ROBERT EDUARD FRANZOI JUNIOR

Integrated scheduling optimization in the crude oil refinery industry: from crude oil unloading to fuel deliveries

São Paulo

2021

ROBERT EDUARD FRANZOI JUNIOR

Integrated scheduling optimization in the crude oil refinery industry: from crude oil unloading to fuel deliveries

Revised Edition

Ph.D. thesis presented to the Graduate Program in Chemical Engineering at the Escola Politécnica, University of São Paulo, to obtain the degree of Doctor of Science.

Concentration Area: Chemical Engineering

Advisor: Prof. Dr. Jorge Andrey Wilhelms Gut
Co-Advisor: Prof. Dr. Brenno Castrillon Menezes

São Paulo

2021

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 01 de abril de 2021

Assinatura do autor: Robert Franzoi

Assinatura do orientador:

Catalogação-na-publicação

Franzoi Junior, Robert Eduard
Integrated scheduling optimization in the crude oil refinery industry: from crude oil unloading to fuel deliveries / R. E. Franzoi Junior -- versão corr. -- São Paulo, 2021.
290 p.

Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Química.

1.Crude oil refining 2.Scheduling 3.Modeling 4.Optimization 5.Machine learning I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Química II.t.

ACKNOWLEDGEMENTS

I would like to start truly thanking God, beyond everything, for the unique opportunity of existing. I deeply appreciate my life and my family, and all the countless blessings we have received. I thank God for the protection, for carrying me when I most needed, and for giving me the strength and faith to always persevere. I also want to thank all the angels, saints, and souls that looked after me in any moment of my life.

Agradeço profundamente à minha família, em especial meus pais, Vânia e Robert, minha irmãzinha, Gabriela, e minha esposa, Juliana. O suporte que minha família me deu, financeiramente e psicologicamente, foi essencial para eu ter chegado aqui e para eu ter me tornado quem sou. O amor, os cuidados e a preocupação incansáveis da minha mãe me guiaram da melhor maneira possível. O amor, a confiança, a motivação e o incentivo contínuos e ilimitados do meu pai foram fundamentais para iniciar minha jornada em busca de meus sonhos e despertar minha vontade incessante de crescer e me tornar cada dia melhor. O amor, a bondade e a fé da minha irmã me ajudaram a ser um irmão, um filho e uma pessoa melhor, e sempre ser verdadeiro comigo mesmo.

I could not be more grateful to my beautiful and beloved wife, Juliana. I deeply thank her for all the sacrifices she made for us, especially during all my travels and exchange programs abroad, for her unconditional love, and for supporting me to pursue my personal and professional dreams. She always believes and sees in me something that even I cannot see. She gives me faith and hope when I most need, helps me to be my best every day, and encourages me to always want to become better tomorrow. Her great heart and brilliant mind continuously provide me the strength and courage I need to fulfill my dreams and goals. Her importance for both my personal and professional lives are unmeasurable.

I thank my academic advisor, Prof. Jorge A. W. Gut, for all his support that guided me to pursue my academic goals, for being patient, and for helping me grow as a PhD student and as a researcher. Prof. Gut provided a hardworking and fun environment with straightforward and insightful feedbacks. His friendly guidance and valuable advices were fundamental throughout all stages of my work.

I thank my academic co-advisor, Prof. Brenno C. Menezes, for the uncountable meetings, for engaging me in new ideas, and for all his time and effort discussing research, debugging codes, and writing papers. Prof. Menezes has a brilliant mind and an unlimited source of great ideas. I could not have accomplished that much without his support and my PhD would not have been so enjoyable without his guidance and encouragement.

I thank Mr. Jeffrey D. Kelly from Industrial Algorithms Limited for being a supportive collaborator in multiple discussions throughout the development of my research, and for the license provided for using his modeling and optimization software.

I would like to gratefully acknowledge Prof. Christos T. Maravelias and his group at University of Wisconsin–Madison for the support provided on the development of an exchange research program in Madison, WI, United States. Prof. Maravelias provided a great support with meaningful discussions and insightful feedbacks, always in a fun and friendly environment.

I would like to gratefully acknowledge Prof. Ignacio E. Grossmann and his group at Carnegie Mellon University for the support provided on the development of an exchange research program in Pittsburgh, PA, United States. Prof. Grossmann is not only an outstanding researcher, but also a kind and peaceful person. It was a great pleasure working with him.

I would like to gratefully acknowledge Prof. Christopher L. E. Swartz and his group at McMaster University for the support provided on the development of my exchange research program in Hamilton, ON, Canada. Prof. Swartz guided me through a very enjoyable research experience with his knowledge, valuable insights, and helpful feedbacks.

I would like to thank the financial supports from CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico (grant #131213/2016-4), from FAPESP - Fundação de Amparo à Pesquisa do Estado de São Paulo (grants #2017/03310-1, #2018/04943-0 and #2018/26252-0), and from ELAP - Emerging Leaders in the Americas Program.

I thank everyone who directly or indirectly prayed, helped, and looked after me in any moment of my life. I am deeply thankful for all the gestures of love, compassion, and friendship that you all have provided me. Sometimes the smallest details make the biggest difference.

RESUMO

FRANZOI, R. E. **Otimização integrada da programação de produção no refino de petróleo: da descarga de óleo à entrega de combustíveis**. 2021. 290 p. Tese (Doutorado) – Escola Politécnica, Universidade de São Paulo, 2021.

A otimização da programação de produção em refinarias de petróleo é um problema complexo e desafiador devido a sua formulação MINLP não convexa em tamanho industrial. Três conceitos principais vêm sendo adotados na indústria e academia para lidar com esse problema. Primeiro, utiliza-se uma formulação simplificada que não inclui todas as unidades de processo, tanques, fluxos e variáveis do problema industrial. Segundo, o modelo de programação de produção é dividido em subproblemas a serem resolvidos hierarquicamente. Terceiro, ainda se utiliza abordagens baseadas em simulação ao invés de otimização devido à complexidade de tal formulação. Contudo, avanços recentes em métodos de tomada de decisão, na capacidade de processamento dos computadores e nos algoritmos de otimização permitem a modelagem e otimização de problemas anteriormente intratáveis, de modo a fornecer recursos para novas aplicações industriais em tempo real e gerar oportunidades para o desenvolvimento de melhores estratégias de modelagem e otimização. Neste trabalho são abordadas formulações complexas baseadas em problemas industriais de programação de produção em refinarias de petróleo. A novidade desta pesquisa consiste em modelar e otimizar tais modelos, incluindo características de design de processo para operações de mistura e processamento; abordagens de decomposição para formulações intratáveis; estratégias de reprogramação para aplicações em tempo real; e modelos aproximados para sistemas de otimização integrados. Abordagens de decomposição permitem construir formulações mais simples a partir de problemas complexos e de grande escala. Design aprimorados para operações de processamento e mistura fornecem previsões mais precisas, flexibilidade de produção e maior valor econômico para o processo. Heurísticas são utilizadas para reduzir significativamente o esforço computacional, limitando o espaço de busca na otimização através de estratégias em horizonte rolante e de técnicas de relaxação iterativas para problemas misto-inteiro lineares. Estratégias de reprogramação da produção e de atualização de parâmetros reduzem as incompatibilidades entre modelo e planta ao lidar com incertezas e distúrbios de maneira eficaz, reduzindo imprecisões, mantendo o sistema atualizado e fornecendo um modo sistemático para aplicações em tempo real. Modelos aproximados substituem formulações complexas e permitem a integração de modelos de unidades de processo em ambientes de otimização de programação de produção. As formulações e metodologias propostas são coerentes com aplicações industriais de grande escala em relação a restrições operacionais, valor agregado do processo e complexidade e tamanho do problema. Os resultados indicam que formulações MINLP não convexas de problemas de programação de produção em refinarias podem ser resolvidas eficientemente utilizando estratégias de decomposição, heurísticas e machine learning, o que pode potencialmente fornecer metodologias de modelagem e otimização adequadas para aplicações em problemas reais em escala industrial.

Palavras-chave: Refino de Petróleo, Programação de Produção Online, Modelagem e Otimização, Estratégias heurísticas, Modelos Surrogados.

ABSTRACT

FRANZOI, R. E. **Integrated scheduling optimization in the crude oil refinery industry: from crude oil unloading to fuel deliveries**. 2021. 290 p. Tese (Doutorado) – Escola Politécnica, Universidade de São Paulo, 2021.

The crude oil refinery scheduling optimization is a complex and challenging problem because of its large-scale and complex-scope non-convex MINLP formulation. Three main concepts have been adopted in both industry and academia to handle this issue. First, a simplified formulation is typically considered, which does not include all the processing units, tanks, flows, and variables from the real industrial problem. Second, the refinery scheduling formulation is broken down into subproblems to be hierarchically solved. Third, simulation-based instead of optimization-based approaches are still employed due to the intractability of such formulation. However, the recent advancements in decision-making modeling, computer-aided resources, and solution algorithms allow the modeling and optimization of previously intractable problems, provide resources for novel real-time industrial applications, and open opportunities for the development of novel and improved modeling and optimization strategies. The research topics addressed herein focus on handling complex formulations typically found in crude oil refinery scheduling applications. The novelty of this research consists of modeling and optimizing a complete crude oil refinery scheduling problem, including decomposition approaches for handling intractable formulations, improved network designs for blending and processing operations, rescheduling strategies for online applications, and surrogate modeling for integrated optimization environments. Decomposition approaches are useful for building simpler and tractable formulations from complex and large-scale problems. Improved processing and blending designs provide more accurate predictions, production flexibility, and increased economic value for the process. Modeling and solving heuristics are used to significantly reduce the computational effort by limiting the optimization search space in constructive rolling horizon strategies and by introducing iterative relaxations on mixed-integer linear programming problems. Rescheduling and parameter updating strategies mitigate plant-model mismatches by effectively handling uncertainties and disturbances, reducing inaccuracies, maintaining the state of the system updated, and providing a systematic fashion for online applications. Surrogate models can effectively replace complex formulations in order to allow the integration of unit-operation models within refinery scheduling optimization environments. The formulation and methodologies addressed herein are coherent with large-scale and complex-scope industrial applications in terms of applicability, operational constraints, refinery economics, and problem complexity and size. The results indicate that complex non-convex MINLP refinery scheduling formulations can be efficiently solved by utilizing decomposition, heuristic, machine learning, and rescheduling strategies, which would potentially provide improved modeling and optimization capabilities for real industrial applications.

Keywords: Crude Oil Refining, Online Scheduling, Modeling and Optimization, Heuristic approaches, Surrogate Modeling.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

ALAMO	Automated Learning of Algebraic Models for Optimization
APC	Advanced Process Control
API	American Petroleum Institute
ASTM	Association Society for Testing and Materials
ATR	Atmospheric Residue
CDH	Chronological Decomposition Heuristic
CDU	Crude Distillation Unit
CHGO	Coker Heavy Gas Oil
CLGO	Coker Light Gas Oil
CLN	Coker Light Naphtha
CHN	Coker Heavy Naphtha
CLNH	Coker Light Naphtha Hydrotreater
CMGO	Coker Medium Gas Oil
CO	Crude Oil
CPU	Central Processing Unit
CRU	Catalytic Reforming Unit
CSW	Conventional Swing-Cut
DCU	Delayed Coker Unit
DHT	Diesel Hydrotreater
DO	Decanted Oil
ETOH	Ethanol
FCCU	Fluid Catalytic Cracking
FDH	Flowsheet Decomposition Heuristic
FG	Fuel Gas
FI	Fractionation Index
FP	First Principles
FRS	Full Rescheduling Strategy
FY	Fixed Yield
GAMS	General Algebraic Modeling Language
GO	Gas Oil
HCN	Heavy Cracked Naphtha
HD	Heavy Diesel

HN	Heavy Naphtha
HVGO	Heavy Vacuum Gas Oil
IML	Industrial Modeling Language
IPL	Industrial Programming Language
ISW	Improved Swing-Cut
K	Kerosene
IMPL	Industrial Modeling & Programming Language
IP	Institute of Petroleum
LCNH	Light Cracked Naphtha Hydrotreater
LCO	Light Cycle Oil
LCN	Light Cracked Naphtha
LD	Light Diesel
LHS	Latin Hypercube Sampling
LN	Light Naphtha
LP	Linear Programming
LPG	Liquefied Petroleum Gas
LVGO	Light Vacuum Gas Oil
MILP	Mixed-Integer Linear Programming
MINLP	Mixed-Integer Nonlinear Programming
MIQP	Mixed-Integer Quadratic Programming
MIP	Mixed-integer Programming
MISO	Multiple-Input-Single-Output
N	Naphtha
NLP	Nonlinear Programming
OA	Outer Approximation
PDH	Phenomenological Decomposition Heuristic
QLQP	Quantity-Logic-Quality Phenomena
RTN	Resource Task Network
RTO	Real-time Optimization
S	Sulfur Content
SG	Specific Gravity
SLP	Sequential Linear Programming
SM	Surrogate Model

STN	State Task Network
SW	Swing-Cut
TBP	True Boiling Point
UOPSS	Unit-Operation-Port-State Superstructure
VDU	Vacuum Distillation Unit
VGO	Vacuum Gas Oil
VR	Vacuum Residue
WO	Williams-Otto
WTR	Weight Transfer Ratio
X	Independent Variable
Y	Dependent Variable
YLD	Yield

CONTENTS

1
General introduction and literature review

1.1 General Introduction

Petroleum is a complex mixture of hydrocarbons with diverse applications. It has been used since biblical times by many distinct peoples including the Babylonians, Egyptians, Greeks, Romans, among others (TRIGGIA et al., 2001). The large-scale industrial processing of petroleum dates from the 19th century, and it was initially used for the production of kerosene. Since then, technological advances and the increasing demand for fuels have led to the expansion of the refineries in terms of both the production capacity and the variety and quality of products (AL-QAHTANI and ELKAMEL, 2011).

Petroleum is currently one of the most important and valuable commodities, mostly because it can be processed into fuels that contribute to a large amount of the world energy generation. That includes fuel gas and natural gas, which are used for the heating of residential and commercial buildings, and for generating electricity; and gasoline, kerosene, and diesel, which are the mostly consumed fuels for vehicles, such as cars, trucks, ships, and aircrafts (SPEIGHT, 2006).

The production chain of the petrochemical industry begins with the exploration and extraction processes, in which the petroleum can be found in sedimentary rocks in gaseous (natural gas), liquid (crude oil), semi-solid (bitumen), and solid (wax and asphaltite) forms. Among all these forms, the liquid crude oil is the most valuable and the most attractive in terms of processing, and hence, it is the raw material processed in refineries (RIAZI, 2005). The process of transforming the crude oil into fuels, also known as the refining process, aims to split it into smaller molecules and clustering the molecules with similar characteristics in crude oil fractions. These fractions undergo several conversion and treatment processes in order to enhance their purity and improve their quality. The final steps consist of mixing operations whose purpose is to guarantee specific characteristics in terms of quality (e.g., maximum sulfur content and specific gravity). Subsequently, the products are stored in tanks or pools and are further sold and distributed via logistic operations (AL-QAHTANI and ELKAMEL, 2011).

Throughout the entire crude oil refinery network, there are multiple decisions to be carried out. This decision-making process is highly important for the refinery economics and it is typically referred to as crude oil scheduling. Some of these decisions include where to allocate arriving feedstocks, what is the optimal blend of crudes to be mixed and processed at a given day and at a given unit, what is the optimal blend of final products considering their qualities and market dynamics, where to store the final products, what are the quality issues that may arise from the refinery operations, etc. (KELLY et al., 2017b).

Historically, the crude oil scheduling production was carried out using spreadsheets, simulators, and rudimentary tools, with most decisions being carried out manually in a trial and error procedure (MENEZES et al., 2015a). However, with the recent advances in the decision-making modeling, solving algorithms, and computer-aided resources, the scheduling decision-making has greatly improved and has become increasingly automated. Many opportunities arise from these technological advances, in which a proper and efficient scheduling has become essential for the competitiveness of crude oil refineries (KELLY and MANN, 2003). In that context, there are open opportunities for employing computational tools to obtain better scheduling solutions for real industrial applications (REDDY et al., 2004a). The main benefits include finding a wide and varied set of feasible (and potentially better) optimal solutions; reducing the computational effort within simulation and optimization applications, which leads to economic and operational advantages; and adapting the incumbent scheduled operations under the occurrence of adverse or unexpected events (e.g., finding new feasible and/or optimal solutions whenever any event makes the current scheduling operations infeasible or suboptimal) (JIA et al., 2003; KELLY and MANN, 2003).

Due to economic and operational factors, including market competitiveness, market fluctuations (e.g., demand and price of products, demand and quality of feedstocks), uncertainties and other operational disturbances, and strict environmental legislation, crude oil refineries have to achieve improved and highly efficient operations by using advanced technology in addition to exploiting all the opportunities for increasing profit and reducing costs. One of these opportunities is to use mathematical modeling and computational algorithms to optimize the crude oil scheduling operations in order to find optimal, and potentially better, solutions (LI et al., 2016). Aiming to achieve an accurate and reliable modeling and optimization of the process, there are two essential considerations to be addressed. First, the mathematical formulation must be properly designed to include complete and accurate information from the quantity, logistics, and quality processes, and to determine the scope of the problem in terms of both the scheduling operations based on the real plant and the modeling aspects related to the tuning of parameters such as the future time horizon, time discretization, etc. Second, efficient problem solving techniques must be employed to properly optimize the mathematical formulation (KELLY et al., 2017b).

There are three distinct types of information that should be considered in crude oil scheduling problems: quantity (amounts and flows), logistics (binary decision variables), and quality (properties such as sulfur content, specific gravity, pour point, among others). Simultaneously considering the quantity (linear), logistics (mixed-integer linear), and quality (nonlinear) information leads to a mixed-integer nonlinear programming (MINLP) problem. Moreover, the crude oil refining process is highly complex, with a large number of continuous and binary variables and linear and nonlinear equations. This results in a complex large-scale MINLP problem, which is hard to solve and highly time consuming. Thus, several studies in the literature present strategies for decomposing this problem, originally an MINLP, into two subproblems, a mixed-integer linear programming (MILP) problem and a nonlinear programming (NLP) problem, which are solved sequentially (WENKAI et al., 2002 ; MOURET et al., 2009; CASTRO and GROSSMANN, 2014; CAFARO et al., 2015; KELLY et al., 2017a). This type of strategy helps to reduce the size of the problem and the computational effort in simulation/optimization environments.

Although the crude oil refinery operations are a continuous process, it is necessary to establish a finite time horizon for mathematical simulation and optimization purposes. Furthermore, it is important to tune the balance between the size (i.e., future time and time discretization) and the complexity of the problem so as to provide computational tractability, especially for applications that require fast solutions. To achieve scheduling solutions that are coherent with industrial applications (i.e., under similar operational conditions to those industrially used), with good accuracy, and in reasonable computational time (minutes), Kelly et al. (2017b) propose scheduling optimizations for the future seven days with time steps of two hours, with an emphasis on scheduling solution towards online applications.

Aiming to handle the complexity of the refinery scheduling formulation, most studies in this literature considers several simplifications, related to either the modeling of specific units or to the overall scope of the problem. The former regards the use of simplified models for the processing units throughout the refinery, including the atmospheric and vacuum distillation towers, hidrotreaters, hidrocrackers, fluid catalytic cracking unit, naphtha reformer, etc. Instead of using complex or rigorous models to model these units, simplified correlations may be employed, which allows the integration of processing unit models in refinery optimization applications. The latter regards simplifying the overall scheduling scope to reduce or limit the complexity of the problem in terms of the number of variables and constraints. It is worth mentioning that most works in the literature simplify the overall scope of the refinery because of the complexity and difficulties that arise by including even simplified processing unit models in the refinery scheduling optimization, and focus only on the crude oil selection, blending, and feeding to the crude distillation unit (CDU). Despite several works on crude oil scheduling in terms of both modeling and optimization, to the best of our knowledge there are only one study towards integrated industrial-scale operations that fully consider the refinery process, including the crude oil blending, the processing units (i.e., distillation towers, naphtha reformer, catalytic cracking, delayed cooker, hydrotreaters, etc.), and the product blending (XU et al., 2017). However, a reduced number of variables is used in a continuous time model (in the order of dozens binary variables), which is not coherent with real industrial operations including a systematic production with detailed logistics and quality operations. Therefore, embedding unit-operation models in refinery scheduling application remains an open gap in the literature.

The refinery unit-operation most studied and discussed in the literature is the distillation unit, especially because its crucial and important role for refinery operations. When addressing the distillation unit modeling for optimization applications, most works have also introduced simplifications so as to build formulations that can be embedded into optimization problems. One of the first and most simple methods considers that the yields and properties of the outlet fractions from the CDU are fixed and depend exclusively on the crude oil assay data (FU and MAHALEC, 2015). Such simplification was widely used for two main reasons. First, if a rigorous simulator or a complex model is employed to calculate the yields and properties of the distillates, the resulting formulation would be intractable, i.e., the optimization would be too time consuming to be used for real industrial applications. Second, the fixed yield method is simple, straightforward, and provide reasonable accuracy, which was suitable for some applications when better models had not been developed yet. Authors such as Li, Hui, and Li (2005), Guerra and Le Roux (2011a, 2011b), Menezes et al. (2013), Fu, Sanchez, and Mahalec (2015), and Franzoi et al. (2020), proposed more accurate methodologies that improved the predictions of the distillation unit outputs. Moreover, some of these models are small in size and have low complexity, therefore they can potentially be embedded in scheduling optimization applications with small time steps (e.g., two hours) for improved predictions and for broader applications.

Another important feature related to scheduling operations concerns the integration of the modeling, optimization, validation, and implementation towards properly determining the optimal schedule. First, the modeling should be coherent, reliable and accurate in terms of mathematically representing the real process. That includes considering a deep level of details and as much information as possible, but also finding a balance between the accuracy and complexity of the model aiming to achieve computational tractability. Second, the formulation should be properly optimized utilizing computational software, tools, and solving strategies. Third, the scheduling solution needs to be validated (i.e., the solution to be implemented must be reliable and accurate, and plant-process mismatches should be mitigated) for achieving improved operations. If the three previous steps are properly performed, the schedule implementation is expected to be as smooth as possible, given that the refinery operations are highly complex, in which there are uncertainties, noises, and disturbances throughout the entire plant.

After the schedule is modeled, optimized, validated, and implemented, new information becomes available over time and should be considered in the schedule as soon as possible (GUPTA et al., 2016a). Moreover, noises, disturbances, and disruptions change the incumbent (expected) process conditions, which typically leads to suboptimality and to eventual infeasibilities (GUPTA et al., 2016b). Thus, authors such as GUPTA et al. (2016a) and GUPTA et al. (2016b) address the need of rescheduling to improve and adapt the incumbent schedule to account for uncertainties and disturbances, in addition to considering new available information. That leads to sequential re-optimization approaches focused on restoring feasibility and optimality to the incumbent solution, in which improved operations are expected.

Some of the missing gaps on the crude oil refinery scheduling, including the mathematical modeling and optimization strategies, are addressed herein. These topics encompass: a) the design and development of the mathematical formulation based on industrial refinery operations or other relevant industrial processes; b) the design and development of optimization strategies employed to solve the mathematical formulation; c) developing and/or including simplified formulations for unit-operations (e.g., distillation unit) in the refinery modeling and optimization; and d) introducing online scheduling features in the formulation.

1.2 Objectives

Considering the scope discussed in the General Introduction section, this thesis addresses the modeling and optimization aspects for crude oil refinery scheduling applications, and aims to contribute for the state-of-the-art literature on the topic. The following specific objectives are proposed:

- Design and build the mathematical formulation for crude oil refinery scheduling applications using as modeling platform the software IMPL (Industrial Modeling & Programming Language), from Industrial Algorithms Limited. The scope considered encompasses the crude oil unloading, blending, and feeding to the CDU, the processing units throughout the refinery, and the blending and storage of final products.
- Utilize an MILP-NLP decomposition strategy to reduce the size of the large-scale nonconvex MINLP model from the refinery scheduling formulation, which arises due to the large number of logistic decisions (binary variables) and nonlinear terms (mostly in the blending equations). In this approach, the MILP and the NLP subproblems are sequentially and iteratively solved.
- Utilize a linear reformulation of blending equations to include nonlinear quality information (e.g., specific gravity, sulfur content, etc.) in the MILP model, improving the accuracy of the MILP solution. By applying this technique, it is expected a lower

gap between the MILP and NLP solutions, and hence, better convergence of the decomposition strategy.

- Utilize and/or develop simplified models or correlations for the refinery unit-operations and embed them in the refinery scheduling optimization. These models are typically employed to predict the outputs of the respective units.
- Implement online scheduling approaches using data feedback and/or simulated measurements to provide a systematic approach that handles uncertainties, noises, and disturbances in the process in a continuous rescheduling fashion.
- Employ commercial optimization solvers (e.g., CPLEX and GUROBI) to optimize the mathematical formulations.

1.3 Thesis Novelties and Contributions

Considering the state-of-the-art literature on process optimization for crude oil refining processes, in addition to the objectives presented in Section 1.3, the main contributions and novelties of this thesis are highlighted as follows. This thesis focuses on the development, modeling, and optimization of complex scope and industrial scale crude oil refinery scheduling problems, aiming to solve highly complex, large, and intractable models that have never been addressed in such complexity in the scheduling optimization literature. The problems are built within a discrete time formulation as non-convex mixed-integer nonlinear models (MINLP) containing hundreds of thousands of variables and constraints. Integrated modeling and optimization approaches are developed and embedded in systematic and efficient frameworks, whereby are employed decomposition, linearization, heuristic-based, and machine learning strategies for handling intractable formulations, the investigation of process design features for blending and processing operations, continuous rescheduling strategies for mitigating plant-model mismatches and handling uncertainties, noises, and disturbances in the process, and surrogate models to efficiently replace rigorous and complex formulations in order to allow the integration of unit-operation models within refinery scheduling environments.

1.4 Thesis Outline

This PhD thesis is divided in eight chapters:

(I) General Introduction and Literature Review (Chapter 1).

(II) Refinery Scheduling: Mathematical Formulation (Chapter 2).

(III) Refinery Scheduling: Modeling, Configuration, and Optimization (Chapter 3).

(IV) Refinery Scheduling: Case Studies and Discussion (Chapter 4).

(V) A Closed-Loop Rescheduling Framework for Continuous Nonlinear Processes with Disturbances (Chapter 5).

(VI) Cutpoint Temperature Surrogate Modeling for Refinery Applications (Chapter 6).

(VII) Online Large-Scale Refinery Scheduling with Surrogate Approximations (Chapter 7).

(VIII) General Conclusions and Future Outlook (Chapter 8).

Chapter 1 presents a general introduction and the literature review on the topics of petroleum, crude oil refining, and modeling and solving aspects of crude oil scheduling. Chapter 2 introduces a generic mathematical formulation for the crude oil refinery scheduling problem. Chapter 3 discusses the modeling, configuration, and optimization methodologies applied for tackling and solving the scheduling formulation. Chapter 4 presents several case studies typically found within crude oil refinery scheduling applications, including proper discussions and their respective results. Chapter 5 introduces the online scheduling topic, in which a rescheduling framework is developed for handling uncertainties and disturbances in continuous nonlinear processes. Chapter 6 presents a cutpoint temperature surrogate modeling approach in order to include unit-operation models in the refinery scheduling optimization environment. Chapter 7 discusses online large-scale refinery scheduling applications including the use of surrogate modeling. Chapter 8 presents general conclusions of this work and highlights the future outlook on the topic of refinery scheduling optimization.

1.5 Literature Review

In this section it will be presented a review on the main topics addressed in this thesis, including petroleum, petrochemical industry, crude oil scheduling, mathematical modeling and approaches, online scheduling strategies, and refinery processing unit modeling.

1.5.1 Petroleum

The word petroleum comes from the Latin: petra (stone) and oleum (oil). Petroleum is an oily substance in its liquid state, flammable, and less dense than water, it has a strong smell and its color varies between black and light brown (TRIGGIA et al., 2001). It is considered a fossil fuel because it is formed through the decomposition of plants and animals. The remains of dead organisms accumulate on the bottom of lakes and seas, mixed with sand and other sediments. Over time, the combination of pressure, heating, and bacterial action, transforms this complex organic matter into products such as hydrocarbons and water. Due to its chemical process of formation, petroleum is a complex mixture of thousands distinct hydrocarbons, in addition to small amounts of other elements such as sulfur, nitrogen, oxygen, metals, and some salts. (ROBINSON, 2006).

1.5.2 Petroleum Composition

The composition of the hydrocarbon mixture that originates petroleum varies significantly depending on the source or reservoir, so that there are no identical petroleums (MARIANO, 2001). Table 1.1 presents the typical mass fraction ranges of the chemical elements that form petroleum, including carbon, hydrogen, nitrogen, oxygen, sulfur, and metals.

Table 1.1: Mass fraction of chemical elements in petroleum.

Elements	Mass Faction (%)
C	83.0 – 87.0
H	10.0 – 14.0
N	0.1 – 2.0
O	0.05 – 1.5
S	0.05 – 6.0
Metals	< 0.1

Source: Speight (2006).

1.5.2.1 Hydrocarbons elements

The hydrocarbon molecules present in the petroleum are divided into four classes, namely, paraffins, naphthenes, aromatics, and olefins. Paraffins are open and saturated hydrocarbon chains, stable and with a generic molecular formula C_nH_{2n+2}, which may have ramifications. They are named with the suffix “ane” (e.g., methane, isopentane). Naphthenes are a class of cyclic aliphatic hydrocarbons characterized by having one or more rings of saturated carbon atoms. They have the generic molecular formula C_nH_{2n} and the prefix “cycle” (cyclopropane). In the aromatics molecules there are alternating single and double bonds in rings of six carbon atoms, such as the benzene. Olefins are unsaturated hydrocarbons with generic formula C_nH_{2n} rarely found in nature, as they are highly unstable due to their high reactivity (TRIGGIA et al., 2001).

1.5.2.2 Non-Hydrocarbons elements

The non-hydrocarbon elements present in the petroleum consist in organic compounds containing nitrogen, oxygen, and sulfur atoms, inorganic compounds (e.g., salts), and organometallic compounds. These elements are typically found in most fractions of petroleum, especially in the heavier ones because of their large chains. Even in small concentrations, they greatly impact the refining operations due to the several risks to the process, including corrosion of equipment by organic acids and inorganic salts;

disruptions in the process due to the deposit of salts; and poisoning of catalysts by organometallic compounds (YAMANISHI, 2007).

1.5.3 Forms of Petroleum

In the exploration and extraction petrochemical processes, petroleum can be found in sedimentary rocks in gaseous (natural gas), liquid (crude oil), semi-solid (bitumen), and solid (wax and asphaltite) forms, depending on the nature of its chemical constituents and the temperature and pressure conditions where it exists. The main forms of petroleum typically found in the exploration process are the natural gas, which does not condense into a liquid at surface conditions; the condensate, which is gaseous at the reservoir conditions and condenses into a liquid at the surface; and the crude oil, the liquid part of petroleum which is the most valuable and attractive to be processed due to the products (distillates) obtained (RIAZI, 2005; DEMBICKI, 2016). Therefore, as the crude oil is the feedstock used in the refinery processing operations, it will be further discussed herein.

1.5.4 Crude oil properties

Some of the main crude oil properties that are economically relevant to the refining process are the specific gravity, sulfur content, and the predominant hydrocarbon type (MARTÍNEZ, 1999 apud ZYLBERBERG, 2006). Regarding specific gravity, the crude oil is classified according to a degree that ranges from light (less dense) to heavy (more dense). This classification is defined according to the American Petroleum Institute, and is referred to as API degree. Equation 1.1 presents the formula to calculate the API degree, in which $SG\,(60\,°F)$ is the specific gravity of the crude oil at the temperature of 60 °F. The lower the specific gravity, the higher the API degree and hence, the greater the commercial value of the crude oil, because it will yield in a larger amount of valuable distillates, such as gasoline and diesel.

$$°API = \frac{141.5}{SG\,(60\,°F)} - 131.5 \tag{1.1}$$

Crude oil may also be classified according to its predominant type of hydrocarbon, into paraffinic, naphthenic, aromatic, and olefins, and according to its sulfur content, as sweet (less than 0.5% of sulfur in mass) or acid (more than 0.5% of sulfur in mass) (MARTINS, 2003). These properties are highly important due to their economic and technical impact on the refinery process. Therefore, for improved operations it is fundamental to analyze and estimate the quality and the properties of the crude oil prior to its processing at the refinery.

1.5.5 Crude oil characterization

One of the most well-known and used methods for crude oil characterization are the laboratory distillation experiments, in which a sample of crude is distilled at a laboratory facility under specific conditions, following methods standardized by organizations such as ASTM (Association Society for Testing and Materials) and IP (Institute of Petroleum) (YAMANISHI, 2007). One of the most used tests is the TBP (True Boiling Point) due to its accuracy and reliability. According to Nedelchev et al. (2011), the TBP analysis using the standard test ASTM D-2892 is the best method for crude characterization in terms of its boiling points distribution.

The TBP analysis consists of reproducing a distillation process on a laboratory scale, in which samples are taken over time in order to estimate the yields and properties (e.g., specific gravity and sulfur content) at distinct boiling temperature ranges (YAMANISHI, 2007). This analysis uses an arbitrary number of pseudocomponents, also known as cuts, boiling cuts, distillation cuts, or micro-cuts (FAHIM et al., 2009), and it is typically performed with a high number of theoretical stages (15 - 100) and a high reflux rate (RIAZI, 2005). The TBP experiment generates the TBP curve, which provides reliable information to properly estimate the yields and properties of each crude oil distilled fraction. This is especially useful for designing the technical, operational, and modeling/optimization strategies for crude oil refineries (BATISTELLA et al., 2007). However, the TBP analysis is an expensive procedure and requires about two days to be completed (PASQUINI and BUENO; 2007).

1.5.6 Crude oil fractions (distillates)

The crude oil fractions obtained from the distillation process are also referred to as distillates. Table 1.2 presents the crude oil distillates with their respective typical boiling ranges, as well as the main final products (mostly fuels) derived from them at the end of the refinery process.

Table 1.2: Main crude oil fractions obtained from the distillation process.

Distillate	Typical boiling range (°C)	Final refinery product
Liquefied Petroleum Gas (LPG)	-40 – 0	Propane (fuel)
Light Naphtha (LN)	39 – 85	Gasoline
Heavy Naphtha (HN)	85 – 200	Gasoline, Aromatics
Kerosene (K)	170 – 270	Jet fuel, Diesel N° 1
Gas oil (GO)	180 – 340	Heating Oil, Diesel N° 2
Vacuum gas oil (VGO)	340 – 566	Gasoline, Fuel Oil, Lubricant, FCC feed
Vacuum residue (VR)	> 540	Coke, Asphalt, FCCU feed

Source: Adapted from Robinson (2006).

1.5.7 Petrochemical Industry

This section discusses the petrochemical industry, including the exploration and extraction of petroleum, the crude oil refinery, and the crude oil processing operations.

1.5.7.1 Exploration and Extraction of Petroleum

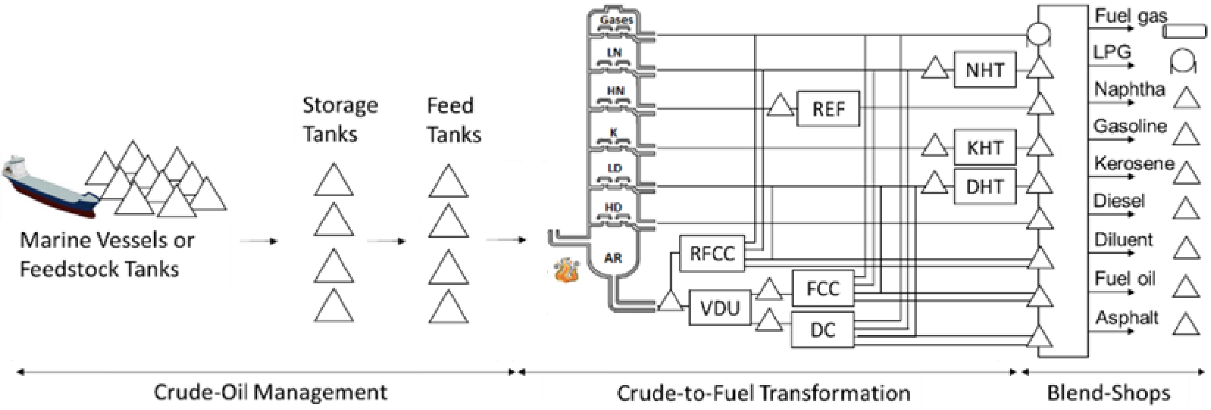
The production chain in the petrochemical industry begins with the exploration of petroleum. Detailed analyses are carried out on the rocky layers of the soil, which provide radiographs of the subsoil and information on the petroleum availability. Some candidate sites are then selected to be drilled, and depending on the technical and economic viability analyses, the petroleum is extracted using special equipment and pipes, stored in tanks or pools, transported through pipelines to the maritime terminals (special ports for loading and unloading), and finally sent to the crude oil refineries to be further processed (SILVA, 2009).

1.5.7.2 Crude oil refinery

The first crude oil refineries date from the 1860s and were predominantly composed of batch distillation units whose main objective was the production of kerosene (AL-QAHTANI and ELKAMEL, 2011). From the 1930s, cracking and coking processing units (i.e., for breaking large molecules into smaller molecules so as to increase the economic value of the material) began to emerge aiming to meet the growing demand for gasoline, both in terms of quality (mostly related to the octane number) and amount produced. Since the Second World War, there have been major development on refinery operations, increasing the complexity of the refineries, as well as the number of different products to meet an increasing and more sophisticated market demand (YAMANISHI, 2007). In general, the main objective of crude oil refineries is the production of energy products (fuels and gases) due to their large demand and vast application, in which the most important and valuable products are gasoline, diesel, and kerosene (MARIANO, 2001).

Crude oil refineries are a complex system network containing multiple unit-operations and flows throughout the process. The refinery design highly relies on the characteristics of the crude to be processed and the products of interest, so that each refinery has its specificities (EPA, 1995). Typically, a crude oil refinery is composed of three main segments that are connected to form a continuous production network, including the crude oil management, the crude to fuel transformation (process-shops), and the product blending and logistic transportation (blend-shops) (AL-QAHTANI and ELKAMEL, 2011). Figure 1.1 presents a simplified typical production network of crude oil refineries.

Figure 1.1: Typical crude oil refinery network.



Source: Adapted from Kelly et al. (2017b).

The first segment concerns the crude oil management. It begins with the arrival of crude at the refinery, usually via ships or underground pipelines, which is stored in storage tanks. From the storage tanks, one or more different types of crude are sent to a blender unit, mixed, and sent to the feed tanks. Properly preparing the crude oil blend to be used to feed the distillation unit is crucial for the refinery due to the high impact in both the economic and technical operations (KELLY et al., 2017b).

The second segment encompasses the crude to fuel transformation processes, also referred to as process-shops, which involve the refinery production units that transform the crude oil into intermediate products. The crude oil in the feed tanks feeds the distillation unit to produce some output fractions referred to as distillates. These fractions undergo multiple conversion and treatment operations to improve their quality and to aggregate additional economic value (DO, 2014; MARIANO, 2001). In summary, the main objectives of these transformation processes are to fractionate the crude oil into more valuable and desired fractions or products, and then improving their quality and purity by converting heavy molecules into lighter molecules, which increases their economic values. In addition, this segment also provides utilities for the refinery, including fuels, electricity, steam, hot water, chilled water, compressed air, nitrogen, etc., which are required in the refinery (AL- QAHTANI and ELKAMEL, 2011).

The third segment including the blending and transportation operations and is referred to as blend-shops. The intermediate products incoming from the processing units are mixed according to final product specifications in order to meet the market demands or contracts. For example, two or more different diesel streams, incoming from different operations in the refinery and with different qualities, can be blended to produce a final diesel stream that meets some required quality specifications (e.g., maximum sulfur content or minimum cetane number). The final fuels are stored in tanks or pools to be later shipped via logistical modes (AL-QAHTANI and ELKAMEL, 2011).

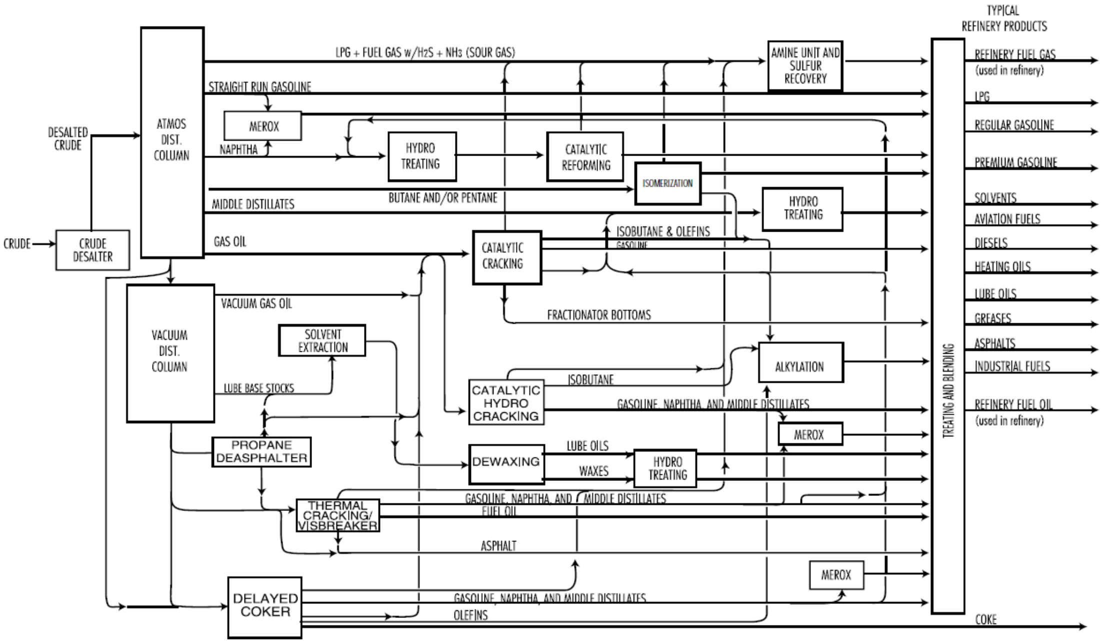
The operations to transform the crude oil into final fuels are highly complex and present a fundamental role in the crude oil refinery. Therefore, the main operations typically found in crude oil refineries are described in the following.

1.5.7.3 Crude oil refining process

There are only a few and not so valuable applications for the raw crude oil. Aiming to increase its economic value, it undergoes several chemical and physical refining processes, which break it down into simpler and more valuable compounds (DO, 2014; MARIANO, 2001). The refining process is typically divided in three categories: separation, conversion and treatment, and mixing.

The separation processes are physical in nature and typically consist of separating one or more valuable or desired components among a mix or blend. At crude oil refineries, the most important separation process is the distillation unit, in which the crude blend is separated into fractions, but maintaining the nature of the molecules unchanged. These distilled fractions are sent to conversion units, where they are chemically altered by adding energy, pressure, catalysts, and hydrogen. The streams from the separation and conversion processes are then treated to eliminate impurities that may reduce or compromise the quality of the final product, and are subsequently blended to produce the final refinery fuels (BUENO, 2003; DO, 2014). Figure 1.2 presents the flow chart of a typical crude oil refining process, including the separation, conversion, treatment, and mixing units.

Figure 1.2: Flow chart of a typical crude oil refining process.



Source: EPA (1995).

The most typical and important processes in crude oil refineries are discussed in the following.

1.5.7.4 Separation Processes

The separation processes separate valuable or desired components from a multicomponent blend or mixture. They are considered a physical-based process because they do not change the structure of the molecules (i.e., breaking chains or rearranging atoms). In the crude oil refining field, they are commonly associated with the distillation unit, in which the hydrocarbon molecules are separated by the difference in their temperature boiling points. The distillation unit can be composed of four different towers, including the atmospheric and vacuum distillation units, the pre-flash column, and the debutanizer (YAMANISHI, 2007). However, according to Zahedi et al. (2011), prior to the distillation process, it is fundamental to treat the feed for the removal of contaminants, a separation process referred to as desalination.

1.5.7.4.1 Desalination

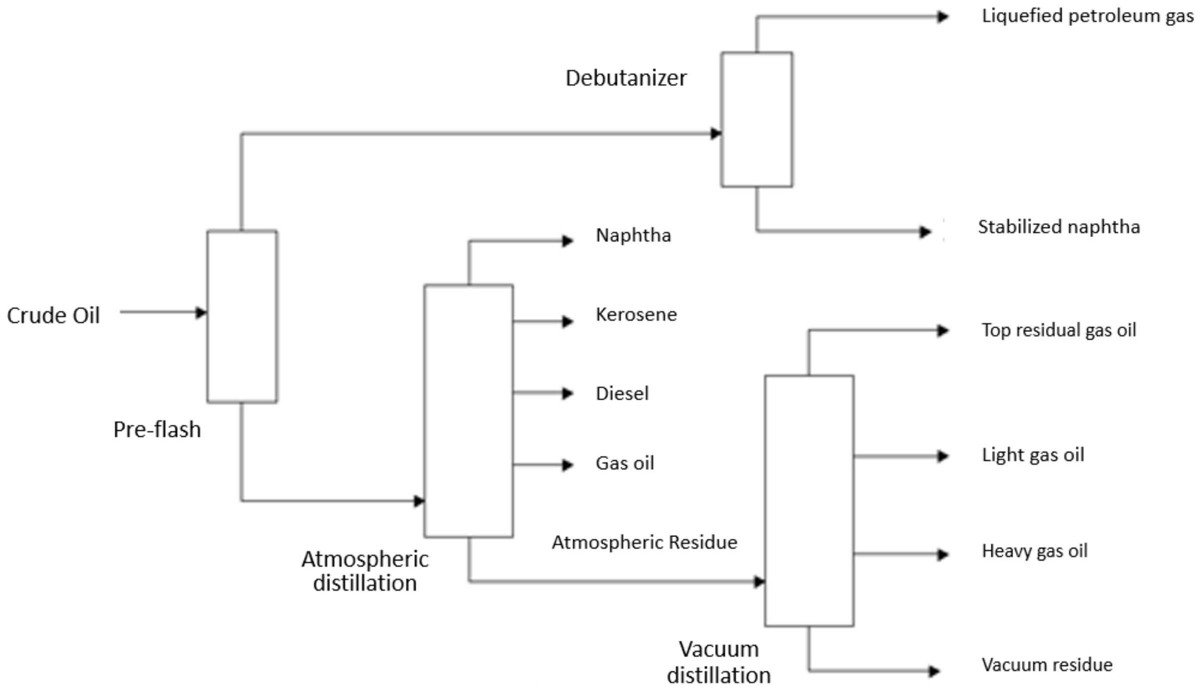
Before being processed in the distillation towers, the crude oil undergoes a pre-treatment for removing contaminants such as dissolved salts that corrodes pipelines and units (SPEIGHT, 2006) and metallic ions and suspended solids cause catalyst deactivation (EPA, 1995). This pre-treatment is referred to as desalination and consists of adding about 3 - 10% water (volumetric base) to the crude oil, and heating that mixture to induce the migration and dissolution of the salts to the aqueous phase. Then, the crude oil is separated from the aqueous phase in a separation vessel by adding demulsifiers or by applying a high electrical potential to coalesce droplets from the aqueous phase. The process of desalination of crude oil generates two liquid residues: an oily sludge and a stream of residual salt water, at high temperature, which is sent to the effluent treatment stations of the refinery (MARIANO, 2001). After the desalination process, the crude oil is preheated in heat exchangers and sent to the distillation unit (YAMANISHI, 2007).

1.5.7.4.2 Distillation unit

Distillation is the process in which a liquid or gaseous mixture of two or more components is separated into fractions by applying or removing heat. It relies on the fact that the components of the mixture have different boiling points, so that the vapor phase formed has a greater number of components with a low boiling point (more volatile) than the original liquid phase. The distillation process is widely used in the chemical industry and represents key operation for the crude oil refinery because of its importance in the production chain. Moreover, it consumes large amounts of energy and can contribute with more than 50% of the refinery operating costs. Thus, efficiently optimizing and controlling the distillation unit represents a potential cost reduction opportunity for the oil refinery (CHEREMISINOFF, 2000).

The distillation unit has great importance in the refining process because of the need to decompose the crude oil into smaller and simpler fractions. Traditionally, two main towers form the distillation unit, namely, the atmospheric distillation column, and the vacuum distillation column. However, more complex and sophisticated refineries also have the pre-fractionating (pre-flash) column and the debutanizer (naphtha stabilizing) column (YAMANISHI, 2007). Figure 1.3 shows the flowchart of an entire distillation unit containing the pre-flash, atmospheric distillation, debutanizer, and vacuum distillation columns.

