volume compared to the volume to be

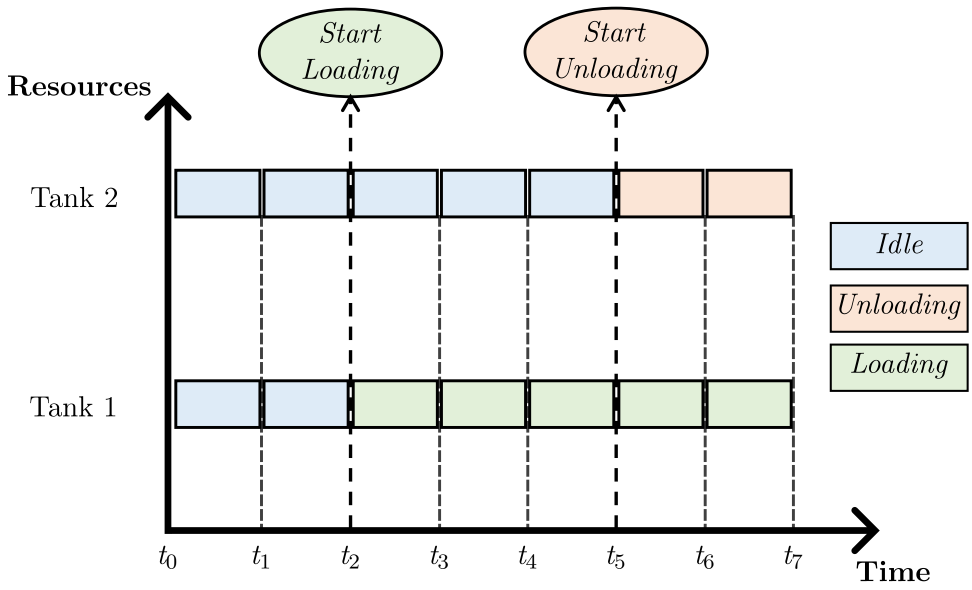

Figure 3.2: Discrete-time representation.

unloaded.

5. A maximum number of tanks can be loaded simultaneously.

6. No tank can be simultaneously loaded and unloaded.

7. After receiving crude oil, refinery and loading tanks must remain idle for a certain period to allow for brine settling and removal before feeding a CDU or a refinery tank, respectively.

8. There is a maximum number of refinery tanks that can feed a CDU at the same time, and the time to change over tanks is negligible.

9. A new grade, "undefined" (TUND), is established to represent the status of tanks when they do not meet any of the classification rules.

10. A perfect mixing of the crudes supplied to a CDU from different tanks occurs in the mixing pipelines.

11. Each intermediate tank is dedicated to storing the residue of a specific mixture and can only be loaded or unloaded if the corresponding recipe is being processed.

12. Each final tank is dedicated to the storage of residues destined for the standard process, so it cannot be loaded unless the standard process is being carried out.

13. The inventory level per crude in the final tanks is not tracked, as no properties or yields are calculated downstream of these tanks. This prevents the inclusion of a large number of non-linear constraints.

14. The processing of blends in different units (CDUs, intermediate units, and the coker unit) occurs simultaneously, with no delay time between units.

15. It is not allowed to stop feeding the processing units (CDUs, intermediate units, and coker unit).

Under these assumptions, a scheduling model of the process operation, which includes balances, constraints, and other relevant factors, was developed. The following sections describe the nomenclature of sets, parameters, and variables, as well as the model constraints and the objective function, respectively.

3.3 Notation

The notation of the developed model is shown below. Sets and parameters are written in uppercase, while variables are written in lowercase. Besides, each one of them is listed in alphabetical order.

3.3.1 Sets

- C: crude oils $\{c_1, c_2, ..., c_C\}$.
- CCL: mapping of crude oils to their respective classes $\{(c_1, TBIA), ..., (c_C, TASF)\}$.
- CL: tank and crude grades $\{TASF, TPES, TM10, TMMF, TMBF, TBIA, TLGR, TUND\}$.
- IU: intermediate units $\{V3, VB3\}$.
- K: key properties of crude oil $\{TAN, CTI, RA, SPG, MDS, SPGRA, RV\}$.
- MCL: classes allowed in each recipe $\{(mx_1, TASF), ..., (mx_4, TBIA)\}$.
- MX: mixtures or recipes for processes $\{mx_1, mx_2, mx_3, mx_4\}$.
- P: processes $\{Standard, Asphalt, LSfuel\}$.
- PM: recipes allowed in each process $\{(Standard, mx_1), (Asphalt, mx_2), ..., (LSfuel, mx_4)\}$.
- Q: all tanks in the refinery $\{q_1, q_2, ..., qf_2\}$.

- QF: final tanks $\{qf_1, qf_2\}$.
- $QFIU$: mapping of final tanks to intermediate units. This set indicates which final tanks can be loaded by intermediate unit iu $\{(qf_1, V3), (qf_2, VB3)\}$.
- QI: intermediate tanks $\{qi_1, qi_2, qi_3, qi_4\}$.
- QIM: allocation of tanks to recipes. This set indicates which mixture may be stored in the intermediate tank qi $\{(qi_1, mx_1), (qi_2, mx_2), ..., (qi_4, mx_4)\}$.
- QIU: mapping of intermediate tanks to crude distillation units. This set indicates which intermediate tanks can be loaded by CDU u $\{(qi_1, CDU1), ..., (qi_4, CDU2)\}$.
- QL: loading tanks $\{q_{12}, q_{13}, q_{14}\}$.
- QR: refinery tanks $\{q_1, ..., q_{11}\}$.
- S: time slots $\{s_1, ..., s_S\}$.
- U: crude distillation units $\{CDU1, CDU2\}$.
- UIU: alignment of CDUs with intermediate units. This set indicates which CDU u is connected to the intermediate unit iu $\{(CDU1, V3), (CDU2, VB3)\}$.
- UM: recipes allowed in each crude distillation unit $\{(CDU1, mx_1), ..., (CDU2, mx_4)\}$.
- UP: processes allowed in each CDU $\{(CDU1, Standard), ..., (CDU2, LSfuel)\}$.
- USQ: upstream tanks (loading and refinery tanks) $\{q_1, ..., q_{14}\}$.
- VC: mapping of vessels to the crude oils they carry. This set indicates which type of crude oil c is transported by each vessel v, i.e., its elements are pairs (v, c), where v is in V, and c is in C $\{(v_1, c_1)..., (v_V, c_C)\}$.
- V: vessels $\{v_1, ..., v_V\}$.

3.3.2 Parameters

- AT_v: arrival time of vessel.
- $CAPQL_q$: minimum inventory level limit in a tank.
- $CAPQU_q$: maximum capacity of a tank.

- $CDMG_v$: demurrage or sea waiting cost.
- $CCLP_{c,cl}$: priority of unloading crude oil c in a tank of grade cl.
- $CEP_{iu,p}$: cost due to overproduction concerning the demand of process p in intermediate unit iu.
- $CTDN_v$: departure tardiness cost.
- $CSLKMAX_{k,u}$: cost related to violation of upper boundary of key property.
- $CSLKMIN_{k,u}$: cost related to violation of lower boundary of key property.
- $CSP_{iu,p}$: cost due to underproduction concerning the demand of process p in intermediate unit iu.
- $DEMIUP_{iu,p}$: demand of process p in intermediate unit iu.
- DH: length of scheduling horizon.
- EDT_v: expected departure time of vessel.
- ETA: end-time of asphalt campaign.
- $FRIUL_{iu}$: minimum inlet flow to intermediate unit.
- $FRIUU_{iu}$: maximum inlet flow to intermediate unit.
- $FRCKL$: minimum inlet flow to coker unit.
- $FRCKU$: maximum inlet flow to coker unit.
- $FRUL_u$: minimum inlet flow to CDU.
- $FRUU_u$: maximum inlet flow to CDU.
- $FRVL_v$: minimum crude oil transfer rate from vessel.
- $FRVU_v$: maximum crude oil transfer rate from vessel.
- H: end-time of the scheduling horizon.
- $H0$: start-time of scheduling horizon.
- $LFRQL_q$: minimum tank loading flow rate.
- $LFRQU_q$: maximum tank loading flow rate.
- NQV: maximum number of tanks a vessel can load simultaneously.

- NQU: maximum number of refinery tanks from which a CDU can be fed simultaneously.
- $NTQU$: maximum number of refinery tanks that can be unloaded at the same time.
- $PRC_{c,k}$: key property k value in crude c.
- $PRL_{u,k}$: lower bound of property k in unit u.
- $PRU_{u,k}$: upper bound of property k in unit u.
- $SETT$: time to settle and remove the brine.
- SLI_q: initial inventory level in tank.
- $SLIC_{q,c}$: initial amount of crude c in tank q.
- STA: start-time of asphalt campaign.
- $TUNDUB$: maximum proportion of tanks with grade “undefined" allowed in recipes.
- $UFRQL_q$: minimum tank unloading flow rate.
- $UFRQU_q$: maximum tank unloading flow rate.
- $VOL_{v,c}$: amount of crude c in vessel v.
- WD: water density.

3.3.3 Continuous variables

The domain of continuous variables is the set of non-negative real numbers.

- dmg_v: demurrage of vessel v.
- $dmg1_{v,s}$: auxiliary variable to calculate dmg_v.
- ds_s: length of slot s.
- $exprod_{iu,p}$: overproduction volume in intermediate unit iu for process p.
- is_s: start-time of slot s.
- $shprod_{iu,p}$: shortage volume in intermediate unit iu for process p.
- $slkmax_{k,u,s}$: slack variable for property upper bounds.
- $slkmin_{k,u,s}$: slack variable for property lower bounds.

- $stock_{q,s}$: total inventory level in q at the beginning of s.
- $stockc_{q,c,s}$: amount of c in q at the beginning of s.
- $stockcend_{q,c,s}$: amount of c in q at the end of the horizon.
- $stockend_{q,s}$: total level in q at the end of the scheduling horizon.
- tdn_v: departure tardiness of vessel v.
- $tdn1_{v,s}$: auxiliary variable to calculate tdn_v.
- ts_s: end-time of slot s.
- $vciu_{c,iu,s}$: amount of crude oil c fed into intermediate unit iu during slot s.
- $vcqiu_{c,qi,iu,s}$: amount of crude c transferred from intermediate tank qi to intermediate unit iu during slot s.
- $vcqq_{c,ql,qr,s}$: volume of crude c transferred from ql to qr during s.
- $vcqu_{c,qr,u,s}$: amount of crude c transferred from refinery tank qr to CDU u during s.
- $vcra_{c,u,s}$: output volume of atmospheric residue by crude from CDU u during s.
- $vcuq_{c,u,qi,s}$: volume of crude c loaded from CDU u into intermediate tank qi during slot s.
- $vcuiu_{c,u,iu,s}$: volume of atmospheric residue by crude transferred from CDU u to intermediate unit iu.
- $vcvq_{c,v,q,s}$: amount of crude c transferred from v to q during s.
- $vcvqcl_{c,v,q,s,cl}$: amount of crude c transferred from vessel v to tank q with grade cl during slot s.
- $vcv_{c,v,s}$: total amount of crude c unloaded from v during s.
- $vcku_s$: total input volume to the coker unit during slot s.
- $vcl_{cl,q,u,s}$: volume transferred from tank qr with grade cl to CDU u during slot s.
- $vcra_{c,u,s}$: output volume of atmospheric residue by crude from CDU u during s.
- $viu_{iu,s}$: input volume to intermediate unit iu during slot s.

- $viucku_{iu,s}$: input volume to the coker unit from line relative to intermediate unit iu, during slot s.
- $viuq_{iu,qf,s}$: volume loaded from intermediate unit iu into final tank qf during slot s.
- $vmu_{mx,u,s}$: volume of recipe mx processed by CDU u during slot s.
- $vprod_{iu,p,s}$: output volume from intermediate unit iu to satisfy demand of process p during slot s.
- $vqcku_{qf,s}$: volume transferred from final tank qf to coker unit during slot s.
- $vqq_{ql,qr,s}$: amount transferred from ql to qr during s.
- $vqiu_{qi,iu,s}$: volume transferred from intermediate tank qi to intermediate unit iu.
- $vqu_{qr,u,s}$: volume transferred from refinery tank qr to CDU u during s.
- $vra_{u,s}$: output volume of atmospheric residue from CDU u during s.
- $vrv_{iu,s}$: output volume of residue from intermediate unit iu during s.
- $vu_{u,s}$: total amount of crude mix transferred to CDU u during s.
- $vuq_{u,qi,s}$: volume loaded from CDU u into intermediate tank qi during slot s.
- $vuiu_{u,iu,s}$: volume of atmospheric residue transferred from CDU u to intermediate unit iu.
- $vvq_{v,q,s}$: total amount of crude transferred from v to q during s.

3.3.4 Binary variables

- $fuv_{v,s}$: is equal to 1 if vessel v finishes its unloading at the end of s.
- $lq_{q,s}$: is equal to 1 if tank q is receiving crude during s.
- $suv_{v,s}$: is equal to 1 if vessel v starts its unloading at the beginning of s.
- $tbia1_{q,s}, tbia2_{q,s}$: auxiliary binary variables for classifying TBIA tanks.
- $tlgr1_{q,s}, tlgr2_{q,s}$: auxiliary binary variables for classifying TLGR tanks.
- $tmbf1_{q,s}, tmbf2_{q,s}$: auxiliary binary variables for classifying TMBF tanks.

- $tmmf1_{q,s}, tmmf2_{q,s}$: auxiliary binary variables for classifying TMMF tanks.

- $tpes1_{q,s}, tpes2_{q,s}, tpes3_{q,s}, tpes4_{q,s}, tpes5_{q,s}, tpes6_{q,s}$: auxiliary binary variables for classifying TPES tanks.

- $tq_{cl,q,s}$: is equal to 1 if tank q belongs to grade cl at the beginning of slot s.

- $uq_{q,s}$: is equal to 1 if tank q is delivering crude during slot s.

- $uqu_{qr,u,s}$: is equal to 1 if refinery tank qr feeds CDU u during slot s.

- $uv_{v,s}$: is equal to 1 if vessel v is unloading during s.

- $xmu_{mx,u,s}$: is equal to 1 if recipe mx is selected to be processed in CDU u during slot s.

3.4 Constraints

Vessel unloading and upstream tank loading

A vessel is unloaded during a slot s ($uv_{v,s}$) if it was unloading during the previous slot ($uv_{v,s-1}$) and has not finished yet ($fuv_{v,s-1}$), or if it starts at the beginning of the current slot ($suv_{v,s}$) (3.1).

$$uv_{v,s} = uv_{v,s-1} + suv_{v,s} - fuv_{v,s-1} \quad \forall v \in V, \ \forall s \in S \mid s > 1 \tag{3.1}$$

A vessel can finish unloading at the end of a slot as long as it was unloading during that slot (3.2). Note that if the ship is not being unloaded ($uv_{v,s} = 0$), then $fuv_{v,s}$ will be equal to zero.

$$uv_{v,s} \geq fuv_{v,s} \quad \forall v \in V, \ \forall s \in S \tag{3.2}$$

All ships may start and end unloading only once within the scheduling horizon, (3.3) and (3.4) respectively.

$$\sum_{s \in S} suv_{v,s} = 1 \quad \forall v \in V \tag{3.3}$$

$$\sum_{s \in S} fuv_{v,s} = 1 \quad \forall v \in V \tag{3.4}$$

Only one vessel can unload at any moment (assumption 1).

$$\sum_{v \in V} uv_{v,s} \leq 1 \quad \forall s \in S \tag{3.5}$$

Unloading of vessels and loading tanks cannot be carried out simultaneously.

$$uv_{v,s} + uq_{ql,s} \leq 1 \quad \forall ql \in QL, \ \forall v \in V, \ \forall s \in S \tag{3.6}$$

A maximum of NQV tanks can be loaded simultaneously (assumption 5). That is, the sum of the binary variable indicating that a tank is being loaded ($lq_{q,s}$), over all tanks, must be less than or equal to NQV. Note that this constraint applies only to upstream tanks, i.e. loading and refinery tanks.

$$\sum_{q \in USQ} lq_{q,s} \leq NQV \quad \forall s \in S \tag{3.7}$$

When unloading crude oil from a vessel, it is necessary to have at least one tank being loaded. Note that if no tank is receiving a load, then $\sum_{q \in USQ} lq_{q,s}$ equals zero.

$$\sum_{q \in USQ} lq_{q,s} \geq uv_{v,s} \quad \forall v \in V,\ \forall s \in S \tag{3.8}$$

A loading tank cannot be unloaded if there is no refinery tank connected.

$$\sum_{qr \in QR} lq_{qr,s} \geq uq_{ql,s} \quad \forall ql \in QL,\ \forall s \in S \tag{3.9}$$

Only one loading tank can unload at a time.

$$\sum_{ql \in QL} uq_{ql,s} \leq 1 \quad \forall s \in S \tag{3.10}$$

A loading tank cannot be loaded if there is no vessel docked.

$$lq_{ql,s} \leq \sum_{v \in V} uv_{v,s} \quad \forall ql \in QL,\ \forall s \in S \tag{3.11}$$

A refinery tank cannot be loaded if there is no vessel docked or a loading tank unloading.

$$lq_{qr,s} \leq \sum_{v \in V} uv_{v,s} + \sum_{ql \in QL} uq_{ql,s} \quad \forall qr \in QR,\ \forall s \in S \tag{3.12}$$

At most NQU refinery tanks are allowed to concurrently feed a CDU (assumption 8).

$$\sum_{qr \in QR} uqu_{qr,u,s} \leq NQU \quad \forall u \in U,\ \forall s \in S \tag{3.13}$$

At most NTQU refinery tanks can operate at a time.

$$\sum_{qr \in QR} \sum_{u \in U} uqu_{qr,u,s} \leq NTQU \quad \forall s \in S \tag{3.14}$$

Each CDU must continually process feedstock coming from refinery tanks (assumption 15). This means that each CDU must be fed by at least one tank at all times.

$$\sum_{qr \in QR} uqu_{qr,u,s} \geq 1 \quad \forall u \in U,\ \forall s \in S \tag{3.15}$$

A tank must be in one of the three states (i.e., loading, unloading, or idle) during a given slot. If a tank is neither loading nor unloading, then it is idle. This is represented by the case where both variables $lq_{q,s}$ and $uq_{q,s}$ take value zero during the same time slot.

$$lq_{q,s} + uq_{q,s} \leq 1 \quad \forall q \in Q, \ \forall s \in S \tag{3.16}$$

A refinery tank must be discharging ($uq_{qr,s}$) if it is feeding a CDU ($uqu_{qr,u,s}$).

$$uq_{qr,s} \geq uqu_{qr,u,s} \quad \forall qr \in QR, \ \forall u \in U, \ \forall s \in S \tag{3.17}$$

A refinery tank must be feeding at least one unit ($\sum_{u \in U} uqu_{qr,u,s}$) if it is being unloaded ($uq_{qr,s}$).

$$\sum_{u \in U} uqu_{qr,u,s} \geq uq_{qr,s} \quad \forall qr \in QR, \ \forall s \in S \tag{3.18}$$

The end-time of a slot is equal to its start-time plus its length.

$$ts_s = is_s + ds_s \quad \forall s \in S \tag{3.19}$$

The start-time of a slot coincides with the end-time of the previous slot. This implies that the durations of operations of all resources are synchronized in each slot.

$$is_s = ts_{s-1} \quad \forall s \in S \mid s > 1 \tag{3.20}$$

The total length of the time slots must be equal to the length of the scheduling horizon.

$$\sum_{s \in S} ds_s = DH \tag{3.21}$$

The big-M method, explained by Winston and Goldberg [37], is applied to compute the amount of crude unloaded to tanks.

This is a common approach in mixed-integer programming used to represent implications or conditional constraints by introducing a sufficiently large constant M. Moreover, the big-M method is widely used to avoid formulating nonlinear constraints, especially when the original condition involves the product of continuous and binary variables. By replacing such nonlinear expressions with linear inequalities involving M, it becomes possible to preserve the linearity of the model.

To illustrate the big-M method, consider the following case where x and w are non-negative continuous variables and y is a binary variable. We want to model the following logic:

- If $y = 1$, then $x \leq w$
- If $y = 0$, then $x = 0$

A direct nonlinear formulation would be:

$$x \leq w \cdot y \tag{3.22}$$

$$x \geq 0 \tag{3.23}$$

$$w \geq 0 \tag{3.24}$$

$$y \in \{0, 1\} \tag{3.25}$$

However, constraint (3.22) is nonlinear, which we want to avoid. To linearize it, we apply the big-M method as follows:

$$x \leq w + M \cdot (1 - y) \tag{3.26}$$

$$x \leq M \cdot y \tag{3.27}$$

$$x \geq 0 \tag{3.28}$$

$$w \geq 0 \tag{3.29}$$

$$y \in \{0, 1\} \tag{3.30}$$

In this way, when the binary variable $y = 1$:

- Constraint (3.26) becomes $x \leq w$
- Constraint (3.27) becomes $x \leq M$, with M being sufficiently large

Thus, x is forced to take a value less than or equal to w.

On the other hand, if $y = 0$:

- Constraint (3.26) becomes $x \leq w + M$
- Constraint (3.27) becomes $x \leq 0$

Therefore, x is forced to be equal to 0.

In this way, the nonlinear constraint is replaced by a linear system that models the same behavior. The constant M must be chosen large enough to avoid infeasibilities, but not excessively large to prevent numerical issues.

In this case, the M values are determined based on physical limits. For example, M1 takes into account the volume carried by each vessel.

It is important to mention that there is an alternative formulation known as the convex hull method [38], which yields tighter relaxations but results in larger models. Given that the inherent complexity of the problem addressed already leads to a large-scale model, we have decided to adopt the big-M method. As previously mentioned, the drawback of this method compared to the convex hull approach is that it results in models with looser relaxations. Therefore, when opting for the big-M method, it is crucial to define M values that are coherent and based on the specific characteristics of the process.

We calculate the upper bound of the volume of crude oil unloaded from a ship to a tank ($vcvq_{c,v,q,s}$) as a function of the maximum loading flow rate that the receiving tank admits ($LFRQU_q$) and the duration of the operation (ds_s).

$$vcvq_{c,v,q,s} \leq LFRQU_q \cdot ds_s \quad \forall q \in USQ,\ \forall (v,c) \in VC,\ \forall s \in S \tag{3.31}$$

We calculate the lower bound of the volume of crude oil unloaded from a ship to a tank ($vcvq_{c,v,q,s}$) as a function of the minimum loading flow rate the tank admits ($LFRQL_q$) and the duration of the operation. Note that this constraint is activated only if ship v is unloading ($uv_{v,s} = 1$) and tank q is receiving a load ($lq_{q,s} = 1$).

$$\begin{aligned} &vcvq_{c,v,q,s} \geq LFRQL_q \cdot ds_s - M1_v \cdot (2 - uv_{v,s} - lq_{q,s}) \\ &\forall q \in USQ,\ \forall (v,c) \in VC,\ \forall s \in S \end{aligned} \tag{3.32}$$

If vessel v is not being unloaded, then $vcvq_{c,v,q,s}$ will be equal to zero.

$$vcvq_{c,v,q,s} \leq VOL_{v,c} \cdot uv_{v,s} \quad \forall q \in USQ,\ \forall (v,c) \in VC,\ \forall s \in S \tag{3.33}$$

If tank q is not being loaded, then $vcvq_{c,v,q,s}$ will be equal to zero.

$$vcvq_{c,v,q,s} \leq VOL_{v,c} \cdot lq_{q,s} \quad \forall q \in USQ,\ \forall (v,c) \in VC,\ \forall s \in S \tag{3.34}$$

Also, we use the big-M method to calculate the crude volume unloaded from a vessel during a slot s ($vcv_{c,v,s}$). The upper and lower bounds are obtained by 3.35 and 3.36, respectively.

$$vcv_{c,v,s} \leq FRVU_v \cdot ds_s \quad \forall (v,c) \in VC,\ \forall s \in S \tag{3.35}$$

$$vcv_{c,v,s} \geq FRVL_v \cdot ds_s - M1_v \cdot (1 - uv_{v,s}) \quad \forall (v,c) \in VC,\ \forall s \in S \tag{3.36}$$

If ship v is not being unloaded, then $vcv_{c,v,s}$ will be equal to zero.

$$vcv_{c,v,s} \leq VOL_{v,c} \cdot uv_{v,s} \quad \forall (v,c) \in VC,\ \forall s \in S \tag{3.37}$$

The crude volume unloaded from a vessel during a slot s is equal to the sum of volumes unloaded to each tank.

$$vcv_{c,v,s} = \sum_{q \in USQ} vcvq_{c,v,q,s} \quad \forall (v,c) \in VC,\ \forall s \in S \tag{3.38}$$

The total volume loaded into a tank during a slot s is calculated by using (3.39).

$$vvq_{v,q,s} = \sum_{\substack{c \in C \\ (v,c) \in VC}} vcvq_{c,v,q,s} \quad \forall q \in USQ, v \in V,\ \forall s \in S \tag{3.39}$$

To make each vessel unload fully during the scheduling horizon (assumption 2), we use (3.40).

$$\sum_{s \in S} vcv_{c,v,s} = VOL_{v,c} \quad \forall (v,c) \in VC \tag{3.40}$$

Figure 3.3 illustrates the part of the process modeled based on the previous constraints, specifically the unloading of crude oil from vessels and the charging of loading and refinery tanks. The figure highlights the main variables involved in the constraints: the volume of crude oil unloaded from a vessel ($vcv_{c,v,s}$), the volume charged into a loading tank ($vcvq_{c,v,ql,s}$), and the volume charged into a refinery tank ($vcvq_{c,v,qr,s}$).

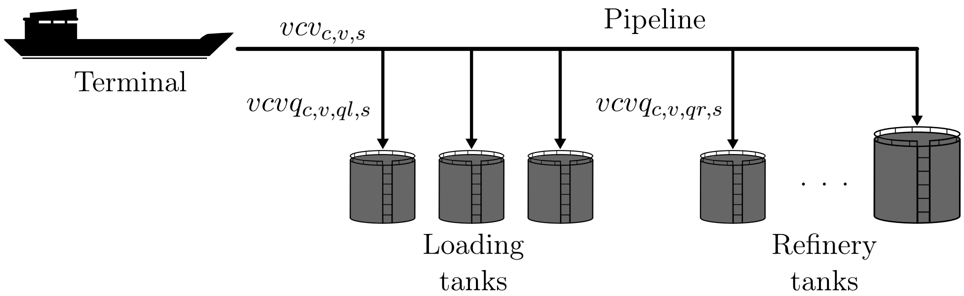

Figure 3.3: Vessel unloading representation.

Refinery tank unloading and CDU feeding

Constraints (3.41)-(3.43) compute the amount of crude unloaded from refinery tanks.

Constraint 3.41 establishes the upper bound of the amount of crude oil discharged from tank qr to unit u ($vqu_{qr,u,s}$) as a function of the maximum discharge flow rate from the tank ($UFRQU_{qr}$) and the duration of the operation.

$$vqu_{qr,u,s} \leq UFRQU_{qr} \cdot ds_s \quad \forall qr \in QR, \ \forall u \in U, \ \forall s \in S \tag{3.41}$$

We compute the lower bound as a function of the minimum discharge flow rate of tank qr ($UFRQL_{qr}$) and the duration of the operation. This constraint will be activated if refinery tank qr is feeding CDU u ($uqu_{qr,u,s} = 1$). Here, the value of M2 is based on the maximum capacity of the tank.

$$\begin{aligned} & vqu_{qr,u,s} \geq UFRQL_{qr} \cdot ds_s - M2_{qr} \cdot (1 - uqu_{qr,u,s}) \\ & \forall qr \in QR, \ \forall u \in U, \ \forall s \in S \end{aligned} \tag{3.42}$$

If tank qr is not feeding unit u, then $vqu_{qr,u,s}$ will be equal to zero.

$$vqu_{qr,u,s} \leq CAPQU_{qr} \cdot uqu_{qr,u,s} \quad \forall qr \in QR, \ \forall u \in U, \ \forall s \in S \tag{3.43}$$

The total volume unloaded from refinery tank qr to crude distillation unit u during slot s is calculated by using (3.44).

$$vqu_{qr,u,s} = \sum_{c \in C} vcqu_{c,qr,u,s} \quad \forall qr \in QR,\ \forall u \in U,\ \forall s \in S \tag{3.44}$$

The total feed to CDU u during slot s ($vu_{u,s}$) is equal to the sum of the volumes transferred from each refinery tank ($vqu_{qr,u,s}$).

$$vu_{u,s} = \sum_{qr \in QR} vqu_{qr,u,s} \quad \forall u \in U,\ \forall s \in S \tag{3.45}$$

Constraints (3.46) and (3.47) set the upper and lower limits for $vu_{u,s}$, respectively

$$vu_{u,s} \leq FRUU_u \cdot ds_s \quad \forall u \in U,\ \forall s \in S \tag{3.46}$$

$$vu_{u,s} \geq FRUL_u \cdot ds_s \quad \forall u \in U,\ \forall s \in S \tag{3.47}$$

Figure 3.4 represents the unloading of refinery tanks and the feeding of crude distillation units. In addition, the most relevant variables defined in the modeling of this process are shown. In this case, two refinery tanks, qr and qr', are depicted. The first is associated with variables $vqu_{qr,CDU1,s}$ and $vqu_{qr,CDU2,s}$, which represent the volume discharged from this tank to CDUs 1 and 2, respectively. Similarly, the variables associated with tank qr' are shown. Finally, the variables representing the total input volume to the CDUs are also included, where $vu_{CDU1,s}$ corresponds to CDU 1 and $vu_{CDU2,s}$ to CDU 2.

Volume transfer from loading tank to refinery tank

The following set of constraints allows for the computation of the volumes transferred from the loading tanks to the refinery tanks.

Constraint 3.48 determines the upper bound of the total volume unloaded from a loading tank to all receiving refinery tanks, depending on the maximum discharge flow rate limit of loading tank ql and the duration of slot s.

$$\sum_{qr \in QR} vqq_{ql,qr,s} \leq UFRQU_{ql} \cdot ds_s \quad \forall ql \in QL,\ \forall s \in S \tag{3.48}$$

The lower limit of the volume discharged from a loading tank to the refinery tanks is determined by constraint 3.49. This cosntraint is activated if and only if the loading tank ql is unloading ($uq_{ql,s} = 1$)

$$\sum_{qr \in QR} vqq_{ql,qr,s} \geq UFRQL_{ql} \cdot ds_s - M2_{ql} \cdot (1 - uq_{ql,s}) \quad \forall ql \in QL,\ \forall s \in S \tag{3.49}$$

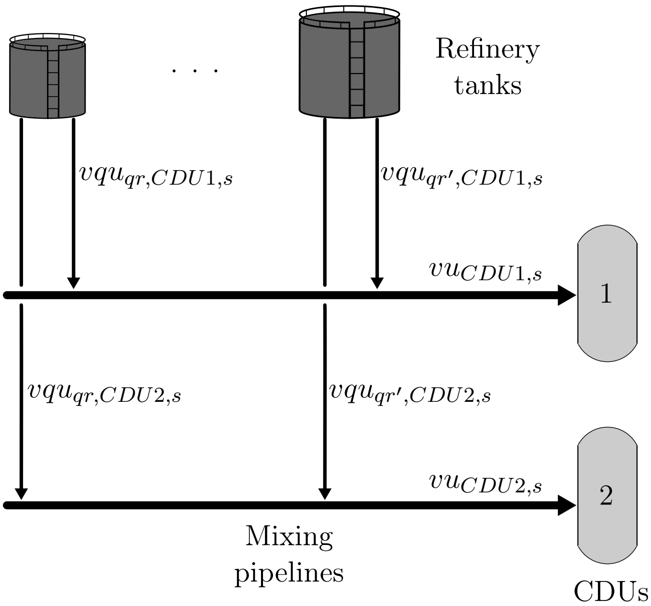

Figure 3.4: Refinery tank unloading and CDU feeding representation.

Constraint 3.50 sets the upper bound for the inlet volume of a refinery tank being fed by a loading tank. Unlike equation 3.48, in this case, the summation is performed over the loading tanks and considers the maximum loading flow rate of the refinery tank qr. Additionally, only one term in the summation can take a positive value, as only one loading tank is allowed to unload at a time.

$$\sum_{ql \in QL} vqq_{ql,qr,s} \leq LFRQU_{qr} \cdot ds_s \quad \forall qr \in QR, \ \forall s \in S \tag{3.50}$$

The following constraint calculates the lower bound of the inlet volume to a refinery tank fed by a loading tank. When formulating this constraint, it is important to consider that a refinery tank may receive inflow from either a loading tank or a ship. Additionally, a loading tank cannot be unloaded at the same time as a ship. Therefore, this constraint is activated only if, during the same time slot, the refinery tank qr is being loaded and the loading tank ql is being unloaded.

$$\sum_{ql' \in QL} vqq_{ql',qr,s} \geq LFRQL_{qr} \cdot ds_s - M2_{qr} \cdot (2 - lq_{qr,s} - uq_{ql,s}) \quad \forall ql \in QL, \ \forall qr \in QR, \ \forall s \in S \tag{3.51}$$

If tank ql is not unloading ($uq_{ql,s} = 0$), then it cannot feed any refinery tank ($\sum_{qr \in QR} vqq_{ql,qr,s} = 0$).

$$\sum_{qr \in QR} vqq_{ql,qr,s} \leq CAPQU_{ql} \cdot uq_{ql,s} \quad \forall ql \in QL, \ \forall s \in S \tag{3.52}$$

If the refinery tank qr is not being charged ($lq_{qr,s} = 0$), then it cannot receive any load from a loading tank.

$$\sum_{ql \in QL} vqq_{ql,qr,s} \leq CAPQU_{qr} \cdot lq_{qr,s} \quad \forall qr \in QR, \ \forall s \in S \tag{3.53}$$

The volume transferred from loading tank ql to refinery tank qr during slot s is computed by (3.54).

$$vqq_{ql,qr,s} = \sum_{c \in C} vcqq_{c,ql,qr,s} \quad \forall ql \in QL, \ \forall qr \in QR, \ \forall s \in S \tag{3.54}$$

Figure 3.5 depicts the unloading of loading tanks and the charging of refinery tanks. In this case, the variables associated with the volumes loaded into refinery tanks qr ($vqq_{ql,qr,s}$) and qr' ($vqq_{ql,qr',s}$), both supplied from loading tank ql, are shown.

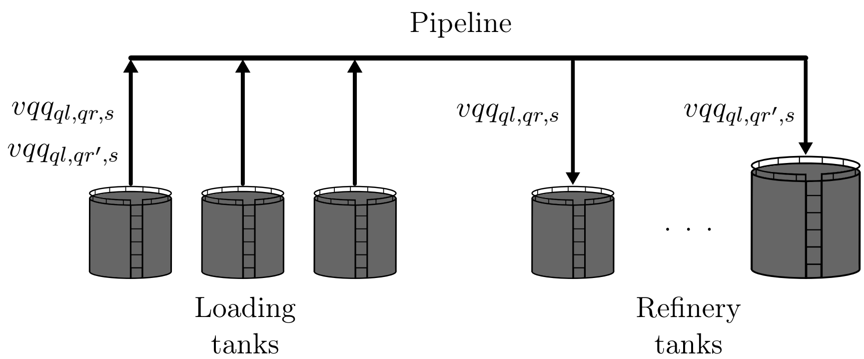

Figure 3.5: Loading tank unloading and refinery tank charging representation.

Demand fulfillment

To calculate the difference between processed volume and required demand of each process, we use (3.55) and (3.56). Constraint 3.55 sets the over-production volume at unit iu for process p ($exprod_{iu,p}$) as the difference between the volume processed over the horizon and the established demand. Constraint 3.56 computes the shortage volume at unit iu ($shprod_{iu,p}$) as the difference between the required demand and the volume processed over the horizon. Note that if the demand is not met, then $shprod_{iu,p}$ will be greater than zero, and constraint 3.55 will be idle. Otherwise, if the demand value is exceeded, then $exprod_{iu,p}$ will be greater than zero, and constraint 3.56 will be idle.

It is important to remember that UP is the set that defines which processes can be performed in each CDU u, while UIU is the set that specifies which intermediate unit iu is connected to CDU u. In this way, the UIU set is used in conjunction with UP to link processes to intermediate units.

$$exprod_{iu,p} \geq \sum_{s \in S} vprod_{iu,p,s} - DEMIUP_{iu,p} \quad \forall (u, iu) \in UIU, \ \forall (u,p) \in UP \tag{3.55}$$

$$shprod_{iu,p} \geq DEMIUP_{iu,p} - \sum_{s \in S} vprod_{iu,p,s} \quad \forall (u, iu) \in UIU, \ \forall (u,p) \in UP \tag{3.56}$$

Feed mixture production

Each CDU is fed with only one recipe at a time. Set UM specifies which recipes or mixtures are allowed in each unit.

$$\sum_{\substack{mx \in MX \\ (u,mx) \in UM}} xmu_{mx,u,s} = 1 \quad \forall u \in U, \ \forall s \in S \tag{3.57}$$

The volume of the mixture processed in unit u must be equal to the total feed volume (3.58). For all other recipes not selected, it is zero (3.59). The value of M3 is defined according to the maximum input flow rate to the CDU u and the duration of the horizon.

$$\sum_{\substack{mx \in MX \\ (u,mx) \in UM}} vmu_{mx,u,s} = vu_{u,s} \quad \forall u \in U, \ \forall s \in S \tag{3.58}$$

$$vmu_{mx,u,s} \leq M3_u \cdot xmu_{mx,u,s} \quad \forall (u, mx) \in UM, \ \forall s \in S \tag{3.59}$$

To calculate the volume that each CDU receives from each tank belonging to a certain grade, we use (3.60) and (3.61).

