# Large-Scale Optimization of Nonconvex MINLP Refinery Scheduling

*Robert E. Franzoi,* [a,b] *Brenno C. Menezes,* [c] *Jorge A. W. Gut,* [b*] *Ignacio E. Grossmann,* [d]

[a] Andlinger Center for Energy and the Environment, Princeton University, Princeton, NJ, United States.

[b] Department of Chemical Engineering, Universidade de São Paulo, São Paulo, Brazil.

[c] Division of Engineering Management and Decision Sciences, College of Science and Engineering, Hamad Bin Khalifa University, Qatar Foundation, Doha, Qatar.

[d] Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, United States.

*Corresponding Author. E-mail address: jorgewgut@usp.br (Jorge A. W. Gut)

## Abstract

The modeling and optimization of large-scale refinery scheduling problems is challenging because of their complexity and size. There are six particularly important features addressed herein. From a modeling perspective, we consider an integrated refinery scheduling network, coherent mathematical formulation based on plant requirements, and advanced process-unit modeling. From a solving perspective, this includes optimization decision-making, evaluation of scheduling parameters, and re-optimization mechanisms. We propose a novel mathematical model to represent refinery scheduling operations more accurately and realistically. Then, we develop a state-of-the-art modeling and optimization framework to solve industrial-size refinery scheduling problems. The framework leverages the use of mathematical optimization and algorithmic methods by combining modeling approaches (process design, model decompositions), solving strategies (rescheduling procedures, heuristic algorithms), and machine learning regression (reduced-order models). An industrial-size refinery scheduling problem formulated as a nonconvex mixed-integer nonlinear programming (MINLP) model is successfully optimized, providing higher profitability and more efficient scheduling operations.

Keywords: Mathematical programming, optimization, refinery scheduling, nonconvex MINLP, decision-making framework, heuristic algorithms, machine learning.

## 1. Introduction

Over the last decades, crude oil refinery operations have grown increasingly complex. Factors including tighter competition, stricter environmental regulations, and lower margin profits, require companies to employ enhanced computer-aided decision-making capabilities for achieving higher efficiency and reducing costs in planning, scheduling, and supply chain applications (Shah et al., 2015). However, such operations are very complex and involve a large number of feedstocks, flows, units, properties, and products within an enterprise-wide optimization (EWO) problem.

A key goal of crude oil refineries is to properly determine the scheduling operations, which comprise the decision-making of tasks to be performed in the plant, and significantly affect the economics and operations in crude oil refineries. Refinery scheduling involves multiple sources of uncertainties and includes a large number of parameters, variables, and constraints. Its mathematical formulation includes information referred to as quantity-based (e.g., material flows, amounts, yields, and inventories), logic-based (e.g., binary or discrete decisions), and quality-based (e.g., nonlinear information of properties such as specific gravity, sulfur concentration, pour point, octane number, etc.). The resulting model is a large-scale nonconvex mixed-integer nonlinear programming (MINLP) problem, which is particularly challenging because of its complexity (combinatorial decisions, nonlinear and nonconvex equations) and size (large number of variables and constraints). This imposes difficulties for the solution of complex industrial cases, and which is by far intractable given the current state-of-the-art modeling tools and optimization capabilities (e.g., computer power, optimization solvers).

Many strategies have been developed and adopted in both industry and academia to provide tractable formulations for refinery scheduling problems, which introduce simplifications for the modeling of processing units, representation of the process network scope, and search space in optimization algorithms. However, not only are these simplifications often too drastic and unrealistic relative to the industrial needs, but there is still a huge gap between the state-of-the-art solution methods and the refinery scheduling problems that need to be solved in the industry. There have been recent advances in computer-aided resources, solution algorithms, and decision-making (Franzoi et al., 2018a; Rafiei and Ricardez-Sandoval, 2020). Improved modeling, solving, implementation, and integration approaches, as well as machine learning and big data strategies, increasingly lead to opportunities to lower costs and improved operations. Commercial optimization solvers have become increasingly efficient, in addition to the enhancements of computational processing. Remarkable success has been achieved in improving refinery scheduling optimization, including the development of novel methods, models, and algorithms (Li et al., 2020). The remaining challenges rely on developing mathematical models and decision-making frameworks to address the typical industrial challenges and provide implementable solution for industrial scheduling decision-making.

## 1.1 Current limitations on refinery scheduling decision-making

Simplifications have been traditionally introduced in refinery scheduling problems to achieve tractability. However, they often compromise the solution quality or lead to schedules that are not easily implementable in the plant. In the following, we highlight key considerations for the modeling and optimization of industrial-scale refinery scheduling.

First, due to modeling, solving, and computational limitations, building unit operation models and optimizing the integrated refinery units in multilevel decisions remains intractable. Hence, the refinery scheduling problem has been traditionally decomposed into smaller sub-problems that are modeled and solved separately, namely, crude oil scheduling, production network, and product blending network (Shah et al., 2015). Solving an integrated scheduling formulation comprised of all three segments is denoted herein as refinery scheduling. It is well established that better solutions are obtained by solving integrated formulations rather than their hierarchical counterparts (Yang et al., 2020). Feasible solutions found in the hierarchical approach may not be feasible in the integrated formulation, and the conditions for optimality are also different. Despite extensive literature on scheduling optimization for each refinery subproblem, such an integrated scheduling formulation for industrial-sized applications has not been studied (Li et al., 2020).

Second, standard refinery scheduling formulations do not include some of the key variables and requirements from the plant. Hence, their mathematical representations are not coherent with real-world industrial operations that involve a systematic production with detailed logistics and quality operations (Mouret et al., 2009; Castro and Grossmann, 2014; Xu et al., 2017). As pointed out by Franzoi et al. (2019), it is imperative to account for a detailed and accurate refinery network. Moreover, it is fundamental to consider logistics limitations by introducing constraints that model drawing and filling operations in tanks (e.g., to avoid simultaneous operations), operational modes (i.e., same physical unit operating at distinct conditions), minimum operating time of units, and unit downtimes (i.e., shut-down or maintenance periods), are useful to incorporate relevant plant requirements into the mathematical model (Zyngier and Kelly, 2009). The lack of such constraints in refinery scheduling formulations compromises the implementation of the solution in the plant.

Third, several industrial processes still employ trial-and-error decision-making rather than optimization-based approaches (Chryssolouris et al., 2005; Zhang et al., 2010). However, the space search and exploration employed by optimization methods are greatly superior and especially important for industrial operations, which comprise complex, uncertain, and dynamic processes and that consider process safety, operational requirements, scheduling implementation, and plant-model mismatches. Employing optimization approaches for scheduling problems provides several benefits in terms of improved operations and profitability.

Fourth, simulation-based rigorous formulations are often employed for processing units throughout the refinery, such as distillation towers. However, they are computationally expensive and not suitable for scheduling optimization. Instead of using complex or rigorous models to model these units, simplified correlations may be employed (e.g., surrogate equations), which allow the integration of processing unit models in refinery scheduling optimization applications (Cuadros Bohorques et al., 2020; Franzoi et al., 2020). These surrogates are supposedly accurate representations that provide relevant additional information at a low computational cost.

Fifth, proper calibration of scheduling parameters such as time-step and time horizon is fundamental. On one hand, the computational effort is key and imposes limitations on the model complexity. On the other hand, an ideal solution would benefit from the gains achieved with small time-steps and a large time horizon. Small time-steps provide better decision-making and better solutions due to the additional degrees of freedom available, which results in better management of resources and production configuration. A long time horizon provides additional information that impacts the optimal scheduling through spot market opportunities and more efficient production and management of resources.

Sixth, noises, disturbances, and other unforeseen events happen over time and lead to plant-model mismatches, which significantly worsen the refinery operations and economics. Some optimization fields such as stochastic programming and robust optimization aim to provide proper methodologies for such problems. Although efficient for diverse applications, stochastic approaches require probability functions that may be difficult to estimate or that needlessly increase the model complexity, and may fail to handle constrained problems because the stochastic search operators often produce infeasible solutions (Francisco et al., 2005); and robust optimization methods are very conservative as they aim to hedge against all possible worst-case uncertain scenarios, which often leads to overly expensive or needlessly complex solutions (Garcia et al., 2016). Therefore, these approaches may not be the most suitable alternatives for refinery scheduling. Conversely, re-optimization procedures, known as online scheduling or rescheduling, provide efficient capabilities for revising the optimal schedule whenever necessary or beneficial, thereby mitigating plant-model mismatches (Franzoi et al., 2021a). This is especially important in complex processing sites that require integrated plant-model environments for data updating and accurate tracking of quality information throughout the process network (Franzoi et al., 2018a).

### 1.2 Objectives and contributions of this work

The objective of this paper is threefold. First, we discuss the state-of-the-art limitations of large-scale refinery scheduling applications, and we provide an overview of efficient modeling and optimization methods for refinery scheduling. Second, we present a novel mathematical model that comprises process-unit models and operational constraints particularly relevant to industrial scheduling operations. Third, we propose a novel decision-making framework and illustrate its application to solve an industrial-scale nonconvex MINLP refinery scheduling problem.

There are six particularly important features addressed herein, some of which have not been typically considered in the literature and industry. From a modeling perspective, this includes the consideration of an integrated refinery scheduling network, coherent mathematical formulation based on plant requirements, and advanced process-unit modeling. Specifically, we propose a novel mathematical model that represents refinery scheduling operations in a more accurate and realistic fashion. The model comprises an integrated refinery topology with crude oil scheduling, production network, and product blending network; critical constraints are introduced to manage the logistics operations of tanks and units; and the process-unit models for the crude distillation towers are represented through data-driven surrogate equations for improved accuracy.

From a solving perspective, this includes optimization decision-making, consideration of scheduling parameters, and re-optimization mechanisms. We propose a novel decision-making framework that employs advanced modeling and solving methods, and which relies on optimization decision-making. We study and evaluate how scheduling parameters affect the solution method. Finally, we indicate how our proposed methodology is suitable for re-optimization mechanisms within a moving horizon fashion, which aims to further improve the scheduling solution and provide desired characteristics for industrial applications.

The framework is shown to efficiently solve large-scale nonconvex MINLP refinery scheduling problems comprised of hundreds of thousands of variables and constraints that to the best of our knowledge have never been addressed in such complexity in the literature. It comprises a) discrete-time modeling using the unit operation port state superstructure (UOPSS) representation; b) temporal decomposition; c) phenomenological decomposition; d) efficient design for the mathematical formulation (model design) and the operations (process design); e) model reduction strategies; f) relaxation heuristics for MILP models; g) rescheduling strategies to minimize plant-model mismatches by handling process uncertainties and disturbances; h) surrogate models as an alternative to rigorous counterparts.

The methodologies discussed in this work are valid for diverse types of formulations (e.g., NLP, MILP, MINLP), problems (e.g., planning, scheduling, logistics, supply chain), applications (e.g., simulation, optimization, machine learning), and industries (e.g., petrochemical, mining, production, transportation). They represent a great potential for improving state-of-the-art capabilities with significant benefits for the academic literature and industrial decision-making. This work aims to provide guidelines for further research by solving large-scale and complex problems within an efficient decision-making optimization framework. We believe this is a fundamental step towards further breakthroughs in the mathematical optimization field.

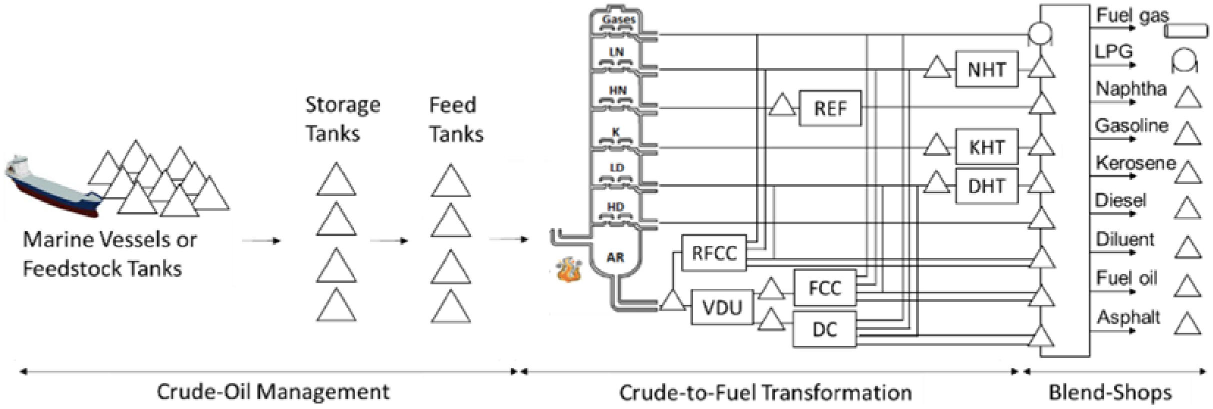
This paper is outlined as follows. An overview of refinery scheduling and algorithmic methods for refinery scheduling optimization is provided in Section 2. A novel mathematical formulation for refinery scheduling is presented in Section 3. A modeling and optimization decision-making framework is proposed in Section 4, which solves a large-scale refinery scheduling example presented in Section 5. The remarks and future outlook are highlighted in Section 6.

## 2. Literature review

### 2.1 Crude oil refinery network

The refinery network shown in Figure 1 includes three segments, namely, crude oil scheduling, production network, and product blending network.

Figure 1: Crude oil refinery network. (adapted from Kelly et al., 2017).



The crude oil scheduling comprises the unloading and mixing operations of crude oil feedstocks. Upon arrival, the crude oils are unloaded and stored in storage tanks. Mixing operations are carried out to properly prepare the blend recipe to feed the distillation units according to the refinery needs. The quality and composition of the distillation feed significantly affect all the subsequent material streams throughout the plant. This impacts the entire refinery operations, the yields and quality of products, and the scheduling feasibility and profitability. Hence, it represents a key decision for the refinery due to the high impact both in terms of economics and operations (Kelly et al., 2017). Because of its importance and complexity, crude oil scheduling has been well-studied in the literature (Lee et al., 1996; Jia et al., 2003; Reddy et al., 2004; Saharidis et al, 2009; Robertson et al., 2011; Castro and Grossmann, 2014; Franzoi et al., 2019).

The crude oil blends that feed the distillation unit are physically separated into fractions referred to as distillates. Throughout the production network, the distillates undergo multiple conversion and treatment processes (e.g., catalytic reforming, fluid catalytic cracking, delayed coker, hydrocracking, hydrotreating, debutanizer, depropanizer), which convert the crude oil fractions into more valuable products whereby providing quality enhancement and increased economic value (Do, 2014). Given the molecular and compositional interaction complexity within such physical and chemical processes, most works on the topic are based on rigorous simulation approaches and are mostly employed for real-time optimization and control. These complex refinery unit models introduce computational complexity (e.g., a large number of variables and constraints, and a high degree of nonlinearity) in the solution of large-scale MINLP problems (Li et al., 2020). Therefore, simplified correlations are often adopted to be used within scheduling environments for several refinery unit operations, including distillation (Fu et al., 2020; Franzoi et al., 2020), fluid catalytic cracking (Cuadros Bohorquez et al., 2020), catalytic reforming (Mencarelli et al., 2020), hydrocracking (Zhong et al., 2019; Song et al., 2021), hydrotreating (Wang et al., 2019; Xia et al., 2021), among others.

The product blending segment concerns the mixing of intermediate refinery streams into final products to be sold and distributed via logistics operations. Blending formulations are highly nonconvex and nonlinear because of the quality balances for property calculation and tracking, which are required for meeting product specifications (Al-qahtani and Elkamel, 2011; Shah et al., 2015). Product blend scheduling has been extensively studied (Mendez et al., 2006; Li and Karimi, 2011; Cuiwen et al., 2013; Castillo-Castillo et al., 2016; Franzoi et al., 2019, Ali et al., 2022).

## 2.2 Algorithmic methods for industrial refinery scheduling optimization

The detailed mathematical formulation needed to represent refinery scheduling problems is difficult to be solved as full-space nonconvex MINLP problems. To overcome the computational limitations that arise from the high complexity and combinatorial size of such systems, algorithmic methods based on heuristics and decompositions have been often employed. In this section, we introduce strategies that are particularly useful for tackling large-scale refinery scheduling models. They provide significant benefits and have good cost-effectiveness in reducing computational effort while maintaining solution quality.

### 2.2.1 Phenomenological decomposition heuristic

The phenomenological decomposition heuristic (PDH) was proposed by Menezes et al. (2015) for strategic refinery planning. The original model is formulated as an MINLP problem, whereas a quantity-logistics-quality decomposition breaks it down into smaller MILP-NLP sub-models to be sequentially and iteratively solved. The MILP sub-model is built and optimized considering only logistic and quantity information (i.e., by neglecting nonlinear variables and equations). The binary variables from the MILP optimal solution are fixed in the original MINLP, and the resulting NLP model is optimized considering only quantity and quality information. The yields and properties from the NLP optimal solution are used as updated parameters in the next MILP iteration. Important hyperparameters to be calibrated include the criteria for convergence and the selection of MILP/NLP solutions to be used within a retro-feeding fashion in the next sub-problem optimization.

### 2.2.2 Linear programming reformulation of nonconvex equations

Despite the computational benefits provided by decomposition heuristics, such methodologies often introduce gaps within their iterative procedure. In the phenomenological decomposition heuristic, the MILP sub-model neglects relevant quality information to avoid nonlinearities and nonconvexities in the model. For improved performance of the method, Kelly et al. (2018) develop a linear programming reformulation of nonconvex blending equations that introduces linearized quality information in the MILP sub-problems. Nonconvex quality constraints are approximated by linear formulas valid exclusively for mixing equations (e.g., blending of streams) to be incorporated as an additional set of constraints. The property variables are considered invariant coefficients in quality material balances that match the product specification by including slack or surplus variables in an equality constraint.