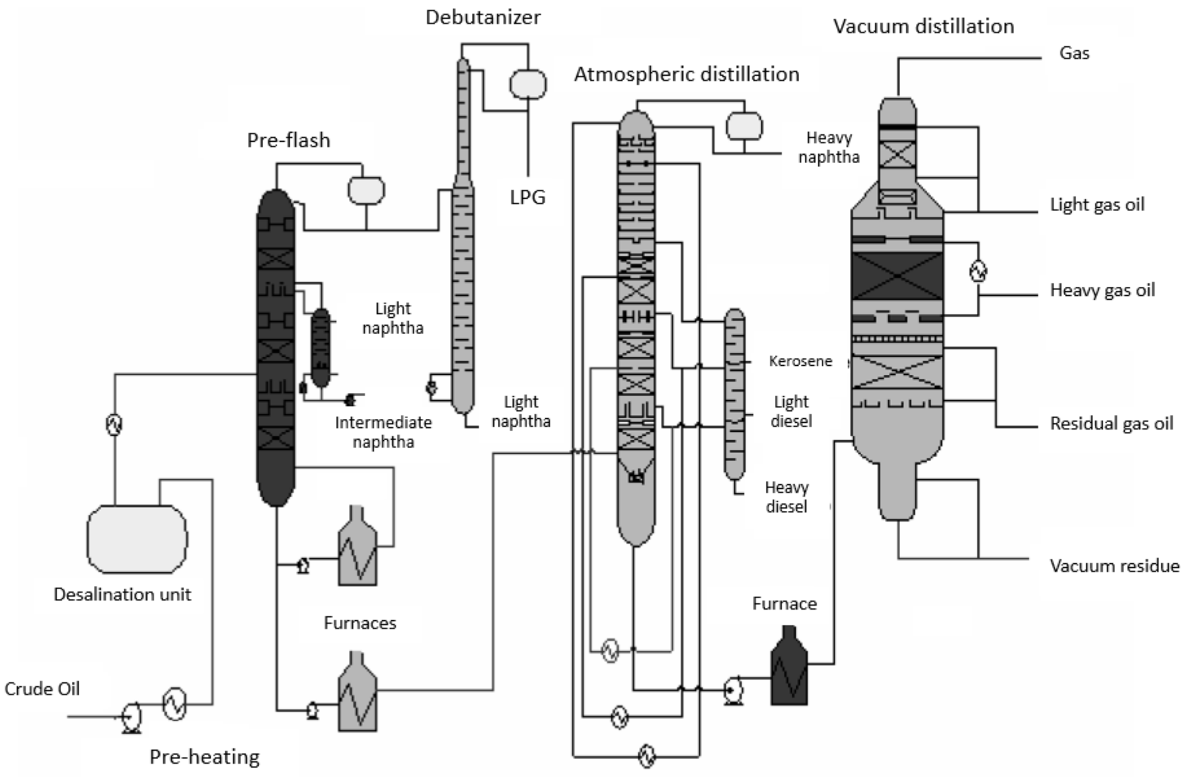
Figure 1.3: Flowchart of the crude oil distillation unit with four towers.



Source: Adapted from Yamanishi (2007).

Figure 1.4 presents a simplified network of a distillation unit composed of four towers, in addition to the heating and desalination systems. Preheated crude oil is fed into the pre-flash tower, also referred to as the pre-vaporizing tower, whose main purpose is to increase the processing capacity of the distillation unit by processing part of the total feed. In this tower, the lighter fractions of crude oil are separated from the main stream, reducing the total load of the atmospheric distillation tower (SILVESTRE, 2005). According to Cheremisinoff (2000), a secondary objective of the pre-flash tower is to improve the energy efficiency of the process.

Figure 1.4: Simplified distillation unit network with four towers.



Source: Adapted from Silvestre (2005).

The top stream of the pre-flash tower, which is basically composed of fuel gas (FG), liquefied petroleum gas (LPG), and light naphtha, is sent to the debutanizer tower, whose purpose is to “stabilize” the naphtha produced in the distillation towers and separate it from the light gases (VENTIN, 2010), in addition to specifying some chemical properties, such as naphtha vapor pressure and LPG weathering (SILVA, 2009). The impure LPG is sent for a treatment to remove sulfurous compounds and it is further sold as LPG (cooking gas). The stabilized naphtha is stored in a tank or pool, and is further blended with naphtha streams from other units (e.g., catalytic cracking) to produce gasoline (SILVA, 2009).

The bottom product of the debutanizer is sent to an atmospheric distillation column, also referred to as the crude distillation unit. However, the CDU can also be fed directly with raw crude oil. In this case, the desalinated crude is pre-heated in a furnace to a maximum temperature of around 400 °C to avoid thermal decomposition. (MARIANO, 2001).

Inside the towers, there are fractionation trays (plates) that provide the separation of the crude oil in various fractions due to the difference in the temperature boiling points of the molecules, because the closer to the top of the column, the lower the temperature. Thus, the rising steam, when in contact with each tray, undergoes a partial condensing. Hydrocarbons whose boiling points are greater than or equal to the temperature of a given tray are retained in that tray. The composition of the liquid varies in each stage of the tower, in which the liquid becomes heavier as it approaches the bottom of the tower (and lighter as it approaches the top). As the fractions condense in a given tray, the excess of liquid is spilled on the bottom tray. Upon reaching the bottom tray, which is at a higher temperature, the light fractions from the upper tray are vaporized. The liquid that overflows from tray to tray is known as internal reflux. The light components that have not condensed in any tray leave the top of the column (light naphtha and LPG); the side products are heavy naphtha, kerosene, and diesel; and the heavier fractions (atmospheric residue, ATR), not vaporized, are sent to the vacuum distillation column (EPA, 1995; MARIANO, 2001).

To decompose the atmospheric residue into distinct fractions, high temperatures would be needed inside the distillation column (operating at atmospheric pressure). For circumventing this expensive process due to the large amount of energy required to achieve high temperatures, the vacuum distillation column is used, which operates at low pressures to provide the fractionation of the atmospheric residue fraction while operating at temperatures below the formation of coke and the cracking of the crude oil (YAMANISHI, 2007). In the vacuum distillation, the heavier fractions of crude are distilled again, but under low pressure (0.2 to 0.7 psia) to increase the volatilization of the compounds and hence, achieve an efficient fractionation. The vacuum inside the tower is typically maintained through steam ejectors and vacuum pumps, in addition to injecting superheated steam at the base of the column to reduce the partial pressure of hydrocarbons in the tower, facilitating the vaporization and separation (GARY and HANDWERK, 1994). The main side stream products are the light and heavy vacuum gas oils, while the bottom product is the vacuum residue (YAMANISHI, 2007).

1.5.7.5 Conversion Processes

The conversion processes aim to aggregate value to the distillate fractions and to improve their qualities. Unlike separation processes, conversion processes are chemical in nature and use break down, regrouping, or molecular restructuring reactions. Some of the main conversion processes used in the crude oil refining includes cracking (thermal and catalytic), coking, alkylation, isomerization, polymerization, and catalytic reform (MARIANO, 2001). Some of these conversion processes are described in the following.

1.5.7.5.1 Thermal cracking

Thermal cracking uses moderate pressure and high temperatures (above 350 °C) to break large hydrocarbon chain into smaller and lighter molecules. These reactions are thermodynamically favorable at high temperatures and involve breaking carbon-carbon bonds (HOCKING, 2016; SPEIGHT, 2006). The thermal cracking feed are typically heavy diesel and vacuum residue. The outputs fractions are fuel gas, LPG, and naphtha. Diesel is also produced in the thermal cracking, but it is typically recycled to improve the efficiency of the unit. The thermal cracking process has been increasingly replaced by the catalytic cracking (EPA, 1995).

1.5.7.5.2 Catalytic cracking

Catalytic cracking utilizes heat, pressure, and a catalyst, to break large hydrocarbon molecules into smaller, lighter, and more valuable molecules (BUENO, 2003). Its main feed stream is the vacuum gasoil, which when subjected to high pressures and temperatures under the presence of a catalyst, is decomposed into several lighter fractions, such as fuel gas, liquefied petroleum gas (LPG), cracked naphtha, and light cycle oil (LCO). The bottom product is the decanted oil (MOREIRA, 2006). The main products are gasoline (50 - 65% by volume) and LPG (25 - 40% by volume) (BUENO, 2003).

1.5.7.5.3 Delayed Coking

The delayed coking unit thermally converts crude oil residues into liquid streams with increased economic value. The vacuum residue is fed to the delayed coking unit, heated up to 487 - 520 ° C, and sent to coke drums, where the cracking process occurs (ROBINSON, 2006). The main products of this unit include the fuel gas, LPG, naphtha, gas oils, and coke (LIMA, 2012).

1.5.7.5.4 Catalytic reforming

The main objective of the catalytic reforming is to transform heavy low-octane naphtha into high-octane petrochemical products used for the production of gasoline (octane is a property that indicates the resistance of the air-fuel mixture under high pressures and temperatures, so that the higher the octane the greater the energy efficiency of the fuel). These products are referred to as reformates and have a high content of benzene, toluene, and xylene, which is an excellent source of aromatics for the refinery (ROBINSON, 2006). The catalytic reforming also produces liquefied petroleum gas, fuel gas, acid gas, and a stream rich in hydrogen, which can be used in catalytic hydrotreatment units that do not require large streams or high purity (BUENO, 2003).

1.5.7.5.5 Isomerization

The main application of the isomerization process in the crude oil refining is to increase the octane number of paraffins through the conversion of paraffins into iso-paraffins. A very common example is the conversion of n-butane into iso-butane, which can later be alkylated in liquid hydrocarbons within the gasoline boiling range (SPEIGHT, 2006). The isomerization process involves the contact of hydrocarbons in the presence of a catalyst and in favorable conditions, with a temperature range of 90 °C - 200 °C (MARIANO, 2001).

1.5.7.5.6 Alkylation

Alkylation is a process of combining paraffins and olefins to form larger, higher molecular weight, and more branched molecules. One of the most common alkylation process is the reaction of iso-butane with iso-butene or butene, producing iso-octane. A catalyst is typically used to increase the conversion rate of the process. According to Mariano (2001), a typical alkylation unit consists of two main sections, for the reaction and for the recovery and purification of the catalyst.

1.5.7.5.7 Polymerization

Polymerization is a process in which low molecular weight molecules are transformed into higher molecular weight molecules, but maintaining the same molecular arrangement (SPEIGHT, 2006). According to Robinson (2006), this process is employed to convert C3 and C4 olefins into C6 to C9 olefins (e.g., conversion of iso-butylene to diisobutylene). These reactions typically occur under high pressure and in the presence of a catalyst (MARIANO, 2001).

1.5.7.5.8 Super Fractionation

The super fractionation unit aims to separate compounds with similar volatilities, such as propane and propylene. A high number of trays and a high reflux rate are typically necessary to obtain an efficient separation (GOKHALE, 1995).

1.5.7.6 Treatment Processes

After the separation and conversion processes, it is typically required to treat the intermediate products for the removal of impurities and contaminants, which improves both their quality and economic value. In some cases, treatment processes may also be performed before specific units in order to treat their feed. One of the most used is the hydrotreating.

1.5.7.6.1 Hydrotreating

According to Robinson (2006), hydrotreating is not considered a conversion process because the breaks in carbon-carbon bonds are minimal. Its main purpose is to remove impurities such as sulfur, nitrogen, oxygen, halides, and metals from the product stream (which can be either an intermediate or final product, or the feed to a separation or conversion unit), in addition to improving the stream quality by converting olefins and diolefins into paraffins. Hydrotreating can be used not only for the purification of products, but also before processes where catalytic deactivation by sulfur or nitrogen can occur, such as catalytic reform and hydrocracking (EPA, 1995; ROBINSON, 2006). In this process, fixed bed reactors are generally used with large amounts of hydrogen in the presence of catalyst and under high pressure and temperature. Some of the fractions that typically undergo to the hydrotreating are naphtha, kerosene, and diesel (EPA, 1995).

1.5.7.7 Blending, Specification, and Logistic Distribution

After the separation, conversion and treatment processes, the intermediate products are sent to mixing units (blenders) to produce the final products according to the market demands or contracts to be met. These products have to be specified regarding their properties, such as maximum sulfur content, minimum cetane or octane number, etc. After the blending operations, the final products are stored and further sent via logistical modes (DO, 2014).

1.5.8 Crude oil refinery scheduling

For improved, more efficient, and safer operations, the highly complex crude oil refinery problem has to be determined or scheduled in advance, prior to the implementation in the plant. This is one of the most relevant and important applications for the refinery, with great economic and technical impact, and is referred to as the crude oil scheduling. This decision-making process consists of determining the decisions to be made and the operations to be carried out throughout the entire refinery. These decisions include choosing the process variables, which can be either binary (e.g., which units and flows are operating) or continuous (e.g., amounts, flows, compositions). The decision-making in crude oil refineries typically focuses on the processing of crude oil and intermediate products to produce the final fuels, in which some of the decisions include which and how much of each crude to process, where to store the feedstocks and intermediate products, and when to perform each of these decisions. The decision-making for the crude oil scheduling is typically assisted by mathematical formulation and problem-solving techniques.

Modeling the crude oil scheduling within the interval of hours, shifts, or days, is a very difficult problem due to the complexity of the process, including the multiple logistics decisions and the processing of hydrocarbon streams in liquid and gaseous states. In the petroleum refining industry, equipment, tanks, and the physical separation and chemical reaction units, operate continuously by processing feedstocks of different qualities composed by hydrocarbon streams with heteroatoms (sulfur, nitrogen, and metals) to produce treated fractions formed by similar compounds (i.e., same boiling points). After the refining process, the final products or fuels are suitable in terms of specifications to be sold and used over multiple applications (KELLY et al., 2017b).

Due to the complexity of the crude oil refining process, a proper, accurate, and efficient modeling requires dozens of thousands of variables and constraints within a non-convex MINLP formulation (KELLY et al., 2017b). Hence, the crude oil refinery scheduling was historically carried out via spreadsheets and simulators in which most decisions were made manually in a trial and error procedure (MENEZES et al., 2015a). However, with the recent advances in decision-making modeling and optimization, solving algorithms, and computer-aided resources (FRANZOI et al., 2018; BRUNAUD et al., 2020), the automation of an efficient scheduling decision-making for crude oil refinery applications has become possible and essential for the competitiveness of the refineries (KELLY and MANN, 2003).

According to Li et al. (2016), because of the increasing market competitiveness, in addition to factors such as the price fluctuations for both the feedstocks and the demand of fuels, deterioration of product quality, and strict environmental regulations, crude oil refineries have to explore the use of advanced software and tools aiming to improve the refinery operations both technically and economically. In that context, the use of computational tools for industrial scheduling applications emerges as an important field of study (REDDY et al., 2004a). The main benefits include the systematic search for optimal solutions, which are typically better than the solutions found manually or based on simulation software; significantly reduce the time and effort required to build a new or updated schedule; and provide tools for handling adverse or unexpected events (e.g., disruptions in the production chain) by automatically updating the incumbent schedule (JIA et al., 2003; KELLY and MANN, 2003).

Due to the complexity of crude oil refining operations, it is common to divide the refining process in three sections, namely the crude oil scheduling, the process-shops, and the blend-shops (JIA and IERAPETRITOU, 2004; SHAH et al., 2010). The crude oil scheduling is the most impactful section for the refinery mostly because both the technical operations and the refinery economics depend on the feed to be processed, which is determined from the scheduling operations (KARUPPIAH et al., 2008; LOTERO et al., 2016; SHAH and IERAPETRITOU, 2015). The potential for optimizing and improving the crude oil refinery operations includes several steps, since the beginning of the production chain. Kelly et al. (2017a) state that the crude oil quality (i.e., the yields and properties of each crude fraction) significantly affects the yields and qualities of the final products. This is enhanced by the large combination of distinct crude oil blends that can be chosen. Thus, in addition to its key importance for the refinery, the crude oil scheduling is also one of greatest challenges in terms of modeling, optimization, and implementation in the plant (SHAH et al., 2010).

1.5.9 Mathematical Modeling

Mathematical modeling has been increasingly employed to achieve improved operations and to exploit cost reduction opportunities for industrial applications. The models are a mathematical representation of the process and are helpful to predict, calculate, simulate, and investigate possible scenarios. Thus, the best or desired scheduling solutions can be computationally tested and analyzed in advance prior to their selection and implementation in the real process. These models are especially helpful for allowing simulation and optimization techniques to be applied, in which the variables can be estimated and the operational conditions can be analyzed, providing better process control and safer, smoother, and more efficient operations.

The mathematical models are classified over distinct categories according to their formulation, including linear programming (LP), nonlinear programming (NLP), mixed-integer programming (MIP), which englobes the mixed-integer linear programming (MILP) and the mixed-integer nonlinear programming (MINLP). The nonlinear models include nonlinear information, either in the variables, constraints, or objective function, while the mixed-integer models include binary decisions or variables (WINSTON and GOLDBERG, 2004). The crude oil refinery scheduling is typically formulated as an MINLP problem due to the simultaneous binary decisions and nonlinear information (HOU et al., 2016; YÜZGEÇ et al., 2010), and it is one of the most challenging tasks for the crude oil refineries.

1.5.9.1 Time horizon and discretization

When building the mathematical model for a given problem, it is needed to establish the time limits or bounds in which the problem is formulated (i.e., defining the initial and final instants of time to be modeled), which is referred to as the time horizon. The time horizon length may be in the order of seconds or minutes (e.g., real time optimization and control applications), hours or days (e.g., crude oil scheduling), months or years (e.g., procurement planning and process design). Aiming to achieve improved and more accurate solutions, the time horizon is typically decomposed or discretized into time steps, providing additional degrees of freedom in the modeling and optimization.

There are two distinct approaches for the time discretization. In the discrete time formulation, the time is discretized uniformly and all events must begin and end at the limit (i.e., beginning or end) of a time step. The continuous time formulation considers the time as a continuous variable, so the exact start and end dates of each event can be freely specified within the time horizon (HOU et al., 2016; SAHARIDIS et al., 2009; MÉNDEZ et al., 2006a). While the main drawback of the discrete time formulation is the combinatorial complexity that scales with the number of binary decisions, implementation issues arise from continuous time formulations, in which the execution of tasks by the operators in the plant cannot be easily performed, in addition to the need for defining the time events to be selected to represent the operations in the plant (KELLY et al., 2017b).

Because of the issues for solving mixed-integer industrial problems using discrete time formulations given their intractability due to the large number of binary decisions, the literature on crude oil scheduling optimization in the last few decades mostly employed continuous time approaches. However, with the recent technological advances on decision-making modeling, solving algorithms, and computer-aided resources, the utilization of discrete time approaches has become possible for large-scale industrial applications (KELLY et al., 2017b), although it still requires assistance of additional modeling (decomposition) and solving (heuristic) strategies.

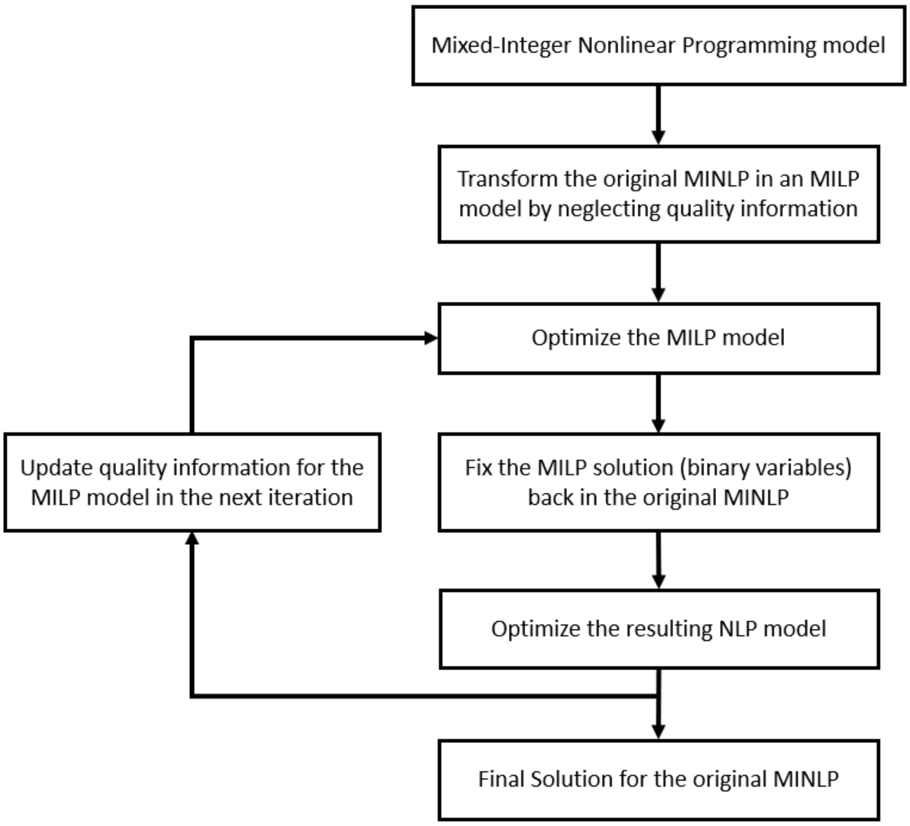
1.5.9.2 Heuristic approaches and decomposition algorithms

Heuristic approaches and decomposition algorithms have been widely used for handling intractable problems. For the industrial-sized scheduling of complex phenomenological (separating, converting, blending) and procedural (sequences, setups, startups) optimizations, the mathematical formulations may be difficult to be solved as a full space MINLP model. In order to properly determine the optimal scheduling operations, it is required an accurate modeling including the logistics and quality information throughout the process, in addition to utilize efficient problem solving techniques via software optimization, which typically uses commercial solvers (e.g., CPLEX). However, there are modeling and optimization limitations for handling large-scale MINLP formulations such as the crude oil refinery scheduling problem.

Therefore, several works in the literature have been developed strategies to break down the MINLP formulations into smaller, simpler, and less complex subproblems (MOURET, GROSSMANN, and PESTIAUX, 2009; LOTERO et al., 2016; ASSIS et al., 2019) in order to handle such complicated models that vary in a three-dimensional MINLP quantity-logic-quality (QLQ) relationship space. Typical decompositions lead to MILP-NLP formulations to be sequentially solved until a convergence criterion is met (WENKAI et al., 2002; MOURET et al., 2009; CASTRO and GROSSMANN, 2014; CAFARO et al., 2015; KELLY et al., 2017a).

Menezes et al. (2015b) propose a phenomenological decomposition heuristic (PDH) approach for a strategic planning problem to determine the refinery configuration and the process unit dimensions. The problem is formulated as an MINLP, which is decomposed into MILP and NLP sub-models. First, the MILP model is optimized considering only logistic and quantity information (i.e., by neglecting the quality information in the original MINLP model). Then, the binary variables from the optimal MILP 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 the new coefficients or parameters in the MILP problem in the next iteration, with logic and quantity variables under consideration. Kelly et al. (2017b) applied the same method for scheduling applications, in which multiple MILP solution are generated for improved performance and convergence of the algorithm. An iterative procedure is performed, in which the best NLP solution (i.e., yields and quality information) is retro-fed to the MILP in the next iteration, until a convergence criterion is met. Figure 1.5 illustrates the phenomenological decomposition heuristic procedure.

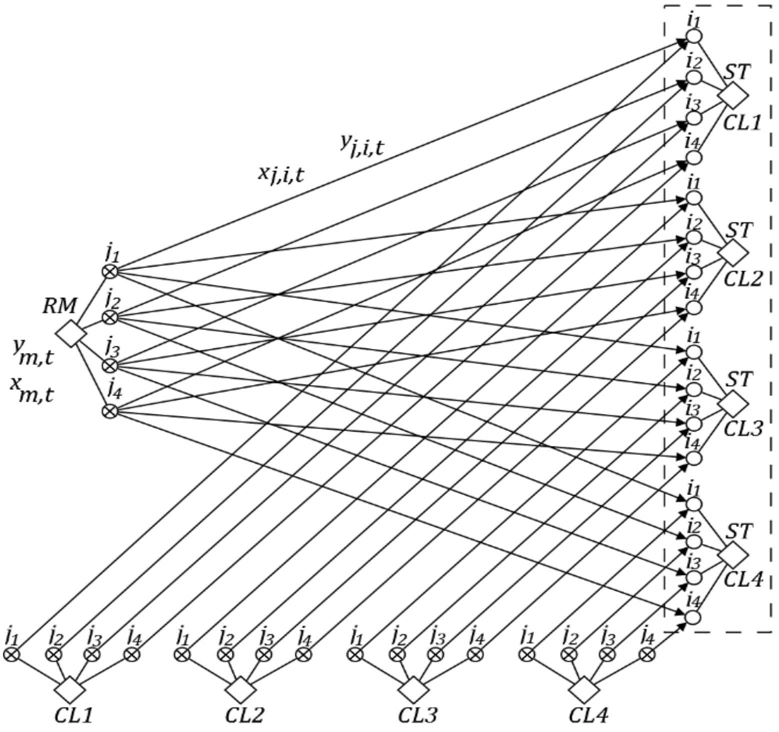
Figure 1.5: Phenomenological decomposition heuristic.


Mixed-Integer Nonlinear Programming model
Transform the original MINLP in an MILP model by neglecting quality information
Optimize the MILP model
Update quality information for the MILP model in the next iteration
Fix the MILP solution (binary variables) back in the original MINLP
Optimize the resulting NLP model
Final Solution for the original MINLP

Source: Author (2021).

Several other heuristic-based strategies have been developed to reduce the computational burden for time-limited applications. Kelly et al. (2017a) develop a mixed-integer linear programming model to design pre-assignments of distinct feedstocks with different qualities when storing them from supply sources to shared storages (i.e., clustering multiple feedstocks in the same pool). This method assigns individual units or sources (e.g., crude oils, tanks, feedstocks) to a limited number of storage or sinks, and specifies the variables to be clustered (i.e., compound-properties). This clustering strategy is similar to the concept found in many sequence-dependent changeover heuristics (e.g., using product-wheels and blocking), in which the individuals are grouped in families according to some common criteria. This is especially helpful when there is a limited storage space, and for large-scale applications typically found in the crude oil, metal, and food processing industries. The storage assignment formulation minimizes the quality deviation in the clustering of a larger number of feedstocks into a smaller number of pools. This method designs simple and straightforward segregation rules in order to achieve better crude oil management, crude oil blend control, and improved blend scheduling optimizations. Figure 1.6 illustrates the feedstock storage assignment method, in which the raw materials RM are assigned to the storage sinks ST under distinct clustering groups CL.

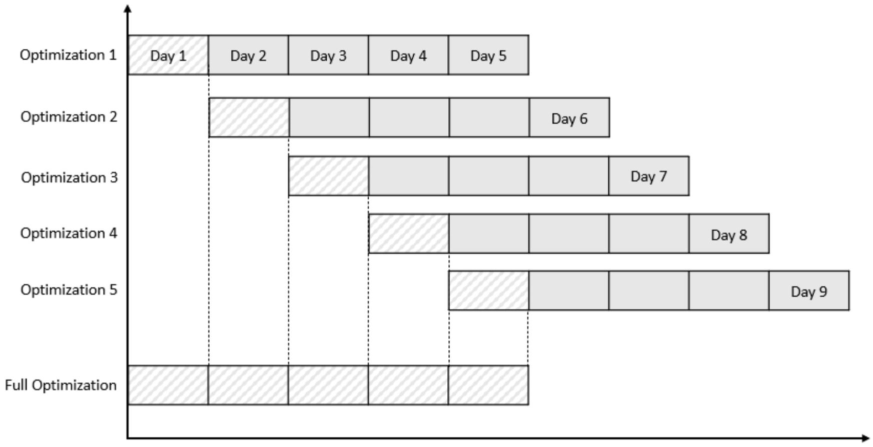
Figure 1.6: Feedstock storage assignment flowsheet.



Source: Kelly et al. (2017a).

Kelly (2002) developed the chronological decomposition heuristic (CDH), a straightforward time-based strategy used in the search of integer-feasible solutions. The CDH is specifically designed for discrete-time scheduling optimization problems typically found in the petrochemical, chemical, and pharmaceutical industries. This heuristic decomposes the model regarding its time dimension (i.e., the full time horizon is discretized or decomposed into smaller steps). Each sub-model is then solved using mixed-integer linear programming (MILP) techniques starting from the first model onwards. Thus, instead of optimizing one large problem over the entire time horizon, multiple sequential time-discretized models are solved. Figure 1.7 illustrates the concept of the chronological decomposition heuristic over a rolling horizon.

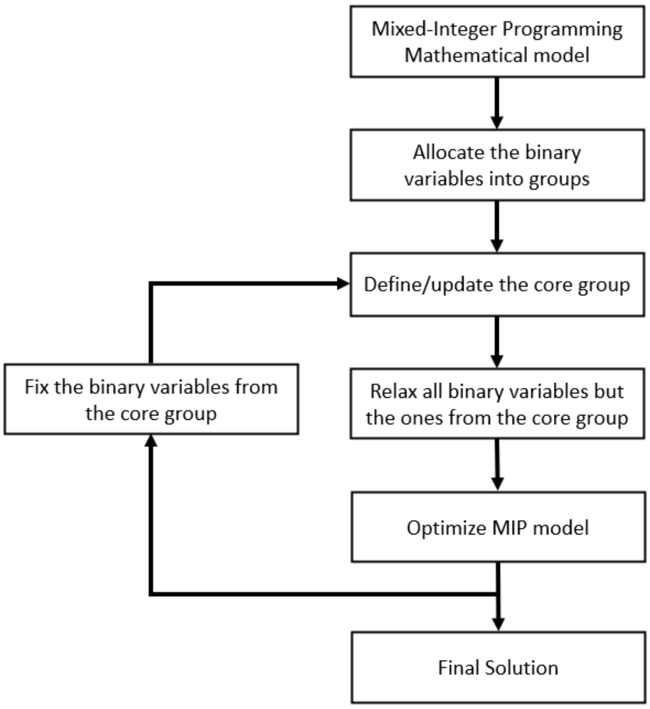
Figure 1.7: Rolling horizon chronological decomposition strategy.



Source: Author (2021).

Aiming to tackle large-scale MILP problems, Kelly and Mann (2004) developed the flowsheet decomposition heuristic (FDH) to reduce the computational time needed to find good integer-feasible solutions for industrial applications. This method is a sort of relax-and-fix heuristic, which allocates individual units or sources (e.g., feedstocks, tanks) into groups, and solves mixed-integer linear programming problems (as many as there are groups), given a pre-specified order of importance. In each MILP subproblem, a group of units is defined as the core (most important) group. All binary variables that are not included in the core group are relaxed, and the MILP is optimized. From the optimal solution, the binary variables that are included in the core group are fixed to their optimal values, and the algorithm moves to the next iteration until convergence is met (i.e., after optimizing as many MILPs as there are groups). At this point, all binaries will have been fixed, so that the solution for the last MILP optimization is also a feasible solution of the original MILP. Figure 1.8 illustrates the flowsheet decomposition heuristic relax-and-fix approach.

Figure 1.8: Relax-and-Fix decomposition approach.


Mixed-Integer Programming Mathematical model
Allocate the binary variables into groups
Define/update the core group
Fix the binary variables from the core group
Relax all binary variables but the ones from the core group
Optimize MIP model
Final Solution

Source: Author (2021).

The phenomenological decomposition heuristic, the feedstock assignment strategy, and the Unit-Operation-Port-State Superstructure (UOPSS) (KELLY, 2005) are simultaneously applied for the modeling and optimization of an industrial size problem for the future 7-day time horizon discretized in 2-hour intervals. The formulation includes 5 atmospheric distillation units in 9 operational modes and 35 storage and feed tanks. The logistics problem (MILP) has around 30 thousand continuous variables and 30 thousand binary variables, 6.5 thousand equations, 80 thousand inequalities, and 54 thousand degrees of freedom, and is solved in 128.8 seconds using 8 threads in the solver CPLEX 12.6 (International Business Machine IBM, USA). The quality problem (NLP) has over 102 thousand continuous variables, 58 equality constraints, 768 inequality constraints, and 45 thousand degrees of freedom, and is solved in 10.3 minutes in an sequential linear programming (SLP) approach using the solver CPLEX 12.6. The MILP-NLP gap between the two solutions is below 3.5% after two iterations of the algorithm.

1.5.10 Crude oil scheduling optimization

The first optimization approach for crude oil scheduling problems used a discrete time-based formulation (LEE et al., 1996). The main advantage of this type of formulation is introducing and simplifying how the events are discretized throughout the time horizon, so that the decisions and operations to be performed in the real process must belong to a specific time period (FLOUDAS and LIN, 2004). That is especially useful to facilitate the implementation, execution, and coordination of the schedule by the operators and workers in the industrial plant (KELLY et al., 2017a). However, due to the combinatorial complexity of discrete time modeling, most of the works in the literature addressed the use of continuous time formulations (JIA et al., 2003; REDDY et al., 2004a, b; MOURET et al., 2009, CASTRO and GROSSMANN, 2014; CAFARO et al ., 2015; XU et al., 2017), although Kelly et al. (2017a) state that technological improvements in the last decades have reduced the simulation and optimization computational effort by two orders of magnitude in comparison with the 1990s, which opens opportunities for large-scale industrial applications using discrete time formulations.

Shah (1996) develop an MILP model for the crude oil scheduling using a discrete time formulation, in which the problem is decomposed into two stages to be sequentially solved. The first stage defines the refinery operating conditions and the crude oil supply to the refinery, while the second stage defines the loading and unloading schedules between the refinery and the ports or ships. This approach guarantees feasibility but not optimality for the scheduling solution.

Lee et al. (1996) consider the short-term crude oil scheduling problem, including the crude unloading from ships or pipelines to the refinery storage tanks, and the prepare of the crude blend to feed the distillation units. The problem is initially formulated as an MINLP, in which the nonlinearities arise from the blending equations. However, the authors substitute bilinear terms for individual flow component terms, yielding an MILP problem. The model is formulated in discrete time and a branch and bound method is employed for the solving procedure.

Wenkai et al. (2002) state that the linearization of bilinear terms proposed by Lee et al. (1996) often leads to inconsistent solutions due to the composition discrepancies (i.e., the crude oil composition sent from a tank to the CDU is different from the composition of the blend inside the tank). In order to overcome this issue, Wenkai et al. (2002) propose an algorithm that iteratively solves two MILP and one NLP problems. Their algorithm also reduces the total number of binary variables and uses heuristic techniques for large-scale problems, reducing the required computational effort.

Jia et al. (2003) address the short-term crude oil scheduling problem including the crude oil unloading, blending, and processing in the distillation units, and develop an MILP continuous time formulation based on the state-task network (STN) representation.

Reddy et al. (2004a) prove the existence of the composition discrepancy in the work of Lee et al. (1996) and show that the algorithm proposed by Wenkai et al. (2002) eliminates the composition discrepancy but may eventually fail even to obtain feasible solutions. Considering the same crude oil scheduling problem, Reddy et al. (2004b) propose an MILP continuous time formulation and develop an iterative algorithm that eliminates the composition discrepancies and solves the MILP models for time horizons of seven days. The authors compare the formulations in continuous and discrete time and claim that the former is preferable for complex problems while the latter is preferable for simpler and smaller problems. Alternatively, Reddy et al. (2004a) propose an MINLP formulation with a hybrid representation of time, in which the time horizon is discretized into time steps of 8 hours, but the events do not necessarily need to start and the end of a given time interval. The solution algorithm proposed by the authors is able to mitigate composition discrepancies, in addition to simultaneously solving the entire problem without the need to separately solve MILP and NLP subproblems.

Li et al. (2007) review the algorithm proposed by Reddy et al. (2004a) and Reddy et al. (2004b) and develop an improved, faster, more robust, and more efficient algorithm, in which the problem is reformulated for the future 20-day time horizon. Van Elzakker et al. (2010) review the algorithm proposed by Li et al. (2007) and perform cuts to remove infeasible combinations of binary variables so as to reduce the computational time. Moreover, their updated algorithm also increases the total profit by including inventory costs in the optimization. Li et al. (2012) study the same crude oil scheduling problem addressed in Li et al. (2007), and introduce three additional contributions: a

MINLP model formulated in continuous time, a linear piecewise estimate of bilinear terms, and a global optimization algorithm based on the branch and bound method.

Méndez et al. (2006b) develop an approach that simultaneously optimizes the crude oil scheduling and the gasoline offline blending. The proposed method uses linear approximations and an iterative procedure through successive LP or MILP problems to estimate the properties of the products, and is flexible to consider both discrete and continuous time representations depending on the characteristics of the problem.

Furman et al. (2007) propose a non-convex MINLP based on a continuous time formulation to optimize the transfer of streams between tanks in the crude oil refinery. According to the authors, in order to reduce the number of decision events (i.e., binary variables) and, consequently, the computational time, the model allows simultaneous inlet and outlet flows at the same time step, as long as they are not from the same tank.

Karuppiah et al. (2008) present an outer approximation (OA) algorithm to globally solve a non-convex MINLP using a continuous time formulation for the crude oil scheduling problem, including the steps from the crude unloading to the distillation unit. The proposed algorithm solves a relaxed MILP problem of the initial MINLP, in addition to using cutting plane techniques to reduce the computational time.

Pan et al. (2009) study the crude oil scheduling problem and use heuristic rules to linearize bilinear terms and to fix part of the binary variables in order to simplify the model by converting it from an MINLP to an MILP. The proposed algorithm is able to avoid composition discrepancies without using iterative techniques, and maximizes the refinery profit while meeting the product demand.

Zhang and Xu (2015) study the crude oil scheduling problem and consider additional information such as the brine decantation time and multiple crude inlet flows to the refinery. The authors implement an iterative outer approximation algorithm based on the techniques applied by Karuppiah et al. (2008) to obtain an optimal solution to the non-convex MINLP scheduling problem.

Lotero et al. (2016) address the MINLP blending problem of final distillates from the crude oil refinery process, and propose an algorithm that decomposes the original MINLP in two levels. In the first level, a relaxed MILP of the original MINLP is solved, and in the second level, a reduced MINLP is solved by fixing some of the binary variables of the original model.

Kelly et al. (2017a) propose a discrete time MILP formulation for the feedstock assignment problem, in which the feedstocks arriving at the refinery are clustered in groups according to their qualities (i.e., the feedstocks with similar characteristics are stored in the same pool). The objective function minimizes the quality deviation when mixing two or more different crudes. That provides a significant improvement in the control of the properties of the crude oil blends, as well as a reduction in the size of the problem (and hence, a reduction in the computational effort for the optimization procedure).

To the best of our knowledge, despite an extensive literature on the crude oil refinery scheduling, the works on this topic address a partial modeling of the refining process (i.e., the models include either the crude oil scheduling, the blend-shops, or specific process units, but do not integrate all of them within a simultaneous solving procedure). There is a single study (XU et al., 2017) that models the entire refinery process, including the crude oil scheduling, the process-shops, and the blend-shops. However, the continuous time model for a time horizon of two weeks considered less than 100 binary variables, which is not coherent with large-scale industrial operations.

1.5.11 Online Scheduling

Although the refinery scheduling represents a network with continuous processes in operation, a finite time horizon must be defined so as to build and solve the mathematical model for this problem. The time horizon length should be chosen according to the desired application and the computational resources available. After the modeling and optimization, as the schedule is implemented in the real process, uncertainties and disturbances arise, and new information becomes available and should be used as soon as possible for achieving improved operations (GUPTA et al., 2016a), especially because changes as disruptions, delays, and market fluctuations, may result in sub-optimality and infeasibilities in the incumbent schedule (GUPTA et al., 2016b). The operational data used in the crude oil scheduling are typically out of date or not integrated with the production, which leads to inconsistencies in the prediction throughout the process (MENEZES et al., 2015c). Thus, a continuous cycle of improvement is required to reduce the deviation between the model predictions and the actual values in the plant. Therefore, an efficient online scheduling (rescheduling) procedure facilitates the adaptation of the schedule under uncertainties and unforeseen events, in addition to considering new information as soon as possible, which aims to increase the economic value of the process (GUPTA et al., 2016a).

The term online scheduling is also referred to as rescheduling and indicates the act of updating the incumbent schedule as a corrective action due to changes, uncertainties, or disturbances in the process, which may be beneficial (e.g., arrival of new feedstocks, new market demands of products) or not (e.g., malfunction of units, disruptions in the process) (VIERIA, HERRMANN and LIN, 2003; LI and IERAPETRITOU, 2008). According to Gupta et al. (2016b), the rescheduling should be performed not only when necessary, but on a regular basis.

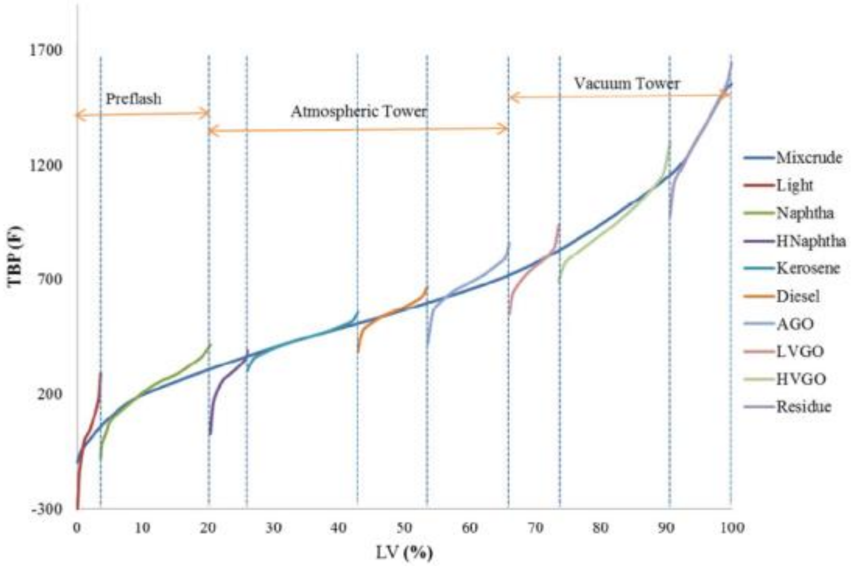
Zhang et al. (2015) develop a full rescheduling strategy (FRS) for the crude oil refinery front-end problem (i.e., crude oil management). The refinery production is rescheduled whenever there is a malfunction in a tank due to mechanical problems. That mitigates operational infeasibilities and achieves a better efficiency and process control. Pattison et al. (2016) propose a periodic rescheduling mechanism for an air separation unit that includes in the model information regarding prices and demands, as well as disturbances such as process disruption.

1.5.12 Cutpoint temperature modeling

A fundamental step in the crude oil refinery scheduling optimization is to calculate the yields and properties of distillation units (FU and MAHALEC, 2015). For such calculation, both rigorous and surrogate representations can be used. The rigorous or first principals modeling considers mass (or molar) and energy balances in the columns. As a result, compositions and flows of internal and external streams as well as operational conditions, such as pressure and temperature of the stages, can be determined. However, despite the sufficient robustness and accuracy to predict the yields and properties of distillation units, rigorous models demand higher computational effort, imposing difficulties for their application in large-scale problems. On the other hand, non-rigorous modeling can use surrogate or simplified correlations based on measured data to represent the modifications promoted by the mass and energy balances over the hydrocarbon components. Due to their simplicity, straightforward application, and relatively good accuracy, they are commonly used for process optimization in oil refineries (LI *et al.*, 2005).

Crude oil is a complex mixture of hydrocarbons with different volatilities. When it is processed in a distillation column, which has limitations regarding the number of stages and the reflux rate, the non-sharp fractionation or overlaps between adjacent distillates, as seen in Figure 1.9, must be considered to properly formulate cutpoint optimization methods (LI *et al.*, 2005). According to Fu (2015), this overlap typically occurs with properties such as sulfur content, specific density, and viscosity.

Figure 1.9: Crude oil TBP (true boiling point) distillation curve.



Source: Fu (2015).

To model such imperfect fractionation in the estimation of product yields and properties, the main simplification addressed in the literature considers averaged cutpoints to divide the TBP curve into small sections (cuts). These simplified cutpoints are commonly defined as the mid-point of the adjacent TBP overlapping temperatures (TBP cutpoint = 0.5(EP_L + IBP_H)), in which EP_L is the end boiling point of the lighter fraction and the IBP_H is the initial boiling point of the adjacent heavier fraction. A traditional empirical approach to represent the distillation unit is known as delta-based modeling and uses small increments for product deviations in the TBP curve. Common types of delta-based models are the swing-cut methods, which require the estimation of the swing-cut size and the TBP range for each product. This information, combined with the properties of the crude oil in the corresponding TBP range, is used to calculate the distillate properties (FU and MAHALEC, 2015). Due to their simplicity, they are one of the most used models in the distillation unit modeling (GUERRA and LE ROUX, 2011).

The swing-cut model proposed by Zhang *et al.* (2001) considers the properties of the feed in the distillation unit and the operating conditions of the process as variables and optimize the amount produced from each distillate. However, this method considers that the properties of adjacent products are fixed throughout the swing-cut, failing to represent the high nonlinearity of the distillation process. Li *et al.* (2005) use an empirical procedure to calculate the mass transfer rates of each product in the CDU and determine the size of each swing-cut. In addition, the authors use regression models based on the properties of the feed load to consider the variation of properties in each swing-cut. However, due to the possibility of processing more than one type of crude oil simultaneously, the model requires additional procedures to calculate the TBP curve of the crude oil mixture. To deal efficiently with the variation of properties within the swing-cut, Menezes et al. (2013) improve the traditional swing-cut method by separating each swing-cut into two fractions (light and heavy) using 10 °C increments as micro cuts. This method adds property information for both fractions, and additional variables and nonlinear constraints. The properties of each fraction are calculated individually using interpolated quality information regarding their respective amounts. This method improves the prediction of distillation unit outputs and provide more accurate results.

Other methods have also been developed to model the distillation unit. Trierwiler and Tan (2001) used an iterative procedure referred to as the adherent recursion model, in which the cutpoints are sent to a rigorous simulator in order to update the yields and properties of the CDU output fractions. However, the high computational time required for the convergence of the problem, mainly due to the use of the rigorous simulator, limits the application of this method. Li, Hui, and Li (2005) use an empirical procedure to calculate the mass transfer rates of each product in the CDU and determine the size of each swing-cut. In addition, the authors use regression models based on the properties of the feed load to consider the variation of properties in each swing-cut. However, due to the possibility of processing more than one type of oil simultaneously, the model requires additional procedures for calculating the TBP curve of the mixture fed into the CDU. Guerra and Le Roux (2011a, 2011b) apply a modified swing-cut model using volumetric transfer rates in a refinery planning problem. The proposed model uses empirical correlations based on the oil TBP curves, failing to deal efficiently with the variation of properties within the swing-cut. Alattas *et al.* (2011) develop a simplified nonlinear CDU model for production planning based on a fractionation index (FI). They represent the tower as flash units operating in series and the FI for each section are computed using characteristics of the columns such as tower configuration and temperature distribution. Using FI and molar balances among the units, the model predicts the distillation tower operations more accurately than the traditional swing-cut models. Mahalec and Sanchez (2012) propose a hybrid model based on first principles considering mass and energy balances to optimize distillation unit towers. Surrogate correlation using partial least-squares models relates the operating variables to product distillation curves in order to predict the vertical deviation between front and back sections of the curve. By relying partially on a statistical modeling, the method manages to reduce the prediction error of the fractions in the distillation unit (FU and MAHALEC, 2015). Fu, Sanchez and Mahalec (2015) propose a simple hybrid model with few linearities to optimize a distillation unit containing three towers. The model does not assume that the final boiling temperature of a lighter fraction and the initial boiling temperature of a heavier fraction are equidistant from the oil TBP curve; and that the midpoint of a cut lies on the TBP curve. Hence, a smaller error is achieved for the distillation unit predictions. However, the increment used to define the cutpoints (that range from approximately 14 to 67 °C) may not be small enough to provide sufficiently good accuracy.

Many authors propose models to predict more efficiently the outputs of the crude distillation unit. However, using a crude distillation unit model within a complete industrial-sized scheduling optimization is still to be addressed. That would be especially helpful when consider time steps in the order of hours (approximating to an online scheduling cycle) and a complete distillation tower topology (for a more realistic and accurate representation). These features provide a better process accuracy and control towards mitigating plant-model mismatches within a more complex and accurate crude oil refinery modeling and optimization.

1.6 References

AL-QAHTANI, K. Y.; ELKAMEL, A. Planning and integration of refinery and petrochemical operations. John Wiley & Sons, 2011.

ASSIS, L. S.; CAMPONOGARA, E.; MENEZES, B. C.; GROSSMANN, I. E. An MINLP formulation for integrating the operational management of crude oil supply. Computers & Chemical Engineering, v. 123, p. 110-125, 2019.

BROOKS, R. W; VAN WALSEM F. D.; DRURY J. Choosing cutpoints to optimize product yields. Hydrocarbon Processing, v. 78, n. 11, p. 53-58, 1999.

BRUNAUD, B., PEREZ, H. D., AMARAN, S., BURY, S., WASSICK, J., GROSSMANN, I. E. Batch scheduling with quality-based changeovers. Computers & Chemical Engineering, v. 132, p. 106617, 2020.

BUENO, C. Planejamento operacional de refinarias. Dissertação de Mestrado, Universidade Federal de Santa Catarina, 2003.

CAFARO, V. G.; CAFARO, D. C.; MÉNDEZ, C. A.; CERDÁ, J. Optimization model for the detailed scheduling of multi-source pipelines. Computers & Industrial Engineering, 88, 395-409, 2015.

CASTRO, P. M.; GROSSMANN, I. E. Global optimal scheduling of crude oil blending operations with RTN continuous-time and multiparametric disaggregation. Industrial & Engineering Chemistry Research, v. 53, n. 39, p. 15127-15145, 2014.

CHEREMISINOFF, N. P. Handbook of chemical processing equipment. Butterworth-Heinemann, 2000.

DEMBICKI, H. Practical petroleum geochemistry for exploration and production. Elsevier, 2016.

DO, H. Risk management of oil refinery. Master thesis, University of Texas, 2014.

FAHIM, M. A.; AL-SAHHAF, T. A.; ELKILANI, A. Fundamentals of petroleum refining. Elsevier, 2009.

FLOUDAS, C. A.; LIN, X. Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. Computers & Chemical Engineering, v. 28, n. 11, p. 2109-2129, 2004.

FRANZOI, R. E., MENEZES, B. C., KELLY, J. D., GUT, J. A. W. Effective scheduling of complex process-shops using online parameter feedback in crude oil refineries. In Computer Aided Chemical Engineering, v. 44, p. 1279-1284, Elsevier, 2018.

FU, G. Hybrid Model for Optimization of Crude Distillation Units. Master thesis, McMaster University, 2015.

FU, G.; MAHALEC, V. Comparison of Methods for Computing Crude Distillation Product Properties in Production Planning and Scheduling. Industrial & Engineering Chemistry Research, v. 54, n. 45, p. 11371-11382, 2015.

FU, G.; SANCHEZ, Y.; MAHALEC, V. Hybrid model for optimization of crude oil distillation units. AIChE Journal, v. 62, n. 4, p. 1065-1078, 2015.

FURMAN, K. C.; JIA, Z.; IERAPETRITOU, M. G. A robust event-based continuous time formulation for tank transfer scheduling. Industrial & Engineering Chemistry Research, v. 46, n. 26, p. 9126-9136, 2007.

GARY, J. H.; HANDWERK, G. E. Petroleum Refining -Technology and Economics, 3rd Edition, Marcel Dekker, Inc., New York, N.Y., 1994.