The variable $tq_{cl,qr,s}$ takes the value 1 if tank qr belongs to class cl during slot s. Since a tank can belong to only one class at a time, constraint (3.60) ensures that the variable $vcl_{cl,qr,u,s}$ can take a positive value only for the class cl associated with qr during s, and it will be equal to zero for all other classes.

$$vcl_{cl,qr,u,s} \leq CAPQU_{qr} \cdot tq_{cl,qr,s} \quad \forall cl \in CL, \ \forall qr \in QR, \ \forall u \in U, \ \forall s \in S \tag{3.60}$$

Constraint (3.61) allows relating the variables $vcl_{cl,qr,u,s}$ and $vqu_{qr,u,s}$, and states that the volume of class cl transferred from refinery tank qr to crude distillation unit u must be equal to the total volume transferred from qr to u. It should be noted that only one term will be positive in the summation since the refinery tanks belong to only one class at a time.

$$\sum_{cl \in CL} vcl_{cl,qr,u,s} = vqu_{qr,u,s} \quad \forall qr \in QR, \ \forall u \in U, \ \forall s \in S \tag{3.61}$$

The set MCL indicates the tank grades allowed in each recipe mx. Constraint (3.62) states that if the recipe mx is being processed in unit u during slot s ($xmu_{mx,u,s} = 1$), then the volume transferred from tanks of class cl, which is not supported in the mixture, must be zero.

$$\begin{aligned} &\sum_{qr \in QR} vcl_{cl,qr,u,s} \leq M3_u \cdot (1 - xmu_{mx,u,s}) \\ &\forall (u, mx) \in UM, \ \ \forall s \in S, \ \ \forall (mx, cl) \notin MCL \end{aligned} \tag{3.62}$$

Equation (3.63) indicates the maximum proportion of tanks with grade "undefined" allowed in the recipes.

$$\sum_{qr \in QR} vcl_{cl,qr,u,s} \leq TUNDUB \cdot vu_{u,s} \quad \forall u \in U, \ \ \forall s \in S, \ \ cl = \text{"TUND"} \tag{3.63}$$

As mentioned in Chapter 2, a single asphalt campaign is conducted each month, with known start and end dates. Constraint (3.64) states that an asphalt mixture mx can only be processed in the corresponding unit u during slot s if the start date of that slot (is_s) is later than the planned start date of the asphalt campaign (STA).

Although the variable $xmu_{mx,u,s}$ used in this constraint does not explicitly depend on the process p, defining the constraint in this way is consistent. It is important to note that not all mixtures are compatible with all processes, and not all processes can be carried out in every CDU. These restrictions reflect actual operational constraints at the refinery. Based on this, the sets UM and PM were introduced. The former contains pairs (u, mx), where u represents a CDU and mx represents a mixture or recipe; such pairs indicate the mixtures admitted in each distillation unit. Similarly, the set PM contains pairs (p, mx), where p represents a specific process and mx a mixture; these pairs indicate the mixtures allowed in each process p.

Since constraint (3.64) refers specifically to the asphalt campaign, it is only defined for p equal to *Asphalt*, and consequently, only for the recipes mx that are compatible with this process. Although $xmu_{mx,u,s}$ is not indexed by p, restricting the domain of the constraint in this way ensures that it only applies to the relevant mixtures. Otherwise, the constraint would be enforced for all (p, mx) pairs, including those where p refers to processes other than *Asphalt*, incorrectly implying that the start date for each of those processes must also be greater than or equal to STA.

$$\begin{aligned} &is_s \geq STA \cdot xmu_{mx,u,s} \\ &\forall (u, mx) \in UM, \ \ \forall s \in S, \ \ \forall (p, mx) \in PM, \ \ p = \text{"Asphalt"} \end{aligned} \tag{3.64}$$

In addition, the end date of the slot (ts_s) during which the asphalt mixture mx is being processed cannot be later than the end date established for the asphalt campaign (ETA), as stated in (3.65).

$$ts_s \leq ETA + H \cdot (1 - xmu_{mx,u,s})$$
$$\forall (u, mx) \in UM,\ \forall s \in S,\ \forall (p, mx) \in PM,\ p = \text{"Asphalt"} \tag{3.65}$$

Constarints (3.66) and (3.67) calculate the upper and lower limits, respectively, of the volumes of mixtures fed into each unit, considering the limits on input flow rates to the distillation units and the duration of the slots. It should be noted that these constraints apply exclusively to the combinations of mx, u, and p present in the sets UM, UP, and PM.

$$vmu_{mx,u,s} \leq FRUU_u \cdot ds_s$$
$$\forall (u, p) \in UP,\ \forall (u, mx) \in UM,\ \forall (p, mx) \in PM,\ \forall s \in S \tag{3.66}$$

$$vmu_{mx,u,s} \geq FRUL_u \cdot ds_s - M3_u \cdot (1 - xmu_{mx,u,s})$$
$$\forall (u, p) \in UP,\ \forall (u, mx) \in UM,\ \forall (p, mx) \in PM,\ \forall s \in S \tag{3.67}$$

Using (3.68) and (3.69), we calculate the volume of crude oil unloaded from a vessel in a tank whose grade is *cl*.

The variable $vcvqcl_{c,v,q,s,cl}$ represents the volume of crude oil c unloaded from vessel v into tank q whose grade is cl during slot s. Constraint (3.68) specifies that this variable can only take a value greater than zero if tank q belongs to class cl ($tq_{cl,q,s}$), otherwise it takes the value zero.

$$vcvqcl_{c,v,q,s,cl} \leq CAPQU_q \cdot tq_{cl,q,s}$$
$$\forall (v, c) \in VC,\ \forall q \in USQ,\ \forall s \in S,\ \forall cl \in CL \tag{3.68}$$

Constraint (3.69) relates the variable $vcvqcl_{c,v,q,s,cl}$ to the variable $vcvq_{c,v,q,s}$ which represents the volume of crude oil c discharged from vessel v into tank q. In case $vcvq_{c,v,q,s}$ is greater than zero, only one term of the summation on the left-hand side will be positive, since each tank is associated with only one class during each slot.

$$\sum_{cl \in CL} vcvqcl_{c,v,q,s,cl} = vcvq_{c,v,q,s} \quad \forall (v, c) \in VC,\ \forall q \in USQ,\ \forall s \in S \tag{3.69}$$

Feed mixture properties computation

The following is the set of constraints that establishes the concentration limits of the properties (total acid number, atmospheric residue yield, cetane index, and specific gravity) when feeding the CDUs. The total acid number (TAN) and specific gravity (SPG) have only an upper bound, while the cetane index (CTI) has only a lower bound. The atmospheric residue yield (RA) has both a lower and an upper limit. Furthermore, the calculation of the values of all properties are linear in mass, except SPG, which is linear in volume, so we must convert the volume $vcqu_{c,qr,u,s}$ to mass when calculating them.

To make this clearer to the reader, let us take as an example constraint (3.70), which sets the upper bound for total acid number.

Starting from the left side of the inequality, a double summation is presented, first over the set of crudes and then over the set of refinery tanks.

Within the double summation, the variable $vcqu_{c,qr,u,s}$ is multiplied by $PRC_{c,SPG}$ (specific gravity of crude c) and by WD (density of water) to obtain the mass value of each crude transferred from qr to u. This amount is then multiplied by the parameter $PRC_{c,TAN}$, which is the value of TAN for each crude oil c.

The reader may expect that this entire term should be divided by the total mass entering the u unit during s to calculate TAN concentration. However, this would introduce nonlinearities.

To avoid these nonlinearities, the total mass ($\sum_{c\in C}\sum_{qr\in QR} vcqu_{c,qr,u,s} \cdot PRC_{c,SPG} \cdot WD$) is multiplied on the right side of the inequality by the parameter $PRU_{u,TAN}$, which represents the upper limit of property TAN in unit u. Moreover, the slack variable $slkmax_{k,u,s}$ is added, which allows the upper bound to be violated, but incurs a penalty cost in the objective function.

$$\begin{aligned}&\sum_{c\in C}\sum_{qr\in QR} vcqu_{c,qr,u,s} \cdot PRC_{c,SPG} \cdot WD \cdot PRC_{c,k} \leq \\ &(PRU_{u,k} \cdot \sum_{c\in C}\sum_{qr\in QR} vcqu_{c,qr,u,s} \cdot PRC_{c,SPG} \cdot WD) + slkmax_{k,u,s} \\ &\forall u \in U,\ \forall s \in S,\ k = \text{"TAN"}\end{aligned} \tag{3.70}$$

It is important to mention that the value of the property CTI must be calculated based on the middle distillate mass. Therefore, constraint (3.71) includes the factor $PRC_{c,MDS}$, which represents the middle distillate yield for crude oil c.

$$\begin{aligned}&\sum_{c\in C}\sum_{qr\in QR} vcqu_{c,qr,u,s} \cdot PRC_{c,SPG} \cdot WD \cdot PRC_{c,MDS} \cdot PRC_{c,k} \geq \\ &(PRL_{u,k} \cdot \sum_{c\in C}\sum_{qr\in QR} vcqu_{c,qr,u,s} \cdot PRC_{c,SPG} \cdot WD \cdot PRC_{c,MDS}) - slkmin_{k,u,s} \\ &\forall u \in U,\ \forall s \in S,\ k = \text{"CTI"}\end{aligned} \tag{3.71}$$

Constraints (3.72) and (3.73) set the upper and lower bounds, respectively, for the atmospheric residue yield of the feed mixture.

$$\begin{aligned}&\sum_{c\in C}\sum_{qr\in QR} vcqu_{c,qr,u,s} \cdot PRC_{c,SPG} \cdot WD \cdot PRC_{c,k} \leq \\ &(PRU_{u,k} \cdot \sum_{c\in C}\sum_{qr\in QR} vcqu_{c,qr,u,s} \cdot PRC_{c,SPG} \cdot WD) + slkmax_{k,u,s} \\ &\forall u \in U,\ \forall s \in S,\ k = \text{"RA"}\end{aligned} \tag{3.72}$$

$$\sum_{c\in C}\sum_{qr\in QR} vcqu_{c,qr,u,s}\cdot PRC_{c,SPG}\cdot WD\cdot PRC_{c,k} \geq$$
$$(PRL_{u,k}\cdot \sum_{c\in C}\sum_{qr\in QR} vcqu_{c,qr,u,s}\cdot PRC_{c,SPG}\cdot WD) - slkmin_{k,u,s} \tag{3.73}$$
$$\forall u\in U,\ \forall s\in S,\ k = \text{"RA"}$$

For SPG, since it is linear in volume, it is not necessary to convert the value of $vcqu_{c,qr,u,s}$ to mass. Furthermore, it is possible to use $vu_{u,s}$ directly on the right-hand side of the inequality.

$$\sum_{c\in C}\sum_{qr\in QR} vcqu_{c,qr,u,s}\cdot PRC_{c,k} \leq (PRU_{u,k}\cdot vu_{u,s}) + slkmax_{k,u,s} \tag{3.74}$$
$$\forall u\in U,\ \forall s\in S \text{ where } k = \text{"SPG"}$$

Upstream tank inventory level

The amount of crude c in each refinery tank at the start of slot s ($stockc_{qr,c,s}$) is calculated as the amount of crude oil c at the beginning of the previous slot ($stockc_{qr,c,s-1}$), plus the load of crude oil c received during the previous slot ($\sum_{v\in V} vcvq_{c,v,qr,s-1} + \sum_{ql\in QL} vcqq_{c,ql,qr,s-1}$), minus the volume of crude oil c discharged from tank qr to the units during the previous slot ($\sum_{u\in U} vcqu_{c,qr,u,s-1}$).

$$stockc_{qr,c,s} = stockc_{qr,c,s-1} + \sum_{\substack{v\in V\\(v,c)\in VC}} vcvq_{c,v,qr,s-1}$$
$$- \sum_{u\in U} vcqu_{c,qr,u,s-1} \quad \forall qr\in QR,\ \forall c\in C,\ \forall s\in S \mid s>1 \tag{3.75}$$

Similarly, the amount of crude c in each loading tank at the start of slot s ($stockc_{ql,c,s}$) is calculated as the amount of crude oil c at the beginning of the previous slot ($stockc_{ql,c,s-1}$), plus the load of crude oil c received during the previous slot ($\sum_{v\in V} vcvq_{c,v,qr,s-1}$), minus the volume of crude oil c discharged from tank ql to the refinery tanks during the previous slot ($\sum_{qr\in QR} vcqq_{c,ql,qr,s-1}$).

$$stockc_{ql,c,s} = stockc_{ql,c,s-1} + \sum_{\substack{v\in V\\(v,c)\in VC}} vcvq_{c,v,ql,s-1} - \sum_{qr\in QR} vcqq_{c,ql,qr,s-1}$$
$$\forall ql\in QL,\ \forall c\in C,\ \forall s\in S \mid s>1 \tag{3.76}$$

The amount of crude c in each tank at the beginning of the horizon is given by (3.77).

$$stockc_{q,c,s} = SLIC_{q,c} \quad \forall q\in Q,\ \forall c\in C,\ s=1 \tag{3.77}$$

The amount of crude c in each refinery tank at the end of the horizon ($stockcend_{qr,c,s}$) is calculated as the amount of crude oil c at the beginning of the last slot of the horizon ($stockc_{qr,c,s}$), plus the amount of crude oil c received during the final slot ($\sum_{v \in V} vcvq_{c,v,qr,s} + \sum_{ql \in QL} vcqq_{c,ql,qr,s}$), minus the volume of crude oil c discharged from tank q to the units during that slot ($\sum_{u \in U} vcqu_{c,qr,u,s}$).

$$\begin{aligned} stockcend_{qr,c,s} &= stockc_{qr,c,s} + \sum_{\substack{v \in V \\ (v,c) \in VC}} vcvq_{c,v,qr,s} + \sum_{ql \in QL} vcqq_{c,ql,qr,s} \\ &- \sum_{u \in U} vcqu_{c,qr,u,s} \quad \forall qr \in QR,\ \forall c \in C,\ s = |S| \end{aligned} \tag{3.78}$$

The amount of crude c in each loading tank at the end of the horizon ($stockcend_{ql,c,s}$) is defined as the volume of crude oil c at the beginning of the last slot of the horizon ($stockc_{ql,c,s}$), plus the amount of crude oil c received from vessels during the final slot ($\sum_{v \in V} vcvq_{c,v,ql,s}$), minus the volume of crude oil c transferred from loading tank ql to the refinery tanks during that slot ($\sum_{qr \in QR} vcqq_{c,ql,qr,s}$).

$$\begin{aligned} stockcend_{ql,c,s} &= stockc_{ql,c,s} + \sum_{\substack{v \in V \\ (v,c) \in VC}} vcvq_{c,v,ql,s} - \sum_{qr \in QR} vcqq_{c,ql,qr,s} \\ &\forall ql \in QL,\ \forall c \in C,\ s = |S| \end{aligned} \tag{3.79}$$

The total level in each refinery tank at the start of slot s ($stock_{qr,s}$) and at the end of the horizon ($stockend_{qr,s}$) is given by (3.80)-(3.81).

$$\begin{aligned} stock_{qr,s} &= stock_{qr,s-1} + \sum_{v \in V} vvq_{v,qr,s-1} + \sum_{ql \in QL} vqq_{ql,qr,s-1} \\ &- \sum_{u \in U} vqu_{qr,u,s-1} \quad \forall qr \in QR,\ \forall s \in S \mid s > 1 \end{aligned} \tag{3.80}$$

$$\begin{aligned} stockend_{qr,s} &= stock_{qr,s} + \sum_{v \in V} vvq_{v,qr,s} + \sum_{ql \in QL} vqq_{ql,qr,s} \\ &- \sum_{u \in U} vqu_{qr,u,s} \quad \forall qr \in QR,\ s = |S| \end{aligned} \tag{3.81}$$

Regarding the loading tanks, the total level at the beginning of each slot ($stock_{ql,s}$) and at the end of the horizon ($stockend_{ql,s}$) is given by equations (3.82) and (3.83), respectively.

$$\begin{aligned} stock_{ql,s} &= stock_{ql,s-1} + \sum_{v \in V} vvq_{v,ql,s-1} - \sum_{qr \in QR} vqq_{ql,qr,s-1} \\ &\forall ql \in QL,\ \forall s \in S \mid s > 1 \end{aligned} \tag{3.82}$$

$$stockend_{ql,s} = stock_{ql,s} + \sum_{v \in V} vvq_{v,ql,s} - \sum_{qr \in QR} vqq_{ql,qr,s}$$
$$\forall ql \in QL, \ s = |S| \tag{3.83}$$

Moreover, both at the beginning of each slot (3.84) and at the end of the horizon (3.85), the total level in a tank q is equal to the sum of the volumes of each crude oil c stored in that tank.

As will be seen later, there is no inventory tracking by crude oil for the final tanks, so these constraints are only defined for loading tanks, refinery tanks, and intermediate tanks.

$$\sum_{c \in C} stockc_{q,c,s} = stock_{q,s} \quad \forall q \in (USQ \cup QI), \ \forall s \in S \tag{3.84}$$

$$\sum_{c \in C} stockcend_{q,c,s} = stockend_{q,s} \quad \forall q \in (USQ \cup QI), \ s = |S| \tag{3.85}$$

Equations (3.86)-(3.89) establish the physical limits for the inventory levels.

$$stock_{q,s} \leq CAPQU_q \quad \forall q \in Q, \ \forall s \in S \tag{3.86}$$

$$stock_{q,s} \geq CAPQL_q \quad \forall q \in Q, \ \forall s \in S \tag{3.87}$$

$$stockend_{q,s} \leq CAPQU_q \quad \forall q \in Q, \ s = |S| \tag{3.88}$$

$$stockend_{q,s} \geq CAPQL_q \quad \forall q \in Q, \ s = |S| \tag{3.89}$$

To ensure minimum settling time (assumption 7), we use (3.90). If a tank q receives a charge during slot s ($lq_{q,s} = 1$) and is discharged during slot s' ($uq_{q,sbis} = 1$), where s' is later than s, then the start time of slot s' must be greater than or equal to the end time of slot s, plus the time required to settle ($SETT$). This constraint is defined exclusively for loading tanks and refinery tanks.

$$is_{s'} - ts_s \geq SETT \cdot (lq_{q,s} + uq_{q,s'} - 1)$$
$$\forall q \in USQ, \ \forall s \in S, \ \forall s' \in S \mid s' > s \tag{3.90}$$

Vessel demurrage and tardiness

The discharge of crude oil from vessel v cannot start before its arrival time.

$$is_s \geq AT_v \cdot suv_{v,s} \quad \forall v \in V, \ \forall s \in S \tag{3.91}$$

The demurrage is calculated as the time elapsed between the arrival of a ship and the start of its unloading.

From constraints 3.92-3.93, the auxiliary variable $dmg1_{v,s}$ is calculated, which represents how many hours after its arrival a ship v has started unloading in slot s.

In case the ship has not started unloading at the beginning of slot s ($suv_{v,s} = 0$), then constraints 3.92 and 3.93 will be inactive. Otherwise, if the ship starts unloading at the beginning of slot s ($suv_{v,s} = 1$), then constraints 3.92 and 3.93 are activated, and the value of $dmg1_{v,s}$ is computed.

$$dmg1_{v,s} \geq is_s - AT_v \cdot suv_{v,s} - H \cdot (1 - suv_{v,s}) \quad \forall v \in V, \ \forall s \in S \tag{3.92}$$

$$dmg1_{v,s} \leq is_s - AT_v \cdot suv_{v,s} \quad \forall v \in V, \ \forall s \in S \tag{3.93}$$

From constraint 3.94, the demurrage of each ship is calculated as the summation of the auxiliary variable $dmg1_{v,s}$ over all slots.

$$dmg_v = \sum_{s \in S} dmg1_{v,s} \quad \forall v \in V \tag{3.94}$$

If vessel v leaves the terminal after its expected departure time EDT_v, it should pay a penalty that will be proportional to the departure tardiness (tdn_v). Analogously to the calculation of the demurrage, the tardiness of each ship (tdn_v) is computed from the auxiliary variable $tdn1_{v,s}$, using constraints (3.95)-(3.97).

$$tdn1_{v,s} \geq ts_s - EDT_v \cdot fuv_{v,s} - H \cdot (1 - fuv_{v,s}) \quad \forall v \in V, \ \forall s \in S \tag{3.95}$$

$$tdn1_{v,s} \leq H \cdot fuv_{v,s} \quad \forall v \in V, \ \forall s \in S \tag{3.96}$$

$$tdn_v = \sum_{s \in S} tdn1_{v,s} \quad \forall v \in V \tag{3.97}$$

Tank classification by composition

The following constraints classify tanks based on their composition, i.e., according to the types of crude oil and the volumes stored. These constraints are defined according to the criteria presented in Chapter 2.

Constraints (3.98) and (3.99) determine whether a tank belongs to the class TASF. The variable $tq_{TASF,q,s}$ equals 1 if the stock level of all TASF-class crudes present in q at the beginning of s represents more than 65% of the total stock level. It should be noted that if $tq_{TASF,q,s} = 1$, then (3.98) is idle, and (3.99) is activated. Otherwise, if tank q does not meet this criterion, then $tq_{TASF,q,s} = 0$, making constraint (3.99) idle while activating constraint (3.98).

Note that the summations are defined over the crude types c belonging to the set C, such that the pair (c, cl) belongs to the set CCL. The latter corresponds to the set that links each raw to its class. In both constraints, the element cl is explicitly declared equal to $TASF$. This notation for summations is used consistently throughout the thesis.

$$\sum_{\substack{c \in C \\ (c,TASF) \in CCL}} stockc_{q,c,s} - 0.65 \cdot stock_{q,s} \leq CAPQU_q \cdot tq_{TASF,q,s} \quad \forall q \in USQ, \ \forall s \in S \tag{3.98}$$

$$\sum_{\substack{c \in C \\ (c,TASF) \in CCL}} stockc_{q,c,s} - 0.65 \cdot stock_{q,s} \geq CAPQU_q \cdot (tq_{TASF,q,s} - 1) \quad \forall q \in USQ, \ \forall s \in S \tag{3.99}$$

Constraints (3.100)–(3.116) define the classification criteria for the class TPES. We must remember that there are two possible cases in which a tank can belong to TPES. First, a tank is classified as TPES if it contains between 40% and 65% of a mixture of TPES, TASF, and TM10 crude oils. Second, a tank is classified as TPES if it contains between 65% and 90% TM10 crude oil. Unlike the constraints developed for the class TASF, the definition of the constraints for the class TPES requires the introduction of auxiliary binary variables ($tpes1$ to $tpes6$).

In constraints (3.100) and (3.101), the auxiliary binary variable $tpes4_{q,s}$ equals 1 if the sum of the stock levels of all crudes of class TASF, TPES, and TM10 exceeds 40% of the total stock level present in q at the beginning of slot s.

$$\sum_{\substack{c \in C \\ (c,TASF) \in CCL}} stockc_{q,c,s} + \sum_{\substack{c \in C \\ (c,TPES) \in CCL}} stockc_{q,c,s} + \sum_{\substack{c \in C \\ (c,TM10) \in CCL}} stockc_{q,c,s} - 0.4 \cdot stock_{q,s} \geq CAPQU_q \cdot (tpes4_{q,s} - 1) \quad \forall q \in USQ, \ \forall s \in S \tag{3.100}$$

$$\sum_{\substack{c \in C \\ (c,TASF) \in CCL}} stockc_{q,c,s} + \sum_{\substack{c \in C \\ (c,TPES) \in CCL}} stockc_{q,c,s} + \sum_{\substack{c \in C \\ (c,TM10) \in CCL}} stockc_{q,c,s} - 0.4 \cdot stock_{q,s} \leq CAPQU_q \cdot tpes4_{q,s} \quad \forall q \in USQ, \ \forall s \in S \tag{3.101}$$

In constraints (3.102) and (3.103), the auxiliary binary variable $tpes6_{q,s}$ equals 1 if the sum of the stock levels of all crudes of class TASF, TPES, and TM10 is less than 65% of the total stock level present in q at the beginning of s.

$$\sum_{\substack{c \in C \\ (c,TASF) \in CCL}} stockc_{q,c,s} + \sum_{\substack{c \in C \\ (c,TPES) \in CCL}} stockc_{q,c,s} + \sum_{\substack{c \in C \\ (c,TM10) \in CCL}} stockc_{q,c,s} - 0.65 \cdot stock_{q,s} \leq CAPQU_q \cdot (1 - tpes6_{q,s}) \quad \forall q \in USQ,\ \forall s \in S \tag{3.102}$$

$$\sum_{\substack{c \in C \\ (c,TASF) \in CCL}} stockc_{q,c,s} + \sum_{\substack{c \in C \\ (c,TPES) \in CCL}} stockc_{q,c,s} + \sum_{\substack{c \in C \\ (c,TM10) \in CCL}} stockc_{q,c,s} - 0.65 \cdot stock_{q,s} \geq CAPQU_q \cdot (-tpes6_{q,s}) \quad \forall q \in USQ,\ \forall s \in S \tag{3.103}$$

Then, from constraints (3.104)–(3.106), it is established that the auxiliary binary variable $tpes1_{q,s}$ equals 1 if and only if both $tpes4_{q,s}$ and $tpes6_{q,s}$ are equal to 1. Otherwise, $tpes1_{q,s}$ equals zero.

$$tpes1_{q,s} \leq tpes4_{q,s} \quad \forall q \in USQ,\ \forall s \in S \tag{3.104}$$

$$tpes1_{q,s} \leq tpes6_{q,s} \quad \forall q \in USQ,\ \forall s \in S \tag{3.105}$$

$$tpes1_{q,s} \geq tpes4_{q,s} + tpes6_{q,s} - 1 \quad \forall q \in USQ,\ \forall s \in S \tag{3.106}$$

In constraints (3.107) and (3.108), the auxiliary binary variable $tpes3_{q,s}$ equals 1 if the sum of the stock levels of all TM10 crudes exceeds 65% of the total stock level present in q at the start of slot s.

$$\sum_{\substack{c \in C \\ (c,TM10) \in CCL}} stockc_{q,c,s} - 0.65 \cdot stock_{q,s} \geq CAPQU_q \cdot (tpes3_{q,s} - 1) \quad \forall q \in USQ,\ \forall s \in S \tag{3.107}$$

$$\sum_{\substack{c \in C \\ (c,TM10) \in CCL}} stockc_{q,c,s} - 0.65 \cdot stock_{q,s} \leq CAPQU_q \cdot tpes3_{q,s} \quad \forall q \in USQ,\ \forall s \in S \tag{3.108}$$

In constraints (3.109) and (3.110), the auxiliary binary variable $tpes5_{q,s}$ equals 1 if the sum of the stock levels of all TM10 crudes is less than 90% of the total stock level present in q at the start of s.

$$\sum_{\substack{c \in C \\ (c,TM10) \in CCL}} stockc_{q,c,s} - 0.9 \cdot stock_{q,s} \leq CAPQU_q \cdot (1 - tpes5_{q,s}) \quad \forall q \in USQ, \ \forall s \in S \tag{3.109}$$

$$\sum_{\substack{c \in C \\ (c,TM10) \in CCL}} stockc_{q,c,s} - 0.9 \cdot stock_{q,s} \geq CAPQU_q \cdot (-tpes5_{q,s}) \quad \forall q \in USQ, \ \forall s \in S \tag{3.110}$$

Then, from constraints (3.111)–(3.113), it is established that the auxiliary binary variable $tpes2_{q,s}$ equals 1 if and only if both $tpes3_{q,s}$ and $tpes5_{q,s}$ are equal to 1. Otherwise, $tpes2_{q,s}$ equals zero.

$$tpes2_{q,s} \leq tpes3_{q,s} \quad \forall q \in USQ, \ \forall s \in S \tag{3.111}$$

$$tpes2_{q,s} \leq tpes5_{q,s} \quad \forall q \in USQ, \ \forall s \in S \tag{3.112}$$

$$tpes2_{q,s} \geq tpes3_{q,s} + tpes5_{q,s} - 1 \quad \forall q \in USQ, \ \forall s \in S \tag{3.113}$$

Finally, constraints (3.114)–(3.116) determine that a tank belongs to class TPES if one of the two cases is fulfilled. That is, $tq_{TPES,q,s}$ equals 1 if $tpes1_{q,s}$ equals 1 or $tpes2_{q,s}$ equals 1.

$$tq_{TPES,q,s} \geq tpes1_{q,s} \quad \forall q \in USQ, \ \forall s \in S \tag{3.114}$$

$$tq_{TPES,q,s} \geq tpes2_{q,s} \quad \forall q \in USQ, \ \forall s \in S \tag{3.115}$$

$$tq_{TPES,q,s} \leq tpes1_{q,s} + tpes2_{q,s} \quad \forall q \in USQ, \ \forall s \in S \tag{3.116}$$

The constraints related to the remaining classification criteria follow the same logical structure as the cases explained above.

Constraints (3.117)–(3.122) define the classification criteria for class TMMF.

$$\sum_{\substack{c \in C \\ (c,TMMF) \in CCL}} stockc_{q,c,s} - 0.4 \cdot stock_{q,s} \geq CAPQU_q \cdot (tmmf1_{q,s} - 1) \quad \forall q \in USQ, \ \forall s \in S \tag{3.117}$$

$$\sum_{\substack{c \in C \\ (c,TMMF) \in CCL}} stockc_{q,c,s} - 0.4 \cdot stock_{q,s} \leq CAPQU_q \cdot tmmf1_{q,s} \\ \forall q \in USQ, \ \forall s \in S \tag{3.118}$$

$$\sum_{\substack{c \in C \\ (c,TASF) \in CCL}} stockc_{q,c,s} + \sum_{\substack{c \in C \\ (c,TPES) \in CCL}} stockc_{q,c,s} \\ - 0.1 \cdot stock_{q,s} \geq CAPQU_q \cdot (tmmf2_{q,s} - 1) \quad \forall q \in USQ, \ \forall s \in S \tag{3.119}$$

$$\sum_{\substack{c \in C \\ (c,TASF) \in CCL}} stockc_{q,c,s} + \sum_{\substack{c \in C \\ (c,TPES) \in CCL}} stockc_{q,c,s} \\ - 0.1 \cdot stock_{q,s} \leq CAPQU_q \cdot tmmf2_{q,s} \quad \forall q \in USQ, \ \forall s \in S \tag{3.120}$$

$$tq_{TMMF,q,s} \geq tmmf1_{q,s} \quad \forall q \in USQ, \ \forall s \in S \tag{3.121}$$

$$tq_{TMMF,q,s} \leq tmmf1_{q,s} + tmmf2_{q,s} \quad \forall q \in USQ, \ \forall s \in S \tag{3.122}$$

Constraints (3.123)–(3.129) represent the classification criteria for class TMBF.

$$\sum_{\substack{c \in C \\ (c,TMBF) \in CCL}} stockc_{q,c,s} - 0.65 \cdot stock_{q,s} \geq CAPQU_q \cdot (tmbf1_{q,s} - 1) \\ \forall q \in USQ, \ \forall s \in S \tag{3.123}$$

$$\sum_{\substack{c \in C \\ (c,TMBF) \in CCL}} stockc_{q,c,s} - 0.65 \cdot stock_{q,s} \leq CAPQU_q \cdot tmbf1_{q,s} \\ \forall q \in USQ, \ \forall s \in S \tag{3.124}$$

$$\sum_{\substack{c \in C \\ (c,TASF) \in CCL}} stockc_{q,c,s} + \sum_{\substack{c \in C \\ (c,TPES) \in CCL}} stockc_{q,c,s} \\ + \sum_{\substack{c \in C \\ (c,TMMF) \in CCL}} stockc_{q,c,s} - 0.05 \cdot stock_{q,s} \leq CAPQU_q \cdot (1 - tmbf2_{q,s}) \\ \forall q \in USQ, \ \forall s \in S \tag{3.125}$$

$$\sum_{\substack{c \in C \\ (c,TASF) \in CCL}} stockc_{q,c,s} + \sum_{\substack{c \in C \\ (c,TPES) \in CCL}} stockc_{q,c,s} + \sum_{\substack{c \in C \\ (c,TMMF) \in CCL}} stockc_{q,c,s} - 0.05 \cdot stock_{q,s} \geq CAPQU_q \cdot (-tmbf2_{q,s}) \quad \forall q \in USQ,\ \forall s \in S \tag{3.126}$$

$$tq_{TMBF,q,s} \leq tmbf1_{q,s} \quad \forall q \in USQ,\ \forall s \in S \tag{3.127}$$

$$tq_{TMBF,q,s} \leq tmbf2_{q,s} \quad \forall q \in USQ,\ \forall s \in S \tag{3.128}$$

$$tq_{TMBF,q,s} \geq tmbf1_{q,s} + tmbf2_{q,s} - 1 \quad \forall q \in USQ,\ \forall s \in S \tag{3.129}$$

Constraints (3.130)–(3.136) establish the classification criteria for class TBIA.

$$\sum_{\substack{c \in C \\ (c,TBIA) \in CCL}} stockc_{q,c,s} - 0.95 \cdot stock_{q,s} \geq CAPQU_q \cdot (tbia1_{q,s} - 1) \quad \forall q \in USQ,\ \forall s \in S \tag{3.130}$$

$$\sum_{\substack{c \in C \\ (c,TBIA) \in CCL}} stockc_{q,c,s} - 0.95 \cdot stock_{q,s} \leq CAPQU_q \cdot tbia1_{q,s} \quad \forall q \in USQ,\ \forall s \in S \tag{3.131}$$

$$\sum_{\substack{c \in C \\ (c,TASF) \in CCL}} stockc_{q,c,s} + \sum_{\substack{c \in C \\ (c,TPES) \in CCL}} stockc_{q,c,s} + \sum_{\substack{c \in C \\ (c,TMMF) \in CCL}} stockc_{q,c,s} - 0.05 \cdot stock_{q,s} \leq CAPQU_q \cdot (1 - tbia2_{q,s}) \quad \forall q \in USQ,\ \forall s \in S \tag{3.132}$$

$$\sum_{\substack{c \in C \\ (c,TASF) \in CCL}} stockc_{q,c,s} + \sum_{\substack{c \in C \\ (c,TPES) \in CCL}} stockc_{q,c,s} + \sum_{\substack{c \in C \\ (c,TMMF) \in CCL}} stockc_{q,c,s} - 0.05 \cdot stock_{q,s} \geq CAPQU_q \cdot (-tbia2_{q,s}) \quad \forall q \in USQ,\ \forall s \in S \tag{3.133}$$

$$tq_{TBIA,q,s} \leq tbia1_{q,s} \quad \forall q \in USQ,\ \forall s \in S \tag{3.134}$$

$$tq_{TBIA,q,s} \leq tbia2_{q,s} \quad \forall q \in USQ,\ \forall s \in S \tag{3.135}$$

$$tq_{TBIA,q,s} \geq tbia1_{q,s} + tbia2_{q,s} - 1 \quad \forall q \in USQ,\ \forall s \in S \tag{3.136}$$

Constraints (3.137)–(3.143) establish the classification criteria for class TLGR.

$$\sum_{\substack{c \in C \\ (c,TLGR) \in CCL}} stockc_{q,c,s} - 0.65 \cdot stock_{q,s} \geq CAPQU_q \cdot (tlgr1_{q,s} - 1) \quad \forall q \in USQ,\ \forall s \in S \tag{3.137}$$

$$\sum_{\substack{c \in C \\ (c,TLGR) \in CCL}} stockc_{q,c,s} - 0.65 \cdot stock_{q,s} \leq CAPQU_q \cdot tlgr1_{q,s} \quad \forall q \in USQ,\ \forall s \in S \tag{3.138}$$

$$\sum_{\substack{c \in C \\ (c,TASF) \in CCL}} stockc_{q,c,s} + \sum_{\substack{c \in C \\ (c,TPES) \in CCL}} stockc_{q,c,s} + \sum_{\substack{c \in C \\ (c,TMMF) \in CCL}} stockc_{q,c,s} - 0.05 \cdot stock_{q,s} \leq CAPQU_q \cdot (1 - tlgr2_{q,s}) \quad \forall q \in USQ,\ \forall s \in S \tag{3.139}$$

$$\sum_{\substack{c \in C \\ (c,TASF) \in CCL}} stockc_{q,c,s} + \sum_{\substack{c \in C \\ (c,TPES) \in CCL}} stockc_{q,c,s} + \sum_{\substack{c \in C \\ (c,TMMF) \in CCL}} stockc_{q,c,s} - 0.05 \cdot stock_{q,s} \geq CAPQU_q \cdot (-tlgr2_{q,s}) \quad \forall q \in USQ,\ \forall s \in S \tag{3.140}$$

$$tq_{TLGR,q,s} \leq tlgr1_{q,s} \quad \forall q \in USQ,\ \forall s \in S \tag{3.141}$$

$$tq_{TLGR,q,s} \leq tlgr2_{q,s} \quad \forall q \in USQ,\ \forall s \in S \tag{3.142}$$

$$tq_{TLGR,q,s} \geq tlgr1_{q,s} + tlgr2_{q,s} - 1 \quad \forall q \in USQ, \ \forall s \in S \tag{3.143}$$

Constraints (3.144)–(3.145) define the criteria for class TM10.

$$\sum_{\substack{c \in C \\ (c,TM10) \in CCL}} stockc_{q,c,s} - 0.9 \cdot stock_{q,s} \geq CAPQU_q \cdot (tq_{TM10,q,s} - 1) \quad \forall q \in USQ, \ \forall s \in S \tag{3.144}$$

$$\sum_{\substack{c \in C \\ (c,TM10) \in CCL}} stockc_{q,c,s} - 0.9 \cdot stock_{q,s} \leq CAPQU_q \cdot tq_{TM10,q,s} \quad \forall q \in USQ, \ \forall s \in S \tag{3.145}$$

Finally, constraint (3.146) states that tank q can belong to only one class cl at the beginning of each slot s.

$$\sum_{cl \in CL} tq_{cl,q,s} = 1 \quad \forall q \in USQ, \ \forall s \in S \tag{3.146}$$

If a refinery tank is being discharged, then the crude oil concentration in the outflow must be equal to the concentration inside the tank. In other words, this principle states that the proportion of each crude in the volume transferred ($vcqu_{c,qr,u,s}/vqu_{qr,u,s}$) must be the same as the proportion of each crude in the volume stored ($stockc_{qr,c,s}/stock_{qr,s}$). This rule is satisfied by (3.147). It should be noted that this equation yields two bilinear terms which are non-convex.

$$vcqu_{c,qr,u,s} \cdot stock_{qr,s} = vqu_{qr,u,s} \cdot stockc_{qr,c,s} \quad \forall c \in C, \ \forall qr \in QR, \ \forall u \in U, \ \forall s \in S \tag{3.147}$$

The same principle must be satisfied when loading tanks are discharged. The proportion of each crude oil in the volume transferred from a loading tank to a refinery tank must be equal to its proportion in the total volume stored in the loading tank (3.148).

$$vcqq_{c,ql,qr,s} \cdot stock_{ql,s} = vqq_{ql,qr,s} \cdot stockc_{ql,c,s} \quad \forall c \in C, \ \forall ql \in QL, \ \forall qr \in QR, \ \forall s \in S \tag{3.148}$$

Modeling downstream units

From this point on, the constraints related to the units downstream of the CDUs are presented. Figure 3.6 shows a schematic of these units, highlighting the most relevant variables defined when modeling this part of the process. The picture also includes the names of the tanks and the types of residues each tank admits.