### 2.2.3 Temporal decomposition heuristic

Temporal decomposition heuristics (TDH) are straightforward time-based strategies that provide significant model reduction over the time dimension. Previous works used similar approaches for the optimization of discrete-time scheduling problems typically found in the petrochemical industry (Kelly, 2002). This heuristic proposes a temporal decomposition in which the entire time horizon $t$ is discretized into $tc$ time chunks. The sub-models are sequentially solved, whereby the initial values of the decision variables are based on the optimal solution of the previous sub-model. Thus, multiple time-discretized sub-models are built over a smaller time horizon $t/tc$, which are solved within a rolling horizon fashion. The complete optimal scheduling (referred to as closed-loop solution) is then achieved by sequentially aggregating the optimal solutions from each sub-model (referred to as open-loop solutions). For additional information on rolling horizon strategies, see Franzoi et al. (2021a).

### 2.2.4 Relaxation heuristic

Relaxation heuristics can be a powerful method for tackling large-scale mixed-integer programming (MIP) problems by reducing the computational burden in branch-and-bound discrete searches. A common example are the relax-and-fix strategies. For details, see Absi and van den Heuvel (2019).

### 2.2.5 Surrogate modeling for refinery unit-operations

Process unit models utilize computationally expensive high-fidelity formulations that are not suitable for scheduling optimization. Surrogate models have been increasingly developed as a simplified yet accurate alternative. They can be integrated into scheduling optimization with minimal increase in computational effort and can be recursively updated based on measurement feedback using plant data (Franzoi et al., 2021b,c). Franzoi et al. (2020) propose a data-driven surrogate modeling framework for the crude distillation unit (CDU) based on mixed-integer quadratic programming (MILP) that correlates independent variables (crude oil compositions and temperatures) to dependent variables (yields and properties of distillates). Other refinery unit-operations (e.g., naphtha reformer, fluid catalytic cracker, delayed coker) can also be approximated by surrogate formulations.

### 2.2.6 Online optimization of refinery scheduling

The implementation of the scheduling solution in the plant is a critical and challenging task. Such dynamic systems are subject to uncertainties and disturbances, which often lead disruptions, delays, and market fluctuations. This significantly affects the scheduling solution and result in sub-optimality or infeasibilities (Gupta and Maravelias, 2016). Furthermore, operational data used in the scheduling are typically out of date or not integrated with the production network, which leads to inconsistencies in the prediction throughout the process.

Online optimization of refinery scheduling is required to reduce plant-model mismatches and to ensure optimal operations. Such a periodic or event-triggered scheduling updating mechanism handles the occurrence of disturbances and unforeseen events, improves the scheduling solution, and minimizes losses. Ideally, it should be performed within the shift of the operators or even in smaller time windows. Feedback from the plant can be automatically connect to the decision-making core, whereby systematically performing rescheduling optimization to maintain the state of the system up-to-date and to handle changes and disturbances in the process. This is a fundamental step to achieve more efficient production and management of resources (Franzoi et al., 2021a).

## 3. Mathematical formulation for refinery scheduling

We propose a novel mathematical formulation for industrial-scale refinery scheduling problems. The model is built based on discrete-time and uses the UOPSS representation. Previous works on the topic have shown good performance of the UOPSS representation for modeling and optimizing large-scale refinery scheduling (Kelly et al., 2017; Brunaud et al., 2020). Our proposed full-space MINLP model is comprised of three parts:

- The base formulation includes constraints to manage the relationships between unit-operations and mass balances. All constraints from the base formulation are also used in the MILP and NLP formulations.
- The MILP formulation (Logistics problem) includes the base formulation in addition to logistics constraints.
- The NLP formulation (Quality problem) includes the base formulation in addition to the quality balances.

The novelty of our formulation is twofold. First, we propose an integrated model for the entire refinery network, which includes all three segments (crude oil scheduling, production network, and product blending network) within an industrial-scale formulation. We introduce equations to estimate the outputs from process units based on surrogate modeling (for the distillation network) and fixed yield models (for the other process units in the production network). The full-space MINLP model comprises around 70,000 continuous and 50,000 binary variables, 180,000 constraints and 90,000 degrees of freedom.

Second, we introduce critical constraints to represent the logistics operations in the refinery site. Although refinery scheduling operations are associated with many types of logistics limitations and requirements, mathematical formulations in the industry and literature typically neglect these aspects due to their high complexity, detailed information, and difficult implementation. The operational limitations introduced in our model are fundamental to achieving realistic scheduling solutions. This is particularly relevant from an implementation perspective. Given that the logistics requirements are a critical part of the refinery scheduling, our novel formulation leads to solutions that can be more easily and smoothly implemented in the plant. Another advantage is that our constraints implicitly account for changeover operations in the distillation network. This eliminates the need of explicitly adding changeover equations, whose parameters are difficult to estimate and tune, leading to inaccuracies.

In this section, we introduce the:

- Most important UOPSS characteristics, objects, and groups of objects.
- Lists of indices, sets, subscripts, superscripts, parameters, and variables.
- Equations, constraints, and objective function.

### 3.1 UOPSS objects

The network shown in Figure 2 is comprised of a group of main objects that represent the sources and sinks (◇), tanks and inventories (Δ), and continuous processes (⊠). These objects are referred to as unit-operations hereinafter. The unit-operations can be classified into four categories, namely, perimeter-based, pool-based, mixed-based, and splitter-based. There is a relationship between how the unit-operations of each category are represented and modeled in the formulation and their logistics and processing operations in the refinery site.

- Perimeter-based represent quantity-flows incoming to or outgoing from the plant (e.g., arrival of crude or delivery of final products through pipelines).
- Pool-based refers to intermediate tanks throughout the plant, often allocated before or after processing units. They are important for ensuring smooth operations (e.g., constant feed for downstream units) and to provide logistics flexibility.
- Mixer-based are continuous stirred blender units typically used to prepare the feed for processing units or for final product specifications by blending multiple intermediate streams with boosters (e.g., for the production of fuels). They can receive multiple simultaneous inlet streams, although a single outlet stream is operationally allowed.
- Splitter-based refers to physical and chemical processing units (e.g., distillation, fluid catalytic cracking, hydrotreating, delayed coking). For efficient operations and control, the profiles of the feed amount and quality should be as smooth as possible, with minimum or no variations over time. Splitter-based units support a single inlet stream, but can have multiple outlet streams, typically associated with the conversion and separation processes.

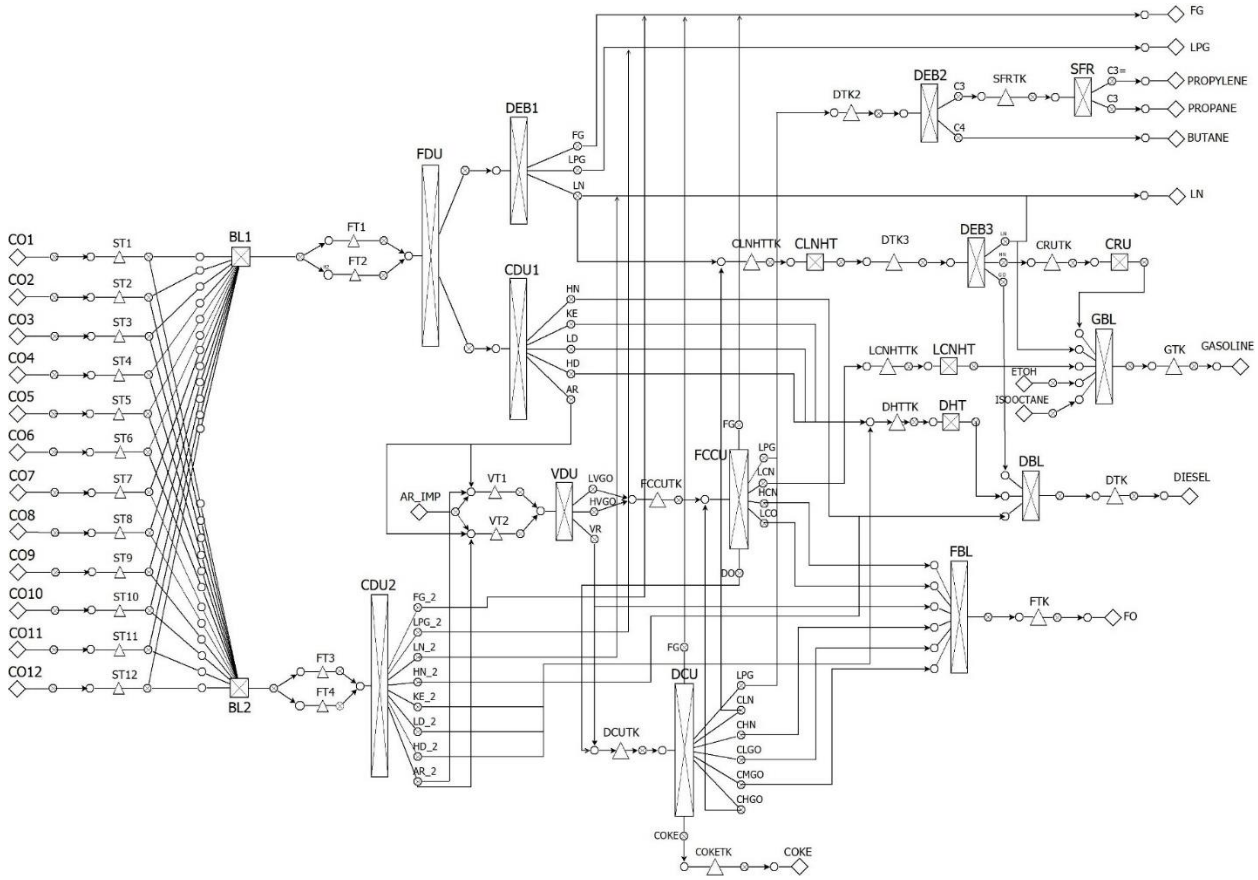


Figure 2: Refinery scheduling network.

There are auxiliary objects before and after each unit-operation, referred to as in-ports and out-ports (O and ⊗) and arrows (→) that represent connections between ports (e.g., material flows). The unit-operations are configurable to have operational modes and are connected to inlet and outlet ports. The ports are combined in pairs $(j, i)$ to represent transfer operations throughout the network, whereby establishing the material balances for each unit-operation. States are used for configuring non-material flows and elements related to non-material flows. For example, states can configure the energy transfer throughout the network to establish energy balances. They also allow the inclusion of hypothetical modeling configurations based on non-physical relationships.

The UOPSS representation is formed by groups of objects that can be categorized as unit-operation (UO), unit-operation-port-state (UOPS), and unit-operation-port-state-unit-operation-port-state (UOPS-UOPS). The UO groups represent units with operational modes, the UOPS groups additionally include port and state information, and the UOPS-UOPS groups connect adjacent UOPS groups to ensure consistency in the flows throughout the network. The UO groups have binary and continuous variables associated with processing rates, orders, or inventories. The UOPS groups have continuous variables associated with inlet and outlet operations, which are useful for handling multiple inlet or outlet streams within the same unit-operation (e.g., splitters, mixers). The arrows connecting adjacent ports establish the UOPS-UOPS relationships, which are modeled with binary and continuous variables to represent material or energy transfers between unit-operations. An illustrative example providing additional details on the UOPSS representation can be found in the Supplementary Material A.

### 3.2 Indices and Sets

The following indices and sets are considered in the refinery scheduling model.

- $i \in \{I, I_{BL}, I_{FT}, I_{TK} I_{UN},\}$ for outlet ports of unit-operations.
- $j \in \{J, J_{BL}, J_{ST}, J_{TK}, J_{UN}\}$ for inlet ports of unit-operations.
- $m \in M = \{M_{BL}, M_{CDU}, M_{FE}, M_{PR}, M_{TK}, M_{UN}\}$ for unit-operations $m$ of blenders, crude distillation units, feedstocks, products, tanks, and processing units.
- $op \in OP = \{op_1, op_2, \dots, op_{modes}\}$ for operational modes of processing units.
- $p \in \{P_v, P_w\}$ for volume-based and mass-based properties.
- $t \in T = \{t_1, t_2, \dots, t_n\}$ for $n$ time periods.
- $tt \in T = \{tt_1, tt_2, \dots, tt_n\}$ for $n$ time periods.

Figure 3 presents an illustrative diagram to provide an easier understanding of the relationship among unit-operations and their interconnected ports. A generic unit-operation $m$ is directly connected to in-port $i$ and out-port $j$. This inlet port $i$ handles upstream processes to $m$, i.e., incoming transfers or logistics operations. Moreover, $i$ is connected to $j_{up}$, which is the out-port of an upstream unit-operation to $m$, i.e., $m_{up}$. The arrow connection for the in-port-out-port pair $(j_{up}, i)$ establishes the UOPS-UOPS connection between the unit-operations $m_{up}$ and $m$. It is similar for the pair $(j, i_{do})$ that connects unit-operation $m$ with its downstream counterpart $m_{do}$.

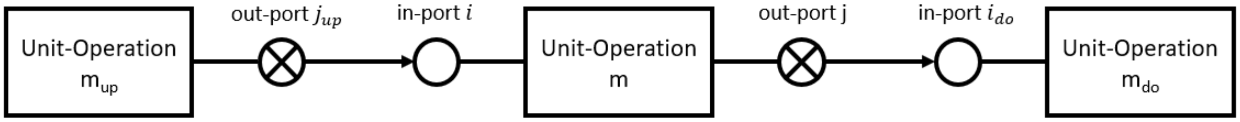


Figure 3: Relationship among unit-operations and ports.

### 3.3 Subscripts and superscripts

The following subscripts and superscripts are used in the refinery scheduling model.

- $BL$: blender unit.
- $c$: crude oil component.
- $CDU$: crude distillation unit.
- $cut$: crude oil cutpoint fraction.
- $DEV$: total flowrate deviation.
- $DEV1$: flowrate deviation (increase).
- $DEV2$: flowrate deviation (decrease).
- $do$: downstream unit-operation or flow.
- $DRAW$: drawing operations.
- $FAC$: factor.

- $FE$: feedstock.
- $FILL$: filling operations.
- $FT$: feed tank.
- $L$: lower bound.
- $NLP$: nonlinear programming.
- $op$: operational mode of process units.
- $opt$: optimal value.
- $p$: property.
- $PR$: product.
- $sg$: specific gravity.
- $SP$: starting point.
- $ST$: storage tank.
- $sul$: sulfur concentration.
- $TK$: tank.
- $U$: upper bound.
- $UN$: processing unit.
- $up$: upstream unit-operation or flow.

### 3.4 Parameters

The following parameters are included in the refinery scheduling model.

- $\overline{\Delta \mathrm{D}}_m^L$: lower bound for the fill-draw delay constraint of unit-operation m ∈ $M_{TK}$.
- $\overline{\Delta \mathrm{D}}_m^U$: upper bound for the fill-draw delay constraint of unit-operation m ∈ $M_{TK}$.
- $\overline{\Delta t}$: time-step size.
- $\overline{dv}^L$: lower bound for a generic decision variable $dv$.
- $\overline{dv}^{SP}$: starting point of a generic decision variable $dv$.
- $\overline{dv}^U$: upper bound for a generic decision variable $dv$.
- $\bar{f}_{i,p,t}$: quality factor for property $p$ from in-port $i$ at time period t.
- $\bar{f}_{j,p,t}$: quality factor for property $p$ to out-port $j$ at time period t.
- $\overline{mu}_{i,t}^L$: lower bound for the multi-use upstream operations from out-port i at time period t.
- $\overline{mu}_{i,t}^U$: upper bound for the multi-use upstream operations from out-port i at time period t.
- $\overline{mu}_{j,t}^L$: lower bound for the multi-use downstream operations from out-port j at time period t.
- $\overline{mu}_{j,t}^U$: upper bound for the multi-use downstream operations from out-port j at time period t.
- $\bar{n}_p$: number of time periods.
- $\overline{pen}_m$: penalty over the performance of distillation unit $m \in M_{CDU}$.
- $\overline{price}_{m,t}$: market value of product $m \in M_{PR}$ or cost of feedstock $m \in M_{FE}$ at time period t.
- $\bar{r}_{\mathrm{i,t}}^{\mathrm{L}}$: lower bound for the inverse yield from out-port j at time period t.
- $\bar{r}_{\mathrm{i,t}}^{\mathrm{U}}$: upper bound for the inverse yield from out-port j at time period t.
- $\bar{r}_{\mathrm{j,t}}^{\mathrm{L}}$: lower bound for the direct yield from out-port j at time period t.
- $\bar{r}_{\mathrm{j,t}}^{\mathrm{U}}$: upper bound for the direct yield from out-port j at time period t.

- $\bar{t}_{end}$: time horizon length.
- $\overline{upt}_{m}^{L}$: lower bound for the uptime constraint of unit-operation $m$.
- $\overline{upt}_{m}^{U}$: upper bound for the uptime constraint of unit-operation $m$.
- $\mathrm{\bar{x}_{j,i,t}^{L}}$: lower bound for the material flow from out-port j to in-port i at time period t.
- $\mathrm{\bar{x}_{j,i,t}^{U}}$: upper bound for the material flow from out-port j to in-port i at time period t.
- $\mathrm{\bar{x}_{m,t}^{L}}$: lower bound for the flowrate of unit-operation m at time period t.
- $\mathrm{\bar{x}_{m,t}^{U}}$: upper bound for the flowrate of unit-operation m at time period t.
- $\mathrm{\overline{xh}_{m,t}^{L}}$: lower bound for the inventory of unit-operation m ∈ $M_{TK}$ at time period t.
- $\mathrm{\overline{xh}_{m,t}^{U}}$: upper bound for the inventory of unit-operation m ∈ $M_{TK}$ at time period t.
- $\overline{xh}_{m,t}^{DRAW}$: lower bound for the inventory of unit-operation m ∈ $M_{TK}$ at time period t upon completion of a drawing operation.
- $\overline{xh}_{m,t}^{FILL}$: upper bound for the inventory of unit-operation m ∈ $M_{TK}$ at time period t upon completion of a filling operation.
- $\overline{xh}_{m,t}^{Max}$: big-M parameter to represent the maximum inventory of unit-operation m ∈ $M_{TK}$ at time period t.