GOKHALE, Vikram; HUROWITZ, Scott E.; RIGGS, James B. A Dynamic Model of a Superfractionator: A Test Case for Comparing Distillation Control Techniques. IFAC Proceedings Volumes, v. 28, n. 9, p. 311-316, 1995.

GUERRA, O. J.; LE ROUX, G. A. C. Improvements in petroleum refinery planning: 1. Formulation of process models. Industrial & Engineering Chemistry Research, v. 50, n. 23, p. 13403-13418, 2011.

GUERRA, O. J.; LE ROUX, G. A. C. Improvements in petroleum refinery planning: 2. Case studies. Industrial & Engineering Chemistry Research, v. 50, n. 23, p. 13419-13426, 2011.

GUPTA, D.; MARAVELIAS, C. T.; WASSICK, J. M. From rescheduling to online scheduling. Chemical Engineering Research and Design, v. 116, p. 83-97, 2016a.

GUPTA, D.; MARAVELIAS, C. T. On deterministic online scheduling: major considerations, paradoxes and remedies. Computers & Chemical Engineering, v. 94, p. 312-330, 2016b.

HOCKING, M. BB. Handbook of chemical technology and pollution control. Elsevier, 2016.

HOU, Y.; WU, N.; LI, Z. A genetic algorithm approach to short-term scheduling of crude oil operations in refinery. IEEJ Transactions on Electrical and Electronic Engineering, v. 11, n. 5, p. 593-603, 2016.

JIA, Z.; IERAPETRITOU, M.; KELLY, J. D. Refinery short-term scheduling using continuous time formulation: Crude oil operations. Industrial & Engineering Chemistry Research, v. 42, n. 13, p. 3085-3097, 2003.

JIA, Z.; IERAPETRITOU, M. Efficient short-term scheduling of refinery operations based on a continuous time formulation. Computers & chemical engineering, v. 28, n. 6, p. 1001-1019, 2004.

KARUPPIAH, R.; FURMAN, K. C.; GROSSMANN, I. E. Global optimization for scheduling refinery crude oil operations. Computers & Chemical Engineering, v. 32, n. 11, p. 2745-2766, 2008.

KELLY, J. D. Chronological decomposition heuristic for scheduling: divide and conquer method. AIChE Journal, v. 48, p. 2995-2999, 2002.

KELLY, J.D.; MANN J.L. Crude Oil Blend Scheduling Optimization: an Application with Multimillion dollar benefits - Part 1: Process/plant optimization. Hydrocarbon Processing, 82, 6, 47-53, 2003.

KELLY, J. D.; MANN, J. L. Flowsheet Decomposition Heuristic for Scheduling: a Relax-and-Fix Method. Computers & Chemical Engineering, v. 28, p. 2193-2200, 2004.

KELLY, J. D. The Unit-Operation-Stock Superstructure (UOSS) and the Quantity-Logic-Quality Paradigm (QLQP) for Production Scheduling in The Process Industries. In Multidisciplinary International Scheduling Conference Proceedings: New York, United States, p. 327-333, 2005.

KELLY, J. D.; MENEZES, B. C.; GROSSMANN, I. E. Distillation blending and cutpoint temperature optimization using monotonic interpolation. Industrial & Engineering Chemistry Research, v. 53, n. 39, p. 15146-15156, 2014.

KELLY J. D.; ZYNGIER, D. An improved MILP modeling of sequence-dependent switchovers for discrete-time scheduling problems. Ind Eng Chem Res 46: 4964, 2007.

KELLY, J. D.; ZYNGIER, D. Continuously improve the performance of planning and scheduling models with parameter feedback. In: Proceedings of the foundations of computer-aided process operations (FOCAPO), 2008.

KELLY, J. D.; ZYNGIER, D. Unit-operation nonlinear modeling for planning and scheduling applications. Optimization and Engineering, v. 18, p. 133–154, 2017.

KELLY, J. D.; MENEZES, B. C.; GROSSMANN, I. E.; ENGINEER, F. Feedstock Storage Assignment in Process Industry Quality Problems. In Foundations of Computer Aided Process Operations, Tucson, United States, 2017a.

KELLY, J. D.; MENEZES, B. C.; GROSSMANN, I. E.; ENGINEER, F. Crude oil Blend Scheduling Optimization of an Industrial-Sized Refinery: a Discrete-Time Benchmark. In Foundations of Computer Aided Process Operations, Tucson, United States, 2017b.

LEE, H.; PINTO, J. M.; GROSSMANN, I. E.; PARK, S. Mixed-integer linear programming model for refinery short-term scheduling of crude oil unloading with inventory management. Industrial & Engineering Chemistry Research, v. 35, n. 5, p. 1630-1641, 1996.

LI, J.; LI, W.; KARIMI, I. A.; SRINIVASAN, R. Improving the robustness and efficiency of crude scheduling algorithms. AIChE Journal, v. 53, n. 10, p. 2659-2680, 2007.

LI, J.; MISENER, R.; FLOUDAS, C. A. Continuous-time modeling and global optimization approach for scheduling of crude oil operations. AIChE Journal, v. 58, n. 1, p. 205-226, 2012.

LI, J.; XIAO, X.; FLOUDAS, C. A. Integrated gasoline blending and order delivery operations: Part I. Short-term scheduling and global optimization for single and multi-period operations. AIChE Journal, 2016.

LI, W.; HUI, C.; LI, A. Integrating CDU, FCC and product blending models into refinery planning. Computers & chemical engineering, v. 29, n. 9, p. 2010-2028, 2005.

LI, Z.; IERAPETRITOU, M. Process scheduling under uncertainty: Review and challenges. Computers & Chemical Engineering, v. 32, n. 4, p. 715-727, 2008.

LIMA, R. de O. Pirólise térmica e catalítica de resíduos de vácuo gerados no refino de petróleo. Dissertação de Mestrado. Universidade Federal do Rio Grande do Norte, 2012.

LOTERO, I.; TRESPALACIOS, F.; GROSSMANN, I. E.; PAPAGEORGIOU, D. J.; CHEON, M. S. An MILP-MINLP decomposition method for the global optimization of a source based model of the multiperiod blending problem. Computers & Chemical Engineering, v. 87, p. 13-35, 2016.

MARIANO, J. B. Impactos ambientais do refino de petróleo. Dissertação de Mestrado, Universidade Federal do Rio de Janeiro, 2001.

MARTINS, C. de A. Introdução da concorrência e barreiras à entrada na atividade de refino de petróleo no Brasil. Dissertação de Mestrado em Economia. Rio de Janeiro: Instituto de Economia/UFRJ, 2003.

MÉNDEZ, C. A.; CERDÁ, J.; GROSSMANN, I. E.; HARJUNKOSKI, I.; FAHL, M. State-of-the-art review of optimization methods for short-term scheduling of batch processes. Computers & Chemical Engineering, v. 30, n. 6, p. 913-946, 2006a.

MÉNDEZ, C. A.; GROSSMANN, I. E.; HARJUNKOSKI, I.; KABORÉ, P. A simultaneous optimization approach for off-line blending and scheduling of oil-refinery operations. Computers & chemical engineering, v. 30, n. 4, p. 614-634, 2006b.

MENEZES, B. C.; KELLY, J. D.; GROSSMANN, I. E. Improved swing-cut modeling for planning and scheduling of oil-refinery distillation units. Industrial & Engineering Chemistry Research, v. 52, n. 51, p. 18324-18333, 2013.

MENEZES, B. C.; JOLY, M.; MORO, L. F. L. Crude oil Scheduling Technology: moving from simulation to optimization. In: Europeam Symposium in Computer Aided Process Engineering, 2015, Copenhagen. 12th International Symposium on Process System

Engineering and 25th Europeam Symposium in Computer Aided Process Engineering, 2015a. p. 22-22.

MENEZES, B. C.; KELLY, J. D.; GROSSMANN, I. E. Phenomenological decomposition heuristic for process design synthesis of oil-refinery Units. Computer Aided Chemical Engineering, v. 37, p. 1877-1882, 2015b.

MENEZES, B. C.; KELLY, J. D.; GROSSMANN, I. E.; VAZACOPOULOS, A. Generalized Capital Investment Planning of Oil-Refineries using MILP and Sequence-Dependent Setups. Computers & Chemical Engineering, v. 80, p. 140-154, 2015c.

MOREIRA, F. S. Alternativas tecnológicas para a maximização da produção de olefinas leves a partir de petróleos pesados. Projeto de Conclusão de Curso. Escola de Química, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2006.

MOURET, S., GROSSMANN, I. E., PESTIAUX, P. A Novel Priority-Slot Based Continuous-Time Formulation for Crude oil Scheduling Problems. Industrial & Engineering Chemistry Research, v. 48, n. 18, p. 8515-8528, 2009.

NEDELCHEV, A.; STRATIEV, D.; IVANOV, A.; STOILOV, G. Boiling point distribution of crude oils based on TBP and ASTM D-86 distillation data. Petroleum & Coal, v. 53, n. 4, p. 275-290, 2011.

PAN, M.; LI, X.; QIAN, Y. New approach for scheduling crude oil operations. Chemical Engineering Science, v. 64, n. 5, p. 965-983, 2009.

PASQUINI, C; BUENO, A. F. Characterization of petroleum using near-infrared spectroscopy: Quantitative modeling for the true boiling point curve and specific gravity. Fuel, v. 86, n. 12, p. 1927-1934, 2007.

PATTISON, R. C.; TOURETZKY, C. R.; HARJUNKOSKI, I.; BALDEA, M. Moving horizon closed-loop production scheduling using dynamic process models. AIChE Journal, 2016.

REDDY, P.; KARIMI, I. A.; SRINIVASAN, R. Novel solution approach for optimizing crude oil operations. AIChE Journal, v. 50, n. 6, p. 1177-1197, 2004a.

REDDY, P. C. P.; KARIMI, I. A.; SRINIVASAN, R. A new continuous-time formulation for scheduling crude oil operations. Chemical Engineering Science, v. 59, n. 6, p. 1325-1341, 2004b.

RIAZI, M. R. Characterization and properties of petroleum fractions. ASTM international, 2005.

ROBINSON, P. R. Petroleum processing overview. Practical advances in petroleum processing. Springer New York, p. 1-78, 2006.

SAHARIDIS, G. K. D.; MINOUX, M.; DALLERY, Y. Scheduling of loading and unloading of crude oil in a refinery using event-based discrete time formulation. Computers & Chemical Engineering, v. 33, n. 8, p. 1413-1426, 2009.

SHAH, N. Mathematical programming techniques for crude oil scheduling. Computers & Chemical Engineering, v. 20, p. S1227-S1232, 1996.

SHAH, N. K.; LI, Z.; IERAPETRITOU, M. G. Petroleum refining operations: key issues, advances, and opportunities. Industrial & Engineering Chemistry Research, v. 50, n. 3, p. 1161-1170, 2010.

SHAH, N. K.; IERAPETRITOU, M. G. Lagrangian decomposition approach to scheduling large-scale refinery operations. Computers & Chemical Engineering, v. 79, p. 1-29, 2015.

SILVA, R.T. da. Aplicações da Teoria de Controle em uma Refinaria de Petróleo. Trabalho de Conclusão de Curso, Universidade de São Paulo, 2009.

SILVESTRE, D. S. Inferência da curva de destilação ASTM da destilação atmosférica para controle avançado. Monografia submetida à Universidade Federal de Santa Catarina. Florianópolis, 2005.

SPEIGHT, J. G. The chemistry and technology of petroleum. 4th edition, CRC Press, 2006.

TRIERWILER, D.; TAN, R. Advances in Crude Oil LP Modeling. Hydrocarbon Asia 2001, 11, 52–58.

TRIGGIA, A. A., CORREIA, C. A., FILHO, C. V., XAVIER, J. A. D., MACHADO, J. C. V., THOMAS, J. E., ... & GOUVÊA, P. C. V. M. Fundamentos de engenharia de petróleo. Rio de Janeiro: Interciência/PETROBRAS, 2001.

van ELZAKKER, M. A. H.; ZONDERVAN, E.; FRANSOO; J. C.; de HAAN, A. B. An improved feasibility algorithm for the optimization of crude unloading, blending and charging in an oil refinery. Computer Aided Chemical Engineering, v. 20, p. 1805, 2010.

U.S. Environmental Protection Agency, EPA Office of Compliance Sector Notebook Project: Profile of the Petroleum Refining Industry, EPA/310-R-95-013, 1995.

VENTIN, Fabyana Freire. Controle robusto de uma torre estabilizadora de nafta. Dissertação de Mestrado. Universidade Federal do Rio de Janeiro, 2010.

VIEIRA, G. E.; HERRMANN, J. W.; LIN, E. Rescheduling manufacturing systems: a framework of strategies, policies, and methods. Journal of scheduling, v. 6, n. 1, p. 39-62, 2003.

WENKAI, L.; HUI, C. W.; HUA, B.; TONG, Z. Scheduling crude oil unloading, storage, and processing. Industrial & engineering chemistry research, v. 41, n. 26, p. 6723-6734, 2002.

WINSTON, W. L.; GOLDBERG, J. B. Operations research: applications and algorithms. Belmont: Thomson Brooks/Cole, 2004.

WINTER, A.; BATISTELLA, C. B.; WOLF MACIEL, M. R.; MACIEL FILHO, R.; LOPES, M. S.; MEDINA, L. C. A true boiling point curve through molecular distillation using framol correlation. 8th International Conference on Chemical and Process Engineering. Italy, p. 641-646, 2007.

XU, J.; ZHANG, S.; ZHANG, J.; WANG, S.; XU, Q. Simultaneous scheduling of front-end crude transfer and refinery processing. Computers & Chemical Engineering, 2017.

YAMANISHI, E. Simulação, analise e otimização das colunas atmosférica e debutanizadora da unidade de destilação do refino de petróleo. Dissertação de mestrado, 2007.

YÜZGEÇ, U.; PALAZOGLU, A.; ROMAGNOLI, J. A. Refinery scheduling of crude oil unloading, storage and processing using a model predictive control strategy. Computers & Chemical Engineering, v. 34, n. 10, p. 1671-1686, 2010.

ZAHEDI, G.; SABA, S.; AL-OTAIBI, M.; MOHD-YUSOF, K. Troubleshooting of crude oil desalination plant using fuzzy expert system. Desalination, v. 266, p. 162-170, 2011.

ZHANG, S.; XU, Q. Refinery continuous-time crude scheduling with consideration of long-distance pipeline transportation. Computers & Chemical Engineering, v. 75, p. 74-94, 2015.

ZHANG, S.; WANG, S.; XU, Q. A New Reactive Scheduling Approach for Short-Term Crude Oil Operations under Tank Malfunction. Industrial & Engineering Chemistry Research, v. 54, n. 49, p. 12438-12454, 2015.

ZHANG, J.; ZHU, X. X.; TOWLER, G. P. A level-by-level debottlenecking approach in refinery operation. Industrial & Engineering Chemistry Research, v. 40, n. 6, p. 1528-1540, 2001.

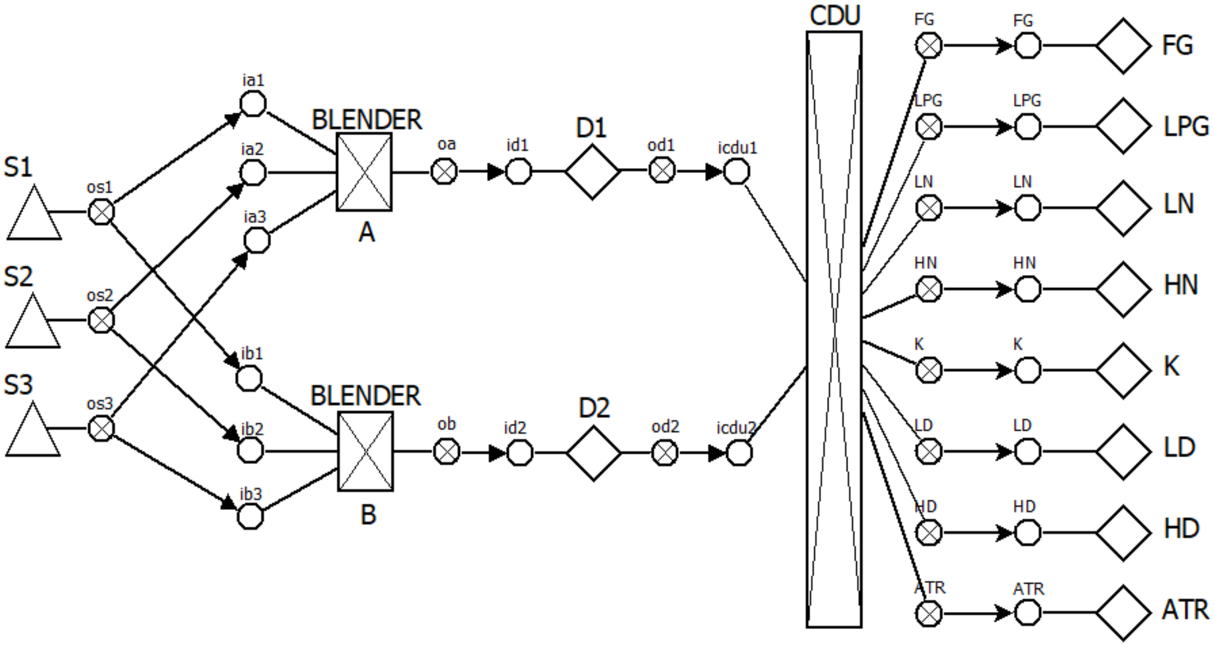
ZYLBERBERG, R. S. Impactos Econômicos e Sociais da Nova Refinaria no Brasil: Uma Análise Comparativa. Monografia de bacharelado, Instituto de Economia, Universidade Federal do Rio de Janeiro, 2006.

2

Refinery Scheduling: Mathematical Formulation

In this chapter we present and discuss a generic mathematical formulation used to model the refinery scheduling examples developed in the present work. For providing a better understating on the mathematical formulation, the simplified network presented in Figure 2.1 is employed, which is based on the Unit-Operation-Port-State-Superstructure (UOPSS) formulation from Kelly et al. (2005). The network is composed by the following objects: a) sources and sinks (◇), tanks, pools, or inventories (△), continuous processes (☒) with distinct modes of operation m; and b) their connectivity involving arrows (→), inlet-ports i (O) and outlet-ports j (⊗). Binary y and continuous x variables are engineered to model unit-operations and their connections, where states can be set to in-ports and out-ports to add more constraints and continuous (or even discrete) variables. In-port and out-port structures are connectors between upstream and downstream unit-operations known as internal-streams (the arrows in Figure 2.1), whereby inlet and outlet flows of material can still be mixed in the in-port-states or split from the out-port-states. In the UOPSS modeling, the arc-flows attached to a port can be connected to any unit-operations in different time windows and ports. A connection between a unit-operation-port-state to another unit-operation-port-state (up or downstream) is managed by binary variables for the unit-operation-port-state to unit-operation-port-state flows or continuous variables in the following classical semi-continuous constraints as $yx^L \leq x \leq yx^U$, with x^L and x^U as the lower and upper bounds of x.

Figure 2.1: Blend scheduling problem network.



Source: Author (2021).

The objective function in Equation 2.1 maximizes the product revenues by subtracting the costs with feedstocks and the performance of the CDU throughputs, defined as the difference of the processed crude oil amounts between the previous and the current time periods.

$$Max\ Z = \sum_{t}\left(\sum_{m\in M_{FU}} price_{m,t}\, x_{m,t} - \sum_{m\in M_{FEED}} price_{m,t}\, x_{m,t} - \sum_{m\in M_{CDU}} weight_m\left(x_{m,t}^{LOD} + x_{m,t}^{UPD}\right)\right) \tag{2.1}$$

Where Z is the objective function to me maximized, and at each time period t, $price_{m,t}$ is the selling price of each product (fuels) $m \in M_P$ or the cost of each feedstock $m \in M_{FEED}$, $x_{m,t}$ are the amounts or flows of the respective feedstocks or products, $weight_m$ is a parameter related to the performance term employed to improve the operating conditions in the distillation unit $m \in M_{CDU}$ considering the lower $x_{m,t}^{LOD}$ and upper $x_{m,t}^{UPD}$ deviation variables of the amount of crude processed in the distillation unit in sequential time periods, which is expected to be as constant as possible. This performance penalty smooths the CDU throughputs $x_{m,t}$ ($m \in M_{CDU}$) by calculating the variations of its adjacent amounts and minimizing the 1-norm or linear deviation of the flow in consecutive time periods. Then,

If $x_{m,t+1} \leq x_{m,t} \implies x_{m,t}^{LOD} = x_{m,t} - x_{m,t+1}$ and $x_{m,t}^{UPD} = 0.$

If $x_{m,t+1} \geq x_{m,t} \implies x_{m,t}^{UPD} = x_{m,t+1} - x_{m,t}$ and $x_{m,t}^{LOD} = 0.$

These deviation variables are set as the same as the bounds of the CDU throughputs, i.e., $0 \leq x_{m,t}^{LOD} \leq \bar{x}_{m,t}^{U}$ and $0 \leq x_{m,t}^{UPD} \leq \bar{x}_{m,t}^{U}$ for $m \in M_{CDU}$. The smoothing relationship for the CDU flow in Equation 2.2 is satisfied if $x_{m,t+1} = x_{m,t}$, in which $x_{m,t}^{LOD} = x_{m,t}^{UPD} = 0$.

$$x_{m,t+1} - x_{m,t} + x_{m,t}^{LOD} - x_{m,t}^{UPD} = 0 \;\; \forall \, m \in M_{CDU}, t \tag{2.2}$$

The UOPSS mathematical formulation is defined in Equations (2.3) to (2.22) and involves objects and their connectivity as shown in Figure 2.1. The sets M_{TK}, M_{BL}, M_{FEED}, and M_P are, respectively, for the unit-operations m of tanks, blenders, feedstocks, and products. The arrows $x_{j,i,t}$ represent the quantity-flows of the connections between distinct unit-operation-port-states, and $x_{m,t}$ $(m \notin M_{TK})$ represents the throughput of unit-operations m.

The quantity-flows of the arrows vary between their bounds ($x_{j,i,t}^{L}$ and $x_{j,i,t}^{U}$) if their binary variables $y_{j,i,t}$ are active, as shown in Equation (2.3). It is similar for the binary variables of unit-operations $y_{m,t}$ $(m \notin M_{TK})$ with respect to their bounds ($x_{m,t}^{L}$ and $x_{m,t}^{U}$) in Equation (2.4), and for the binary variables $y_{m,t}$ $(m \in M_{TK})$ of tank holdups or inventory levels $xh_{m,t}$ between their bounds ($xh_{m,t}^{L}$ and $xh_{m,t}^{U}$) in Equation (2.5).

Equation (2.3) ensures that the flow $x_{j,i,t}$, is between its lower and upper bounds ($\bar{x}_{j,i,t}^{L}$ e $\bar{x}_{j,i,t}^{U}$) if its respective binary variable is active. Similarly, Equation (2.4) imposes bounds ($\bar{x}_{m,t}^{L}$ and $\bar{x}_{m,t}^{U}$) for the flows of unit-operations $x_{m,t}$ $(m \notin M_{TK})$ with respect to their bounds ($x_{m,t}^{L}$ and $x_{m,t}^{U}$) using binaries $y_{m,t}$ $(m \in M_{TK})$, and Equation (2.5) imposes bounds for the tank holdups or inventory levels $xh_{m,t}$ between their bounds ($xh_{m,t}^{L}$ and $xh_{m,t}^{U}$) using the binary variables $y_{m,t}$ $(m \in M_{TK})$.

$$\bar{x}_{j,i,t}^{L} \, y_{j,i,t} \leq x_{j,i,t} \leq \bar{x}_{j,i,t}^{U} \, y_{j,i,t} \;\; \forall \; (j,i), t \tag{2.3}$$

$$\bar{x}_{m,t}^{L} \, y_{m,t} \leq x_{m,t} \leq \bar{x}_{m,t}^{U} \, y_{m,t} \;\; \forall \; m \notin M_{TK}, t \tag{2.4}$$

$$\overline{xh}_{m,t}^{L}\, y_{m,t} \leq xh_{m,t} \leq \overline{xh}_{m,t}^{U}\, y_{m,t} \;\; \forall \;\; m \in M_{TK}, t \tag{2.5}$$

If the binary variable $y_{m,t}$ of the unit-operation m at the time period t is active, Equations (2.6) to (2.9) bound ($x_{m,t}^{L}$ and $x_{m,t}^{U}$) the summation of the quantity-flows (arrows $x_{j,i,t}$) incoming to or outgoing from its port-states. Thus, Equations (2.6) and (2.7) impose bounds for the sum of the $x_{j,i,t}$ quantity-flows outgoing from the out-port-states j of m_{up} (unit upstream of m) and incoming to the in-port-states i of m. Similarly, Equations (2.8) and (2.9) impose bounds for the sum of the $x_{j,i,t}$ quantity-flows outgoing from the out-port-states j of m and incoming to the in-port-states i of m_{do} (unit downstream of m).

$$\sum_{j} x_{j,i,t} \geq \bar{x}_{m,t}^{L}\, y_{m,t} \quad \forall \,(i, m, t) \tag{2.6}$$

$$\sum_{j} x_{j,i,t} \leq \bar{x}_{m,t}^{U}\, y_{m,t} \quad \forall \,(i, m, t) \tag{2.7}$$

$$\sum_{i} x_{j,i,t} \geq \bar{x}_{m,t}^{L}\, y_{m,t} \quad \forall \,(m, j, t) \tag{2.8}$$

$$\sum_{i} x_{j,i,t} \leq \bar{x}_{m,t}^{U}\, y_{m,t} \quad \forall \,(m, j, t) \tag{2.9}$$

Equations (2.10) to (2.13) define bounds on yields, which can be inverse/input ($r_{i,t}^{L}$ and $r_{i,t}^{U}$) and direct/output ($r_{j,t}^{L}$ and $r_{j,t}^{U}$) since the unit-operations m ($m \notin M_{TK}$) allow multiple streams incoming to or outgoing from their independent connected ports. It is worth noting that in industrial operations, tanks typically have only one active inlet flow and only one active outlet flow (i.e., tank do not have more than one inlet flow or more than one outlet flow simultaneously). Equations (2.10) and (2.11) define bounds of inverse yields ($r_{i,t}^{L}$ and $r_{i,t}^{U}$) while (2.12) and (2.13) define bounds of direct yields ($r_{j,t}^{L}$ and $r_{j,t}^{U}$) for the unit-operation throughputs $x_{m,t}$ of unit-operations $m \notin M_{TK}$.

$$\sum_{j} x_{j,i,t} \geq r_{i,t}^{L}\, x_{m,t} \quad \forall \,(i, t), m \notin M_{TK} \tag{2.10}$$

$$\sum_{j} x_{j,i,t} \leq r_{i,t}^{U}\, x_{m,t} \quad \forall \,(i, t), m \notin M_{TK} \tag{2.11}$$

$$\sum_{i} x_{j,i,t} \geq r_{j,t}^{L}\, x_{m,t} \quad \forall\, (j,t), m \notin M_{TK} \tag{2.12}$$

$$\sum_{i} x_{j,i,t} \leq r_{j,t}^{U}\, x_{m,t} \quad \forall\, (j,t), m \notin M_{TK} \tag{2.13}$$

Equations (2.14) represents the material balance required to calculate the inventory or holdup $xh_{m,t}$ of storage and feed tanks $\mathrm{m} \in M_{TK}$. The current holdup amount $xh_{m,t}$ is the remaining amount of material in the past time period plus the amount of material incoming to the tank (upstream connections j_{up}) minus the outgoing material (downstream connections i_{do}).

$$xh_{m,t} = xh_{m,t-1} + \sum_{j_{up}} x_{j_{up},i,t} - \sum_{i_{do}} x_{j,i_{do},t} \;\; \forall\, (i,j,t), \mathrm{m} \in M_{TK} \tag{2.14}$$

The material balances to impose no accumulation of material in continuous-processes are defined in Equation (2.15) for the unit-operation $m \notin M_{TK}$.

$$\sum_{j_{up}} x_{j_{up},i,t} = \sum_{i_{do}} x_{j,i_{do},t} \;\; \forall\, (i,j,t), \mathrm{m} \notin M_{TK} \tag{2.15}$$

Equation (2.16) defines the non-negative continuous variables $x_{m,t}, x_{j,i,t}$, and $xh_{m,t}$, and Equation (2.17) defines the binary variables $y_{j,i,t}, y_{m,t}$.

$$x_{m,t}, x_{j,i,t}, xh_{m,t} \geq 0 \tag{2.16}$$

$$y_{j,i,t}, y_{m,t} = \{0,1\} \tag{2.17}$$

2.1 Logistics Problem: MILP Refinery Scheduling

The logistics problem includes Equations (2.1) to (2.17) previously shown in the UOPSS flowsheet formulation, in addition to Equations (2.18) to (2.35), which involve: a) constraints of structural transitions and selection of operating 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 or uptime and zero downtime of units; and e) fill-draw delay, fill-to-full, and draw-to-empty operations for tanks.

The structural transition constraint Equation (2.18) coordinates setups of connected unit-operations between the out-port-state j_{up} of m_{up} and the in-port-state i of m. If the binary variables of the unit-operations m_{up} and m are active (i.e., if the units are in operation), then the binary variable $y_{j,i,t}$ related to the stream connecting them must be active (i.e., there might be a flow between them). However, if at least one of the units is not operating (binary not active and hence, equal to zero), there cannot be a material flow connecting these units. This logic valid cut, forming a group of 4 objects (m_{up}, j_{up}, i, m), reduces the search in branch-and-bound methods.

$$y_{m_{up},t} + y_{m,t} \geq 2y_{j_{up},i,t} \ \forall \left(m_{up}, j_{up}, i, m, t\right) \tag{2.18}$$

Equation (2.19) ensures that at most one operational mode, procedure, or task $y_{m,t}$ is simultaneously allowed for each physical unit m at each time period t.

$$\sum_{m} y_{m,t} \leq 1 \ \forall \, t \tag{2.19}$$

The temporal transition constraints represented by Equations (2.20) to (2.22) 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 unit-operation. These three variables are relaxed in the interval [0,1] instead of being considered as logic variables, but Equation (2.22) ensures their integrality.

$$y_{m,t} - y_{m,t-1} - zsu_{m,t} + zsd_{m,t} = 0 \ \forall \, m \in M_{BL}, t \tag{2.20}$$

$$y_{m,t} + y_{m,t-1} - zsu_{m,t} - zsd_{m,t} \ - 2zsw_{m,t} = 0 \quad \forall \, m \in M_{BL}, t \tag{2.21}$$

$$zsu_{m,t} + zsd_{m,t} + zsw_{m,t} \leq 1 \ \forall \, m \in M_{BL}, t \tag{2.22}$$

Equations (2.23) and (2.24) are multi-use procedure constraints, in which downstream in-port-states i_{do} connected to the out-port-states ($j \in J_{USE}$) are limited by lower and upper bounds $USE^{L}_{j,t}$ and $USE^{U}_{j,t}$. This is helpful to avoid simultaneous drawing operations from blenders $m \in M_{BL}$ to their downstream tanks. Similarly, (2.25) and (2.26) impose bounds $USE^{L}_{i,t}$ and $USE^{U}_{i,t}$ for the upstream out-port-states j_{up} connected to the in-port-states ($i \in I_{USE}$). That is helpful to control the maximum number of simultaneous filling operations to downstream units such as CDUs ($m \in M_{CDU}$).

$$\sum_{i_{do}} y_{j,i_{do},t} \geq USE_{j,t}^{L}\, y_{m,t} \ \ \forall\, j \in J_{USE}, \forall\, m \in M_{BL}, t \tag{2.23}$$

$$\sum_{i_{do}} y_{j,i_{do},t} \leq USE_{j,t}^{U}\, y_{m,t} \ \forall\, j \in J_{USE}, \forall\, m \in M_{BL}, t \tag{2.24}$$

$$\sum_{j_{up}} y_{j_{up},i,t} \geq USE_{i,t}^{L}\, y_{m,t} \ \ \forall\, i \in I_{USE}, \forall \in M_{CDU}\, , t \tag{2.25}$$

$$\sum_{j_{up}} y_{j_{up},i,t} \leq USE_{i,t}^{U}\, y_{m,t} \forall\, i \in I_{USE}, \forall \in M_{CDU}\, , t \tag{2.26}$$

Equations (2.27) and (2.28) model the uptime or run-length constraints in which UPT^{L} and UPT^{U} are the respective lower and upper bounds of time, t_{end} is the time horizon length, and Δt is the time step. Equation (2.29) models the unit-operation uptime temporal aggregation cut in which n_p is the number of periods. Additional details on Equations (2.27) to (2.29) can be found in Kelly and Zyngier (2007) and Zyngier and Kelly (2009). Equation (2.30) is the zero-downtime constraint for the CDU to select at least one mode of operation m to be continuously operating.

$$\sum_{tt=1}^{UPT^{L}-1} zsu_{m,t-tt|t-tt\geq NTP} \leq y_{m,t} \ \forall\, \mathrm{m}, t > 1 \tag{2.27}$$

$$\sum_{tt=t}^{\frac{UPT^{U}}{\Delta t}} y_{m,t} \leq \frac{UPT^{U}}{\Delta t} \ \forall\, m, (t < t_{end} - UPT^{U}) \tag{2.28}$$

$$\Delta t \sum_{t} zsu_{m,t} \leq n_p \ \forall\, m \in M_{BL} \tag{2.29}$$

$$\sum_{m} y_{m,t} \geq 1 \quad \forall\, (\mathrm{m}, t) \tag{2.30}$$

Equations (2.31) and (2.32) manage the minimum (ΔD_{min}) and maximum (ΔD_{max}) time between the latest filling and the following drawing operations, referred to as the fill-draw delay, for the upstream j_{up} and downstream i_{do} connections of the tank $\mathrm{m}{\in}M_{TK}$. Additional details can be found in Zyngier and Kelly (2009).

$$y_{m,j_{up},i,t} + y_{m,j,i_{do},t+tt} \leq 1 \;\; \forall \left(j_{up}, i, j, i_{do}\right), \mathrm{m}{\in}M_{TK}, tt = 0..\Delta D_{min},$$
$$t = 1..t, t + tt < t_{end} \qquad (2.31)$$

$$y_{m,j_{up},i,t-1} - y_{m,j_{up},i,t} - \sum_{tt=1}^{\Delta D_{max}} y_{m,j,i_{do},t-1} \leq 0 \; \forall \left(j_{up}, i, j, i_{do}\right), \mathrm{m}{\in}M_{TK},$$
$$t = 1..t - \Delta D_{max}, t + tt < t_{end} \qquad (2.32)$$

Equations (2.33) to (2.36) represent the remaining constrains involving tanks, namely, the fill-to-full and draw-to-empty operations. They add the logic variable $yd_{m,t}$, related to the filling and drawing operations, which is active (equal to one) for drawing operations and not active (equal to zero) for filling operations, avoiding the use of two distinct logic variables. The coefficients $xh_{m,t}^{FULL}$ and $xh_{m,t}^{EMPTY}$ are, respectively, the fill-to-full and draw-to-empty inventories to ensure the tank to be filled or drawn at least to their respective values. Hence, the filling operation of a tank must be carried out until the tank reaches a minimum inventory $\overline{xh}_{m,t}^{FULL}$, and the drawing operations must be carried out until the tank reaches a maximum inventory $\overline{xh}_{m,t}^{EMPTY}$. That is helpful to reduce 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).

$$y_{j_{up},i,t} + yd_{m,t} \leq 1 \;\; \forall \left(j_{up}, i, m\right) \text{ for } m \in M_{TK}, t \qquad (2.33)$$

$$y_{j,i_{do},t} \leq yd_{m,t} \;\; \forall \left(m, j, i_{do}\right) \text{ for } m \in M_{TK}, t \qquad (2.34)$$

$$xh_{m,t} - \overline{xh}_{m,t}^{U}\left(yd_{m,t} - yd_{m,t-1}\right) + \left(\overline{xh}_{m,t}^{U} - xh_{m,t}^{FULL}\right) \geq 0 \;\; \forall\, m \in M_{TK}, t \qquad (2.35)$$

$$xh_{m,t} + \overline{xh}_{m,t}^{U}\left(yd_{m,t-1} - yd_{m,t}\right) - \left(\overline{xh}_{m,t}^{U} + xh_{m,t}^{EMPTY}\right) \leq 0 \;\; \forall\, m \in M_{TK}, t \qquad (2.36)$$

2.2 Quality Problem: NLP Refinery Scheduling

The quality problem includes Equations (2.1) to (2.17) from the UOPSS flowsheet formulation (all binary variables are fixed), Equations (2.37) to (2.42) for the blending or pooling constraints, and Equations (2.43) to (2.45) for the transformations from crude oil components c to amounts and properties of product distillates in fractionators (see Kelly and Zyngier (2017) for additional details on the fractionation equations with renderings).

Considering p as property (component concentration, specific gravity, sulfur content, etc.) in which v and w are the volume- and weight-based properties, respectively, Equation (2.37) defines the volume-based properties ($p \in P_v$) as components (crude oil compositions) and density (specific gravity) in the in-port-states i (or mixer) for $i \notin I_{TK}$. Similarly, Equation (2.38) defines the mass-based properties ($p \in P_w$) such as sulfur concentration, whereby $v_{j_{up},p=sg,t}$ is the density ($p = sg$) of the upstream flow from the out-port-states j_{up} incoming to the in-port-state i. For the sake of simplicity, the subsets in the summations involving the in-port-states i in Equations (2.37) and (2.38) are omitted, since this is valid for all $i \notin I_{TK}$, by which j_{up} represents upstream out-port-states of the connected unit-operations m_{up}.

$$v_{i,p,t} \sum_{j_{up}} x_{j_{up},i,t} = \sum_{j_{up}} v_{j_{up},p,t} x_{j_{up},i,t} \quad \forall\ (i,t) \notin I_{TK}, p \in P_v \tag{2.37}$$

$$w_{i,p,t} \sum_{j_{up}} v_{j_{up},p=sg,t}\ x_{j_{up},i,t} = \sum_{j_{up}} w_{j_{up},p,t}\ v_{j_{up},p=sg,t}\ x_{j_{up},i,t} \quad \forall\ (i,t) \notin I_{TK}, p \in P_w \tag{2.38}$$

Equations (2.39) and (2.40) represent the quality balances for volume- and mass-based properties in tanks. The subsets in the summations involving the in-port-states i and out-port-states j in Equations (2.39) and (2.40) are also omitted, since this is valid for all $(i,m,j) \in M_{TK}$, by which j_{up} represents the upstream out-port-states of the connected unit-operations m_{up} arriving in the sole in-port-state $i \in I_{TK}$, and i_{do} represents the downstream in-port-states of the connected unit-operations m_{do} outgoing from the sole out-port-state $j \in J_{TK}$. The quality variable for the out-port-states of a tank unit-operation ($m \in M_{TK}$) is the quality of the blend within the tank, as defined by Equations (2.41) and (2.42).

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

$$\forall\; (i,m,j,t) \in M_{TK}, p \in P_v \qquad (2.39)$$

$$w_{m,p,t}\; v_{m,p=sg,t}\; xh_{m,t}$$

$$= w_{m,p,t-1}\; v_{m,p=sg,t-1}\; xh_{m,t-1} + \sum_{j_{up}} w_{j_{up},p,t}\; v_{j_{up},p=sg,t}\; x_{j_{up},i,t}$$

$$- w_{m,p,t}\; v_{m,p=sg,t} \sum_{i_{do}} x_{j,i_{do},t} \;\; \forall\; (i,m,j) \in M_{TK}, p \in P_w, t \qquad (2.40)$$

$$v_{j,p,t} \;= v_{m,p,t} \;\; \forall\; (m,j) \in M_{TK}, p \in P_v, t \qquad (2.41)$$

$$w_{j,p,t} \;= w_{m,p,t} \;\; \forall\; (m,j) \in M_{TK}, p \in P_w, t \qquad (2.42)$$

For distillation unit operations, Equation (2.43) converts the CDU throughputs $\left(\sum_{j_{up}} x_{j_{up},i_{CDU},t}\right)$ in the i_{CDU} inlet into amounts or yields of distillates $\left(\sum_{i_{do}} x_{j,i_{do},t}\right)$ outgoing from the CDU out-port-states $j \in J_{DIST}$ to the connected downstream in-port-states i_{do}. Equations (2.44) and (2.45) calculate, respectively, the volume- and weight-based properties for J_{DIST} considering each crude oil component c in the crude oil assay with respect to their defined cuts (defined as cut). The renderings for yields and properties are $re^{yie}_{j,c,cut}$ and $re^{p}_{j,c,cut}$ and the upper script sg stands for specific gravity. The subsets in the summations are omitted for the sake of simplicity.

$$\sum_{i_{do}} x_{j,i_{do},t} = \sum_{j_{up}} x_{j_{up},i_{CDU},t} \sum_{c} \sum_{cut} v_{i_{CDU},p=c,t} re^{yie}_{j,c,cut} \;\; \forall\; j \in J_{DIST}, t \qquad (2.43)$$

$$v_{j,p,t} \sum_{i_{do}} x_{j,i_{do},t} = \sum_{j_{up}} x_{j_{up},i_{CDU},t} \sum_{c} \sum_{cut} v_{i_{CDU},p=c,t} re^{yie}_{j,c,cut} re^{p}_{j,c,cut} \;\; \forall\; j \in J_{DIST}, t \qquad (2.44)$$

$$v_{j,p=sg,t} w_{j,p,t} \sum_{i_{do}} x_{j,i_{do},t} = \sum_{j_{up}} x_{j_{up},i_{CDU},t} \sum_{c} \sum_{cut} v_{i_{CDU},p=c,t} re^{yie}_{j,c,cut}\; re^{p=sg}_{j,c,cut} re^{p}_{j,c,cut} \;\; \forall j$$

$$\in J_{DIST}, t \qquad (2.45)$$

2.3 Linear Approximation of Blending Equations: MILP Factor Blending

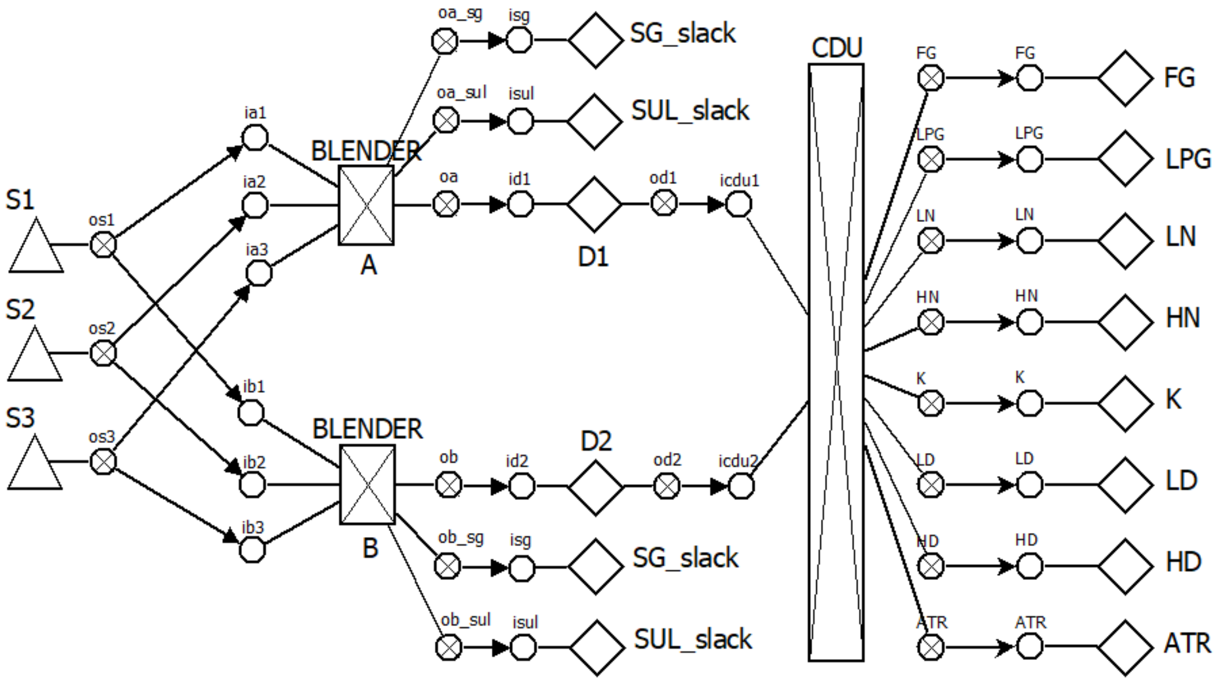
To improve the accuracy of the MILP formulation and optimization, we utilize an approximation strategy in which non-convex NLP blending constraints are formulated as LP quantity-quality balances of streams using factors for qualities (KELLY, MENEZES, and GROSSMANN, 2018). The extended quality amount of the property p is considered in the constraint 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-flows $x_{j^F,p,t}$ outgoing from the slack or surplus out-port-states j^F $(\mathrm{oa\_sg}, \mathrm{oa\_sul})$ in Figure 2.2 are considered in the proposed factor-flow balance in Equation (2.46) for the specific gravity and sulfur content properties.

$$\sum_{i \in I_{BL}} f_{i,p,t} \sum_{j_{up} \in J_{ST}} x_{j_{up},i,t} = f_{j,p,t} \sum_{i_{do} \in I_{FT}} x_{j,i_{do},t} + x_{j^F,p,t} \quad \forall\, j, j^F, p, t \tag{2.46}$$

In the proposed LP approximation to be included in the MILP formulation, for each property p considered to be calculated in the blender $m \in M_{BL}$, amounts of raw materials $\sum_{j_{up}} x_{j_{up},i,t}$ incoming to multiple in-port-states i with factors for qualities $f_{i,p,t}$, in the left side of Equation (2.46) counterbalance the total amount $\sum_{i_{do}} x_{j,i_{do},t}$ of the blended material factor or property specification $f_{j,p,t}$ added to slacks or surpluses of the factor-flow variables $x_{j^F,p,t}$, in the right side of Equation (2.46). From the summation $\sum_{i_{do}} x_{j,i_{do},t}$ in Equation (2.46), although there are diverse outlet ports for the blended material, operationally there is only one simultaneous outlet stream from the blender. The factor in $j^F$is considered as unitary. Therefore, the value of the slack or surplus factor-flows $x_{j^F,p,t}$ represents the insufficient or exceeded amount of qualities for the LP factor flow of each respective property p.

The factor-flow variable $x_{j^F,p,t}$ closes the balance in Equation (2.46) considering the blended material amounts and the product factor $f_{j,p,t}$. For an upper bound of property specification, a slack or negative value is needed, so that $x_{j^F,p,t} \leq 0$. Similarly, for a lower bound, a positive factor-flow or surplus ($x_{j^F,p,t} \geq 0$) applies. Also, as transformation from property to property index may change the signal of the number, to avoid infeasibilities, the factor-flow is modeled as $x_{j^F,p,t} \leq 0$ and $x_{j^F,p,t} \geq 0$ for property indices.

Figure 2.2: Blend scheduling problem network including the factors approximation.



Source: Author (2021).

2.4 References

KELLY, J. D. The Unit-Operation-Stock Superstructure (UOSS) and the Quantity-Logic-Quality Paradigm (QLQP) for Production Scheduling in The Process Industries. In Multidisciplinary International Scheduling Conference Proceedings: New York, United States, p. 327-333, 2005.

KELLY, J. D.; MENEZES, B. C.; GROSSMANN, I. E. Successive LP approximation for non-convex blending in milp scheduling optimization using factors for qualities in the process industry. Industrial & Engineering Chemistry Research, v. 57, n. 32, p. 11076-11093, 2018.

KELLY, J. D.; ZYNGIER, D. An Improved MILP Modeling of Sequence-Dependent Switchovers for Discrete-Time Scheduling Problems. Industrial & Engineering Chemistry Research, v. 46, p. 4964, 2007.

KELLY, J. D.; ZYNGIER, D. Unit-operation Nonlinear Modeling for Planning and Scheduling Applications. Optimization and Engineering, v. 18, n. 1, p. 133–154, 2017.

ZYNGIER, D., KELLY, J. D. Multi-product inventory logistics modeling in the process industries. In: Wanpracha Chaovalitwongse, Kevin C. Furman, Panos M. Pardalos (Eds.) Optimization and logistics challenges in the enterprise. Springer optimization and its applications, p. 61-95, 2009.

3

Refinery Scheduling: Modeling, Configuration, and Optimization

In this chapter we present a generic overview on the modeling and configuration (Section 3.1), and optimization (Section 3.2) procedures used herein to address the refinery scheduling problems, including the software, tools, and packages employed.

3.1 Refinery Scheduling: Modeling and Configuration

After defining the mathematical formulation (Chapter 2), it has to be properly written and built within a modeling platform or software. The modeling platform used to address the refinery scheduling problems proposed in this work is the Industrial Modeling & Programming Language (IMPL), from Industrial Algorithms Limited, which had been previously applied by Kelly et al. (2017b) to solve a highly complex industrial-scale production scheduling problem.

There are several features in IMPL that provide the support required to handle complex large-scale applications, including the specific formulation for modeling logistics and quality processes using the UOPSS superstructure, the Sequential Linear Programming (SLP) technology, the calculation of derivatives using complex numbers and groups of variables with the same dispersion pattern in the matrix, and the use of Reverse Polish notation.