To keep the figure clear and easy to interpret, only the variables in the area of intermediate unit V3 are shown. However, the same structure applies to unit VB3. Additionally, subscripts in the variables have been omitted for simplicity.

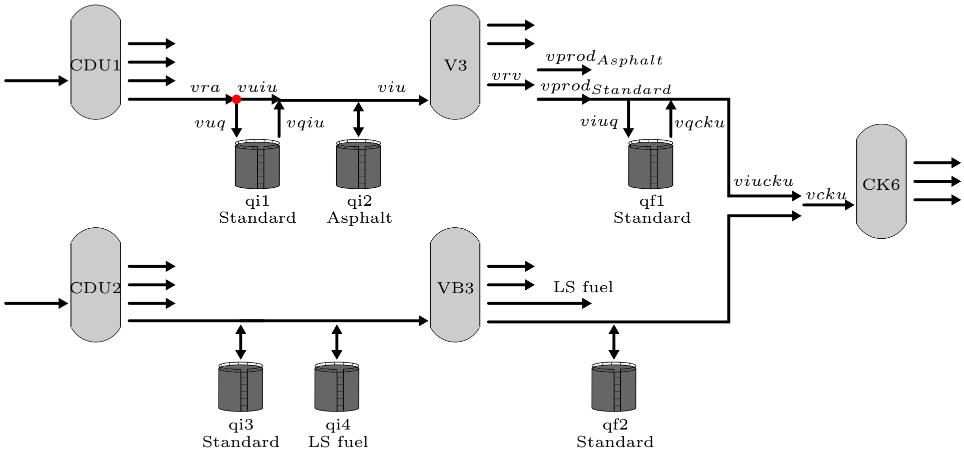

Figure 3.6: Schematic of downstream units and main variables.

CDU residue, intermediate tank operation, and intermediate unit feeding

The volume of atmospheric residue ($vra_{u,s}$) is calculated by (3.149) based on the input volume to the unit, which corresponds to the sum of the volumes of crude oil discharged from the refinery tanks. To determine this, the mass of atmospheric residue is first obtained by multiplying $vcqu_{c,qr,u,s}$ (volume of crude c transferred from qr to u), $PRC_{c,SPG}$ (specific gravity of crude c), and WD (water density), along with $PRC_{c,RA}$ (atmospheric residue yield of crude oil c). Then, this mass of RA is converted to volume by dividing by $PRC_{c,SPGRA}$ (specific gravity of RA for crude c) and WD. Note that both WD values cancel out. Finally, we must sum over the set of crude oil and refinery tanks.

$$vra_{u,s} = \sum_{c \in C} \sum_{qr \in QR} vcqu_{c,qr,u,s} \cdot \left(\frac{PRC_{c,SPG}}{PRC_{c,SPGRA}} \right) \cdot PRC_{c,RA} \qquad (3.149)$$
$$\forall u \in U, \ \forall s \in S$$

Similar to the previous constraint, the volume of atmospheric residue per crude is obtained using constraint (3.150).

$$vcra_{c,u,s} = \sum_{qr \in QR} vcqu_{c,qr,u,s} \cdot \left(\frac{PRC_{c,SPG}}{PRC_{c,SPGRA}} \right) \cdot PRC_{c,RA} \quad \forall c \in C,\ \forall u \in U,\ \forall s \in S \tag{3.150}$$

We should remember that each CDU is aligned with an intermediate unit (set UIU), CDU1 with V3, and CDU2 with VB3. In turn, each CDU can feed a set of intermediate tanks (set QIU).

The following constraint states that the volume of atmospheric residue is equal to the sum of the volume of residue sent from unit u to the intermediate tanks (connected to this unit) and the volume sent to the corresponding intermediate unit iu.

$$vra_{u,s} = \sum_{\substack{qi \in QI \\ (qi,u) \in QIU}} vuq_{u,qi,s} + vuiu_{u,iu,s} \quad \forall (u, iu) \in UIU,\ \forall s \in S \tag{3.151}$$

Similarly, equation (3.152) establishes the volume of atmospheric residue per crude oil.

$$vcra_{c,u,s} = \sum_{\substack{qi \in QI \\ (qi,u) \in QIU}} vcuq_{c,u,qi,s} + vcuiu_{c,u,iu,s} \quad \forall c \in C,\ \forall (u, iu) \in UIU,\ \forall s \in S \tag{3.152}$$

Constraint (3.153) states that the volume of residue loaded into the intermediate tank qi ($vuq_{u,qi,s}$) is equal to the sum of the input volumes of each crude oil c.

$$vuq_{u,qi,s} = \sum_{c \in C} vcuq_{c,u,qi,s} \quad \forall (qi, u) \in QIU,\ \forall s \in S \tag{3.153}$$

From constraint (3.154), the volume of residue transferred from crude distillation unit u to intermediate unit iu during slot s ($vuiu_{u,iu,s}$) is calculated as the sum of the volumes of each crude oil.

$$vuiu_{u,iu,s} = \sum_{c \in C} vcuiu_{c,u,iu,s} \quad \forall (u, iu) \in UIU,\ \forall s \in S \tag{3.154}$$

As shown in Figure 3.6, there is a node where the atmospheric residue flow is divided in two, one in the direction of the intermediate tanks and the other in the direction of the intermediate unit.

The non-linear constraint (3.155) states that the concentration of each crude oil present in the volume sent to an intermediate tank must be equal to its concentration in the volume sent to the intermediate unit iu.

$$vcuq_{c,u,qi,s} \cdot vuiu_{u,iu,s} = vcuiu_{c,u,iu,s} \cdot vuq_{u,qi,s} \quad \forall c \in C,\ \forall (u, iu) \in UIU,\ \forall (qi, u) \in QIU,\ \forall s \in S \tag{3.155}$$

From constraint (3.156), the input volume to intermediate unit iu during slot s ($viu_{iu,s}$) is calculated as the sum of the volume coming from CDU u and the volumes coming from the intermediate tanks.

$$viu_{iu,s} = vuiu_{u,iu,s} + \sum_{\substack{qi \in QI \\ (qi,u) \in QIU}} vqiu_{qi,iu,s} \quad \forall (u, iu) \in UIU, \ \forall s \in S \tag{3.156}$$

In a similar way, the input volume to unit iu is calculated for each crude oil using constraint (3.157).

$$\begin{aligned} &vciu_{c,iu,s} = vcuiu_{c,u,iu,s} + \sum_{\substack{qi \in QI \\ (qi,u) \in QIU}} vcqiu_{c,qi,iu,s} \\ &\forall c \in C, \ \forall (u, iu) \in UIU, \ \forall s \in S \end{aligned} \tag{3.157}$$

As stated in previous constraints related to tanks, the volume discharged from the intermediate tank qi is equal to the sum of the volumes discharged of each crude oil.

$$vqiu_{qi,iu,s} = \sum_{c \in C} vcqiu_{c,qi,iu,s} \quad \forall (u, iu) \in UIU, \ \forall (qi, u) \in QIU, \ \forall s \in S \tag{3.158}$$

Analogous to (3.147) and (3.148), constraint (3.159) is a non-linear constraint that guarantees that the concentration of crude oil c in the outlet volume from the tank qi is equal to the concentration inside the tank.

$$\begin{aligned} &vcqiu_{c,qi,iu,s} \cdot stock_{qi,s} = vqiu_{qi,iu,s} \cdot stockc_{qi,c,s} \\ &\forall c \in C, \ \forall (u, iu) \in UIU, \ \forall (qi, u) \in QIU, \ \forall s \in S \end{aligned} \tag{3.159}$$

Constraints (3.160)–(3.162) calculate the volume transferred to tank qi from CDU u.

Constraint (3.160) establishes the upper bound of the quantity transferred from unit u to intermediate tank qi ($vuq_{u,qi,s}$), based on the maximum tank load flow rate ($LFRQU_{qi}$) and the duration of the operation.

$$vuq_{u,qi,s} \leq LFRQU_{qi} \cdot ds_s \quad \forall (qi, u) \in QIU, \ \forall s \in S \tag{3.160}$$

Constraint (3.161) sets the lower limit of the volume loaded into tank qi and is activated only if the tank is being loaded ($lq_{qi,s} = 1$); otherwise, the constraint remains idle.

$$vuq_{u,qi,s} \geq LFRQL_{qi} \cdot ds_s - M2_{qi} \cdot (1 - lq_{qi,s}) \quad \forall (qi, u) \in QIU, \ \forall s \in S \tag{3.161}$$

Constraint (3.162) states that if the tank is being loaded, the inlet volume cannot exceed the tank's maximum storage capacity. Conversely, if the tank is not being loaded, the inlet volume must be zero.

$$vuq_{u,qi,s} \leq CAPQU_{qi} \cdot lq_{qi,s} \quad \forall (qi, u) \in QIU, \ \forall s \in S \tag{3.162}$$

Constraints (3.163)-(3.165) compute the output volume from intermediate tank qi to intermediate unit iu as a function of the tank output flow limits and the duration of operation.

$$vqiu_{qi,iu,s} \leq UFRQU_{qi} \cdot ds_s \quad \forall (u, iu) \in UIU,\ \forall (qi, u) \in QIU,\ \forall s \in S \tag{3.163}$$

$$\begin{aligned} & vqiu_{qi,iu,s} \geq UFRQL_{qi} \cdot ds_s - M2_{qi} \cdot (1 - uq_{qi,s}) \\ & \forall (u, iu) \in UIU,\ \forall (qi, u) \in QIU,\ \forall s \in S \end{aligned} \tag{3.164}$$

$$vqiu_{qi,iu,s} \leq CAPQU_{qi} \cdot uq_{qi,s} \quad \forall (u, iu) \in UIU,\ \forall (qi, u) \in QIU,\ \forall s \in S \tag{3.165}$$

The input volume to unit iu is calculated from (3.166) and (3.167), taking into account the limits of its flow rate and slot duration. Note that these units, like CDUs, must operate without interruption.

$$viu_{iu,s} \leq FRIUU_{iu} \cdot ds_s \quad \forall iu \in IU,\ \forall s \in S \tag{3.166}$$

$$viu_{iu,s} \geq FRIUL_{iu} \cdot ds_s \quad \forall iu \in IU,\ \forall s \in S \tag{3.167}$$

Before presenting the following constraint, we must remember that each intermediate tank accepts the residue of a specific recipe, except for tank $qi2$, which can receive residue from both mixtures associated with the asphalt process.

Constraint (3.168) states that intermediate tank qi can only be loaded or unloaded during slot s if, during that slot, the mixture admitted in that tank is being processed. In the case of tank $qi2$, this applies if either of the asphalt mixtures is being processed.

Additionally, it should be noted that the summation can only take a value of 0 or 1, ensuring that a tank cannot be loaded and unloaded simultaneously.

$$lq_{qi,s} + uq_{qi,s} \leq \sum_{\substack{mx \in MX \\ (qi,mx) \in QIM}} xmu_{mx,u,s} \quad \forall (qi, u) \in QIU,\ \forall s \in S \tag{3.168}$$

Constraints (3.169) and (3.170) state that the inventory level, total and by crude respectively, in the intermediate tank qi at the beginning of slot s is equal to the inventory level at the beginning of the previous slot, plus the volume of atmospheric residue received from CDU u during the previous slot, minus the volume unloaded to unit iu during the previous slot.

$$\begin{aligned} & stock_{qi,s} = stock_{qi,s-1} + vuq_{u,qi,s-1} - vqiu_{qi,iu,s-1} \\ & \forall (u, iu) \in UIU,\ \forall (qi, u) \in QIU,\ \forall s \in S \mid s > 1 \end{aligned} \tag{3.169}$$

$$stockc_{qi,c,s} = stockc_{qi,c,s-1} + vcuq_{c,u,qi,s-1} - vcqiu_{c,qi,iu,s-1}$$
$$\forall (u, iu) \in UIU,\ \forall c \in C,\ \forall (qi, u) \in QIU,\ \forall s \in S \mid s > 1 \tag{3.170}$$

From constraints (3.171) and (3.172), the stock level at the end of the horizon, total and for each crude oil, is calculated as the stock level at the start of the last slot of the horizon, plus the amount of residue received from crude distillation unit u during the last slot, minus the amount discharged to intermediate unit iu during the last slot.

$$stockend_{qi,s} = stock_{qi,s} + vuq_{u,qi,s} - vqiu_{qi,iu,s}$$
$$\forall (u, iu) \in UIU,\ \forall (qi, u) \in QIU,\ s = |S| \tag{3.171}$$

$$stockcend_{qi,c,s} = stockc_{qi,c,s} + vcuq_{c,u,qi,s} - vcqiu_{c,qi,iu,s}$$
$$\forall (u, iu) \in UIU,\ \forall c \in C,\ \forall (qi, u) \in QIU,\ s = |S| \tag{3.172}$$

Similar to the calculation of the volume of atmospheric residue in constraint (3.149), the volume of residue produced by intermediate unit iu during slot s is calculated by constraint (3.173).

$$vrv_{iu,s} = \sum_{c \in C} vciu_{c,iu,s} \cdot \left(\frac{PRC_{c,SPG}}{PRC_{c,SPGRV}} \right) \cdot PRC_{c,\%RV} \quad \forall iu \in IU,\ \forall s \in S \tag{3.173}$$

Intermediate unit residue, final tank operation, and coker unit feeding

It is important to remember that each intermediate unit iu is aligned with a CDU u (set UIU), and each CDU u supports specific processes (set UP). Therefore, each unit iu is only associated with these specific processes.

Equation 3.174 states that the volume of residue obtained in unit iu during slot s is equal to the sum of the volumes allocated to the processes allowed in this unit. Since only one process can be carried out at a time, the sum on the right-hand side will contain only a single positive term.

$$vrv_{iu,s} = \sum_{\substack{p \in P \\ (u,p) \in UP}} vprod_{iu,p,s} \quad \forall (u, iu) \in UIU,\ \forall s \in S \tag{3.174}$$

Constraint (3.175) states that the variable $vprod_{iu,p,s}$ can take a value greater than zero only if, during slot s, a mixture mx associated with process p (set PM) is being processed. Since only one mixture can be processed at a time, the summation is at most equal to one. If no mixture associated with process p is processed, the variable $vprod_{iu,p,s}$ is equal to zero. The value of M4 is defined as a function of the maximum input flow rate to the intermediate unit iu and the duration of the horizon.

$$vprod_{iu,p,s} \leq M4_{iu} \cdot \sum_{\substack{mx \in MX \\ (p,mx) \in PM \\ (u,mx) \in UM}} xmu_{mx,u,s} \quad \forall (u,iu) \in UIU,\ \forall (u,p) \in UP,\ \forall s \in S \tag{3.175}$$

The input volume to the coker unit from each intermediate unit iu during slot s is calculated by equation (3.176). It is equal to the volume of residue produced by iu allocated to the standard process, minus the input volume to the final tank qf associated with iu (set $QFIU$), plus the output volume from tank qf to the coker unit.

Note that at most one of the sums will be positive, as simultaneous loading and unloading of a tank is not allowed.

$$viucku_{iu,s} = vprod_{iu,p,s} - \sum_{\substack{qf \in QF \\ (qf,iu) \in QFIU}} viuq_{iu,qf,s} + \sum_{\substack{qf \in QF \\ (qf,iu) \in QFIU}} vqcku_{qf,s} \quad \forall iu \in IU,\ p = \text{Standard},\ \forall s \in S \tag{3.176}$$

Equations (3.177) and (3.178) calculate the inventory volumes in the final tanks at the beginning of each slot ($stock_{qf,s}$) and at the end of the horizon ($stockend_{qf,s}$), respectively.

It is important to note that, given the scope defined in this study, it is not necessary to track inventory by crude in the final tanks, as no properties or yields are calculated beyond this point.

$$stock_{qf,s} = stock_{qf,s-1} + viuq_{iu,qf,s-1} - vqcku_{qf,s-1} \quad \forall (qf,iu) \in QFIU,\ \forall s \in S \mid s > 1 \tag{3.177}$$

$$stockend_{qf,s} = stock_{qf,s} + viuq_{iu,qf,s} - vqcku_{qf,s} \quad \forall (qf,iu) \in QFIU,\ s = |S| \tag{3.178}$$

The inlet volume to the final tank qf from the intermediate unit iu during slot s ($viuq_{iu,qf,s}$) is calculated by constraints (3.179)-(3.181).

$$viuq_{iu,qf,s} \leq LFRQU_{qf} \cdot ds_s \quad \forall (qf,iu) \in QFIU,\ \forall s \in S \tag{3.179}$$

$$viuq_{iu,qf,s} \geq LFRQL_{qf} \cdot ds_s - M2_{qf} \cdot (1 - lq_{qf,s}) \quad \forall (qf,iu) \in QFIU,\ \forall s \in S \tag{3.180}$$

$$viuq_{iu,qf,s} \leq CAPQU_{qf} \cdot lq_{qf,s} \quad \forall (qf, iu) \in QFIU, \ \forall s \in S \tag{3.181}$$

Similarly, the outlet volume from tank qf to the coker unit ($vqcku_{qf,s}$) is calculated by constraints (3.182)-(3.184).

$$vqcku_{qf,s} \leq UFRQU_{qf} \cdot ds_s \quad \forall qf \in QF, \ \forall s \in S \tag{3.182}$$

$$vqcku_{qf,s} \geq UFRQL_{qf} \cdot ds_s - M2_{qf} \cdot (1 - uq_{qf,s}) \quad \forall qf \in QF, \ \forall s \in S \tag{3.183}$$

$$vqcku_{qf,s} \leq CAPQU_{qf} \cdot uq_{qf,s} \quad \forall qf \in QF, \ \forall s \in S \tag{3.184}$$

Constraint (3.185) states that a final tank can only be loaded during slot s if a standard process is running during this time. We should remember that final tanks only allow the storage of residues associated with the standard process.

$$\begin{aligned} lq_{qf,s} \leq & \sum_{\substack{mx \in MX \\ (p,mx) \in PM \\ (u,mx) \in UM}} xmu_{mx,u,s} \\ & \forall (u, iu) \in UIU, \ \forall (qf, iu) \in QFIU, \ p = \text{Standard}, \ \forall s \in S \end{aligned} \tag{3.185}$$

From equation (3.186), the total input volume to the coker unit during slot s is calculated, which is equal to the sum of the volumes supplied by each intermediate unit iu.

$$vcku_s = \sum_{iu \in IU} viucku_{iu,s} \quad \forall s \in S \tag{3.186}$$

From constraints (3.187) and (3.188), the total input volume to the coker unit during slot s is calculated ($vcku_s$), considering the upper and lower thresholds respectively.

$$vcku_s \leq FRCKU \cdot ds_s \quad \forall \forall s \in S \tag{3.187}$$

$$vcku_s \geq FRCKL \cdot ds_s \quad \forall \forall s \in S \tag{3.188}$$

3.5 Objective function

The objective function is given by (3.189). The first and second terms refer to demurrage and departure tardiness costs, respectively. The third and fourth terms represent the costs due to the difference between processed volume and required demand. The fifth and sixth terms involve the cost of violating the established limits for the concentration of key properties in feed mixtures. Finally, the last term maximizes the unloading of crude oil into priority tanks according to their grades.

$$
\begin{aligned}
\text{Minimize} \quad & \sum_{v \in V} CDMG_v \cdot dmg_v + \sum_{v \in V} CTDN_v \cdot tdn_v \\
+ & \sum_{iu \in IU} \sum_{\substack{\sum_{p \in P} \\ (u,p) \in UP \\ (u,iu) \in UIU}} CEP_{iu,p} \cdot exprod_{iu,p} \\
+ & \sum_{iu \in IU} \sum_{\substack{\sum_{p \in P} \\ (u,p) \in UP \\ (u,iu) \in UIU}} CSP_{iu,p} \cdot shprod_{iu,p} \\
+ & \sum_{k \in K} \sum_{u \in U} \sum_{s \in S} CSLKMAX_{k,u} \cdot slkmax_{k,u,s} \\
+ & \sum_{k \in K} \sum_{u \in U} \sum_{s \in S} CSLKMIN_{k,u} \cdot slkmin_{k,u,s} \\
- & \sum_{c \in C} \sum_{\substack{v \in V \\ (v,c) \in VC}} \sum_{q \in USQ} \sum_{s \in S} \sum_{cl \in CL} CCLP_{c,cl} \cdot vcvqcl_{c,v,q,s,cl}
\end{aligned}
\tag{3.189}
$$

The optimization model is a mixed-integer nonlinear programming (MINLP) model, as it involves both continuous and binary variables. The objective function is given by (3.189), and the problem is subject to constraints (3.1)-(3.188), among which (3.147), (3.148), (3.155), and (3.159) are nonlinear.

Chapter 4

Solution strategy

An important point to take into account when formulating a model is that there is a trade-off between model complexity and resolution time. Simple models allow us to obtain solutions in a short time, but they are hardly applicable in reality. On the other hand, complex models give detailed solutions but take a long time to achieve them, so we cannot respond within the time demanded by the process itself. Thus, the difficulty lies in formulating a model that faithfully represents the process and provides us with solutions applicable to the real operation but that, simultaneously, can be solved in a time frame that meets the user's needs.

Initially, the MINLP model presented in Chapter 3 performs properly for short scheduling horizons. However, as discussed in Chapter 2, the refinery operates based on a monthly scheduling horizon. Due to the complexity of the model, as well as its nonlinear and nonconvex nature, solving it in a monolithic manner is not viable.

To address this challenge, we have developed a temporal decomposition technique in conjunction with a linear approximation (i.e., yielding a MILP model). This approach allows us to determine the scheduling for a monthly horizon within a time frame that meets user requirements.

4.1 Piecewise linear approximation using planes

In this chapter, we develop a new method based on piecewise linear approximation using planes to approximate the product of two non-negative continuous variables.

The motivation for this development stems from the problem of blending crude oils in tanks, which is present in crude oil operations scheduling.

Typically, in the mathematical modeling of the crude oil operations scheduling problem, a constraint must be included to ensure that the concentration of crude oils in the output volumes of the tanks remains equal to the concentration within the tank. This constraint is nonlinear and non-convex because it includes two bilinear terms, each representing the product of two continuous variables related to inventory level and the volume transferred from the tanks.

For the sake of clarity and to avoid complicated nomenclature, the method is presented through a toy example. Let us assume that we want to approximate the product of two non-negative continuous variables $x1$ and $x2$.

First, the domain of each variable is partitioned into a certain number of intervals. Additionally, each quadrilateral formed by the intersection of intervals is divided into two triangles. For this, the diagonal is drawn from the lower left vertex to the upper right vertex in each quadrilateral. Then, from the vertices of each triangle, a plane is defined that will approximate the product of these variables in each interval. It should be noted that depending on the location of the point to be evaluated, one plane or another will be used. If the point is located above the diagonal, the equation formed by the vertices at the lower left, upper left, and upper right is used. If the point is below the diagonal, the equation formed by the vertices at the lower left, lower right, and upper right is used (Figure 4.1).

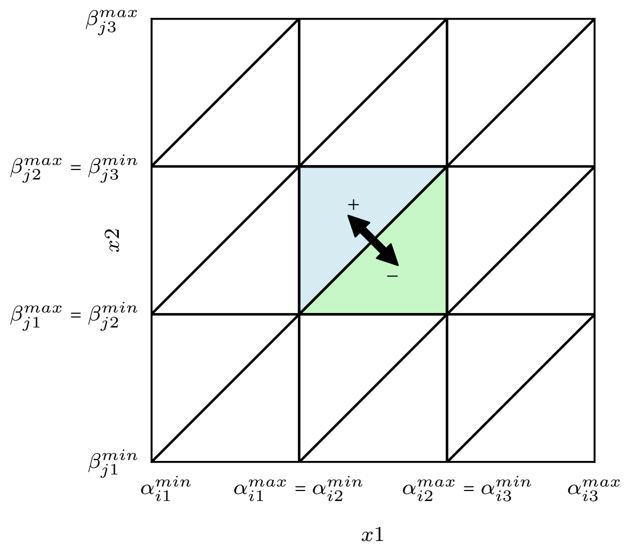

Figure 4.1: Piecewise approximation by planes.

Based on the above premise, the following set of variables and constraints is defined.

Each variable is disaggregated into two variables: "upper" and "lower". The "upper" variable ($x1_{i,j}^{u}$ for $x1$, and $x2_{i,j}^{u}$ for $x2$) is related to the upper plane, and the "lower" variable ($x1_{i,j}^{l}$ for $x1$, and $x2_{i,j}^{l}$ for $x2$) to the lower plane. Constraints (4.1) and (4.2) represent the disaggregation of variables

$x1$ and $x2$, respectively. Set I is the set of intervals for $x1$, and set J is the set of intervals for $x2$.

$$x1 = \sum_{i \in I} \sum_{j \in J} x1^{u}_{i,j} + x1^{l}_{i,j} \tag{4.1}$$

$$x2 = \sum_{i \in I} \sum_{j \in J} x2^{u}_{i,j} + x2^{l}_{i,j} \tag{4.2}$$

Two binary variables ($w^{u}_{i,j}$ and $w^{l}_{i,j}$) are defined to determine to which quadrilateral the point to be evaluated belongs, and to which plane (upper or lower). For example, if $x1$ is in interval i, $x2$ in interval j, and the point $(x1, x2)$ is above the diagonal; then $w^{u}_{i,j}$ will be equal to 1. The constraints (4.3)-(4.6) establish the upper and lower bounds of each disaggregated variable in each quadrilateral. The parameters α^{min}_{i} and α^{max}_{i} represent the minimum and maximum values, respectively, in interval i for the variables $x1^{u}_{i,j}$ and $x1^{l}_{i,j}$. Similarly, the parameters β^{min}_{j} and β^{max}_{j} represent the minimum and maximum values, respectively, in interval j for the variables $x2^{u}_{i,j}$ and $x2^{l}_{i,j}$.

$$\alpha^{min}_{i} * w^{u}_{i,j} \leq x1^{u}_{i,j} \leq \alpha^{max}_{i} * w^{u}_{i,j} \quad \forall i \in I, j \in J \tag{4.3}$$

$$\alpha^{min}_{i} * w^{l}_{i,j} \leq x1^{l}_{i,j} \leq \alpha^{max}_{i} * w^{l}_{i,j} \quad \forall i \in I, j \in J \tag{4.4}$$

$$\beta^{min}_{j} * w^{u}_{i,j} \leq x2^{u}_{i,j} \leq \beta^{max}_{j} * w^{u}_{i,j} \quad \forall i \in I, j \in J \tag{4.5}$$

$$\beta^{min}_{j} * w^{l}_{i,j} \leq x2^{l}_{i,j} \leq \beta^{max}_{j} * w^{l}_{i,j} \quad \forall i \in I, j \in J \tag{4.6}$$

The variable z, representing the value of the product of $x1$ and $x2$, is also disaggregated (4.7).

$$z = \sum_{i \in I} \sum_{j \in J} z^{u}_{i,j} + z^{l}_{i,j} \tag{4.7}$$

The disaggregated variables are calculated by (4.8) and (4.9). Note that these constraints correspond to the equations of the upper and lower planes for each quadrilateral, respectively. Additionally, if a plane is not selected, the value of the disaggregated variable associated with it will be zero.

$$z^u_{i,j} = \beta^{max}_j * x1^u_{i,j} + \alpha^{min}_i * x2^u_{i,j} - \alpha^{min}_i * \beta^{max}_j * w^u_{i,j} \quad \forall i \in I, j \in J \tag{4.8}$$

$$z^l_{i,j} = \beta^{min}_j * x1^l_{i,j} + \alpha^{max}_i * x2^l_{i,j} - \alpha^{max}_i * \beta^{min}_j * w^l_{i,j} \quad \forall i \in I, j \in J \tag{4.9}$$

Constraint (4.10) states that only one plane can be selected.

$$\sum_{i \in I} \sum_{j \in J} w^u_{i,j} + w^l_{i,j} = 1 \tag{4.10}$$

Depending on whether the distance from the point to the diagonal is positive (4.11) or negative (4.12), the binary variable $w^u_{i,j}$ or the binary variable $w^l_{i,j}$ can be activated, respectively.

$$(\alpha^{max}_i - \alpha^{min}_i) * x2^u_{i,j} - (\beta^{max}_j - \beta^{min}_j) * x1^u_{i,j} \geq (\alpha^{min}_i * \beta^{max}_j - \alpha^{max}_i * \beta^{min}_j) * w^u_{i,j} \quad \forall i \in I, j \in J \tag{4.11}$$

$$(\alpha^{max}_i - \alpha^{min}_i) * x2^l_{i,j} - (\beta^{max}_j - \beta^{min}_j) * x1^l_{i,j} \leq (\alpha^{min}_i * \beta^{max}_j - \alpha^{max}_i * \beta^{min}_j) * w^l_{i,j} \quad \forall i \in I, j \in J \tag{4.12}$$

In the model developed in Chapter 3, four nonlinear and nonconvex constraints were introduced ((3.147), (3.148), (3.155), and (3.159)), involving the product of nonnegative real variables, resulting in a mixed-integer nonlinear programming model.

It should be noted that the method developed above can also be applied to the product of indexed variables, allowing it to be extended to the bilinear terms of these nonlinear constraints.

In Appendix B, the set of constraints that approximate the nonlinear constraint (3.148) related to the concentration of crude oils in the output volumes of the loading tanks is presented. The approximations for the remaining nonlinear constraints can be derived in a similar manner.

It is important to mention that other methods, such as McCormick relaxation, were also considered throughout the development of the thesis. However, we found that the approach presented in this chapter is more efficient than the McCormick approximation when applied to the crude oil operations scheduling optimization problem. In [39], a comparison was made between the piecewise linear approximation using planes and the piecewise McCormick approximation, applied to a benchmark case study from the literature [14]. The results showed that the former method yielded a better solution in terms of both profit (an increase of over 3%) and computational time (a reduction of approximately 40%).

4.2 Temporal decomposition method

We present below the method developed to solve the scheduling problem for a monthly horizon. While this method does not guarantee a global optimum, it allows us to obtain feasible and high-quality solutions within a reasonable time frame.

The method involves the iterative resolution of two models: an aggregate model and a detailed model. Regarding the aggregate model, it employs a continuous-time formulation based on general precedence, and involves the following decisions:

- Assignment of mixtures to units and their sequencing.
- Assignment of tanks to mixtures and their sequencing.
- Calculation of transferred volumes.
- Calculation of operation times.

The aggregate model is based on the premise of optimizing crude oil operations scheduling, assuming that only current resources are available (initial refinery conditions) and that no crude oil supplies will be received over the horizon. It focuses on decisions to be made on refinery tanks, those feeding the distillation units, and therefore does not take into account the loading tanks.

Although the model does not account for crude supplies (i.e. the arrival of supply vessels), the scheduling horizon is divided into a number of sub-horizons equal to the number of vessels expected within the month, respecting their arrival order.

The start and end dates of each sub-horizon are defined as follows. The first sub-horizon starts at the beginning of the global horizon (time zero). Subsequent sub-horizons start at the end date of the previous sub-horizon. The end date of each sub-horizon matches the expected departure date of the corresponding vessel, except for the last sub-horizon which ends at the end of the overall horizon.

Since no crude supplies are received and loading of intermediate and final tanks is not allowed, tank compositions and their respective classes remain unchanged, allowing the aggregate model to be formulated as a MILP.

4.2.1 Aggregate model formulation

Sets

- IU: intermediate units.
- K: key properties of crude oil.

- MCL: classes allowed in each recipe.
- MX: mixtures or recipes for processes.
- P: processes.
- PM: recipes allowed in each process.
- Q: all tanks in the refinery (except loading tanks).
- QCL: class associated to each tank.
- QF: final tanks.
- $QFIU$: mapping of final tanks to intermediate units. This set indicates which final tanks can be loaded by intermediate unit iu.
- QI: intermediate tanks.
- QIM: allocation of tanks to recipes. This set indicates which mixture may be stored in the intermediate tank qi.
- QIU: mapping of intermediate tanks to crude distillation units. This set indicates which intermediate tanks can be loaded by CDU u.
- QR: refinery tanks.
- T: sub-horizons.
- U: crude distillation units.
- UIU: alignment of CDUs with intermediate units. This set indicates which CDU u is connected to the intermediate unit iu.
- UM: recipes allowed in each crude distillation unit.
- UP: processes allowed in each CDU.

4.2.2 Parameters

- $CAPQL_q$: minimum inventory level limit in a tank.
- $CAPQU_q$: maximum capacity of a tank.
- $CEP_{iu,p}$: cost due to overproduction concerning the demand of process p in intermediate unit iu.
- $CSLKMAX_{k,u}$: cost related to violation of upper boundary of key property.

- $CSLKMIN_{k,u}$: cost related to violation of lower boundary of key property.
- $CSP_{iu,p}$: cost due to underproduction concerning the demand of process p in intermediate unit iu.
- ETA: end-time of asphalt campaign.
- $FRIUL_{iu}$: minimum inlet flow to intermediate unit.
- $FRIUU_{iu}$: maximum inlet flow to intermediate unit.
- $FRCKL$: minimum inlet flow to coker unit.
- $FRCKU$: maximum inlet flow to coker unit.
- $FRUL_u$: minimum inlet flow to CDU.
- $FRUU_u$: maximum inlet flow to CDU.
- $LFRQL_q$: minimum tank loading flow rate.
- $LFRQU_q$: maximum tank loading flow rate.
- NTM: maximum number of tanks allowed in mixtures.
- NQU: maximum number of refinery tanks from which a CDU can be fed simultaneously.
- $PRL_{u,k}$: lower bound of property k in unit u.
- $PRU_{u,k}$: upper bound of property k in unit u.
- SLI_q: initial inventory level in tank.
- STA: start-time of asphalt campaign.
- $TUNDUB$: maximum proportion of tanks with grade "undefined" allowed in recipes.
- $UFRQL_q$: minimum tank unloading flow rate.
- $UFRQU_q$: maximum tank unloading flow rate.
- WD: water density.
- H: end-time of the scheduling horizon.
- HF_t: end time of subhorizon.
- HI_t: start time of subhorizon.