## 3.5 Variables

The decision variables considered in the refinery scheduling model comprise quantity (i.e., material streams, inventory levels), logistics (i.e., binary decisions regarding unit-operations and connection flows), and quality (i.e., yields, compositions, properties) information.

### 3.5.1 Continuous variables

- $dv^{opt}$: optimal value of a generic decision variable $dv$.
- $rand$: random number generated.
- $re_{j,c,cut}^{yie}$: rendering for the yield of crude oil cutpoint $cut$ of crude oil component $c$ in the in-port $j$.
- $re_{j,c,cut}^{p}$: rendering for the quality of crude oil cutpoint $cut$ of crude oil component $c$ in the in-port $j$.
- $v_{i,p,t}$: volume-based quality $p$ of the downstream flow in the in-port i of its respective unit-operation at time period $t$.
- $v_{j,p,t}$: volume-based quality $p$ of the upstream flow in the in-port j of its respective unit-operation at time period $t$.
- $w_{i,p,t}$: mass-based property $p$ of the downstream flow in the in-port i of its respective unit-operation at time period $t$.
- $w_{j,p,t}$: mass-based property $p$ of the upstream flow in the in-port j of its respective unit-operation at time period $t$.
- $x_{j,i,t}$: material flow from out-port $j$ to in-port $i$ at time period $t$.
- $x_{j^{FAC},p,t}$: surplus/slack quality amount of the factor-flow $j^{FAC}$ for each property $p$ at time period t.
- $x_{m,t}$: material flow of unit-operation $m$ at time period $t$.
- $x_{m,t}^{DEV}$: flowrate deviation in unit-operation $m \in M_{CDU}$ between time periods $t-1$ and $t$.

- $x_{m,t}^{DEV1}$: flowrate deviation (increase) in unit-operation $m \in M_{CDU}$ between time periods $t-1$ and $t$.
- $x_{m,t}^{DEV2}$: flowrate deviation (decrease) in unit-operation $m \in M_{CDU}$ between time periods $t-1$ and $t$.
- $xh_{m,t}$: material inventory of tank unit-operation $m$ at time period $t$.
- $zsd_{m,t}$: semi-continuous variable [0,1] to represent shut-down operations of unit-operation $m$ at time period $t$.
- $zsu_{m,t}$: semi-continuous variable [0,1] to represent start-up operations of unit-operation $m$ at time period $t$.
- $zsw_{m,t}$: semi-continuous variable [0,1] to represent switch-over-to-itself operations of unit-operation $m$ at time period $t$.
- $Z$: objective function to be maximized.

### 3.5.2 Binary variables

- $y_{j,i,t}$: discrete decision associated with the connection between out-port $j$ and in-port $i$ at time period $t$.
- $y_{m,t}$: discrete decision associated with unit-operation $m$ at time period $t$.
- $y_{m^{op},t}$: discrete decision for operational mode $op$ of unit-operation $m$ at time period $t$.
- $yd_{m,t}^{DRAW}$: discrete decision to represent drawing operations in unit-operation m $\in M_{TK}$ at time period $t$.
- $yd_{m,t}^{FILL}$: discrete decision to represent filling operations in unit-operation m $\in M_{TK}$ at time period $t$.

## 3.6 Base formulation

The base formulation is defined in Equations (1) to (15) and includes both continuous and binary variables within UO, UOPS, and UOPS-UOPS relationships. All out-ports $j$ and in-ports $i$ are associated with a unit-operation $m$ (i.e., $j_m$ and $i_m$), but the subscript $m$ is omitted for the sake of simplicity. Equation (1) defines the non-negative real variables $x_{m,t}, x_{j,i,t}$, and $xh_{m,t}$, and Equation (2) defines the binary variables $y_{j,i,t}, y_{m,t}$.

$$x_{m,t},\ x_{j,i,t},\ xh_{m,t} \geq 0 \tag{1}$$

$$y_{j,i,t}, y_{m,t} = \{0,1\} \tag{2}$$

The arrows throughout the network shown in Figure 3 represent the UOPS-UOPS material flows $x_{j,i,t}$, which are constrained by the inlet and outlet flowrate (e.g., hydraulic limits) and by the processing capacities of units. They vary between lower and upper bounds ($\bar{x}_{j,i,t}^{L}$ and $\bar{x}_{j,i,t}^{U}$) if their binary variables $y_{j,i,t}$ are active, as shown in Equation (3). If the binary variable is zero, the flows are automatically set to zero, which indicates that the respective UOPS-UOPS transfer $(\mathrm{j,i,t})$ is not permitted. This could represent maintenance procedures in the downstream or upstream unit connected to the flow $x_{j,i,t}$, or in the pipelines as well. Similarly, Equation (4) imposes bounds ($\bar{x}_{m,t}^{L}$ and $\bar{x}_{m,t}^{U}$) for the UO flows of unit-operations $x_{m,t}$ ($m \in \{M_{UN}, M_{BL}\}$) using binary variables

$y_{m,t}$ ($m \in \{M_{UN}, M_{BL}\}$), and Equation (5) imposes bounds ($\overline{xh}^{L}_{m,t}$ and $\overline{xh}^{U}_{m,t}$) for the UO tank holdups or inventory levels $xh_{m,t}$ ($m \in M_{TK}$) using the binary variables $y_{m,t}$ ($m \in M_{TK}$).

$$\bar{x}^{L}_{j,i,t}\, y_{j,i,t} \leq x_{j,i,t} \leq \bar{x}^{U}_{j,i,t}\, y_{j,i,t} \;\; \forall \;(j,i), t \tag{3}$$

$$\bar{x}^{L}_{m,t}\, y_{m,t} \leq x_{m,t} \leq \bar{x}^{U}_{m,t}\, y_{m,t} \;\; \forall \;\; t, m \in \{M_{BL}, M_{UN}\} \tag{4}$$

$$\overline{xh}^{L}_{m,t}\, y_{m,t} \leq xh_{m,t} \leq \overline{xh}^{U}_{m,t}\, y_{m,t} \;\; \forall \;\; m \in M_{TK}, t \tag{5}$$

If the binary variable $y_{m,t}$ of the unit-operation $m$ at time period $t$ is active, Equations (6) to (9) constrain the summation of the quantity-flows $x_{j,i,t}$ incoming to or outgoing from its port-states. Equations (6) and (7) impose lower ($\bar{x}^{L}_{m,t}$) and upper ($\bar{x}^{U}_{m,t}$) bounds for the sum of the UOPS-UOPS quantity-flows $x_{j,i,t}$ outgoing from the out-port $j_{up}$ of $m_{up}$ (unit-operation upstream of $m$) and incoming to the in-ports $i$ of $m$. Similarly, Equations (8) and (9) impose bounds for the sum of the $x_{j,i,t}$ quantity-flows outgoing from the out-ports j of m and incoming to the in-ports $i_{do}$ of $m_{do}$ (unit-operation downstream of $m$). If $y_{m,t}$ is inactive, the quantity-flows in Equations (6) to (9) are set to 0.

$$\sum_{j_{up}} x_{j_{up},i,t} \geq \bar{x}^{L}_{m,t}\, y_{m,t} \;\;\; \forall\, (i,m,t) \tag{6}$$

$$\sum_{j_{up}} x_{j_{up},i,t} \leq \bar{x}^{U}_{m,t}\, y_{m,t} \;\;\; \forall\, (i,m,t) \tag{7}$$

$$\sum_{i_{do}} x_{j,i_{do},t} \geq \bar{x}^{L}_{m,t}\, y_{m,t} \;\;\; \forall\, (m,j,t) \tag{8}$$

$$\sum_{i_{do}} x_{j,i_{do},t} \leq \bar{x}^{U}_{m,t}\, y_{m,t} \;\;\; \forall\, (m,j,t) \tag{9}$$

Unit-operations $m \in \{M_{BL}, M_{UN}\}$ can have multiple streams incoming to or outgoing from their independent connected ports. Equations (10) to (13) introduce bounds on yields, either inverse/input or direct/output. Inverse yields model the mixer-based unit-operations, in which there are multiple inlet streams within a given time window. Direct yields model splitter-based unit-operations, which split the incoming quantities in multiple outlet streams within a given time window. This set of constraints is useful to configure yields for inlet and outlet flows. For flows $x_{m,t}$ of unit-operation $m \in M_{BL}$, Equations (10) and (11) define bounds of inverse yields ($\bar{r}^{L}_{i,t}$ and $\bar{r}^{U}_{i,t}$) for the upstream operations to $m$ (e.g., to constrain the percentual yields of streams to be blended). For throughputs $x_{m,t}$ of unit-operation $m \in M_{UN}$, Equations while (12) and (13) define bounds of direct yields ($\bar{r}^{L}_{j,t}$ and $\bar{r}^{U}_{j,t}$) for downstream operations to $m$ (e.g., specifying yields of unit outputs). Equations (12) and (13) are applied for all unit-operations, given that they behave similarly to splitter units, i.e., with a single input stream but potentially multiple output streams. The yields $\bar{r}^{L}_{i,t}$ and $\bar{r}^{U}_{i,t}$ can be configured to be fixed (i.e., $\bar{r}^{L}_{i,t} = \bar{r}^{U}_{i,t}$) or to provide an optimizable range as an additional degree of freedom (i.e., $\bar{r}^{L}_{i,t} < \bar{r}^{U}_{i,t}$). Depending on the process network or configuration, Equations (10) and (11) can be extended to include splitter-based units, and Equations (12) and (13) can be extended to include mixer-based units. It is worth noting that Equations (10) to (13) are not valid for tanks, which typically have only one active inlet flow and only one active outlet flow at the same time window (i.e., due to operational limitations, tanks cannot have multiple inlet flows or multiple outlet flows simultaneously).

$$\sum_{j_{up}} x_{j_{up},i,t} \geq \bar{r}^{L}_{i,t}\, x_{m,t} \quad \forall\, (i,t), m \in \{M_{BL}\} \tag{10}$$

$$\sum_{j_{up}} x_{j_{up},i,t} \leq \bar{r}^{U}_{i,t}\, x_{m,t} \quad \forall\, (i,t), m \in \{M_{BL}\} \tag{11}$$

$$\sum_{i_{do}} x_{j,i_{do},t} \geq \bar{r}^{L}_{j,t}\, x_{m,t} \quad \forall\, (j,t), m \in \{M_{UN}\} \tag{12}$$

$$\sum_{i_{do}} x_{j,i_{do},t} \leq \bar{r}^{U}_{j,t}\, x_{m,t} \quad \forall\, (j,t), m \in \{M_{UN}\} \tag{13}$$

Equation (14) represents the material balance required to calculate the inventory or holdup $xh_{m,t}$ of storage and feed tanks m ∈ $M_{TK}$. At each time window $t$, the holdup amount $xh_{m,t}$ is the remaining amount of material in the past time period ($xh_{m,t-1}$), increased by the incoming material to the tank from the upstream connections ($x_{j_{up},i,t}$) and subtracted by the outgoing material from the downstream connections from the tank ($x_{j,i_{do},t}$).

$$xh_{m,t} = xh_{m,t-1} + \sum_{j_{up}} x_{j_{up},i,t} - \sum_{i_{do}} x_{j,i_{do},t} \;\; \forall\, (i,j,t), \mathrm{m} \in M_{TK} \tag{14}$$

The material balances to impose no accumulation of material in continuous processes are defined in Equation (15) for the unit-operation $m \in \{M_{BL}, M_{UN}\}$, whereby the total incoming flow to the unit-operation equals its total outlet flow. The pairs $(j_{up}, i)$ and $(j, i_{do})$ are the upstream and downstream connections of $m$.

$$\sum_{j_{up}} x_{j_{up},i,t} = \sum_{i_{do}} x_{j,i_{do},t} \quad \forall\, \Big( (j_{up}, i), (j, i_{do}), t \Big), m \in \{M_{BL}, M_{UN}\} \tag{15}$$

### 3.7 Logistics mathematical formulation: MILP refinery scheduling

The logistics problem includes Equations (1) to (15) previously shown in the UOPSS flowsheet formulation, in addition to Equations (16) to (37), which involve: a) constraints of structural transitions and selection of operational modes; b) temporal transitions of unit-operations in sequence-dependent cycles; c) sharing of objects (units, ports, etc.) in multi-use constraints; d) operational time and zero downtime of units; e) fill-draw delay, fill-to-full, and draw-to-empty operations for tanks; and f) linear reformulation of blending equations.

Equations (16) and (17) are structural transition constraints that coordinate setups of connected unit-operations through the out-port $j_{up}$ of $m_{up}$ and the in-port $i$ of $m$. If the binary variables of a given pair of unit-operations $(m_{up}, m)$ are active, then the binary variable $y_{j,i,t}$ that represents their connection must be active (i.e., material streams are allowed between them). However, if at least one of the units is not operating (i.e., binary not active and hence, equal to zero), there cannot be a material stream connecting these units. This logic valid cut forms a group of four objects $\left(m_{up}, j_{up}, i, m\right)$ and is especially helpful to reduce the search in branch-and-bound algorithms.

$$y_{m_{up},t} \geq y_{j_{up},i,t} \quad \forall \ \left(m_{up}, j_{up}, i, t\right) \tag{16}$$

$$y_{m,t} \geq y_{j_{up},i,t} \quad \forall \ \left(m, j_{up}, i, t\right) \tag{17}$$

Equation (18) ensures that at most one operational mode or task $op$ is simultaneously allowed for each physical unit $m$ at each time period $t$. Equation (19) correlates each unit-operation $m$ with its respective operational modes $op$, so that if the unit-operation is active, one operational mode must be selected at each time window. Equation (20) is the zero-downtime constraint to ensure the continuous selection of a mode of operation $op$. This is useful for unit-operations that must be continuously operated for long periods, such as CDUs.

$$\sum_{op} y_{m^{op},t} \leq 1 \ \ \forall \ \mathrm{t}, m \in \{M_{UN}\} \tag{18}$$

$$y_{m,t} = \sum_{op} y_{m^{op},t} \ \ \forall \ \mathrm{t}, m \in \{M_{UN}\} \tag{19}$$

$$\sum_{op} y_{m^{op},t} = 1 \quad \forall \ \mathrm{t}, m \in \{M_{CDU}\} \tag{20}$$

The temporal transition constraints represented by Equations (21) to (23) coordinate the operation of the semi-continuous blender units. The binary variable $y_{m,t}$ manages the variables related to the start-up ($zsu_{m,t}$), shut-down ($zsd_{m,t}$), and switch-over-to-itself ($zsw_{m,t}$) operations, which are respectively associated with starting, shutting down, or changing the operational mode of a blender unit-operation. The combination of Equations (21) to (23) also ensures these three variables to be discrete variables. In the proposed formulation, the temporal transition constraints are considered only for blender unit-operations because the processing unit-operations $m \in M_{UN}$ are assumed to continuously operate. However, such constraints can be useful for handling non-continuous or semi-continuous operations and to introduce scheduled maintenances for units.

$$y_{m,t} - y_{m,t-1} - zsu_{m,t} + zsd_{m,t} = 0 \ \ \forall \ m \in M_{BL}, t \tag{21}$$

$$y_{m,t} + y_{m,t-1} - zsu_{m,t} - zsd_{m,t} \ - 2zsw_{m,t} = 0 \quad \forall \ m \in M_{BL}, t \tag{22}$$

$$zsu_{m,t} + zsd_{m,t} + zsw_{m,t} \leq 1 \;\; \forall\, m \in M_{BL}, t \qquad (23)$$

Equations (24) and (25) are multi-use procedure constraints, in which downstream in-ports $i_{do}$ connected to the out-ports $j$ are limited by lower and upper bounds $\overline{mu}^{L}_{j,t}$ and $\overline{mu}^{U}_{j,t}$. This is helpful to control the number of simultaneous drawing operations allowed from a blender $m \in M_{BL}$ to its downstream tanks. Similarly, (26) and (27) impose bounds $\overline{mu}^{L}_{i,t}$ and $\overline{mu}^{U}_{i,t}$ for the upstream out-ports $j_{up}$ connected to the in-ports $i$, which is helpful to control the number of simultaneous filling operations to unit-operations $m \in M_{UN}$.

$$\sum_{i_{do}} y_{j,i_{do},t} \geq \overline{mu}^{L}_{j,t}\, y_{m,t} \;\; \forall \;\; (\mathrm{j}, m \in M_{BL}, t) \qquad (24)$$

$$\sum_{i_{do}} y_{j,i_{do},t} \leq \overline{mu}^{U}_{j,t}\, y_{m,t} \; \forall \; (\mathrm{j}, m \in M_{BL}, t) \qquad (25)$$

$$\sum_{j_{up}} y_{j_{up},i,t} \geq \overline{mu}^{L}_{i,t}\, y_{m,t} \;\; \forall \;\; (\mathrm{i}, m \in M_{UN}\,, t) \qquad (26)$$

$$\sum_{j_{up}} y_{j_{up},i,t} \leq \overline{mu}^{U}_{i,t}\, y_{m,t} \;\; \forall \; (\mathrm{i}, m \in M_{UN}\,, t) \qquad (27)$$