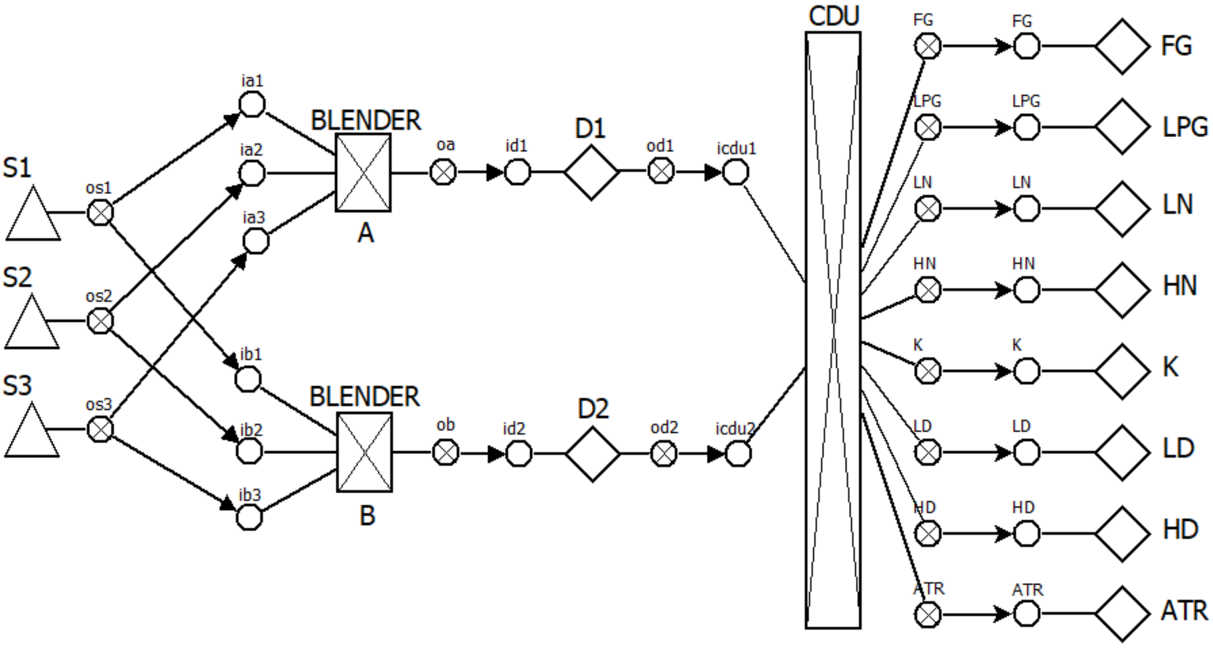
The modeling and configuration procedures in IMPL can be divided in two main steps related to designing and creating the problem superstructure, and coding the formulation within IMPL's specific language.

3.1.1 Superstructure

The first step concerns designing and creating the superstructure to be used in the modeling. The superstructure is built based on the UOPSS formulation, and in a proper format which can be read and understood by IMPL. This is required because the information contained in the superstructure (e.g., units, flows, conditions, etc.) is a fundamental part of the formulation, and that information is later converted into a programming language code which can be read and understood by IMPL as well.

The superstructure is created and designed by using the software DIA Diagram Editor version 0.97.2 (originally developed by Alexander Larsson for the GNOME desktop environment, under general public license). Figure 3.1 presents the blend scheduling problem network as an illustrative example of a superstructure built in the DIA software.

Figure 3.1: Blend scheduling problem network.



Source: Author (2021).

The superstructure presented in Figure 3.1 is composed of five distinct types of components, including triangles (S1, S2, and S3) to represent tanks, diamonds (D1 and D2) for sources of feedstocks or sinks of products, crossed rectangles (BLENDER and CDU) for continuous process units such as blenders and distillation units, and crossed and blank circles to represent outlet flows and inlet flows, respectively, which are a fundamental part of the UOPSS formulation to establish the material flows throughout the process. These circles are respectively referred to as out-ports and in-ports, and must be included after unit-operations that have outlet flows (e.g., sources, tanks, units) and before unit-operations that have inlet flows (e.g., sinks, tanks, units). Therefore, the inlet and outlet ports are used to establish the material balances throughout the network.

The superstructure contains essential information to be used within IMPL to build the mathematical model for a given problem, which is first converted from superstructure-based to code-based information, and is later read by IMPL to build the equation-oriented mathematical formulation.

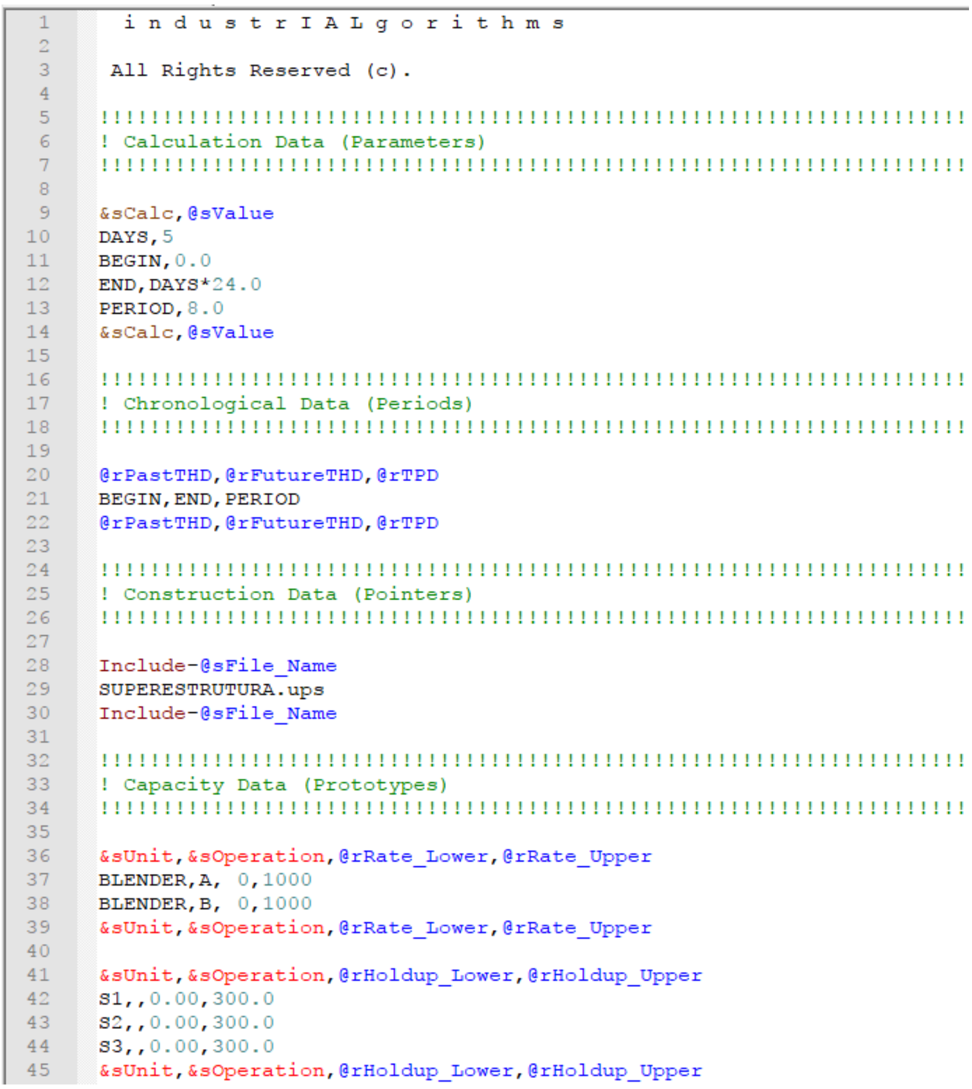
3.1.2 Coding

The information from the superstructure is part of the coding required for the modeling, configuration, and optimization of a given problem in IMPL. The remaining information concerns establishing the process parameters and variables with their respective lower and upper bounds, modeling the flows, inventories, and capacities of pools, tanks, and other unit-operations, defining quality specifications for (intermediate and final) streams, products, and units, including additional information on the crude oil assay and market prices, defining the objective function and other problem-specific constraints, as well as including specific equations of the UOPSS-oriented formulation.

The coding is software-specific, i.e., it must be written in a proper language to be read and understood by IMPL, similarly to any programming languages such as Python and C++, or to other optimization tools/software such as Pyomo or GAMS. The problem to be modeled and optimized can be coded using either industrial modeling language (IML), which directly utilizes the intrinsic configuration of IMPL in a mixture of coding and configuration, or industrial programming language (IPL), in which a third-part language (e.g., Python or Fortran) or software (e.g., Excel or MATLAB) is employed as the main coding platform and interface, and which access IMPL to build the mathematical model and to solve the optimization problem. While the IML language is more focused on the end-user and hence, is more limited in terms of complexity and programming flexibility, the IPL language is more complex and robust, and provides the resources required for a high level programming approach.

The IML language does not provide an interface for the user, so that the text editing software Notepad++ version 7.5.6 (developed by Don Ho, under general public license) is environment used. Figure 3.2 presents a generic piece of code built using the IML language in Notepad++. This simplified piece of code consists of defining some parameters, reading the information from the superstructure (which was previously stored in another file), and utilizing IMPL-based functions to build the required constraints for the problem. In general, there is no need of defining the variables, specifying whether they are logic or continuous, or creating the material balances throughout the flowsheet. This information is automatically and implicitly generated by IMPL based on the configuration of the problem, mostly obtained from the superstructure. Therefore, there is no need to explicitly write the equations and constraints that rule a given problem.

Figure 3.2: Problem coded in the IML language using Notepad++.



```
1    i n d u s t r I A L g o r i t h m s
2
3   All Rights Reserved (c).
4
5  !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
6  ! Calculation Data (Parameters)
7  !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
8
9  &sCalc,@sValue
10 DAYS,5
11 BEGIN,0.0
12 END,DAYS*24.0
13 PERIOD,8.0
14 &sCalc,@sValue
15
16 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
17 ! Chronological Data (Periods)
18 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
19
20 @rPastTHD,@rFutureTHD,@rTPD
21 BEGIN,END,PERIOD
22 @rPastTHD,@rFutureTHD,@rTPD
23
24 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
25 ! Construction Data (Pointers)
26 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
27
28 Include-@sFile_Name
29 SUPERESTRUTURA.ups
30 Include-@sFile_Name
31
32 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
33 ! Capacity Data (Prototypes)
34 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
35
36 &sUnit,&sOperation,@rRate_Lower,@rRate_Upper
37 BLENDER,A,  0,1000
38 BLENDER,B,  0,1000
39 &sUnit,&sOperation,@rRate_Lower,@rRate_Upper
40
41 &sUnit,&sOperation,@rHoldup_Lower,@rHoldup_Upper
42 S1,,0.00,300.0
43 S2,,0.00,300.0
44 S3,,0.00,300.0
45 &sUnit,&sOperation,@rHoldup_Lower,@rHoldup_Upper
```

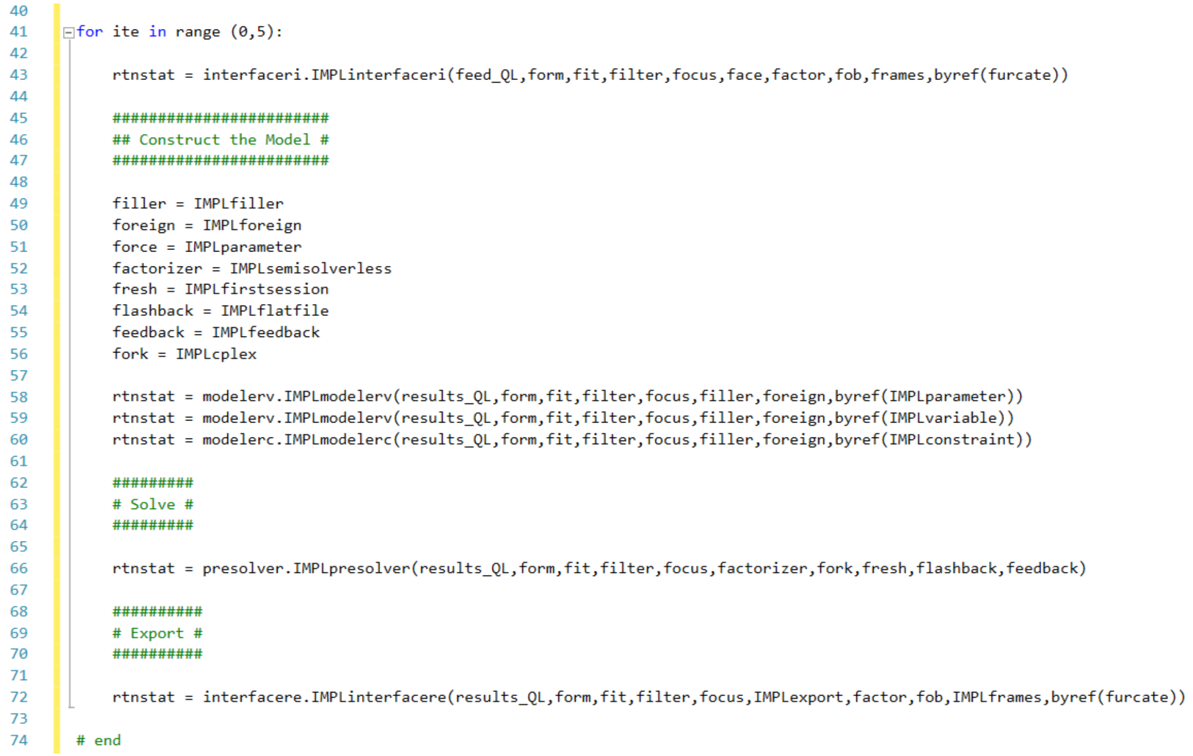
Source: Author (2021).

In the code shown in Figure 3.2, line 9 opens an IMPL function to include parameters, which is closed at line 14. Between them, in lines 10 to 13, the parameters are included by the user. Lines 20 to 22 are used to add information concerning the time horizon, in which the problem is modeled from day 0 to day 5 with time intervals of 8 hours. Lines

28 to 30 read a file containing information related to the superstructure, which was previously and automatically built by IMPL from the superstructure shown in Figure 3.1, which establishes the material flows throughout the process. Lines 36 to 39 impose lower and upper bounds for flows in continuous blenders units and lines 40 to 45 impose lower and upper bounds for inventories of tanks. This is a simplified piece of code built for the problem shown in Figure 3.1 to clarify and exemplify the coding and configuration set up when using the industrial and modeling language within IMPL. The remaining of the problem is similarly configured.

A more sophisticated coding procedure utilizes the so called industrial programming language, in which a third-part language or software is employed as the coding and/or development interface. In this work, the software Visual Studio 2015 (Microsoft, USA) is used for that purpose, which consists of an integrated development environment that supports several programming languages. The main advantages of the IPL language are the possibility of coding designing and automation, performance improvement, development of additional functions and equations, integration with other software and platforms, etc. Hence, several interesting and useful features can be used within the IPL language, including the possibility of developing complex and systematic algorithms and frameworks, performing sequential optimizations of multiple instances of the same problem, which is especially useful to handle nonlinearities and non-convexities of highly nonlinear and non-convex models by multiple optimizations with randomized variable initialization, importing data from and exporting results to other software, etc. Although the coding automation typically requires higher effort, it provides several benefits in terms of coding performance and flexibility, time savings in the long-term, and the possibility of implementing programming strategies and features. Figure 3.3 presents a piece of an optimization code written in Python 3 as programming language and implemented in the Visual Studio 2015 software.

Figure 3.3: IPL code written in Python and implemented in Visual Studio 2015.



```
40
41  for ite in range (0,5):
42
43      rtnstat = interfaceri.IMPLinterfaceri(feed_QL,form,fit,filter,focus,face,factor,fob,frames,byref(furcate))
44
45      ########################
46      ## Construct the Model #
47      ########################
48
49      filler = IMPLfiller
50      foreign = IMPLforeign
51      force = IMPLparameter
52      factorizer = IMPLsemisolverless
53      fresh = IMPLfirstsession
54      flashback = IMPLflatfile
55      feedback = IMPLfeedback
56      fork = IMPLcplex
57
58      rtnstat = modelerv.IMPLmodelerv(results_QL,form,fit,filter,focus,filler,foreign,byref(IMPLparameter))
59      rtnstat = modelerv.IMPLmodelerv(results_QL,form,fit,filter,focus,filler,foreign,byref(IMPLvariable))
60      rtnstat = modelerc.IMPLmodelerc(results_QL,form,fit,filter,focus,filler,foreign,byref(IMPLconstraint))
61
62      #########
63      # Solve #
64      #########
65
66      rtnstat = presolver.IMPLpresolver(results_QL,form,fit,filter,focus,factorizer,fork,fresh,flashback,feedback)
67
68      ##########
69      # Export #
70      ##########
71
72      rtnstat = interfacere.IMPLinterfacere(results_QL,form,fit,filter,focus,IMPLexport,factor,fob,IMPLframes,byref(furcate))
73
74  # end
```

Source: Author (2021).

Using the IPL language provides the programming support required for the development of advanced and systematic algorithms and frameworks, which are essential to handle complex problems and formulations. In Figure 3.3, a strategy involving loops is employed to perform sequential and automatic optimizations, which encompasses lines 41 to 74. More specifically, line 43 reads a secondary file containing information on the problem; lines 49 to 56 specify intern parameters of IMPL, mostly related to the type of problem, solver to be used in the optimization, etc.; lines 58 to 60 call the internal modeling routine of IMPL (i.e., these are the commands for IMPL to build the mathematical model); line 66 calls the optimization routine (i.e., IMPL calls the specified commercial solver, such as CPLEX and GUROBI, to optimize the mathematical model previously built); and line 72 export the results for a new file.

In summary, using the IML language is simpler and easier to implement, which is typically a good option to learn how to use the IMPL software for modeling and optimization purposes, building illustrative and small-scale cases, performing initial tests, etc. On the other hand, the IPL language requires more complex and sophisticated coding, as well as a software or platform for development and interfacing, but provides several advantages for an improved coding and for the development of algorithms and frameworks. Hence, the IPL language is mostly used for the modeling and optimization applications addressed herein. It is also worth mentioning that when optimizing a model written either in the IML or IPL languages, the exact same solutions are expected to be found.

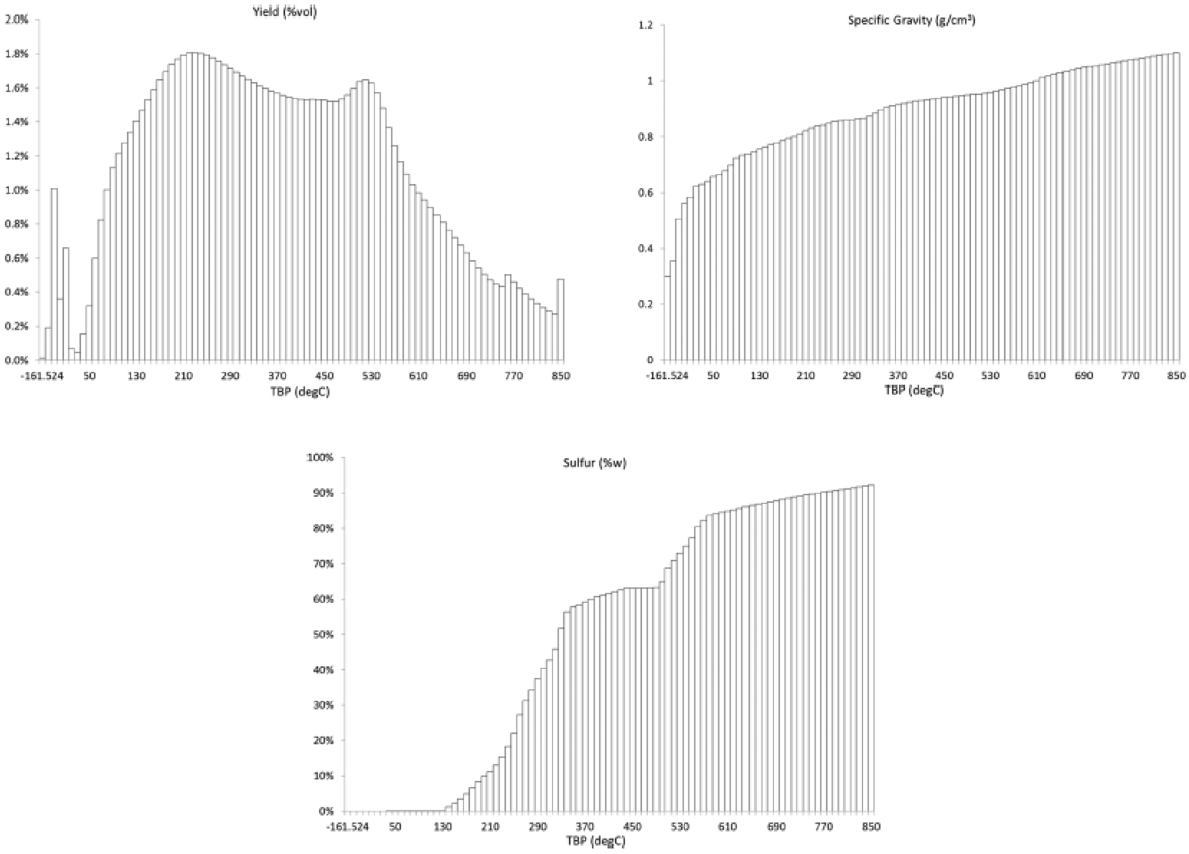
3.1.3 Crude oil assay data

Crude oil refinery operations require knowledge on the crude oil information, especially regarding the quantities (yields) and qualities (properties). This information is highly important for improved industrial operations to provide a better accuracy in the estimations and calculations in planning, scheduling, and control environments, among others.

The characterization of crude oils is typically carried out through rigorous experimental analysis (the TBP curve is the most common) in order to accurately estimate the properties of each component. This is required because when the crude oil is fractionated in the distillation unit, there are distinct qualities for each fraction, so that the crude characterization is fundamental to properly estimate information such as yields and properties of the raw crudes, crude blends, and intermediate and final products (MENEZES et al., 2013). In fact, tracking crude information throughout the network is not trivial and represents a very challenging an important role in the refinery operations.

However, there are economic and technical limitations and difficulties in carrying out this type of analysis, mostly because the relatively large temperature ranges required to collect the samples. Therefore, alternative tools have been increasingly used to obtain or estimate the crude oil assay data. Menezes et al. (2013) used the software for rigorous simulation PetroSIM version 4.1 (KBC Advanced Technologies, England) to estimate the properties of crude oil fractions divided in small temperature ranges of 10 °C, referred to as cutpoints. The data containing the yields and properties of each cut (i.e., fraction of crude oil) is referred to as the crude oil assay. Figure 3.4 presents an example of a crude oil assay with crude fractions divided in 10 °C, and shows how the yield, specific gravity, and sulfur content vary with the TBP temperature.

Figure 3.4: Graphical crude oil assay containing the yields, specific gravity, and sulfur content variables over the temperature.



Source: Adapted from Menezes et al. (2013).

In the modeling and optimization of the refinery problems addressed herein, a data set containing 28 distinct crude oils is used. A simplified version of this data set is presented in Table 3.1. The full crude oil assay data set contains additional information regarding the yields, specific gravity, and sulfur content of each fraction of crude oil divided in micro cuts or ranges. This full crude assay data was generated by the rigorous simulator PetroSIM in Menezes et al. (2013).

Table 3.1: Crude oil assay data.

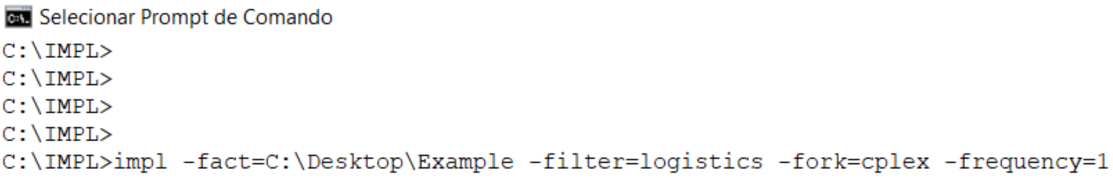
Crude oil index	Crude oil name	° API	Specific gravity (g/cm^3)	Sulfur content (% mass-based)
CO1 (light – medium)	RONCADOR52	27.58	0.8894	0.5025
CO2 (heavy)	MARLIM35	20.79	0.9291	0.6077
CO3 (medium – heavy)	PCONCHAS	22.93	0.9162	0.2310
CO4 (light – medium)	LULA	29.90	0.8766	0.3443
CO5 (extra light)	AGBAMI	45.59	0.7990	0.0487
CO6 (medium – heavy)	BARRACUDA	25.48	0.9014	0.5188
CO7 (light)	BRENT	36.58	0.8418	0.3878
CO8 (heavy)	PBALEIAS	20.57	0.9304	0.4303
CO9 (extra light)	SAHARAN	43.82	0.8070	0.0711
CO10 (heavy)	RONCADOR54	17.35	0.9506	0.6860
CO11 (extra light)	OKONO	40.96	0.8204	0.0571
CO12 (light)	PENNINGTON	33.50	0.8575	0.0909
CO13 (heavy)	MARLIM51	21.36	0.9256	0.6390
CO14 (medium – heavy)	MARLIM40	23.30	0.9141	0.6377
CO15 (medium – heavy)	MARLIMLS	23.90	0.9105	0.5991
CO16 (heavy)	MARLIM56	18.31	0.9445	0.7272
CO17 (medium – heavy)	MARLIM53	22.32	0.919	0.5595
CO18 (heavy)	MARLIM47	20.07	0.9335	0.7668
CO19 (medium – heavy)	MARLIM37	23.52	0.9127	0.6804
CO20 (heavy)	MARLIM33	20.88	0.9286	0.7065
CO21 (heavy)	MARLIM32	20.07	0.9335	0.7668
CO22 (light – medium)	JABUTI	28.53	0.8842	0.4939
CO23 (light – medium)	GOLFINHO	27.22	0.8914	0.1520
CO24 (heavy)	ESPADARTE	20.75	0.9293	0.4781
CO25 (medium – heavy)	CARATINGA	23.47	0.9130	0.5057
CO26 (light – medium)	BAZ	28.86	0.8823	0.2714
CO27 (heavy)	ALBACORA	20.56	0.9305	0.5614
CO28 (extra light)	AKPO	45.32	0.8002	0.0655

Source: Author (2021).

3.2 Refinery Scheduling: Optimization

The modeling and configuration steps include the design and creation of the superstructure, as well as the code used by IMPL to build the mathematical model. After the code is compiled and the model is built, IMPL calls a commercial optimization solver. When working with the IML language, this task can be performed via the prompt of command, as shown in Figure 3.5, in which some flags are required and must be specified by the user to properly call the main IMPL routine. In the case shown in Figure 3.5, there are specified the directory where the files are ("fact" flag), the type of problem, which can be quantity, logistics, or quality ("filter" flag), the solver to be called for optimization purposes ("fork" flag), and the number of sequential optimizations to be carried out ("frequency" flag).

Figure 3.5: Code execution through the prompt of command.

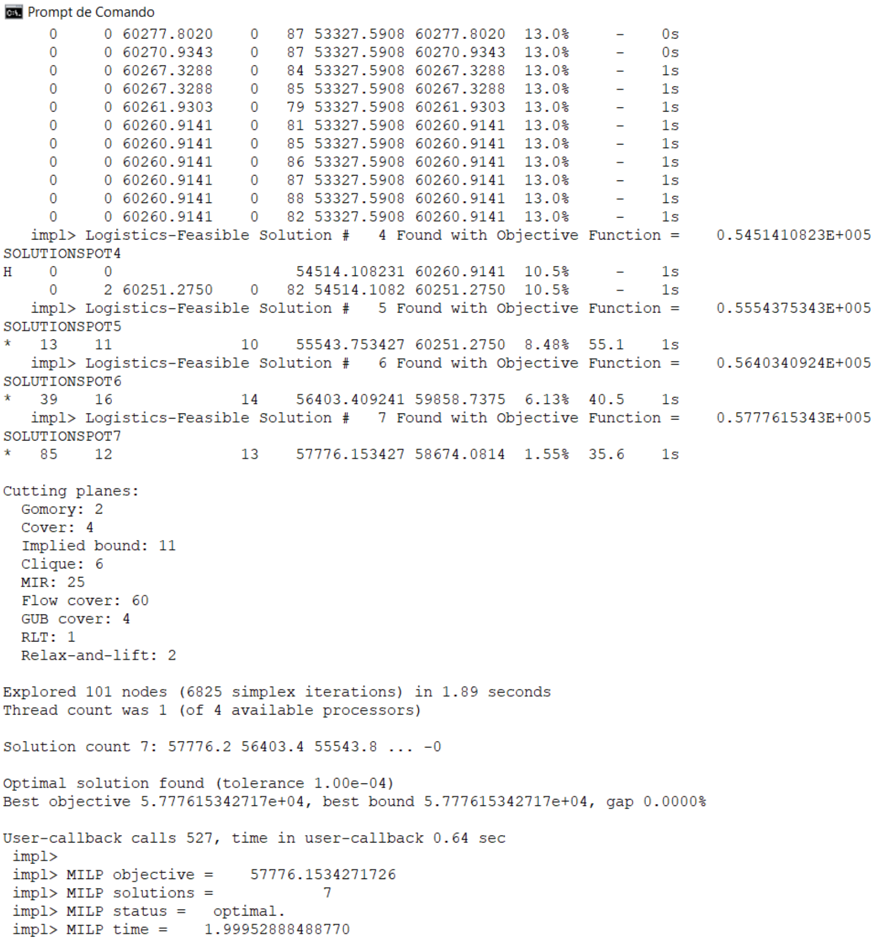


```
C:\IMPL>
C:\IMPL>
C:\IMPL>
C:\IMPL>
C:\IMPL>impl -fact=C:\Desktop\Example -filter=logistics -fork=cplex -frequency=1
```

Source: Author (2021).

The command executed as shown in Figure 3.5 regards a logistics problem (containing quantity and logic information) to be optimized by the commercial solver CPLEX. For MILP problems, some of the most common solvers are CPLEX, GUROBI, and COINMP. For NLP problems, IPOPT can be used, although in this work we employ linearization strategies that allow the use of MILP solvers for nonlinear problems. A summary of the optimization steps and results is written in the prompt of command, as shown in Figure 3.6. Moreover, IMPL creates a set of files containing the solution from the optimization procedure, as well as additional information regarding the mathematical formulation (e.g., equations, variables, parameters), statistics of the problem, etc.

Figure 3.6: Results shown in the prompt of command.

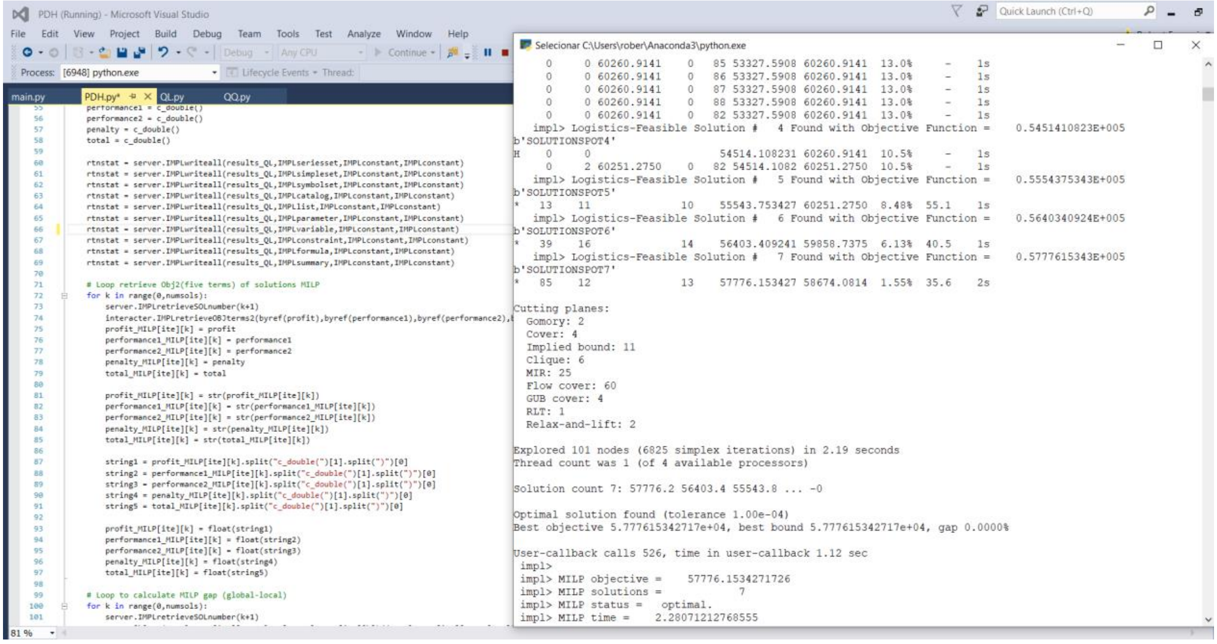


```
     0     0 60277.8020    0   87 53327.5908 60277.8020  13.0%     -    0s
     0     0 60270.9343    0   87 53327.5908 60270.9343  13.0%     -    0s
     0     0 60267.3288    0   84 53327.5908 60267.3288  13.0%     -    1s
     0     0 60267.3288    0   85 53327.5908 60267.3288  13.0%     -    1s
     0     0 60261.9303    0   79 53327.5908 60261.9303  13.0%     -    1s
     0     0 60260.9141    0   81 53327.5908 60260.9141  13.0%     -    1s
     0     0 60260.9141    0   85 53327.5908 60260.9141  13.0%     -    1s
     0     0 60260.9141    0   86 53327.5908 60260.9141  13.0%     -    1s
     0     0 60260.9141    0   87 53327.5908 60260.9141  13.0%     -    1s
     0     0 60260.9141    0   88 53327.5908 60260.9141  13.0%     -    1s
     0     0 60260.9141    0   82 53327.5908 60260.9141  13.0%     -    1s
   impl> Logistics-Feasible Solution #   4 Found with Objective Function =    0.5451410823E+005
SOLUTIONSPOT4
H    0     0                    54514.108231 60260.9141  10.5%     -    1s
     0     2 60251.2750    0   82 54514.1082 60251.2750  10.5%     -    1s
   impl> Logistics-Feasible Solution #   5 Found with Objective Function =    0.5554375343E+005
SOLUTIONSPOT5
*   13    11              10    55543.753427 60251.2750  8.48%  55.1    1s
   impl> Logistics-Feasible Solution #   6 Found with Objective Function =    0.5640340924E+005
SOLUTIONSPOT6
*   39    16              14    56403.409241 59858.7375  6.13%  40.5    1s
   impl> Logistics-Feasible Solution #   7 Found with Objective Function =    0.5777615343E+005
SOLUTIONSPOT7
*   85    12              13    57776.153427 58674.0814  1.55%  35.6    1s

Cutting planes:
  Gomory: 2
  Cover: 4
  Implied bound: 11
  Clique: 6
  MIR: 25
  Flow cover: 60
  GUB cover: 4
  RLT: 1
  Relax-and-lift: 2

Explored 101 nodes (6825 simplex iterations) in 1.89 seconds
Thread count was 1 (of 4 available processors)

Solution count 7: 57776.2 56403.4 55543.8 ... -0

Optimal solution found (tolerance 1.00e-04)
Best objective 5.777615342717e+04, best bound 5.777615342717e+04, gap 0.0000%

User-callback calls 527, time in user-callback 0.64 sec
 impl>
 impl> MILP objective =    57776.1534271726
 impl> MILP solutions =            7
 impl> MILP status =   optimal.
 impl> MILP time =    1.99952888488770
```

Source: Author (2021).

When the IPL language is used, the development platform or software can be used directly to call IMPL in a straightforward fashion. For example, Figure 3.7 shows a piece of code developed in Visual Studio 2015. When the code in executed, there is a specific command calling the main IMPL routine, in which the mathematical model is built and the optimization solver is called in an automatic and systematic fashion, according to the commands within the code.

Figure 3.7: Code execution through Visual Studio 2015.


PDH (Running) - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug
Any CPU
Continue
Process: [6948] python.exe
Lifecycle Events
Thread:
Quick Launch (Ctrl+Q)
main.py
PDH.py*
QL.py
QQ.py
81 %
Selecionar C:\Users\rober\Anaconda3\python.exe

```
    performance1 = c_double()
    performance2 = c_double()
    penalty = c_double()
    total = c_double()

    rtnstat = server.IMPLwriteall(results_QL,IMPLseriesset,IMPLconstant,IMPLconstant)
    rtnstat = server.IMPLwriteall(results_QL,IMPLsimpleset,IMPLconstant,IMPLconstant)
    rtnstat = server.IMPLwriteall(results_QL,IMPLsymbolset,IMPLconstant,IMPLconstant)
    rtnstat = server.IMPLwriteall(results_QL,IMPLcatalog,IMPLconstant,IMPLconstant)
    rtnstat = server.IMPLwriteall(results_QL,IMPLlist,IMPLconstant,IMPLconstant)
    rtnstat = server.IMPLwriteall(results_QL,IMPLparameter,IMPLconstant,IMPLconstant)
    rtnstat = server.IMPLwriteall(results_QL,IMPLvariable,IMPLconstant,IMPLconstant)
    rtnstat = server.IMPLwriteall(results_QL,IMPLconstraint,IMPLconstant,IMPLconstant)
    rtnstat = server.IMPLwriteall(results_QL,IMPLformula,IMPLconstant,IMPLconstant)
    rtnstat = server.IMPLwriteall(results_QL,IMPLsummary,IMPLconstant,IMPLconstant)

    # Loop retrieve Obj2(five terms) of solutions MILP
    for k in range(0,numsols):
        server.IMPLretrieveSOLnumber(k+1)
        interacter.IMPLretrieveOBJterms2(byref(profit),byref(performance1),byref(performance2),
        profit_MILP[ite][k] = profit
        performance1_MILP[ite][k] = performance1
        performance2_MILP[ite][k] = performance2
        penalty_MILP[ite][k] = penalty
        total_MILP[ite][k] = total

        profit_MILP[ite][k] = str(profit_MILP[ite][k])
        performance1_MILP[ite][k] = str(performance1_MILP[ite][k])
        performance2_MILP[ite][k] = str(performance2_MILP[ite][k])
        penalty_MILP[ite][k] = str(penalty_MILP[ite][k])
        total_MILP[ite][k] = str(total_MILP[ite][k])

        string1 = profit_MILP[ite][k].split("c_double(")[1].split(")")[0]
        string2 = performance1_MILP[ite][k].split("c_double(")[1].split(")")[0]
        string3 = performance2_MILP[ite][k].split("c_double(")[1].split(")")[0]
        string4 = penalty_MILP[ite][k].split("c_double(")[1].split(")")[0]
        string5 = total_MILP[ite][k].split("c_double(")[1].split(")")[0]

        profit_MILP[ite][k] = float(string1)
        performance1_MILP[ite][k] = float(string2)
        performance2_MILP[ite][k] = float(string3)
        penalty_MILP[ite][k] = float(string4)
        total_MILP[ite][k] = float(string5)

    # Loop to calculate MILP gap (global-local)
    for k in range(0,numsols):
        server.IMPLretrieveSOLnumber(k+1)
```

```
     0     0 60260.9141    0   85 53327.5908 60260.9141  13.0%     -    1s
     0     0 60260.9141    0   86 53327.5908 60260.9141  13.0%     -    1s
     0     0 60260.9141    0   87 53327.5908 60260.9141  13.0%     -    1s
     0     0 60260.9141    0   88 53327.5908 60260.9141  13.0%     -    1s
     0     0 60260.9141    0   82 53327.5908 60260.9141  13.0%     -    1s
   impl> Logistics-Feasible Solution #   4 Found with Objective Function =    0.5451410823E+005
b'SOLUTIONSPOT4'
H    0     0                    54514.108231 60260.9141  10.5%     -    1s
     0     2 60251.2750    0   82 54514.1082 60251.2750  10.5%     -    1s
   impl> Logistics-Feasible Solution #   5 Found with Objective Function =    0.5554375343E+005
b'SOLUTIONSPOT5'
*   13    11              10    55543.753427 60251.2750  8.48%  55.1    1s
   impl> Logistics-Feasible Solution #   6 Found with Objective Function =    0.5640340924E+005
b'SOLUTIONSPOT6'
*   39    16              14    56403.409241 59858.7375  6.13%  40.5    1s
   impl> Logistics-Feasible Solution #   7 Found with Objective Function =    0.5777615343E+005
b'SOLUTIONSPOT7'
*   85    12              13    57776.153427 58674.0814  1.55%  35.6    2s

Cutting planes:
  Gomory: 2
  Cover: 4
  Implied bound: 11
  Clique: 6
  MIR: 25
  Flow cover: 60
  GUB cover: 4
  RLT: 1
  Relax-and-lift: 2

Explored 101 nodes (6825 simplex iterations) in 2.19 seconds
Thread count was 1 (of 4 available processors)

Solution count 7: 57776.2 56403.4 55543.8 ... -0

Optimal solution found (tolerance 1.00e-04)
Best objective 5.777615342717e+04, best bound 5.777615342717e+04, gap 0.0000%

User-callback calls 526, time in user-callback 1.12 sec
 impl>
 impl> MILP objective =    57776.1534271726
 impl> MILP solutions =           7
 impl> MILP status =   optimal.
 impl> MILP time =    2.28071212768555
```

Source: Author (2021).

Similarly, IMPL creates the files containing the solution from the optimization procedure and additional files regarding the mathematical formulation, statistics of the problem, etc.

3.3 References

KELLY, J. D.; MENEZES, B. C.; GROSSMANN, I. E.; ENGINEER, F. Crude oil Blend Scheduling Optimization of an Industrial-Sized Refinery: a Discrete-Time Benchmark. In Foundations of Computer Aided Process Operations, Tucson, United States, 2017b.

MENEZES, B. C.; KELLY, J. D.; GROSSMANN, I. E. Improved swing-cut modeling for planning and scheduling of oil-refinery distillation units. Industrial & Engineering Chemistry Research, v. 52, n. 51, p. 18324-18333, 2013.

4

Refinery Scheduling: Examples, Discussion, and Results

In this chapter it is presented several examples based on typical crude oil refinery problems, including a discussion on their application, the explanation of the problem, and their respective results and conclusions. The outline of this chapter is as follows.

In section 4.1 (Example 1) it is discussed a blending problem typically used for blending of crude oils prior to their processing in the distillation unit, or blending of intermediate refinery streams to produce specified fuels. An explanation of the phenomenological decomposition heuristic used to break down the MINLP model is presented as well.

Section 4.2 (Example 2) addresses a crude oil distillation problem considering a complex towers in cascade network (instead of considering a single tower) for improved predictions.

Section 4.3 (Example 3) utilizes the complex distillation network from Example 2 in a blend scheduling problem including the crude oil scheduling to prepare the feed for the distillation unit. The phenomenological decomposition is applied to break down the MINLP formulation, and the factors linearization strategy is employed to improve the accuracy of the MILP solution.

Section 4.4 (Examples 4 to 7) discusses the design for online processes and blend scheduling optimization and investigates the impact on distinct blending designs and process designs on crude oil scheduling problems.

Section 4.5 (Example 8) introduces a complete crude oil refinery scheduling network including the entire process-shop in addition to the crude oil scheduling formulation presented in Example 3.

Section 4.6 (Example 9) addresses a case similar to Example 8, and introduces modeling strategies with time reduction purposes. Two heuristic strategies are employed, namely, a rolling horizon and a relax-and-fix approaches.

Section 4.7 (Example 10) discusses the crude oil refinery scheduling optimization problem and introduces an online parameter feedback strategy for handling typical plant-model mismatches that often happen in crude oil refinery operations.

All examples addressed herein are modeled using the Industrial Modeling & Programming Language (IMPL) and are optimized through commercial optimization solvers. The MINLP models are broken down into MILP-NLP sub-models to be solved sequentially. The MILP models are optimized by either CPLEX or GUROBI, while the NLP models are first linearized through a sequential linear programming (SLP) technique to be further optimized through CPLEX or GUROBI. The machine used is an Intel Core i7 (Intel, USA) with 2.90 GHz and 16 GB RAM.

Energy balances are not considered herein. For real applications, the energy balance of the furnaces and the network of parallel crude oil feed integrated to product distillates inside heat exchangers would have to be coordinated to meet the amounts of distillates from the scheduling solution.

The networks in all examples (Figures 4.1 to 4.19) are constructed using the unit-operation-port-state superstructure (UOPSS) formulation (KELLY, 2005) composed by: a) unit-operations m for sources or sinks (◇), continuous-processes (☒) and tanks (△), and b) their connectivity involving arrows (→), in-port-states i (○) and out-port-states j (⊗). Unit-operations and arrows have binary y and continuous x variables.

In the first step of the decomposition approach, the MILP optimal solution is found. Although the optimization solvers find and save intermediate optimal solutions throughout the optimization search, and hence, multiple MILP solutions could be used in the sequence of the decomposition algorithm, only the best solution is chosen in order to limit the computational time spent on each iteration. In the second step of the decomposition approach, multiple optimizations are performed for the NLP problem aiming to find better solutions, which is especially helpful given that refinery scheduling problems are typically highly nonlinear and non-convex, and are often in large size, so that the optimization may converge to poor local optimal solutions. This multiple optimization strategy is performed automatically and systematically by a random generation toolbox within the modeling software IMPL, which generates initial guesses for all variables according to the their lower and upper bounds, and by including a random number, as shown in Equation 4.1.

$$Initial\ Value_{var} = \frac{Up_{var} + \ Lo_{var}}{2} + (Up_{var} - \ Lo_{var})\ (Random - 0.5) \tag{4.1}$$

Where $Initial\ Value_{var}$ is the initial value of the variable var, Lo_{var} and Up_{var} represent the lower and upper bounds of the variable var, and $Random$ is a random number uniformly generated between 0.0 and 1.0.

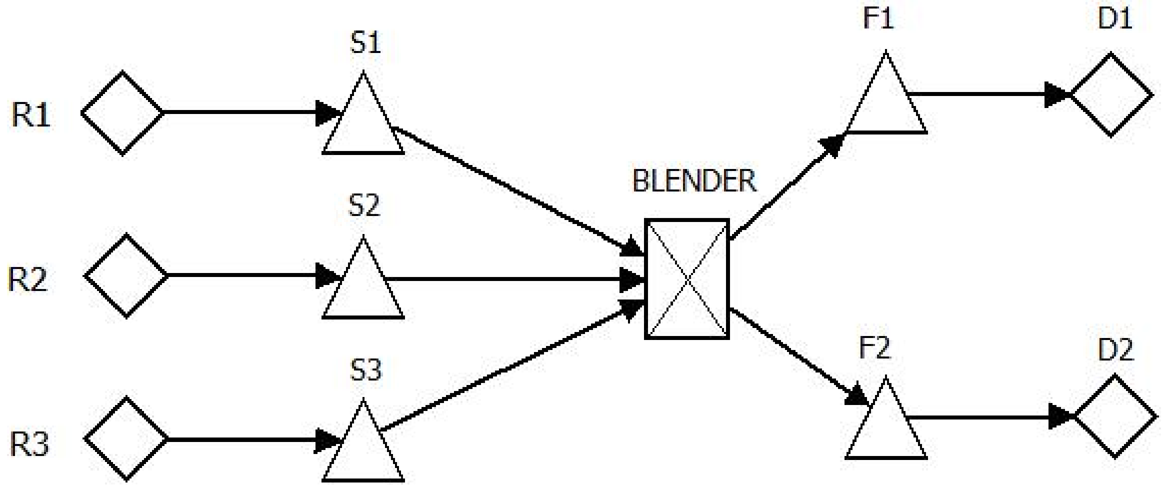
It is also required to establish stopping criteria in which the decomposition framework is assumed to converge. The criterion used in the examples is typically a maximum number of iterations of the algorithm, although several criteria could be employed, such as a maximum decomposition gap MILP-NLP, and non-improvement of the solutions between successive iterations.

4.1 Example 1: Blending operations

The first example consists of a blending problem typically found in the blending of crude oils prior to their processing in the distillation unit, or in the blending of intermediate refinery streams for the production of fuels with quality specifications (e.g., maximum sulfur content, minimum cetane number, etc.). Although it is a small size blending problem, there are several nonlinear and non-convex terms due to the blending equations, as well as binary variables associated with distinct operating modes of the blender unit and from the UOPSS-based formulation (i.e., binary variables for flows and units). This results in a non-convex MINLP formulation, which is broken down in the form of MILP-NLP subproblems to be sequentially solved. An explanation of the phenomenological decomposition heuristic used to break down the MINLP model is presented as well. It is worth noting that this blending example was especially helpful to learn, develop, and improve the methodology aspects addressed in this thesis.

In the problem presented in Figure 4.1, there are three feedstocks R1 to R3, which contain intermediate diesel coming from multiple distinct refinery processing units. Although all of them are diesel streams, their qualities and compositions are slightly different (i.e., distinct hydrocarbon compositions, and distinct properties such as sulfur content and specific gravity). The objective of this example is to maximize the profit in the production of diesel streams with properties within specified ranges, more specifically, maximum values of sulfur content and specific gravity. Therefore, the operations need to ensure that all properties of each product are properly specified. From the feedstock pools, the distinct diesel streams are sent to intermediate tanks S1 to S3, and subsequently feed a blender unit (BLENDER). The diesel blend leaves the mixer and is stored in final tanks F1 and F2 to be later sold and sent via distribution or logistic modes D1 and D2.

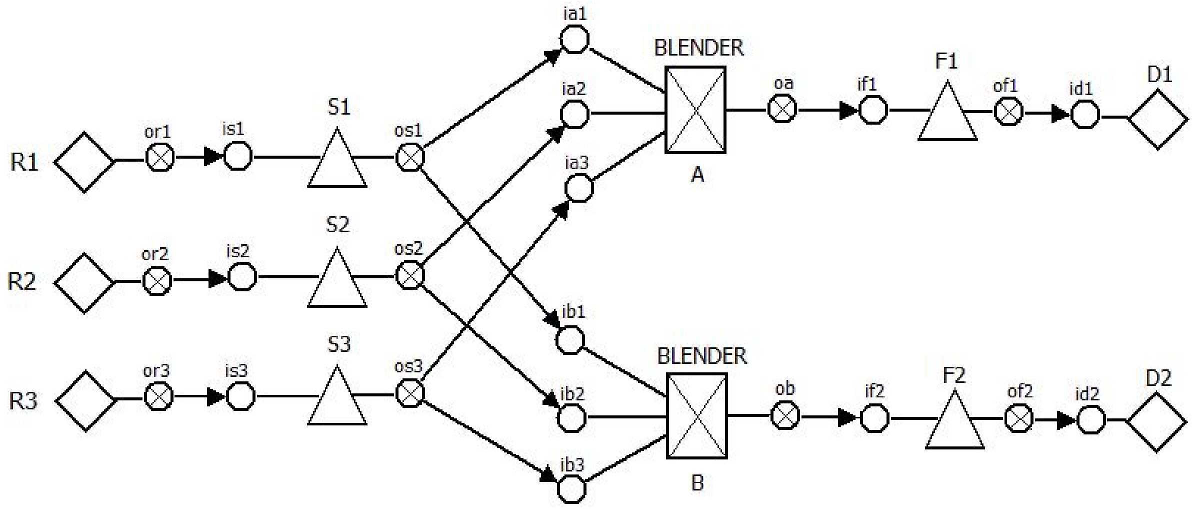
Figure 4.1: Blending operations flowsheet.