- HT_t: sub-horizon length.
- $\text{PDEMUP}_{iu,p,t}$: demand of process p in intermediate unit iu during subhorizon t
- $\text{PRQ}_{q,k}$: property value in tank.
- BM1: big-M value based on the capacity of the tanks.

Continuous variables

The domain of continuous variables is the set of non-negative real numbers.

- $vmu^{agg}_{mx,u,t}$: volume of recipe mx processed by CDU u during t.
- $vq_{qr,mx,t}$: volume unloaded from refinery tank qr for recipe mx during t.
- $dm_{mx,u,t}$: duration of processing of mixture mx in unit u during t.
- $tf_{mx,t}$: end-time of processing mixture mx during t.
- $ti_{mx,t}$: start-time of processing mixture mx during t.
- $stockend^{agg}_{q,t}$: total level in tank q at the end of the subhorizon t.
- $auxq_{q,t}$: auxiliary variable for tank inventory.
- $shprod^{agg}_{iu,p,t}$: shortage volume in intermediate unit iu for process p during t.
- $exprod^{agg}_{iu,p,t}$: overproduction volume in intermediate unit iu for process p during t.
- $slkmax^{agg}_{k,mx,u,t}$: slack variable for property upper bounds.
- $slkmin^{agg}_{k,mx,u,t}$: slack variable for property lower bounds.
- $vra^{agg}_{mx,u,t}$: output volume of atmospheric residue for recipe mx in CDU u during t.
- $vqiu^{agg}_{mx,qi,iu,t}$: volume transferred from intermediate tank qi to intermediate unit iu.
- $viu^{agg}_{mx,iu,t}$: input volume to intermediate unit iu during subhorizon t.
- $vrv^{agg}_{mx,iu,t}$: output volume of residue from intermediate unit iu during t.
- $vqcku^{agg}_{mx,qf,t}$: volume transferred from final tank qf to coker unit during t.

- $viucku^{agg}_{mx,iu,t}$: input volume to the coker unit from line relative to intermediate unit iu, during subhorizon t.
- $vprod^{agg}_{iu,p,t}$: output volume from intermediate unit iu to satisfy demand of process p during t.

Binary variables

- $xmu^{agg}_{mx,u,t}$: is equal to 1 if recipe mx is selected to be processed in CDU u during t.
- $xmm_{mx,mx',t}$: is equal to 1 if recipe mx is processed before recipe mx' during t.
- $uq^{agg}_{q,mx,t}$: is equal to 1 if tank q is assigned to mixture mx during t.

Constraints

The constraints (4.13) and (4.14) determine the fulfillment of demand for the first sub-horizon and the subsequent ones, respectively. To achieve this, they take into account the output volume from the intermediate unit and the shortage or overproduction associated with each sub-horizon t.

$$\begin{aligned} & vprod^{agg}_{iu,p,t} + shprod^{agg}_{iu,p,t} - exprod^{agg}_{iu,p,t} = PDEMUP_{iu,p,t} \\ & \quad \forall (u,iu) \in UIU,\ \forall (u,p) \in UP,\ t = 1 \end{aligned} \tag{4.13}$$

$$\begin{aligned} & vprod^{agg}_{iu,p,t} + shprod^{agg}_{iu,p,t} - exprod^{agg}_{iu,p,t} \\ & = PDEMUP_{iu,p,t} + shprod^{agg}_{iu,p,t-1} - exprod^{agg}_{iu,p,t-1} \\ & \quad \forall (u,iu) \in UIU,\ \forall (u,p) \in UP,\ \forall t \in T \mid t > 1 \end{aligned} \tag{4.14}$$

Constraint (4.15) establishes that the CDUs must operate continuously, meaning that at least one recipe must be processed throughout the sub-horizon.

$$\sum_{\substack{mx \in MX \\ (u,mx) \in UM}} xmu^{agg}_{mx,u,t} \geq 1 \quad \forall u \in U,\ \forall t \in T \tag{4.15}$$

Constraints (4.16) and (4.17) calculate the volume of recipe mx processed in unit u during sub-horizon t, based on the processing time and the bounds on the inlet flow rate to u.

$$vmu^{agg}_{mx,u,t} \geq FRUL_u \cdot dm_{mx,u,t} \quad \forall (u,mx) \in UM,\ \forall t \in T \tag{4.16}$$

$$vmu^{agg}_{mx,u,t} \leq FRUU_u \cdot dm_{mx,u,t} \quad \forall (u,mx) \in UM,\ \forall t \in T \tag{4.17}$$

Constraint (4.18) states that if recipe mx is not assigned to unit u during t, then its processing time is zero.

$$dm_{mx,u,t} \leq HT_t \cdot xmu^{agg}_{mx,u,t} \quad \forall (u, mx) \in UM,\ \forall t \in T \tag{4.18}$$

Constraint (4.19) determines that the sum of the processing times must be equal to the duration of the sub-horizon.

$$\sum_{\substack{mx \in MX \\ (u,mx) \in UM}} dm_{mx,u,t} = HT_t \quad \forall u \in U,\ \forall t \in T \tag{4.19}$$

Constraint (4.20) establishes that the final processing date of mixture mx is equal to the start date plus its processing time.

$$tf_{mx,t} = ti_{mx,t} + dm_{mx,u,t} \quad \forall (u, mx) \in UM,\ \forall t \in T \tag{4.20}$$

Constraints (4.21) and (4.22) establish the precedence between recipes mx and mx'. If both mixtures are assigned to the same unit and mx precedes mx', then the processing start-time of mx' must be later than the processing end-time of mx (4.21). Conversely, if mx' precedes mx, then the processing start date of mx must be later than the final processing date of mx' (4.22).

$$\begin{aligned} &ti_{mx',t} \geq tf_{mx,t} - H \cdot (1 - xmm_{mx,mx',t}) \\ &- H \cdot (2 - xmu^{agg}_{mx,u,t} - xmu^{agg}_{mx',u,t}) \\ &\quad \forall (u, mx) \in UM,\ \ \forall (u, mx') \in UM,\ mx < mx',\ \forall t \in T \end{aligned} \tag{4.21}$$

$$\begin{aligned} &ti_{mx,t} \geq tf_{mx',t} - H \cdot xmm_{mx,mx',t} \\ &- H \cdot (2 - xmu^{agg}_{mx,u,t} - xmu^{agg}_{mx',u,t}) \\ &\quad \forall (u, mx) \in UM,\ \forall (u, mx') \in UM,\ mx < mx',\ \forall t \in T \end{aligned} \tag{4.22}$$

The processing start dates of mixtures in sub-horizon t must be later than the final processing dates of the previous sub-horizon. This constraint prevents the processing end date of the last mixture corresponding to $t - 1$ from exceeding the start date of the first mixture corresponding to t.

$$ti_{mx',t} \geq tf_{mx,t-1} \quad \forall mx \in MX,\ \forall mx' \in MX,\ \forall t \in T \mid t > 1 \tag{4.23}$$

Constraints (4.24) and (4.25) determine the start and end dates of the asphalt campaign, respectively.

$$\begin{aligned} &ti_{mx,t} \geq STA \cdot xmu^{agg}_{mx,u,t} \\ &\quad \forall (u, mx) \in UM,\ \forall t \in T,\ \forall (p, mx) \in PM,\ p = \text{"Asphalt"} \end{aligned} \tag{4.24}$$

$$\begin{aligned} &tf_{mx,t} \leq ETA + HF_t \cdot (1 - xmu^{agg}_{mx,u,t}) \\ &\quad \forall (u, mx) \in UM,\ \forall t \in T,\ \forall (p, mx) \in PM,\ p = \text{"Asphalt"} \end{aligned} \tag{4.25}$$

Constraint (4.26) calculates the volume of mixture mx processed in unit u during t as the sum of the volumes discharged from the tanks assigned to that mixture. It should be noted that summing over the set of classes CL is necessary to link the classes associated with each tank (set QCL) with the classes allowed in each recipe (set MCL), and finally, with the mixtures allowed in each CDU (set UM).

$$vmu^{agg}_{mx,u,t} = \sum_{\substack{cl \in CL \\ (mx,cl) \in MCL}} \sum_{\substack{qr \in QR \\ (qr,cl) \in QCL}} vq_{qr,mx,t} \quad \forall (u, mx) \in UM, \ \forall t \in T \quad (4.26)$$

Constraint (4.27) establishes the maximum percentage of volume discharged from tanks whose class is undefined, relative to the inlet volume to unit u.

$$\sum_{\substack{qr \in QR \\ (qr,cl) \in QCL}} vq_{qr,mx,t} \leq TUNDUB \cdot vmu^{agg}_{mx,u,t}$$
$$\forall (u, mx) \in UM, \ \forall (mx, cl) \in MCL, \ \forall t \in T, \ cl = \text{"TUND"} \quad (4.27)$$

Similar to the detailed model, constraints (4.28)-(4.32) compute the property values of the mixtures fed into the CDUs based on the established minimum or maximum bounds. The only difference from the model developed in Chapter 3 is that the property values in the tanks do not change since their compositions remain constant (no tank loading occurs), and thus they are considered parameters ($PRQ_{q,k}$).

$$\sum_{\substack{cl \in CL \\ (mx,cl) \in MCL}} \sum_{\substack{qr \in QR \\ (qr,cl) \in QCL}} vq_{qr,mx,t} \cdot PRQ_{qr,k} \cdot PRQ_{qr,SPG} \cdot WD$$
$$\geq PRL_{u,k} \cdot \sum_{\substack{cl \in CL \\ (mx,cl) \in MCL}} \sum_{\substack{qr \in QR \\ (qr,cl) \in QCL}} vq_{qr,mx,t} \cdot PRQ_{qr,SPG} \cdot WD - slkmin^{agg}_{k,mx,u,t}$$
$$\forall (u, mx) \in UM, \ \forall t \in T, \ k = \text{"RA"} \quad (4.28)$$

$$\sum_{\substack{cl \in CL \\ (mx,cl) \in MCL}} \sum_{\substack{qr \in QR \\ (qr,cl) \in QCL}} vq_{qr,mx,t} \cdot PRQ_{qr,k} \cdot PRQ_{qr,SPG} \cdot WD$$
$$\leq PRU_{u,k} \cdot \sum_{\substack{cl \in CL \\ (mx,cl) \in MCL}} \sum_{\substack{qr \in QR \\ (qr,cl) \in QCL}} vq_{qr,mx,t} \cdot PRQ_{qr,SPG} \cdot WD + slkmax^{agg}_{k,mx,u,t}$$
$$\forall (u, mx) \in UM, \ \forall t \in T, \ k = \text{"RA"} \quad (4.29)$$

$$\sum_{\substack{cl \in CL \\ (mx,cl) \in MCL}} \sum_{\substack{qr \in QR \\ (qr,cl) \in QCL}} vq_{qr,mx,t} \cdot PRQ_{qr,k} \cdot PRQ_{qr,SPG} \cdot WD$$
$$\leq PRU_{u,k} \cdot \sum_{\substack{cl \in CL \\ (mx,cl) \in MCL}} \sum_{\substack{qr \in QR \\ (qr,cl) \in QCL}} vq_{qr,mx,t} \cdot PRQ_{qr,SPG} \cdot WD + slkmax^{agg}_{k,mx,u,t}$$
$$\forall (u,mx) \in UM, \ \forall t \in T, \ k = \text{"TAN"} \tag{4.30}$$

$$\sum_{\substack{cl \in CL \\ (mx,cl) \in MCL}} \sum_{\substack{qr \in QR \\ (qr,cl) \in QCL}} vq_{qr,mx,t} \cdot PRQ_{qr,k} \cdot PRQ_{qr,SPG} \cdot WD \cdot PRQ_{qr,MDS}$$
$$\geq PRL_{u,k} \cdot \sum_{\substack{cl \in CL \\ (mx,cl) \in MCL}} \sum_{\substack{qr \in QR \\ (qr,cl) \in QCL}} vq_{qr,mx,t} \cdot PRQ_{qr,SPG} \cdot WD \cdot PRQ_{qr,MDS}$$
$$- slkmin^{agg}_{k,mx,u,t}$$
$$\forall (u,mx) \in UM, \ \forall t \in T, \ k = \text{"CTI"} \tag{4.31}$$

$$\sum_{\substack{cl \in CL \\ (mx,cl) \in MCL}} \sum_{\substack{qr \in QR \\ (qr,cl) \in QCL}} vq_{qr,mx,t} \cdot PRQ_{qr,k}$$
$$\leq PRU_{u,k} \cdot \sum_{\substack{cl \in CL \\ (mx,cl) \in MCL}} \sum_{\substack{qr \in QR \\ (qr,cl) \in QCL}} vq_{qr,mx,t} + slkmax^{agg}_{k,mx,u,t} \tag{4.32}$$
$$\forall (u,mx) \in UM, \ \forall t \in T, \ k = \text{"SPG"}$$

Constraints (4.33) and (4.34) compute the volume discharged from refinery tank qr, assigned to mixture mx, based on its minimum and maximum discharge bounds and the processing time of the mixture.

$$vq_{qr,mx,t} \geq LFRQU_{qr} \cdot dm_{mx,u,t} - BM1_{qr} \cdot (1 - uq^{agg}_{qr,mx,t})$$
$$\forall qr \in QR, \ \forall (qr,cl) \in QCL, \ \forall (mx,cl) \in MCL, \ \forall (u,mx) \in UM, \ \forall t \in T \tag{4.33}$$

$$vq_{qr,mx,t} \leq UFRQU_{qr} \cdot dm_{mx,u,t}$$
$$\forall qr \in QR, \ \forall (qr,cl) \in QCL, \ \forall (mx,cl) \in MCL, \ \forall (u,mx) \in UM, \ \forall t \in T \tag{4.34}$$

Constraint (4.35) establishes that the volume is zero if the tank has not been assigned to the mixture.

$$vq_{qr,mx,t} \leq CAPQU_{qr} \cdot uq^{agg}_{qr,mx,t}$$
$$\forall qr \in QR, \ \forall (qr,cl) \in QCL, \ \forall (mx,cl) \in MCL, \ \forall t \in T \tag{4.35}$$

Constraint (4.36) calculates the inventory level of refinery tank qr at the end of sub-horizon t. It considers the initial inventory of qr and subtracts the total volume discharged from qr during sub-horizon t and previous sub-horizons. One important point to note is that, since crude oil supply is not considered in this aggregated model, the system may run out of crude in the tanks, making it impossible to find a feasible solution. To avoid this infeasibility, the auxiliary variable $auxq_{qr,t}$ has been included in the inventory balance. This variable is penalized in the objective function.

$$stockend^{agg}_{qr,t} = SLI_{qr} - \sum_{\substack{cl \in CL \\ (qr,cl) \in QCL}} \sum_{\substack{mx \in MX \\ (mx,cl) \in MCL}} \sum_{\substack{t2 \in T \\ t2 \leq t}} vq_{qr,mx,t2} + \sum_{\substack{t2 \in T \\ t2 \leq t}} auxq_{qr,t2}$$
$$\forall qr \in QR, \;\; \forall t \in T \tag{4.36}$$

Constraints (4.37) and (4.39) establish the minimum inventory level and maximum capacity of each tank, respectively.

$$stockend^{agg}_{q,t} \geq CAPQL_q \quad \forall q \in Q, \;\; \forall t \in T \tag{4.37}$$

$$stockend^{agg}_{q,t} \leq CAPQU_q \quad \forall q \in Q, \;\; \forall t \in T \tag{4.38}$$

Constraint (4.39) establishes the maximum number of tanks that can be assigned to each mixture during sub-horizon t.

$$\sum_{q \in QR} uq^{agg}_{qr,mx,t} \leq NTM \quad \forall mx \in MX, \;\; \forall t \in T \tag{4.39}$$

Constraint (4.40) states that a tank cannot be assigned to a mixture if that mixture is not being processed.

$$uq^{agg}_{qr,mx,t} \leq xmu^{agg}_{mx,u,t}$$
$$\forall qr \in QR, \;\; \forall (qr,cl) \in QCL, \;\; \forall (mx,cl) \in MCL, \;\; \forall (u,mx) \in UM, \;\; \forall t \in T \tag{4.40}$$

From constraint (4.41), the volume of atmospheric residue obtained in CDU u when processing mixture mx during sub-horizon t is calculated.

$$vra^{agg}_{mx,u,t} = \sum_{\substack{cl \in CL \\ (mx,cl) \in MCL}} \sum_{\substack{qr \in QR \\ (qr,cl) \in QCL}} vq_{qr,mx,t} \cdot PRQ_{qr,RA} \cdot \left(\frac{PRQ_{qr,SPG}}{PRQ_{qr,SPGRA}} \right)$$
$$\forall (u,mx) \in UM, \;\; \forall t \in T \tag{4.41}$$

Constraint (4.42) prevents the intermediate tank qi dedicated to mixture mx from being discharged during t if that mixture is not being processed.

$$uq^{agg}_{qi,mx,t} \leq xmu^{agg}_{mx,u,t}$$
$$\forall (qi,mx) \in QIM, \;\; \forall (qi,u) \in QIU, \;\; \forall (u,mx) \in UM, \;\; \forall t \in T \tag{4.42}$$

From constraints (4.43)-(4.45), the volume discharged from the intermediate tank qi dedicated to mixture mx, which is processed in the intermediate unit iu during t, is calculated.

$$vqiu^{agg}_{mx,qi,iu,t} \leq UFRQU_{qi} \cdot dm_{mx,u,t}$$
$$\forall (qi,mx) \in QIM,\ \forall (qi,u) \in QIU,\ \forall (u,mx) \in UM,\ \forall (u,iu) \in UIU,\ \forall t \in T \tag{4.43}$$

$$vqiu^{agg}_{mx,qi,iu,t} \leq CAPQU_{qi} \cdot uq^{agg}_{qi,mx,t}$$
$$\forall (qi,mx) \in QIM,\ \forall (qi,u) \in QIU,\ \forall (u,mx) \in UM,\ \forall (u,iu) \in UIU,\ \forall t \in T \tag{4.44}$$

$$vqiu^{agg}_{mx,qi,iu,t} \geq LFRQU_{qi} \cdot dm_{mx,u,t} - BM1_{qi} \cdot (1 - uq^{agg}_{qi,mx,t})$$
$$\forall (qi,mx) \in QIM,\ \forall (qi,u) \in QIU,\ \forall (u,mx) \in UM,\ \forall (u,iu) \in UIU,\ \forall t \in T \tag{4.45}$$

Constraint (4.46) states that the inlet volume of mixture mx into the intermediate unit iu is equal to the sum of the volume of atmospheric residue obtained in CDU u and the volume discharged from the intermediate tank qi.

$$viu^{agg}_{mx,iu,t} = vra^{agg}_{mx,u,t} + \sum_{\substack{qi \in QI \\ (qi,mx) \in QIM}} vqiu^{agg}_{mx,qi,iu,t}$$
$$\forall (u,mx) \in UM,\ \forall (u,iu) \in UIU,\ \forall t \in T \tag{4.46}$$

Constraints (4.47) and (4.48) compute the inlet volume of mixture mx into the intermediate unit iu, based on the processing time of that mixture and the minimum and maximum feed bounds of the intermediate unit iu, respectively.

$$viu^{agg}_{mx,iu,t} \geq FRVUL_{iu} \cdot dm_{mx,u,t}$$
$$\forall (u,mx) \in UM,\ \forall (u,iu) \in UIU,\ \forall t \in T \tag{4.47}$$

$$viu^{agg}_{mx,iu,t} \leq FRVUU_{iu} \cdot dm_{mx,u,t}$$
$$\forall (u,mx) \in UM,\ \forall (u,iu) \in UIU,\ \forall t \in T \tag{4.48}$$

Analogous to refinery tanks, the inventory level of the intermediate tanks at the end of sub-horizon t is calculated using constraint (4.49).

$$stockend_{qi,t} = SLI_{qi} - \sum_{\substack{mx \in MX \\ (qi,mx) \in QIM}} \sum_{\substack{t2 \in T \\ t2 \leq t}} vqiu^{agg}_{mx,qi,iu,t2} + \sum_{\substack{t2 \in T \\ t2 \leq t}} auxq_{qi,t2}$$
$$\forall (qi,u) \in QIU,\ \forall (u,iu) \in UIU,\ \forall t \in T \tag{4.49}$$

From constraint (4.50), the volume of residue obtained in the intermediate unit iu when processing the volume of mixture mx coming from the corresponding CDU and intermediate tank during sub-horizon t is determined. It is important to highlight that the first term, which represents the volume contributed by the CDU, is expressed in terms of the participating refinery tanks, since the vacuum residue yield of these tanks is known. Thus, it is possible to calculate how much of the residue produced in the intermediate unit is due to the mixture supplied by the refinery tanks, which is analogous to determining how much residue is obtained in the intermediate unit from the stream coming from the CDU.

$$vrv^{agg}_{mx,iu,t} = \sum_{\substack{cl \in CL \\ (mx,cl) \in MCL}} \sum_{\substack{qr \in QR \\ (qr,cl) \in QCL}} vq_{qr,mx,t} \cdot PRQ_{qr,RV} \cdot \left(\frac{PRQ_{qr,SPG}}{PRQ_{qr,SPGRV}} \right)$$
$$+ \sum_{\substack{qi \in QI \\ (qi,mx) \in QIM}} vqiu^{agg}_{mx,qi,iu,t} \cdot PRQ_{qi,RV} \cdot \left(\frac{PRQ_{qi,SPG}}{PRQ_{qi,SPGRV}} \right)$$
$$\forall (u,mx) \in UM, \ \forall (u,iu) \in UIU, \ \forall t \in T \tag{4.50}$$

Constraint (4.51) calculates the volume produced in the intermediate unit iu intended for the process p during t, considering the volume of residue obtained in the intermediate unit from the mixture associated with each process.

$$vprod^{agg}_{iu,p,t} = \sum_{\substack{mx \in MX \\ (u,mx) \in UM \\ (p,mx) \in PM}} vrv^{agg}_{mx,iu,t} \qquad \forall (u,iu) \in UIU, \ \forall (u,p) \in UP, \ \forall t \in T \tag{4.51}$$

Constraint (4.52) establishes that while a mixture associated with the Standard process is being processed, the inlet volume to the coker unit from the intermediate unit iu is equal to the sum of the residue volume obtained in iu and the volume supplied by the corresponding final tank.

$$viucku^{agg}_{mx,iu,t} = vprod^{agg}_{iu,p,t} + \sum_{\substack{qf \in QF \\ (qf,iu) \in QFIU}} vqcku^{agg}_{mx,qf,t}$$
$$\forall (u,mx) \in UM, \ \forall (u,iu) \in UIU, \ \forall (p,mx) \in PM, \ \forall t \in T, \ p = \text{"Standard"} \tag{4.52}$$

On the other hand, if a mixture associated with Standard is not being processed, then the inlet volume to the coker unit will be equal to the volume supplied by the final tank. In this case, the variable $vqcku^{agg}_{mx,qf,t}$ is interpreted as the volume supplied by the final tank qf to the coker unit during the period in which mixture mx, which does not correspond to a Standard mixture, is being processed. It is important to consider both cases since the feed to the coker unit cannot be interrupted.

$$viucku^{agg}_{mx,iu,t} = \sum_{\substack{qf \in QF \\ (qf,iu) \in QFIU}} vqcku^{agg}_{mx,qf,t}$$
$$\forall (u,mx) \in UM,\ \forall (u,iu) \in UIU,\ \forall (p,mx) \in PM,\ \forall t \in T,\ p \neq \text{"Standard"} \tag{4.53}$$

From constraint (4.54), the inventory level of the final tank qf at the end of sub-horizon t is determined.

$$stockend^{agg}_{qf,t} = SLI_{qf} - \sum_{\substack{mx \in MX \\ (u,mx) \in UM}} \sum_{\substack{t2 \in T \\ t2 \leq t}} vqcku^{agg}_{mx,qf,t2} + \sum_{\substack{t2 \in T \\ t2 \leq t}} auxq_{qf,t2}$$
$$\forall (qf,iu) \in QFIU,\ \forall (u,iu) \in UIU,\ \forall t \in T \tag{4.54}$$

Constraints (4.55) and (4.56) compute the volume discharged from the final tank qf based on the processing time and the minimum and maximum discharge bounds, respectively.

$$vqcku^{agg}_{mx,qf,t} \geq LFRQU_{qf} \cdot dm_{mx,u,t} - BM1_{qf} \cdot (1 - uq^{agg}_{qf,mx,t})$$
$$\forall (qf,iu) \in QFIU,\ \forall (u,iu) \in UIU,\ \forall (u,mx) \in UM,\ \forall t \in T \tag{4.55}$$

$$vqcku^{agg}_{mx,qf,t} \leq UFRQU_{qf} \cdot dm_{mx,u,t}$$
$$\forall (qf,iu) \in QFIU,\ \forall (u,iu) \in UIU,\ \forall (u,mx) \in UM,\ \forall t \in T \tag{4.56}$$

If the final tank qf is not unloaded, then the supplied volume must be zero.

$$vqcku^{agg}_{mx,qf,t} \leq CAPQU_{qf} \cdot uq^{agg}_{qf,mx,t}$$
$$\forall (qf,iu) \in QFIU,\ \forall (u,iu) \in UIU,\ \forall (u,mx) \in UM,\ \forall t \in T \tag{4.57}$$

Constraint (4.58) states that the binary variable $uq^{agg}_{qf,mx,t}$ indicating whether a tank is being discharged during the processing of mixture mx in sub-horizon t can take a value of 1 if and only if mixture mx is processed in unit u during t ($xmu^{agg}_{mx,u,t}$).

$$uq^{agg}_{qf,mx,t} \leq xmu^{agg}_{mx,u,t}$$
$$\forall (qf,iu) \in QFIU,\ \forall (u,iu) \in UIU,\ \forall (u,mx) \in UM,\ \forall t \in T \tag{4.58}$$

Constraints (4.59) and (4.60) compute the inlet volume to the coker unit based on the minimum and maximum feed bounds and the processing time.

$$viucku^{agg}_{mx,iu,t} \geq FRCKL \cdot dm_{mx,u,t}$$
$$\forall (u,iu) \in UIU,\ \forall (u,mx) \in UM,\ \forall t \in T \tag{4.59}$$

$$viucku^{agg}_{mx,iu,t} \leq FRCKU \cdot dm_{mx,u,t}$$
$$\forall (u,iu) \in UIU,\ \forall (u,mx) \in UM,\ \forall t \in T \tag{4.60}$$

Objective Function

The objective function seeks to minimize:

- The shortage and overproduction of residue volume produced in the intermediate unit iu intended to meet the demand of process p throughout the scheduling horizon.
- The violation of the lower and upper limits of the properties evaluated in the CDU feed.
- The sum over all tanks and sub-horizons of the auxiliary variable $auxq_{q,t}$.

$$
\begin{aligned}
\text{Minimize} & \sum_{iu \in IU} \sum_{\substack{p \in P \\ (u,p) \in UP \\ (u,iu) \in UIU}} \sum_{t \in T} CSP_{iu,p} \cdot shprod^{agg}_{iu,p,t} \\
& + \sum_{iu \in IU} \sum_{\substack{p \in P \\ (u,p) \in UP \\ (u,iu) \in UIU}} \sum_{t \in T} CEP_{iu,p} \cdot exprod^{agg}_{iu,p,t} \\
& + \sum_{k \in K} \sum_{mx \in MX} \sum_{\substack{u \in U \\ (u,mx) \in UM}} \sum_{t \in T} CSLKMIN_{k,u} \cdot slkmin^{agg}_{k,mx,u,t} \\
& + \sum_{k \in K} \sum_{mx \in MX} \sum_{\substack{u \in U \\ (u,mx) \in UM}} \sum_{t \in T} CSLKMAX_{k,u} \cdot slkmax^{agg}_{k,mx,u,t} \\
& + \sum_{q \in Q} \sum_{t \in T} CAUXQ_q \cdot auxq_{q,t}
\end{aligned}
\tag{4.61}
$$

Interestingly, it should be noted that downstream operations have a direct impact on the objective function. As shown, this function involves the variables $shprod^{agg}_{iu,p,t}$ and $exprod^{agg}_{iu,p,t}$, which represent the shortage and overproduction, respectively, of the volume produced for each process relative to the established demand. This produced volume corresponds to the residue obtained in each intermediate unit, meaning that meeting the demand depends on the operation of these units.

It is also important to recall that the intermediate units are fed by the intermediate tanks and by the residue produced in the CDUs, which in turn are fed by the refinery tanks. Therefore, the operation of the intermediate units also affects decisions related to the discharge of refinery tanks and the fulfillment of the established quality specifications (i.e., key property values) for the feed blends to the CDUs.

Once the aggregate model has been presented, we continue with the explanation of the temporal decomposition method. As mentioned before, the method implies an iterative procedure.

In the first iteration, the aggregate model is solved for the entire horizon. Based on the solution for the first sub-horizon, specific binary variables are fixed in the detailed model. If refinery tank qr is not assigned to any mixture during the first sub-horizon, then tank qr must not be assigned to any CDU in the detailed model.

It should be noted that during the first iteration, the detailed model is solved only for the first sub-horizon.

Furthermore, before solving the detailed model, the following parameters must be set:

- Number of time slots.
- Start and end dates of the scheduling horizon, corresponding to the current sub-horizon.
- Demand for each process, for each unit, to be satisfied within the horizon.

Once these parameters are defined, the detailed model is solved. If the model is infeasible, the proposed number of slots is increased, and the model is solved again. This procedure is repeated until a solution is found or until a maximum number of attempts is reached.

If a feasible solution is obtained, the algorithm proceeds to the next iteration, in which the second sub-horizon is solved. At the start of the second iteration, the first sub-horizon is discarded, as it was optimized in the previous iteration. Consequently, the global horizon now extends from the beginning of the second sub-horizon onward.

Before solving the aggregated model, the initial conditions of the refinery are updated based on the solution obtained from the detailed model. This update considers the inventory levels and tank properties at the end of the first sub-horizon. Additionally, the remaining demand is recalculated.

Both models are then solved as described above. This procedure is repeated until all sub-horizons have been completed or, equivalently, until all vessels have been iterated over.

Chapter 5

Deterministic results

This chapter analyzes the performance of the detailed model developed in Chapter 3, in combination with the linear approximation and temporal decomposition strategies presented in Chapter 4.

To this end, two case studies based on real data from the refinery under study are solved. These case studies correspond to the refinery configuration described in Chapter 2, which is illustrated in Figure 2.1.

The resulting solutions are analyzed through various visualizations, including Gantt charts, evolution of property values in the feed blends to the CDUs, as well as solution times and model statistics.

5.1 Case study 1

The scheduling horizon spans one month, during which the arrival of nine vessels is expected. Each vessel is associated with an expected arrival and departure date, a specific crude type, and the corresponding mass. This information is detailed in Table 5.1.

Table 5.2 displays the demand requirements over the planning horizon, broken down by process and associated intermediate unit. The asphalt campaign is scheduled from days 18 to 21. It is important to remember that the demand fulfillment is based on the output volume of the intermediate units (variable $vprod_{iu,p,s}$).

The initial inventory of each tank, as well as the storage capacity limits, are reported in Table 5.3. For the final tanks, only the total initial volume is indicated, as there is no tracking of the individual crude types stored, only the aggregated quantity is considered. It is worth noting that refinery tanks correspond to tanks q1 to q11, loading tanks to q12, q13, and q14, intermediate tanks to qi1, qi2, qi3, and qi4, and final tanks to qf1 and qf2.

The crude oil characteristics are provided in two separate tables. Table 5.4 summarizes the values of the properties for each crude type: cetane index (CTI), total acid number (TAN), specific gravity (SPG), atmospheric residue

Table 5.1: Vessel arrival and departure dates, crude type, and mass.

Vessel	Arrival date	Departure date	Transported crude	Mass (x10^3 ton)
B1	01/06/2024	03/06/2024	CPC	94
B2	04/06/2024	06/06/2024	UBP	132
B3	07/06/2024	09/06/2024	MAY	90
B4	11/06/2024	13/06/2024	GRA	90
B5	14/06/2024	16/06/2024	COL	148
B6	17/06/2024	19/06/2024	CPC	94
B7	22/06/2024	24/06/2024	BHP	94
B8	25/06/2024	27/06/2024	GRA	95
B9	28/06/2024	30/06/2024	ISX	69

Table 5.2: Demand for the planning horizon.

Unit	Demand (x10^3 m^3)		
	Standard	Asphalt	LS Fuel
V3	37	3	-
VB3	28	-	2

specific gravity (SPGRA), atmospheric residue yield (RA), middle distillate yield (MDS), and vacuum residue yield (RV); while Table 5.5 displays the priority scale between crude types and grades.

It is worth noting that the data presented were provided by professionals from the planning team at the Petronor refinery.

The problem size is also reported, including the number of variables and constraints for each subhorizon after applying the linear approximation strategy presented in Chapter 4 to the detailed MINLP model introduced in Chapter 3. These statistics, along with the corresponding solution times, are shown in Table 5.6. It should be noted that the column labeled "Solution Time" refers specifically to the time required to solve the detailed MILP model for each subhorizon.

It is important to know that the original model reaches an approximate size of 700000 constraints, 165000 of which are nonlinear, along with 400000 continuous variables and 12000 binary variables, and no feasible solution was obtained when attempting to solve the monolithic model after four hours of computation.

In addition to the analysis presented earlier, the solution is examined using several graphs.

Table 5.3: Initial inventory and capacity limits of the tanks.

Tank	Initial inventory (x10^3 m^3)													Total	Capacity (x10^3 m^3)	
	BUZ	CPC	ESX	FOX	ISX	JOS	M1T	M1Y	MAY	MTL	SUC	UBP	WTM		Min	Max
q1				30		10								40	8	62
q2							20	10						30	8	63
q3				25		15								40	8	63
q4			10							10				20	8	63
q5								2	45					47	8	63,5
q6	5				5				5			5		20	6	63
q7			8							8				16	6	63,5
q8		10										30		40	5	63
q9		5					5					60	5	75	14	102
q10									24			30		54	13	103
q11	30													30	10	103
q12	40	7							10					57	8	63
q13	2				2				2			4		10	8	63
q14									5			15		20	7	63
qi1				2			40		3		3			48	2	100
qi2		20	5		5			10		15				55	2	100
qi3				5					25			10	10	50	2	100
qi4	15					15		20			30			80	2	100
qf1														48	2	60
qf2														48	2	60

Table 5.4: Crude oil properties.

Crude	CTI	TAN (mg KOH/g)	SPG	SPGRA	RA	MDS	RV
BHP	49,06	0,22	0,91	1,03	0,47	0,17	0,28
BUZ	52,63	0,19	0,88	0,96	0,48	0,18	0,25
COL	37,97	1,27	0,93	1,03	0,58	0,14	0,36
CPC	59,12	0,07	0,79	0,92	0,15	0,19	0,05
ESX	60,79	0,13	0,84	0,94	0,32	0,21	0,14
FOX	58,41	0,11	0,83	0,95	0,26	0,19	0,10
GRA	46,97	1,06	0,89	0,97	0,41	0,23	0,19
ISX	59,90	0,21	0,85	0,99	0,32	0,21	0,18
JOS	47,36	0,36	0,88	0,97	0,43	0,21	0,20
M1T	57,86	0,18	0,95	0,98	0,79	0,13	0,50
M1Y	39,95	0,18	0,98	1,02	0,75	0,17	0,50
MAY	49,15	0,30	0,93	1,03	0,53	0,16	0,35
MTL	41,88	2,95	0,90	0,95	0,38	0,29	0,14
SUC	53,32	0,15	0,87	0,98	0,39	0,24	0,17
UBP	53,83	0,09	0,87	0,97	0,40	0,20	0,18
WTM	63,35	0,10	0,81	0,91	0,25	0,19	0,09

Table 5.5: Crude type and grade priority scale.

Crude	TASF	TPES	TMMF	TMBF	TBIA	TLGR	TM10	TUND
BHP	7	8	0	0	0	0	0	0
BUZ	0	0	6	7	8	0	0	0
COL	8	7	0	0	0	0	0	0
CPC	0	0	0	7	0	8	0	0
ESX	0	0	0	6	8	7	0	0
FOX	0	0	0	7	0	8	0	0
GRA	0	0	6	8	0	7	0	0
ISX	0	0	0	8	0	7	0	0
JOS	0	0	6	7	0	8	0	0
M1T	0	7	0	6	0	0	8	0
M1Y	0	7	0	6	0	0	8	0
MAY	8	7	0	0	0	0	0	0
MTL	0	0	6	7	8	0	0	0
SUC	0	0	0	0	0	0	0	8
UBP	0	0	7	8	0	6	0	0
WTM	0	0	0	7	0	8	0	0

Table 5.6: Model statistics.