Equations (28) and (29) model the uptime constraints in which $\overline{upt}^{L}_{m}$ and $\overline{upt}^{U}_{m}$ are the respective uptime lower and upper bounds, $tt$ is an auxiliary time index, $\bar{t}_{end}$ is the time horizon length, and $\overline{\Delta t}$ is the time step. They impose operational time limits for unit-operations $m$, so that minimum and maximum amounts of time can be configured (i.e., upon starting an operation, a given unit cannot operate less than the uptime lower bound or more than the uptime upper bound). In Equation (28), if a unit is operating at time window $t$, there cannot have been more than one start-up operation within a past time range $\overline{upt}^{L}_{m}$; and if the unit is not operating, no start-up operation within a past time range $\overline{upt}^{L}_{m}$ is allowed. Equation (29) imposes that unit-operations m cannot continuously operate over a time range higher than the upper uptime bound of $\overline{upt}^{U}_{m}$. Equation (30) models a unit-operation uptime temporal aggregation cut. Given the uptime lower bound, number of time periods $\bar{n}_{p}$, and time step size, there is a limit in the number of possible start-up operations that can be performed over the time horizon length. Equation (30) reduces the search space of the start-up variable $zsu_{m,t}$, whereby the global optimum region is guaranteed to remain feasible after the cut. It is worth noting that Equations (28) to (30), combined with Equations (21) to (23), can properly manage the correlation between the variables $zsu_{m,t}$, $zsd_{m,t}$, and $zsw_{m,t}$ and the operational transitions for unit-operations $m$.

$$\sum_{tt=1|\mathrm{tt}<\mathrm{t}}^{\left(\frac{\overline{upt}^{L}_{m}}{\overline{\Delta t}}-1\right)} zsu_{m,t-tt} \leq y_{m,t} \;\;\; \forall \;\; (m, t \geq 2) \qquad (28)$$

$$\sum_{tt=t}^{(t+\frac{\overline{upt}_m^U}{\overline{\Delta t}})} y_{m,t} \le \frac{\overline{upt}_m^U}{\overline{\Delta t}} \quad \forall \ (m, t \le \bar{t}_{end} - \frac{\overline{upt}_m^U}{\overline{\Delta t}}) \tag{29}$$

$$\sum_{t} zsu_{m,t} \le (\frac{\bar{n}_p * \overline{\Delta t}}{\overline{upt}_m^L + \overline{\Delta t}}) \ \ \forall \ m \tag{30}$$

Equations (31) and (32) manage the minimum ($\overline{\Delta \mathrm{D}}_m^L$) and maximum ($\overline{\Delta \mathrm{D}}_m^U$) time between the latest filling and the next drawing operations, referred to as the fill-draw delay, for the upstream $j_{up}$ and downstream $i_{do}$ connections of a tank m$\in M_{TK}$. In Equation (31), whenever there is an incoming flow to a tank, any outgoing flow from that tank is only allowed after a waiting period of $\overline{\Delta \mathrm{D}}_m^L$. Additionally, Equation (31) ensures that filling and drawing operations cannot be simultaneously performed to the unit-operation $m \in M_{TK}$ at time window $t$, which represents a logistics constraint commonly used in the plant. In Equation (32), whenever there is an incoming flow to a tank, there must be at least one outgoing flow from that tank within the period $\overline{\Delta \mathrm{D}}_m^U$. These constraints are useful for handling materials that require a decantation period prior to their processing; and for avoiding long storage periods of materials.

$$y_{m,j_{up},i,t} + y_{m,j,i_{do},t+tt} \le 1 \ \ \forall \left(j_{up}, i, j, i_{do}\right), m \in M_{TK}, tt = 0..\overline{\Delta \mathrm{D}}_m^L, \\ t = 1..\bar{t}_{end}, t + tt \le \bar{t}_{end} \tag{31}$$

$$y_{m,j_{up},i,t} - \sum_{tt=1}^{\overline{\Delta \mathrm{D}}_m^U} y_{m,j,i_{do},t+tt} \le 0 \ \forall \left(j_{up}, i, j, i_{do}\right), \mathrm{m} \in M_{TK}, t + tt \le \bar{t}_{end} \tag{32}$$

Equations (33) to (36) represent the fill-to-full and draw-to-empty operations for tanks. Equation (33) introduces the logic variable $yd_{m,t}^{FILL}$, which is active when there is a filling operation. Similarly, Equation (34) introduces the logic variable $yd_{m,t}^{DRAW}$ for drawing operations. Equation (35) enforces that once a filling operation starts, there is a minimum inventory $\overline{xh}_{m,t}^{FILL}$ that must be reached before allowing any drawing operation from the same tank. Similarly, whenever there is a drawing operation, Equation (36) imposes a maximum inventory $\overline{xh}_{m,t}^{DRAW}$ that must be reached before any filling operation is allowed to the same tank. The coefficients $\overline{xh}_{m,t}^{FILL}$ and $\overline{xh}_{m,t}^{DRAW}$ are the fill-to-full and the draw-to-empty inventories, which may be static or dynamic over the time horizon. Equations (35) and (36) use the maximum inventory available $\overline{xh}_{m,t}^{Max}$ at each time window $t$ to control the auxiliary binary variables using big-M constraints (see Winston and Goldberg, 2004 for detailed information on big-M methods). Thus, filling operations continuously fill the tank at least until it reaches a minimum inventory $\overline{xh}_{m,t}^{FILL}$, and drawing operations must be carried out at least until the tank reaches a maximum inventory $\overline{xh}\,_{m,t}^{DRAW}$. This is especially helpful for reducing the number of operations involving tanks (i.e., for improved industrial operations, it is beneficial to perform fewer operations in larger amount rather than multiple operations in smaller amount). Such constraints are alternatives to the commonly employed changeover costs, which are arbitrary limitations to account for implicit costs of processing operations (e.g., opening valves).

$$y_{m,j_{up},i,t} \le yd^{FILL}_{m,t} \quad \forall \left(j_{up}, i\right), m \in M_{TK}, t \tag{33}$$

$$y_{m,j,i_{do},t} \le yd^{DRAW}_{m,t} \quad \forall \left(j, i_{do}\right), m \in M_{TK}, t \tag{34}$$

$$\left(\overline{xh}^{FILL}_{m,t-1} - xh_{m,t-1}\right) \le yd^{FILL}_{m,t} * \overline{xh}^{Max}_{m,t} \qquad \forall\, m \in M_{TK}, t \ge 2 \tag{35}$$

$$\left(\overline{xh}^{DRAW}_{m,t-1} - xh_{m,t-1}\right) \ge -yd^{DRAW}_{m,t} * \overline{xh}^{Max}_{m,t} \quad \forall\, m \in M_{TK}, t \ge 2 \tag{36}$$

To improve the accuracy of the MILP formulation, we utilize a linear approximation for nonconvex blending constraints proposed by Kelly et al. (2018). The extended quality amount of the property $p$ is considered as an in-out quantity and quality product or factors $f$ multiplied by volume flows $x$ around the blender unit-operations. To close the quantity-quality balance in the blender, the factor-flow $x_{j^{FAC},p,t}$ outgoing from the slack or surplus out-ports is considered in the balance in Equation (37) for the properties (e.g., specific gravity and sulfur concentration).

$$\sum_{i \in I_{BL}} \bar{f}_{i,p,t} \sum_{j_{up} \in J_{ST}} x_{j_{up},i,t} = \bar{f}_{j,p,t} \sum_{i_{do} \in I_{FT}} x_{j,i_{do},t} + x_{j^{FAC},p,t} \quad \forall\, j, j^{FAC}, p, t \tag{37}$$

For each property $p$ to be calculated in the blender $m \in M_{BL}$, amounts of feedstocks $\sum_{j_{up}} x_{j_{up},i,t}$ incoming from storage tanks to the blender through multiple in-ports $i$ with associated quality factors $\bar{f}_{i,p,t}$ counterbalance the quality of the total amount $\sum_{i_{do}} x_{j,i_{do},t}$ of blended material (to be sent to feed tanks) with associated quality factors $\bar{f}_{j,p,t}$. A slack or surplus variable for the factor-flow $x_{j^{FAC},p,t}$ is added to provide a degree of freedom for the quality balance. The factor in $j^{FAC}$is considered as unitary; therefore, the value of the slack or surplus factor-flow $x_{j^{FAC},p,t}$ represents the insufficient or exceeded quality amount of the factor-flow for each property $p$. For an upper bound of property specification, a slack or negative value is needed, so that $x_{j^{FAC},p,t} \le 0$. Similarly, a positive factor-flow or surplus ($x_{j^{FAC},p,t} \ge 0$) applies for a lower bound. Also, as the transformation from property to property index may change the signal of the number, to avoid infeasibilities, the factor-flow is modeled as $x_{j^{FAC},p,t} \le 0$ and $x_{j^{FAC},p,t} \ge 0$ for property indices.

### 3.8 Quality mathematical formulation: NLP refinery scheduling

The quality problem includes Equations (1) to (15) from the UOPSS flowsheet formulation, Equations (38) to (43) for the blending constraints, and Equations (44) to (46) for the mass balances and transformations of crude oil components $c$ into subsequent fractions. The NLP formulation cannot handle binary variables, so that the UOPSS equations are reformulated either by setting the binary variables to unity or by retrieving any previous logistics solution, whereby setting them to their respective optimal values.

For the quality balances throughout the network, we consider $p$ as property (component concentration, specific gravity, sulfur content) in which $v$ and $w$ are the volume- and weight-based properties, respectively. The variables $v_{j_{up},p,t}$ and $w_{j_{up},p,t}$ are the volume- and mass-based qualities of the upstream flow from the out-ports $j_{up}$ (of unit-operation $m_{up}$) incoming to the in-port $i$ (of unit-operation $m$), whereby $v_{j_{up},p=sg,t}$ refers specifically for specific gravity; and $v_{i,p,t}$ and $w_{i,p,t}$ are the volume- and mass-based qualities of inlet flows $i \in I$. It is similar for the downstream variables $v_{i_{do},p,t}$ and $w_{i_{do},p,t}$. Figure 4 illustrates the quality variables associated with the inlet and outlet ports. It is worth mentioning that the quality at a given port is also the quality of the material flow outgoing from this port (i.e., assumption of perfect mixing).

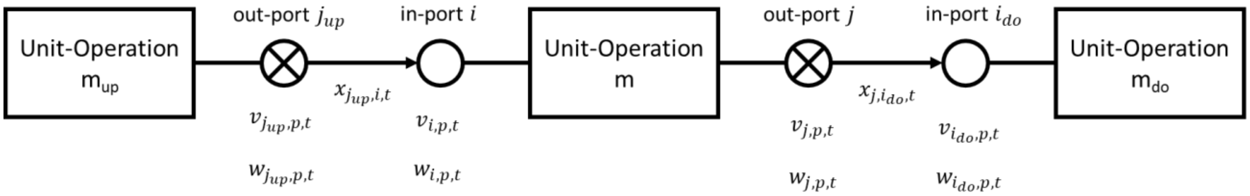


Figure 4: Quality variables for unit-operations and ports within the UOPSS representation.

Equation (38) is an overall mass-balance at the in-port $i \in \{I_{BL}, I_{UN}\}$ of mixer-based and splitter-based unit-operations $m \in \{M_{BL}, M_{UN}\}$. It introduces the volume-based properties $p \in P_v$ such as components (crude oil compositions) and densities (specific gravity). Similarly, Equation (39) is a property mass-balance (e.g., sulfur concentration), which introduces the mass-based properties $p' \in P_w$.

$$v_{i,p,t} \sum_{j_{up}} x_{j_{up},i,t} = \sum_{j_{up}} v_{j_{up},p,t} x_{j_{up},i,t} \quad \forall\, t, i \in \{I_{BL}, I_{UN}\}, p \in P_v \tag{38}$$

$$w_{i,p',t} \sum_{j_{up}} v_{j_{up},p,t}\, x_{j_{up},i,t} = \sum_{j_{up}} w_{j_{up},p',t}\, v_{j_{up},p,t}\, x_{j_{up},i,t} \quad \forall\, t, i \in \{I_{BL}, I_{UN}\}, p \in P_v, p' \in P_w \tag{39}$$

Equations (40) and (41) represent the quality balances for volume- and mass-based properties in tanks. They are a combination of Equation (14), that ensures consistency for the material-balance in tanks, with Equations (38) and (39) for the overall and component mass-balances when mixing two or more streams. The subsets in the summations involving the in-ports $i$ and out-ports $j$ in Equations (40) and (41) are also omitted, since this is valid for all $(i, m, j) \in M_{TK}$, by which $j_{up}$ represents the upstream out-ports of the connected unit-operations $m_{up}$ arriving in the in-port $i \in I_{TK}$, and $i_{do}$ represents the downstream in-ports of the connected unit-operations $m_{do}$ outgoing from the out-port $j \in J_{TK}$. The quality variable for the out-ports of a tank unit-operation ($m \in M_{TK}$) is the quality of the blend within the tank, as defined by Equations (42) and (43) for volume- and mass-based properties, respectively.

$$v_{m,p,t}\, xh_{m,t} = v_{m,p,t-1}\, xh_{m,t-1} + \sum_{j_{up}} v_{j_{up},p,t}\, x_{j_{up},i,t} - v_{m,p,t} \sum_{i_{do}} x_{j,i_{do},t} \tag{40}$$

$$\forall\, \mathrm{t}, (i,m,j) \in \{I_{TK}, M_{TK}, J_{TK}\}, p \in P_v$$

$$\begin{aligned} w_{m,p',t}\, v_{m,p,t}\, xh_{m,t} &= w_{m,p',t-1}\, v_{m,p,t-1}\, xh_{m,t-1} + \sum_{j_{up}} w_{j_{up},p',t}\, v_{j_{up},p,t}\, x_{j_{up},i,t} - \\ w_{m,p',t}\, v_{m,p,t} \sum_{i_{do}} x_{j,i_{do},t} \quad &\forall\, \mathrm{t}, (i,m,j) \in \{I_{TK}, M_{TK}, J_{TK}\}, p \in P_v, p' \in P_w \end{aligned} \tag{41}$$

$$v_{j,p,t} = v_{m,p,t} \;\; \forall\, (m,j) \in \{M_{TK}, J_{TK}\}, p \in P_v, t \tag{42}$$

$$w_{j,p',t} = w_{m,p',t} \;\; \forall\, (m,j) \in \{M_{TK}, J_{TK}\}, p' \in P_w, t \tag{43}$$

Equations (44), (45), and (46) respectively are the overall material-balance, overall mass-balance, and component mass-balance for the outputs of unit-operations based on distillation processes. Equation (44) converts the CDU throughputs $\left(\sum_{j_{up}} x_{j_{up},i_{CDU},t}\right)$ from the $i_{CDU}$ inlet into distillate yields $\left(\sum_{i_{do}} x_{j,i_{do},t}\right)$ outgoing from the CDU out-ports $j$ to the connected downstream in-ports $i_{do}$. This is performed for all crude oil components $c$ (from the crude oil assay) over all pre-defined distillation fractions $cut$ by introducing the rendering variable $re_{j,c,cut}^{yie}$, which provides the yield value for each set of $(j, c, cut)$. Similarly, Equations (45) and (46) respectively calculate the volume- and weight-based properties. The rendering variable for qualities, $re_{j,c,cut}^{p}$, can be accordingly adapted for volume-based qualities (e.g., $re_{j,c,cut}^{p=sg}$, in which the superscript $sg$ stands for specific gravity) and mass-based qualities (e.g., $re_{j,c,cut}^{p=sul}$, in which the superscript $sul$ stands for sulfur concentration).

$$\sum_{i_{do}} x_{j,i_{do},t} = \sum_{j_{up}} x_{j_{up},i_{CDU},t} \sum_{c} \sum_{cut} v_{i_{CDU},p=c,t} re_{j,c,cut}^{yie} \;\; \forall\, (j,t) \tag{44}$$

$$v_{j,p,t} \sum_{i_{do}} x_{j,i_{do},t} = \sum_{j_{up}} x_{j_{up},i_{CDU},t} \sum_{c} \sum_{cut} v_{i_{CDU},p=c,t} re_{j,c,cut}^{yie} re_{j,c,cut}^{p} \;\; \forall\, (j,t), p \in P_v \tag{45}$$

$$\begin{aligned} & v_{j,p,t} w_{j,p',t} \sum_{i_{do}} x_{j,i_{do},t} \\ &\quad = \sum_{j_{up}} x_{j_{up},i_{CDU},t} \sum_{c} \sum_{cut} v_{i_{CDU},p=c,t} re_{j,c,cut}^{yie}\, re_{j,c,cut}^{p} re_{j,c,cut}^{p'} \;\; \forall\, (j,t), p \\ &\quad \in P_v, p' \in P_w \end{aligned} \tag{46}$$

For material- and mass-balances of other unit-operations $m \in M_{UN}$, the mathematical equations depend on the model employed for calculating the input-output correlation. Such models can be based on fixed non-composition-based yields (e.g., regardless of the input flow composition, there is a fixed set of yields used for calculating each one of the pre-defined outputs), fixed composition-based yields (i.e., similar to the distillation model in Equations (44) to (46), in which the outlet flows of distillates depend on the composition of the crude oil blend incoming to the CDU), process-based yields (i.e., that depend on a set of processing variables such as pressure and temperature) which can be composition-based or not, or any other representation that can accurately estimate the unit-operation outputs. In this work, the remaining processing units consider simplified correlations based on non-composition yields or process-based yields that depend on processing variables such as temperature.