Source: Author (2021).

It is also considered that it is possible to produce distinct products in the same blender unit. Thus, the problem is formulated so as to include the operating modes used to produce each product, and the superstructure of the problem is modified to account for such information. Figure 4.2 presents the updated flowsheet, which contains two blender units (both are referred to as BLENDER because they represent the same physical unit). However, there are different operating modes, represented by the capital letters A and B just below each blender. The UOPSS representation is used in Figure 4.2 as well, in which the in-ports and out-ports are included.

Figure 4.2: Blending operations flowsheet using the UOPSS representation.



Source: Author (2021).

There is an increase in the number of variables due to the additional blender unit modeled, including its respective additional flows. The problem is modeled such as BLENDER (A) and BLENDER (B) are different units, used for the production of products D1 and D2, respectively. Then, it is also required to constrain the blenders not to operate simultaneously.

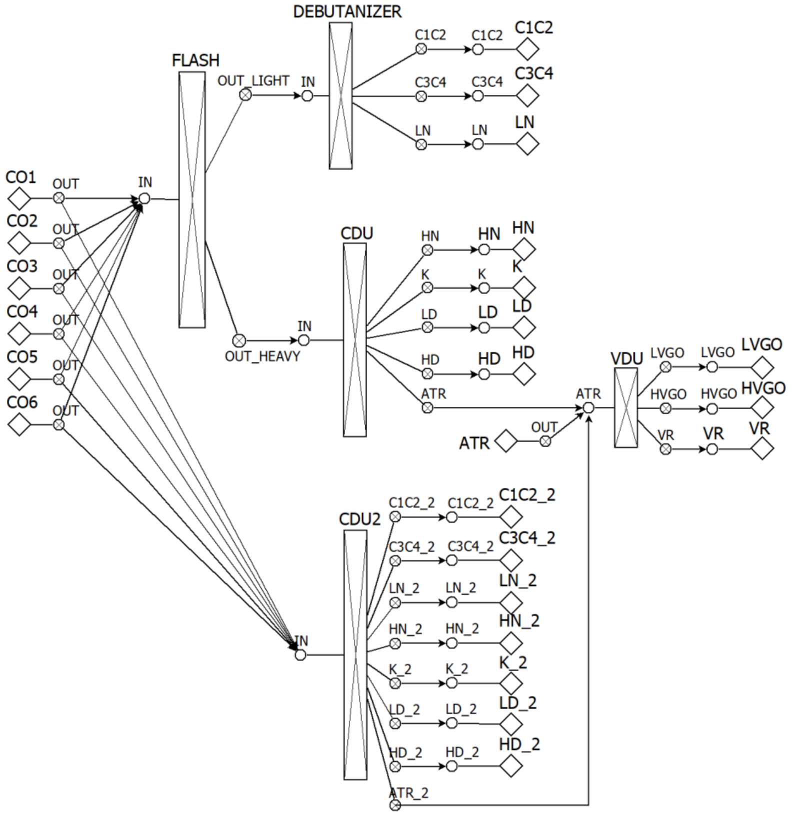
Data on the availability and cost of feedstocks R1 to R3, price of products D1 and D2, capacity of tanks and units, and specification properties of the products (specific gravity and sulfur content) are included in the formulation. There are binary and continuous decision variables, which are associated with the logistic decisions (blender operating modes, active units/flows) and flows throughout the network. The problem is modeled considering the future 15-day time horizon, with 24-hour intervals, in a total of 15 time periods. The problem is formulated as an MINLP and is broken down into two subproblems, an MILP and an NLP, to be sequentially solved. The decomposition strategy is discussed in Section 1.4.9.2.

The MILP solution to be saved and fixed for the NLP optimization includes the binary variables associated with flows (whether they are active or not) and process units (blenders). In the optimal MILP solution after two iterations of the decomposition algorithm, the blender operates at maximum capacity, and at only one operating mode at each time step. The least expensive feedstock is chosen, as quality information is not considered in the formulation. For the best NLP solution (i.e., the NLP model is optimized 10 times using the random generation tool described in Section 1.4.9.2, and only the best solution is chosen), all feedstocks are utilized to ensure the quality constraints for the proper specification of products. The final objective function for profit maximization is around $ 810 for the MILP model and it is around $ 676 for the NLP model. The decomposition approach leads to a decomposition gap between the MILP and NLP solutions, which is indeed expected.

4.2 Example 2: Towers in cascade distillation network

Example 2 considers a complex network of towers in cascade, composed of flash distillation (FLASH), debutanizer or naphtha stabilizer (DEBUTANIZER), two atmospheric distillation (CDU and CDU2), and vacuum distillation (VDU) towers. The main distillates produced are the fuel gas (C1C2), liquefied petroleum gas (C3C4), light naphtha (LN), heavy naphtha (HN), kerosene (K), light diesel (LD), heavy diesel (HD), and atmospheric residue (ATR). An external feed of ATR is also connected directly to the VDU, which produces light vacuum gas oil (LVGO), heavy vacuum gas oil (HVGO), and vacuum residue (VR). The crude oil assay or composition defined in micro-cut, hypo- or pseudo-component distribution is used to calculate the yields and properties of all towers. The production of distillates requires the temperature cutpoints to determine the aggregation of the micro-cuts to the final cuts (product distillates). Details on micro-cuts calculation and distribution can be found in Menezes et al. (2013). Figure 4.3 presents the distillation unit network, which is formulated as a single-period NLP problem.

Figure 4.3: Complex distillation network flowsheet.



Source: Author (2021).

The six sources CO1 to CO6 supply the operations with distinct types of crude oil, which can be either sent to the pre-flash tower or to the distillation unit CDU2. The debutanizer and the two atmospheric distillation columns process their respective feeds to produce fuel gas, LPG, light naphtha, heavy naphtha, kerosene, light diesel, heavy diesel, and atmospheric residue. The atmospheric residue streams are sent to the VDU, while the other streams are sent directly to product storage pools. It is also assumed that an additional amount of atmospheric residue can be imported and processed from the ATR pool.

There are several operational limitations on this example, especially the absence of blender units to prepare the distillation feed. The formulation includes nonlinear terms from the quality balances (e.g., the mass flow that enters the pre-flash tower is the product between the volumetric flow at the entrance of the tower and the specific gravity of the feed, which are both decision variables in the model). The bilinear and trilinear terms (i.e., related to the specific gravity sulfur content equations, respectively) result in a nonlinear programming (NLP) problem.

The decision variables are related to the amounts to be processed of each crude oil and of the atmospheric residue imported. Information on availability of feedstocks, price of products, processing capacity of the units, in addition to the crude oil quality (composition, specific gravity, and sulfur content) are also included in the formulation. Binary variables from the UOPSS formulation (i.e., related to whether the flows and units are active or not) are considered as active (i.e., set to one) and neglected in the model. The objective function maximizes the profit for a single time step for the future 24-hour time horizon. Data on the availability of feedstocks and the production capacity of the units are presented in Table 4.1.

Table 4.1: Data used in the formulation of Example 2.

Unit	Availability ($\boldsymbol{m^3/day}$)	Unit	Processing Capacity ($\boldsymbol{m^3/day}$)
CO1	100	CDU	80
CO2	100	CDU2	80
CO3	100	FLASH	100
CO4	100	DEBUTANIZER	20
CO5	100	VDU	45
CO6	100		
ATR	20		

Source: Author (2021).

The example is modeled using IMPL and optimized using the solver CPLEX 12.8.0 via a sequential linear programming technique for the linearization of nonlinear terms. The optimal solution (i.e., decision variables) of the cascaded distillation towers example is shown in Table 4.2.

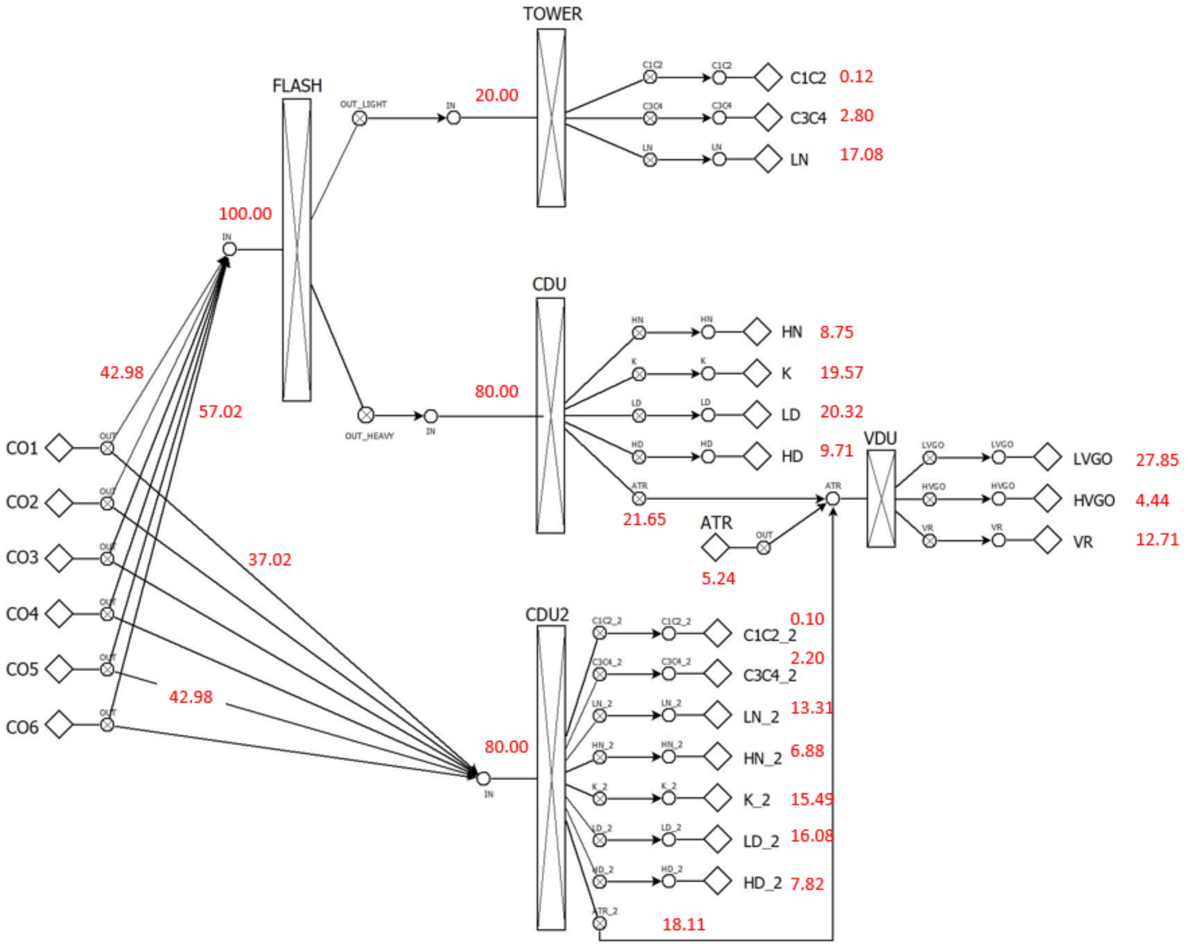
Table 4.2: Optimal solution of the cascaded distillation towers example.

Decision Variable (Flows)	Optimal Value (m^3/day)
CO1 - FLASH	42.98
CO1 – CDU2	37.02
CO2 – FLASH	0.00
CO2 – CDU2	0.00
CO3 – FLASH	0.00
CO3 – CDU2	0.00
CO4 – FLASH	0.00
CO4 – CDU2	0.00
CO5 – FLASH	57.02
CO5 – CDU2	42.98
CO6 – FLASH	0.00
CO6 – CDU2	0.00
ATR - VDU	5.24

Source: Author (2021).

The flowsheet of the cascaded distillation example, including the optimal mass flows (m^3/day), is shown in Figure 4.4.

Figure 4.4: Cascaded distillation network with optimal values.



Source: Author (2021).

The optimal solution indicates that the light crudes (i.e., with better quality), which are crudes CO1 and CO5, are most likely to be used due to the quality specification constraints. The full availability ($100\ m^3/day$) of CO5 is used, whereas only $80\ m^3/day$ of CO1 are used due to the bottleneck in the production capacity of the pre-flash tower ($100\ m^3/day$) and CDU2 ($80\ m^3/day$). The atmospheric residue from the ATR pool is also used to complete the processing capacity of the vacuum distillation column. In addition, it is worth noting that the CDU and DEBUTANIZER columns operate at maximum capacity to maximize the refinery profit.

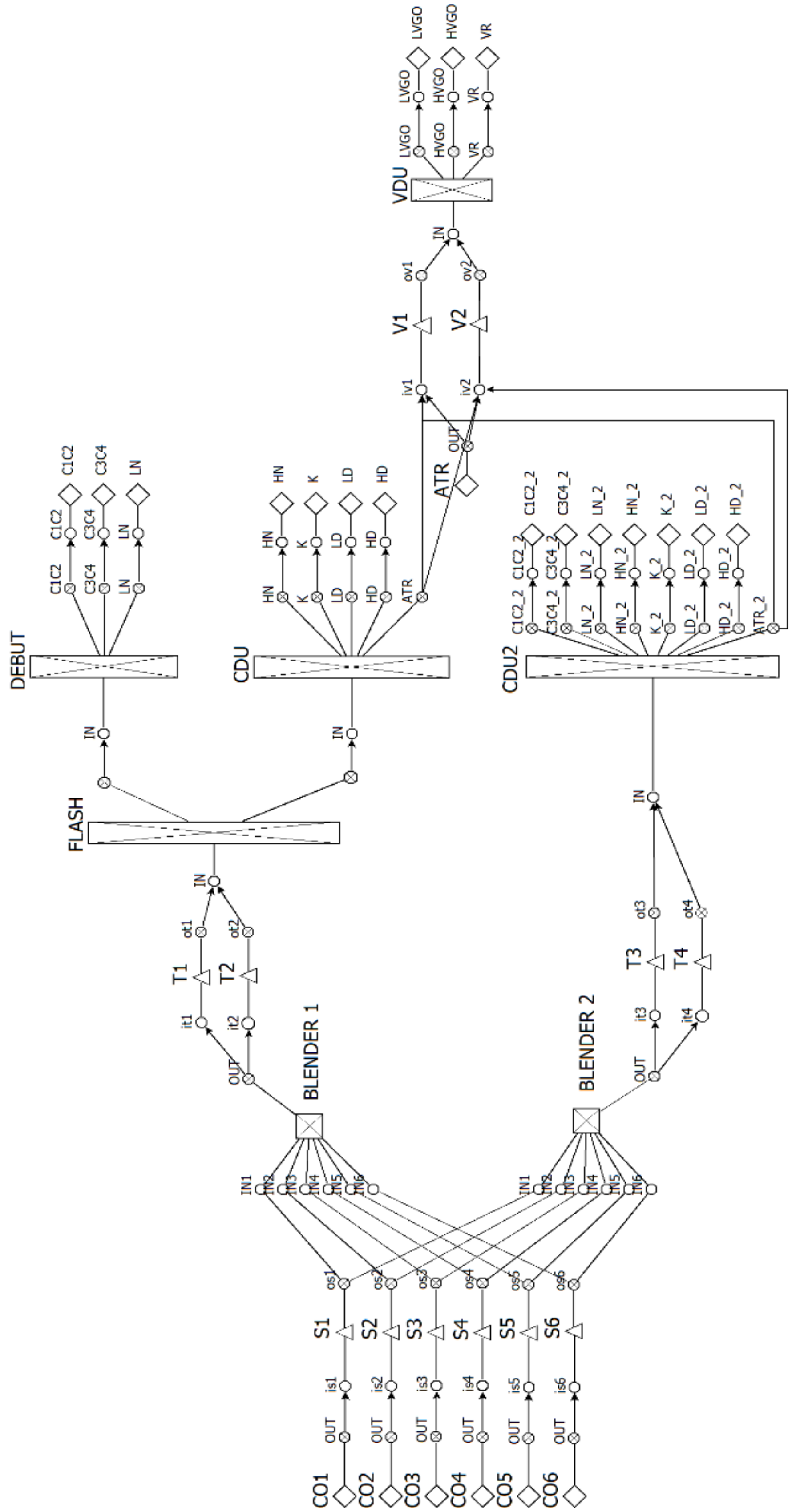
The optimal solution achieved is consistent and expected, as it uses mostly the crude with best quality, utilizes full capacity of the units, and uses atmospheric residue to complete the feed for the vacuum distillation column.

4.3 Example 3: Crude oil blend scheduling operations[1]

After utilizing the phenomenological decomposition heuristic for a blending problem (Example 1), it is applied to the crude oil blend scheduling example, which incorporates the blending of crude oils to produce the distillation feed, and a complex towers in cascade network to represent the distillation unit (from Example 2). The crude oil blend scheduling example is formulated as a discrete-time MINLP problem and the phenomenological decomposition heuristic is used to break it down into two sub-models to be sequentially and iteratively optimized. The crude oil blend scheduling flowsheet is presented in Figure 4.5 and considers the blending and processing of six crude oil supplies (CO1 to CO6), six storage tanks (S1 to S6), in addition to two mixers (BLENDER1, BLENDER2), six feed tanks (T1 to T4, V1 and V2), a flash tower (FLASH), a naphtha debutanizer tower (DEBUT), two atmospheric distillation units (CDU, CDU2), a vacuum distillation column (VDU) and an atmospheric residue import (ATR). The distillation unit is modeled by a detailed series of towers in cascade to provide higher accuracy in the predictions. Intermediate hydrocarbon streams, such as kerosene (K), light naphtha (LN), light diesel (LD), etc., are also included in the formulation.

[1] This section is based on the following manuscript:
FRANZOI, R. E.; MENEZES, B. C.; KELLY, J. D.; GUT, J. A. W. Blend scheduling optimization using factors for qualities in cascaded distillation towers in crude oil refineries. In Blucher Chemical Engineering Proceedings, v. 1, n. 5, p. 1233-1236, 2018.

Figure 4.5: Crude oil blend scheduling flowsheet.



Source: Author (2021).

Example 3 assumes that there are six crude oils available to be processed at the refinery operations, which are supplied by the limited sources CO1 to CO6 and are stored in the storage tanks S1 to S6. These crude oils may be used to prepare the feed for the distillation unit by any of the two distinct blender units. The crude blend formed at BLENDER1 is sent to the feed tanks (T1 and T2) prior to the pre-flash tower. The pre-flash tower separates the feed into light and heavy fractions. The lighter fraction, composed of fuel gas, LPG, and light naphtha, is sent to the naphtha stabilizing tower (also referred to as debutanizer), where the separation and subsequent storage of these three streams take place. The heavier output from the pre-flash tower is sent to an atmospheric distillation unit, which separates the feed into heavy naphtha, kerosene, light diesel, heavy diesel, and atmospheric residue. These streams are stored, except the atmospheric residue, which is sent to the vacuum distillation column.

The blend formed at BLENDER2 is sent to the feed tanks T3 and T4, which feed another atmospheric distillation unit (CDU2). In refinery operations, this blend is typically a mixture of heavy crudes, as it is not operationally or economically interesting to process it in a pre-flash tower prior to the processing in the atmospheric distillation column. The CDU2 separates its feed into fuel gas, LPG, light naphtha, heavy naphtha, kerosene, light diesel, heavy diesel, and atmospheric residue). All streams are stored in pools, except the atmospheric residue, which is sent to the vacuum distillation unit (VDU). The VDU receives the atmospheric residue from the two atmospheric distillation columns. In addition, there is also the possibility of using the atmospheric residue stored in the ATR pool, which is assumed to be imported by the refinery to be used as a feedstock in the operations.

The binary and continuous decision variables are associated with whether and how much is processed of each crude oil and of the imported atmospheric residue, the operation of tanks and units, and the flows throughout the network, for each period of time. Data on the cost and availability of feedstocks, product prices, processing capacity of the units, in addition to the crude oil quality (composition, specific gravity, and sulfur content) are included in the formulation as well.

The mathematical formulation is based on a discrete-time grid with a 5-day time-horizon and time steps of 4 hours, and considers initial inventories of crude oil supply, storage, and charging tanks, integrated to the distillation units arranged in the form of cascaded towers. The qualities of the crude oil tanks are assumed to be constant (at a given time step) since there are constraints to avoid simultaneous or multiple filling to the same tank (as well as to avoid simultaneous drawing from a tank to multiple sinks). The yields of the distillation units are calculated from the crude oil assays. There are binary variables associated with the logistic decisions (e.g., selection of crudes and flows at each time step, operations of units, etc.), as well as nonlinear variables and constraints due to the nonlinear nature of the crude oil properties and the blending operations. Moreover, the blending equations for qualities or properties (e.g., sulfur content) are by nature non-convex, resulting in an non-convex MINLP model. Thus, the phenomenological decomposition heuristic (see Section 1.4.9.2) is used to tackle such complex problem, which is broken down into an MILP and an NLP sub-models. The mathematical MILP model includes only quantity and logic information while the NLP model includes quantity and quality information. The heuristic procedure initially neglects quality information (variables and constraints) from the original MINLP model. This results in an MILP model, which is optimized and its logistics solution (i.e.., binary decisions) is fixed back in the original MINLP. This converts the MINLP into an NLP to be optimized. The NLP solution is also a feasible solution for the original MINLP. An iterative procedure is performed, in which the NLP quality solution (i.e., yields and properties) is retro-fed as initial guesses in the MILP model of the next iteration. Furthermore, because the NLP problem is highly nonlinear by the blending and distillation transformations, multiple NLP optimizations are performed to avoid poor optimal solutions. For that, a randomization tool generates initial points to be used in the different NLP optimizations. The statistics of the mathematical formulation are presented as follows. In the MILP model there are 375 continuous and 672 binary variables, 1700 constraints (220 equality) and 546 degrees of freedom. In the NLP model there are 4442 continuous variables, 5098 constraints (3328 equality) and 30 degrees of freedom. The UOPSS formulation is utilized and it is built-in within the modeling platform IMPL (Industrial Modeling & Programming Language), in which GUROBI 12.7.1 is used as optimization solver in an Intel Core i7 with 2.7 GHz and 16 GB RAM.

When the phenomenological decomposition heuristic is applied, that typically leads to a MILP-NLP decomposition gap due to the limitations of the method (e.g., neglecting quality information). Then, a linear programming approximation of nonlinear constraints is used to improve the accuracy of the MILP modeling. This strategy significantly improves the accuracy of the MILP modeling by including linearized (and approximated) quality information. Hence, it is expected a better and faster convergence of the PDH algorithm in the iterative MILP-NLP procedure by reducing the gap between both solutions. Figure 4.6 presents the crude oil blend scheduling problem including the linear factor reformulation to approximate nonlinear blend streams. Other than the addition of these linear reformulations for the MILP model, the problems in Figure 4.5 and 4.6 are similar. In the MILP model, there are 641 continuous and 728 binary variables, 2948 constraints (316 equality) and 660 degrees of freedom. In the NLP model there are 4470 continuous variables, 5242 constraints (3328 equality) and 63 degrees of freedom.

The decomposition heuristic applied in these two examples allows the optimization of complex crude oil refinery scheduling problems, whereby the cascaded distillation design provides an improved level of detail by considering individual fractionations in each column instead of only one black box distillation tower to account for the whole distillation system, which improves both the prediction and the control of the towers.

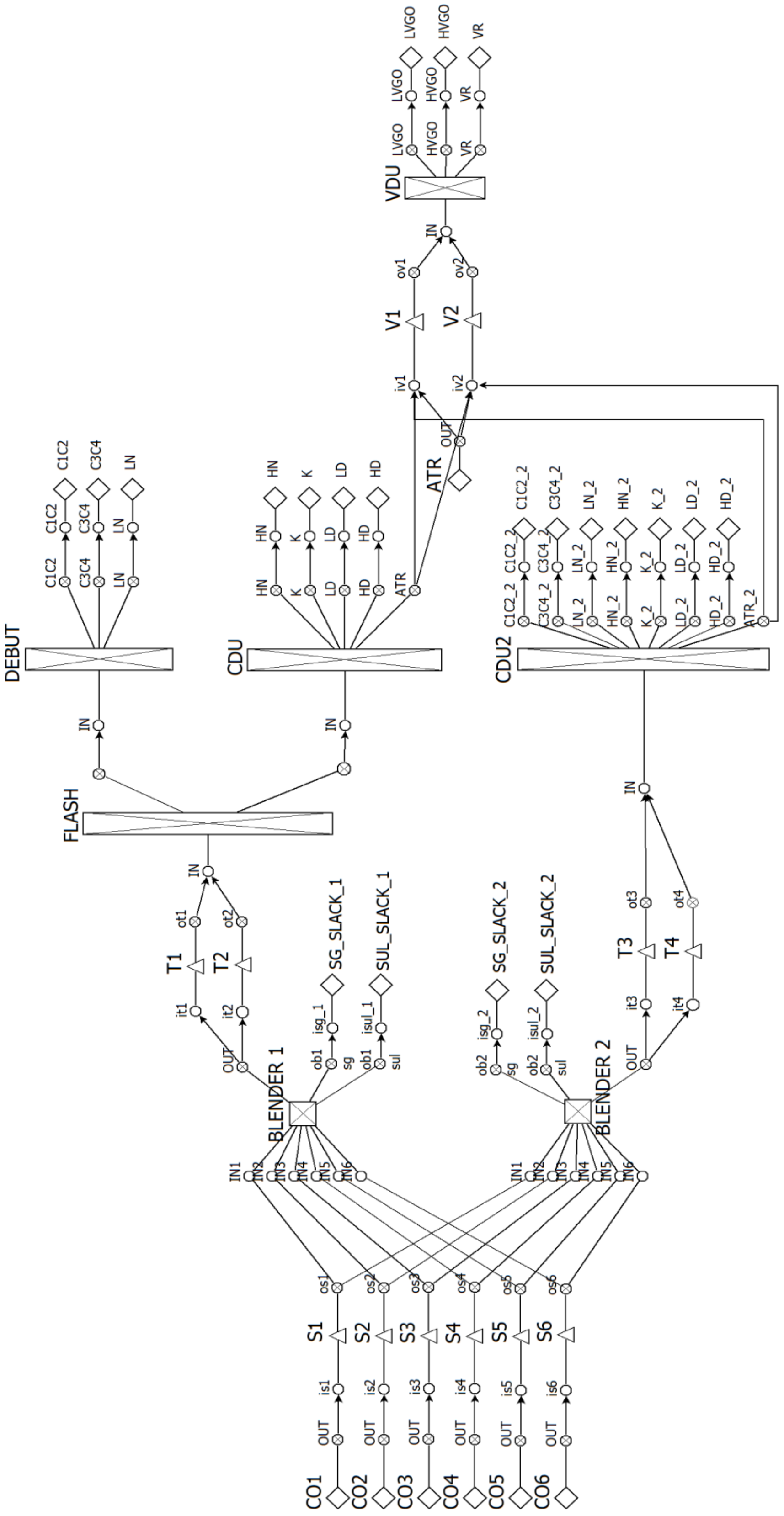
The computational time spent to run the PDH algorithm (using a total of three MILP-NLP iterations) is around 6 minutes for each example. The decomposition heuristic strategy successfully decomposed the MINLP model to provide acceptable optimization times of few minutes instead of what would typically take hours when optimizing a non-convex MINLP of this size and complexity. By including the linear reformulation of nonlinear constraints (factorizing strategy) in the MILP model, the MILP-NLP decomposition gap is effectively reduced whereas improving the MINLP objective function, as shown in Table 4.3.

Table 4.3: Results for the crude oil blend scheduling example.

	Without Factors	With Factors
MILP Solution (US$)	30,116	29,039
NLP Solution (US$)	28,076	28,869
MINLP Solution (US$)	28,076	28,869
Decomposition Gap (%)	6.77	0.59

Source: Author (2021).

Figure 4.6: Crude oil blend scheduling flowsheet including the factors reformulation.



Source: Author (2021).

Table 4.4 presents more detailed results for the crude oil blend scheduling problem with and without factors over three iterations of the decomposition heuristic approach.

Table 4.4: Results for the crude oil blend scheduling example with factors over three iterations of the decomposition algorithm.

	1st Iteration	2nd Iteration	3rd Iteration
MILP Solution (US$)	28,437	29,272	29,039
NLP Solution (US$)	25,985	26,439	28,869
MINLP Solution (US$)	25,985	26,439	**28,869**
Decomposition Gap (%)	8.62	9.68	0.59

Source: Author (2021).

The linearization strategy improves the quality of the MILP solution by considering the proxied information on qualities in the LP factor reformulation, achieving better MINLP solutions with an increase of 2.82% in the objective function. This improvement relies directly on the increase in the NLP objective function, and indirectly on the lower MILP-NLP gap (it is expected that a more accurate modeling leads to a lower decomposition gap, i.e., if the MILP formulation is more accurate and complete, the MILP solution will be more accurate and closer to the NLP solution). Therefore, the results indicate the efficiency of this strategy to improve the overall objective function of the MINLP model. This strategy may also lead to additional improvements, such as a better convergence of the decomposition algorithm by a reduced MILP-NLP gap. Although this example represents a medium-scale problem, as the strategy is applied only for blending operations, it is expected that the time increase for industrial problems would not limit its application.

The number of NLP optimizations is an important parameter to be tuned and affects the performance of the decomposition approach. Therefore, instead of 10 NLP optimization per iteration of the framework, additional tests are carried out using 50 NLP optimizations. The results are presented in Table 4.5.

Table 4.5: Results for the crude oil blend scheduling example with factors over three iterations of the decomposition algorithm using 50 NLP optimizations.

	1st Iteration	2nd Iteration	3rd Iteration
MILP Solution (US$)	29,617	30,338	30,085
NLP Solution (US$)	27,952	28,895	28,825
MINLP Solution (US$)	27,952	**28,895**	28,825
Decomposition Gap (%)	5.62	4.76	4.19

Source: Author (2021).

Increasing the number of NLP optimizations improves the overall MINLP solution but at higher computational time. This is a parameter to be tuned in the framework and depends on the desired purpose or application, as it significantly affects the total computational effort spent by the algorithm.

4.4 Examples 4 to 7: Design for Online Process and Blend Scheduling Optimization[2]

In the manufacturing with transformation of natural resources into products, different quality raw materials varying in composition are segregated, stocked, and blended to prepare the bulk feed quality of the plant. In such blend scheduling problem, the topology of the storage and blending operations, as well as the process design network, significantly affects the performance of the scheduling decision-making, which impacts the intermediate streams and final products. Considering the industrial advances toward online scheduling approaches, this section discusses the design aspects for improved blending and processing of compositional-level raw materials found in the petroleum (fractional), petrochemical (molecular), and metallurgic (atomic) industries. There are needs of diverse feed quality due to the highly complex and dynamic environment. Hence, a detailed and efficient design of the blend and processing scheduling is required for high-performance operations considering

[2] This section is based on the following manuscript:
FRANZOI, R. E.; MENEZES, B. C.; KELLY, J. D.; GUT, J. A. Design for Online Process and Blend Scheduling Optimization." In Computer Aided Chemical Engineering, v. 47, p. 187-192. Elsevier, 2019.

flexibility, responsiveness, management ability, etc. The proposed examples aim to compare computational efforts, and modeling and solution efficiency, depending on how the blending operations are modeled (using continuous blending or batch mixtures between tanks), and on the level of detail considered to model the distillation unit. The examples focus on the design for raw material- or feed-edge blend scheduling operation into online perspectives, although the same analyses are valid for the product-edge.

4.4.1 Introduction

The use of blender units in continuous operations is widespread in the product-edge manufacturing (i.e., back-end border of a plant), and there are several recent publications on blend scheduling applications in the petrochemical industry (CASTILLO-CASTILLO and MAHALEC, 2014; KELLY, MENEZES, and GROSSMANN, 2018). However, at the raw material- or feed-edge (or front-end border of a plant), the blend scheduling operations are commonly designed considering the standard topology of storage and feed raw material tanks or piles without a continuous blender. Thus, the blending operations of feedstocks and products use batch mixture of sequential streams for liquids such as crude oils using pipelines in refineries (LEE et al., 1996; JIA, IERAPETRITOU, and KELLY, 2003; CASTRO and GROSSMANN, 2014). It is similar for solids such as concentrated mineral flowing through conveyor-belts in metal processing sites (SONG et al., 2018).

Modeling and optimization of blending problems can be found in Misener and Floudas (2009) and considerations on blend scheduling solutions such as a sequential MILP approach in product blend-shops is presented in Kelly, Menezes, and Grossmann (2018). Towards online scheduling strategies, Franzoi et al. (2018a) shows improved operations using continuous blending of crude oils from the storage tanks to prepare the feed (charging) tanks for processing in the distillation unit. In the blend scheduling problem, the towers are represented as complex as they are designed, and hence, the model can include online scheduling aspects to be integrated to the solution, such as variable-feedback (GUPTA and MARAVELIAS, 2017a), parameter-feedback (FRANZOI et al., 2018b) and gradient-feedback (yet to be addressed within the online scheduling topic by the process systems engineering community).

The examples proposed in this section address the design for blending and processing of raw materials at the compositional level by utilizing a continuous blender unit for improved blending predictions and operations, and a detailed fractionation system in which the distillation unit is represented by towers in cascade. This is a typical problem in the petrochemical industry, in which most crude oil refineries still prepare the crude oil blend (feed for the distillation unit) without including a continuous blending unit. Industrial refinery operations often design blend scheduling problems by mixing batches of streams from the storage inventories to the feed inventories (stream by stream), since drawing and filling operations are typically not simultaneously allowed within the same tank or pool.

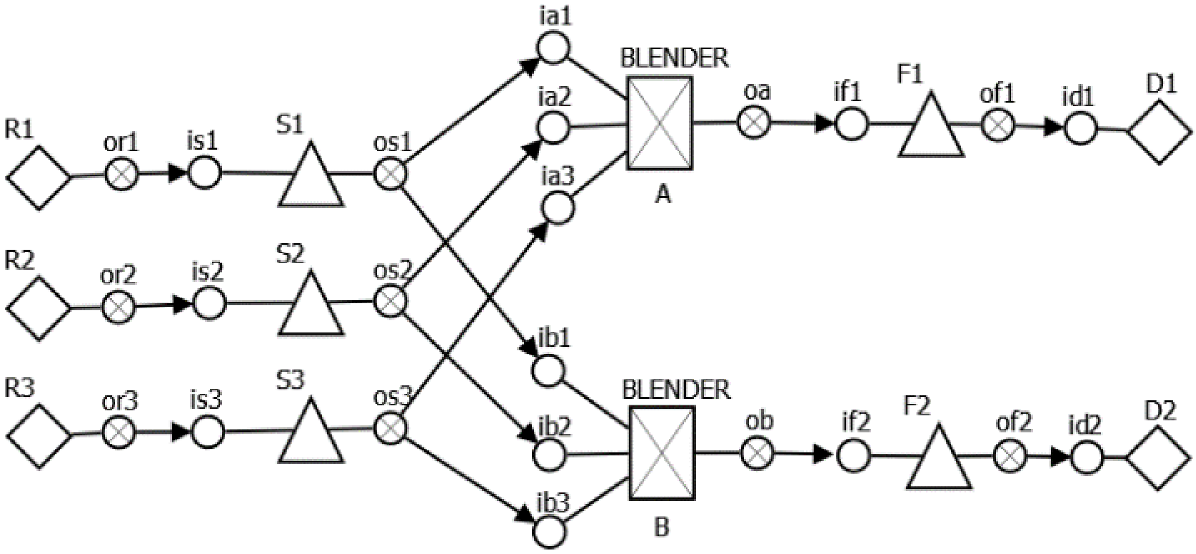
The research addressed in this section emphasizes the importance of efficient design and modeling approaches, and aims to provide information on the need of continuous blender units, measurement apparatus, and computational algorithms (such as control strategies) at the feed-edge of a plant. The process design, integrated to the proposed blend design, is modeled as a complex set of towers in cascade for the distillation unit. In previous literature, except by Franzoi et al. (2018a), the complex distillation unit is simplified within optimization problems by considering one or mostly two towers. The combination of the blend design (without or with a blender) and the process design (simplified or real design) demonstrates the operational pros and cons of the installation of continuous blending units and the modeling of a complex and detailed process design within online scheduling propositions.

4.4.2 Example 4: Blend scheduling operations

The blend scheduling operations presented in Figures 4.7 and 4.8 represent a typical problem in industries such as the petrochemical and metallurgic, in which the available feedstocks have to be blended to produce either a final product or an intermediate blend to be further used as the feed of other processing units. The scenarios considered in Figures 4.7 and 4.8 represent the same processing operation, but using or not a blender unit. The raw materials R1 to R3 are sent to the storage tanks S1 to S3, which are either sent directly to the feed tanks F1 to F2 or mixed in the blender unit BLENDER. In this case, there are two operating modes for the same physical blender unit, which allows to consider distinct conditions for each product or pool to be produced and stored in D1 or D2. Quality specifications on the maximum specific gravity and maximum sulfur content for the final products are constrained and must be met.

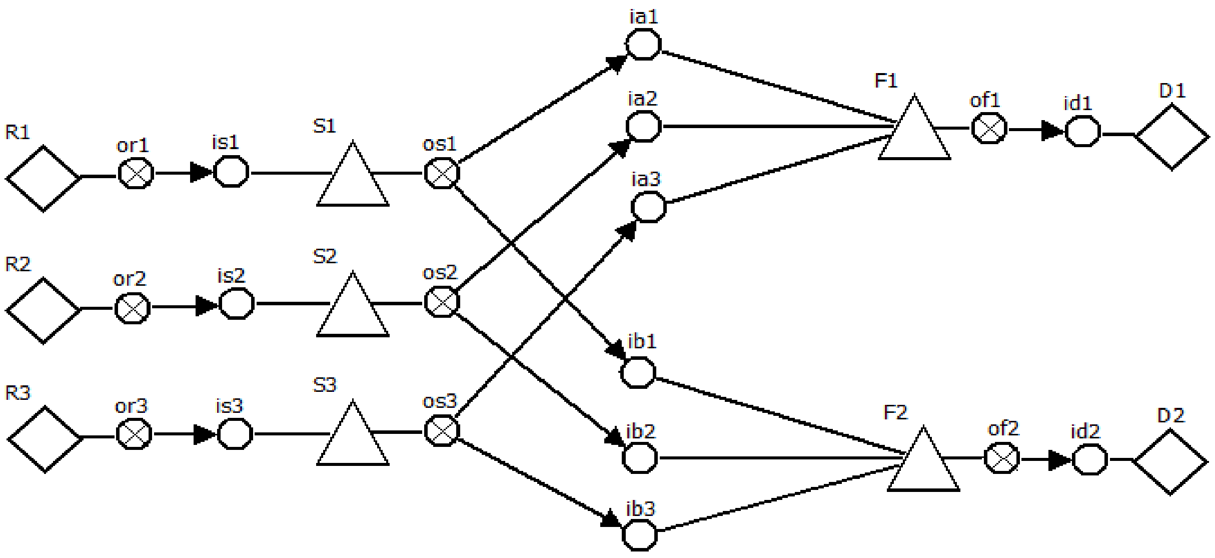
The blend scheduling problem is a non-convex mixed-integer nonlinear programming (MINLP) problem due to the nonlinearities from the blending equations, as well as the binary decisions regarding both the flows and the blender operating modes. Therefore, efficient modeling and solving techniques should be employed to provide a tractable formulation that can be solved in acceptable computational times (KELLY et al., 2017a). Then, the MILP-NLP phenomenological decomposition heuristic (MENEZES, KELLY, and GROSSMANN, 2015) is applied. The logistic constraints considered in the MILP formulation are: (a) Fill draw delay (i.e., after any filling operation, drawing operations for the same tank are only allowed after a certain amount of time); (b) Drawing-empty and filling-full (i.e., there are lower bounds for drawing and filling operations); (c) Multi-use (i.e., tanks can only send material to one sink, and can only receive material from one source simultaneously). For additional details on the logistic constraints, see Zyngier and Kelly (2009).

Figure 4.7: Blend scheduling problem (with blender) flowsheet.



Source: Author (2021).

Figure 4.8: Blend scheduling problem (without blender) flowsheet.



Source: Author (2021).

Table 4.6 compares the statistics of both cases, with and without using the continuous blender unit. The optimization maximizes the total profit and it is performed for the future 10-day time horizon with 4-hour time step using GUROBI (8.1.0) for the MILP sub-problem and IMPL's SLP linked to GUROBI (8.1.0) for the NLP sub-problem. The NLP objective function, MILP-NLP decomposition gap, and CPU time (s), are also presented in Table 4.6.

Table 4.6: Statistics of the blend scheduling problem with and without a continuous blender.

		With Blender	Without Blender
MILP	Binary Variables	1300	956
	Continuous Variables	1552	1068
	Constraints	9188	6664
	(Equality)	(1084)	(302)
	Degrees of Freedom	2070	1722
NLP	Continuous Variables	4018	3701
	Constraints	17704	9904
	(Equality)	(3826)	(6184)
	Degrees of Freedom	192	196
Profit ($)		583	430
Gap MILP-NLP (%)		1.4	13.0
Total CPU time (s)		200	1100

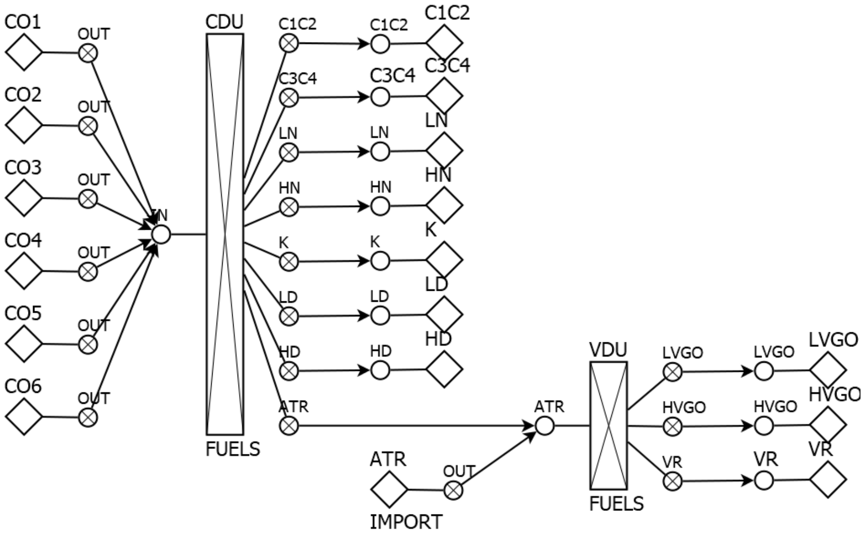
Source: Author (2021).

The continuous blender unit used in the formulation has two operating modes, which leads to 8 possible flows connecting the storage to the feed tanks, instead of 6 possible flows when there is no continuous blender (i.e., the storage and feed tanks are connected directly). Hence, a larger number of variables and constraints is expected for the scenario that utilizes the blender unit. However, that does not imply in an inefficient or more difficult to solve formulation. The results indicate that including a continuous blender in this blend scheduling example achieves improved operations with higher profit, smaller decomposition gap MILP-NLP, and lower computational effort. The increased economic value of the blending operations is achieved due to the limitations imposed by the logistic constraints (i.e., fill draw delay, drawing empty and filling full, and multi-use), which have a higher impact when there is no continuous blender in the process. Moreover, although including the blender leads to a larger model in size, a smaller decomposition gap and a faster optimization are achieved. Therefore, including the continuous blender is the preferred modeling and operational strategy for this example.

4.4.3 Example 5: Distillation unit operations with simplified and complex networks

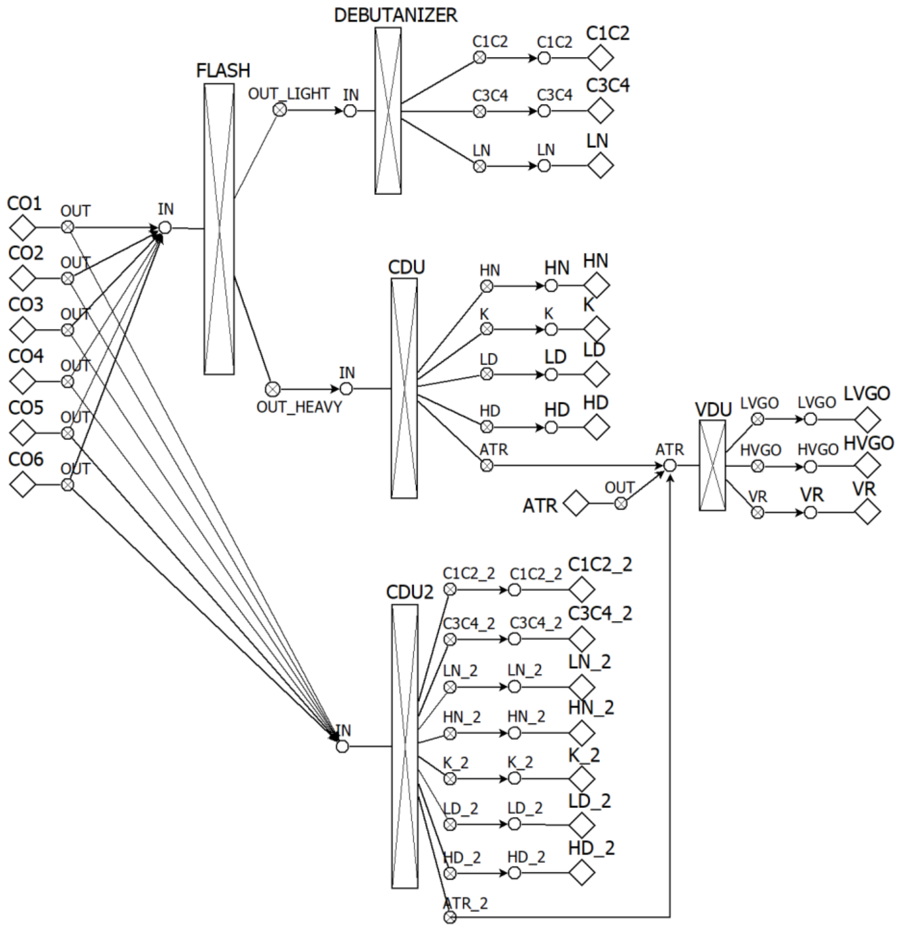
The processing example in Figure 4.9 presents the supply of crude oil feedstocks feeding a simplified set of distillation towers composed of atmospheric distillation (CDU) and vacuum distillation (VDU) towers. Similarly, Figure 4.10 considers a complex network of towers in cascade, composed of flash distillation (FLASH), debutanizer or naphtha stabilizer (DEBUTANIZER), two atmospheric distillation (CDU and CDU2), and vacuum distillation (VDU) towers. The main distillates produced are the fuel gas (C1C2), liquefied petroleum gas (C3C4), light naphtha (LN), heavy naphtha (HN), kerosene (K), light diesel (LD), heavy diesel (HD), and atmospheric residue (ATR). An external feed of ATR is also connected directly to the VDU, which produces light vacuum gas oil (LVGO), heavy vacuum gas oil (HVGO), and vacuum residue (VR). The crude oil assay or composition defined in micro-cut, hypo- or pseudo-component distribution is used to calculate the yields and properties of all towers. The production of distillates requires the temperature cutpoints to determine the aggregation of the micro-cuts to the final cuts (product distillates). Details on micro-cuts calculation and distribution can be found in Menezes, Kelly, and Grossmann (2013). The distillation unit networks in Figures 4.9 and 4.10 are modeled as single-period NLP problems.

Figure 4.9: Simplified distillation network flowsheet.



Source: Author (2021).

Figure 4.10: Complex distillation network flowsheet.



Source: Author (2021).

Both problems are optimized within seconds using IMPL’s SLP linked to the solver GUROBI (8.1.0). Four distinct optimizations are performed in order to maximize the production of each distillate, LN, HN, K, and LD, one at a time (i.e., the objective function in each optimization maximizes the production of one of the four distillates considered). The objective of these examples is to investigate how flexible are the operations to produce the final products and what is the impact on the scheduling solution. Depending on market demands and prices, flexible operations in which larger amounts can be produced of a given product present several economic and operational benefits (i.e., easier to schedule larger productions of specific products, and achieving higher economic value as well). Unlimited amounts of raw materials (CO1 to CO6) are considered. As the design with towers in cascaded network is more detailed in terms of processing flexibility, the optimization search space is wider and yields in a more efficient production, but at higher computational effort. No cutpoint optimization was considered for fractionation; therefore, the distillate products were calculated by the variations in the crude oils and the networked towers. The maximum amount produced of the distillates are shown in Table 4.7, in which Maximum LN refers to the maximum amount of LN that can be achieved in each distillation unit, and similarly for the other distillates.

Table 4.7: Maximum product yields (%) for each distillation network.