Iteration	Constraints	Binary variables	Continuous variables	Solution time (s)
0	900317	167376	523130	80
1	900317	167376	523130	99
2	1200444	223168	697500	632
3	900317	167376	523130	67
4	900317	167376	523130	305
5	1200444	223168	697500	507
6	900317	167376	523130	111
7	900317	167376	523130	49
8	1200444	223168	697500	255

5.1.1 Resource operations

First, Figure 5.1 shows a Gantt chart that represents the operation of the different resources over the horizon. As can be seen, the Y-axis shows the resources, namely the vessels, the processing units, and the tanks; then, the X-axis shows the time, with the displayed dates corresponding to the moments when events occur. Finally, the operation of each resource and its duration are represented using bars. These bars are associated with a color depending on the resource and the operation carried out.

Starting from the bottom of the Y-axis, the vessels are shown first; the associated bars are light blue and represent their unloading. For the processing units, the bars represent the processing of the feedstock blends, and their colors vary according to the process carried out: gray for "Standard", green for "LS Fuel", and black for "Asphalt". Regarding the tanks, for the loading tanks and refinery tanks, the blue bars represent loading from vessels; in the case of refinery tanks, red bars are also shown, which represent unloading to the CDUs. As for the intermediate tanks, the purple bars represent unloading to intermediate processing units. Finally, in the final tanks, the brown bars represent unloading to the coker unit.

When analyzing the unloading of the ships, we can note that vessel 1 unloads into tanks 1, 2, 3, 8, and 9; vessel 2 supplies crude to tanks 6, 11, 13, and 14; vessel 3 to tanks 4, 5, and 11; vessel 4 feeds tanks 1, 7, and 11; vessel 5 supplies tanks 4, 6, and 10; vessel 6 discharges into tanks 6, 7, and 8; vessel 7 into tanks 3, 4, and 5; vessel 8 feeds tanks 1, 7, and 11; finally, vessel 9 supplies crude to tanks 4 and 11.

Moreover, we find that no vessel incurs demurrage or tardiness costs, except for the last one (vessel 9), which incurs a demurrage of almost 34 hours and a tardiness of 10 hours.

Regarding the processing units, it is evident that all of them comply with the constraint of operating without interruptions, as the diagram shows bars throughout the horizon. It is worth noting that most of the time is dedicated to processing Standard blends, which is due to this being the process with the highest demand. On the other hand, we can observe that four LS fuel campaigns are carried out, represented by green bars, both in CDU 2 and in the intermediate unit VB3. Additionally, the allowed time window for executing the Asphalt campaign is respected, represented by black bars in units CDU 1 and V3.

Other conclusions that arise from the analysis of Figure 5.1 are the following. We can see that loading tank 12 and intermediate tanks 1, 2, and 4 do not operate at any point, meaning they are neither loaded nor unloaded. From this last observation, it can be inferred that the solution is efficient, since it allows operations to be carried out using fewer tanks than those available. This opens the possibility of increasing the established production levels, provided that the refinery's processing capacity allows it. Additionally, from a logistical point of view, the underutilization of the tank infrastructure could facilitate the reception of larger volumes of crude, which would further support potential increases in production, subject to refinery constraints.

Furthermore, there are no transfers from loading tanks to refinery tanks, nor transfers to intermediate or final tanks. Among the intermediate tanks, only tank 3 operates, feeding the intermediate unit VB3 toward the end of the horizon; recall that this tank was associated with the Standard process, as confirmed by the diagram.

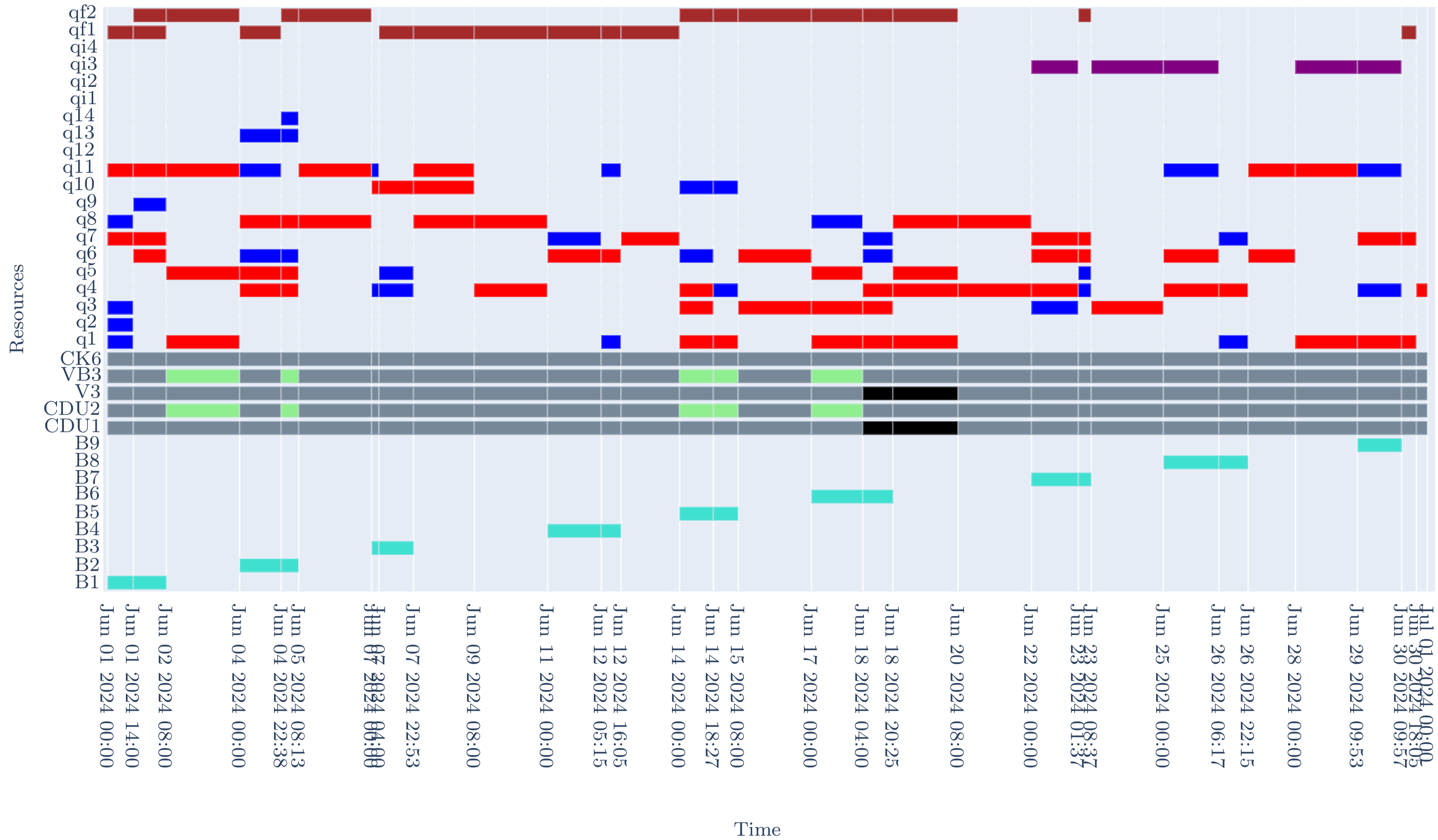

Figure 5.1: Gantt chart of the refinery operations.

5.1.2 Inventory levels

Figure 5.2 shows the inventory level evolution in refinery tank *q1*. From this figure, it is possible to verify that inventory control in tanks is carried out correctly. To do so, we analyze whether the inventory evolution is consistent with the operations represented in the Gantt diagram (Figure 5.1).

In Figure 5.2, we observe that tank 1 starts with a level of 40000 m^3 and increases to 61020 m^3. This is because it receives a transfer of 21020 m^3 from vessel 1, as shown in the Gantt chart. Then, tank 1 remains idle for a period, and its inventory level remains unchanged. During the third slot, tank 1 feeds both CDUs, discharging a total of 18920 m^3, and therefore its inventory level decreases to 42100 m^3. Over the next eight slots, the tank remains inactive until it receives a load of 19900 m^3 from vessel 4, increasing its inventory level. It then stays idle during slot 13. In slots 14 and 15, it discharges again: first, it transfers 9610 m^3 to CDU 2, and then feeds both units with a total volume of 16240 m^3. The tank remains inactive in the following slot, and discharges again in slots 17, 18, and 19, transferring 11210 m^3 to CDU 2, 7730 m^3 to CDU 1, and 5340 m^3 to CDU 2, respectively. As a result, the inventory level of tank 1 decreases to 11870 m^3 at the beginning of slot 20.

The tank stays idle until slot 25, when it receives a load of 32480 m^3 from vessel 8, increasing its inventory level to 44350 m^3. The tank participates again in feeding the units during slots 27, 28, and 29. In slot 27, it transfers 13750 m^3 to CDU 2, and during slots 28 and 29, it transfers 9490 m^3 and 6550 m^3 to CDU 1, respectively. Therefore, its inventory level decreases, reaching 14560 m^3 at the end of the horizon.

It should be noted that at no point was the maximum capacity of 62000 m^3 exceeded, nor did the inventory level fall below the lower limit of 8000 m^3.

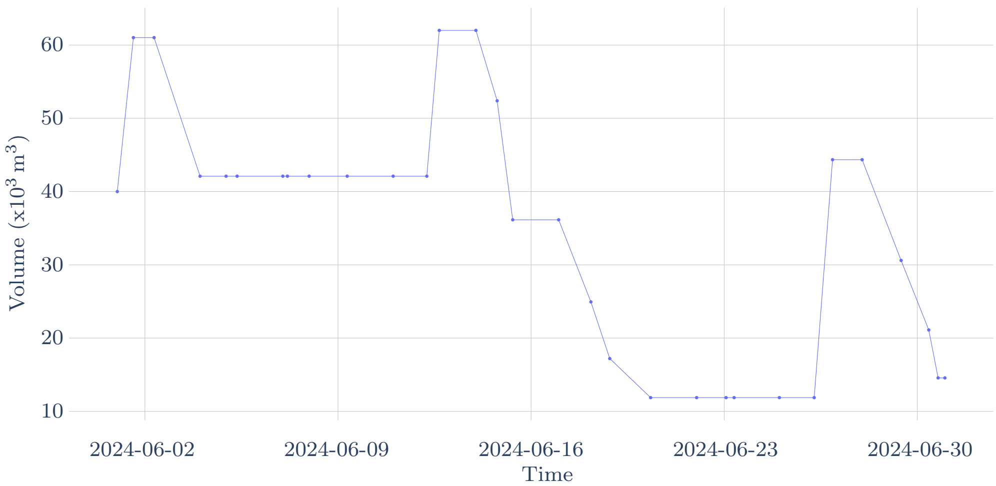

Figure 5.2: Inventory level evolution in tank 1.

5.1.3 Properties of the feed mixtures

Next, a series of figures related to the evolution of properties in the feed mixtures to the CDUs are analyzed.

First, Figure 5.3 shows the evolution of the total acid number (TAN) property for CDU 1, which has an upper bound of 1.5 mgKOH/g, represented by a green dashed line. The Y-axis represents the property value, while the X-axis shows the monthly horizon. Figure 5.4 shows the evolution of the same property but for CDU 2. We can observe that in neither case is the maximum limit exceeded.

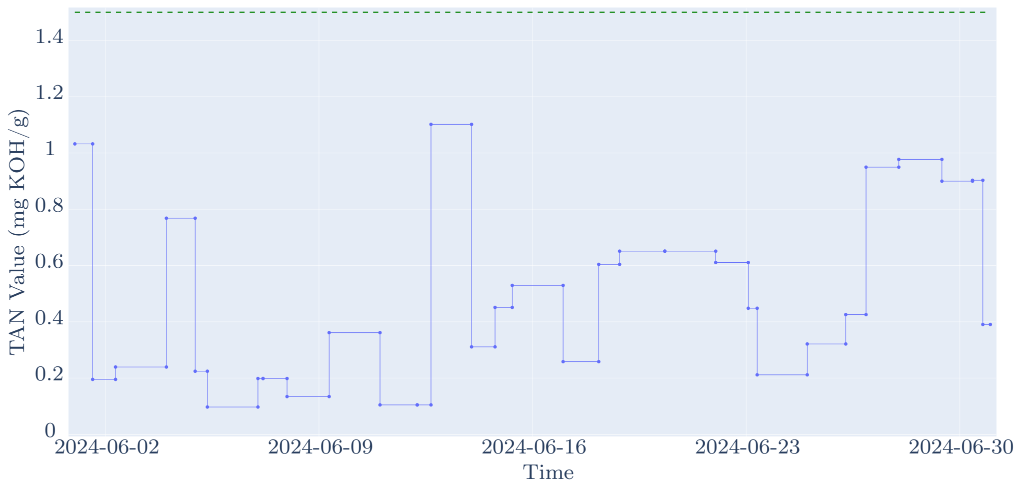

Figure 5.3: TAN property evolution in CDU 1.

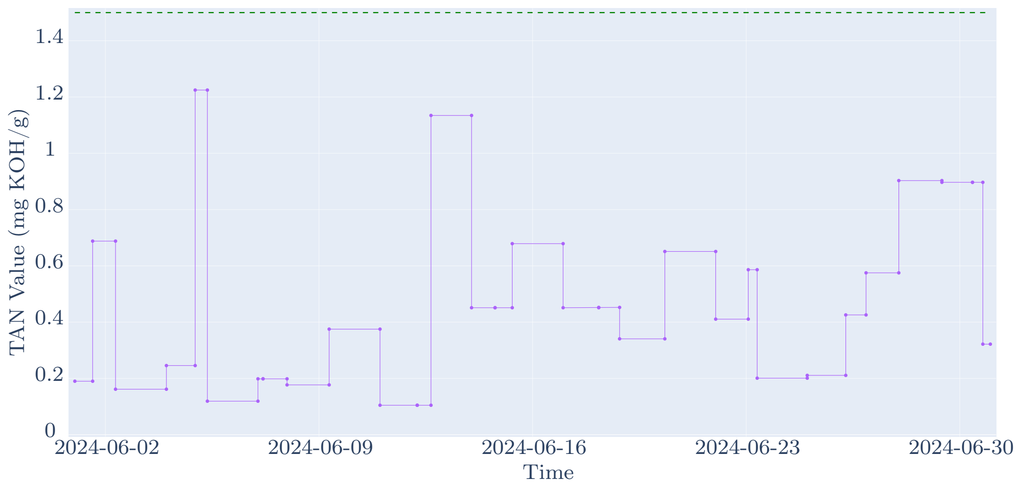

Figure 5.4: TAN property evolution in CDU 2.

Figures 5.5 and 5.6 show the value of the cetane index (CTI) property over the horizon for the feed mixtures to crude distillation units 1 and 2, respectively. This property is the most critical of all, with a lower bound of

47 for CDU 1 and 46 for CDU 2. As observed in both figures, the mixtures never fall below the established limit.

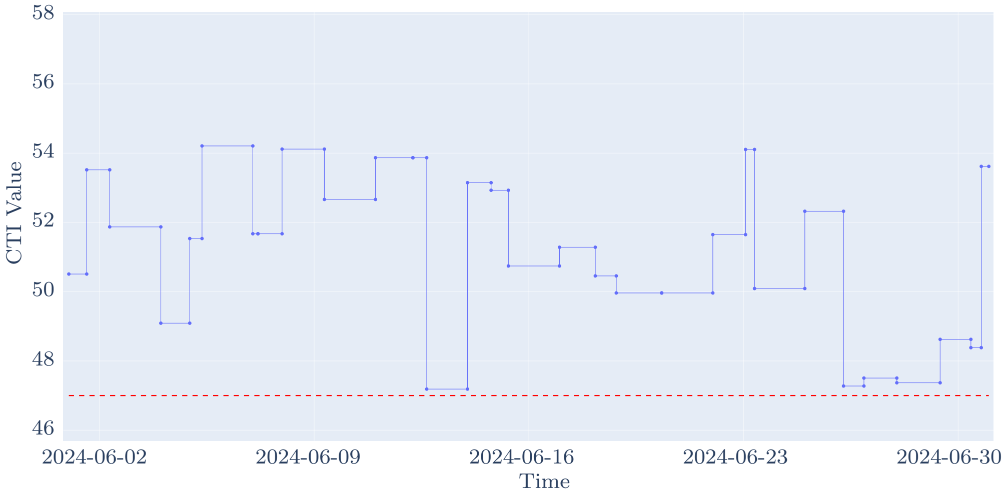

Figure 5.5: CTI property evolution in CDU 1.

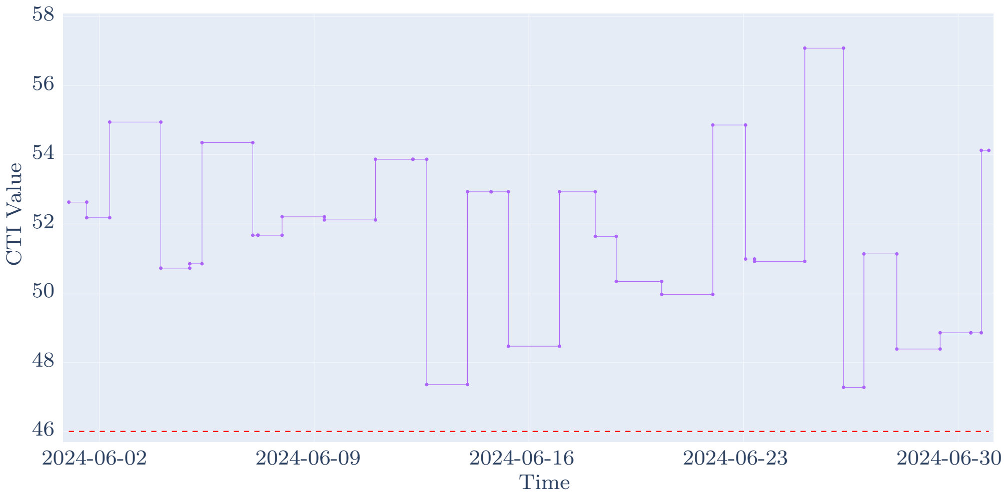

Figure 5.6: CTI property evolution in CDU 2.

Figures 5.7 and 5.8 display the evolution of the specific gravity (SPG) property for the feed to CDU 1 and CDU 2, respectively. This property has an upper bound of 0.91 in both cases. We can see that the limit is respected in both CDUs, except during a period close to the end of the horizon. However, the property value during this period is 0.914 for both cases, so this violation of the limit could be considered negligible.

Finally, the evolution over the horizon of the atmospheric residue yield (RA) is shown in Figures 5.9 and 5.10 for CDU 1 and CDU 2, respectively. There is an upper bound of 50% for both units, and a lower bound of 37% for CDU 1 and 35% for CDU 2.

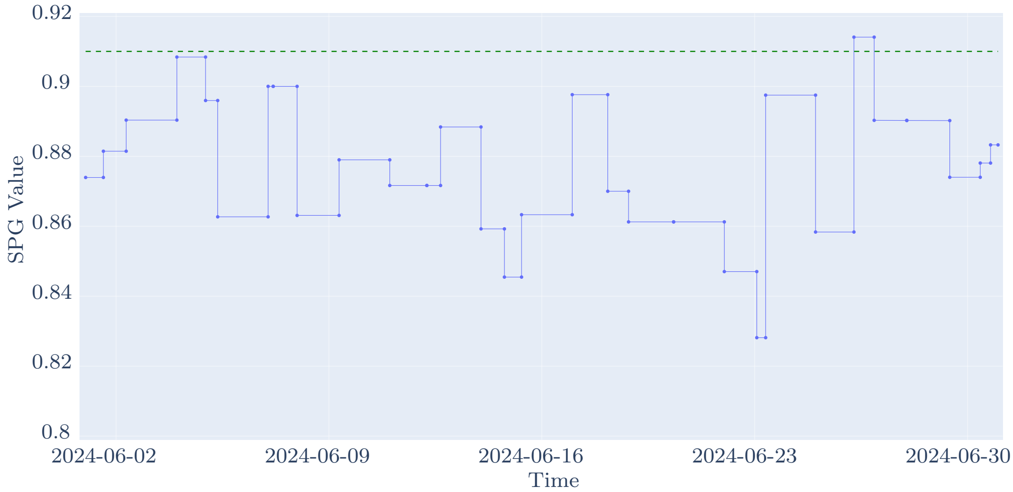

Figure 5.7: SPG property evolution in CDU 1.

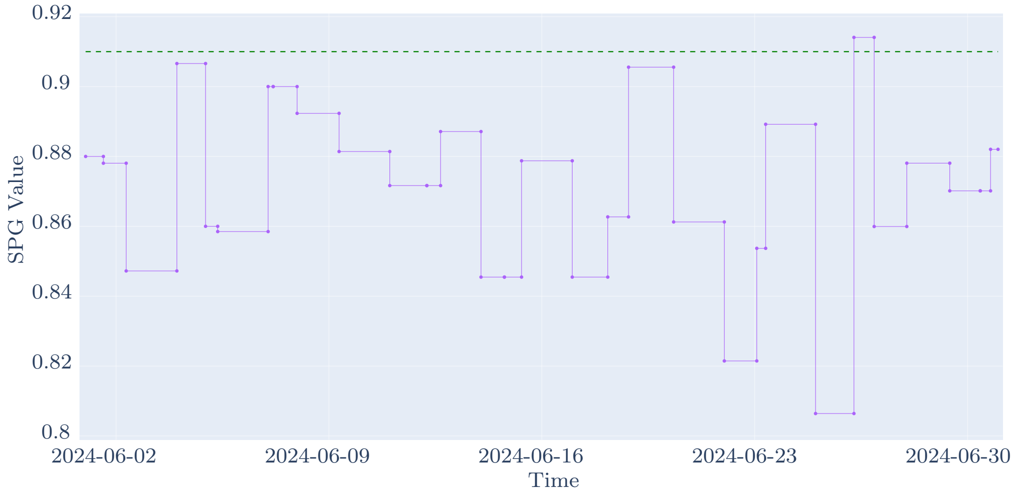

Figure 5.8: SGP property evolution in CDU 2.

By observing both figures, we can see that the maximum bound is never exceeded. However, the lower limit is violated at certain time intervals, for both CDU 1 and CDU 2; with the latter showing the greatest deviation of approximately 15% during the period closest to the end of the horizon.

The main reason for this result is that over 25% of the crude available throughout the planning horizon for forming feed mixtures consists of crude types with atmospheric residue yields below 35%. It is also important to recall that the constraints related to the calculation of property concentrations include slack variables to prevent infeasibilities. As a result, strict enforcement of the specified limits may not always be guaranteed, which is precisely the case for this property.

Finally, it should be mentioned that the total production of each process, expressed in thousands of cubic meters, was: 38.6 (Standard-V3), 30.9

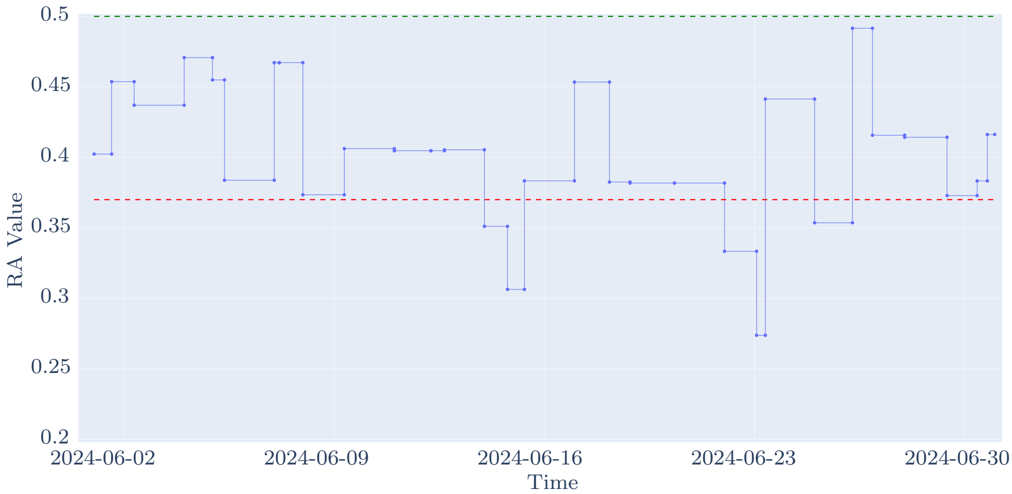

Figure 5.9: RA property evolution in CDU 1.

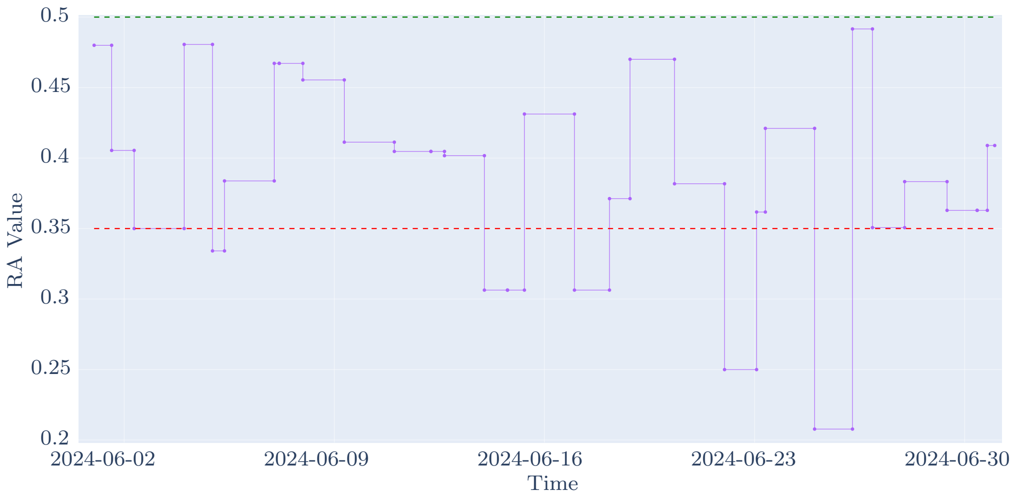

Figure 5.10: RA property evolution in CDU 2.

(Standard-VB3), 3.8 (Asphalt), and 2.2 (LS Fuel). As can be seen, overproduction occurred in all cases.

The example was solved using Pyomo and Gurobi 11.0.0 for MILPs on a computer with an Intel Core i9-13900K 3.00 GHz processor and 128 GB RAM. The problem was solved with a relative gap of less than 1%, with a total solution time of 35 minutes. Three partitions were used for each variable involved in the linear approximation, and the initial number of proposed slots was set to three.

5.2 Case study 2

A second case study is solved, considering the same initial conditions as in Case 1, with the key difference being that twelve vessels are expected to

Table 5.7: Vessel arrival and departure dates, crude type, and mass (Case 2).

Vessel	Arrival date	Departure date	Transported crude	Mass (x10^3 ton)
B1	01/06/2024	03/06/2024	BUZ	90
B2	03/06/2024	05/06/2024	CPC	90
B3	06/06/2024	08/06/2024	UBP	132
B4	08/06/2024	10/06/2024	BHP	95
B5	11/06/2024	13/06/2024	MAY	132
B6	13/06/2024	15/06/2024	BUZ	70
B7	16/06/2024	18/06/2024	ESX	90
B8	18/06/2024	20/06/2024	CPC	95
B9	21/06/2024	23/06/2024	COL	148
B10	23/06/2024	25/06/2024	BUZ	69
B11	26/06/2024	28/06/2024	BHP	70
B12	28/06/2024	30/06/2024	COL	90

arrive during the monthly planning horizon. Each vessel is characterized by its expected arrival and departure dates, the specific type of crude it transports, and the corresponding mass. This information is summarized in Table 5.7.

5.2.1 Resource operations

Figure 5.11 presents a Gantt chart illustrating the operation of the various resources throughout the planning horizon for the second case study. The operation and duration of each resource are depicted using bars, with colors assigned according to the type of resource and the specific operation performed.

As observed in the first case study, it is noteworthy that loading tank 12 and intermediate tanks 1, 2, and 4 remain idle throughout the horizon, they are neither loaded nor unloaded. This observation reinforces the previous analysis regarding the potential to increase the established production levels, provided that the refinery's processing capacity allows it, and that the underutilization of tank infrastructure could facilitate the reception of larger volumes of crude.

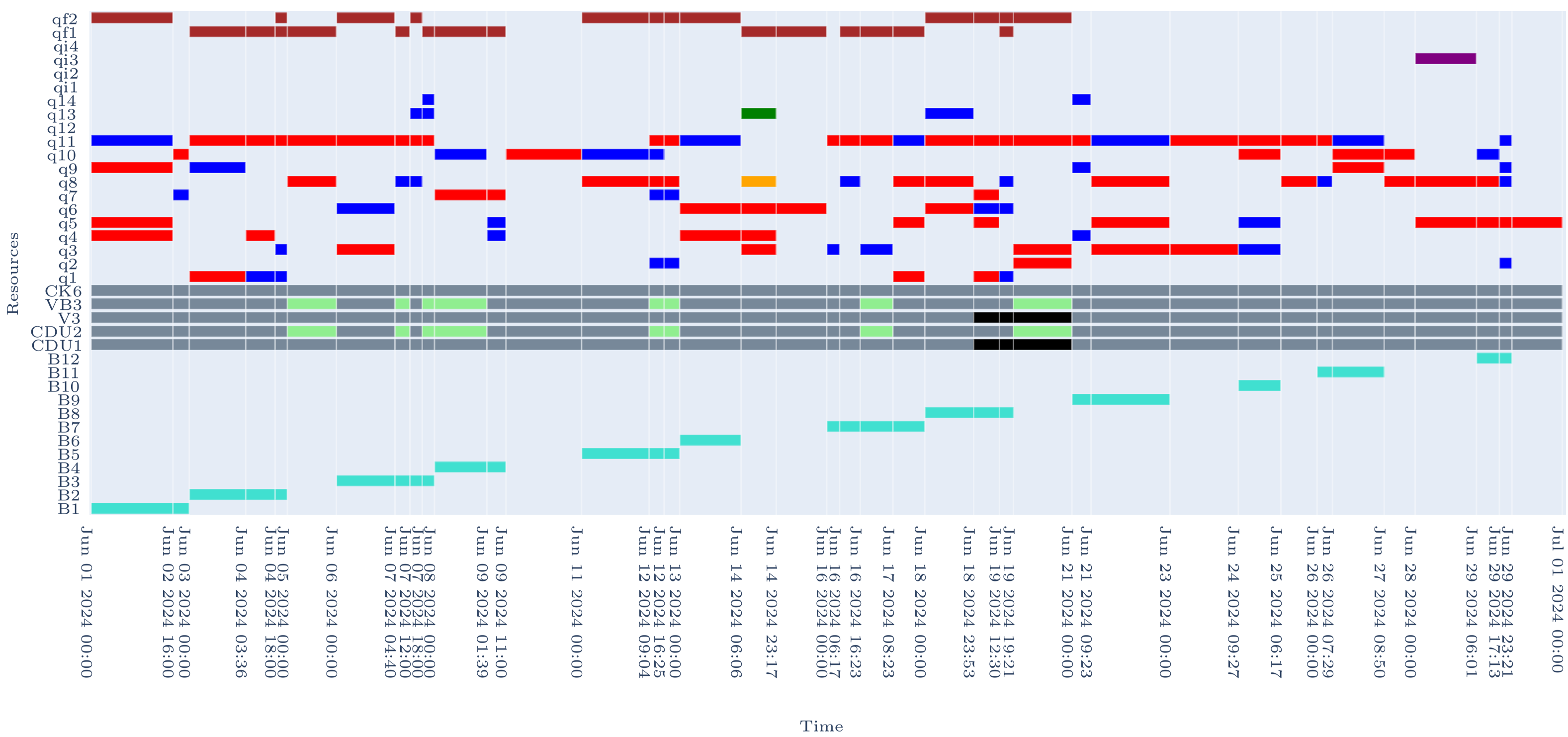

Figure 5.11: Gantt chart of the refinery operations (Case 2).

5.2.2 Properties of the feed mixtures

As in the first case study, the limits established for TAN, CTI, and SPG are met for both CDUs. Therefore, the corresponding graphs are omitted, and only the evolution of the atmospheric residue yield (RA) over the horizon is shown in Figures 5.12 and 5.13 for CDU 1 and CDU 2, respectively.

Examining both figures reveals that the upper limit is never exceeded, except for CDU 2 during a period close to the end of the horizon. However, the property value at that point is 0.516, so this slight violation could be considered negligible.

Unlike the result obtained in the first case study, the lower limit for CDU 2 is not violated, and for CDU 1, the lower bound is only breached during a single time interval, reaching a value of approximately 0.3.

Evidently, the increased availability of crudes with higher atmospheric residue yields allows for improved compliance with the property limits compared to the first case study.

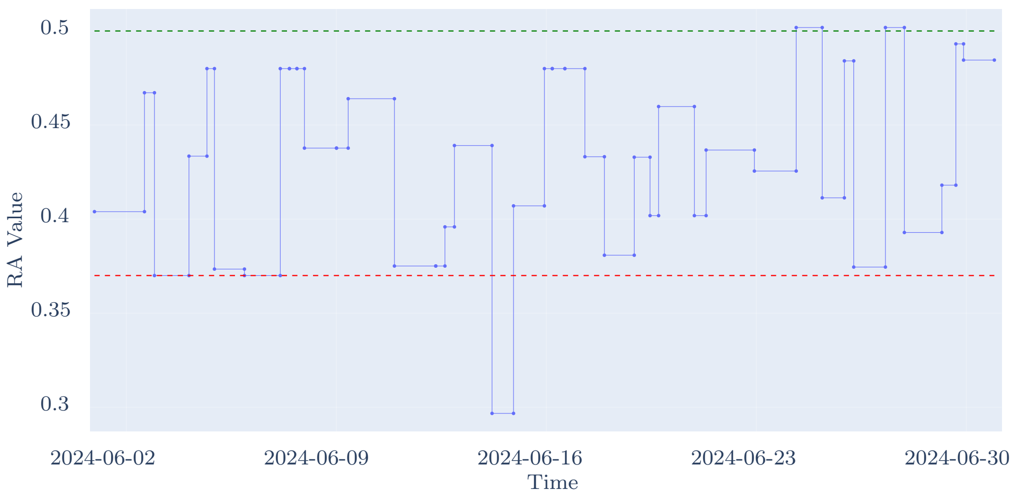

Figure 5.12: RA property evolution in CDU 1 (Case 2).

The example was solved using Pyomo and Gurobi 11.0.0 for MILPs on a computer with an Intel Core i9-13900K 3.00 GHz processor and 128 GB RAM. The problem was solved with a relative gap of less than 1%, with a total solution time of 50 minutes. Three partitions were used for each variable involved in the linear approximation, and the initial number of proposed slots was set to five.

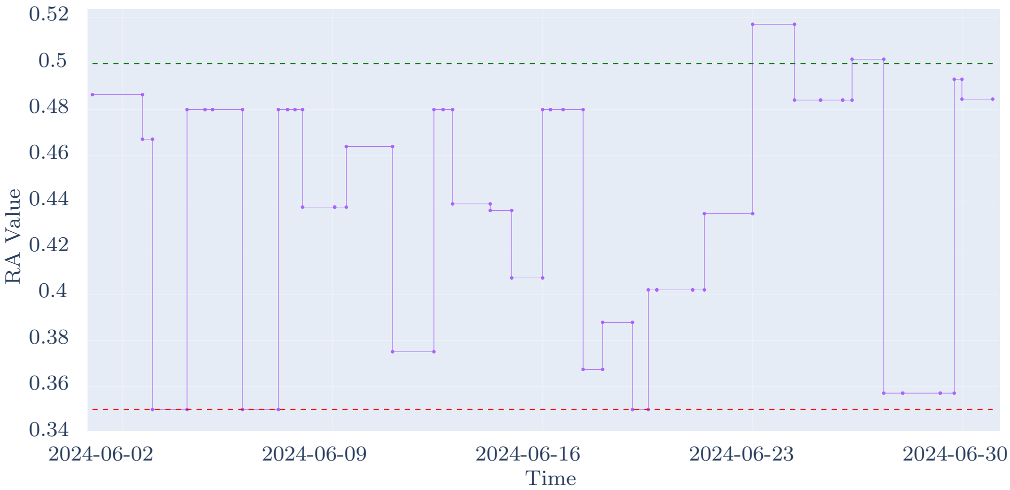

Figure 5.13: RA property evolution in CDU 2 (Case 2).

Chapter 6

Stochastic model

As mentioned in Chapter 1, another decision we need to make before developing a mathematical programming model is whether to formulate a stochastic or a deterministic model.

Deterministic models are very useful and reliable as long as the process to be optimized is not subject to large uncertainties or is not very sensitive to variations in the parameter values. Otherwise, there is a risk that the solution obtained will not be robust enough to adapt to changes in the process environment if the reality differs from what is assumed in the model, and therefore the quality of the solution is likely to decrease.

When optimizing the scheduling of crude oil operations, we should consider that weather conditions may have an impact on the arrival time of vessels and thus on the availability of crude oil. In turn, these events affect downstream decisions. Based on this premise, we have considered the arrival dates of ships at the terminal as the most important uncertainty and, consequently, it must be considered explicitly. This uncertainty is represented by a set of scenarios that cover possible ship arrival dates with different probabilities that can be obtained with historical data and current conditions.