### 3.9 Objective function for the refinery scheduling optimization

The objective function for the refinery scheduling optimization in Equation (47) maximizes the product revenues by subtracting the feedstock costs and a performance term for the CDU throughputs to minimize process fluctuations.

$$Max\ Z = \sum_{t} \left( \sum_{m \in M_{PR}} \overline{price}_{m,t}\, x_{m,t} - \sum_{m \in M_{FE}} \overline{price}_{m,t}\, x_{m,t} - \sum_{m \in M_{CDU}} \overline{pen}_{m}\, x_{m,t}^{DEV} \right) \tag{47}$$

Where $Z$ is the objective function to me maximized over the entire time horizon with time periods $t$, $\overline{price}_{m,t}$ is the market value of each product $m \in M_{PR}$ or the cost of each feedstock $m \in M_{FE}$, $x_{m,t}$ are the amounts or flows of the respective products or feedstocks, $\overline{pen}_{m}$ is a penalty parameter introduced to manage the performance (i.e., for improved operating conditions) of distillation units $m \in M_{CDU}$, and $x_{m,t}^{DEV}$ is the deviation between the amount of crude processed in the distillation unit over consecutive time periods. In industrial operations, the operational conditions are expected to be as smooth as possible, without abrupt changes or variations that may compromise the process control within such highly dynamic environments. This performance term smooths the CDU throughputs $x_{m,t}$ ($m \in M_{CDU}$) by calculating the variations of its adjacent amounts and minimizing the linear deviation of the flow in consecutive time periods. Then, $x_{m,t}^{DEV} = |x_{m,t} - x_{m,t-1}|$. In order to achieve a linear formulation of such constraint, we propose the use of Equations (48) to (52) that introduce distinct terms $x_{m,t}^{DEV1}$ and $x_{m,t}^{DEV2}$ to represent penalties when the CDU flowrate increases or decreases, and constrain them accordingly. The upper bound $\bar{x}_{m,t}^{U}$ is the maximum flowrate of unit-operation $m \in M_{CDU}$ at time period $t$.

$$x_{m,t}^{DEV} = x_{m,t}^{DEV1} + x_{m,t}^{DEV2} \quad \forall \quad t \geq 2, m \in M_{CDU} \tag{48}$$

$$x_{m,t}^{DEV1} \geq x_{m,t} - x_{m,t-1} \quad \forall \quad t \geq 2, m \in M_{CDU} \tag{49}$$

$$x_{m,t}^{DEV2} \geq x_{m,t-1} - x_{m,t} \quad \forall \quad t \geq 2, m \in M_{CDU} \tag{50}$$

$$0 \leq x_{m,t}^{DEV1} \leq \bar{x}_{m,t}^{U} \quad \forall \quad t \geq 2, m \in M_{CDU} \tag{51}$$

$$0 \leq x_{m,t}^{DEV2} \leq \bar{x}_{m,t}^{U} \quad \forall \quad t \geq 2, m \in M_{CDU} \tag{52}$$

## 4. State-of-the-art modeling and optimization of industrial-scale refinery scheduling

The second main contribution of this work is the development of a state-of-the-art decision-making framework based on modeling and optimization algorithms to effectively solve industrial-scale refinery scheduling. Figure 5 provides a conceptual diagram that illustrates the core concepts employed in the proposed framework, which is introduced in Figure 6. In addition to the novel mathematical formulation developed for the refinery scheduling problem, the framework comprises a combination of computer-aided approaches such as phenomenological and temporal decompositions, linear reformulation of nonconvex blending equations, surrogate modeling techniques to represent complex processing units, a novel heuristic algorithm based on relaxation methods, a novel solving method to tackle NLP models, and rescheduling mechanisms.

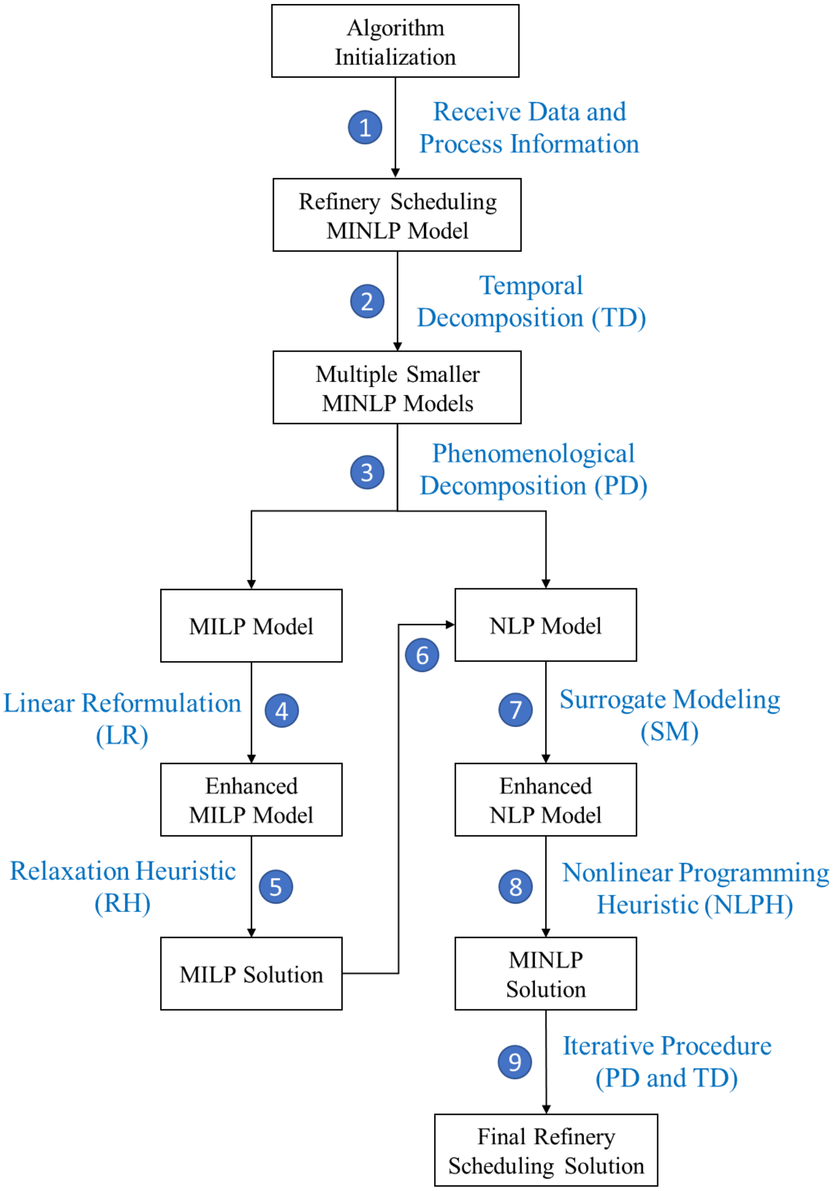

Algorithm Initialization
1
Receive Data and Process Information
Refinery Scheduling MINLP Model
2
Temporal Decomposition (TD)
Multiple Smaller MINLP Models
3
Phenomenological Decomposition (PD)
MILP Model
NLP Model
6
Linear Reformulation (LR)
4
7
Surrogate Modeling (SM)
Enhanced MILP Model
Enhanced NLP Model
Relaxation Heuristic (RH)
5
8
Nonlinear Programming Heuristic (NLPH)
MILP Solution
MINLP Solution
9
Iterative Procedure (PD and TD)
Final Refinery Scheduling Solution

Figure 5: Conceptual diagram for the proposed framework.

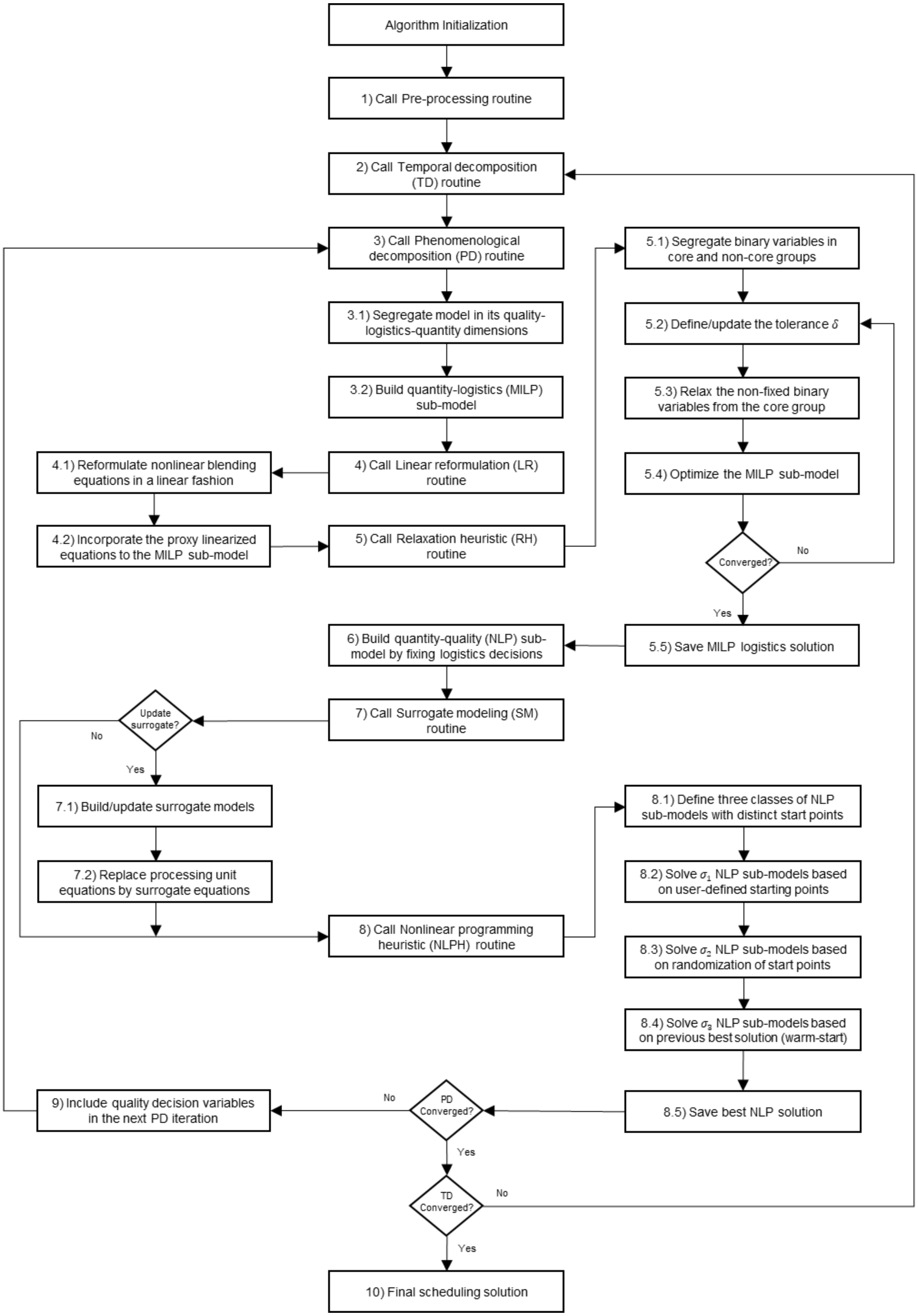

Algorithm Initialization
1) Call Pre-processing routine
2) Call Temporal decomposition (TD) routine
3) Call Phenomenological decomposition (PD) routine
3.1) Segregate model in its quality-logistics-quantity dimensions
3.2) Build quantity-logistics (MILP) sub-model
4) Call Linear reformulation (LR) routine
4.1) Reformulate nonlinear blending equations in a linear fashion
4.2) Incorporate the proxy linearized equations to the MILP sub-model
5) Call Relaxation heuristic (RH) routine
5.1) Segregate binary variables in core and non-core groups
5.2) Define/update the tolerance $\delta$
5.3) Relax the non-fixed binary variables from the core group
5.4) Optimize the MILP sub-model
Converged?
No
Yes
5.5) Save MILP logistics solution
6) Build quantity-quality (NLP) sub-model by fixing logistics decisions
7) Call Surrogate modeling (SM) routine
Update surrogate?
No
Yes
7.1) Build/update surrogate models
7.2) Replace processing unit equations by surrogate equations
8) Call Nonlinear programming heuristic (NLPH) routine
8.1) Define three classes of NLP sub-models with distinct start points
8.2) Solve $\sigma_1$ NLP sub-models based on user-defined starting points
8.3) Solve $\sigma_2$ NLP sub-models based on randomization of start points
8.4) Solve $\sigma_3$ NLP sub-models based on previous best solution (warm-start)
8.5) Save best NLP solution
PD Converged?
No
9) Include quality decision variables in the next PD iteration
Yes
TD Converged?
No
Yes
10) Final scheduling solution

Figure 6: Modeling and optimization framework for industrial-scale refinery scheduling

In the following we present a pseudo-code of the algorithm with further information on the capabilities employed within the framework.

```
1) Call Pre-processing routine
    Define parameters and hyperparameters required (e.g., scheduling parameters, optimization gap for commercial solvers, tolerances, convergence criteria, etc.).
    Read incumbent data to establish initial process conditions (e.g., actual inventories, qualities, etc.).
2) Call Temporal Decomposition (TD) routine
    2.1) Segregate the time horizon into multiple time chunks to be sequentially solved. The initial process conditions (i.e., inventories, qualities) of time chunk $tc + 1$ are the respective process conditions at the end of time chunk $tc$.
    3) Call Phenomenological Decomposition (PD) routine
        3.1) Segregate the original MINLP model according to its quantity-logistics-quality phenomena into MILP (quantity and logistics) and NLP (quantity and quality) sub-models.
        3.2) Build quantity-logistics (MILP) sub-model by fixing/neglecting quality information, which ensures the variables, constraints, and objective function to be necessarily linear.
        4) Call Linear Reformulation (LR) routine
            4.1) Reformulate nonlinear blending equations in a linear fashion by introducing hypothetical slack/surplus variables. Such a heuristic procedure is applied only for the crude oil blending equations.
            4.2) Incorporate the linearized equations into the MILP sub-model to provide more accurate predictions.
        5) Call Relaxation Heuristic (RH) routine
            5.1) Segregate binary variables into core and non-core groups.
            5.2) Define or update the tolerance $\delta$.
            5.3) Relax the non-fixed binary variables from core group.
            5.4) Optimize the MILP sub-model. If convergence criteria are met, go to 5.5. Otherwise, go back to 5.2.
            5.5) Save incumbent MILP logistics solution (binary variables).
        6) Build NLP sub-model
            6.1) Build the quantity-quality NLP sub-model by fixing the previously saved logistics solution in the MINLP model.
        7) Call Surrogate Modeling (SM) routine
            7.1) Build or update surrogates according to a pre-defined criterion (upon changes in the distillation feed recipe).
            7.2) Replace processing unit equations by surrogate equations in the NLP sub-model.
        8) Call Nonlinear Programming Heuristic (NLPH) routine
            8.1) Define three classes of NLP sub-models.
            8.2) Solve $\sigma_1$ NLP sub-model based on user-defined starting points (e.g., personal knowledge or experience).
            8.3) Solve $\sigma_2$ NLP sub-models based on randomization of starting points.
            8.4) Solve $\sigma_3$ NLP sub-models using a warm-start procedure (the starting points are randomly generated around the optimal values of the best NLP solution found so far).
            8.5) The NLP solutions are sorted according to a performance factor (chosen as the objective function value) and the solution with best performance is saved. While the PD routine does not converge, the algorithm goes to Step 9 and subsequently returns to Step 3. While the TD routine does not converge, the algorithm returns to Step 2.
    9) Include quality information in the next PD iteration
```

While the phenomenological decomposition does not converge, quality decision variables from the incumbent NLP solution are saved and updated into the MILP problem of the next PD iteration in Step 3.

**10) Final Scheduling Solution**

Upon convergence of the temporal decomposition, the algorithm provides a solution for the original MINLP problem. Within each PD iteration, the best solution is selected and saved. Each TD iteration originates a piece of a solution, whereas the final scheduling solution is comprised of one or multiple pieces depending on the time horizon segregation in the TD routine.

Some considerations with respect to the methods used in the framework are further discussed in the following. Prior to the algorithm initialization, we highlight the importance of coherent process topology as well as the selection of additional features such as model representation, scheduling parameters, and tuning hyperparameters that are relevant to achieve more efficient formulations.

*UOPSS Representation*: It has been used in many works (Kelly et al., 2017; Brunaud et al., 2020; Zyngier et al., 2018) for large-scale applications and provides efficient capabilities for handling large number of variables and constraints. In fact, despite the introduction of auxiliary variables and constraints (which are not mandatory), it is apparent in many problems that the UOPSS leads to more tractable formulations that can be more easily and quickly solved by optimization solvers.

*Scheduling Parameters*: An important consideration concerns the parameters to be determined prior to solving the scheduling problem, including the time horizon length, time-step size, number of time periods. Extensive testing was performed in this work with distinct scheduling parameters to evaluate the interplay among model size, optimization tractability, and solution quality.

*Rescheduling*: The scheduling parameters play a key role in providing easier and smoother integration of the mathematical optimization core with the schedule implementation in the plant. The proposed framework enables a rescheduling mechanism that exploits the re-optimization in an online moving horizon fashion, in which plant data define the initial process conditions for the incumbent optimization. The idea of rescheduling has been previously discussed in Franzoi et al. (2021a) and provides significant operational benefits and economic gains.

*Algorithm hyperparameters*: There are hyperparameters to be tuned in order to achieve efficient optimization of such large-scale and complex formulations. They are mostly associated with the modeling and solving strategies employed within the framework. This includes the convergence criteria for the phenomenological decomposition, time discretization in the temporal decomposition, segregation of variables and choice of tolerance in the relaxation heuristic, surrogate updating criteria in the surrogate modeling routine, MILP optimization gap, and the number of sequential NLP optimizations,. In the following we present a list of hyperparameters with further considerations and proper explanation for their respective choices.