	Cascaded Towers Network	Simplified Network
Maximum LN	30.03	27.04
Maximum HN	15.55	13.48
Maximum K	34.11	30.82
Maximum LD	39.51	39.49

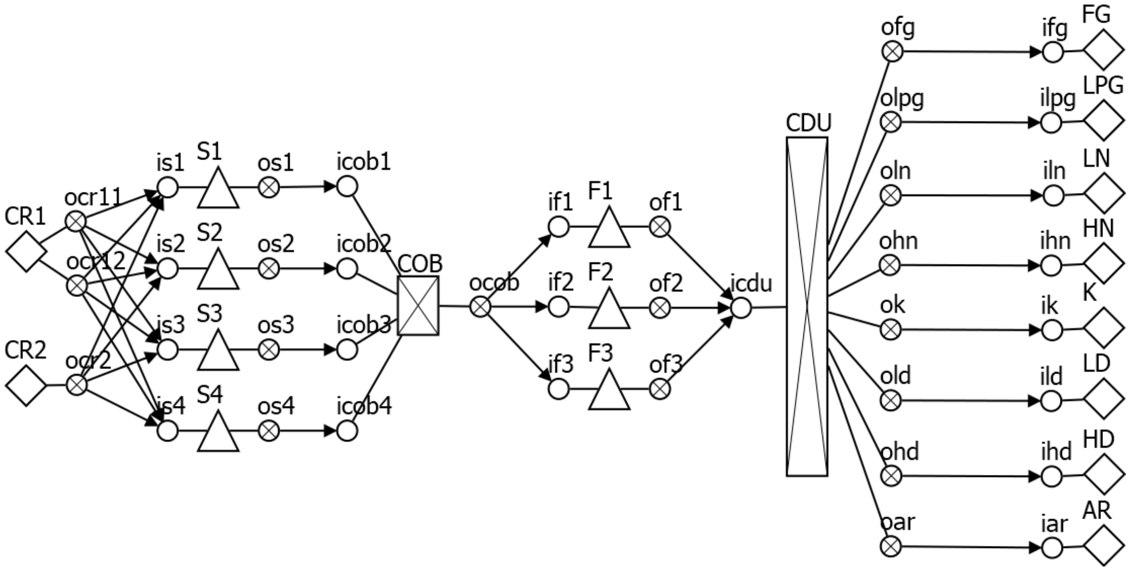
Source: Author (2021).

When the complex distillation network is used in the formulation, the scheduling optimization is more flexible and hence, can achieve a higher production of a specific distillate. That is especially helpful if the demand of a given distillate is high, or if there is an urgent need to increase the production of a particular distillate. Hence, that may significantly affect the economic value of the process.

4.4.4 Example 6: Blend scheduling with simplified processing operations

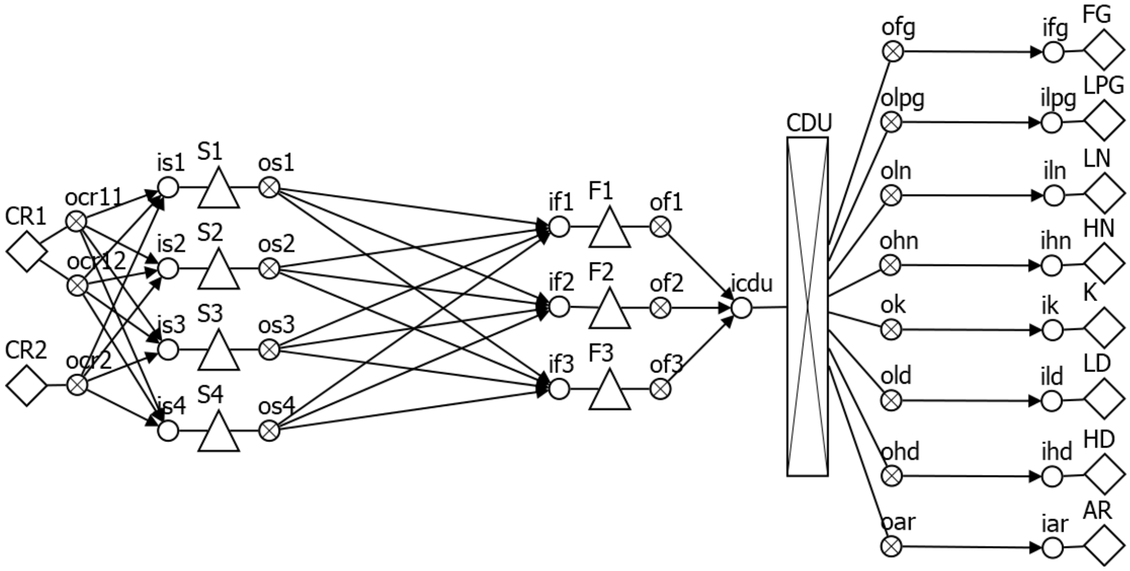
The common design for segregation, stock, and blending of feedstock raw materials (CR1 to CR2) for processing operations is represented in Figure 4.11. There are the storage (S1 to S4) and feed (F1 to F3) tanks (△) to temporarily store the raw materials with different qualities and compositions (i.e., yields and properties) to be processed in the separation and conversion units (e.g., distillation towers for separation of crude oils and smelting furnace for conversion of concentrate of minerals). Similarly, Figure 4.12 represents the same problem but without using a continuous blender unit. In both scenarios, a crude distillation unit is included to process the crude oil to produce the distillates.

Figure 4.11: Crude oil blend scheduling flowsheet with a continuous blender.



Source: Author (2021).

Figure 4.12: Crude oil blend scheduling flowsheet without a continuous blender.



Source: Author (2021).

The blend scheduling designs presented in Figures 4.11 and 4.12 utilize a simplified processing network composed of a single distillation unit. The problems are formulated as an MINLP. The modeling and optimization are performed for a 14-day time horizon with 4-hour time step. The modeling statistics for both scenarios, with and without using the continuous blender unit, are presented in Table 4.8.

Table 4.8. Modeling statistics of the blend scheduling with simplified processing problem.

		With Blender	Without Blender
MILP	Binary Variables	2,824	3,117
	Continuous Variables	3,331	3,374
	Constraints (equality)	10,841 (2,356)	9,146 (1,430)
	Degrees of Freedom	4,391	5,061
NLP	Continuous Variables	11,785	14,389
	Constraints (equality)	12,055 (11,041)	13,819 (13,141)
	Degrees of Freedom	744	1248

Source: Author (2021).

In this example, including the continuous blender in the formulation leads to a model smaller in size, because there is only one operating mode and hence, there are fewer possible flows when the blender is used. Tables 4.9 and 4.10 present the results of the MILP-NLP optimization for the blend scheduling with simplified processing problem. The results indicate that for both MILP and NLP optimizations the objective functions are very close independently on whether the blender unit is used or not. Furthermore, the computational times are slightly higher for the case without blender. In Table 4.9, the MILP problem is solved with CPLEX (12.8.0) and the NLP problem is solved with IMPL's SLP linked to this solver. It is similar in Table 4.10 but using the solver GUROBI (8.1.0) for both optimizations.

Table 4.9: Optimization results of the blend scheduling with simplified processing using CPLEX (12.8.0).

	MILP		NLP	
	Profit ($)	CPU (s)	Profit ($)	CPU (s)
With Blender	35,387	2.3	35,120	60.2
Without Blender	35,387	5.1	35,120	65.3

Source: Author (2021).

Table 4.10: Optimization results of the blend scheduling with simplified processing using GUROBI (8.1.0).

	MILP		NLP	
	Profit ($)	CPU (s)	Profit ($)	CPU (s)
With Blender	35,387	1.6	35,520	69.4
Without Blender	35,387	6.4	35,520	72.2

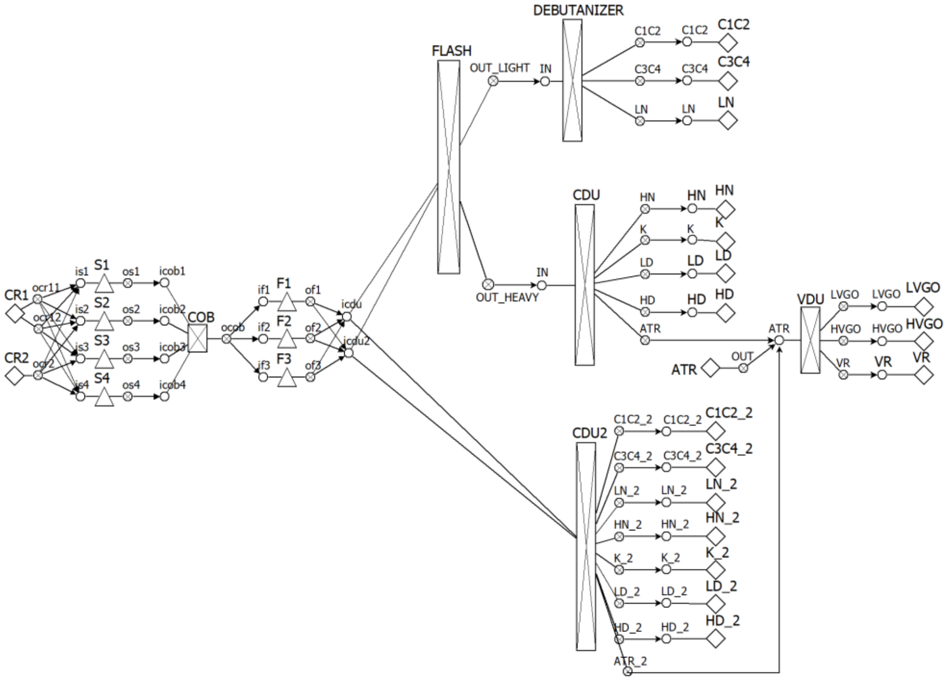
Source: Author (2021).

Table 4.9 shows that the same objective function is obtained in both scenarios (using or not the continuous blender), although a lower computational effort is required for the optimization with blender, hence this would be the preferred design for that example. Similar results are obtained in Table 4.10, when GUROBI is used instead of CPLEX for the optimizations, in which a slightly higher objective function is obtained at a higher computational effort.

4.4.5 Example 7: Blend scheduling with complex processing operations

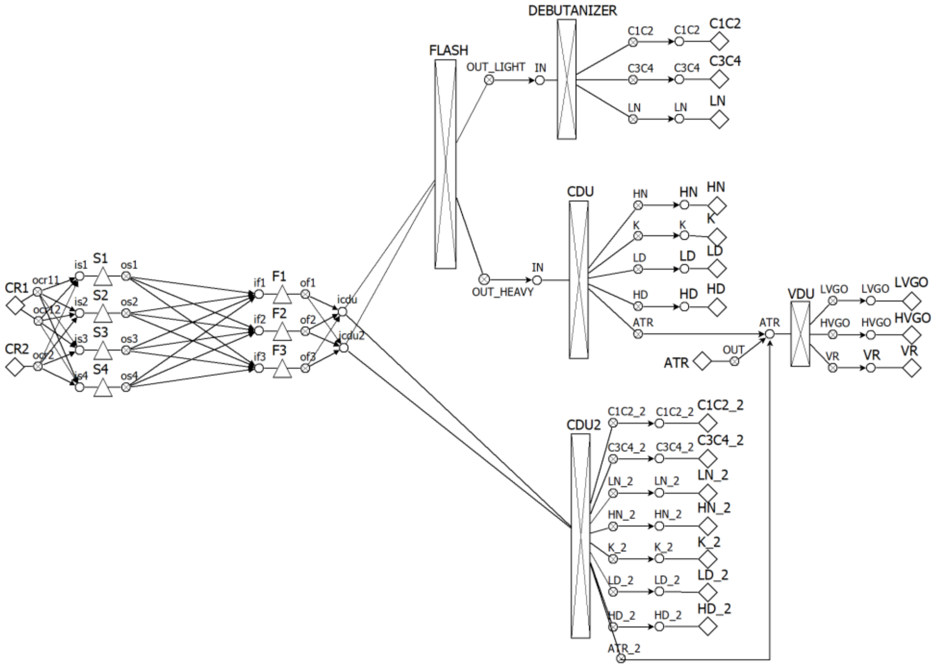
The blend scheduling with simplified process design examples (Figures 4.11 and 4.12) are merged with the complex towers in cascade distillation (Figure 4.10) to generate the blend scheduling with complex processing examples shown in Figures 4.13 and 4.14. Additional tanks are also included to manage the three different feeds outgoing from the VDU unit. These are more realistic examples of real industrial operations, which include the blending operations required to prepare the feed for the distillation unit, as well as a complex set of towers in cascade to provide a proper and detailed modeling of the distillation process.

Figure 4.13: Crude oil blend scheduling flowsheet with continuous blender.



Source: Author (2021).

Figure 4.14: Crude oil blend scheduling flowsheet without continuous blender.



Source: Author (2021).

The examples are formulated as MINLP problems and are optimized for a 14-day time horizon with 4-hour time step. The modeling statistics are shown in Table 4.11.

Table 4.11: Modeling statistics of the blend scheduling with complex processing problem.

		With Blender	Without Blender
MILP	Binary Variables	4,166	4,620
	Continuous Variables	7,647	8,151
	Constraints (equality)	16,519 (3,195)	12,849 (2,859)
	Degrees of Freedom	8,568	9,912
NLP	Continuous Variables	20,199	21,331
	Constraints (equality)	21,471 (19,675)	22,779 (21,015)
	Degrees of Freedom	524	484

Source: Author (2021).

The MILP problem is solved with CPLEX (12.8.0) and GUROBI (8.1.0), and the NLP problem is solved with the IMPL's SLP tool linked to each solver. The optimization results for the optimization with each solver are presented in Tables 4.12 and 4.13.

Table 4.12: Optimization results of the blend scheduling with complex processing problem using CPLEX (12.8.0).

	MILP		NLP	
	Profit ($)	CPU (s)	Profit ($)	CPU (s)
With Blender	48,618	2.3	42,675	416.8
Without Blender	47,569	6.6	39,048	1079.1

Source: Author (2021).

Table 4.13: Optimization results of the blend scheduling with complex processing problem using GUROBI (8.1.0).

	MILP		NLP	
	Profit ($)	CPU (s)	Profit ($)	CPU (s)
With Blender	48,618	2.1	41,774	421.1
Without Blender	47,569	3.7	36,377	3316.1

Source: Author (2021).

By the results in Tables 4.12 and 4.13, the complex process design using a blender unit for improved blending design achieves higher objective functions at lower CPU times in both the MILP and NLP problems, as well as reduced MILP-NLP decomposition gaps regarding their objective functions. The continuous blender unit is shown to be especially useful when the formulation includes a complex distillation network. That indicates potential benefits on the integration of improved blending and processing design towards improved operations.

The processing and blend scheduling optimization are proposed for both blend design and process design with the purpose of comparing the simple and the complex network structures. The installation of a continuous blender unit for feedstocks (as in Figures 4.11 and 4.13) is the preferred design for improved blend scheduling operations, as it improves the process feed quality matching at the compositional level, leads to a wider optimization search space, and provides the possibility of using better advanced process control (APC) and real-time optimization (RTO) approaches for blending operations. The use of a complex distillation network is the preferred design for improved processing operations, which provides improved and more accurate predictions, as well as a better processing flexibility.

The perfect match of blend design and process design must be coherent with the blend scheduling optimization of the composition of raw materials. Hence, the installation of continuous blender units (better blend design) and the efficient process design modeling (better process design) is recommended for high-performance operations. Despite the higher computational time due to the more complex blend scheduling and process designs, to achieve better solutions of NP-hard scheduling problems in acceptable time is becoming possible due to: a) advancements in decision-making modeling (with the use of the UOPSS flowsheet and the heuristic decomposition of MINLP as MILP+NLP); b) solving algorithms (improvements in commercial solver updates); and c) computer-aided resources (more powerful computer processing and memory).

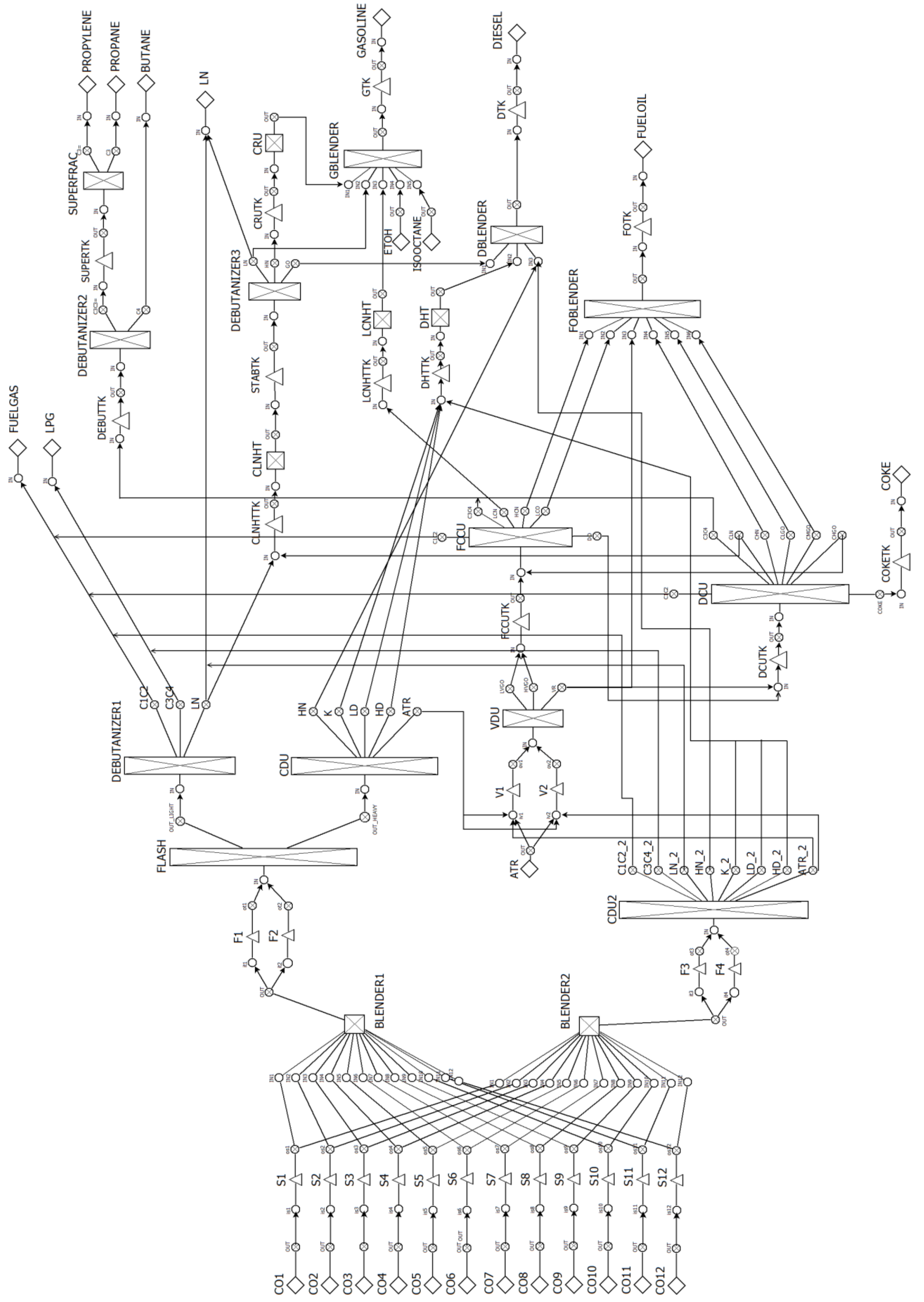
4.5 Example 8: Crude oil refinery scheduling operations

Example 8 extends the scope of the crude oil blend scheduling problem to include more feedstocks (a total of twelve distinct crude oils), multiple storage tanks, two blenders to prepare the feed for the distillation unit, a cascaded distillation unit with five towers (two atmospheric distillation units, a vacuum distillation unit, a pre-flash tower, and a debutanizer), and the whole process-shops, including additional blenders (GBLENDER, DBLENDER, FOBLENDER), fluid catalytic cracker (FCCU), hydrotreaters for coke naphtha (CLNHT), cracked light naphtha (LCNHT), and diesel (DHT), delayed coker (DCU), debutanizers (DEBUTANIZER1 to DEBUTANIZER3), superfractionator (SUPERFRAC), and catalytic reformer (CRU). Figure 4.15 presents the flowsheet of the crude oil refinery scheduling problem.

There are 12 distinct sources of crude oil CO1 to CO12 to feed the refinery operations. These crudes are stored in the storage tanks S1 to S12, which are blended through the blender units (BLENDER1 and BLENDER2). The crude blend is sent to the feed tanks F1 to F4 to further feed the distillation unit composed of a set of five towers.

The DEBUTANIZER1 tower produces fuel gas, LPG, and light naphtha. The distillation column CDU produces heavy naphtha, kerosene, light diesel, heavy diesel, and atmospheric residue. The distillation column CDU2 produces fuel gas, LPG, light naphtha, heavy naphtha, kerosene, light diesel, heavy diesel, and atmospheric residue. The fuel gas and LPG produced in these units are sent to product storage tanks (FUELGAS and LPG, respectively). Light naphtha is sent either to a reservoir (LN) or to a coke naphtha hydrotreating unit (tank CLNHTTK and hydrotreater CLNHT). The heavy naphtha is sent to a diesel mixer (DBLENDER). Kerosene, light diesel, and heavy diesel are sent to a diesel hydrotreating unit (tank DHTTK and hydrotreater DHT). The atmospheric residue is sent to the vacuum distillation unit (VDU). In the diesel hydrotreating unit, kerosene and diesel are hydrotreated and sent to the diesel blender (DBLENDER), whose product is sent to a tank (DTK) and later stored in a final pool of diesel (DIESEL). The vacuum distillation column produces light vacuum diesel, heavy vacuum diesel, and vacuum residue. The two 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).

Figure 4.15: Crude oil refinery scheduling flowsheet.



Source: Author (2021).

Light hydrotreated naphtha in the CLNHT is sent to a debutanizer unit for the stabilization of naphtha, composed of a tank (STABTK) and a debutanizer column (DEBUTANIZER2). The output fractions are light naphtha, which is sent either to a naphtha storage tank (LN) or to a gasoline mixer (GBLENDER); heavy naphtha, sent to a catalytic reform unit (tank CRUTK and column CRU); and diesel, sent to the diesel mixer (DBLENDER). The heavy naphtha is processed in the catalytic reformer unit and is subsequently sent to the blender GBLENDER for gasoline production.

The fluidized catalytic cracking unit (FCCU) produces fuel gas, LPG, cracked light naphtha, cracked heavy naphtha, light recycling oil, and decanted oil. Fuel gas and liquefied gas are sent to product storage tanks. Cracked light naphtha is sent to a hydrotreating unit (tank LCNHTTK and hydrotreater LCNHT). Heavy cracked naphtha and light cracked oil are sent to a fuel oil mixer (FOBLENDER). Decanted oil is sent to the delayed coking unit (tank DCUTK DCU column DCU).

In the cracked light naphtha hydrotreatment unit, hydrotreated naphtha is sent to the gasoline mixer (GBLENDER), in which ethanol and isooctane (from the pools ETOH and ISOOCTANE) are also added. The GBLENDER output is sent to a tank (GTK) and later stored in a gasoline tank (GASOLINE).

The delayed coking unit (DCU) processes vacuum residue and decanted oil, producing fuel gas, liquefied gas, light coke naphtha, heavy coke naphtha, light coke diesel, medium coke diesel, heavy coke diesel, and coke. Fuel gas and LPG are sent to storage tanks. Light coke naphtha is sent to the CLNHT hydrotreatment unit. Heavy coke naphtha, light coke diesel, and medium coke diesel are sent to the fuel oil mixer (FOBLENDER). Heavy coke diesel is sent to the FCCU. The coke produced is sent to a tank (COKETK) and later stored in a coke reservoir (COKE).

The problem to be optimized maximizes the profit of the refinery operations. Two scenarios with time horizons of 5 days and intervals of 12 and 4 hours, with a total of 10 and 30 time periods, respectively, are proposed and discussed as follows. The original crude oil refinery scheduling problem is formulated as an MINLP, which is broken down using the phenomenological decomposition heuristic into an MILP problem to be solved with CPLEX (12.8.0), and an NLP problem to be solved with the IMPL's SLP built-in linked to CPLEX.

4.5.1 Scenario 4.5.1: Time horizon of 5 days and time steps of 12 hours

In the MILP formulation there are 1,274 equality and 8,474 inequality constraints, 2,039 continuous variables, 2,178 binary variables, and 2,276 degrees of freedom. In the NLP formulation there are 9,724 equality constraints, 6,010 inequality restrictions, 12,104 continuous variables, and 292 degrees of freedom. Four iterations of the decomposition algorithm are performed, and the stopping criterion adopted is the maximum number of iterations. Table 4.14 presents the objective functions for the MILP solutions in each iteration of the phenomenological decomposition heuristic.

Table 4.14: Objective functions for the MILP solutions in each iteration.

1ª iteration	2ª iteration	3ª iteration	4ª iteration
66,907	62,978	63,964	63,964

Source: Author (2021).

Each NLP problem is optimized 10 times by using the automatic and systematic random generation tool to randomize the initial values for the decision variables. The objective functions for the NLP optimizations in each iteration of the phenomenological decomposition heuristic are presented in Table 4.15.

Table 4.15: Objective functions of the NLP solutions at each iteration.

1ª iteration	2ª iteration	3ª iteration	4ª iteration
39.203	39.575	42.893	40.344
56.152	46.714	57.474	45.728
46.053	-	43.111	42.912
53.935	56.864	49.178	49.228
-	48.414	47.929	46.253
57.761	51.004	49.245	51.478
60.467	49.066	46.714	49.061
48.242	46.738	46.949	46.909
57.112	46.398	-	-
58.173	-	55.688	56.576

Source: Author (2021).

The blank spaces in Table 4.15 are infeasible NLP solutions (12.5 %) because the optimization solver reached a maximum number of iterations in the optimization routine without finding feasible solutions. The best solution is found in the first iteration with a decomposition gap MILP-NLP of 9.35%. The computational time spent to run the entire algorithm is around 5 minutes.

4.5.2 Scenario 4.5.2: Time horizon of 5 days and time steps of 4 hours

In the MILP formulation there are 3,814 equality and 25,594 inequality constraints, 5,739 continuous variables, 6,138 binary variables, and 6,824 degrees of freedom. In the NLP formulation there are 29,164 equality constraints, 18,030 inequality restrictions, 34,104 continuous variables, and 771 degrees of freedom. Four iterations of the decomposition algorithm are performed, and the stopping criterion adopted is the maximum number of iterations. Table 4.16 presents the objective functions obtained for MILP problems in each PDH iteration.

Table 4.16: Objective functions for the MILP solutions in each iteration.

1ª Iteration	2ª Iteration	3ª Iteration	4ª Iteration
68.831	75.283	58.417	58.363

Source: Author (2021).

Each NLP problem is optimized 10 times using distinct initial values for the variables. The objective functions for each PDH iteration are shown in Table 4.17.

Table 4.17: Objective functions of the NLP solutions at each iteration.

1ª Iteration	2ª Iteration	3ª Iteration	4ª Iteration
-	-	-	-
47.494	-	43.934	-
-	45.975	46.266	-
-	48.626	-	**60.544**
46.604	-	51.599	47.272
53.616	-	-	56.058
-	-	-	45.811
55.214	46.246	47.626	56.005
50.679	-	-	-
57.829	-	-	-

Source: Author (2021).

A total of 22 NLP solutions did not converge (55%) due to the optimization solver stop criterion regarding the maximum number of iterations. The best objective NLP function (60,544) is found in iteration 4 with a deviation of -3.73% regarding its respective MILP solution, with the negative sign representing an MILP solution smaller than its respective NLP solution. The computational time spent to optimize all MILP and NLP iterations was 40 minutes. Reducing the time step from 12 hours to 4 hours improved the optimal solution 60.467 to 60.544, although there is an increase in the computational time of 300 %. There is a tradeoff between the quality of the solution and the computational effort required for the optimization.

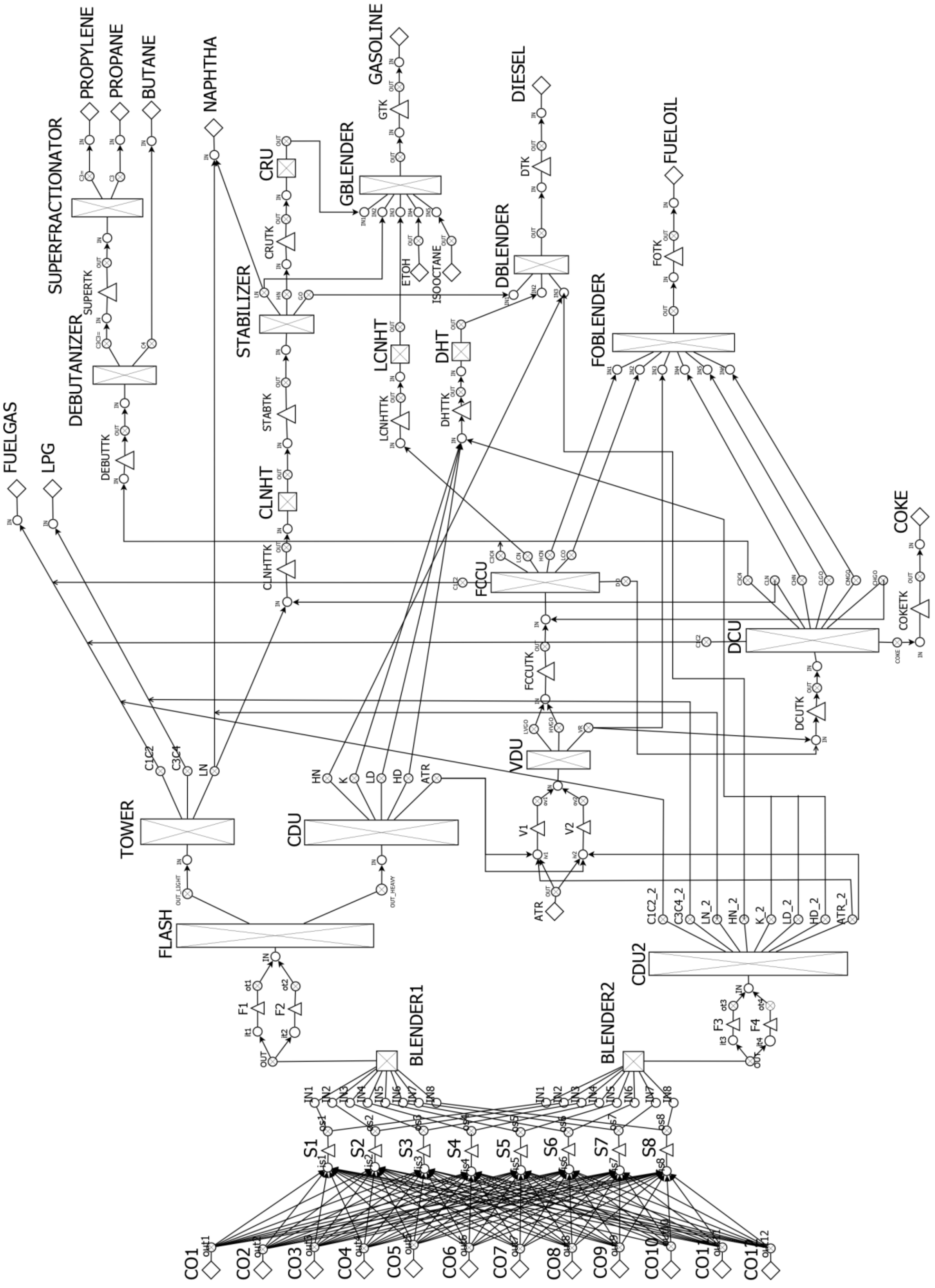
4.6 Example 9: Refinery scheduling operations using a chronological decomposition heuristic approach

Example 9 extends the scope of the crude oil blend scheduling problem to include more feedstocks (a total of twelve distinct crudes), eight storage tanks, two blenders, the cascaded distillation unit with five towers (two atmospheric distillation units, a vacuum distillation unit, a pre-flash tower, and a debutanizer), and the whole process-shops, which includes additional blenders, catalytic cracker, hydrotreaters, delayed coker, debutanizers, superfractionator, and reformer. Figure 4.16 presents the flowsheet of the crude oil refinery scheduling problem.

A discrete-time model is formulated for the future 15 days with time steps of 8 hours, in a total of 45 time periods. Additional modeling strategies are included in the formulation to achieve a more efficient modeling and hence, a faster and more efficient optimization: a) exclusions in rolling horizon strategies (KELLY, 2002); and b) relaxations in relax-and-fix iterations to construct the complete MILP problem by integrating the solution of the subproblems. For additional details, see Kelly and Mann (2004).

In the base case scenario (not using the rolling horizon and relax-and-fix approaches), the MILP model has around 8000 continuous and 9000 binary variables, 44000 constraints (5000 equality) and 10000 degrees of freedom, whereas the NLP model has around 50000 continuous variables, 70000 constraints (44000 equality) and 1000 degrees of freedom. To handle such complex model, the phenomenological decomposition heuristic (MENEZES, KELLY, and GROSSMANN, 2015) is employed.

Figure 4.16: Crude oil refinery scheduling flowsheet.



Source: Author (2021).

The chronological decomposition heuristic in Kelly (2002) consists of a rolling horizon approach that splits the time horizon in time chunks and optimize the problem in successive integrated subproblems with or without crossover between time windows, i.e., with recalculation of the time periods in the neighborhood. There are six scenarios proposed to investigate the impact of the rolling horizon strategy on the optimal solution. Scenario 4.6.1 is the optimization without using the chronological decomposition heuristic (i.e., this is the base case scenario in which the problem is optimized for the entire time horizon without decomposition). Scenarios 4.6.2 to 4.6.6 are formulated using the rolling horizon approach, so that multiple sequential optimizations are performed for each of them so as to consider the entire time horizon. For example, scenario 4.6.2 is composed of three sequential optimizations, whereby the first optimizes days 1 to 5, the second optimizes days 6 to 10, and the third optimizes days 11 to 15. When crossover is considered, the time horizon within the crossover range is optimized twice. For example, there is a crossover from days 6 to 10 in scenario 4.6.5, which means that the time horizon for the first optimization is from day 1 to 10, and the time horizon for the second optimization is from day 6 to 15. The optimizations are carried out using the solver CPLEX 12.7.1 in an Intel Core i7 with 2.7 GHz and 16 GB RAM. Table 4.18 shows the optimization results (objective function and computational time) for each scenario.

Table 4.18: Optimization results for the scenarios using the rolling horizon approach.

Scenario	Rolling Horizon	Optimization 1	Optimization 2	Optimization 3	Profit (US$)	CPU Time (s)
4.6.1	No	Days 0 – 15	-	-	85,558	295
4.6.2	Yes	Days 0 – 5	Days 6 - 10	Days 11 - 15	84,995	115
4.6.3	Yes	Days 0 -5	Days 6 – 15	-	85,348	112
4.6.4	Yes	Days 0 – 10	Days 11 – 15	-	84,232	83
4.6.5	Yes	Days 0 -10	Days 6 - 15	-	85,447	38
4.6.6	Yes	Days 0 -10	Days 8 - 15	-	85,522	40

Source: Author (2021).

The results in Table 4.18 indicate that the best rolling horizon configuration among the configurations tested are Scenarios 4.6.5 and 4.6.6, which include crossover. Scenario 4.6.6 achieves a reduction of 86.4 % in the computational time and a reduction of only 0.042 % in the objective function when compared to the base case, which is expected due to the decomposition applied (i.e., decomposition strategies typically lead to solving limitations such as smaller search space in the optimization, and hence, to worse solutions). It is worth mentioning that the main objective of this type of strategy is achieve faster solutions by allowing acceptable losses in the objective function. This is especially useful to provide tractable formulations for the optimization of large-scale time-limited applications, such as the crude oil refinery scheduling problem.

The other heuristic strategy applied for the example shown in Figure 4.16 is the relax-and-fix approach, which optimizes a relaxed version of the original MINLP refinery scheduling problem, fixes all the relaxed variables that are active (i.e., equal to one according to certain tolerance), and reoptimizes the MINLP problem. The first optimization is much faster because the relaxed MINLP becomes an NLP problem, and the second optimization is expected to have a significantly lower number of binary variables than the original problem. Therefore, this strategy may be useful to reduce the computational time for the optimizations, which is especially helpful for handling intractable large-scale formulations. The optimization results for Scenario 4.6.7, which uses the relax-and-fix approach, are presented in Table 4.19, and compared to the original MINLP results from Scenario 4.6.1.

Table 4.19: Optimization results for the scenario using the relax-and-fix approach.

Scenario	Relax-and-fix approach	Profit ($)	CPU Time (s)
4.6.1	No	85,558	295
4.6.7	Yes	85,421	37

Source: Author (2021).

The results in Table 4.19 indicate that a computational time 85.5 % lower is achieved by using the relax-and-fix approach, with a reduction of only 0.16 % in the objective function.

4.7 Example 10: Effective scheduling of complex process-shops using online parameter feedback in crude oil refineries[3]

Integrated scheduling optimization comprising the battery limits of crude oil refineries is a challenging problem to be solved as it includes decisions concerning the quantity and quality of the crude oil feedstocks and final products (such as fuels and petrochemicals), as well as the refinery production network. So far, the literature on crude oil scheduling optimization has covered the problem from the crude oil unloading and storage up to the distillation straight-run streams. To go further, this section extends the scope of the problem from the raw material deliveries up to product liftings through the refinery process-shop by using closed-loop, online and routine process feedback data from field and laboratory measurements for better process predictions, integrated within the scheduling cycle. For such engine, past routine operating data calibrates gains and biases as $y_{measured} = gain * y_{model} + bias$, whereby the data updating in y considers both process yields and variables such as throughputs, flows, holdups, and properties, whose effects propagate throughout the process network. Parameter feedback is applied after data reconciliation computation in a complete crude oil refinery blend scheduling problem considering real tank topology, cascaded distillation towers, process-shops, and blend-shops, so as to effectively optimize the complex process system. The feedback strategy is solved within an iterative mixed-integer linear and nonlinear programming (MILP-NLP) decomposition by updating NLP results of process-shop's yields and properties, and recipes of blend-shops in the next MILP solution until convergence is achieved.

4.7.1 Introduction

An effective optimization integrating the operations of scheduling in refining of crude oils into fuels, lubes, asphalts, and petrochemical feeds is a challenging problem to be solved. It involves decisions concerning logistics and storage of crude oil feedstocks

[3] This section is based on the following manuscript:

FRANZOI, R. E.; MENEZES, B. C.; KELLY, J. D.; GUT, J. A. W. Effective scheduling of complex process-shops using online parameter feedback in crude-oil refineries. In Computer Aided Chemical Engineering, v. 44, p. 1279-1284, 2018.

and final products as well as the processing and blending in the refinery network. Approaches to promote the integration of these multi-entity, multi-activity and multi-resource decision-making require efficient modeling and solving strategies that considers the difficulties posed by these logistics and quality aspects.

In terms of modeling, previous literature on crude oil scheduling optimization covers the scheduling from the crude oil unloading and storage up to the representation of the crude oil atmospheric and vacuum distillation towers and their immediate straight-run streams (KELLY et al., 2017a). Predictions of product yields and properties from process-shops, where the crude oil blend diet feed is separated into straight-run distillation intermediate products for further downstream reactions, conversions, separations, treatments, and blending throughout the remaining refinery unit-operations and tanks, remain a substantial limitation for better scheduling operations. Modeling such processes reliably and accurately is difficult due to the complexities and uncertainties in the feed quality, operating conditions, process measurements, etc. and therefore parameter feedback is required.

In terms of solving, the search for optimized solutions in the crude oil scheduling may succeed with an MILP-NLP decomposition strategy instead of a full space MINLP approach, since in the latter a binary variable relaxation of the logistics problem is solved along with the non-convex blending and processing relationships in the NLP steps. Recently, Kelly et al. (2017a) solved a discrete-time benchmark for a 1-week time horizon with 2-hour time step involving the logistics details of ship arrivals, tanks, and process units in an MILP as well as the quality calculations in an NLP for the blending of streams (e.g., feed diet of distillation towers) and for the transformation of crude oils into distillates. However, the benchmark excludes further physical and chemical transformations in other process units beyond the initial distillation towers. In this example, a problem covering the entire refinery scheduling process is modeled, from crude oil unloading and storage up to fuel deliveries by including process- and blend-shops as well as the fine tune adjustments regarding on online measurement updates from the field.

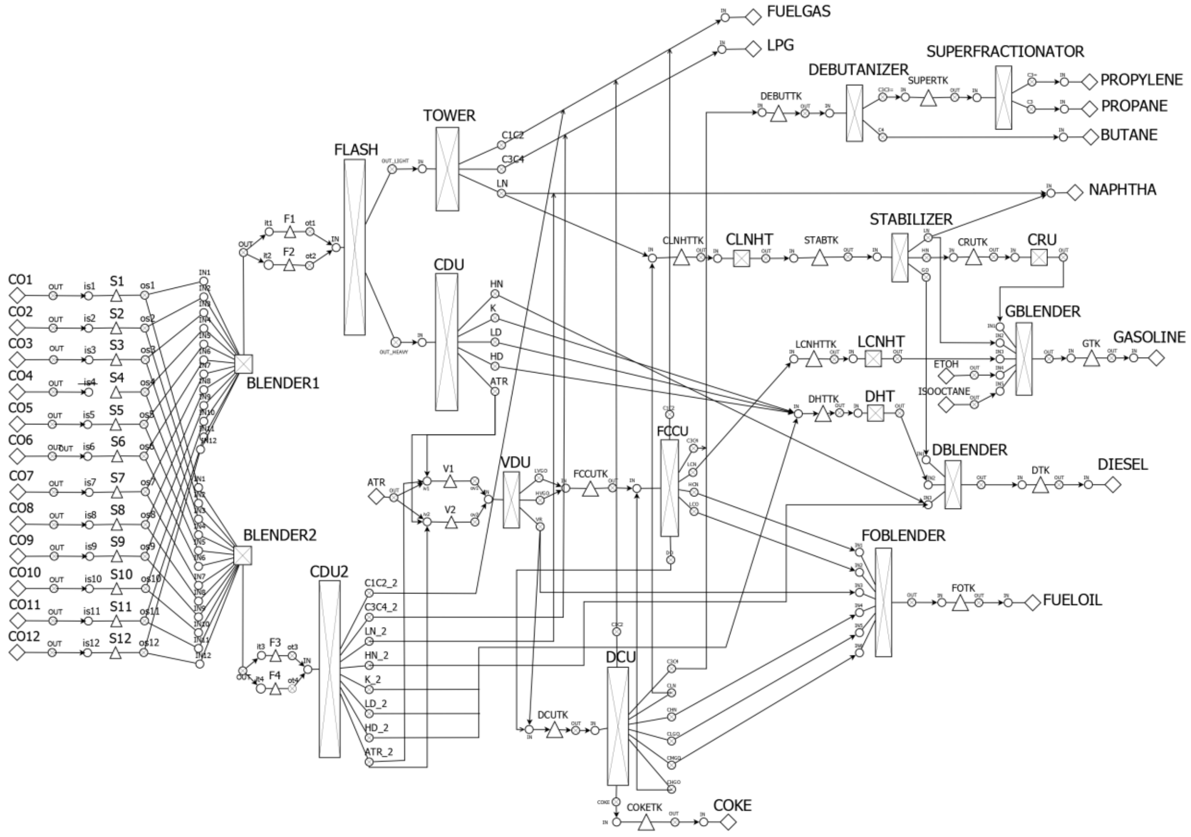
With the evolution of solving capabilities determining crude oil blend scheduling problems with 1-hour discrete time step, there are reasons to evolve into a more complex scope and scale in the modeling frontier by including cascaded distillation towers, downstream process-shops and blend-shops (e.g., to specify products at their contracted deliveries). Furthermore, an online scheduling algorithm can better predict the model by including the valid plant state (measured) by using parameter feedback and operational updates from the field. According to Subramanian, Maravelias, and Rawlings (2012), disruptions or arrival of new information can make the incumbent schedule suboptimal or infeasible, motivating the need for online scheduling. Major considerations on strategies (GUPTA and MARAVELIAS, 2016; GUPTA, MARAVELIAS, and WASSICK, 2016), design (GUPTA and MARAVELIAS, 2017b) and formulations (GUPTA and MARAVELIAS, 2017a) introduce the online scheduling modeling aspects, although other essential elements as re-scheduling activation and frequency as well as handling of process uncertainty are imperative.

For such online scheduling improvement, the use of parameter feedback is proposed considering the past routine operating data to calibrate gains and biases such as $y_{measured} = gain * y_{model} + bias$. Data reconciliation and estimation steps are necessary for the next scheduling cycle. The parameter updating is performed by offsetting the lower, upper, and target bounds of variables for output-uncertainty, and by creating new flow and quality variables for input-uncertainty overriding when these variables are propagating in the flowsheet.

4.7.2 Problem statement

The flowsheet in Figure 4.17 shows the operational scheduling optimization of a refinery with real scale topology considering the crude oil supply, storage and feed tank operations, production in cascaded distillation towers, operations of process-shops and blend-shops of both crude oils and products. The marine vessels or feedstock tanks (CO1 to CO12) supply a crude oil refinery with different quality of raw materials. The assignment of crude oil feedstocks to storage tanks (S1 to S12) can be pre-defined by clustering similar quality raw materials (KELLY et al., 2017b). Storage tanks are connected to crude oil blenders for preparation of distillation tower feeds or diet to be stocked temporarily in feed tanks (F1 to F4) before starting the crude oil charging.

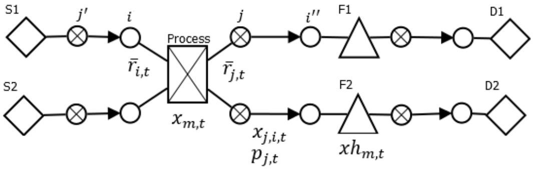
Figure 4.17: Crude oil refining scheduling flowsheet.



Source: Author (2021).

The network in Figure 4.17 is constructed in the unit-operation-port-state superstructure (UOPSS) formulation (KELLY, 2005). The UOPSS objects are detailed in the illustrative example in Figure 4.18 as: a) unit-operations m for sources and sinks (◇), tanks (△) and continuous-processes (⊠) and b) the connectivity involving arrows (→), inlet-ports i (○) and outlet-ports j (⊗). Unit-operations and arrows have binary and continuous variables (y and x, respectively) and the ports can hold the states as process yields or properties. The port-states j' and i'' represent upstream and downstream ports connected, respectively, to the in-port i and out-port j of a unit-operation m.

Figure 4.18: Illustrative example for the parameter feedback approach.



Source: Author (2021).

4.7.3 Online parameter feedback

The parameter updating of variables ($x_{m,t}$, $xh_{m,t}$, $x_{j,i,t}$, $p_{j,t}$) and hard bounds of yields ($\bar{r}^L_{i,t}$, $\bar{r}^U_{i,t}$, $\bar{r}^L_{j,t}$, $\bar{r}^U_{j,t}$) uses real, measured, or valid values defined as $y_{measured} = gain * y_{model} + bias$, considering gain and bias as the proportional and the linear deviations between the measured and model values. Two sets are defined for the data updating in the next scheduling cycle. For variables of unit-operation throughputs $x_{m,t}$, tank holdups $xh_{m,t}$, and parameters of inverse yields ($\bar{r}^L_{i,t}$, $\bar{r}^U_{i,t}$) and yields ($\bar{r}^L_{j,t}$ and $\bar{r}^U_{j,t}$), there is no need to create intermediate variables to update their ordinated values in the model since the flowsheet structure is identical (same rows and columns in the modeling, although with different coefficients). For this type of updating, gains and biases are added directly in their soft bounds (target) and hard bounds of unit-operations variables ($x_{m,t}$ and $xh_{m,t}$) and their yields, procedure named as offsetting.

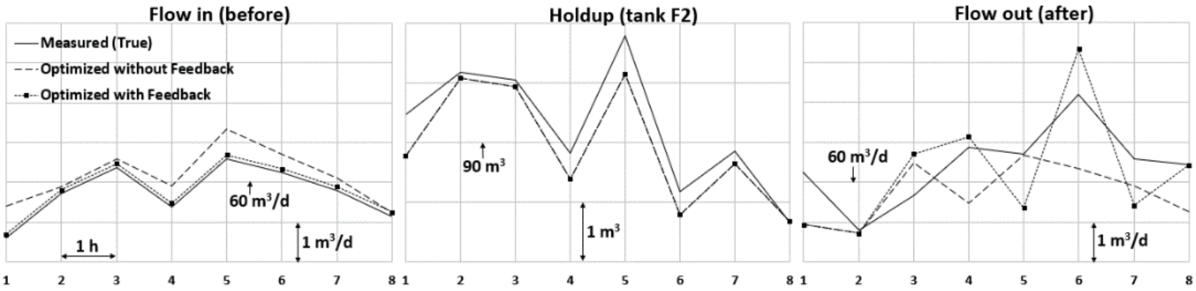
However, for flow of material and properties of unit-operation to unit-operation connections ($x_{j,i,t}$, $p_{j,t}$), there is a need to create intermediate variables to update their flowsheet propagation that overrides neighbor structures connected by arcs, paths, or arrows (→). Therefore, the total number of variables increases by the additional flows and properties overriding the connected objects for the re-scheduling step with parameter feedback, representing new columns in the re-scheduled model. In such way, the effects of all measurement updates propagate throughout the network.

By updating the overriding variables in the illustrative case (Figure 4.18), the number of continuous variables increases around 40%, hence for industrial cases this type of variable can be applied only in the initial time windows for less complex solving steps.

For each scheduling cycle before the parameter feedback updating, a simultaneous data reconciliation and parameter estimation is performed considering the raw data collected in the past time window. The examples use the reconciled data of the measured variables as targets and the gains and bias are feedback in the next re-scheduling model to be executed continuously.

Figure 4.19 shows the measured and the optimized values with and without the parameter feedback (both using reconciled data as target) for the illustrative case in Figure 4.18. A time horizon of 8 hours considers fixed setups of unit-operations and arrow flows, resulting in an NLP problem. The plots for flow-in, flow-out, and holdup of the tank F2 consider $60\ m^3/day$ and $90\ m^3$ as flow and holdup baseline, respectively.

Figure 4.19: Measured and optimized solutions for tank F2 (see Figure 4.18).



Source: Author (2021).

For the optimized case with feedback (--▪--), the flow after the tank F2 (on the right side) is the result of the effects of its flow-in (slightly) and holdup (significantly). The variations of tank flow-out delay one-time window to follow the new holdup value at each re-scheduling step and only the first time-window in the future of the optimized values are shown in Figure 4.19, therefore there are 8 scheduling cycles in this case.

In the industrial-sized example in Figure 4.17, the logistics MILP optimization for 5 days of time-horizon with 4 hours as time step (30 time periods) is solved in 86 seconds (with GUROBI 7.5.1) at 3.9% MILP gap. There are 12,568 constraints (2,545 equalities) for 5,777 continuous variables and 3,412 binary variables with 6,664 degrees-of-freedom. The quality NLP optimization is solved in 184 seconds (with

IMPL’s SLPQPE linked to GUROBI 7.5.1). There are 4,103 constraints (3,663 equalities) for 3,821 continuous variables with 278 degrees-of-freedom.