The objective of this chapter is to assess the impact of incorporating this type of uncertainty into the crude oil operations scheduling problem. Numerous approaches exist for incorporating this kind of uncertainty, including chance-constrained, robust optimization, and two-stage methods, among others.

In chance-constrained, we calculate an optimal policy such that the probability of fulfilling the constraints is greater than a certain user-defined value. The main problem of this approach is the complexity of the numerical solution in a mixed-integer context.

In robust optimization, the decision variable is optimized for the worst case of the uncertain variable, so the constraints are satisfied for all values of the uncertainty. The robust solution is guaranteed to remain feasible over the entire range of uncertain parameter realizations [40], but the solution may be too conservative.

Finally, the two-stage stochastic programming approach involves two types of decision variables: the first-stage variables ("here-and-now" variables) which have to be implemented now and influence all future decisions, and the second-stage ones that will be implemented later on when more information about the process is available so that they can be adjusted to the realization of the uncertainty (recourse variables or "wait-and-see" variables). This provides solutions that are less conservative than robust formulations.

In the present work, we propose a model based on two-stage stochastic programming with recourse [8] to tackle the crude oil operations scheduling optimization, considering uncertainty in the arrival date of the vessels, because of the following reasons:

- The uncertainty does not depend on the decisions made and several scenarios can be selected in a sensible way as small variations of the planned arrival dates of ships.
- A discrete probability distribution can be defined for the uncertainty, i.e., for the arrival time of vessels.
- The structure of the problem makes it possible to clearly define which are first-stage decisions and which are second-stage decisions.
- It gives robust solutions without being extremely conservative.

In this problem, the first-stage variables refer to those related to the supply of blends to the CDUs. This involves the allocation of tanks to units, the output flow rates from tanks, and the input flow rates to CDUs. The start time, end time, and length of slots are also considered first-stage decisions.

As for the second-stage decisions, they encompass decisions related to: activities carried out at the marine terminal after a ship has arrived, inventory management (including types of crudes and levels stored in each tank), crude concentration in the output flow from tanks, and therefore crude concentration in the input flows to CDUs. The crude oil concentration indicates the proportion of each type of crude oil in a flow rate. It could also be thought of as the flow rate of each crude oil.

The decision regarding which variables correspond to the first stage is based on the constraint of continuous operation of the CDUs, as well as on the exercise of putting ourselves in the operator's shoes. If we were the decision-makers, then we would have to decide on the feeding of the CDUs at all times, even without knowing the exact arrival dates of the ships. This involves selecting the tanks that will feed the CDUs, the output flow rates from tanks, and the start and end times of these feeding operations. All these decisions determine the flows and composition of the CDU feeds.

On the other hand, each time a ship arrives, we must decide in which tanks the crude oil will be unloaded and the quantity to be transferred. These decisions constitute the recourse variables since they allow us to correct the choices made in the first stage and maintain the solution's feasibility in the scenarios under study.

An important point to consider is related to concentrations. Among the first-stage variables, we have mentioned the flow rate from each tank to each CDU. This flow rate can be defined as the sum of the output flow of each crude oil while respecting their in-tank concentrations. However, the output flow of each type of crude oil is a second-stage variable since the amount of each crude oil in the feeding tank may vary between scenarios. This is equivalent to saying that the composition of each tank is a second-stage variable.

6.1 Scenario definition

As mentioned earlier, the crude oil supply is subject to uncertainty owing to possible deviations in the scheduled arrival time of the ships. This uncertainty is depicted by a discrete set of scenarios that contemplates different arrival times. These scenarios are generated as follows.

For each ship, we assess a most likely arrival date according to the company planning and the potential deviations from it, both in terms of arriving earlier or later. This process generates three possibilities per ship, each accompanied by a corresponding probability of occurrence. Afterward, we build scenarios by considering the combinations of all dates. Here, we assume that the arrivals of the ships are mutually independent events.

Regarding this last assumption, one could argue that local weather conditions at the refinery site might introduce correlations between the arrival times of the vessels. However, it is important to note that the ships do not necessarily share the same origin or route; therefore, their arrival times may also be influenced by weather conditions at the point of departure or along their respective routes. Moreover, in practice, vessel arrivals are typically scheduled with one- or two-day intervals, which tends to mitigate the effect of potential dependencies.

6.2 Assumptions and model

While the obvious way to do this would be to apply the two-stage stochastic approach to the deterministic model presented in Chapter 3, it is important to keep in mind that this model is already large and complex. Introducing uncertainty would further increase its size, making it impossible to solve.

Therefore, a simpler case is studied, based on the scenario shown in Figure 1.2. The system consists of a single-dock marine terminal and an oil storage and processing section, both connected by a pipeline. Only storage tanks are available, and they are not dedicated to a single type of crude oil;

that is, blends of crude oils can be stored. The tanks are connected by mixing pipelines to the crude distillation unit area, where the final mixtures are prepared to achieve the desired flows and properties required by the different CDUs. Two CDUs are considered in this study. Additionally, the model is intended for short-term scheduling applications, where neither specific processes nor tank classifications are considered, and properties are treated in a generic manner.

Although the resulting model does not fully represent the real refinery studied in this thesis, it serves as proof of concept, allowing us to analyze the effects and benefits of considering uncertainty in this type of problem. Moreover, it provides insights that could be valuable when applying this approach to larger models.

Most of the assumptions made for the deterministic model also apply to the stochastic model, since the latter is based on a simplified case of the former. However, exceptions arise in assumptions related to loading tanks and downstream units, as these resources are not considered in the stochastic model. Nevertheless, all assumptions are explicitly stated below to ensure clarity regarding which ones apply.

1. There is only one pipeline connecting the terminal with the refinery, so only one vessel can unload at any moment.

2. A vessel that has started unloading crude can leave the terminal only once it is completely emptied.

3. Each vessel carries a single type of crude oil, and it is considered that the pipeline has a negligible volume compared to the volume to be unloaded.

4. A tank cannot receive crude from a vessel and feed a CDU at the same time. After receiving crude, a tank should stay idle for some time for brine settling and removal.

5. A maximum number of tanks can be loaded simultaneously, and transfers between tanks are not allowed.

6. There is a maximum number of CDUs that a single tank can feed simultaneously.

7. There is a maximum number of tanks that can feed a CDU at the same time, and the time to change over tanks is negligible.

8. Perfect mixing of crudes occurs in the mixing pipelines.

9. It is not allowed to stop feeding the crude distillation units.

Based on these assumptions, we developed the two-stage stochastic programming model. Appendix A includes the nomenclature of sets, parameters, and variables. Additionally, it contains the model's constraints, except for the objective function, which is explained in the next section.

6.3 Objective function

The cost associated with each scenario e, including first and second stage terms, is calculated by using (6.1). The first term, the costs due to the difference between processed volume and required demand, comprises the first-stage cost. The variables $exprod_u$ and $shprod_u$ represent the overproduction and underproduction, respectively, concerning the demand for CDU u. The second term, demurrage and departure tardiness costs, represents the second-stage cost, where the variables $dmg_{v,e}$ and $tdn_{v,e}$ refer to the demurrage and departure tardiness of vessel v under scenario e, respectively. In both cases, the parameters in capital letters correspond to the unit costs related to each variable.

$$\begin{aligned} ze_e = & \sum_{u \in U} (CEP_u * exprod_u + CSP_u * shprod_u) \\ & + \sum_{v \in V} (CDMG_v * dmg_{v,e} + CTDN_v * tdn_{v,e}) \quad \forall e \in E \end{aligned} \tag{6.1}$$

The objective function is composed of the first-stage cost and the expected value of the second-stage cost, considering all scenarios e. To calculate it, we sum the costs associated with each scenario (ze_e), weighting them according to their probability of occurrence (π_e), as shown in (6.2).

$$MIN \sum_{e \in E} \pi_e * ze_e \tag{6.2}$$

6.4 Deterministic equivalent program

There are different ways to solve the two-stage stochastic programming model. In this paper, we use the deterministic equivalent program approach, which consists of solving the first and second-stage variables together, and thus simultaneously obtaining a feasible solution for each scenario.

The optimization model results in a mixed-integer nonlinear programming (MINLP) model. The objective function is given by (6.2), and the problem is subject to constraints (a.1-a.59, and 6.1), among which only (a.59) is nonlinear. This nonlinearity is caused by the fact that the calculation of the outlet volume for each crude oil type involves the product of the total outlet volume and the concentration of that crude oil type, both of which are also variables.

6.5 MINLP solution procedure

As in the deterministic model, the two-stage stochastic programming model is formulated as a MINLP, where the non-linear constraint involved is non-convex. Therefore, it is essential to apply strategies that effectively address this issue.

For this case, we do not implement the strategy developed in Chapter 4. Instead, we propose a different approach consisting of two steps. This approach is suitable given the nature of the problem, as no significant variations in tank composition are expected within a short-term horizon.

Initially, a mixed integer linear programming (MILP) model, which is an approximation of the original MINLP, is solved. The approximate MILP formulation is obtained by replacing the nonlinear constraint (a.59), the one which states that the crude oil concentration at the outlet of a tank must be the same as the one inside the tank, with the linear constraints (6.3) and (6.4), which determine that a tank maintains the initial crude c concentration until the moment it receives crude oil from a ship.

In more detail, if a tank has not received any loading until slot s inclusive, then the binary variable $lq_{q,s',e}$, indicating if the tank is receiving a load, will be equal to zero for slot s and all previous slots ($s' \leq s$). Thus, from equations (6.3) and (6.4), the volume of crude oil type c unloaded ($vcqu_{c,q,u,s,e}$) will be equal to the total volume unloaded ($vqu_{q,u,s}$) during slot s, multiplied by the initial concentration of crude oil c in that tank ($CONC_{q,c}$). In case the tank has previously received a load, then the constraints become idle.

$$vcqu_{c,q,u,s,e} \leq CONC_{q,c} * vqu_{q,u,s} \ + \ M5_q * \sum_{s' \in S, s' \leq s} lq_{q,s',e} \qquad \forall c \in C,\ \forall q \in Q,\ \forall u \in U,\ \forall s \in S,\ \forall e \in E \tag{6.3}$$

$$vcqu_{c,q,u,s,e} \geq CONC_{q,c} * vqu_{q,u,s} \ - \ M5_q * \sum_{s' \in S, s' \leq s} lq_{q,s',e} \qquad \forall c \in C,\ \forall q \in Q,\ \forall u \in U,\ \forall s \in S,\ \forall e \in E \tag{6.4}$$

Subsequently, the values of the binary variables in the original MINLP (allocation of vessels to tanks, allocation of tanks to units, et cetera) are fixed according to the solution obtained for the MILP, and the resulting nonlinear programming (NLP) model is solved to get the values of volumes loaded into the tanks, volumes unloaded from the tanks, start-time and duration of operations, among others. In case no feasible solution can be reached, the original MINLP model is solved using an outer approximation solver (DICOPT [41]).

This procedure is summarized in algorithm 1, and depicted in Figure 6.1. In this figure, each circle represents a type of model and its components: the objective function, constraints, and variables. Initially, starting from the MINLP model, we obtain the MILP model (linear approximation) and solve it (MILP solution). Then, based on this solution, we fix the values of the binary variables in the MINLP model (binary variables MILP) and solve the resulting NLP model (solution).

Algorithm 1 MINLP solution strategy

Replace equation (a.59) of MINLP model with equations (6.3) and (6.4).
Solve resulting MILP model.
Fix binary variables from MILP solution in MINLP model.
Solve resulting NLP model.
if NLP solution is infeasible **then**
    Solve MINLP model with DICOPT solver.
end if

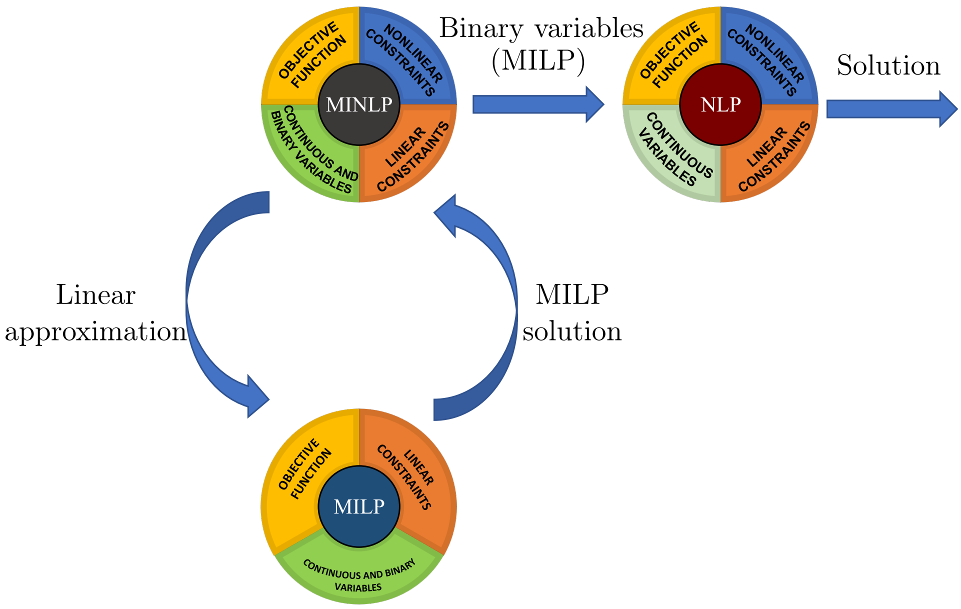

Figure 6.1: Solution procedure.

It should be noted that the algorithm is performed only once. However, it could be extended to an iterative approach as follows. If a feasible solution is not obtained for the NLP, "no good" constraints could be added to the MILP. These constraints force at least one binary variable to change its value in the new solution. This procedure would be repeated until a feasible solution for the NLP is found or a certain number of iterations is exceeded.

6.6 Stochastic model performance evaluation

One of the objectives of this study is to evaluate whether two-stage stochastic programming offers any advantage over simpler deterministic approaches. To this end, Birge and Louveaux [8] proposed the value of the stochastic solution (VSS) and the expected value of perfect information (EVPI) as performance indicators. Before defining both measures, it should be noted that the two-stage stochastic programming model is also known as recourse problem (RP).

6.6.1 Expected value of perfect information

Suppose that we have perfect information about uncertainty, i.e., we know with complete certainty the arrival date of the ships every time we have to make a decision; in other words, we know what the future scenario will be. Furthermore, let us assume that there is at least one feasible solution for each of the scenarios considered. Thus, we could solve each of them separately obtaining the corresponding optimal solution and the associated value of the objective function. If we were to repeat the procedure of applying the appropriate optimal solution every time we have to make a decision, we would obtain the minimum expected cost in the long run.

Therefore, the minimum expected cost is equal to the sum of the optimal costs associated with each scenario weighted by its probability of occurrence what is known as the wait-and-see solution (WS).

Finally, we obtain the expected value of perfect information (EVPI) as the difference between the RP solution and the WS solution. The EVPI represents how much we would be willing to pay, each time we have to make a decision, to obtain perfect information about the arrival of the ships.

In practice, gaining access to better information might involve, for example, paying a specialized company to track the position and status of incoming ships in real time or purchasing detailed weather forecasts that improve the accuracy of arrival predictions, among other options. However, such an investment would only be justified if the EVPI is greater than or equal to the cost of acquiring this information.

$$EVPI = RP - WS \tag{6.5}$$

6.6.2 Value of the stochastic solution

One may ask which is the advantage of using the stochastic solution over the deterministic one. If we do not wish to use the RP, then we can solve a deterministic problem with the expected arrival times of the ships. This approach is known as the expected value problem (EV), and its solution is called the expected value solution.

With this solution, we will apply the first-stage variable values. Then, when a ship arrives, the best we could do is to solve another deterministic problem in which we fix the values of the first-stage variables and optimize the second-stage variables.

In the long run, the cost will correspond to the weighted average of all these solutions, and it is known as the expected result of using the EV solution (EEV).

Finally, the value of the stochastic solution (VSS) compares the EEV solution and the RP solution in order to quantify the reduction of the expected cost when considering the randomness of the uncertainty versus its weighted average.

$$VSS = EEV - RP \tag{6.6}$$

It should be noted that there is a risk that the solution obtained from the EV model may be infeasible for one or more scenarios. Therefore, when optimizing the second-stage variables, we have incorporated slack variables related to vessel unloading start dates to achieve convergence in those scenarios for which the solution is infeasible.

6.7 Risk management

The approach described above (a.1-a.59, 6.1, and 6.2) does not evaluate the risk associated with the objective function, that is, it is risk-neutral and only seeks to minimize the expected cost in the long run without taking into account the probability distribution of the objective function, that is, the probability of having very bad values of the cost function in case some scenarios are realized. However, it is often important to consider this distribution to reduce the risk that the solution obtained could produce high costs in certain scenarios.

For example, consider a situation in which a vessel arrives very late, corresponding to a low-probability scenario. If the first-stage decisions are based solely on expected arrival times, it could happen that the solution requires processing large volumes of a specific type of crude oil—the same type carried by the delayed vessel. As a result, the unavailability of that crude due to the vessel's delay could force the plant to shut down, leading to high operational costs. This highlights the importance of not only minimizing expected costs but also mitigating exposure to worst-case scenarios.

For this purpose, there are two popular risk measures: Value-at-Risk (VaR) and Conditional Value-at-Risk (CVaR). On the one hand, VaR with confidence level $1-\alpha$ determines the minimum value ω^*, where ω^* is the value of the cost function J such that the probability of obtaining a value of J less than ω^* is $1-\alpha$. On the other hand, CVaR with confidence level

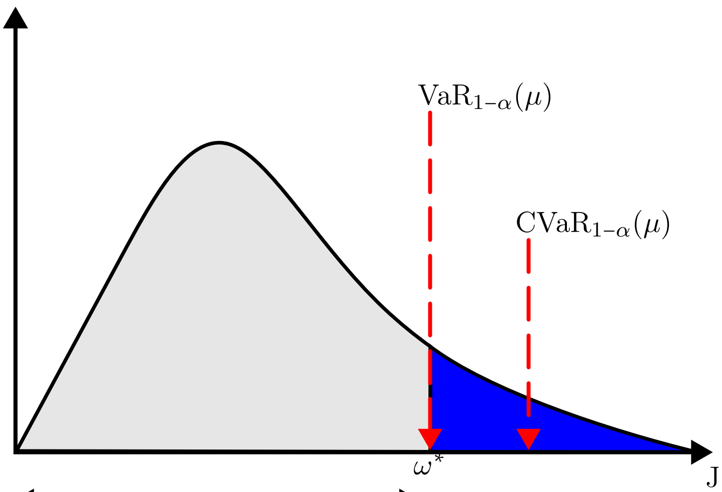

Figure 6.2: Graphical representation of the CVaR.

$1-\alpha$ represents the average value of the tail of the distribution, above the $\mathrm{VaR}_{1-\alpha}$ value (Figure 6.2).

In Figure 6.2, the y-axis represents the values of the probability density function. The x-axis represents the values of the random variable: the objective function (J). The symbol ω* represents the value of J obtained from $\mathrm{VaR}_{1-\alpha}$. The gray area under the curve represents the value of the cumulative distribution function evaluated at ω*, which is the probability that J is less than ω*. The blue area under the curve (distribution tail) represents the probability that J is greater than ω*.

Both risk measures have similar meanings but, even if VaR is easier to interpret, the solution of the optimization problem becomes quite complex, so in this work, we use CVaR because it is simple to calculate and consistent since if the cost function is convex with respect to the decision variables (μ), then the CVaR function is also convex.

The formulation of the two-stage stochastic programming model with risk management maintains the same constraints as the original two-stage model (a.1-a.59, and 6.1) and incorporates the following.

In (6.7), a non-negative auxiliary variable is defined for each scenario (ϕ_e). It takes a value greater than zero if the cost of the scenario (ze_e) is greater than VaR variable (var); otherwise, it can be made zero. In (6.8), the value of the CVaR variable ($cvar$) is calculated. Constraint (6.9) ensures that the auxiliary variable (ϕ_e) is non-negative.

$$ze_e - var \leq \phi_e \quad \forall e \in E \tag{6.7}$$

$$cvar \ = \ var \ + \ (1/\alpha) * (\sum_{e \in E} \pi_e * \phi_e) \tag{6.8}$$

$$\phi_e \geq 0 \quad \forall e \in E \tag{6.9}$$

Moreover, the objective function (6.2) is replaced by (6.10).

$$MIN \ cvar \tag{6.10}$$

Chapter 7

Stochastic results

This chapter focuses on evaluating the performance of the stochastic optimization approaches developed in Chapter 6. In this context, the EVPI and VSS measures are computed to assess the impact of incorporating uncertainty into the model. Furthermore, results from the risk-neutral and risk-averse formulations are compared at different confidence levels, highlighting the effect of incorporating risk management into the decision-making process.

7.1 Stochastic case study

To evaluate the application of the proposed stochastic models, we consider a case study involving a maritime terminal with a pipeline connection, five storage tanks, two crude distillation units, and five crude oil types characterized by a single key property. The planning horizon spans 120 hours (5 days), during which the arrival of two vessels is expected. The arrival times and associated scenario probabilities are presented in Table 7.1. Each vessel is assumed to depart 12 hours after arrival. The demand for CDU 1 is 100000 m^3 and for CDU 2 is 65000 m^3 over the scheduling horizon.

The different formulations mentioned in Chapter 6 have been solved and compared in the following sections.

7.2 Risk-neutral approach analysis

In this section, we analyze the results obtained from the two-stage risk-neutral stochastic programming model (a.1-a.59, 6.1, and 6.2), also known as the recourse problem (RP).

First, in Table 7.2, we observe that the expected cost of the solution is 20.97 k€. This value corresponds to the sum of the costs of each scenario, weighted according to their probability of occurrence.

Second, the costs associated with each scenario are detailed in Table 7.3 under the "Risk-neutral (RP)" row. Here, we can see that they are

Table 7.1: Arrival times and probabilities.

Scenarios	Probabilities	Arrival time (h)	
		Ship 1	Ship 2
1	0.01	5	35
2	0.03	45	35
3	0.01	85	35
4	0.2	5	65
5	0.5	45	65
6	0.2	85	65
7	0.01	5	95
8	0.03	45	95
9	0.01	85	95

greater than zero in all cases except for scenarios 4 and 6, and the worst case corresponds to scenario 2, which has a cost of 159 k€. It should be mentioned that none of the scenarios incur first-stage costs; in other words, the demand is met exactly, and these costs are exclusively attributed to demurrage and tardiness in the unloading of the ships.

Next, we conduct a detailed analysis of Figure 7.1, corresponding to a Gantt chart of ship unloading for the RP solution. On the vertical axis of this diagram, the scenarios are indicated, and on the horizontal axis, the timeline (scheduling horizon). Then, blue and red bars represent the operation of vessels 1 and 2, respectively, and gray bars represent the time intervals during which no vessel is operating. Please note that slot durations are first-stage variables, and therefore, they are the same for all scenarios.

In scenario 1, vessel 1 unloads the crude oil at the scheduled times. In contrast, vessel 2 experiences demurrage and tardiness, beginning its discharge 10 hours later than the scheduled arrival time and finishing at hour 65 instead of hour 47 (12 hours after arrival). In scenario 2, ship 1 starts unloading at the scheduled time (hour 45) but finishes at hour 65, eight hours later than expected. As for ship 2, it incurs a 30-hour demurrage, and consequently, it also fails to meet the stipulated departure time. In this case, we can observe that the order of ship arrivals is not respected when unloading. Regarding scenario 3, we observe that vessel 2 starts unloading 10 hours later than expected, and its tardiness is 18 hours, as it should have left the terminal by hour 47. In scenarios 4 and 6, both vessels adhere to their scheduled times for starting and finishing unloading, resulting in a cost of zero for these scenarios. In scenario 5, both ships start unloading at the expected time, but ship 1 incurs an eight-hour demurrage. Finally, in scenarios 7, 8, and 9, ship 1 begins unloading at the scheduled time and only finishes later than expected in scenario 8. As for ship 2, in all three scenarios, it starts and finishes unloading at hours 97 and 120, respectively, which means there is a two-hour demurrage and a 13-hour tardiness.

Table 7.2: Expected costs associated with proposed models, and values of EVPI and VSS.

RP	WS	EEV	EVPI	VSS
		(x10^3 €)		
20.97	0.45	1078.13	20.52	1057.16

Table 7.3: Cost per scenario.

Model	Cost per scenario (x10^3 €)								
	e1	e2	e3	e4	e5	e6	e7	e8	e9
$CVaR_{0.99}$	0	30	0	24	54	24	42	72	42
$CVaR_{0.7}$	45	99	45	30	30	30	42	42	42
$CVaR_{0.6}$	45	147	45	24	24	24	42	42	42
Risk-neutral (RP)	69	159	69	0	24	0	42	66	42
EEV	205	123	4123	196	0	4000	3196	3000	7000
WS	0	12	0	0	0	0	0	0	9

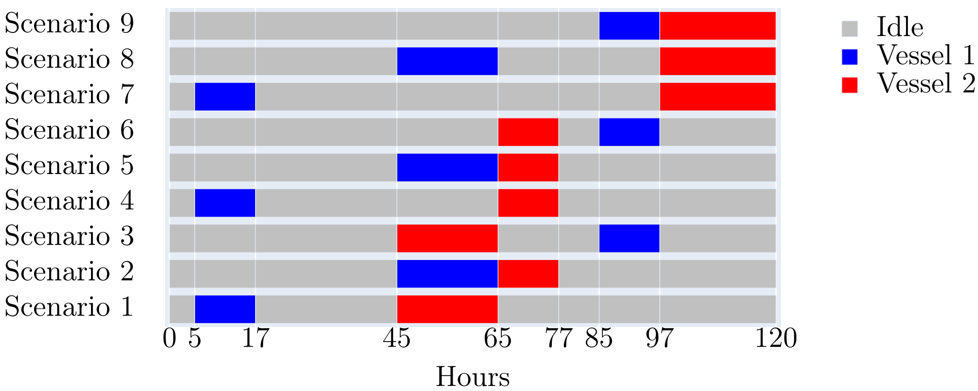

Figure 7.1: Gantt chart of risk-neutral solution.

7.3 EVPI and VSS calculation

The cost function values obtained for the recourse problem solution (RP), the expected result of using the EV solution (EEV), and the wait-and-see solution (WS) are shown in Table 7.2. Moreover, the expected value of perfect information (EVPI) and the value of the stochastic solution (VSS) are presented in the same table to analyze the effect of considering uncertainty. From the value of the EVPI, we can conclude that if we have access to perfect information, the RP solution will improve by up to 20.52 k€, so this is the maximum that we would be willing to pay to obtain that information. Besides, the VSS indicates that it is worth using the two-stage stochastic optimization since the expected cost when using the mean value of the uncertain parameter increases by 1057.16 k€. It is important to mention that this value is very large since there are scenarios for which the first-stage EV solution is infeasible, and this feature is represented by high costs. Therefore, it is preferable to apply the RP solution.

In Table 7.3, we can see the cost per scenario for the risk-neutral (RP), EEV, and WS models. The solution obtained for the first-stage variables using the EV model is infeasible for scenarios 3, 6, 7, 8, and 9, as indicated by the high associated cost for each of these scenarios. It is worth mentioning that when optimizing the second-stage variables, we introduced slack variables related to vessel unloading start dates to ensure convergence in scenarios where the initial solution is infeasible. These slack variables were incorporated into the objective function with a high cost. As a result, the objective function value is higher than three thousand in these cases, highlighting the extent of infeasibility in the first-stage solution for these scenarios.

We can also observe that the WS solution is zero in several scenarios because, with perfect information, they fulfill all specifications. However, it is non-zero in scenarios 2 and 9. This is because when ships arrive on dates close to each other, the second ship to be unloaded will inevitably incur demurrage and tardiness, even if the first ship unloads at the maximum flow rate.

7.4 CVaR analysis

The results obtained for the two-stage stochastic programming model with risk management (CVaR model) are presented below. Three cases with different confidence levels have been analyzed: 0.99, 0.7, and 0.6.

Table 7.3 shows the cost associated with each scenario for each of the evaluated cases: CVaR 0.99, CVaR 0.7, and CVaR 0.6. When we compare these three cases, we can observe that only for CVaR 0.99 are there scenarios with zero cost, specifically scenarios 1 and 3. This is due to the fact that, based on the solution obtained for CVaR 0.99, the demand is met exactly, and ships 1 and 2 are unloaded on the scheduled dates in scenarios 1 and 3. For the remaining scenarios, costs are greater than zero due to demurrage or tardiness in the operation of the ships.

Continuing with the analysis of the three solutions, we can observe that the worst-case scenario for CVaR 0.99 is scenario 8. For the CVaR 0.7 and CVaR 0.6 solutions, the worst scenario is number 2. Additionally, when comparing the three cases, we notice that the highest cost is associated with scenario 2 of the CVaR 0.6 solution.

Interestingly, we can observe from Table 7.3 that for scenarios 7 and 9, the costs associated with the RP and CVaR models are the same. This is because, in all four solutions, vessel 2 starts unloading two hours later than planned and finishes 13 hours later than its expected departure date, resulting in the same demurrage and tardiness costs for these models. It is worth mentioning that in none of the cases are first-stage costs incurred.

Table 7.4 displays the VaR and CVaR values at confidence levels 0.99, 0.7, and 0.6. In addition, the expected costs for all cases, including the risk-neutral approach, are shown. When we compare the solutions that include risk management, we can observe that as the significance level (α) decreases, the VaR and CVaR values increase. This is because a greater weight is given to the worst-case scenarios, i.e., those scenarios that present a high cost.

Likewise, as α decreases, the expected cost (value of the objective function) increases because more conservative policies are adopted. These policies help obtain solutions that avoid high costs in less probable scenarios but come at the expense of increasing the cost associated with scenarios that have a higher probability of occurring. Furthermore, we can note that the RP solution is better in terms of expected cost, but it may have very high costs in some scenarios, as can be seen in Table 7.3 for scenario 2.

Figures 7.2 to 7.4 show Gantt charts of vessel operations for three risk-managed solutions. In these figures, we can observe the arrival and departure times of the two vessels for each of the nine scenarios.

When comparing scenarios within the same case, we observe that the decisions related to the unloading of the vessels vary among them, since they are not first-stage decisions. Moreover, when comparing the three risk-managed solutions, we notice variations in the decisions regarding the start and end times of the time slots.

We also notice that the decisions (unloading dates and operation durations) related to ship 1 for scenarios 1, 3, 4, 6, 7, and 9 remain the same in all cases, including the risk-neutral case. Meanwhile, concerning vessel 2, the decisions related to scenarios 7, 8, and 9 are the same in all cases. Furthermore, only in the case of higher risk aversion ($\text{CVaR}_{0.99}$) is the order of arrival of the vessels in scenario 2 respected. This results in the lowest cost case for scenario 2, excluding the WS solution, as can be seen in Table 7.3.

Figure 7.5 illustrates the costs per scenario in each of the three risk-managed solutions and the risk-neutral solution. From this figure, we can see more clearly the effect of the adopted risk level on the obtained solution. As we assume less risk (lower α), the costs of the scenarios tend to be more uniform. Additionally, we can notice that the possibility of having scenarios with very high costs is avoided.

It is worth mentioning that although the total volume of the feed mixture is the same for all scenarios since it is a first-stage variable, the concentration of the property can be different between scenarios if the composition of the mixture changes. This fact is shown in Figures 7.6, 7.7, and 7.8, which depict the evolution of key property concentration in the feed mixture of CDU 1 for scenarios 3, 5, and 7, corresponding to the $\text{CVaR}_{0.99}$ solution.

By analyzing these figures, we can observe that the evolution of the concentration maintains a very similar profile up to hour 65 in the three scenarios, and in none of the cases, the established limits are violated. However, scenarios 3 and 7 are more critical than scenario 5 since, at the end of the horizon, the property's concentration value is at the lower bound or very close to it.

Table 7.4: VaR and CVaR values.

1-α	$\text{VaR}_{1-\alpha}$ (x10^3 €)	$\text{CVaR}_{1-\alpha}$ (x10^3 €)	Expected cost (x10^3 €)
0.99	72	72	40.5
0.7	30	39.9	32.97
0.6	24	36.53	29.01
Risk-neutral (RP)	-	-	20.97

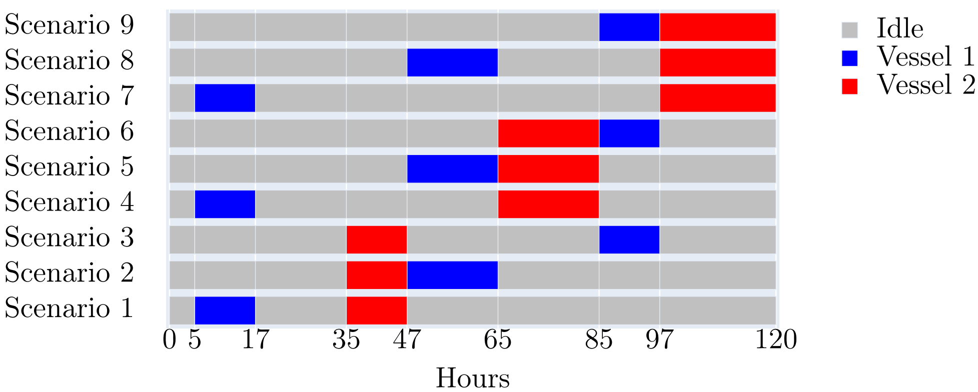

Figure 7.2: Gantt chart of $\text{CVaR}_{0.99}$ solution.

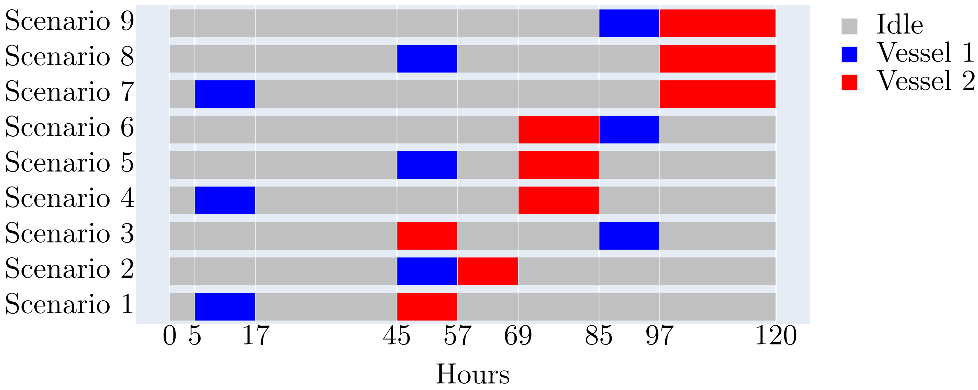

Figure 7.3: Gantt chart of $\text{CVaR}_{0.7}$ solution.

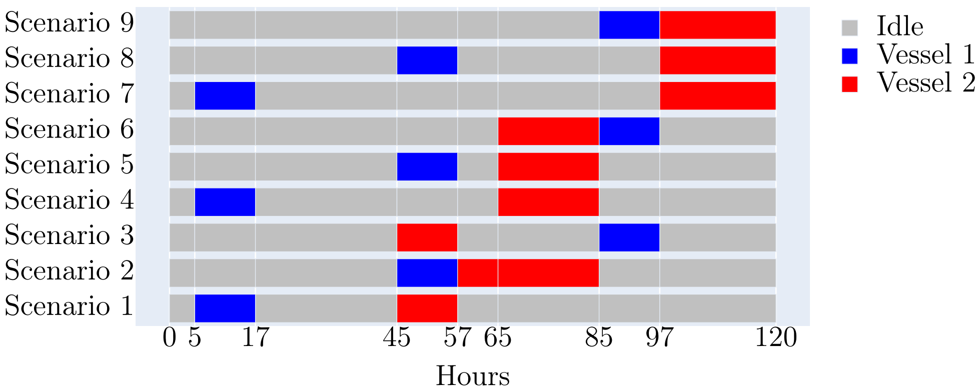

Figure 7.4: Gantt chart of $\text{CVaR}_{0.6}$ solution.

7.5 Time analysis and model statistics

In order to evaluate the complexity of the presented method, an analysis of the computational time has been carried out by varying the number of evaluated scenarios. For this purpose, four cases have been proposed: the first involves 4 scenarios, the second 9 scenarios (the same as those shown in Table 7.1), the third 16 scenarios, and finally, the fourth case with 25 scenarios. Each of the cases has been solved for the models: RP, $\text{CVaR}_{0.7}$, and $\text{CVaR}_{0.99}$; and in each of them, eight slots have been used.

As can be seen from the values in Table 7.5, as the number of scenarios increases, the solution time increases exponentially, increasing by three orders of magnitude for the RP case. Moreover, it should be noted that it was not possible to obtain solutions in less than 3600 seconds for any of the three proposed models.