- $\rho^{PD} = 3$: First convergence criterion (maximum number of iterations) in the phenomenological decomposition algorithm.
- $\phi^{PD} = 0.1\%$: Second convergence criterion (minimum solution improvement from iteration $t$ to iteration $t + 1$) in the phenomenological decomposition algorithm.
- $\tau^{TD} = [1, 10]$: Number of time chunks from the temporal decomposition. Distinct values of $\tau^{TD}$ are considered over multiple scenarios.

- $\delta^{NLP}$ = {0.025, 0.05, 0.10}: Range used for the starting points of decision variables in the third class of NLP optimizations.
- $\delta^{RH}$ = {0.01, 0.05, 0.10}: Tolerance in the relaxation heuristic. It starts assuming an initial value of 0.01, and reaches 0.10 over three iterations.
- $\sigma^{SM} = \{0,1\}$: Binary variable to represent the need for surrogate model updating. This is triggered whenever there is arrival of new feedstocks, change in the qualities of feedstocks, or changes in the process conditions that impact the respective processing unit predictions.
- $\varepsilon^{MILP} = 1.0\%$: Optimization gap in the MILP optimizations.
- $\eta^{NLP} = \{1,3,3\}$: Number of sequential optimizations in each class of NLP sub-models.

*Temporal Decomposition*: This is an efficient size-reduction technique, which should be carefully employed to avoid compromising the solution quality. Extensive testing was performed to understand the interplay among the model reduction from the temporal decomposition, the gains in terms of computational tractability, and the loss in terms of solution quality. This provides meaningful insights on how we can properly and effectively employ this method.

*Phenomenological Decomposition*: This is similar to the heuristic developed in Menezes et al. (2015), which is shown to be appropriate for crude oil refinery scheduling problems. The MINLP model is partitioned and sequentially solved within an iterative procedure, which provides a solution approach for such an intractable large-scale nonconvex scheduling problem.

*Linear Reformulation*: This linear reformulation of nonconvex blending equations has been shown in previous work (Kelly et al., 2018) to significantly improve the solution quality of the MILP sub-models and to reduce the MILP-NLP decomposition gap. It provides quality information to the MILP sub-model, which leads to more realistic logistics solutions.

*Relaxation Heuristic*: We propose a relaxation heuristic (RH) to tackle large-scale MIP problems. The number of binary variables is significantly reduced whereas sequentially solving multiple semi-relaxed MIP subproblems. This is a useful size-reduction method that requires careful calibration and consideration. Extensive testing was performed for the heuristic design and parameter calibration to ensure good cost-effectiveness in terms of achieving significant computational savings while maintaining the solution quality. The proposed relaxation heuristic is illustrated in Figure 7.

The algorithm initializes segregating the binary variables from the MIP formulation (often associated with feedstocks, units, tanks, operations, and logistics constraints) into a core group A that includes setup and start-up variables, and a non-core group B that concerns logistic constraints. At each optimization, the non-fixed binary variables from core-group A are relaxed and the model is optimized. From the optimal solution, the binary variables from core-group A within a tolerance $\delta$ are fixed to 1 and remain unchanged over future optimizations. While the convergence criteria for $\delta$ are not met, the algorithm iteratively updates the tolerance $\delta$ and re-optimizes the problem, until meeting any of the following criteria: a) δ reaches its upper bound; or b) all relaxed binaries are fixed. When the convergence criteria for $\delta$ are met, the problem is optimized one last time to determine the remaining decision variables in the problem.

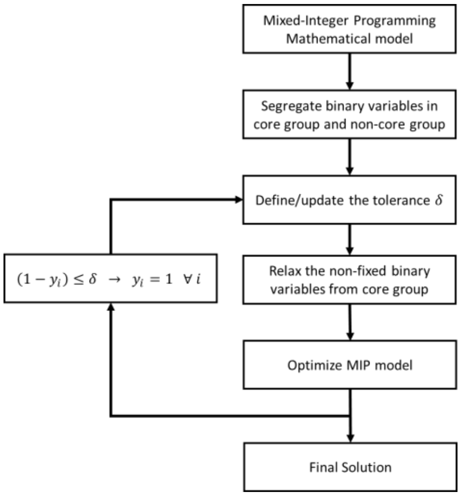

Mixed-Integer Programming Mathematical model
Segregate binary variables in core group and non-core group
Define/update the tolerance $\delta$
$(1 - y_i) \leq \delta \ \rightarrow \ y_i = 1 \ \forall \, i$
Relax the non-fixed binary variables from core group
Optimize MIP model
Final Solution

Figure 7: Algorithm for the proposed relaxation heuristic.

*Surrogate Modeling*: A surrogate model building methodology from Franzoi et al. (2020) is used to build surrogates for the crude distillation unit, which is the most important processing unit in crude oil refinery operations. The method estimates the yields and properties of distillates using the crude oil assay and the hypothetical improved swing-cuts as input training variables. They are built through measurement feedback by using simulated data. The method is shown to be efficient and provides highly reliable surrogates, which can be properly integrated into scheduling problems with minimal increase in the simulation and optimization effort and data requirements. In addition, the surrogates can be updated according to relevant criteria for enhanced predictions.

*Nonlinear Programming Heuristic*: It is well known that starting points for decision variables play a key role in nonlinear optimization problems, especially when there are highly nonlinear or nonconvex terms. This affects the effort needed to find optimal solutions and typically leads to poor or infeasible solutions when the starting points are not properly chosen. Because of the complexity of the formulation addressed herein, which might often lead to poor convergence, we developed a heuristic routine for solving the NLP sub-models. Concepts of exploration and exploitation search procedures (see Franzoi et al., 2021c) are employed to systematically manage the selection of starting points for the decision variables. The idea consist of an initial wide search across the entire space to identify a potentially good source of solutions, and then subsequently look for better solutions in this particular region.

Within the proposed NLPH routine, three classes of NLP sub-models are defined, in which the difference among them is the selection of starting points. The hyperparameters that represent the number of instances to be optimized for each class of NLP sub-problem respectively are $\eta_1^{NLP} = 1$, $\eta_2^{NLP} = 3$, and $\eta_3^{NLP} = 3$. The first class considers starting points defined by the user (typically the scheduler). If such information is missing, the algorithm retrieves the last set of optimal starting points used in the previous optimization (if available), or simply skips this step. The solution quality from the NLP optimization strongly relies on how good the starting points are, which represents a potential to quickly achieve fairly good solutions in early stages of the algorithm.

The second class regards the exploration search, in which a combination of fixed and randomized values is employed. The user has flexibility to determine fixed values for any decision variables based on plant requirements, personal knowledge, or other reliable source of information. The starting point $\overline{dv}^{SP}$ for each remaining variable $dv$ is calculated according to Equation (53), which uniformly lies between the lower and upper bounds $\overline{dv}^{L}$ and $\overline{dv}^{U}$ by generating a random number $rand = [0,1]$. A total of $\eta_2^{NLP}$ optimizations are performed.

$$\overline{dv}^{SP} = rand\left(\overline{dv}^{U} - \overline{dv}^{L}\right) + \overline{dv}^{L} \tag{53}$$

The best solution found from the previous optimizations is saved, which comprises the value of each decision variable $dv^{opt}$. The third class relies on a hybrid idea of exploitation with randomized search, in which the best solution found provides insights for further exploring the optimization space. The starting point of each variable $dv$ is randomly chosen to be as in Equation (54) but subject to the hard bounds in Equation (55). The random number $rand = [0,1]$ is multiplied by a hyperparameter $\delta^{NLP} = \{0.025, 0.05, 0.10\}$ associated with each of the $\eta_3^{NLP} = 3$ optimizations performed for the third class of NLP sub-models.

$$\overline{dv}^{SP} = dv^{opt} * (1 + \delta^{NLP}(2 * rand - 1)) \tag{54}$$

$$\overline{dv}^{L} \leq \overline{dv}^{SP} \leq \overline{dv}^{U} \tag{55}$$

The large degree of nonconvexity typically found in refinery scheduling problems leads to a large number of local optima. This solving procedure provides an efficient fashion to search the neighborhood of the incumbent solution as an alternative to tackle highly nonconvex problems.

## 5. Example: Large-scale refinery scheduling optimization

The proposed framework is employed to solve a large-scale integrated refinery scheduling problem which comprises the entire refinery value chain, i.e., crude oil scheduling, production network, and product blending network. Specifically, the problem topology considers accurate blending and processing networks that are coherent with the physical configuration in the refinery process. This includes the modeling of blending operations with continuous blender units instead of batch mixtures, the inclusion of tanks prior processing units to provide scheduling flexibility, cascaded distillation network with detailed crude-oil assay, a complex processing network of unit-operations, and product blending network for production of fuels.

### 5.1 Problem statement

The refinery network is illustrated in Figure 2. The crude oil scheduling segment includes 12 crude oil feedstocks, 12 storage tanks, 2 blenders, 4 feed tanks, and a distillation network with 5 towers, namely, two atmospheric distillation units, vacuum distillation unit, flash distillation unit, and debutanizer unit. A crude oil assay is used for the calculation of crude oil yields, compositions, and properties, in which a micro-cut distribution (segmented into 10 $°C$ fractions) is considered. Details on the micro-cut calculation and distribution can be found in Menezes et al. (2013). The production network includes fluid catalytic cracking unit, hydrotreaters for coker light naphtha, light cracked naphtha, and diesel, delayed coking unit, two debutanizers, superfractionator, and catalytic reformer unit. The product blending network includes three blenders for gasoline, diesel, and fuel oil to mix intermediate process streams with boosters for final product specification. A comprehensive explanation of the refinery network, discussing the resources, products, unit-operations, and production process can be found in the Supplementary Material B.

The mathematical formulation for the scheduling problem requires information related to:

- Inventory capacity of tanks
- Flowrate capacity of processing units
- Feedstocks (availability and market value)
- Products (demand, market value, and quality specifications)
- Initial inventories and qualities of materials throughout the plant
- Crude oil assay that contains quality information of compositions, yields, and properties over all micro-cut fractions of feedstocks.

The values adopted for such parameters are used in Table 1 (inventory capacity of tanks), Table 2 (flowrate capacity of processing units), Table 3 (information of feedstocks), Table 4 (information of products), and Table 5 (initial inventories and qualities).

Table 1: Parameters for the inventory capacity of tanks.

| Unit name | Unit description | Capacity ($Kbbl$) |
|---|---|---|
| CLNHTTK | Feed tank for the coker light naphtha hydrotreating unit | [0, 100] |
| COKETK | Coke storage tank | [0, 100] |
| CRUTK | Feed tank for the catalytic reformer unit | [0, 100] |
| DCUTK | Feed tank for the delayed coker unit | [0, 100] |
| DHTTK | Feed tank for the diesel hydrotreating unit | [0, 100] |
| DTK | Diesel storage tank | [0, 100] |
| DTK2 | Feed tank for the debutanizer unit (DEB2) | [0, 100] |
| DTK3 | Feed tank for the debutanizer unit (DEB3) | [0, 100] |
| FCCUTK | Feed tank for the fluid catalytic cracking unit | [0, 100] |
| FT1 to FT4 | Crude oil feed tanks | [0, 200] |
| FTK | Fuel oil storage tank | [0, 100] |
| GTK | Gasoline storage tank | [0, 100] |
| LCNHTTK | Feed tank for the light cracked naphtha hydrotreating unit | [0, 100] |
| SFRTK | Feed tank for the superfractionation unit | [0, 100] |
| ST1 to ST12 | Crude oil storage tanks | [0, 200] |
| VT1 to VT2 | Feed tank for the vacuum distillation unit | [0, 200] |

Table 2: Parameters for the flowrate capacity of processing units.

| Unit name | Unit description | Flowrate Capacity ($Kbbl/day$) |
|---|---|---|
| BL1 and BL2 | Crude oil blender units | [0, 100] |
| CDU1 and CDU2 | Crude oil distillation units | [50, 80] |
| CLNHT | Coker light naphtha hydrotreating unit | [0, 50] |
| CRU | Catalytic reformer unit | [0, 50] |
| DBL | Diesel blender unit | [0, 100] |
| DCU | Delayed coker unit | [0, 80] |
| DEB1 | Debutanizer unit | [0, 20] |
| DEB2 | Debutanizer unit | [0, 80] |
| DEB3 | Debutanizer unit | [0, 50] |
| DHT | Diesel hydrotreating unit | [0, 80] |
| FBL | Fuel oil blender unit | [0, 100] |
| FCCU | Fluid catalytic cracking unit | [40, 80] |
| FDU | Flash distillation unit | [50, 100] |
| GBL | Gasoline blender unit | [0, 100] |
| LCNHT | Light cracked naphtha hydrotreating unit | [0, 50] |
| SFR | Superfractionation unit | [0, 20] |
| VDU | Vacuum distillation unit | [60, 90] |

Table 3: Parameters for the availability, quality, and market value of feedstocks.

| Unit name | Unit description | Availability ($Kbbl$) | Specific gravity ($kg/m^3$) | Sulfur concentration ($weight$ %) | Market value ($\$/bbl$) |
|---|---|---|---|---|---|
| AR_IMP | Atmospheric residue (imported) | 200 | 0.9000 | 0.5000 | -35 |
| CO1 | Crude oil C1 | 500 | 0.8894 | 0.5025 | -65 |
| CO2 | Crude oil C1 | 500 | 0.8894 | 0.5025 | -51 |
| CO3 | Crude oil C2 | 500 | 0.9291 | 0.6077 | -56 |
| CO4 | Crude oil C2 | 500 | 0.9291 | 0.6077 | -71 |
| CO5 | Crude oil C3 | 500 | 0.9162 | 0.2310 | -77 |
| CO6 | Crude oil C3 | 500 | 0.9162 | 0.2310 | -70 |
| CO7 | Crude oil C4 | 500 | 0.8766 | 0.3443 | -66 |
| CO8 | Crude oil C4 | 500 | 0.8766 | 0.3443 | -50 |
| CO9 | Crude oil C5 | 500 | 0.7990 | 0.0487 | -57 |
| CO10 | Crude oil C5 | 500 | 0.7990 | 0.0487 | -70 |
| CO11 | Crude oil C6 | 500 | 0.9014 | 0.5188 | -75 |
| CO12 | Crude oil C6 | 500 | 0.9014 | 0.5188 | -72 |
| ETOH | Ethanol (imported) | 100 | 0.7800 | 0.0000 | -100 |
| ISOOCTANE | Isooctane (imported) | 100 | 0.6900 | 0.0000 | -500 |

Table 4: Parameters for the demand, quality specification, and market value of products.

| Unit name | Unit description | Minimum Production (K$bbl$/day) | Specific gravity ($kg/m^3$) | Sulfur concentration ($weight$ %) | Market value ($\$/bbl$) |
|---|---|---|---|---|---|
| BUTANE | Butane storage sink | 10 | 0.90 | 0.20 | 55 |
| COKE | Coke storage sink | - | 0.90 | 0.90 | 30 |
| DIESEL | Diesel storage sink | 60 | 0.90 | 0.01 | 110 |
| FG | Fuel gas storage sink | - | 0.90 | 0.00 | 10 |
| FO | Fuel oil storage sink | 50 | 1.10 | 0.60 | 70 |
| GASOLINE | Gasoline storage sink | 40 | 0.90 | 0.01 | 140 |
| LPG | Liquefied petroleum gas storage sink | - | 0.90 | 0.01 | 50 |
| LN | Light naphtha storage sink | - | 0.90 | 0.01 | 115 |
| PROPANE | Propane storage sink | - | 0.90 | 0.01 | 55 |
| PROPYLENE | Propylene storage sink | - | 0.90 | 0.01 | 60 |

Table 5: Parameters for the initial inventories and qualities of tanks and pools.

| Unit name | Initial inventory ($m^3$) | Crude oil composition | Specific gravity ($kg/m^3$) | Sulfur concentration ($weight$ %) |
|---|---|---|---|---|
| CLNHTTK | 10 | - | 0.7200 | 0.1000 |
| COKETK | 10 | - | 0.9000 | 0.9000 |
| CRUTK | 10 | - | 0.8000 | 0.1000 |
| DCUTK | 10 | - | 0.9000 | 0.2000 |
| DHTTK | 10 | - | 0.8500 | 0.3200 |
| DTK | 10 | - | 0.8000 | 0.0200 |
| DTK2 | 10 | - | 0.8000 | 0.0200 |
| DTK3 | 10 | - | 0.8000 | 0.0200 |
| FCCUTK | 10 | - | 0.9000 | 0.7000 |
| FT1 | 150 | C1 | 0.8894 | 0.5025 |
| FT2 | 10 | C1 | 0.8894 | 0.5025 |
| FT3 | 150 | C1 | 0.8894 | 0.5025 |
| FT4 | 10 | C1 | 0.8894 | 0.5025 |
| FTK | 10 | - | 1.0000 | 0.7000 |
| GTK | 10 | - | 0.4000 | 0.0100 |
| LCNHTTK | 10 | - | 0.8000 | 0.1000 |
| SFRTK | 10 | - | 0.8500 | 0.0100 |
| ST1 | 200 | C1 | 0.8894 | 0.5025 |
| ST2 | 200 | C1 | 0.8894 | 0.5025 |
| ST3 | 10 | C2 | 0.9291 | 0.6077 |
| ST4 | 10 | C2 | 0.9291 | 0.6077 |
| ST5 | 100 | C3 | 0.9162 | 0.2310 |
| ST6 | 10 | C3 | 0.9162 | 0.2310 |
| ST7 | 10 | C4 | 0.8766 | 0.3443 |
| ST8 | 200 | C4 | 0.8766 | 0.3443 |
| ST9 | 10 | C5 | 0.7990 | 0.0487 |
| ST10 | 10 | C5 | 0.7990 | 0.0487 |
| ST11 | 200 | C6 | 0.9014 | 0.5188 |
| ST12 | 200 | C6 | 0.9014 | 0.5188 |
| VT1 | 50 | C1 | 0.8894 | 0.5025 |
| VT2 | 10 | C1 | 0.8894 | 0.5025 |

### 5.2 Computational experiments

The mathematical formulation is built in the commercial modeling and solving platform IMPL (Industrial Modeling & Programming Language). The MILP formulation is solved using the commercial optimization solver Gurobi 9.1.1. The NLP formulation is first linearized through a sequential linear programming (SLP) algorithm in a pre-processing stage, followed by the optimization using Gurobi. The machine used is an Intel Core i7 with 2.7 GHz and 16 GB RAM. The total computational time allotted for optimization is 36,000 seconds.