4.7.4 Conclusions

Efficient scheduling solutions in a near online fashion are becoming reality by the virtue of all advances in decision-making modeling, solving algorithms, and computer-aided resources in terms of faster CPU clock speeds and memory. The major challenge now is to integrate proper or correct data (in timeliness and quality) to the decision automation core. An online scheduling engine with parameter updating might be useful to cope with uncertainties and to reduce the offsets or inaccuracies over the life-time of the problem as an effective way to close the gap among predictions and productions.

4.8 General conclusions on the crude oil refinery scheduling optimization

The crude oil refinery scheduling optimization is a complex and challenging problem due to the high number of continuous and binary variables, in addition to nonlinear and non-convex terms, which results in a non-convex large-scale MINLP formulation. Aiming to study and contribute to the state-of-the-art on the chemical engineering optimization literature, the research topics addressed in this chapter focus on handling complex formulations typically found in the chemical engineering industry through the implementation and development of modeling and optimization approaches, decomposition and heuristic strategies, and machine learning techniques for chemical processes with an emphasis on crude oil refinery scheduling applications. This includes: a) Mathematical modeling of large-scale discrete-time crude oil refinery scheduling problems using the UOPSS representation; b) Phenomenological decomposition of the quantity-logic-quality phenomena, which addresses an iterative two-step solving procedure of MINLP formulations as sequential MILP and NLP sub-models; c) Efficient process design regarding both the mathematical formulation and the operations in the plant; d) Linearization strategies to approximate nonlinear blending terms in a linear programming model, in which the factor-flows of qualities are modeled explicitly as slack or surplus variables; e) Exclusions employed to reduce the scale of the optimization search space in constructive rolling horizon strategies; f) Relaxations in MILP models by using relax-and-fix iterations; g) Parameter feedback to minimize plant-model mismatches, to handle process uncertainties and disturbances, and to improve the reliability and accuracy of the scheduling solution and implementation.

The crude oil refinery scheduling problems addressed are formulated using the unit-operation-port-state superstructure and the quantity-logic-quality phenomena (QLQP) concepts. That allows the modeling and solving of complex-scope industrial-sized scheduling problems using a discrete-time formulation. A phenomenological decomposition heuristic is applied to handle the complex MINLP refinery scheduling formulations by breaking down the problem to significantly reduce the computational burden within optimization approaches.

A proper design plays a key role on industrial operations, including the methodology used to build the mathematical formulation to be solved, as well as the process operations in the real plant. On one hand, the mathematical formulation is expected to accurately represent the real process, especially aiming to mitigate plant-model mismatches over time. On the other hand, the process design directly impacts the economic, technical, and operational conditions. For improved operations, especially when addressing online strategies toward smart processing with real-time feedback from the plant, both an improved blend design and a complex process design are the recommendable networks to be constructed, modeled, and solved, aiming to achieve improved solutions in terms of economic value and production flexibility. The blending formulation considering a continuous blender unit instead of batch mixtures (without the blender), and the processing formulation considering a complex cascaded distillation network instead of a simplified one-tower network, are examples of improved designs that provide more accurate predictions, production flexibility, and increased economic value for the process. This is especially beneficial when the formulation simultaneously includes the blending and the processing improved designs.

Modeling strategies such as the linearization of blending constraints within linear problems improves the mathematical formulation by considering additional information for more accurate predictions, which results in better optimized solutions. This strategy includes proxied information on the qualities of streams by using a linear programming factor reformulation for the blending operations of crude oil. The benefits include improved solutions, with better economic value, and better convergence within the optimization procedure. As the formulation remains unchanged, except for the blending constraints, the computational effort increase is expected not to be a limiting factor for the utilization of this method.

Modeling, solving, and heuristic strategies are employed for handling complex industrial-sized refinery scheduling problems within a discrete-time formulation, including decompositions to reduce the optimization search space in constructive rolling horizon strategies, and relaxations on mixed-integer linear programming problems to construct the problem by an ad-hoc relax-and-fix approach. Both heuristics imply in an expected slight reduction in the objective function, but with great benefits in terms of reduced computational effort. Moreover, this reduced effort is expected to scale with the size of the problem; therefore, it might be especially useful for large-scale applications.

Overall, the formulation proposed is coherent with large-scale industrial applications in terms of operational constraints, refinery economics, and problem complexity and size. The results indicate that complex non-convex MINLP refinery scheduling formulations can be efficiently solved by utilizing decomposition, heuristic, and machine learning strategies, which potentially provides improved modeling and optimization capabilities for real industrial applications.

4.9 References

CASTILLO, P. A.; MAHALEC, V. Inventory Pinch Based Multi-Scale Model for Refinery Production Planning. Computer Aided Chemical Engineering, v. 33, p. 283, 2014.

CASTRO, P.; GROSSMANN, I. E. Global Optimal Scheduling of Crude Oil Blending Operations with RTN Continuous-time and Multiparametric Disaggregation. Industrial & Engineering Chemistry Research, 53, 15127, 2014.

FRANZOI, R. E.; MENEZES, B. C.; KELLY, J. D.; GUT, J. W. Effective Scheduling of Complex Process-Shops Using Online Parameter Feedback in Crude oil Refineries. Computer Aided Chemical Engineering, v. 44, p. 1279-1284, 2018a.

FRANZOI, R. E.; MENEZES, B. C.; KELLY, J. D.; GUT, J. W. Blend Scheduling Optimization Using Factors for Qualities in Cascaded Distillation Towers in Crude oil Refineries. Blucher Chem. Eng. Proceedings, v. 1, p. 1233-1236, 2018b.

GUPTA, D.; MARAVELIAS, C. T. On Deterministic Online Scheduling: Major Considerations, Paradoxes and Remedies. Computers & Chemical Engineering, v. 94, p. 312-330, 2016.

GUPTA, D.; MARAVELIAS, C. T.; WASSICK, J. M. From Rescheduling to Online Scheduling. Chemical Engineering Research and Design, v. 116, p. 83-97, 2016.

GUPTA, D.; MARAVELIAS, C. T. A General State-Space Formulation for Online Scheduling. Processes, v. 5, n. 4, p. 69, 2017a.

GUPTA, D.; MARAVELIAS, C. T. On the Design of an Online Scheduling Algorithm. AICHE Meeting, Minneapolis, MN, 2017b.

JIA, Z.; IERAPETRITOU, M.; KELLY, J. D. Refinery Short-Term Scheduling Using Continuous Time Formulation: Crude oil Operations. Industrial & Engineering Chemistry Research, v. 42, p. 3085, 2003.

KELLY, J. D. The Unit-Operation-Stock Superstructure (UOSS) and the Quantity-Logic-Quality Paradigm (QLQP) for Production Scheduling in The Process Industries. In Multidisciplinary International Scheduling Conference Proceedings: New York, United States, p. 327-333, 2005.

KELLY, J. D. IMPL Manual, www.industrialgorithms.ca, 2018.

KELLY, J. D.; MENEZES, B. C.; ENGINEER, F.; GROSSMANN, I. E. Crude oil Blend Scheduling Optimization of an Industrial-Sized Refinery: a Discrete-Time Benchmark. In Foundations of Computer Aided Process Operations, FOCAPO, Tucson, AR, United States, 10-13 January, 2017a.

KELLY, J. D.; MENEZES, B. C.; GROSSMANN, I. E. Successive LP Approximation for Non-Convex Blending in MILP Scheduling Optimization Using Factors for Qualities in the Process Industry. Industrial & Engineering Chemistry Research, v. 57, n. 32, p. 11076, 2018.

KELLY, J. D.; MENEZES, B. C.; GROSSMANN, I. E.; ENGINEER, F. Feedstock storage assignment in process industry quality problems. In Foundations of Computer Aided Process Operations, FOCAPO, Tucson, AR, United States, 10-13 January, 2017b.

KELLY, J. D.; ZYNGIER, D. Continuously improve the performance of planning and scheduling models with parameter feedback. In Foundations of Computer Aided Process Operations, FOCAPO, Boston, MA, United States, 29-2 July, 2008.

KELLY, J. D.; ZYNGIER, D. Unit-operation Nonlinear Modeling for Planning and Scheduling Applications. Optimization and Engineering, v. 18, n. 1, p. 133–154, 2017.

LEE, H.; PINTO, J. M.; GROSSMANN, I. E.; PARK, S. Mixed-integer Linear Programming Model for Refinery Short-Term Scheduling of Crude Oil Unloading with Inventory Management. Industrial & Engineering Chemistry Research, v. 35, p. 1630, 1996.

MENEZES, B. C.; KELLY, J. D.; GROSSMANN, I. E. Improved Swing-Cut Modeling for Planning and Scheduling of Oil-Refinery Distillation Units. Industrial & Engineering Chemistry Research, v. 52, p. 18324, 2013.

MENEZES, B. C.; KELLY, J. D.; GROSSMANN, I. E. Phenomenological decomposition heuristic for process design synthesis of oil-refinery Units. Computer Aided Chemical Engineering, v. 37, p. 1877-1882, 2015.

MISENER, R.; FLOUDAS, C. A. Advances for the Pooling Problem: Modeling, Global Optimization, and Computational Studies. Appl Comput Math, v. 8, n. 1, p. 3, 2009.

SONG, Y.; MENEZES, B. C.; KELLY, J. D.; GARCIA-HERREROS, P.; GROSSMANN, I. E. Scheduling and Feed Quality Optimization of Concentrate Raw Materials in the Copper Refining Industry. Industrial & Engineering Chemistry Research, v. 57, p. 11686, 2018.

SUBRAMANIAN, K.; MARAVELIAS, C. T.; RAWLINGS, J. B. A state-space model for chemical production scheduling. Computers & chemical engineering, v. 47, p. 97-110, 2012.

ZYNGIER, D., KELLY, J. D. Multi-product inventory logistics modeling in the process industries. In: Wanpracha Chaovalitwongse, Kevin C. Furman, Panos M. Pardalos (Eds.) Optimization and logistics challenges in the enterprise. Springer optimization and its applications, p. 61-95, 2009.

5

A Closed-Loop Rescheduling Framework for Continuous Nonlinear Processes with Disturbances[4]

At any production manufacturing site, scheduling decision-making is often calculated and implemented using unreliable or inaccurate data from the process, therefore infeasibilities and inconsistencies are expected. For improved industrial operations, it is fundamental to minimize the plant-model mismatches through a re-optimization or moving horizon strategy, in which the current state of the system is continuously updated. The online closed-loop approach proposed herein is based on a systematic bi-layer rescheduling framework that simulates the closed-loop scheduling for continuous nonlinear processes within a moving horizon approach. An MINLP blend scheduling problem is addressed, and distinct types of disturbances are introduced. The proposed framework simulates the entire closed-loop scheduling solution, effectively handles the triggered disturbances, reduces inaccuracies and plant-model mismatches by maintaining the state of the system updated, and provides a systematic fashion to improve the scheduling implementation.

5.1 Introduction

Although real-world process industry operations are highly dynamic and fluid environments in which data is uncertain and changes over time, online updates of information in process operations are usually delayed (in terms of timeliness) or incomplete (in their structural and temporal integration) since they are complicated to be synchronized and to be implemented algorithmically in a systematic way (LARSEN and PRANZO, 2018). However, recent advances in network (flowsheet) optimization (BRUNAUD et al., 2020) as well as in solving algorithms and computer-aided

[4] This chapter is based on the following manuscript:
FRANZOI, R. E.; MENEZES, B. C.; KELLY, J. D.; GUT, J. A. A Closed-Loop Rescheduling Framework for Continuous Nonlinear Processes with Disturbances. Computers & Chemical Engineering, 2020. Under review.

resources (FRANZOI et al., 2018) allow more complex decision-making solutions in scheduling propositions. Moreover, the expansion of information and communication technologies can provide an online data measurement of complete process networks for the massive volume and variety of information from the plant (MENEZES, KELLY, and LEAL, 2019). In this direction, this paper addresses the design of an online scheduling framework based on a dynamic or moving horizon rescheduling approach for a typical chemical manufacturing process considering: a) data uncertainty and disturbances; b) rescheduling cycles to update the state of the modeled system and to properly handle the effects of disturbances throughout the network; and c) features for open-loop and closed-loop solutions, the former without integration between the systematic solutions, and the latter generated by sequentially rescheduling the problem using variable feedback (see KELLY and ZYNGIER, 2008).

An effective scheduling optimization is challenging due to the complexities of data updating, modeling re-building or re-formulating of sets, parameters, variables, constraints, and derivatives, as well as solving NP-hard mixed-integer problems fast enough to be integrated within a discrete-time step of hours or less. If the process data used for scheduling is outdated or not properly integrated with the production, there may be inconsistencies in the prediction of amounts and properties of intermediate and final products. Moreover, when data is uncertain (which is common in all industrial fields) it becomes more difficult to reliably and accurately model operational problems and to anticipate future outcomes. In the chemical engineering field, some of the most common uncertainties are related to untracked feed quality (compositions, properties) before the processing, arrival of new information not updated in the current scheduling determinations, programmed and executed deviations from both manual and automated procedures in the process, mis-measurements in resources flows, operational conditions, etc. Exogenous factors such as product demand changes (regarding amount and due date), market fluctuations (price volatility, spot contracts), and feedstock information (date of arrival, amounts, and properties) are likely to happen in the day-to-day production as well. According to Larsen and Pranzo (2019), although dynamic environments and uncertainties are frequent in real-world operations, new information from the production is usually neglected. Furthermore, there might be many unexpected events in real production operations which can invalidate the original schedules so that a re-optimization is essential for smoothing eventual negative impacts (KATRAGJINI, VALLADA, and RUIZ, 2013). From Gupta and Maravelias (2016), this re-optimization consists basically in revising and updating the actual schedule due to disruptions or unexpected events, what is known as reactive scheduling or rescheduling. Although arrival of new information and disruptions in the process may lead to suboptimality and infeasibilities in the incumbent schedule, what by itself motivates the importance of rescheduling (SUBRAMANIAN, MARAVELIAS, and RAWLINGS, 2012), it should be performed not only when necessary but rather on a regular and systematic basis to account for new information in the optimization stage. This potentially contributes to reduced costs and improved economics (GUPTA and MARAVELIAS, 2016), and makes rescheduling a proactive mechanism to be widespread in industry. Similarly, Zhuge and Ierapetritou (2012) argue that even though unexpected events are usually seen negatively, they may not always be unfavorable, and rescheduling should be performed on a regular basis to exploit favorable disturbances instead of rejecting them to maintain the actual schedule (interestingly, there is a similar effect in model predictive control). Therefore, when new data is integrated to the scheduling, re-optimizing the problem provides a continuous and institutionalized cycle of improvement that reduces the deviation between model-based and data-based scheduling solutions and real plant values. Such continuous data-model updating and re-determination of new schedules mitigates or at least reduces the negative effects of uncertainties and disturbances, as well as benefits from any favorable changes in the process.

The rescheduling theory and applications have been emphasized in many works over the past decades (COTT and MACCHIETTO, 1989; RODRIGUES et al., 1996; MCKAY and WIERS, 1999; VIEIRA, HERRMANN, and LIN, 2003; LI and IERAPETRITOU, 2008; KATRAGJINI, VALLADA, and RUIZ, 2013), although rescheduling is still typically formulated as a static open-loop problem in which the main goal is to restore feasibility or optimality. Conversely, when the schedule is revised (and a reschedule is then performed) in a consistent and regular basis, it leads to a moving horizon approach in which sequential open-loop or feedforward solutions might be implemented to generate a closed-loop solution with embedded variable feedback and in which parameter feedback (such as bias updating) may be added as required (KELLY and ZYNGIER, 2008). This can be part of a systematic framework based on an online scheduling algorithm able to handle different types of disturbances in the production system such as a failure or malfunction of a unit or disruption in any link or transfer in the production-chain. Gupta and Maravelias (2016) show that the open- and closed-loop scheduling are distinct problems even when no uncertainties are considered, since in the closed-loop strategy the data is updated in a new model to be re-solved and re-iterated into a new forward rolling rescheduling. Recent studies on closed-loop strategies (GUPTA, MARAVELIAS, and WASSICK, 2016; GUPTA and MARAVELIAS, 2016), formulations (GUPTA and MARAVELIAS, 2017a), and design (GUPTA and MARAVELIAS, 2017b) address the importance of rescheduling in chemical processes. However, additional features such as rescheduling frequency and triggering, approaches to handle uncertainties and disturbances, impact of time horizon length in the simulation/optimization, are still imperative.

The online scheduling framework thus proposed herein is based on a dynamic rescheduling and moving horizon approach and can effectively handle disturbances (either to mitigate/reduce their effects or to take advantage of new information), maintain the state of the system updated, reduce plant-model mismatches, achieve a more accurate and reliable representation of the real process, and provide a systematic fashion to improve the performance of the scheduling implementation in the process. The novelty of this approach relies on developing a systematic bi-layer framework to simulate the closed-loop scheduling solution for continuous nonlinear processes within a moving horizon fashion, in which noises, disturbances, disruptions, and other unforeseen events are assumed to happen with respective triggering probabilities. The framework is employed to test multiple scenarios in order to investigate the impact of the disturbances in the closed-loop scheduling regarding both the economic value and the scheduling operations. Several common types of disturbance are considered, and further analyses on the issues caused by neglecting them are addressed as well. The proposed methodology is applied to a mixed-integer nonlinear programming (MINLP) problem, in which a hierarchical phenomenological decomposition is used (see MENEZES, KELLY, and GROSSMANN, 2015).

The outline of this chapter is as follows. Section 5.2 presents an overview of the online scheduling approaches reported in the literature. The importance of the rescheduling and closed-loop strategies, including a motivating example, is highlighted in Section 5.3. The problem statement is described in Section 5.4 and its mathematical formulation is given in the Supplementary Material. Examples using the proposed framework, including their respective results, are discussed in Section 5.5. The conclusions are highlighted in Section 5.6.

5.2 Literature Review on Online Scheduling (Rescheduling)

In the past decades, several works addressed increasingly more complex rescheduling topics, mostly related to uncertainties in the process, rescheduling features, and open-loop and closed-loop solutions. Kanakamedala, Reklaitis, and Venkatasubramanian (1994) presented a reactive scheduling problem in multipurpose batch plants in which there are unexpected deviations for unit availabilities and processing times, and employed a bi-level least impact heuristic to reschedule the model. Huercio, Espuna, and Puigjaner (1995) used a heuristic-based rescheduling algorithm to cope with real time disturbances through task start time shifting units' reassignment. Honkomp, Mockus, and Reklaitis (1999) proposed a framework to validate and evaluate rescheduling strategies under uncertainties such as processing time variations and unit breakdowns. Vin and Ierapetritou (2000) rescheduled a multiproduct batch plant problem within a continuous time formulation considering as disturbances machine breakdowns and rush order arrivals. Vin and Ierapetritou (2001) developed a strategy to improve the scheduling performance and flexibility under occurrence of unexpected events. Méndez and Cerdá (2003) introduced an MILP formulation for a multiproduct batch plant in which rescheduling is performed either under occurrence of unexpected events or to improve a non-optimal schedule. Besides, multiple rescheduling operations, such as reallocation, resequencing, and reordering, could be performed simultaneously. Janak et al. (2006) developed a rescheduling framework to partially reschedule the problem by determining which tasks have been affected or not under unforeseen events such as unit breakdowns and alteration of orders. Adhitya, Srinivasan, and Karimi (2007) proposed a framework to cope with supply chain disruptions in which the procedure of continuously and frequently rescheduling the problem leads to a closed-loop solution.

Kelly and Zyngier (2008) studied a production-chain problem involving a simple reactor and a tank flowsheet in which the inventory gap between the modeled and the real tank could be reduced to zero offset by performing what they called parameter feedback (gain and/or bias updating). According to these authors, differences between the planned (model) and the real (measured) values in the process do not necessarily result from task execution or operational errors, since the absence of parameter feedback data makes it impossible to distinguish between an inadequate model representation and implementation failure of the problem result. Katragjini, Vallada, and Ruiz (2010) employed three different types of disruptions in a flowshop scheduling problem (arrival of new jobs, machine breakdowns, and release time delays) and developed rescheduling algorithms to find better trade-offs between the quality and the stability of the schedule. Zhuge and Ierapetritou (2012) introduced a closed-loop strategy for the simultaneous integration of scheduling and control, and highlighted the importance of rescheduling not only to handle negative scenarios but also to exploit favorable disturbances in the process. Nie et al. (2014) developed an MILP formulation based on a discrete-time resource task network (RTN) to reschedule a mixed batch/continuous process, which is reformulated in a state-space form and used for continuous rescheduling under process disruption. Kopanos and Postikopoulos (2014) presented a rescheduling approach based on state-space representation, moving horizon framework and multiparametric programming techniques. Lindholm and Nytzén (2014) introduced a bi-level hierarchical approach for a production scheduling problem considering disturbances in the supply of utilities. Du et al. (2015) proposed a time scale bridging framework to integrate closed-loop scheduling and nonlinear control of continuous processes. Gupta, Maravelias, and Wassick (2016) approached rescheduling as an online problem (online scheduling) and showed its benefits even when no disturbances/trigger events occur; therefore, reschedule should be applied whenever new information is available.

Gupta and Maravelias (2016) addressed features of open- and closed-loop scheduling and presented a framework to analyze closed-loop schedules. From an online scheduling perspective, Gupta and Maravelias (2017a) introduced a generic state-space model formulation to routinely handle disturbances and to apply their respective counter-decisions to update the state of the system using parameter feedback. Looking for more efficient closed-loop implementation to properly integrate planning, scheduling and control, Charitopoulos, Papageorgiou, and Dua (2019) introduced a framework based on a rigorous rescheduling mechanism to provide online solutions under dynamic disturbances, which mitigates their impact on operational decisions of planning and scheduling. Larsen and Pranzo (2019) proposed a generic framework for dynamic scheduling problems in which a solver-simulator-controller approach is employed to a job shop scheduling problem to evaluate when uncertainties become relevant, rescheduling triggering and frequency, and solution quality. Stevenson, Fukasawa, and Ricardez-sandoval (2020) evaluated periodic rescheduling policies using a rolling horizon framework in an industrial-scale multipurpose plant to investigate the effect of plant parameters on the plant performance, and highlighted the importance of addressing rescheduling strategies for industrial applications.

Additionally, Franzoi et al. (2018) addressed an industrial-sized crude oil refinery scheduling problem and introduced an online parameter feedback with data reconciliation integrated within the scheduling cycle to achieve better process determinations. The authors highlighted the importance of properly integrating data to the decision automation core to reduce inaccuracies, to handle uncertainties, and to reduce the gap between optimized determinations and productions. For better performance of industrial operations, high-quality predictive analytics based on near past data (validated, reconciled, estimated, etc.) can determine improved near future predictions, which may be used in decision-making (prescriptive analytics) to correct processes mismatches. This framework relies on better diagnostics to improve predictive analytics (by identifying and correcting mismatches) and prescriptive analytics (by handling inconsistencies and infeasibilities) (MENEZES et al., 2019).

For improved industrial scheduling operations, it is fundamental to properly formulate and optimize the problem. That includes minimizing the plant-model mismatches so that the optimal solution matches the process conditions. Because of the high nonlinear and uncertain nature of most industrial problems, unforeseen events are likely to happen constantly and repeatedly throughout the entire network. That motivates for a continuous optimization cycle, in which the current state of the system is updated, and re-optimizations (rescheduling) are performed. The literature on the topic has been increasingly discussed the importance of rescheduling for process operations, although there are still open questions to be addressed, mostly related to open-loop versus closed-loop methods, rescheduling algorithm or framework, tuning of rescheduling elements or parameters, impacts of disturbances and uncertainties on the scheduling operations, etc. Some of these topics will be addressed and discussed as follows.

5.3 Problem Statement

Three main rescheduling topics are addressed herein: a) the framework and the elements of the rescheduling approach; b) the occurrence of disturbances in the process network, their potential impact, and importance of including them in the formulation; and c) open-loop versus closed-loop strategies in the scheduling optimization. In this section, the importance of re-calculation cycles as well as their contribution towards a better scheduling optimization are highlighted.

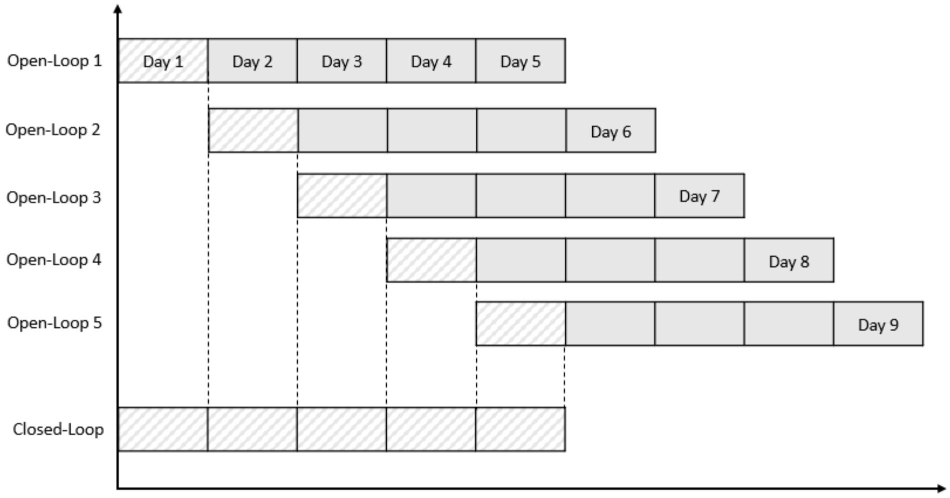
Due to differences between the data used in the scheduling model and the actual data from the process, there are inconsistencies between what would theoretically be produced (predictions and prescriptions) and what is in fact processed (production). These disturbances might arise from the arrival of new information, operational deviations (prescriptions versus productions) from manual and automatic procedures, equipment failures and malfunctions, uncertainty on data (mainly information mismatch), etc. More specifically, common examples of disturbances in chemical engineering encompass: a) product demand changes (amount quantities and/or release, and due dates); b) non-updated and untracked or untraced feedstock information (date of arrival, amounts, and properties); c) uncertainty in the flows and properties throughout the process network; d) breakdowns, malfunctions or unplanned maintenance in process units, storage vessels (tanks), and in their connections; e) uncertainty in the raw material (inlets) to product (outlets) yields and properties in complex units (i.e., distillation column). Some of these disturbances are discussed later in this section. In the following, a motivating example illustrates the main concept of rescheduling and underscores the overall benefits that this type of approach brings to the final scheduling solution.

5.3.1 Motivating Example

Let us consider a moving horizon scheduling optimization to be performed daily within the future 5 days, so that the first optimization is carried out from Day 1 to Day 5 and uses the available past, present, and future information given for that respective time horizon. After the schedule for the first day is implemented, the second optimization is carried out (from Day 2 to Day 6), and new information might be available for this future 5 day-horizon. Therefore, there is a new model to be updated for rescheduling of the next step or cycle. For instance, this new information could be related to a unit breakdown that may cause a future infeasible schedule. It could also be a new future demand order, for Day 6, which was not considered in the first cycle of the scheduling optimization. Market fluctuations may impact the new scheduled program so that the current solution defined in Day 1 does not necessarily meet the product demands and specifications. If the actual schedule becomes infeasible with the new scenario (regarding the production, demand fulfillment, raw material quality, etc.), rescheduling the problem may find a new feasible solution; and even if the schedule is still feasible, a re-optimization may find a distinct and updated optimal solution potentially better than the previous one. In any case, the sooner the new information is included into the modeling and solving, the better in terms of maintaining and/or restoring feasibility or optimality.

By continuously rescheduling the problem, the solution is adapted from an open-loop to a closed-loop fashion. Specifically, optimizing an open-loop schedule for the future five days requires only one optimization at the beginning of the time horizon; however, optimizing the problem in a closed-loop strategy requires a sequence of optimization runs. In the example shown in Figure 5.1, the problem is daily optimized for the future 5 days using time steps of one day. Therefore, there are sequential open-loop solutions, and the closed-loop solution is achieved by using the beginning (first day) of each open-loop solution. By definition, a closed-loop solution is understood as an arrangement of the initial or publishable part of the sequential open-loop solutions through a moving horizon strategy, as shown in Figure 5.1. Similarly, to model predictive control, there are both a prediction horizon for the outputs and a control horizon for the inputs where the initial part of the open-loop schedule is the control horizon.

Figure 5.1: Moving horizon closed-loop scheduling.



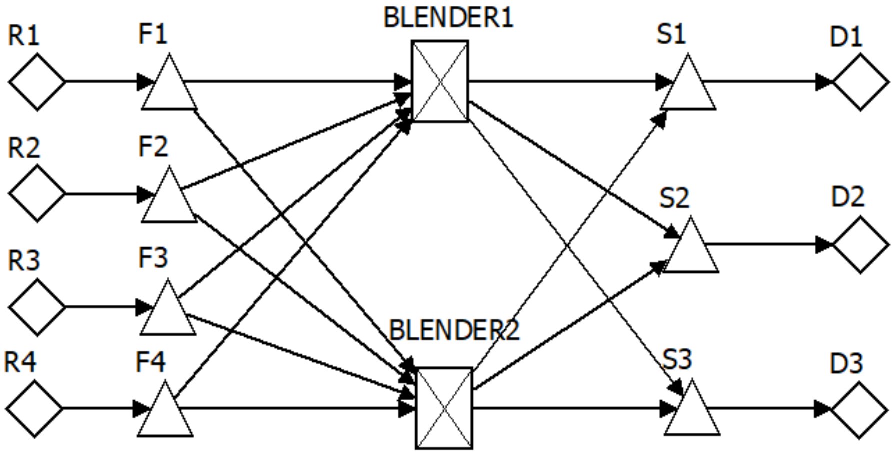
Source: Author (2021).

From the motivating example, the main features of the closed-loop approach require some sophisticated capabilities to: a) periodically or regularly re-optimize the schedule; b) properly integrate operations vis the publishable part of the open-loop schedule; and c) update new information in the model considering process variables (yields, conversion rates, opening inventories, etc.), new process network structures, among others (referred to as variable feedback in Kelly and Zyngier, 2008). Without these closed-loop features implemented into an online fashion, the rescheduling may be unrealistic since the inherent uncertainties and unforeseen disturbances that progressively change the incumbent conditions must be continuously considered to match the ongoing production scenario and context.

5.3.2 Blend Scheduling Optimization Example

To further illustrate the main concepts and advantages of rescheduling approaches, a case study is proposed over a typical blend scheduling optimization problem to investigate the impact of disturbances in the rescheduling modeling and solving, as well as to explain and to discuss the mechanisms of how this framework handles the disturbances and systematically generates a new optimal solution. For each scenario proposed to test our approach, a closed-loop solution is achieved by the sequential integration of the open-loop solutions (similarly as shown in Figure 5.1). In the blend scheduling optimization problem shown in Figure 5.2, there are four feedstocks or raw materials R1 to R4 with distinct qualities (the properties considered are the specific gravity and sulfur content), which are connected to four feed tanks F1 to F4, and sequentially to the mixers BLENDER and BLENDER2. The blended material is sent to the storage tanks S1 to S3 to be stocked in final pools to meet the demand for products D1 to D3. This is a typical example in many distinct types of industry. This could be a problem in which a blend of feedstocks must be prepared for further processing (e.g., preparing the crude oil blend to feed the distillation unit in refinery operations), or when the intermediate products must be blended to ensure the final product to meet specifications in terms of properties or quality control (e.g., mix distinct intermediate streams of diesel to produce a final stream with required specifications in terms of specific gravity and sulfur content).

Figure 5.2: Blend scheduling problem flowsheet.



Source: Author (2021).

The data for this problem are presented in Table 5.1, including the properties of each feedstock R1 to R4, the specifications for each product D1 to D3 (i.e., maximum specific gravity and maximum sulfur content allowed), and their respective market costs or prices.

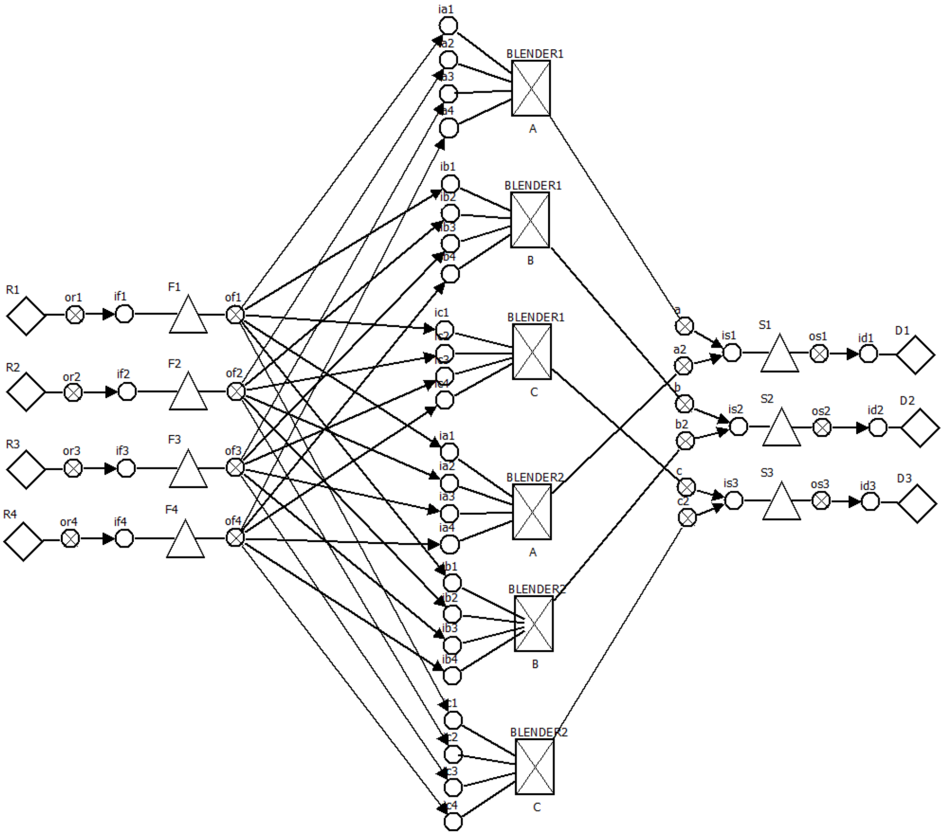
Table 5.1: Data for the blend scheduling optimization problem.

Unit	R1	R2	R3	R4	D1	D2	D3
Specific Gravity (g/mL)	0.80	0.85	0.95	1.00	0.85	0.90	0.95
Sulfur Content Specification (g/g)	0.90	1.00	1.25	1.50	1.05	1.20	1.40
Value (k$/bbl)	-23	-20	-17	-15	80	75	70

Source: Author (2021).

As there are three distinct products that may be produced in each blender unit, operating modes or grades (A, B, and C) are created. Thus, the problem flowsheet is extended as shown in Figure 5.3 where a set of multi-use or unit-commitment constraints are included to avoid simultaneous operation of different modes for each blender unit each time step or time period.

Figure 5.3: Blend scheduling problem flowsheet using the UOPSS formulation.

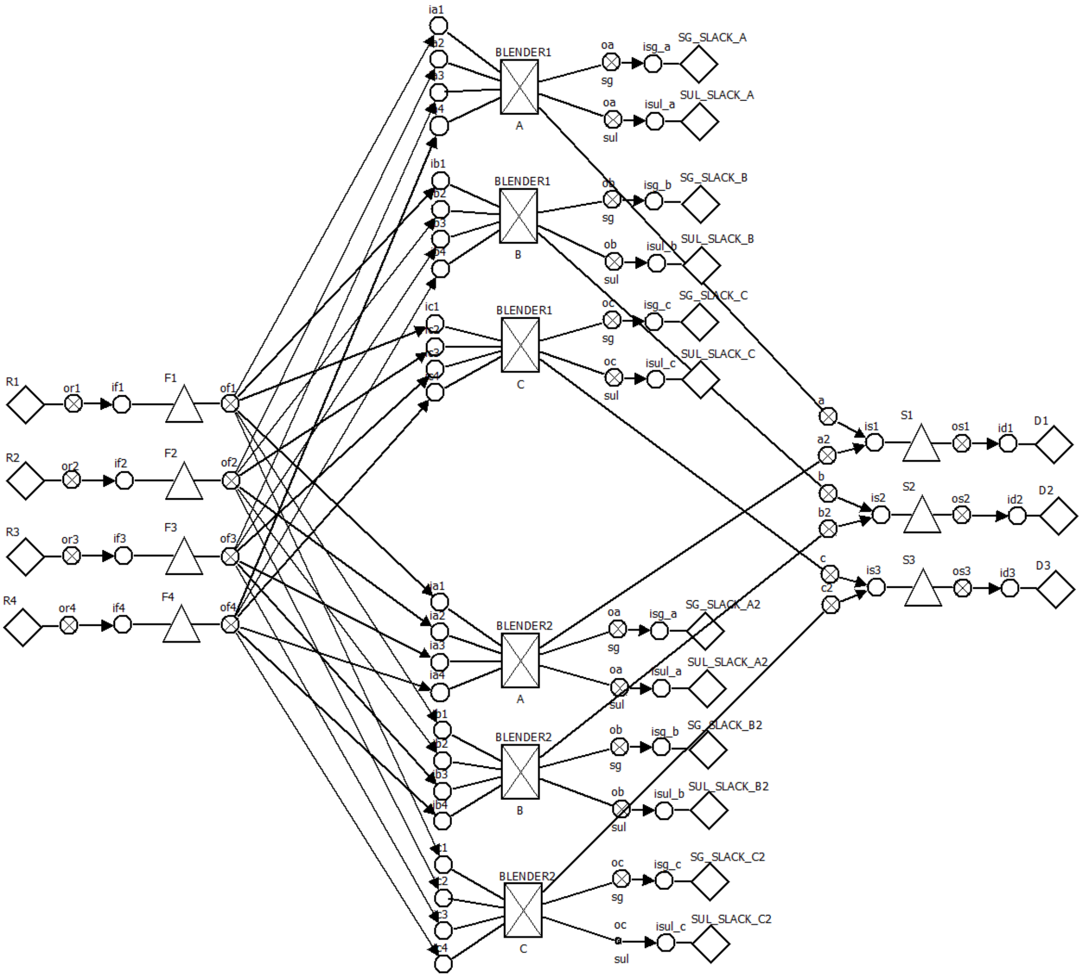


Source: Author (2021).

The flowsheet shown in Figure 5.3 is based on the UOPSS (Unit-Operation-Port-State Superstructure) formulation (KELLY, 2005) and considers six types of UOPSS objects. There are unit-operations m for tanks (△), sources and sinks (◇), and continuous-processes (☒), which are associated with both continuous and binary variables (x and y); and there are connections involving arrows (→), inlet-port-states i (○) and outlet-port-states j (⊗), also associated with both continuous and binary variables (x and y). The capital letters A, B, and C immediately below the blenders (☒) represent their operational modes or grades. Binary variables account for the operational modes as well as connection lineups, routes or paths, and the quality blending equations (for specific gravity and sulfur content) impose both nonlinearities and non-convexities, yielding a non-convex MINLP or what we call a qualogistics problem, as it involves quantity, logic (logistics), and quality phenomenological variables and constraints.

To handle such a complex formulation, the MILP-NLP phenomenological decomposition heuristic addressed by Menezes, Kelly, and Grossmann (2015) and applied in Kelly, Menezes, and Grossmann (2018) for a similar blend-shop, is employed to solve sequential sub-problems in an iterative, sequential, or successive fashion. Each iteration consists in: a) neglecting the quality information in the original MINLP model, and optimizing the resulting mixed-integer linear problem; and b) fixing all the binary variables from the mixed-integer linear problem solution in the original MINLP (i.e., setting the binaries to be either 0 or 1 based on their respective values), and optimizing the resulting nonlinear problem. Although this phenomenological decomposition is an intuitive and straightforward method to tackle the proposed problem, it may result in an undesirable MILP-NLP gap. In addition, the high level of nonlinearities and non-convexities in this type of problem may hinder the NLP optimization to find better optimal solutions (i.e., higher probability to converge to a local optimum or even an infeasible solution). To handle such solution complexities, two additional strategies have been used to improve the performance of the modeling and solving stages within the proposed approach, leading to better convergence (optimization) and results (final solution). First, multiple NLP optimizations are performed in each iteration by generating different starting points for the variables in the modeling stage, which (given the highly nonlinear and non-convex nature of the model) typically leads to different local optimal solutions. The starting points generation is performed automatically by the standard modeling found in the industrial modeling and solving platform (IMPL, Industrial Modeling and Programming Language). Second, the LP approximation for non-convex NLP blending equations proposed by Kelly, Menezes, and Grossmann (2018) and referred to as *factors* is employed, in which slack and surplus variables, as well as linearized quality blending information, are included in the MILP model. Hence, the flowsheet is updated to account for the hypothetical flows related to the slack/surplus variables, in which there are two flows connecting each mode of each blender to the hypothetical perimeters, related to the specific gravity and sulfur content properties, as shown in Figure 5.4.

Figure 5.4: Blend scheduling with factors problem flowsheet using the UOPSS formulation.



Source: Author (2021).

5.3.3 Rescheduling, Open-loop, and Closed-loop Strategies

There are two approaches addressed herein that can be employed to optimize the blend scheduling problem presented in Figure 5.4. An open-loop strategy refers to a single optimization for the entire time horizon. Disturbances cannot be considered in the formulation because rescheduling is not performed, and plant-model mismatches are expected to happen. Moreover, the optimization tends to become suboptimal because eventual unforeseen events and infeasibilities are likely to happen. On the other hand, a closed-loop approach can be employed to sequentially reschedule the incumbent model in a moving horizon fashion. In this strategy, a frequent and predetermined rescheduling optimization handles the occurrence of noises, disturbances, and unforeseen or unexpected events, and takes advantage of additional information (that was not available in the previous optimization). Differences between the formulation (model values) and the plant (real values) can be mitigated, so that a more accurate and realistic process optimization is carried out. Both the formulation and optimization are more reliable in terms of considering the current (real) state of the system. Hence, an easier and smoother implementation of the optimal scheduling solution in the plant is expected.

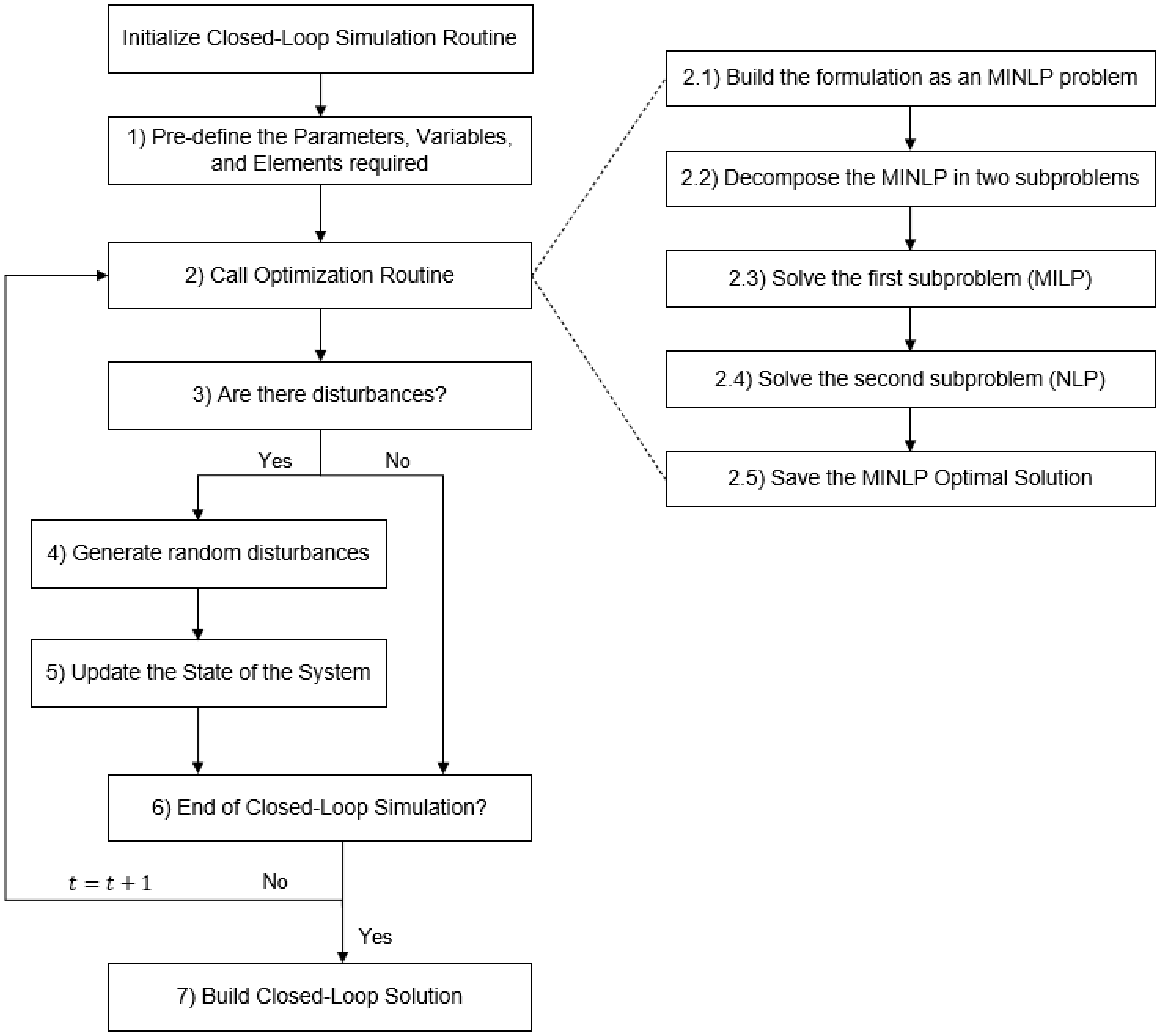
Due to the several advantages of closed-loop approaches, a closed-loop rescheduling framework is proposed herein. Prior to each optimization, the state of the system is updated so as to consider all available information as well as any impacts or changes related to noises, disturbances, disruptions, and unforeseen events. Within a moving horizon fashion, the rescheduling are basically re-optimizations for the future 10-days using a discrete-time formulation. Disturbances are introduced in the formulation in order to simulate a more realistic process system subjected to omnipresent uncertainties. The distinct types of disturbance assumed to happen are predefined in the framework and are randomly generated upon specific triggering probabilities. The framework properly and consistently handles each triggered disturbance by rescheduling the problem to find a new optimal solution. For a real process, the closed-loop strategy is expected to be continuously performed, but for the matter of simulation purposes of the re-optimizations, a limited simulated time horizon of 40-days is defined. This time horizon length is expected to be large enough to avoid or mitigate interferences caused by the end-effects of the simulated horizon. A time step $ts = 24$

h is used, so that the formulation and the optimization have a total of 10 time steps and 40 optimized solutions. Therefore, there are 40 iterations in the simulation of the moving horizon strategy.

5.3.4 Closed-Loop Rescheduling Framework

The framework proposed to build the closed-loop optimized scheduling solution is comprised of two sections. The first is a simulation routine (see Figure 5.5), which is the main structure that forms the framework. Its main objective is to simulate the closed-loop scheduling in a moving horizon fashion. The second section that forms the framework, referred to as optimization routine (see Figure 5.5), is called recursively within the simulation routine. It is worth noting that the optimization routine refers to the future 10-days open-loop optimization within the closed-loop scheduling, while the simulation routine refers to simulate the closed-loop scheduling itself, which is performed for the future 40-days. Therefore, the closed-loop simulation sequentially aggregates multiple open-loop solutions in order to build the closed-loop solution.

Figure 5.5: Moving horizon simulation framework with closed-loop scheduling optimizations.


Initialize Closed-Loop Simulation Routine
1) Pre-define the Parameters, Variables, and Elements required
2) Call Optimization Routine
3) Are there disturbances?
Yes
No
4) Generate random disturbances
5) Update the State of the System
6) End of Closed-Loop Simulation?
$t = t + 1$
No
Yes
7) Build Closed-Loop Solution
2.1) Build the formulation as an MINLP problem
2.2) Decompose the MINLP in two subproblems
2.3) Solve the first subproblem (MILP)
2.4) Solve the second subproblem (NLP)
2.5) Save the MINLP Optimal Solution

Source: Author (2021).

The following steps explain how the framework works towards building the closed-loop optimized scheduling solution in a sequence of simulated iterations.

1) The framework initializes pre-defining all the parameters, variables, and other elements required for the closed-loop simulation. The simulation time horizon is defined as 40-days.
2) The optimization routine is called to model and to optimize the blend scheduling problem within the optimization time horizon (defined as 10-days). That means the first optimization ($t = 1$) is carried out from Days 1 to 10, so that only the information available for that period is considered in the modeling and optimization.

Similar for the second optimization ($t = 2$), carried out from Days 2 to 11, and so on.

2.1) Within the optimization routine, the blend scheduling problem is formulated as an MINLP as there are binary variables related to the operating modes and transfers for the blenders, as well as nonlinear terms related to the blending equations and properties tracking throughout the process.

2.2) The MINLP problem is decomposed in two subproblems using the MILP-NLP phenomenological decomposition heuristic proposed by Menezes, Kelly, and Grossmann (2015).

2.3) The first subproblem (MILP) arises from neglecting all the quality information (variables and constraints) from the original MINLP. Commercial solvers are used to optimize this MILP problem.

2.4) The second subproblem (NLP) arises from setting all the binary variables in the original MINLP by using the optimal solution found in the previous MILP. Commercial solvers are used to optimize this NLP problem. A linear programming (LP) successive linearization technique (found in IMPL) is employed to linearize the nonlinear terms in the NLP formulation whereby these linearized sub-problems are solved using LP solvers. Extensive testing showed that this approach has better performance than applying NLP solvers directly.

2.5) The solution of the NLP problem is a feasible solution of the original MINLP. This solution is saved and the framework moves back to the simulation routine.