From the previous result, we can see that the number of scenarios evaluated constitutes a bottleneck when solving case studies with the proposed

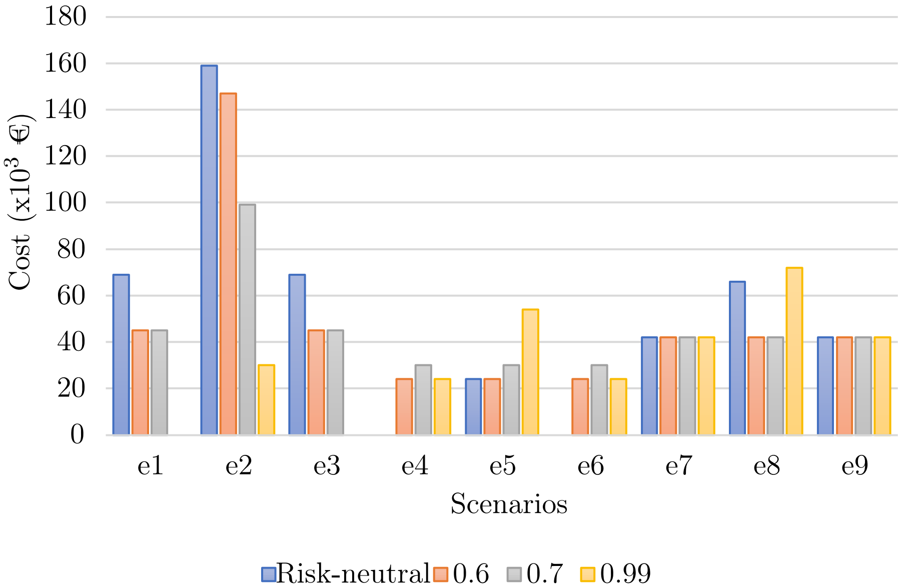

Figure 7.5: Cost of scenarios.

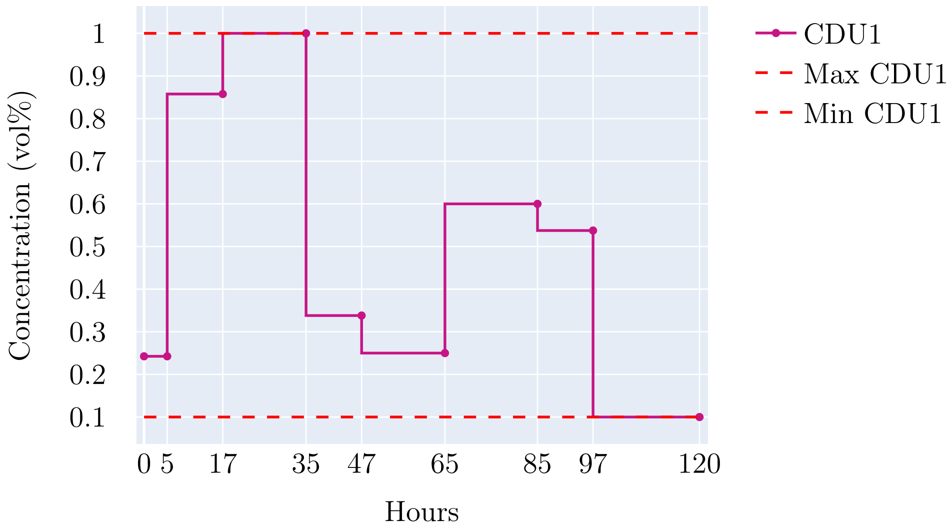

Figure 7.6: Evolution of key property concentration in scenario 3 ($CVaR_{0.99}$).

model, which is why it is of great interest to the authors to apply and develop decomposition methods in future work.

Lastly, Table 7.6 summarizes the statistics for each of the instances solved, including EV and WS models. In the EV and WS models the scenarios are solved separately, therefore, the table shows the number of variables

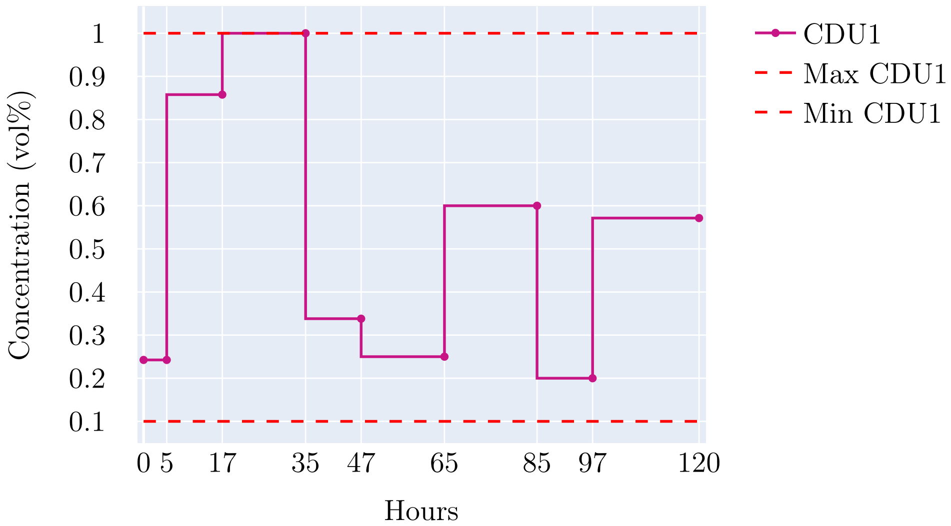

Figure 7.7: Evolution of key property concentration in scenario 5 ($CVaR_{0.99}$).

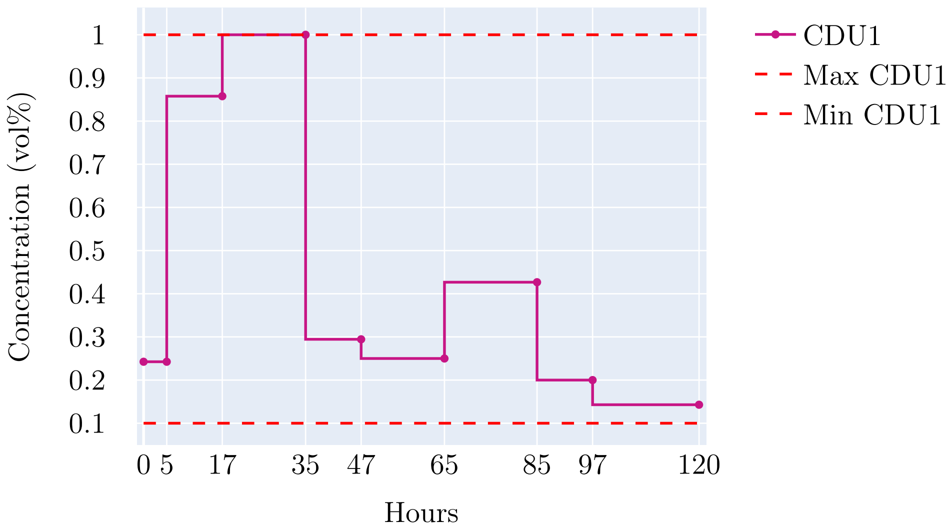

Figure 7.8: Evolution of key property concentration in scenario 7 ($CVaR_{0.99}$).

and constraints corresponding to a single scenario. However, the Time column shows the total time taken to solve all scenarios. The example has been solved using GAMS 41.3.0 software, Gurobi 9.5.2 for MILPs, and CONOPT 4.29 for NLPs on a computer with Intel Core i9-13900K 3.00 GHz processor and 128 GB RAM.

Table 7.5: Solution time for different number of scenarios.

Model	Number of scenarios (CPU time in seconds)			
	4	9	16	25
RP	8.9	40.5	1178.9	-
$\text{CVaR}_{0.7}$	4.3	19.1	482.7	-
$\text{CVaR}_{0.99}$	7.0	38.0	149.1	-

Table 7.6: Model statistics.

<table>
<tr><th rowspan="2">Model</th><th colspan="2">Continuous variables</th><th>Binary variables</th><th colspan="2">Constraints</th><th>Time (s)</th></tr>
<tr><th>MILP</th><th>NLP</th><th>MILP</th><th>MILP</th><th>NLP</th><th>MILP</th></tr>
<tr><td>$\text{CVaR}_{0.99}$</td><td colspan="2" rowspan="3">9356</td><td rowspan="4">1272</td><td rowspan="3">20016</td><td rowspan="3">16416</td><td>38.02</td></tr>
<tr><td>$\text{CVaR}_{0.7}$</td><td>19.12</td></tr>
<tr><td>$\text{CVaR}_{0.6}$</td><td>54.39</td></tr>
<tr><td>Risk-neutral</td><td colspan="2">9345</td><td>20006</td><td>16406</td><td>40.51</td></tr>
<tr><td>EEV</td><td colspan="2">1273</td><td rowspan="2">248</td><td rowspan="2">2670</td><td rowspan="2">2270</td><td>0.77</td></tr>
<tr><td>WS</td><td colspan="2">1257</td><td>2.82</td></tr>
</table>

Chapter 8

Conclusions, contributions and future research

In this work, a mathematical programming model based on a continuous-time formulation has been developed to optimize the crude oil operations scheduling in a refinery, encompassing the coordination of ship arrivals, crude oil inventory management, and the scheduling of feed to crude distillation units (CDUs) and downstream processing units.

Due to the nonlinear nature of the problem and the large model size that arises when considering a one-month planning horizon, a solution strategy was also proposed. This strategy combines a novel piecewise linear approximation technique and a temporal decomposition approach.

The original MINLP model reaches an approximate size of 700000 constraints, 165000 of which are nonlinear, along with 400000 continuous variables and 12000 binary variables. No feasible solution was obtained when attempting to solve the monolithic model after four hours of computation.

Using this methodology, it is possible to obtain high-quality solutions within computational times that meet user requirements. Two case studies based on real data from the refinery under study were solved. Both models were implemented using Pyomo and solved with Gurobi 11.0.0 for MILP models on a computer equipped with an Intel Core i9-13900K 3.00 GHz processor and 128 GB of RAM. The first problem was solved with a relative gap of less than 1%, in a total solution time of 35 minutes; the second problem was solved with a relative gap of less than 1%, in a total solution time of 50 minutes.

However, several aspects remain open for future research. First, to extend the approximation method developed in Chapter 4 to nonlinear functions involving more than two variables (e.g., the product of three variables) and evaluate its performance. Further research is also required on strategies for selecting the number of partitions and on the definition of the corresponding intervals.

Second, to develop a dynamic simulation model to evaluate the performance of the solutions obtained through optimization. While the current model provides optimized schedules, a simulation tool would allow the validation of these schedules in the presence of operational disturbances and uncertainties, providing insights into the robustness and feasibility of the proposed solutions.

Third, while discrete-time formulations often lead to larger models, especially when short-length fixed slots are used, their relaxations tend to be tighter than continuous-time models. Therefore, it would be interesting to apply the solution strategy developed in Chapter 4 to a discrete-time model. Since the approach relies on dividing the horizon into smaller subproblems, the individual models remain of manageable size even with fine discretization.

Finally, the current objective function combines different goals using weighted terms. It would be worth exploring a lexicographic optimization approach, where the original function is split into a set of objectives. These are optimized sequentially, with constraints imposed on previously optimized objectives. This could help obtain solutions that better satisfy the user's most critical objectives.

In addition to the deterministic framework, we developed a stochastic formulation for a scheduling model to characterize the operation of a system comprising a crude oil section and a maritime terminal of an oil refinery. The model is used to decide the best way of operating the crude section while considering the uncertainty linked with ship arrivals. Alongside formulating a two-stage stochastic scheduling problem, we assessed the advantages of concurrently incorporating a two-stage stochastic approach and a continuous-time formulation for optimizing crude oil operations scheduling compared to deterministic approaches. Furthermore, we examined the impact of integrating risk management into the model and how solutions vary across different aversion levels.

From the value of the stochastic solution, we can conclude that the two-stage formulation offers a more robust solution compared to deterministic approaches, mainly because it allows us to correct the consequences of decisions taken now based on future conditions.

Moreover, introducing CVaR enables the penalization of extreme values that may appear if some scenarios are realized, thereby minimizing risk. While the inclusion of risk management in the two-stage stochastic model increases the expected cost with higher risk aversion, Figure 7.5 highlights that risk-aware solutions exhibit greater uniformity compared to risk-neutral ones. Notably, in scenario 2, the cost of the risk-neutral solution exceeds five times that of CVaR at a confidence level of 0.99.

Regarding the stochastic approach, a potential future research direction is to adapt the stochastic formulation to the complete detailed model. However, based on the results obtained, it is evident that such an extension would significantly increase model size. Therefore, it is necessary to apply decomposition methods designed for stochastic programming to solve the model considering multiple scenarios and in an efficient manner.

Lastly, future work could explore the integration of machine learning techniques into crude oil operations scheduling. In particular, recent studies have applied deep reinforcement learning (DRL) to scheduling problems in State Task Networks (STNs), demonstrating that hybrid agents equipped with recurrent neural networks can handle epistemic and aleatoric uncertainty in online scheduling environments. Adapting such methodologies to the crude oil scheduling context could be a promising direction. Additionally, it would be of interest to extend the current methodology by incorporating neural network models to assist in estimating crude oil properties. Although this thesis assumes such properties are given as input data, recent research shows that artificial neural networks (ANNs) can provide fast and accurate estimations of key crude properties, thereby enhancing the efficiency and reliability of scheduling and blending decisions.

To conclude, it is worth mentioning that this thesis led to the publication of one article in a Journal Citation Reports (JCR)-indexed journal [42], four peer-reviewed papers submitted to major international conferences in the field ([43], [44], [45], [46]), and six contributions to national conferences ([47], [48], [49], [50], [51], [39]).

Appendix A

Notation and constraints of the two-stage stochastic programming model

This appendix includes the nomenclature and constraints of the two-stage stochastic programming model presented in Chapter 6. Most of them are the same or very similar to those in the deterministic model introduced in Chapter 3. However, they are restated here for the reader's convenience.

One point that is interesting to mention concerns the way precedence is handled. In this model, the precedence between vessels is not subject to the order of the elements of the set "Vessels" as stated in [19], and there is no pre-allocation of time slots for each vessel as in [12]. Here, the "pre-defined precedence" concept has been applied, which means that the set of slots is pre-ordered, and the optimization algorithm allocates each vessel to some of these slots [52].

A.1 Notation

A.1.1 Sets

- C = types of crude oils.
- E = scenarios.
- K = key components or properties of crude oil.
- Q = tanks.
- S = time slots.
- U = crude distillation units.
- V = vessels.

- VC = vessel-crude pairs. This set indicates which type of crude oil c is transported by each vessel v, i.e., its elements are pairs (v,c), where v is in V, and c is in C.

A.1.2 Parameters

- $AT_{v,e}$ = arrival time of vessel v under scenario e.
- $CAPQL_q$ = minimum inventory of crude mix in a tank.
- $CAPQU_q$ = maximum capacity of a tank.
- $CDMG_v$ = demurrage or sea waiting cost.
- CEP_u = cost due to overproduction concerning the demand of unit u.
- $CONC_{q,c}$ = initial crude c concentration in the tank q.
- CSP_u = cost due to underproduction concerning the demand of unit u.
- $CTDN_v$ = departure tardiness cost.
- DH = length of the scheduling horizon.
- DEM_u = total demand of blended crude for CDU u.
- $EDT_{v,e}$ = departure time of vessel v under scenario e.
- $FRUL_u$ = minimum inlet flow to CDU u.
- $FRUU_u$ = maximum inlet flow to CDU u.
- $FRVL_v$ = minimum rate of crude transfer from vessel v.
- $FRVU_v$ = maximum rate of crude transfer from vessel v.
- H = end-time of the scheduling horizon.
- $LFRQL_q$ = minimum tank loading flow rate.
- $LFRQU_q$ = maximum tank loading flow rate.
- NQU = maximum number of tanks that can feed a CDU.
- NQV = maximum number of tanks that can be loaded.
- NU = maximum number of CDUs that can be loaded simultaneously by a tank.
- $PRC_{c,k}$ = volumetric concentration of the key component k in the crude type c.

- $PRL_{u,k}$ = minimum allowed concentration of key component k in the feedstock of CDU u.
- $PRU_{u,k}$ = maximum allowed concentration of key component k in the feedstock of CDU u.
- $SETT$ = time to settle and remove the brine.
- $SLIC_{q,c}$ = initial amount of crude c in the tank q.
- $UFRQL_q$ = minimum rate of crude transfer from tank q.
- $UFRQU_q$ = maximum rate of crude transfer from tank q.
- $VOL_{v,c}$ = amount of crude c in the vessel v.
- α = significance level.
- π_e = scenario e probability.

A.1.3 Continuous variables

The domain of continuous variables is the set of non-negative real numbers, except for the variables ze_e (cost per scenario) and $cvar$ (Conditional Value-at-Risk), whose domain is the set of all real numbers since they represent the objective functions.

- $cvar$ = Conditional Value-at-Risk.
- $dmg_{v,e}$ = demurrage of vessel v under scenario e.
- $dmg1_{v,s,e}$ = auxiliary variable to calculate $dmg_{v,e}$.
- ds_s = length of slot s.
- $exprod_u$ = overproduction concerning the demand of unit u.
- is_s = start-time of slot s.
- $shprod_u$ = underproduction concerning the demand of unit u.
- $stock_{q,s,e}$ = total level in q at the beginning of s.
- $stockc_{q,c,s,e}$ = amount of c in q at the beginning of s.
- $stockcend_{q,c,s,e}$ = amount of c in q at the end of the horizon.
- $stockend_{q,s,e}$ = total level in q at the end of the scheduling horizon.
- $tdn_{v,e}$ = departure tardiness of vessel v under scenario e.
- $tdn1_{v,s,e}$ = auxiliary variable to calculate $tdn_{v,e}$.

- ts_s = end-time of slot s.
- var = Value-at-Risk.
- $vcqu_{c,q,u,s,e}$ = amount of crude c transferred from q to u during s under scenario e.
- $vcv_{c,v,s,e}$ = total amount of crude c unloaded from v during s.
- $vcvq_{c,v,q,s,e}$ = amount of crude c transferred from v to q during s.
- $vqu_{q,u,s}$ = amount of crude mix transferred from q to u during s.
- $vu_{u,s}$ = total amount of crude mix transferred to u during s.
- $vvq_{v,q,s,e}$ = amount of crude transferred from v to q during s.
- ze_e = cost associated with scenario e.
- ϕ_e = auxiliary variable to assess the CVaR.

A.1.4 Binary variables

- $fuv_{v,s,e}$ = is equal to 1 if vessel v finishes its unloading at the end of s under scenario e; 0 otherwise.
- $lq_{q,s,e}$ = is equal to 1 if tank q is receiving crude during s under scenario e.
- $suv_{v,s,e}$ = is equal to 1 if vessel v starts its unloading at the beginning of s under scenario e; 0 otherwise.
- $uq_{q,s}$ = is equal to 1 if tank q is delivering crude during slot s.
- $uv_{v,s,e}$ = is equal to 1 if vessel v is unloading during s under scenario e; 0 otherwise.
- $uqu_{q,u,s}$ = is equal to 1 if tank q feeds CDU u during slot s.

A.2 Constraints

Vessel unloading and storage tank loading

A vessel is unloaded during a slot s ($uv_{v,s,e}$) if it was unloading during the previous slot ($uv_{v,s-1,e}$) and has not finished yet ($fuv_{v,s-1,e}$), or if it starts at the beginning of the current slot ($suv_{v,s,e}$) (a.1). Note that it applies to every scenario as it involves second-stage variables.

$$uv_{v,s,e} = uv_{v,s-1,e} + suv_{v,s,e} - fuv_{v,s-1,e} \quad \forall v \in V,\ \forall s \in S,\ \forall e \in E \tag{a.1}$$

A vessel can finish unloading at the end of a slot as long as it was unloading during that slot (a.2). Note that if the ship is not being unloaded ($uv_{v,s,e} = 0$), then $fuv_{v,s,e}$ will be equal to 0.

$$uv_{v,s,e} \geq fuv_{v,s,e} \quad \forall v \in V,\ \forall s \in S,\ \forall e \in E \tag{a.2}$$

All ships may start and end unloading only once within the scheduling horizon, (a.3) and (a.4) respectively.

$$\sum_{s \in S} suv_{v,s,e} \ = \ 1 \quad \forall v \in V,\ \forall e \in E \tag{a.3}$$

$$\sum_{s \in S} fuv_{v,s,e} \ = \ 1 \quad \forall v \in V,\ \forall e \in E \tag{a.4}$$

Only one vessel can unload at any moment (assumption 1).

$$\sum_{v \in V} uv_{v,s,e} \ \leq \ 1 \quad \forall s \in S,\ \forall e \in E \tag{a.5}$$

A maximum of NQV tanks can be loaded simultaneously (assumption 5). That is, the sum of the binary variable indicating that a tank is being loaded ($lq_{q,s,e}$), over all tanks, must be less than or equal to NQV.

$$\sum_{q \in Q} lq_{q,s,e} \ \leq \ NQV \quad \forall s \in S,\ \forall e \in E \tag{a.6}$$

When unloading crude oil from a vessel, it is necessary to have at least one tank being loaded. Note that if no tank is receiving a load, then $\sum_{q \in Q} lq_{q,s,e}$ equals zero.

$$\sum_{q \in Q} lq_{q,s,e} \ \geq \ uv_{v,s,e} \quad \forall v \in V,\ \forall s \in S,\ \forall e \in E \tag{a.7}$$

A tank cannot be loaded if there is no ship unloading.

$$lq_{q,s,e} \ \leq \ \sum_{v \in V} uv_{v,s,e} \quad \forall q \in Q,\ \forall s \in S,\ \forall e \in E \tag{a.8}$$

A tank may not charge more than NU crude distillation units simultaneously (assumption 6). Therefore, the sum of the variable indicating that a tank is feeding a unit ($uqu_{q,u,s}$), over all units, must be less than or equal to NU at all times.

$$\sum_{u \in U} uqu_{q,u,s} \ \leq \ NU \quad \forall q \in Q,\ \forall s \in S \tag{a.9}$$

At most NQU tanks are allowed to concurrently feed a CDU (assumption 7).

$$\sum_{q \in Q} uqu_{q,u,s} \ \leq \ NQU \quad \forall u \in U,\ \forall s \in S \tag{a.10}$$

Each CDU must continually process feedstock coming from tanks (assumption 9). This means that each CDU must be fed by at least one tank at all times.

$$\sum_{q \in Q} uqu_{q,u,s} \geq 1 \quad \forall u \in U,\ \forall s \in S \tag{a.11}$$

A tank must be in one of the three states (i.e., loading, unloading, or idle) during a given slot.

$$lq_{q,s,e} + uq_{q,s} \leq 1 \quad \forall q \in Q,\ \forall s \in S,\ \forall e \in E \tag{a.12}$$

A tank must be discharging ($uq_{q,s}$) if it is feeding a CDU ($uqu_{q,u,s}$).

$$uq_{q,s} \geq uqu_{q,u,s} \quad \forall q \in Q,\ \forall u \in U,\ \forall s \in S \tag{a.13}$$

A tank must be feeding at least one unit ($\sum_{u \in U} uqu_{q,u,s}$) if it is being unloaded ($uq_{q,s}$).

$$\sum_{u \in U} uqu_{q,u,s} \geq uq_{q,s} \quad \forall q \in Q,\ \forall s \in S \tag{a.14}$$

The end-time of a slot is equal to its start-time plus its length.

$$ts_s = is_s + ds_s \quad \forall s \in S \tag{a.15}$$

The start-time of a slot coincides with the end-time of the previous slot. This implies that the durations of operations of all resources are synchronized in each slot.

$$is_s = ts_{s-1} \quad \forall s \in S \mid s > 1 \tag{a.16}$$

The total length of the time slots must be equal to the length of the scheduling horizon.

$$\sum_{s \in S} ds_s = DH \tag{a.17}$$

The big-M method, explained by Winston and Goldberg [37], is applied to compute the amount of crude unloaded to tanks. The M values are determined based on physical limits. For example, M1 takes into account the volume carried by each vessel.

We calculate the upper bound of the volume of crude oil unloaded from a ship to a tank ($vcvq_{c,v,q,s,e}$) as a function of the maximum flow rate that the receiving tank admits ($LFRQU_q$) and the duration of the operation.

$$\begin{aligned} &vcvq_{c,v,q,s,e} \leq LFRQU_q \cdot ds_s \\ &\forall q \in Q,\ \forall (v,c) \in VC,\ \forall s \in S,\ \forall e \in E \end{aligned} \tag{a.18}$$

We calculate the lower bound of the volume of crude oil unloaded from a ship to a tank ($vcvq_{c,v,q,s,e}$) as a function of the minimum flow rate the tank admits ($LFRQL_q$) and the duration of the operation. Note that this constraint is activated only if ship v is unloading ($uv_{v,s,e} = 1$) and tank q is receiving a load ($lq_{q,s,e} = 1$).

$$\begin{aligned} & vcvq_{c,v,q,s,e} \geq LFRQL_q \cdot ds_s - M1_v \cdot (2 - uv_{v,s,e} - lq_{q,s,e}) \\ & \forall q \in Q, \ \forall (v,c) \in VC, \ \forall s \in S, \ \forall e \in E \end{aligned} \tag{a.19}$$

If vessel v is not being unloaded, then $vcvq_{c,v,q,s,e}$ will be equal to zero.

$$\begin{aligned} & vcvq_{c,v,q,s,e} \leq VOL_{v,c} \cdot uv_{v,s,e} \\ & \forall q \in Q, \ \forall (v,c) \in VC, \ \forall s \in S, \ \forall e \in E \end{aligned} \tag{a.20}$$

If tank q is not being loaded, then $vcvq_{c,v,q,s,e}$ will be equal to zero.

$$\begin{aligned} & vcvq_{c,v,q,s,e} \leq VOL_{v,c} \cdot lq_{q,s,e} \\ & \forall q \in Q, \ \forall (v,c) \in VC, \ \forall s \in S, \ \forall e \in E \end{aligned} \tag{a.21}$$

Also, we use the big-M method to calculate the crude volume unloaded from a vessel during a slot s ($vcv_{c,v,s,e}$). The upper and lower bounds are obtained by (a.22) and (a.23), respectively.

$$vcv_{c,v,s,e} \leq FRVU_v \cdot ds_s \quad \forall (v,c) \in VC, \ \forall s \in S, \ \forall e \in E \tag{a.22}$$

$$\begin{aligned} & vcv_{c,v,s,e} \geq FRVL_v \cdot ds_s - M1_v \cdot (1 - uv_{v,s,e}) \\ & \forall (v,c) \in VC, \ \forall s \in S, \ \forall e \in E \end{aligned} \tag{a.23}$$

If ship v is not being unloaded, then $vcv_{c,v,s,e}$ will be equal to zero.

$$vcv_{c,v,s,e} \leq VOL_{v,c} \cdot uv_{v,s,e} \quad \forall (v,c) \in VC, \ \forall s \in S, \ \forall e \in E \tag{a.24}$$

The crude volume unloaded from a vessel during a slot s is equal to the sum of volumes unloaded to each tank.

$$vcv_{c,v,s,e} = \sum_{q \in Q} vcvq_{c,v,q,s,e} \quad \forall (v,c) \in VC, \ \forall s \in S, \ \forall e \in E \tag{a.25}$$

The total volume loaded into a tank during a slot s is calculated by using (a.26).

$$vvq_{v,q,s,e} = \sum_{\substack{c \in C \\ (v,c) \in VC}} vcvq_{c,v,q,s,e} \quad \forall v \in V, \ \forall q \in Q, \ \forall s \in S, \ \forall e \in E \tag{a.26}$$

To make each vessel unload fully during the scheduling horizon (assumption 2), we use (a.27).

$$\sum_{s \in S} vcv_{c,v,s,e} = VOL_{v,c} \quad \forall (v,c) \in VC, \ \forall e \in E \tag{a.27}$$

Storage tank unloading and CDU feeding

The big-M method is applied to compute the amount of crude unloaded from tanks (a.28)-(a.30). Constraint (a.28) establishes the upper bound of the amount of crude oil discharged from tank q to unit u ($vqu_{q,u,s}$) as a function of the maximum discharge flow rate from the tank ($UFRQU_q$) and the duration of the operation.

$$vqu_{q,u,s} \le UFRQU_q \cdot ds_s \quad \forall q \in Q, \ \forall u \in U, \ \forall s \in S \tag{a.28}$$

We compute the lower bound as a function of the minimum discharge flow rate of tank q ($UFRQL_q$) and the duration of the operation. This constraint will be activated if tank q is feeding unit u ($uqu_{q,u,s} = 1$).

$$\begin{aligned} vqu_{q,u,s} \ &\ge \ UFRQL_q \cdot ds_s - \ M2_q \cdot (1 \ - \ uqu_{q,u,s}) \\ \forall q \in Q, \ &\forall u \in U, \ \forall s \in S \end{aligned} \tag{a.29}$$

If tank q is not feeding unit u, then $vqu_{q,u,s}$ will be equal to zero.

$$vqu_{q,u,s} \le CAPQU_q \cdot uqu_{q,u,s} \quad \forall q \in Q, \ \forall u \in U, \ \forall s \in S \tag{a.30}$$

The total volume unloaded from a tank during a slot s is calculated by using (a.31). It should be noted that this total volume does not depend on the scenarios as it is a first-stage variable. However, its composition does, as the inventory profile in each tank may be different between scenarios due to receiving crude from ships at different times.

$$vqu_{q,u,s} = \sum_{c \in C} vcqu_{c,q,u,s,e} \quad \forall q \in Q, \ \forall u \in U, \ \forall s \in S, \ \forall e \in E \tag{a.31}$$

The total feed to CDU u during slot s ($vu_{u,s}$) is equal to the sum of the volumes transferred from each tank ($vqu_{q,u,s}$).

$$vu_{u,s} = \sum_{q \in Q} vqu_{q,u,s} \quad \forall u \in U, \ \forall s \in S \tag{a.32}$$

Constraints (a.33) and (a.34) set the upper and lower limits for $vu_{u,s}$, respectively

$$vu_{u,s} \le FRUU_u \cdot ds_s \quad \forall u \in U, \ \forall s \in S \tag{a.33}$$

$$vu_{u,s} \ge FRUL_u \cdot ds_s \quad \forall u \in U, \ \forall s \in S \tag{a.34}$$

Key properties computation

The concentration of key components in the feedstock for the CDUs is given by (a.35)-(a.36). Constraint (a.35) sets the upper bound as the multiplication between the maximum allowed concentration of key component k in the feedstock of CDU u ($PRU_{u,k}$) and the total volume received by u ($vu_{u,s}$). Similarly, constraint (a.36) establishes the lower bound. Note that, in both cases, the variable $vu_{u,s}$ is on the right-hand side of the inequality to avoid nonlinear constraints.

$$\sum_{q \in Q} \sum_{c \in C} vcqu_{c,q,u,s,e} \cdot PRC_{c,k} \leq PRU_{u,k} \cdot vu_{u,s} \quad \forall k \in K,\ \forall u \in U,\ \forall s \in S,\ \forall e \in E \tag{a.35}$$

$$\sum_{q \in Q} \sum_{c \in C} vcqu_{c,q,u,s,e} \cdot PRC_{c,k} \geq PRL_{u,k} \cdot vu_{u,s} \quad \forall k \in K,\ \forall u \in U,\ \forall s \in S,\ \forall e \in E \tag{a.36}$$

Storage tank inventory level

The amount of crude c in each tank at the start of slot s ($stockc_{q,c,s,e}$) is calculated as the amount of crude oil c at the beginning of the previous slot ($stockc_{q,c,s-1,e}$), plus the load of crude oil c received during the previous slot ($vcvq_{c,v,q,s-1,e}$), minus the volume of crude oil c discharged from tank q to the units during the previous slot ($vcqu_{c,q,u,s-1,e}$).

$$stockc_{q,c,s,e} = stockc_{q,c,s-1,e} + \sum_{\substack{v \in V \\ (v,c) \in VC}} vcvq_{c,v,q,s-1,e} - \sum_{u \in U} vcqu_{c,q,u,s-1,e} \quad \forall q \in Q,\ \forall c \in C,\ \forall s \in S \mid s > 1,\ \forall e \in E \tag{a.37}$$

The amount of crude c in each tank at the beginning of the horizon is given by (a.38).

$$stockc_{q,c,s,e} = SLIC_{q,c} \quad \forall q \in Q,\ \forall c \in C,\ s = 1,\ \forall e \in E \tag{a.38}$$

The amount of crude c in each tank at the end of the horizon ($stockcend_{q,c,s,e}$) is calculated as the amount of crude oil c at the beginning of the last slot of the horizon ($stockc_{q,c,s,e}$), plus the crude oil load c received during the final slot ($vcvq_{c,v,q,s,e}$), minus the volume of crude oil c discharged from tank q to the units during that slot ($vcqu_{c,q,u,s,e}$).

$$stockcend_{q,c,s,e} = stockc_{q,c,s,e} + \sum_{\substack{v \in V \\ (v,c) \in VC}} vcvq_{c,v,q,s,e} - \sum_{u \in U} vcqu_{c,q,u,s,e} \quad \forall q \in Q,\ \forall c \in C,\ s = |S|,\ \forall e \in E \tag{a.39}$$

The total level in each tank at the start of slot s ($stock_{q,s,e}$) and at the end of the horizon ($stockend_{q,s,e}$) is given by (a.40)-(a.43).

$$stock_{q,s,e} = stock_{q,s-1,e} + \sum_{v \in V} vvq_{v,q,s-1,e} - \sum_{u \in U} vqu_{q,u,s-1} \quad \forall q \in Q,\ \forall s \in S \mid s > 1,\ \forall e \in E \tag{a.40}$$

$$stockend_{q,s,e} = stock_{q,s,e} + \sum_{v \in V} vvq_{v,q,s,e} - \sum_{u \in U} vqu_{q,u,s}$$
$$\forall q \in Q,\ s = |S|,\ \forall e \in E \qquad \text{(a.41)}$$

Moreover, both at the beginning of each slot (a.42) and at the end of the horizon (a.43), the total level in a tank q is equal to the sum of the volumes of each crude oil s stored in that tank.

$$stock_{q,s,e} = \sum_{c \in C} stockc_{q,c,s,e} \quad \forall q \in Q,\ \forall s \in S,\ \forall e \in E \qquad \text{(a.42)}$$

$$stockend_{q,s,e} = \sum_{c \in C} stockcend_{q,c,s,e} \quad \forall q \in Q,\ s = |S|,\ \forall e \in E \qquad \text{(a.43)}$$

Equations (a.44)-(a.47) establish the physical limits for the inventory levels.

$$stock_{q,s,e} \leq CAPQU_q \quad \forall q \in Q,\ \forall s \in S,\ \forall e \in E \qquad \text{(a.44)}$$

$$stock_{q,s,e} \geq CAPQL_q \quad \forall q \in Q,\ \forall s \in S,\ \forall e \in E \qquad \text{(a.45)}$$

$$stockend_{q,s,e} \leq CAPQU_q \quad \forall q \in Q,\ s = |S|,\ \forall e \in E \qquad \text{(a.46)}$$

$$stockend_{q,s,e} \geq CAPQL_q \quad \forall q \in Q,\ s = |S|,\ \forall e \in E \qquad \text{(a.47)}$$

To ensure minimum settling time (assumption 4), we use (a.48). If a tank q receives a charge during slot s ($lq_{q,s,e} = 1$) and is discharged during slot s' ($uq_{q,s'} = 1$), where s' is later than s, then the start time of slot s' must be greater than or equal to the end time of slot s, plus the time required to settle (SETT).

$$is_{s'} - ts_s \geq SETT \cdot (lq_{q,s,e} + uq_{q,s'} - 1)$$
$$\forall q \in Q,\ \forall s \in S,\ \forall s' \in S,\ s < s',\ \forall e \in E \qquad \text{(a.48)}$$

Demand fulfillment

To calculate the difference between processed volume and required demand by each CDU, we use (a.49)-(a.50). Constraint (a.49) sets the overproduction volume at unit u ($exprod_u$) as the difference between the volume processed over the horizon and the established demand. Constraint (a.50) computes the underproduction volume at unit u as the difference between the required demand and the volume processed over the horizon. Note that if the demand is not met, then $shprod_u$ will be greater than zero, and constraint (a.49) will be idle. Otherwise, if the demand value is exceeded, then $exprod_u$ will be greater than zero, and constraint (a.50) will be idle.

$$exprod_u \geq \sum_{s \in S} vu_{u,s} - dem_u \quad \forall u \in U \tag{a.49}$$

$$shprod_u \geq dem_u - \sum_{s \in S} vu_{u,s} \quad \forall u \in U \tag{a.50}$$

Vessel demurrage and tardiness

The discharge of crude oil from vessel v cannot start before its arrival time.

$$is_s \geq AT_{v,e} \cdot suv_{v,s,e} \quad \forall v \in V, \ \forall s \in S, \ \forall e \in E \tag{a.51}$$

The demurrage is calculated as the time elapsed between the arrival of a ship and the start of its unloading.

From constraints (a.52)-(a.54), the auxiliary variable $dmg1_{v,s,e}$ is calculated, which represents how many hours after its arrival a ship v has started unloading in slot s, for scenario e.