The framework is tested over a large-scale scheduling optimization problem based on the refinery network in Figure 2. Multiple scenarios are designed and solved to investigate the computational tractability and solution quality under distinct conditions, i.e., different model sizes, scheduling parameters, and framework hyperparameters. While larger time horizon length and small time-step size are anticipated to improve profitability, they scale with the model size and therefore should be carefully tuned to provide timely solutions. The scenarios are shown in Table 6 with their respective total time horizon length, segmented time horizon length, and time-step size. The temporal decomposition routine segregates the total time horizon in smaller time chunks. The objective function for the profit maximization and the central processing unit (CPU) time are reported as well. Among the scenarios proposed, the largest MINLP formulation considers time horizon of 30 days and time-step of 2 hours, in a total of 360 time periods. There are around 70,000 continuous and 50,000 binary variables, 180,000 constraints and 90,000 degrees of freedom.

Table 6: Scenarios proposed for the refinery scheduling problem.

| Scenario | Time Horizon (days) | Segmented Time horizon (days) | Time-Step (hours) | Objective Function ($10^3$ US$) | CPU (s) |
|---|---|---|---|---|---|
| 1a | 10 | 10 | 24 | 100,079 | 23 |
| 1b | 10 | 5 | 24 | 98,351 | 14 |
| 2a | 15 | 15 | 24 | 130,969 | 551 |
| 2b | 15 | 5 | 24 | 123,707 | 22 |
| 3a | 20 | 20 | 24 | 157,786 | 1,346 |
| 3b | 20 | 5 | 24 | 150,718 | 30 |
| 4a | 25 | 25 | 24 | 188,103 | 2678 |
| 4b | 25 | 5 | 24 | 175,698 | 41 |
| 5a | 30 | 30 | 24 | 218,270 | 4,498 |
| 5b | 30 | 5 | 24 | 206,143 | 58 |
| 5c | 30 | 3 | 24 | 194,222 | 44 |
| 6a | 30 | 30 | 12 | 233,760 | 36,000+ |
| 6b | 30 | 5 | 12 | 223,060 | 851 |
| 6c | 30 | 3 | 12 | 207,905 | 437 |
| 7a | 30 | 30 | 8 | 244,615 | 36,000+ |
| 7b | 30 | 5 | 8 | 239,152 | 1,806 |
| 7c | 30 | 3 | 8 | 235,800 | 730 |
| 8a | 30 | 30 | 4 | 252,754 | 36,000+ |
| 8b | 30 | 5 | 4 | 245,002 | 3,265 |
| 8c | 30 | 3 | 4 | 241,780 | 2,285 |
| 8d | 30 | 1 | 4 | 225,488 | 254 |
| 9a | 30 | 30 | 2 | 255,899 | 36,000+ |
| 9b | 30 | 5 | 2 | 253,990 | 5,108 |
| 9c | 30 | 3 | 2 | 252,568 | 3,734 |
| 9d | 30 | 1 | 2 | 234,217 | 760 |

The original MINLP formulation is large in size and highly complex given the high degree of nonlinearities and nonconvexities, mostly from the blending equations. The largest scenario considers hundreds of thousands of variables and constraints in a nonconvex MINLP formulation, which is successfully solved by the proposed framework in reasonable computational time.

Table 6 indicates flexibility regarding the selection of appropriate scheduling parameters according to the computational limitations and requirements towards online scheduling strategies. The selection of scheduling parameters is key for achieving proper trade-off. For example, an optimal solution of $239M was found in 30 minutes using time-steps of 8 hours; and an optimal solution of $252M was found in 60 minutes using time-steps of 2 hours. If the total optimization time requires too much effort (e.g., above $2h$), online scheduling procedures may not be appropriate. In this context, Scenarios 9a, 9b, and 9c are potential solution candidates to be implemented in the plant. In general, large time horizon and small time-steps provided significant improvements in the solution quality.

Some results from Table 6 are plotted for easier visualization and comparison. Figure 8 illustrates the computational time and the objective function to be maximized over the increase in the total time horizon length for Scenarios 1a, 2a, 3a, 4a, 5a. Figure 9 illustrates the computational time (a) and the objective function to be maximized (b) over the decrease in the time-step size for all scenarios with 30-days time horizon. The black line with circular markers represents Scenarios 5a, 6a, 7a, 8a, 9a (no time segmentation); the blue line with triangle markers represents Scenarios 5b, 6b, 7b, 8b, 9b (time horizon segmentation of 5 days for each subproblem); and the red line with square markers represents Scenarios 5c, 6c, 7c, 8c, 9c (time horizon segmentation of 3 days for each subproblem).

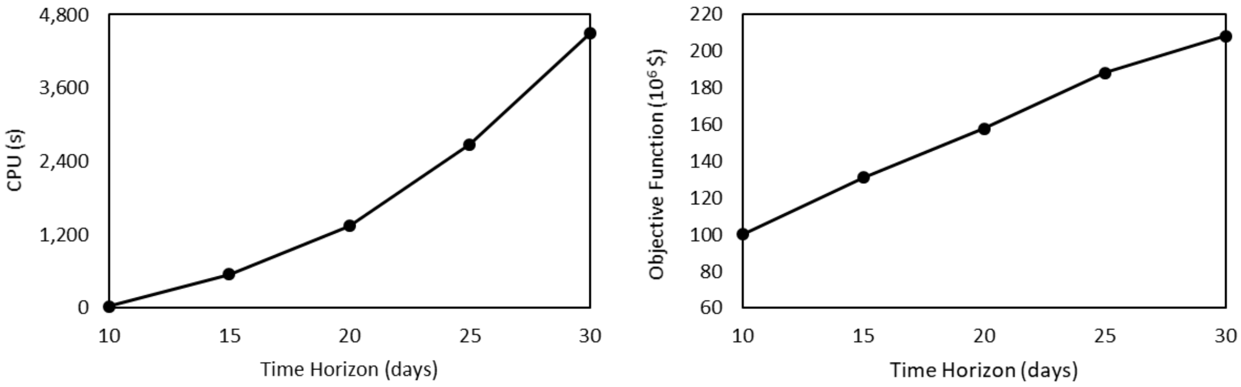


Figure 8: Solution statistics for the: a) computational time and b) objective function over the variation of the time horizon length.

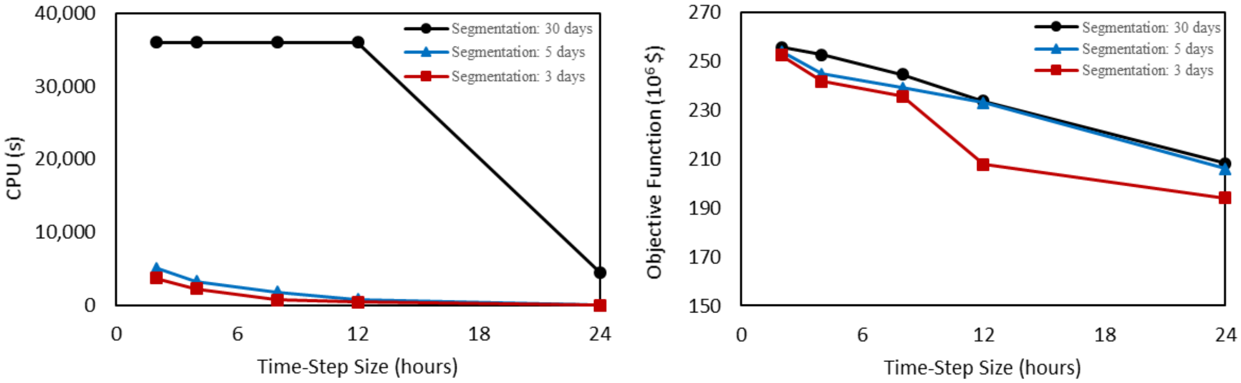


Figure 9: Solution statistics for the: a) computational time and b) objective function over the variation of the time-step size.

The results shown in Table 6 and Figures 8 and 9 illustrate how the scheduling solution is affected by decision-making methods and the calibration of parameters and hyperparameters, which is particularly important toward establishing metrics that are relevant for the application and implementation in industrial processes. This includes identifying a proper interplay among the characteristics of the problem to be solved, the scheduling parameters (e.g., time representation, time horizon length, time-step size), and a combination of mathematical optimization techniques. The algorithmic and computational methods comprised in the decision-making framework developed herein enable solving otherwise intractable large-scale scheduling problems, whereby providing reasonably quick solutions to be carried out in a rescheduling fashion.

Franzoi and Menezes (2022) provide some quantitative insights on the impact of scheduling parameters and rescheduling strategies for typical blend scheduling problems. The scheduling horizon is critical for anticipating spot market opportunities and providing more flexible solutions to handle distinct scenarios that may happen in the plant (e.g., demand peaks, maintenance of units). In this context, the selection of large time horizon length (e.g., 30 days) for scheduling applications is especially relevant. The selection of the time-step size is particularly important from an operational perspective. The comparison among Scenarios 5a to 9a, 5b to 9b, and 5c to 9c, indicates significant improvements in the scheduling solution as the time-step size is decreased from $24h$ to $2h$. Small time-steps provide improved decision-making from the additional degrees of freedom in the optimization, which results in better management of resources and production profile.

The importance of employing small time-steps also affects re-optimization features toward achieving online scheduling solutions. The solution from Scenario 9b (in which a $2h$ time-step scheduling model is solved in less than $2h$) allows addressing online scheduling approaches, whereby the solution is continuously updated within a moving horizon fashion. In such an approach, at certain instants of time (e.g., every $2h$ or upon event triggering), the incumbent status of the system is assessed, additional information including noises and disturbances and evaluated and added to the model, and the system is re-optimized (similarly to the method developed in Franzoi et al., 2021a). Such insights highlight the usefulness of efficient methodologies to tackle industrial problems whereas providing solution approaches that are implementable and can be further adjusted to meet the industrial needs in terms of operations, constraints, and requirements.

Importantly, the solution approach should be chosen according to the specific needs of the industry or process and the scheduling decision-making core should be coordinated with the plant operations. For example, small time-steps are only useful if the solution can be properly implemented in a timely manner; large time horizon requires reliable information about the market and operations over the upcoming near future; and online scheduling strategies are leveraged with a sufficient degree of automation and measurement systems. It is also worth highlighting the importance of such features for the long-term investment planning, which concerns a multi-level and multi-scope decision-making. The optimization scenarios solved in this work provide meaningful insights associated with the benefits of better scheduling, which is especially helpful in operational and economic analyses toward smarter investment planning. Such considerations are fundamental to further enhance the performance of industrial processes.

## 6. Main remarks and contributions of this work

There has been recent research and technological advancements in computer power, solution algorithms, and decision-making with applications in computer-aided and process systems engineering. Toward further breakthroughs in these field, we address the modeling and optimization of industrial-scale refinery scheduling applications. The main contributions of this work are threefold. First, we discuss the state-of-the-art limitations of industrial-scale refinery scheduling and we provide an overview of efficient modeling and algorithmic methods for tackling this problem. Second, we present a novel mathematical model that that accurately represents refinery scheduling operations considering a full-scope refinery topology and relevant operational constraints. Third, we propose a novel decision-making framework based on mathematical modeling and optimization capabilities to solve industrial-scale nonconvex MINLP refinery scheduling systems. To the best of our knowledge, this is the first work that proposes tackling and solving refinery scheduling problems in such a scope, complexity, and size.

The framework leverages the use of mathematical optimization and algorithmic methods by combining modeling approaches (process design, model decompositions), solving strategies (rescheduling procedures, heuristic algorithms), and machine learning regression (reduced-order models). It successfully solves an industrial-size refinery scheduling problem formulated as a nonconvex mixed-integer nonlinear programming (MINLP) model, providing more efficient scheduling operations and higher profitability. The formulation is coherent with industrial applications in terms of operational constraints, complexity, and size. From the results presented herein, we anticipate promising research on the development of solution methods that enhance industrial decision-making in the process systems engineering field.

### 6.1 Future outlook on industrial-scale refinery scheduling optimization

We highlight six features that should be considered toward the future generation of scheduling technology for improved refinery decision-making. From a modeling perspective, this includes the consideration of an integrated refinery scheduling network, realistic mathematical formulation based on plant requirements, and accurate process-unit predictions. From a solving perspective, this includes optimization decision-making, consideration of scheduling parameters, and re-optimization mechanisms.

1. **Integrated Network:** Simultaneously solving an integrated problem including crude oil scheduling, processing network, and product blending network in the same formulation, exploits additional degrees of freedom and leads to improved decision-making.
2. **Realistic Formulation:** The scheduling formulation needs to be coherent with the operational requirements and limitations in the plant. Realistic models lead to realistic schedules that are more easily and smoothly implemented.
3. **Accurate Predictions:** Process unit models need to be represented in an accurate yet tractable fashion. Surrogate modeling plays an important role in the integration between process unit models and refinery scheduling optimization.
4. **Optimization Decision-making:** Refinery scheduling has been transitioning from trial-and-error to simulation-aided and more recently to optimization-aided decision-making. Optimization methods are key to manage complex operations and maximize profitability.
5. **Scheduling Parameters:** Tuning scheduling parameters (e.g., time-step, time horizon, rescheduling frequency) provides better solution and easier implementation in the plant. Trade-offs between solution quality and computational tractability must be assessed.
6. **Rescheduling:** Mechanisms to re-optimize and automate the scheduling decision-making provide efficient and timely capabilities for managing such complex and uncertain process.

The development of advanced computer-aided optimization decision-making is fundamental for further advances in the process systems engineering field, and the abovementioned features are key toward state-of-the-art improvements in refinery scheduling decision-making. An efficient decision-making framework that comprises accurate formulations and computer-aided solving algorithms enables improved solutions for refinery scheduling, with higher efficiency and lower costs. Proper assessment of scheduling parameters and tuning the framework hyperparameters provides a proper trade-off between solution quality and computational effort that enables its application for industrial operations.

ACKNOWLEDGMENTS

The authors would like to acknowledge the support from the São Paulo Research Foundation (FAPESP - Grants 2017/03310-1 and 2018/04942-4) and from the Coordination for the Improvement of Higher Education Personnel (CAPES - Finance Code 001).

**CRediT authorship contribution statement**

**Robert E. Franzoi**: Conceptualization, Methodology, Software, Formal Analysis, Writing - Original Draft, Visualization. **Brenno C. Menezes**: Conceptualization, Methodology. **Jorge A. W. Gut**: Conceptualization, Methodology, Writing – Review & Editing, Supervision, Funding Acquisition. **Ignacio E. Grossmann**: Methodology, Supervision, Writing – Review & Editing.

## 7. References

Absi, N., van den Heuvel, W., 2019. Worst case analysis of Relax and Fix heuristics for lot-sizing problems. European Journal of Operational Research, 279(2), 449-458.

Ali, T.H.M., Franzoi, R.E., Menezes, B.C., 2022. Surrogate modeling for nonlinear gasoline blending operations. Computer Aided Chemical Engineering, 49, 1783-1788. Elsevier.

Al-Qahtani, K.Y., Elkamel, A., 2011. Planning and integration of refinery and petrochemical operations. John Wiley & Sons.

Brunaud, B., Perez, H.D., Amaran, S., Bury, S., Wassick, J., Grossmann, I.E., 2020. Batch scheduling with quality-based changeovers. Computers & Chemical Engineering, 132, 106617.

Castillo-Castillo, P.A., Mahalec, V., 2016. Inventory pinch gasoline blend scheduling algorithm combining discrete- and continuous-time models. Computer and Chemical Engineering, 84, 611-626.

Castro, P.M., Grossmann, I.E., 2014. Global optimal scheduling of crude oil blending operations with RTN continuous-time and multiparametric disaggregation. Industrial & Engineering Chemistry Research, 53(39), 15127-15145.

Chryssolouris, G., Papakostas, N., Mourtzis, D., 2005. Refinery short-term scheduling with tank farm, inventory and distillation management: An integrated simulation-based approach. European Journal of Operational Research, 166(3), 812-827.

Cuadros Bohorquez, J.F., Plazas Tovar, L., Wolf Maciel, M.R., Melo, D.C., Maciel Filho, R., 2020. Surrogate-model-based, particle swarm optimization, and genetic algorithm techniques applied to the multiobjective operational problem of the fluid catalytic cracking process. Chemical engineering communications, 207(5), 612-631.

Cuiwen, C., Xingsheng, G., Zhong, X., 2013. A data-driven rolling horizon online scheduling model for diesel production of a real-world refinery. AIChE J., 59 (4), 1160−1174.

Do, H, 2014. Risk management of oil refinery. Master thesis in Energy and Earth Resources, The University of Texas at Austin.

Franzoi, R. E., Ali, T., Al-Hammadi, A., Menezes, B.C, 2021c. Surrogate Modeling Approach for Nonlinear Blending Processes. In 2021 1st International Conference on Emerging Smart Technologies and Applications, 1-8. IEEE.