In case the ship has not started unloading at the beginning of slot s ($suv_{v,s,e} = 0$), then constraints (a.52) and (a.53) will be inactive, and the variable $dmg1_{v,s,e}$ be zero (a.54). Otherwise, if the ship starts unloading at the beginning of slot s ($suv_{v,s,e} = 1$), then constraints (a.52) and (a.53) are activated, and the value of $dmg1_{v,s,e}$ is computed.

$$dmg1_{v,s,e} \geq is_s - AT_{v,e} \cdot suv_{v,s,e} - H \cdot (1 - suv_{v,s,e}) \quad \forall v \in V, \ \forall s \in S, \ \forall e \in E \tag{a.52}$$

$$dmg1_{v,s,e} \leq is_s - AT_{v,e} \cdot suv_{v,s,e} \quad \forall v \in V, \ \forall s \in S, \ \forall e \in E \tag{a.53}$$

$$dmg1_{v,s,e} \leq H \cdot suv_{v,s,e} \quad \forall v \in V, \ \forall s \in S, \ \forall e \in E \tag{a.54}$$

From constraint (a.55), the demurrage of each ship in each scenario is calculated as the summation of the auxiliary variable $dmg1_{v,s,e}$ over all slots. Note that, at most, a single term of the summation will be greater than zero.

$$dmg_{v,e} = \sum_{s \in S} dmg1_{v,s,e} \quad \forall v \in V, \ \forall e \in E \tag{a.55}$$

If vessel v leaves the terminal after its expected departure time $EDT_{v,e}$, it should pay a penalty that will be proportional to the departure tardiness ($tdn_{v,e}$). Analogously to the calculation of the demurrage, the tardiness of each ship in each scenario ($tdn_{v,e}$) is computed from the auxiliary variable $tdn1_{v,s,e}$, using constraints (a.56)-(a.58).

$$tdn1_{v,s,e} \geq ts_s - EDT_{v,e} \cdot fuv_{v,s,e} - H \cdot (1 - fuv_{v,s,e}) \quad \forall v \in V,\ \forall s \in S,\ \forall e \in E \tag{a.56}$$

$$tdn1_{v,s,e} \leq H \cdot fuv_{v,s,e} \quad \forall v \in V,\ \forall s \in S,\ \forall e \in E \tag{a.57}$$

$$tdn_{v,e} = \sum_{s \in S} tdn1_{v,s,e} \quad \forall v \in V,\ \forall e \in E \tag{a.58}$$

If a tank is being discharged, then the crude oil concentration in the outflow must be equal to the concentration inside the tank. In other words, this principle states that the proportion of each crude in the volume transferred ($vcqu_{c,q,u,s,e}/vqu_{q,u,s}$) must be the same as the proportion of each crude in the volume stored ($stockc_{q,c,s,e}/stock_{q,s,e}$). This rule is satisfied by (a.59). It should be noted that this equation yields two bilinear terms which are non-convex.

$$stock_{q,s,e} \cdot vcqu_{c,q,u,s,e} = stockc_{q,c,s,e} \cdot vqu_{q,u,s} \quad \forall c \in C,\ \forall q \in Q,\ \forall u \in U,\ \forall s \in S,\ \forall e \in E \tag{a.59}$$

Appendix B

Linear approximation by planes for loading tank nonlinear constraint

The notation and constraints associated with the approximation of constraint (3.148) are presented below. Regarding the constraints, they include the same type and number of constraints as the example shown in Chapter 4. The only difference is that here, the constraints associated with both sides of the equation are presented, and a final constraint is included to equate these terms.

B.1 Sets

- I: partition associated with the variables $vcqq\_up_{c,ql,qr,s,i,j}$ and $vcqq\_low_{c,ql,qr,s,i,j}$.
- J: partition associated with the variables $stock\_ql\_up_{c,ql,qr,s,i,j}$ and $stock\_ql\_low_{c,ql,qr,s,i,j}$.
- N: partition associated with the variables $vqq\_up_{ql,qr,c,s,n,p}$ and $vqq\_low_{ql,qr,c,s,n,p}$.
- PP: partition associated with the variables $stockc\_ql\_up_{ql,qr,c,s,n,p}$ and $stockc\_ql\_low_{ql,qr,c,s,n,p}$.
- C: crude oils.
- QL: loading tanks.
- QR: refinery tanks.
- S: time slots.

B.2 Parameters

Below are the parameters associated with the left-hand side of the equation.

- $VCQQ\_MAX_{ql,i}$: upper bound associated with the variables $vcqq\_low_{c,ql,qr,s,i,j}$ and $vcqq\_up_{c,ql,qr,s,i,j}$, in partition i.
- $VCQQ\_MIN_{ql,i}$: lower bound associated with the variables $vcqq\_low_{c,ql,qr,s,i,j}$ and $vcqq\_up_{c,ql,qr,s,i,j}$, in partition i.
- $STOCKQL\_MAX_{ql,j}$: upper bound associated with the variables $stock\_ql\_low_{c,ql,qr,s,i,j}$ and $stock\_ql\_up_{c,ql,qr,s,i,j}$, in partition j.
- $STOCKQL\_MIN_{ql,j}$: lower bound associated with the variables $stock\_ql\_low_{c,ql,qr,s,i,j}$ and $stock\_ql\_up_{c,ql,qr,s,i,j}$, in partition j.

The parameters associated with the right-hand side of the equation are shown below.

- $VQQ\_MAX_{ql,n}$: upper bound associated with the variables $vqq\_low_{ql,qr,c,s,n,p}$ and $vqq\_up_{ql,qr,c,s,n,p}$, in partition n.
- $VQQ\_MIN_{ql,n}$: lower bound associated with the variables $vqq\_low_{ql,qr,c,s,n,p}$ and $vqq\_up_{ql,qr,c,s,n,p}$, in partition n.
- $STOCKCQL\_MAX_{ql,p}$: upper bound associated with the variables $stockc\_ql\_low_{ql,qr,c,s,n,p}$ and $stockc\_ql\_up_{ql,qr,c,s,n,p}$, in partition p.
- $STOCKCQL\_MIN_{ql,p}$: lower bound associated with the variables $stockc\_ql\_low_{ql,qr,c,s,n,p}$ and $stockc\_ql\_up_{ql,qr,c,s,n,p}$, in partition p.

Although certain parameters are associated with variables defined over the set of crude oils C, these parameters are not defined over C because the range of admissible values is the same for each crude, from zero to the maximum capacity of each tank. Furthermore, although this range applies to all parameters, except $STOCKQL\_MIN_{ql,j}$ and $STOCKQL\_MAX_{ql,j}$, they are defined as independent parameters to clearly differentiate the constraints.

For the parameters $STOCKQL\_MIN_{ql,j}$ and $STOCKQL\_MAX_{ql,j}$, an appropriate range could be from the minimum inventory level to the maximum capacity of each tank.

B.3 Variables

As with the parameters, the continuous variables associated with the left-hand side of the equation are shown.

- $vcqq\_up_{c,ql,qr,s,i,j}$: disaggregated variable of variable $vcqq_{c,ql,qr,s}$ related to the upper plane in partition $(i,\, j)$.

- $vcqq\_low_{c,ql,qr,s,i,j}$: disaggregated variable of variable $vcqq_{c,ql,qr,s}$ related to the lower plane in partition $(i,\ j)$.

- $stock\_ql\_up_{c,ql,qr,s,i,j}$: disaggregated variable of variable $stock_{ql,s}$ related to the upper plane in partition $(i,\ j)$.

- $stock\_ql\_low_{c,ql,qr,s,i,j}$: disaggregated variable of variable $stock_{ql,s}$ related to the lower plane in partition $(i,\ j)$.

- $zql_{c,ql,qr,s}$: variable representing the product of the variables $vcqq_{c,ql,qr,s}$ and $stock_{ql,s}$.

- $zql\_up_{c,ql,qr,s,i,j}$: disaggregated variable of variable $zql_{c,ql,qr,s}$ related to the upper plane in partition $(i,\ j)$.

- $zql\_low_{c,ql,qr,s,i,j}$: disaggregated variable of variable $zql_{c,ql,qr,s}$ related to the lower plane in partition $(i,\ j)$.

And the following are the continuous variables associated with the right-hand side of the equation.

- $vqq\_up_{ql,qr,c,s,n,p}$: disaggregated variable of variable $vqq_{ql,qr,s}$ related to the upper plane in partition $(n,\ p)$.

- $vqq\_low_{ql,qr,c,s,n,p}$: disaggregated variable of variable $vqq_{ql,qr,s}$ related to the lower plane in partition $(n,\ p)$.

- $stockc\_ql\_up_{ql,qr,c,s,n,p}$: disaggregated variable of variable $stockc_{ql,c,s}$ related to the upper plane in partition $(n,\ p)$.

- $stockc\_ql\_low_{ql,qr,c,s,n,p}$: disaggregated variable of variable $stockc_{ql,c,s}$ related to the lower plane in partition $(n,\ p)$.

- $oql_{ql,qr,c,s}$: variable representing the product of the variables $vqq_{ql,qr,s}$ and $stockc_{ql,c,s}$.

- $oql\_up_{ql,qr,c,s,n,p}$: disaggregated variable of variable $oql_{ql,qr,c,s}$ related to the upper plane in partition $(n,\ p)$.

- $oql\_low_{ql,qr,c,s,n,p}$: disaggregated variable of variable $oql_{ql,qr,c,s}$ related to the lower plane in partition $(n,\ p)$.

B.4 Binary Variables

Binary variables associated with the left-hand side of the equation.

- $wql\_up_{c,ql,qr,s,i,j}$: binary variable associated with the upper plane in partition $(i,\ j)$.

- $wql\_low_{c,ql,qr,s,i,j}$: binary variable associated with the lower plane in partition $(i,\, j)$.

Binary variables associated with the right-hand side of the equation:

- $vql\_up_{ql,qr,c,s,n,p}$: binary variable associated with the upper plane in partition $(n,\, p)$.
- $vql\_low_{ql,qr,c,s,n,p}$: binary variable associated with the lower plane in partition $(n,\, p)$.

B.5 Constraints

The constraints associated with the left-hand side of the equation are presented below.

$$vcqq_{c,ql,qr,s} = \sum_{i\in I}\sum_{j\in J}\left(vcqq\_up_{c,ql,qr,s,i,j} + vcqq\_low_{c,ql,qr,s,i,j}\right) \quad \forall c \in C, ql \in QL, qr \in QR, s \in S \tag{b.1}$$

$$stock_{ql,s} = \sum_{i\in I}\sum_{j\in J}\left(stock\_ql\_up_{c,ql,qr,s,i,j} + stock\_ql\_low_{c,ql,qr,s,i,j}\right) \quad \forall c \in C, ql \in QL, qr \in QR, s \in S \tag{b.2}$$

$$vcqq\_up_{c,ql,qr,s,i,j} \le VCQQ\_MAX_{ql,i} \cdot wql\_up_{c,ql,qr,s,i,j} \quad \forall c \in C, ql \in QL, qr \in QR, s \in S, i \in I, j \in J \tag{b.3}$$

$$vcqq\_up_{c,ql,qr,s,i,j} \ge VCQQ\_MIN_{ql,i} \cdot wql\_up_{c,ql,qr,s,i,j} \quad \forall c \in C, ql \in QL, qr \in QR, s \in S, i \in I, j \in J \tag{b.4}$$

$$vcqq\_low_{c,ql,qr,s,i,j} \le VCQQ\_MAX_{ql,i} \cdot wql\_low_{c,ql,qr,s,i,j} \quad \forall c \in C, ql \in QL, qr \in QR, s \in S, i \in I, j \in J \tag{b.5}$$

$$vcqq\_low_{c,ql,qr,s,i,j} \ge VCQQ\_MIN_{ql,i} \cdot wql\_low_{c,ql,qr,s,i,j} \quad \forall c \in C, ql \in QL, qr \in QR, s \in S, i \in I, j \in J \tag{b.6}$$

$$stock\_ql\_up_{c,ql,qr,s,i,j} \le STOCKQL\_MAX_{ql,j} \cdot wql\_up_{c,ql,qr,s,i,j} \quad \forall c \in C, ql \in QL, qr \in QR, s \in S, i \in I, j \in J \tag{b.7}$$

$$stock\_ql\_up_{c,ql,qr,s,i,j} \geq STOCKQL\_MIN_{ql,j} \cdot wql\_up_{c,ql,qr,s,i,j} \quad \forall c \in C, ql \in QL, qr \in QR, s \in S, i \in I, j \in J \tag{b.8}$$

$$stock\_ql\_low_{c,ql,qr,s,i,j} \leq STOCKQL\_MAX_{ql,j} \cdot wql\_low_{c,ql,qr,s,i,j} \quad \forall c \in C, ql \in QL, qr \in QR, s \in S, i \in I, j \in J \tag{b.9}$$

$$stock\_ql\_low_{c,ql,qr,s,i,j} \geq STOCKQL\_MIN_{ql,j} \cdot wql\_low_{c,ql,qr,s,i,j} \quad \forall c \in C, ql \in QL, qr \in QR, s \in S, i \in I, j \in J \tag{b.10}$$

$$zql_{c,ql,qr,s} = \sum_{i \in I} \sum_{j \in J} \left(zql\_up_{c,ql,qr,s,i,j} + zql\_low_{c,ql,qr,s,i,j} \right) \quad \forall c \in C, ql \in QL, qr \in QR, s \in S \tag{b.11}$$

$$\begin{aligned} zql\_up_{c,ql,qr,s,i,j} &= STOCKQL\_MAX_{ql,j} \cdot vcqq\_up_{c,ql,qr,s,i,j} \\ &+ VCQQ\_MIN_{ql,i} \cdot stock\_ql\_up_{c,ql,qr,s,i,j} \\ &- STOCKQL\_MAX_{ql,j} \cdot VCQQ\_MIN_{ql,i} \cdot wql\_up_{c,ql,qr,s,i,j} \\ &\forall c \in C, ql \in QL, qr \in QR, s \in S, i \in I, j \in J \end{aligned} \tag{b.12}$$

$$\begin{aligned} zql\_low_{c,ql,qr,s,i,j} &= STOCKQL\_MIN_{ql,j} \cdot vcqq\_low_{c,ql,qr,s,i,j} \\ &+ VCQQ\_MAX_{ql,i} \cdot stock\_ql\_low_{c,ql,qr,s,i,j} \\ &- STOCKQL\_MIN_{ql,j} \cdot VCQQ\_MAX_{ql,i} \cdot wql\_low_{c,ql,qr,s,i,j} \\ &\forall c \in C, ql \in QL, qr \in QR, s \in S, i \in I, j \in J \end{aligned} \tag{b.13}$$

$$\sum_{i \in I} \sum_{j \in J} \left(wql\_up_{c,ql,qr,s,i,j} + wql\_low_{c,ql,qr,s,i,j} \right) = 1 \quad \forall c \in C, ql \in QL, qr \in QR, s \in S \tag{b.14}$$

$$\begin{aligned} &(VCQQ\_MAX_{ql,i} - VCQQ\_MIN_{ql,i}) \cdot stock\_ql\_up_{c,ql,qr,s,i,j} \\ &- (STOCKQL\_MAX_{ql,j} - STOCKQL\_MIN_{ql,j}) \cdot vcqq\_up_{c,ql,qr,s,i,j} \\ &\geq (VCQQ\_MIN_{ql,i} \cdot STOCKQL\_MAX_{ql,j} \\ &- VCQQ\_MAX_{ql,i} \cdot STOCKQL\_MIN_{ql,j}) \cdot wql\_up_{c,ql,qr,s,i,j} \\ &\forall c \in C, ql \in QL, qr \in QR, s \in S, i \in I, j \in J \end{aligned} \tag{b.15}$$

$$(VCQQ\_MAX_{ql,i} - VCQQ\_MIN_{ql,i}) \cdot stock\_ql\_low_{c,ql,qr,s,i,j}$$
$$- (STOCKQL\_MAX_{ql,j} - STOCKQL\_MIN_{ql,j}) \cdot vcqq\_low_{c,ql,qr,s,i,j}$$
$$\leq (VCQQ\_MIN_{ql,i} \cdot STOCKQL\_MAX_{ql,j}$$
$$- VCQQ\_MAX_{ql,i} \cdot STOCKQL\_MIN_{ql,j}) \cdot wql\_low_{c,ql,qr,s,i,j}$$
$$\forall c \in C, ql \in QL, qr \in QR, s \in S, i \in I, j \in J \tag{b.16}$$

The constraints associated with the right-hand side of the equation are presented below.

$$vqq_{ql,qr,s} = \sum_{n \in N} \sum_{p \in PP} \left(vqq\_up_{ql,qr,c,s,n,p} + vqq\_low_{ql,qr,c,s,n,p} \right)$$
$$\forall ql \in QL, qr \in QR, c \in C, s \in S \tag{b.17}$$

$$stockc_{ql,c,s} = \sum_{n \in N} \sum_{p \in PP} \left(stockc\_ql\_up_{ql,qr,c,s,n,p} + stockc\_ql\_low_{ql,qr,c,s,n,p} \right)$$
$$\forall ql \in QL, qr \in QR, c \in C, s \in S \tag{b.18}$$

$$vqq\_up_{ql,qr,c,s,n,p} \leq VQQ\_MAX_{ql,n} \cdot vql\_up_{ql,qr,c,s,n,p}$$
$$\forall ql \in QL, qr \in QR, c \in C, s \in S, n \in N, p \in PP \tag{b.19}$$

$$vqq\_up_{ql,qr,c,s,n,p} \geq VQQ\_MIN_{ql,n} \cdot vql\_up_{ql,qr,c,s,n,p}$$
$$\forall ql \in QL, qr \in QR, c \in C, s \in S, n \in N, p \in PP \tag{b.20}$$

$$vqq\_low_{ql,qr,c,s,n,p} \leq VQQ\_MAX_{ql,n} \cdot vql\_low_{ql,qr,c,s,n,p}$$
$$\forall ql \in QL, qr \in QR, c \in C, s \in S, n \in N, p \in PP \tag{b.21}$$

$$vqq\_low_{ql,qr,c,s,n,p} \geq VQQ\_MIN_{ql,n} \cdot vql\_low_{ql,qr,c,s,n,p}$$
$$\forall ql \in QL, qr \in QR, c \in C, s \in S, n \in N, p \in PP \tag{b.22}$$

$$stockc\_ql\_up_{ql,qr,c,s,n,p} \leq STOCKCQL\_MAX_{ql,p} \cdot vql\_up_{ql,qr,c,s,n,p}$$
$$\forall ql \in QL, qr \in QR, c \in C, s \in S, n \in N, p \in PP \tag{b.23}$$

$$stockc\_ql\_up_{ql,qr,c,s,n,p} \geq STOCKCQL\_MIN_{ql,p} \cdot vql\_up_{ql,qr,c,s,n,p}$$
$$\forall ql \in QL, qr \in QR, c \in C, s \in S, n \in N, p \in PP \tag{b.24}$$

$$stockc\_ql\_low_{ql,qr,c,s,n,p} \leq STOCKCQL\_MAX_{ql,p} \cdot vql\_low_{ql,qr,c,s,n,p}$$
$$\forall ql \in QL, qr \in QR, c \in C, s \in S, n \in N, p \in PP \tag{b.25}$$

$$stockc\_ql\_low_{ql,qr,c,s,n,p} \geq STOCKCQL\_MIN_{ql,p} \cdot vql\_low_{ql,qr,c,s,n,p}$$
$$\forall ql \in QL, qr \in QR, c \in C, s \in S, n \in N, p \in PP \tag{b.26}$$

$$oql_{ql,qr,c,s} = \sum_{n \in N} \sum_{p \in PP} \left(oql\_up_{ql,qr,c,s,n,p} + oql\_low_{ql,qr,c,s,n,p} \right)$$
$$\forall ql \in QL, qr \in QR, c \in C, s \in S \tag{b.27}$$

$$oql\_up_{ql,qr,c,s,n,p} = STOCKCQL\_MAX_{ql,p} \cdot vqq\_up_{ql,qr,c,s,n,p}$$
$$+ VQQ\_MIN_{ql,n} \cdot stockc\_ql\_up_{ql,qr,c,s,n,p}$$
$$- STOCKCQL\_MAX_{ql,p} \cdot VQQ\_MIN_{ql,n} \cdot vql\_up_{ql,qr,c,s,n,p}$$
$$\forall ql \in QL, qr \in QR, c \in C, s \in S, n \in N, p \in PP \tag{b.28}$$

$$oql\_low_{ql,qr,c,s,n,p} = STOCKCQL\_MIN_{ql,p} \cdot vqq\_low_{ql,qr,c,s,n,p}$$
$$+ VQQ\_MAX_{ql,n} \cdot stockc\_ql\_low_{ql,qr,c,s,n,p}$$
$$- STOCKCQL\_MIN_{ql,p} \cdot VQQ\_MAX_{ql,n} \cdot vql\_low_{ql,qr,c,s,n,p}$$
$$\forall ql \in QL, qr \in QR, c \in C, s \in S, n \in N, p \in PP \tag{b.29}$$

$$\sum_{n \in N} \sum_{p \in PP} \left(vql\_up_{ql,qr,c,s,n,p} + vql\_low_{ql,qr,c,s,n,p} \right) = 1$$
$$\forall ql \in QL, qr \in QR, c \in C, s \in S \tag{b.30}$$

$$(VQQ\_MAX_{ql,n} - VQQ\_MIN_{ql,n}) \cdot stockc\_ql\_up_{ql,qr,c,s,n,p}$$
$$- (STOCKCQL\_MAX_{ql,p} - STOCKCQL\_MIN_{ql,p}) \cdot vqq\_up_{ql,qr,c,s,n,p}$$
$$\geq (VQQ\_MIN_{ql,n} \cdot STOCKCQL\_MAX_{ql,p}$$
$$- VQQ\_MAX_{ql,n} \cdot STOCKCQL\_MIN_{ql,p}) \cdot vql\_up_{ql,qr,c,s,n,p}$$
$$\forall ql \in QL, qr \in QR, c \in C, s \in S, n \in N, p \in PP \tag{b.31}$$

$$(VQQ\_MAX_{ql,n} - VQQ\_MIN_{ql,n}) \cdot stockc\_ql\_low_{ql,qr,c,s,n,p}$$
$$- (STOCKCQL\_MAX_{ql,p} - STOCKCQL\_MIN_{ql,p}) \cdot vqq\_low_{ql,qr,c,s,n,p}$$
$$\leq (VQQ\_MIN_{ql,n} \cdot STOCKCQL\_MAX_{ql,p}$$
$$- VQQ\_MAX_{ql,n} \cdot STOCKCQL\_MIN_{ql,p}) \cdot vql\_low_{ql,qr,c,s,n,p}$$
$$\forall ql \in QL, qr \in QR, c \in C, s \in S, n \in N, p \in PP \tag{b.32}$$

Constraint (b.33) equates the left-hand side of the equation with the right-hand side.

$$zql_{c,ql,qr,s} = oql_{ql,qr,c,s} \quad \forall ql \in QL, qr \in QR, c \in C, s \in S \tag{b.33}$$

Bibliography

[1] H. Yang, D. E. Bernal, R. E. Franzoi, F. G. Engineer, K. Kwon, S. Lee, I. E. Grossmann, Integration of crude-oil scheduling and refinery planning by lagrangean decomposition, Computers and Chemical Engineering 138 (7 2020). doi:10.1016/j.compchemeng.2020.106812.

[2] N. Q. Wu, L. P. Bai, M. C. Zhou, An efficient scheduling method for crude oil operations in refinery with crude oil type mixing requirements, IEEE Transactions on Systems, Man, and Cybernetics: Systems 46 (2016) 413–426. doi:10.1109/TSMC.2014.2332138.

[3] R. Saigal, Linear Programming: A Modern Integrated Analysis, International Series in Operations Research & Management Science, Springer US, 2012.

[4] L. Wolsey, Integer Programming, Wiley Series in Discrete Mathematics and Optimization, Wiley, 1998.

[5] L. T. Biegler, Nonlinear Programming, Society for Industrial and Applied Mathematics, 2010. doi:10.1137/1.9780898719383.

[6] I. E. Grossmann, Advanced Optimization for Process Systems Engineering, Cambridge Series in Chemical Engineering, Cambridge University Press, 2021.

[7] C. A. Méndez, J. Cerdá, I. E. Grossmann, I. Harjunkoski, M. Fahl, State-of-the-art review of optimization methods for short-term scheduling of batch processes, Computers and Chemical Engineering 30 (2006) 913–946. doi:10.1016/j.compchemeng.2006.02.008.

[8] J. Birge, F. Louveaux, Introduction to Stochastic Programming, 2nd Edition, Springer New York, NY, 2011.

[9] H. Lee, J. M. Pinto, I. E. Grossmann, S. Park, Mixed-integer linear programming model for refinery short-term scheduling of crude oil unloading with inventory management, Industrial and Engineering Chemistry Research 35 (1996). doi:10.1021/ie950519h.

[10] Z. Jia, M. Ierapetritou, J. D. Kelly, Refinery short-term scheduling using continuous time formulation: Crude-oil operations, Industrial and Engineering Chemistry Research 42 (2003). doi:10.1021/ie020124f.

[11] E. Kondili, C. Pantelides, R. Sargent, A general algorithm for short-term scheduling of batch operations—i. milp formulation, Computers and Chemical Engineering 17 (2) (1993) 211–227, an International Journal of Computer Applications in Chemical Enginnering. doi:https://doi.org/10.1016/0098-1354(93)80015-F.

[12] P. C. P. Reddy, I. A. Karimi, R. Srinivasan, A new continuous-time formulation for scheduling crude oil operations, Chemical Engineering Science 59 (2004). doi:10.1016/j.ces.2004.01.009.

[13] K. C. Furman, Z. Jia, M. G. Ierapetritou, A robust event-based continuous time formulation for tank transfer scheduling, Vol. 46, 2007. doi:10.1021/ie061516f.

[14] S. Mouret, I. E. Grossmann, P. Pestiaux, A novel priority-slot based continuous-time formulation for crude-oil scheduling problems, Industrial and Engineering Chemistry Research 48 (2009) 8515–8528. doi:10.1021/ie8019592.

[15] J. Li, R. Misener, C. A. Floudas, Continuous-time modeling and global optimization approach for scheduling of crude oil operations, AIChE Journal 58 (2012) 205–226. doi:10.1002/aic.12623.

[16] S. Yadav, M. A. Shaik, Short-term scheduling of refinery crude oil operations, Industrial and Engineering Chemistry Research 51 (2012) 9287–9299. doi:10.1021/ie300046g.

[17] A. A. Hamisu, S. Kabantiok, M. Wang, Refinery scheduling of crude oil unloading with tank inventory management, Computers and Chemical Engineering 55 (2013) 134–147. doi:10.1016/j.compchemeng.2013.04.003.

[18] P. M. Castro, I. E. Grossmann, Global optimal scheduling of crude oil blending operations with rtn continuous-time and multiparametric disaggregation, Industrial and Engineering Chemistry Research 53 (2014) 15127–15145. doi:10.1021/ie503002k.

[19] J. Cerdá, P. C. Pautasso, D. C. Cafaro, Efficient approach for scheduling crude oil operations in marine- access refineries, Industrial and Engineering Chemistry Research 54 (2015). doi:10.1021/acs.iecr.5b01461.

[20] B. Zimberg, E. Camponogara, E. Ferreira, Reception, mixture, and transfer in a crude oil terminal, Computers and Chemical Engineering 82 (2015) 293–302. doi:10.1016/j.compchemeng.2015.07.012.

[21] L. S. de Assis, E. Camponogara, B. Zimberg, E. Ferreira, I. E. Grossmann, A piecewise mccormick relaxation-based strategy for scheduling operations in a crude oil terminal, Computers and Chemical Engineering 106 (2017) 309–321. doi:10.1016/j.compchemeng.2017.06.012.

[22] L. S. Assis, E. Camponogara, B. C. Menezes, I. E. Grossmann, An minlp formulation for integrating the operational management of crude oil supply, Computers and Chemical Engineering 123 (2019) 110–125. doi:10.1016/j.compchemeng.2018.12.014.

[23] L. S. Assis, E. Camponogara, I. E. Grossmann, A milp-based clustering strategy for integrating the operational management of crude oil supply, Computers and Chemical Engineering 145 (2 2021). doi:10.1016/j.compchemeng.2020.107161.

[24] S. Lagzi, D. Y. Lee, R. Fukasawa, L. Ricardez-Sandoval, A computational study of continuous and discrete time formulations for a class of short-term scheduling problems for multipurpose plants, Industrial & Engineering Chemistry Research 56 (31) (2017) 8940–8953. arXiv:https://doi.org/10.1021/acs.iecr.7b01718, doi:10.1021/acs.iecr.7b01718.
URL https://doi.org/10.1021/acs.iecr.7b01718

[25] B. P. Patil, R. Fukasawa, L. A. Ricardez-Sandoval, Scheduling of operations in a large-scale scientific services facility via multicommodity flow and an optimization-based algorithm, Ind. Eng. Chem. Res. 54 (2015) 1628.

[26] S. Velez, C. T. Maravelias, Multiple and nonuniform time grids in discrete-time mip models for chemical production scheduling, Comput. Chem. Eng. 53 (2013) 70.

[27] K. G. Menon, R. Fukasawa, L. A. Ricardez-Sandoval, Integration of planning and scheduling for large-scale multijob multitasking batch plants, Industrial & Engineering Chemistry Research 63 (2) (2024) 1039–1054. arXiv:https://doi.org/10.1021/acs.iecr.3c02408, doi:10.1021/acs.iecr.3c02408.
URL https://doi.org/10.1021/acs.iecr.3c02408

[28] C. Zhang, P. Ricoux, Monthly schedule of crude refinery: Multi-scale strategy and multi-criteria objective, Computers and Chemical Engineering 164 (8 2022). doi:10.1016/j.compchemeng.2022.107866.

[29] J. Wang, G. Rong, Robust optimization model for crude oil scheduling under uncertainty, Industrial and Engineering Chemistry Research 49 (2010) 1737–1748. doi:10.1021/ie900358z.

[30] C. Cao, X. Gu, Z. Xin, Stochastic chance constrained mixed-integer nonlinear programming models and the solution approaches for refinery short-term crude oil scheduling problem, Applied Mathematical Modelling 34 (2010) 3231–3243. doi:10.1016/j.apm.2010.02.015.

[31] J. Li, R. Misener, C. A. Floudas, Scheduling of crude oil operations under demand uncertainty: A robust optimization framework coupled with global optimization, AIChE Journal 58 (2012) 2373–2396. doi:10.1002/aic.12772.

[32] F. Oliveira, P. M. Nunes, R. Blajberg, S. Hamacher, A framework for crude oil scheduling in an integrated terminal-refinery system under supply uncertainty, European Journal of Operational Research 252 (2016) 635–645. doi:10.1016/j.ejor.2016.01.034.

[33] K. G. Menon, R. Fukasawa, L. A. Ricardez-Sandoval, A novel stochastic programming approach for scheduling of batch processes with decision dependent time of uncertainty realization, Annals of Operations Research 305 (1-2) (2021) 163–190. doi:10.1007/s10479-021-04141-w. URL https://doi.org/10.1007/s10479-021-04141-w

[34] D. Rangel-Martinez, L. A. Ricardez-Sandoval, Recurrent reinforcement learning strategy with a parameterized agent for online scheduling of a state task network under uncertainty, Industrial & Engineering Chemistry Research 64 (13) (2025) 7126–7140. arXiv:https://doi.org/10.1021/acs.iecr.4c04900, doi:10.1021/acs.iecr.4c04900. URL https://doi.org/10.1021/acs.iecr.4c04900

[35] S. Alizadeh, S. Ta, L. Samavedham, A. K. Ray, Application of artificial neural network for prediction of 10 crude oil properties, The Canadian Journal of Chemical Engineering 101 (11) (2023) 6203–6214. doi:https://doi.org/10.1002/cjce.24938.

[36] S. Mouret, I. E. Grossmann, P. Pestiaux, Time representations and mathematical models for process scheduling problems, Computers and Chemical Engineering 35 (2011) 1038–1063. doi:10.1016/j.compchemeng.2010.07.007.

[37] W. L. Winston, J. B. Goldberg, Operations Research: Applications and Algorithms, 4th Edition, Thomson Brooks/Cole, 2004.

[38] E. Balas, Disjunctive programming and a hierarchy of relaxations for discrete optimization problems, SIAM Journal on Algebraic and Dis-

crete Methods 6 (3) (1985) 466–486. doi:10.1137/0606047.
URL https://doi.org/10.1137/0606047

[39] T. García García Verdier, G. Gutiérrez, C. Méndez, C. de Prada, New linear approximation method applied to crude operations scheduling, XLV Jornadas de Automática (2024). doi:https://doi.org/10.17979/ja-cea.2024.45.10942.

[40] M. Tejeda-Iglesias, N. H. Lappas, C. E. Gounaris, L. Ricardez-Sandoval, Explicit model predictive controller under uncertainty: An adjustable robust optimization approach, Journal of Process Control 84 (2019) 115–132. doi:10.1016/j.jprocont.2019.09.002.

[41] I. Grossmann, J. Viswanathan, A. Vecchietti, R. Raman, E. Kalvelagen, Gams/dicopt: A discrete continuous optimization package 11 (06 2002).

[42] T. G. Garcia-Verdier, G. Gutierrez, C. A. Méndez, C. G. Palacín, C. de Prada, Optimization of crude oil operations scheduling by applying a two-stage stochastic programming approach with risk management, Journal of Process Control 133 (2024) 103142. doi:https://doi.org/10.1016/j.jprocont.2023.103142.

[43] T. G. García-Verdier, G. Gutiérrez, C. Méndez, C. G. Palacín, C. de Prada, Minimizing risk in the scheduling of crudes in an oil refinery, IFAC-PapersOnLine 55 (7) (2022) 809–814, 13th IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems DYCOPS 2022. doi:https://doi.org/10.1016/j.ifacol.2022.07.544.
URL https://www.sciencedirect.com/science/article/pii/S2405896322009508

[44] T. G. García-Verdier, G. Gutiérrez, C. Méndez, C. G. Palacín, C. de Prada, Minimizing risk in the scheduling of crudes in an oil refinery, IFAC-PapersOnLine 55 (2022) 809–814. doi:10.1016/J.IFACOL.2022.07.544.
URL https://linkinghub.elsevier.com/retrieve/pii/S2405896322009508

[45] T. G. Garcia-Verdier, G. Gutierrez, C. Mendez, C. de Prada, Optimizing the monthly scheduling of crudes in a terminal-refinery system, IFAC-PapersOnLine 56 (2) (2023) 7414–7419, 22nd IFAC World Congress. doi:https://doi.org/10.1016/j.ifacol.2023.10.620.
URL https://www.sciencedirect.com/science/article/pii/S2405896323009916

[46] T. G. Garcia-Verdier, G. Gutierrez, C. Mendez, C. de Prada, Optimizing crude oil operations scheduling considering blending in

tanks, IFAC-PapersOnLine 58 (14) (2024) 67–72, 12th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2024. doi:https://doi.org/10.1016/j.ifacol.2024.08.315.
URL https://www.sciencedirect.com/science/article/pii/S2405896324010589

[47] T. G. García-Verdier, G. G. Rodríguez, C. G. Palacín, C. Méndez, C. De Prada, Programación de operaciones en una refinería con acceso marítimo, 2021. doi:10.17979/spudc.9788497498043.450.
URL https://doi.org/10.17979/spudc.9788497498043.450

[48] T. G. Garcia-Verdier, G. Gutierrez, C. Méndez, C. G. Palacin, C. de Prada, Minimización de riesgo en programación de operaciones de crudos en una refinería, Simposio Conjunto de los Grupos Temáticos de CEA "Modelado, Simulación, Optimización e Ingeniería de Control" (2022) 22–27.

[49] T. J. G. Garcia-Verdier, G. G. Rodriguez, C. A. Mendez, C. G. Palacín, C. De Prada, Evaluación de optimización estocástica aplicada a programación de operaciones de crudos en una refinería, 2022. doi:10.17979/spudc.9788497498418.0508.
URL https://doi.org/10.17979/spudc.9788497498418.0508

[50] T. G. Garcia-Verdier, G. Gutierrez, C. Méndez, C. G. Palacin, C. de Prada, Programación mensual de operaciones de crudos en una refinería con terminal marítima, II Simposio Conjunto de los Grupos Temáticos de CEA "Modelado, Simulación, Optimización e Ingeniería de Control" (2023) 9–10.

[51] T. G. Garcia-Verdier, G. Gutierrez, C. Mendez, C. de Prada, Optimizing the monthly scheduling of crude oil operations at a refinery, Editorial de la Universidad de Cantabria, 2023. doi:10.22429/euc2023.026.
URL https://doi.org/10.22429/euc2023.026

[52] C. G. Palacín, Efficient scheduling of batch processes in continuous processing lines (2020). doi:10.35376/10324/43323.
URL http://uvadoc.uva.es/handle/10324/43323