Franzoi, R.E., Kelly, J.D., Menezes, B.C., Swartz, C. L. E., 2021b. An adaptive sampling surrogate model building framework for the optimization of reaction systems. Computers & Chemical Engineering, 152, 107371.

Franzoi, R.E., Menezes, B.C., 2022. Towards improved scheduling: an analysis on time-steps, time horizon, and rescheduling. Computer Aided Chemical Engineering, 51, 1003-1008. Elsevier.

Franzoi, R.E., Menezes, B.C., Kelly, J.D., Gut, J.A., 2021a. A Moving Horizon Rescheduling Framework for Continuous Nonlinear Processes with Disturbances. Chemical Engineering Research and Design, 174, 276-293.

Franzoi, R.E., Menezes, B.C., Kelly, J.D., Gut, J.W., 2018a. Effective scheduling of complex process-shops using online parameter feedback in crude oil refineries. Computer Aided Chemical Engineering, 44, 1279-1284. Elsevier.

Franzoi, R.E., Menezes, B.C., Kelly, J.D., Gut, J.A., 2019. Design for Online Process and Blend Scheduling Optimization. Computer Aided Chemical Engineering, 47, 187-192. Elsevier.

Franzoi, R.E., Menezes, B.C., Kelly, J.D., Gut, J.A., Grossmann, I.E., 2020. Cutpoint temperature surrogate modeling for distillation yields and properties. Industrial & Engineering Chemistry Research, 59(41), 18616-18628.

Fu, G., Sanchez, Y., Mahalec, V., 2016. Hybrid model for optimization of crude oil distillation units. AIChE Journal, 62(4), 1065-1078.

Garcia, D.J., Jian G., Fengqi Y., 2016. Multi-stage adaptive robust optimization over bioconversion product and process networks with uncertain feedstock price and biofuel demand. Computer Aided Chemical Engineering, 38, 217-222. Elsevier.

Gupta D., Maravelias C.T., 2016. On Deterministic Online Scheduling: Major Considerations, Paradoxes and Remedies. Computers & Chemical Engineering, 94, 312-330.

Jia, Z., Ierapetritou, M., Kelly, J.D., 2003. Refinery Short-Term Scheduling Using Continuous Time Formulation: Crude-Oil Operations. Ind. Eng. Chem. Res., 42, 3085−3097.

Kelly, J.D., 2002. Chronological decomposition heuristic for scheduling: Divide and conquer method. AIChE Journal, 48(12), 2995-2999.

Kelly, J.D., Menezes, B.C., Engineer, F., Grossmann, I.E., 2017. Crude oil blend scheduling optimization of an industrial-sized refinery: a discrete-Time benchmark. In: Foundations of Computer Aided Process Operations, Tucson, United States.

Kelly, J.D., Menezes, B.C., Grossmann, I.E., 2018. Successive LP approximation for nonconvex blending in MILP scheduling optimization using factors for qualities in the process industry. Industrial & Engineering Chemistry Research, 57(32), 11076-11093.

Lee, H., Pinto, J.M., Grossmann, I.E., Park, S., 1996. Mixed-integer linear programming model for refinery short-term scheduling of crude oil unloading with inventory management. Industrial & Engineering Chemistry Research, 35(5), 1630-1641.

Li, F., Yang, M., Du, W., Dai, X., 2020. Development and challenges of planning and scheduling for petroleum and petrochemical production. Frontiers of Engineering Management, 1-11.

Li, J., Karimi, I.A., 2011. Scheduling Gasoline Blending Operations from Recipe Determination to Shipping Using Unit Slots. Ind. Eng. Chem. Res., 50 (15), 9156−9174.

Mencarelli, L., Pagot, A., Duchêne, P., 2020. Surrogate-based modeling techniques with application to catalytic reforming and isomerization processes. Computers & Chemical Engineering, 135, 106772.

Mendez, C.A., Grossmann, I.E., Harjunkoski, I., Kabore, P., 2006. A simultaneous optimization approach for off-line blending and scheduling of oil-refinery operations. Comput. Chem. Eng., 30 (4), 614−634.

Menezes, B.C., Kelly, J.D., Grossmann, I.E., 2013. Improved swing-cut modeling for planning and scheduling of oil-refinery distillation units. Industrial & Engineering Chemistry Research, 52 (51), 18324-18333.

Menezes, B.C., Kelly J.D., Grossmann, I.E., 2015. Phenomenological decomposition heuristic for process design synthesis of oil-refinery Units. Computer Aided Chemical Engineering, 37, 1877-1882. Elsevier.

Mouret, S., Grossmann, I.E., Pestiaux, P., 2009. A novel priority-slot based continuous-time formulation for crude oil scheduling problems. Industrial & Engineering Chemistry Research, 48(18), 8515-8528.

Rafiei, M., Ricardez-Sandoval, L.A., 2020. New frontiers, challenges, and opportunities in integration of design and control for enterprise-wide sustainability. Comp. Chem. Eng., 132, 106610.

Reddy, P.C. P., Karimi, I.A., Srinivasan, R., 2004. A novel solution approach for optimizing crude oil operations. AIChE J., 50, 1177−1197.

Robertson, G., Palazoglu, A., Romagnoli, J.A., 2011. A multi-level simulation approach for the crude oil loading/unloading scheduling problem. Computers & chemical engineering, 35(5), 817-827.

Saharidis, G.K.D., Minoux, M., Dallery, Y., 2009. Scheduling of loading and unloading of crude oil in a refinery using event-based discrete time formulation. Comput. Chem. Eng., 33 (8), 1413−1426.

Shah, N.K., Sahay, N., Ierapetritou, M.G., 2015. Efficient decomposition approach for large-scale refinery scheduling. Industrial & Engineering Chemistry Research, 54(41), 9964-9991.

Song, W., Du, W., Fan, C., Yang, M., Qian, F., 2021. Adaptive Weighted Hybrid Modeling of Hydrocracking Process and Its Operational Optimization. Industrial & Engineering Chemistry Research, 60(9), 3617-3632.

Wang, S., Zhou, L., Ji, X., Dang, Y., 2019. Synthesis and optimization of refinery hydrogen network using surrogate models. Computer Aided Chemical Engineering, 46, 655-660. Elsevier.

Winston, W.L., Goldberg, J.B., 2004. Operations research: applications and algorithms, Vol. 3. Belmont: Thomson Brooks/Cole.

Xia, Z., Wang, S., Zhou, L., Dai, Y., Dang, Y., Ji, X., 2021. Surrogate-assisted optimization of refinery hydrogen networks with hydrogen sulfide removal. Journal of Cleaner Production, 310, 127477.

Xu, J., Zhang, S., Zhang, J., Wang, S., Xu, Q., 2017. Simultaneous scheduling of front-end crude transfer and refinery processing. Computers & Chemical Engineering, 96, 212-236.

Yang, H., Bernal, D.E., Franzoi, R.E., Engineer, F.G., Kwon, K., Lee, S. Grossmann, I.E., 2020. Integration of crude-oil scheduling and refinery planning by Lagrangean Decomposition. Computers & Chemical Engineering, 138, 106812.

Zhang, J., Rong, G., Hou, W. Huang, C., 2010. Simulation based approach for optimal scheduling of fuel gas system in refinery. Chemical Engineering Research and Design, 88(1), 87-99.

Zhong, W., Qiao, C., Peng, X., Li, Z., Fan, C., Qian, F., 2019. Operation optimization of hydrocracking process based on Kriging surrogate model. Control Engineering Practice, 85, 34-40.

Zyngier, D., Kelly, J.D., 2009. Multi-product inventory logistics modeling in the process industries. In Optimization and logistics challenges in the enterprise, 61-95. Springer, Boston, MA.

Zyngier, D., Lategan, J., Furstenberg, L., 2018. A process systems approach for detailed rail planning and scheduling applications. Computers & Chemical Engineering, 114, 273-280.

# Supplementary Material

## A. Illustrative Example: UOPSS-based network

The blend scheduling network shown in Figure A1 is used to illustrate some specificities of the UOPSS representation. There are sources of feedstocks such as marine vessels or pipelines (R1 to R3) supplying a crude oil refinery. These source objects are typically continuous incoming flows of material. In such a process, crude oils with distinct compositions and properties are stored, blended, and processed in crude oil distillation units (CDU) to be physically segregated into multiple distillates, namely, fuel gas (FG), liquid petroleum gas (LPG), light naphtha (LN), heavy naphtha (HN), kerosene (K), light diesel (LD), heavy diesel (HD), and atmospheric residuum (ATR). The distillates undergo several chemical and physical operations to be transformed into refined products with increased economic value. At the end of the refinery site, blend stations mix the intermediate products with boosters to ensure proper product quality specification

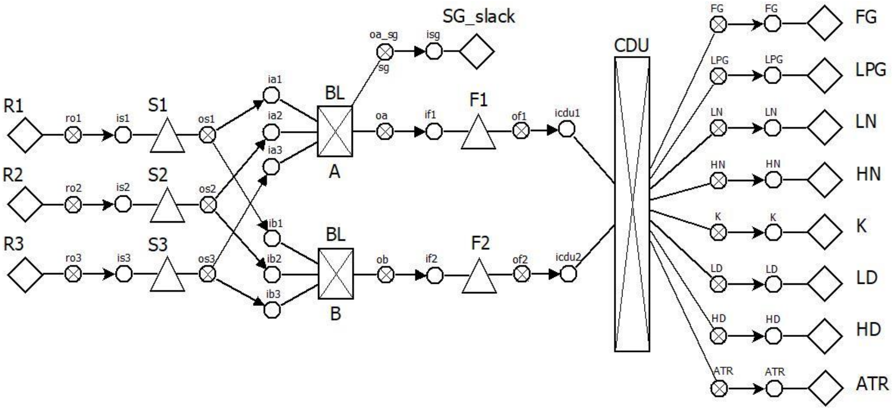


Figure A1: Illustrative blend scheduling problem network.

Let us consider a piece of the network in Figure A1 comprised of a) blender unit (BL) with operational mode (A), out-port ($oa$), and no associated state, and b) feed tank (F1) with no operational mode, in-port ($if1$), and no associated state. Both unit-operations have binary variables, and they can operate (e.g., receive, process, or send material) only if/when their binaries are active (i.e., if both are operating). There are material flows outgoing from the blender unit and incoming to the tank, which are modeled using continuous and binary variables. It is similar for the connection between the two flows, whereby the outgoing flow from the blender is actually the incoming flow to the tank. Traditional mathematical formulations model the flow between the blender and the tank using a single continuous variable, and typically do not consider binary variables. However, the UOPSS representation introduces binaries for the UO groups (blender and tank) and UOPS-UOPS group (connection stream); and continuous variables for the UO groups (blender and tank), UOPS groups (flow outgoing from the blender and flow incoming from the tank), and UOPS-UOPS group (for the material consistency between the UOPS flows). In this example, the continuous variables are associated with material flows, although it is also valid for compositions, properties, and other types of information as well.

## B. Crude oil refinery network

The topology of the crude oil refinery network is presented in Figure 2. There are 12 crude oil feedstocks (CO1 to CO12) that continuously arrive through refinery terminals. They are stored in 12 storage tanks (ST1 to ST12) connected to the blenders (BL1 and BL2). The blend recipe is stored in the feed tanks (FT1 to FT4) to be processed in the distillation network. The cascaded distillation system is comprised of atmospheric distillation units (CDU1 and CDU2), vacuum distillation unit (VDU), flash distillation unit (FDU), and debutanizer tower (DEB). The distillation segregates the complex crude oil blend into crude oil fractions, hereinafter referred to as distillates. A crude oil assay is used for the calculation of crude oil yields, compositions, and properties, in which a micro-cut distribution (segmented into 10 °C fractions) is considered. The production of distillates requires the temperature cutpoints to determine the aggregation of the micro-cuts to the distillates. Details on micro-cuts calculation and distribution can be found in Menezes et al. (2013). Most distillates require further processing in a complex network of unit-operations based on physical separation and chemical transformation processes, whereby enhancing their quality and economic value. This includes the fluid catalytic cracking unit (FCCU), hydrotreaters for coker light naphtha (CLNHT), light cracked naphtha (LCNHT), and diesel (DHT), delayed coking unit (DCU), debutanizers (DEB2 and DEB3), superfractionator (SFR), and catalytic reformer unit (CRU). The product-edge of the network is comprised of additional blenders to mix intermediate streams with boosters for final product specification that meets market and contractual requirements. There are blenders for gasoline (GBL), diesel (DBL), and fuel oil (FBL).

The flash distillation unit (FDU) segregates the crude oil blend to light and heavy fractions respectively sent to the debutanizer tower DEB1 and crude distillation unit CDU1. The debutanizer DEB1 produces fuel gas (FG), liquefied petroleum gas (LPG), and light naphtha (LN). The distillation CDU1 produces heavy naphtha (HN), kerosene (KE), light diesel (LD), heavy diesel (HD), and atmospheric residue (AR). The distillation column CDU2 produces fuel gas, liquefied petroleum gas, light naphtha, heavy naphtha, kerosene, light diesel, heavy diesel, and atmospheric residue. The fuel gas and liquefied petroleum gas produced in these units are sent to product storage sink (FG and LPG, respectively). Light naphtha is sent either to a product storage sink (LN) or to a coker light naphtha hydrotreating system (tank CLNHTTK and hydrotreater CLNHT). Heavy naphtha is sent to a diesel blender (DBL). Kerosene, light diesel, and heavy diesel are sent to a diesel hydrotreating system (tank DHTTK and hydrotreater DHT). The atmospheric residue is sent to storage tanks (V1 and V2) to further feed the vacuum distillation unit (VDU). The feeding for the vacuum distillation includes the atmospheric residue from the CDUs, in addition to an external (imported) feed stored in a pool (AR_IMP). The VDU produces light vacuum gas oil (LVGO), heavy vacuum gas oil (HVGO), and vacuum residue (VR). The vacuum gas oils are sent to the fluidized catalytic cracking unit (tank FCCUTK and column FCCU) and the vacuum residue is sent to the delayed coking unit (tank DCUTK and column DCU).

The fluidized catalytic cracking unit (FCCU) produces fuel gas, liquefied petroleum gas, light cracked naphtha (LCN), heavy cracked naphtha (HCN), light cycle oil (LCO), and decanted oil

(DO). Fuel gas is sent to the product storage sink. Liquefied petroleum gas is sent to a debutanizer unit for stabilization, composed of a tank (DEBTK) and a debutanizer column (DEB2). Light cracked naphtha is sent to a hydrotreating system (tank LCNHTTK and hydrotreater LCNHT). Heavy cracked naphtha and light cracked oil are sent to a fuel oil blender (FBL). Decanted oil is sent to the delayed coking unit (tank DCUTK and delayed coking unit DCU).

The delayed coking unit (DCU) processes vacuum residue and decanted oil to produce fuel gas, liquefied petroleum gas, coker light naphtha (CLN), coker heavy naphtha (CHN), coker light gas oil (CLGO), coker medium gas oil (CMGO), coker heavy gas oil (CHGO), and coke. Fuel gas and liquefied petroleum gas are sent to product storage sinks. Coker light naphtha is sent to a hydrotreatment system (tank CLNHTTK and hydrotreater CLNHT). Coker heavy naphtha, coker light gas oil, and coker medium gas oil are sent to the fuel oil blender (FBL). Coker heavy gas oil is sent to the FCCU. The coke produced is sent to a tank (COKETK) and later stored in a coke reservoir (COKE).

The liquefied petroleum gas from FCCU and DCU is sent to a debutanizer system comprised of a tank (DTK2) and a debutanizer tower (DEB2), which segregates the molecules into propane (C3) and butane (C4). The propane stream is further processed in a superfractionation system comprised of a tank (SFRTK) and a superfractionation unit (SFR), which segregates the molecules into propylene and pure propane. The propylene, propane, and butane final products are sent to storage sinks for distribution (PROPYLENE, PROPANE, and BUTANE, respectively).

Light naphtha from DEB1 and coker light naphtha from DCU are sent to a coker light naphtha hydrotreating system comprised of a tank (CLNHTTK) and a hydrotreater (CLNHT). The light hydrotreated naphtha is sent to a debutanizer system for the stabilization of naphtha, which is comprised of a tank (DEBTK) and a debutanizer column (DEB3). The output fractions are light naphtha, which is sent either to a naphtha storage tank (LN) or to a gasoline blender (GBL); heavy naphtha, sent to a catalytic reform system (tank CRUTK and processing unit CRU); and gas oil, sent to the diesel blender (DBL). The heavy naphtha is processed in the catalytic reformer unit and is subsequently sent to the blender GBL for gasoline production.

The cracked light naphtha hydrotreatment system is comprised of a tank (LCNHTTK) and a hydrotreater (LCNHT) to process light cracked naphtha from the FCCU. The hydrotreated naphtha is further blended in the gasoline blender (GBL) with other naphtha streams, in addition to ethanol and isooctane (from the imported sources ETOH and ISOOCTANE) for gasoline production and specification. The final gasoline product is stored in a tank (GTK) and later sent to a gasoline storage sink for distribution (GASOLINE).

The diesel hydrotreating system is comprised of a tank (DHTTK) and a hydrotreater (DHT), in which multiple intermediate kerosene and diesel streams are hydrotreated. The hydrotreated diesel is further blended in the diesel blender (DBL) with heavy naphtha from the distillation units and gas oil from the debutanizer DEB3 for diesel production specification. The final diesel product is stored in a tank (DTK) and sent to a diesel storage sink for distribution (DIESEL).

Multiple refinery streams, including heavy cracked naphtha, light cycle oil, vacuum residue, coker heavy naphtha, coker light gas oil, and coker medium gas oil, are sent to a fuel oil blender (FBL) for the production and specification of fuel oil. The final fuel oil product is stored in a tank (FTK) and sent to a storage sink for distribution (FO).