3) Disturbances may or may not be considered in the simulation. This is chosen by the user. Not considering them would be useful to simulate an ideal or base scenario, whereas considering them would be much more realistic for industrial operations. Considering and comparing these two possible cases is one of the objectives of this work.

4) When the closed-loop simulation considers disturbances in the process, the framework randomly generates them according to some predefined criteria (i.e., which types of disturbances are assumed to happen, at which probability, and in which size or magnitude).

5) Whenever any disturbance happens, the state of the system is updated so that the simulation is up to date to match the real or ongoing scenario in the plant. For example, information such as flows, holdups logics, and properties, are updated.

Additional details on the distinct types of disturbances and the possible required updates are presented in Section 5.3.5. On the other hand, if disturbances are assumed not to happen, the state of the system is already up to date, so that no corrections are needed.

6) If the simulation ends, the closed-loop scheduling solution is built. Otherwise, the framework moves to iteration $t = t + 1$. The moving horizon is employed to perform sequential optimizations. Thus, on iteration $t = 2$ the problem is optimized from Days 2 to 11 and so on, until the last optimization, which is carried out from Days 40 to 49. The optimal solution is saved at each iteration.
7) Upon the convergence of the Simulation routine, a total of 40 optimizations are carried out and saved. Therefore, a total of 40 optimal solutions have been found. The closed-loop scheduling solution is built by aggregating or combining the solution of the first day from each of these 40 optimal solutions (i.e., Day 1 at iteration $t = 1$, Day 2 at iteration $t = 2$, Day 40 at iteration $t = 40$, and so on). A 40-days closed-loop solution is then achieved.

5.3.5 Disturbances

Uncertainties, noises, and any unforeseen and unexpected events are treated as disturbances in the formulation, allowing the use of variable feedback previously mentioned. Thus, adjustments in the model parameters (gains, biases, etc.) are not required as it remains identical in each scheduling iteration. The disturbances considered in this blend scheduling optimization problem are related to flows (incoming to and outgoing from the blender), feedstocks (their arrival dates and their qualities, such as specify gravity and sulfur content), demands for final products (amounts and release or due dates), and blender units (malfunctions or breakdowns). Whenever a disturbance occurs, the system is updated before the next re-optimization. Therefore, variable feedback is applied to correct the actual status of the plant, as there might be differences between the modeled status and the real status of the plant. The required updates when distinct types of disturbance happen are discussed in the following. It is worth mentioning that we are not interested in what causes or triggers the disturbances, as well as what could be done or performed to mitigate or avoid them in the plant without rescheduling, although that would be an interesting topic with an emphasis on real industrial operations. Rather, it is assumed that the disturbances may eventually happen, and we aim at developing a framework capable of handling them and simulating distinct scenarios to investigate their impact on the closed-loop schedule. More specifically, the interest is in understanding what the impact of disturbances on the schedule are, what are the consequences of neglecting or ignoring them, and what can be done to mitigate their negative impact or to take advantage of any positive conditions or directions.

5.3.5.1 Flows Incoming to the Blender

Flow disturbances are essentially noises in the streams throughout the process. It is assumed that they may happen in each one of the 24 flows incoming to the blender (see Figure 5.4) with a given trigger probability. If there is a noise in any flow, its original value is randomly and uniformly changed up to 20% (i.e., if the optimal flow is 10.0 m^3/h, the updated flow, after the noise is computed, is randomly chosen between 8.0 and 12.0 m^3/h). Whenever there are noises in the flows, in order to correct the actual status of the plant, it must be updated:

- Inventory in the feeding tanks (F);
- Inventory in the storage tanks (S);
- Inventory in the feedstocks (R);
- Flows from the storage tanks S to the demand pools D (to check whether the product demands were met).

If the measured (noisy) flow incoming to the blender is lower than the optimal flow, the inventory in the S tank must be reduced and the inventory in the F tank must be increased by the same value of the respective noise, and vice-versa. Moreover, it is necessary to check whether there was sufficient material in the S tank to supply the required product demand, as well as whether the disturbance interfered in the lower and upper bounds of the F and S tanks (that can also change the flows from the R feedstocks to the F tanks, and from the S tanks to the D demands). There are lower and upper hard bounds for the holdups of each tank and we consider that they must be respected regardless of the disturbance (e.g., a disturbance cannot make the holdup of a tank to be lower than its lower hard bound or higher than its upper hard bound).

5.3.5.2 Flows Outgoing from the Blender

Similarly, it is assumed that there may be noises in the six flows outgoing from the blender (see Figure 5.4) with a given trigger probability. If there is a noise in any flow, its original value is randomly and uniformly changed up to 20%. In that case, it must be updated:

- Inventory in the storage tanks (S);
- Flows from the storage tanks S to the demand pools D (to check whether the product demands were met).

Moreover, whenever there is a noise in one or more flows outgoing from the blender to one of the product pools, there might be discrepancies regarding the pool quality of the respective product. For instance, consider a hypothetical case in which two feedstocks Feed1 (high quality, low sulfur content) and Feed2 (low quality, high sulfur content) are mixed, and the sulfur content of this pool is calculated in the optimization to enforce some specifications constraints. Without noise or disturbances, the sulfur content of that pool in the optimal solution should be the same as the real one in the process. However, with noise in these flows, there are three different scenarios. First, if the ratio Feed1/Feed2 is higher than expected, the final pool has a better quality and a lower sulfur content, therefore the specifications for the final product will still be met although it is expected a product giveaway. Second, if the ratio Feed1/Feed2 is lower than expected, the final pool has worse quality, so that this product specification is not met. In this scenario, we assume that the products can still be sold, but by 90% of their original selling price. Third, if noise does exist although the ratio Feed1/Feed2 remains unchanged (what is not likely to happen), no differences in the pool quality are expected.

5.3.5.3 Arrival of Feedstocks

There are estimated dates of arrival of each feedstock R1 to R4, with expected amounts and properties in terms of specific gravity and sulfur content. The possible disturbances assumed to happen in this case are a sudden change in the arrival date and/or in the quality of a feedstock. The amount is not considered as a possible disturbance because it is typically large enough not to interfere on the short-term schedule. When a feedstock is expected to arrive at certain date, but it does not arrive, or if its properties are different from what they were expected to be, the state of the system must be updated in the model. To clarify the understanding of these two cases, consider a hypothetical example in which the feedstock R2, with specific and known properties, is expected to arrive by the end of day 2, so that from the optimization side it is expected that the feedstock would be available from day 3 onwards. However, if the feedstock R2 does not arrive on the expected date and/or if its properties are different than expected, that new information must be updated in the model, and a rescheduling should be performed to take it into account.

5.3.5.4 Market Fluctuations: Amount and Due Date of Product Demands

There are market fluctuations related to the amounts and due dates of product demands. There are fixed daily demands for each product (and therefore, up to three distinct demands every day), and quality specifications (specific gravity and sulfur content) that must be met. It is assumed there is a probability for the amount of each product demand to change every day. If that disturbance is triggered, the original value of the respective demand is randomly and uniformly changed, which could be either increased or decreased. It is similar for the due dates of each demand, in which it is assumed a probability that they change (in that case the demand would be either preponed to the previous day or postponed to the following day). The exceptions are that the product demands of the first (current) day in the optimization cannot have their amounts or due dates changed, as it would be too difficult to adapt the incumbent schedule. Similarly, it is also valid that the demand of the second day cannot be preponed to day one. In each optimization, it is possible that each of the 27 future demands (three products times nine days) would be disturbed regarding their amounts and due dates. It is worth mentioning that if the daily demand for any product is not met, a daily penalty equal to 10% of the product value is applied, and the demand is postponed to the following day. Furthermore, we also consider to be possible selling up to 20% more than the daily demand for the same price, what gives an additional degree of freedom in the optimization.

5.3.5.5 Blender Breakdown

Eventual malfunctions and breakdowns are expected to happen in any process at some point. It is assumed that each blender is subject to breakdown at a fixed daily probability. It is assumed, however, that at least one blender is available to be used at all times, otherwise the schedule would be impracticable and infeasibilities would likely arise. If a blender breaks down, it cannot be used for a period between 24 h and 96 h (randomly and uniformly chosen). Besides, as the breakdown may happen throughout the day, it is also considered the exact moment of this event. For instance, if a blender with a capacity of 100 m^3/day breaks down in the middle of Day 1 and the time needed to fix it is 42 h, then it is assumed that the blender can operate for half of Day 1, processing at most 50 m^3 of material (12 h broken), does not operate in Day 2 (more 24 h broken), and can operate for 18 h in Day 3, processing at most 75 m^3 (more 6 h broken).

5.3.6 Mathematical Formulation

The blend scheduling problem presented in Figure 5.4 is formulated as a mixed-integer nonlinear programming (MINLP) problem. In each optimization, the MINLP model is decomposed in two subproblems (MILP-NLP decomposition), which are sequentially optimized. The mathematical formulation for the logistics problem (MILP) and for the quality problem (NLP) are presented in detail in the Supplementary Material (Appendix A).

5.4 Case Study, Results, and Discussion

Based on the blend scheduling problem and its respective mathematical formulation (presented in Section 5.3 and in the Supplementary Material, respectively), a case study has been proposed to simulate multiple optimizable scenarios with and without disturbances, to analyze and to compare the results, to investigate the impact of the disturbances in the final closed-loop simulated solution, and to understand what the consequences are of not considering them in the formulation and optimization. Information on the size and complexity of the formulation is presented as follows. In the blend scheduling problem optimization for the future 10-days with discrete time steps of one day, and not considering any disturbance in the formulation, in the MILP model there are 2,184 constraints (332 equalities), 604 continuous variables, 651 binary variables and 961 degrees-of-freedom; in the NLP model there are 490 constraints (410 equalities), 442 continuous variables and 70 degrees-of-freedom. All the proposed scenarios for the case study are modeled using the UOPSS formulation within the modeling platform IMPL (Kelly and Menezes, 2019) from Industrial Algorithms Limited. The optimizations are carried out through the commercial solvers GUROBI 9.0.0 and CPLEX 12.10.0 each linked to IMPL. The computer used was an Intel Core i7 with 2.90 GHz and 16 GB RAM.

Multiple scenarios are proposed to investigate the differences between the open-loop and closed-loop approaches, and the impact of the disturbances in the operations (regarding both the economic value and the operational schedule). The similarities among all of them include the optimization time horizon (10 days), the discrete time steps (1 day), and the full simulated time horizon (40 days). In the following each proposed scenario is presented and discussed, as well as their respective motivation towards understanding the relevance of the topics addressed in the proposed re-scheduling framework.

5.4.1 Impact of Disturbances on the Operations

We are interested in investigating how typical disturbances affect both the economic value of the process (i.e., profitability) and the operations to be scheduled and carried out in the plant. The following scenarios are proposed to simulate the 40-days closed-loop scheduling:

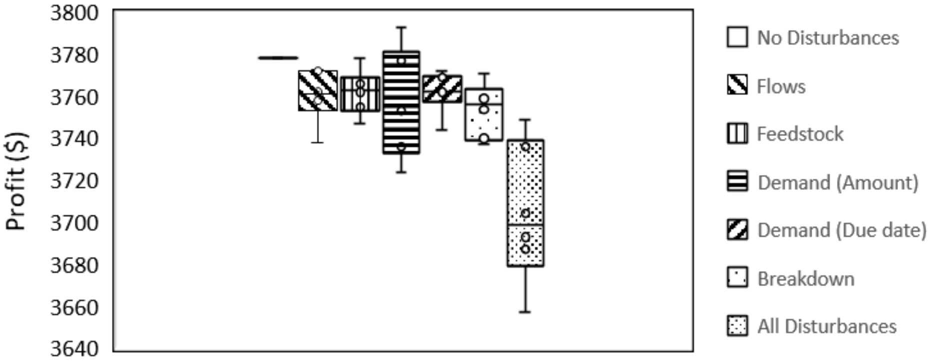
- *Scenario 5.1*: There are no disturbances in the process.
- *Scenario 5.2*: Only disturbances 1 and 2 (i.e., flows), presented in sections 5.3.5.1 and 5.3.5.2, are assumed to happen. There is 50% probability that any flow incoming to or outgoing from any blender is subject to a noise, in which the respective flow is uniformly randomized to be between $\pm\ 20\%$ of its original value.
- *Scenario 5.3*: Only disturbance 3 (i.e., feedstock arrival), presented in section 5.3.5.3, is assumed to happen. The feedstocks R2 and R3 are supposed to arrive at the end of Day 2 and Day 25, respectively. However, each arrival is assumed to

be delayed by three days, so that the feedstocks arrive at Day 5 and at Day 28, respectively. Moreover, the feedstock R2 not only arrives late, but also with unexpected properties ($\mathrm{sg} = 0.86$ instead of $\mathrm{sg} = 0.85$, and $\mathrm{sul} = 1.01$ instead of $\mathrm{sul} = 1.00$).

- *Scenario 5.4*: Only disturbance 4 (i.e., market fluctuations: amounts), presented in section 5.3.5.4, is assumed to happen. There is a probability of 5% that any of the demands (for each product at any of the ten future days) is subject to market fluctuation changes. Whenever there is a demand change, it is assumed a uniform fluctuation of $\pm\ 20\%$ of its original value.
- *Scenario 5.5*: Only disturbance 4 (i.e., market fluctuations: due dates), presented in section 5.3.5.4, is assumed to happen. There is a probability of 3% that the due date of a given product changes at a given day. In that case, the respective demand is preponed or postponed by one day. The only exception is that the demand of the current day (first day of the optimized time horizon) cannot change.
- *Scenario 5.6*: Only disturbance 5 (i.e., blender breakdown), presented in section 5.3.5.5, is assumed to happen. There is a probability of 3% that any blender breaks down at any day.
- *Scenario 5.7*: There are all five disturbances simultaneously. Their respective triggering probabilities are such as in Scenarios 5.2 to 5.6.

When there are no disturbances, there is a unique closed-loop solution because there is no randomization in the algorithm. However, whenever any disturbance is assumed to happen, they are randomly simulated by the algorithm, so that multiple distinct possibilities could happen (e.g., there could be multiple demand peaks of product D1, or sequential breakdowns of a blender), depending on the sequence of random numbers generated. Thus, aiming at more robust analyses and conclusions, multiple closed-loop simulations are performed for each scenario (using distinct seeds for the randomization) to improve the reliability of our approach by providing more representative results. Figure 5.6 presents a box plot chart to illustrate how the disturbances impact the profitability of the process, in which five closed-loop schedules were simulated for each scenario (except for Scenario 5.1, as there is no randomization). The profit shown regards the entire 40-days closed-loop scheduling.

Figure 5.6: Impact of Scenarios 5.1 to 5.7 in the closed-loop scheduling profitability.

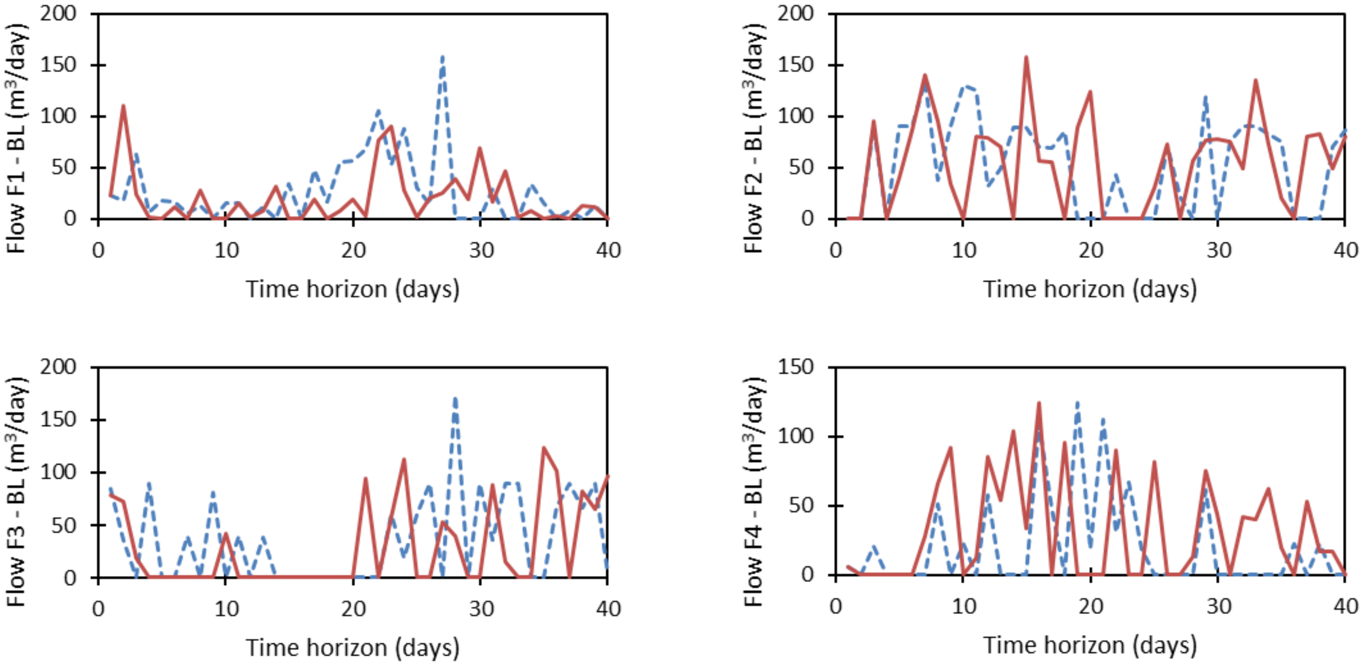


Source: Author (2021).

As expected, although disturbances may eventually lead to an increase in the profit, their impact is usually negative. Moreover, disturbances regarding demands amount and unit breakdowns typically have a worse effect in the closed-loop simulated solution. The worst scenario is when all disturbances are assumed to happen, which is indeed the most representative situation in industrial operations.

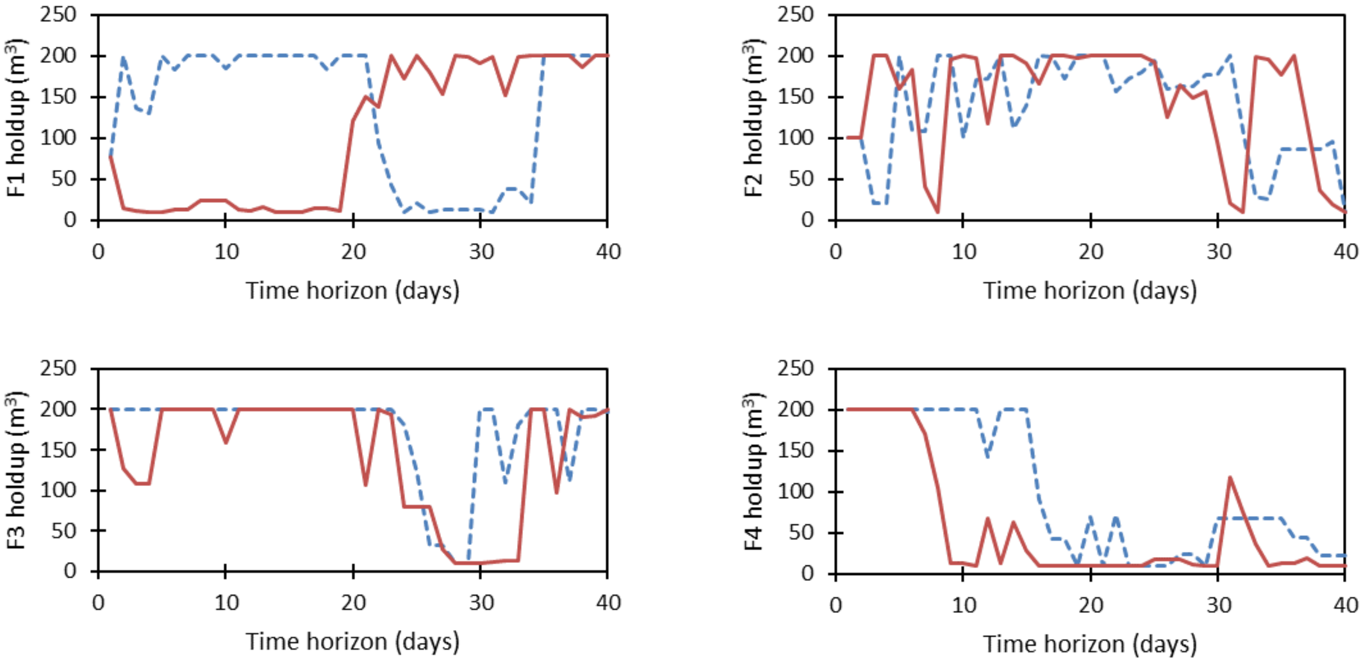
Despite the expected economic impact of noises and disturbances to the closed-loop solution, another meaningful insight regards the process operations and its respective schedule. Not accounting for changes in the process does not only limit the economic value of that process and increase the risk of infeasibilities on the model optimization, but also leads to significant differences regarding the optimal scheduling to be implemented in the real plant. Figures 5.7 to 5.9 present the trend or line plots for the flows from the F tanks to the blenders, and the inventories of F and S tanks, respectively, over the entire simulated time horizon of 40 days for one case of Scenario 5.7. The blue dashed line represents the scheduling operations when disturbances are not assumed to happen, and the red solid line when all the five types of disturbances are considered in the modeling and optimization.

Figure 5.7: Closed-loop operational schedule for flows from the F tanks to the blenders without (blue dashed line) and with (red solid line) disturbances (Scenario 5.7).



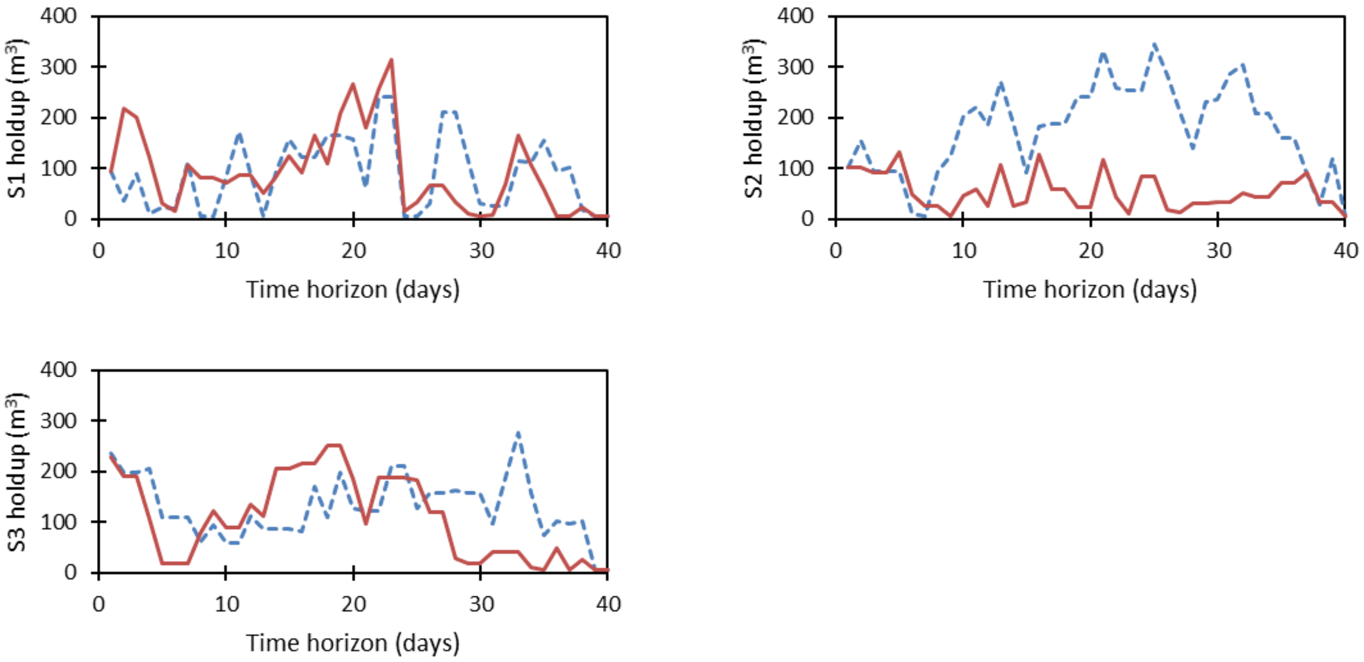
Source: Author (2021).

Figure 5.8: Closed-loop operational schedule for inventories of F tanks without (blue dashed line) and with (red solid line) disturbances (Scenario 5.7).



Source: Author (2021).

Figure 5.9: Closed-loop operational schedule for inventories of S tanks without (blue dashed line) and with (red solid line) disturbances (Scenario 5.7).



Source: Author (2021).

Figures 5.7 to 5.9 show that there is a significant difference between the two plotted cases (with and without disturbances). The direct impact for industrial operations is that not considering noises and disturbances (assuming that they happen) would lead to the implementation of a completely different schedule rather than the real optimal schedule (considering the real state of the process, which includes all the changes resulted from disturbances and unexpected events). It is also worth mentioning that even though we do not address inventory costs in the proposed example, it is clear that they might be relevant for the topic, as there is a significant difference in the inventories of tanks between the closed-loop schedules with and without disturbances, as shown in Figures 5.8 and 5.9. In summary, scheduling operations when disturbances are or are not considered might be completely different from each other, which reinforces the importance of online scheduling strategies for process optimization purposes.

5.4.2 Impact of Neglecting Disturbances on the Operations

We also investigate what is the impact of neglecting the disturbances, i.e., assuming that disturbances do happen, what would be the consequences of not accounting for them in the modeling and optimization in terms of economic value, schedule operations, product specifications, infeasibilities, etc. We discuss the impacts of neglecting each disturbance individually in the following.

5.4.2.1 Flow Disturbances

Noises in the flows might affect the mixture ratio, in which more or less amount of each material is blended. That changes not only the amounts produced, but more importantly, its quality. Besides, it becomes even harder to track the qualities of intermediate and final streams, leading to much more unreliable predictions, and difficulties to manage the control the operations. Some of the operational issues that arise from not accounting for noises in the flows in the blend scheduling problem addressed herein are as follows.

A noise in the flow from tank F1 to the Blender A was considered to happen at Day 1, which reduces that flow in about $5.0\,m^3$. That stream (pure feedstock R1) is blended with another stream composed of pure feedstock R3. The noise affects the ratio R1/R3 in the mixture and hence, also affects the quality of the material sent to and stored in the storage tank S2. That blend is used in the following days to supply the demand of product D2. Not accounting for that noise leads to an incorrect tracking of the properties of the blend stored in the tank S2. The calculated properties of that blend are actually worse than the real properties, so that when that material is used to supply the demand of product D2, that product is unspecified. That implies in operational costs to either reprocess that product or in discounts to sell that product at a lower price. Not accounting for that noise would make it impossible to capture this information and to fix the tank S2 quality as soon as possible (e.g., at Day 2), so that this negative impact could be reduced.

Most importantly, this is just an example regarding one noise that happened at Day 1, but multiple noises are actually likely to happen every day. Tracking information throughout the process is already a difficult task, and if this tracking is not accurate enough, that may lead to severe plant-model mismatches, in which scheduling infeasibilities are highly likely to happen.

5.4.2.2 Arrival of Feedstocks Disturbances

The feedstock R2 was assumed to arrive in the end of Day 2, so that it would be available to be used from Day 3 onwards. Assuming that there is a delay in the arrival of feedstock R2 of 3 days (i.e., feedstock R2 arrives at Day 5 instead of at Day 2), the following operational issues arise:

- At Days 4 and 5, the operations were scheduled to use 25.62 m^3 and 165.27 m^3 of feedstock R2, respectively. These operations could not be performed and they were postponed because the inventory of feedstock R2 was empty. The feedstock R2 was available to be used from Day 6 onwards. Meanwhile, the storage tanks had some additional amount of each product, so the demands for Days 4 onwards could be partially, but not entirely, met.
- The main issue here was that the lack of feedstock R2 completely changed the blends scheduled to produce the final products, and the incumbent schedules could not be carried out. Alternatives would be delaying the processing of that blend, or propose a similar blend but using other feedstocks, although that information (which feedstocks to use and in which ratio) would have to be calculated.
- The blender units were already scheduled to operate at full capacity from Day 6 to Day 13. At Days 14 and 15, they were used to process the blend with feedstock R2 that was already late. The delay to process that material was around 10 days. However, if rescheduling had been performed, the blenders could have processed other feedstocks at Days 4 and 5 (instead of feedstock R2), and could have processed feedstock R2.
- Besides the delay to deliver part of the product demands, which imply in late deliver costs and fees, as well as eventual customer dissatisfaction, there were quality changes in the final products. The product D1 delivered at Day 4 should be a mixture of feedstocks R1, R2, and the material already stocked in the D1 pool. Not mixing the feedstock R2 resulted in a higher-quality final product D1, which represents a product giveaway, and hence, a decrease in the economic value of the process.

- These quality issues also arose because the quality of the feedstock R2 was different than the expected. Without rescheduling the problem that information could not be taken into account, leading to several quality mismatches throughout the entire closed-loop schedule. That resulted not only in product giveaways but also in unspecified products.
- There were similar issues regarding the delay in the arrival of feedstock R3.

5.4.2.3 Market Fluctuation Disturbance: Amount of Demands

Whenever market fluctuations affect the amounts of product demands, there are some interesting changes on the process operations:

- For example, the demand of product D1 at Day 3 was 100 m^3, but there was a market fluctuation reducing that demand to around 95 m^3 (that disturbance was randomly generated).
- The main consequence is that less product D1 is sold, and more product D1 remains in the storage tank S1.
- From Day 4 onwards, more product D1 is produced and stored in the tank S1. However, that stream sent to tank S1 on Day 4 has a slightly different quality than the product already stored, which leads to errors when calculating the properties of that final blend.
- On Day 5 there is another demand of product D1 to be met, and because the amount and properties of the blend already stored in the tank S1 are not accurately known, there is a giveaway in the product D1 sold on Day 5.
- Another disturbance was to increase the demand of product D2 at Day 7, resulting in the lack of demand of product D2 at Day 10, which implied in costs for delivering the product late.
- Most importantly, not accounting for this type of disturbance would result in several plant-model mismatches regarding both the amount and the qualities of final products to be stored and sold. The most common issues would be related to product giveaways, unspecified products, and inventory limits.

5.4.2.4 Market Fluctuation Disturbance: Due Date of Demands

Whenever market fluctuations affect the due date of product demands, there are interesting changes on the process operations as well. Some issues found in the closed-loop simulations of the optimized scenarios are as follows:

- A market fluctuation disturbance happened at Day 6, in which the demand of product D3 (40 m^3), was fully preponed from Day 9 to Day 8. That means there were two days for the schedule operations to prepone by one day the product of 40 m^3 of D3.
- However, if rescheduling is not performed, and hence, the schedule is not updated, meeting this preponed demand relies exclusively on having enough product D3 already produced (which in that case there was only 12 m^3 of product D3 stored, which was not enough to fully supply the demand).
- Moreover, the blenders did not operate at full capacity on Day 8, which means a rescheduling would have mitigated the impact of the disturbance. Additional costs due to the late delivery could have been avoided.
- It is worth noting that this is just one example of due date disturbance in the closed-loop simulation, but multiple of them actually happened and introduced issues for the schedule operations.
- Most importantly, depending on the frequency with which market fluctuations happen, not rescheduling the problem would likely result to late demands almost every day.

5.4.2.5 Blender Breakdown Disturbance

Rescheduling the problem may also reduce or smooth issues caused by unit breakdowns, such as in the following example.

- The second blender suddenly broke down at Day 11. The consequences are that at Days 11 and 12, around 36 m^3 and 80 m^3 of material (product D1) could not be processed, respectively. These productions were postponed to the following days and were performed according to the availability of the blenders.
- Regardless of rescheduling the problem or not, the second blender was not available to be used during that time. However, by rescheduling the problem, a

new optimal solution was found, in which it was preferred to update (change) the scheduled production in the first blender in order to better supply the demand of products.

- More specifically, the first blender was scheduled to produce product D3 at Day 13, but a better schedule, with higher profitability, was achieved by producing product D1 at Day 13 to supply its incumbent demand, whereas product D3 was produced at Day 14 by the second blender. In this new optimal schedule, both demands were met, without any additional costs. On the other hand, not rescheduling the problem would lead to late deliveries and consequent additional costs.

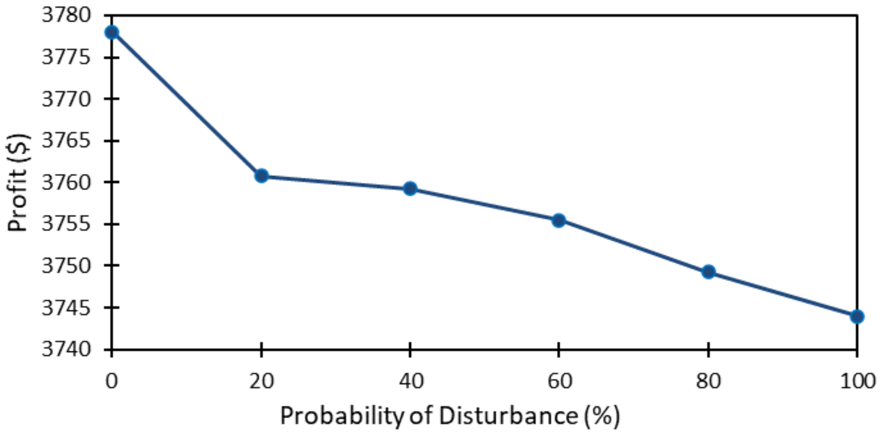
5.4.3 Individual Impact of Disturbances on the Operations

A key aspect regarding the simulation approach developed herein (and that significantly impacts the closed-loop scheduling) is to properly tune or choose the parameters and probabilities by which each disturbance is assumed to happen (e.g., what is the probability of a random unit to breakdown at a specific day, and if that happens, how long will it take to fix the unit). That is especially important in industrial operations to understand how a given noise, disturbance, disruption, or unforeseen event might impact the incumbent and future schedules regarding their profitability and operations. Therefore, Scenarios 5.8 to 5.12 are proposed to investigate how the proposed disturbances impact the closed-loop solution depending on their probability to be triggered as well as the dimension of the changes.

5.4.3.1 A Deeper Analysis on Flow Disturbances

Scenario 5.8 further investigates how noises in the main process streams would affect the scheduling operations. Let us assume that a given main stream (either incoming to or outgoing from the blender) is subject to noises at a fixed probability $d_1 \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$. For each one of the six possible values of d_1, the closed-loop scheduling was simulated five times (using distinct randomization seeds) to provide more robust and reliable results. The average mean closed-loop objective function (i.e., the average mean among the five closed-loop simulations) for each distinct value of d_1 is represented by a dot in Figure 5.10.

Figure 5.10: Impact of the flow disturbances in the closed-loop scheduling.

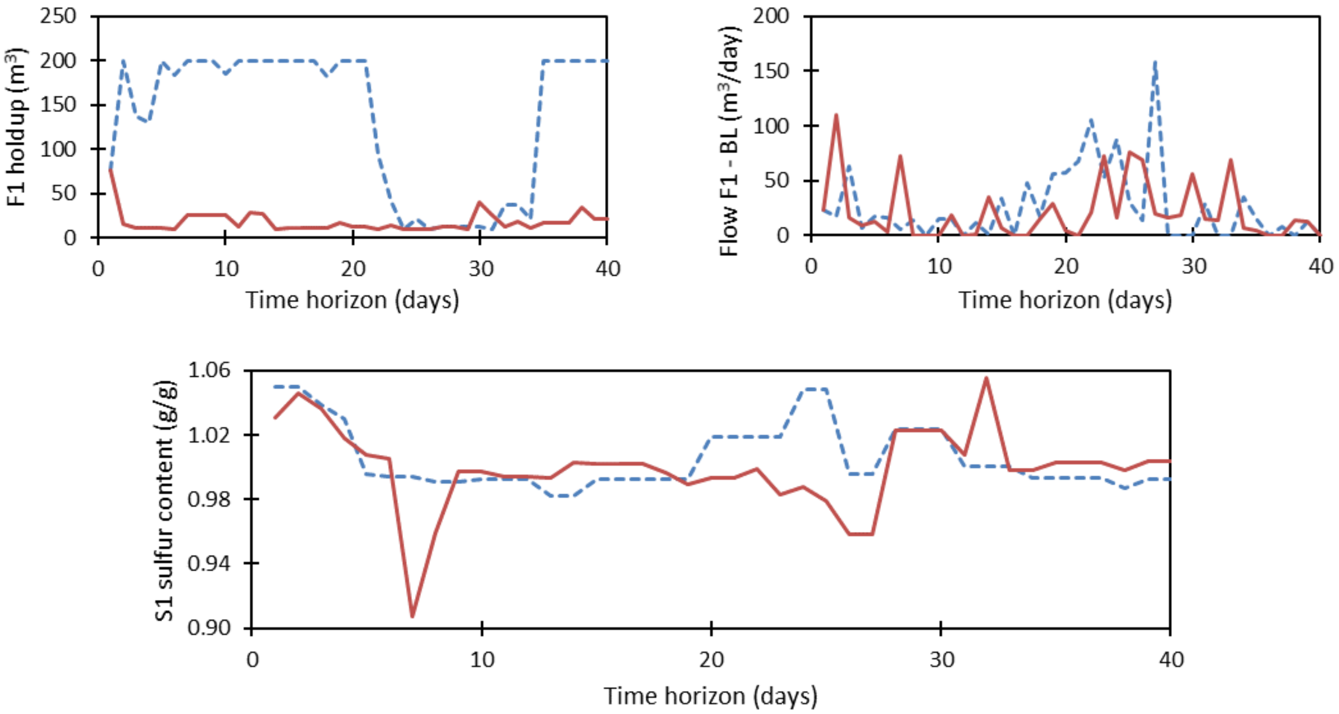


Source: Author (2021).

Although there are quality constraints to impose the products to meet some specifications in terms of their specific gravity and sulfur content, noises in the flows can negatively result in unspecified products. For example, if a disturbance results in mixing a lower amount of a high-quality feedstock or a higher amount of a low-quality feedstock, the final product may not meet the desired specifications, and would either have to be reprocessed or sold by a lower price. For a matter of simplicity, it is considered that every product that does not meet all the quality requirements can still be sold but for 90% of the original price.

Another important feature analyzed is the difference on the operational schedule when disturbances happen or not in the process. Figure 5.11 presents the line plots comparing a case without disturbances (blue dashed line) and a case with flow disturbances ($d_1 = 0.4$), represented as a red solid line. For the sake of simplicity, only the operations for the inventory of tank F1, the total flow from tank F1 to the blenders, and the sulfur content of the tank S1 are shown. It can be noticed from Figure 5.11 that the operations when flow disturbances are assumed to happen are completely different from the operations without considering any disturbance. That highlights the importance of accounting for them in the rescheduling optimization.

Figure 5.11: Closed-loop operational schedule without (blue dashed line) and with (red solid line) disturbances 1 and 2.

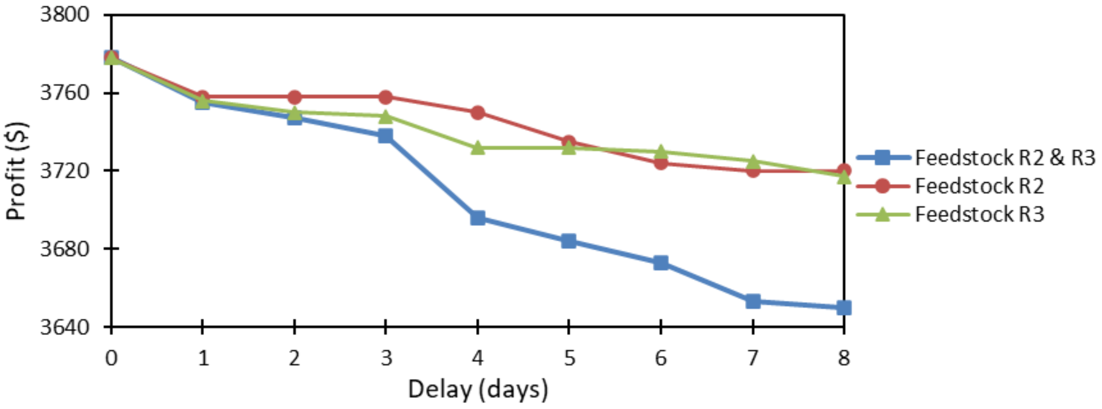


Source: Author (2021).

5.4.3.2 A Deeper Analysis on the Arrival of Feedstock Disturbance

Scenario 5.9 investigates how changes on expected arrival dates of raw materials or feedstocks impact the plant operations. For that, it is assumed that the feedstocks R2 and R3 are expected to arrive at the end of the days 2 and 25, respectively. However, there are hypothetical unforeseen delays of $d_3 \in \{0, 1, 2, 3, 4, 5, 6, 7, 8\}$ days in their arrival. Moreover, three possibilities are considered, in which the delay happens only for feedstock R2, only for feedstock R3, and for both. The closed-loop objective functions for each case and for each value of d_3 are plotted in Figure 5.12.

Figure 5.12: Impact of the feedstock arrival disturbance in the closed-loop scheduling.

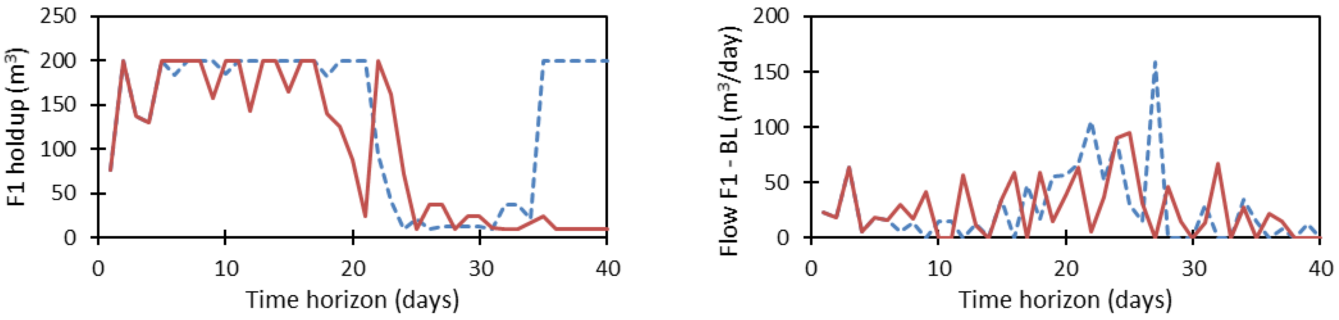


Source: Author (2021).

When different delays (up to eight days) for each scenario are assumed, there is a non-increasing objective function behavior, as shown in Figure 5.12. The highest objective function is from the case without delays, and as they occur and increase in size, there is typically an expected decrease in the profit. It is important to note, however, that a higher delay does not necessarily affect the optimal solution (e.g., in the case with delay of feedstock R2, the optimal solution does not change when the delay increases from one to two days). In general, delays in the arrival of feedstocks decrease the profit because there might be fewer combinations of feedstocks that could be used in the process. It is worth mentioning that critical issues may arise if there are multiple long delays, resulting in higher reduction in the profit as well as eventual infeasibilities.

Similarly to Scenario 5.8, the line plots representing the operational schedule (holdup of tank F1 and total flow from tank F1 to the blenders) for the cases with disturbance 3 using $d_3 = 4$ (red solid line), and without disturbances (blue dashed line) are presented in Figure 5.13. There are significant differences between the closed-loop operational scheduling in these two cases.

Figure 5.13: Closed-loop operational schedule without (blue dashed line) and with (red solid line) disturbance 3.

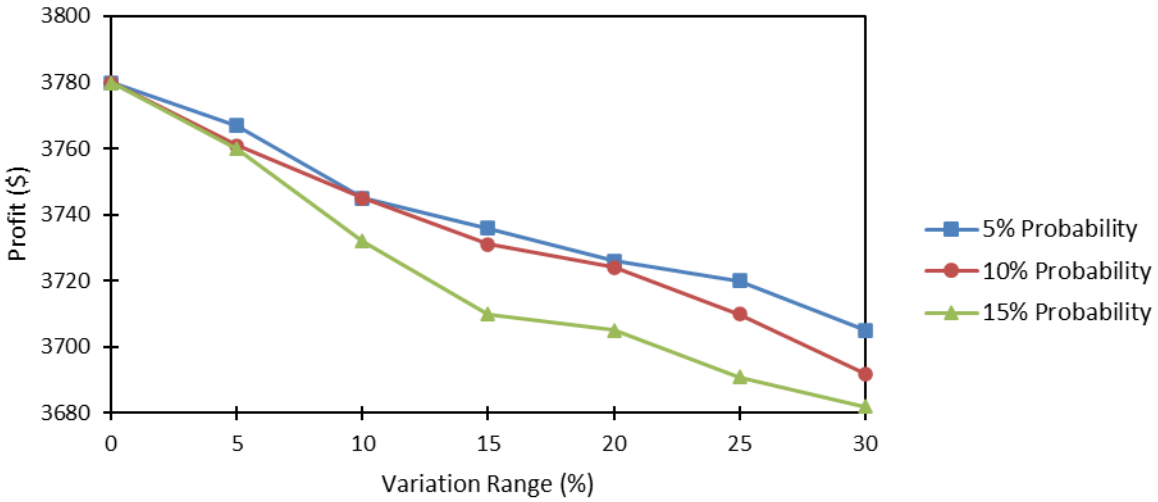


Source: Author (2021).

5.4.3.3 A Deeper Analysis on Market Fluctuation (Amount) Disturbance

Scenario 5.10 investigates the impact of market fluctuations on the final product demands. It is assumed that at any day (except for the current day, i.e., the first day in the optimization time horizon), there is a probability that the expected demand of a given product would change. Two main effects are derived from this disturbance. First, whether the demand decreases or increases affects the amount of product to be produced and sold (which could lead to either a negative or positive effect to the profitability). And second, any changes result in the need to review, adapt, or reprogram the incumbent schedule (which is typically negative for the operations). It is considered three distinct probabilities for the disturbance to be triggered of $d_{41} \in \{0.05, 0.10, 0.15\}$, and in those cases, there is a variation range in the amount of the respective demand of $d_{42} \in \{0.00, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30\}$, which can be positive (increasing the demand) or negative (decreasing the demand). The closed-loop scheduling is built for each case and the objective functions are plotted in Figure 5.14.

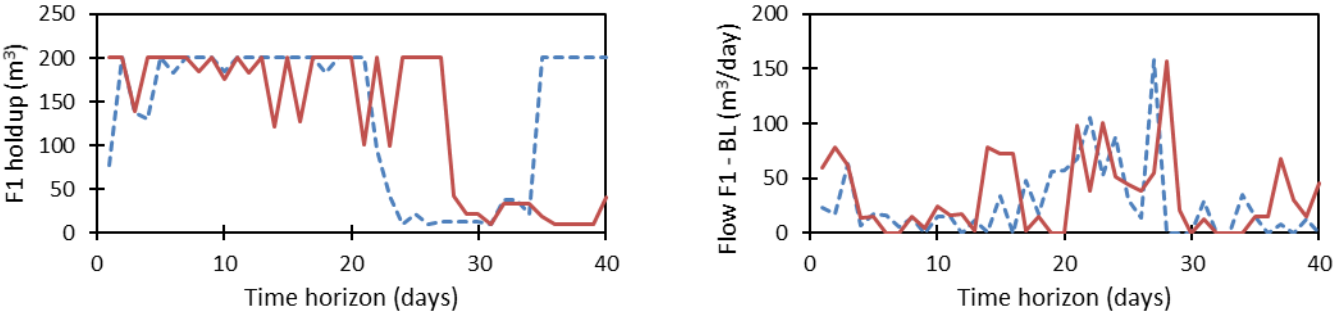
Figure 5.14: Impact of the amount of demand disturbance in the closed-loop scheduling.



Source: Author (2021).

It is typically expected that a higher probability of disturbance leads to a lower operational profit, although it is worth to be mentioned that the overall change in the profit may be positive due to a sufficiently high increase in the amount of demand even with the need to adapt against the changes (i.e., the increase in the profit due to a higher demand overcome its eventual decrease because of the disturbance in the process). In general, however, it is expected a lower profit as the probability of a disturbance to happen increases or the more the demand changes. The line plots representing the operational schedules (for the holdup of tank F1 and the total flow from tank F1 to the blenders) for the cases with disturbance 4 (red solid line) using $d_{41} = 0.10$ and $d_{42} = 0.15$, and without disturbances (blue dashed line) are presented in Figure 5.15.

Figure 5.15: Closed-loop operational schedule without (blue dashed line) and with (red solid line) disturbance 4.

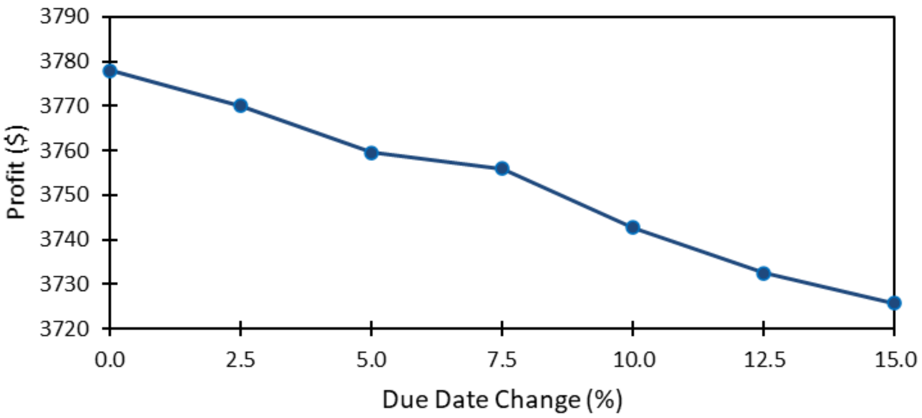


Source: Author (2021).

5.4.3.4 A Deeper Analysis on Market Fluctuation (Due Date) Disturbance

Scenario 5.11 considers market fluctuations on the due date of the demands. Each demand considered in the model (for each product at each day) is subject to a probability $d_5 \in \{0.000, 0.025, 0.050, 0.075, 0.100, 0.125, 0.150\}$ in which its respective due date changes by one day (the demand can be preponed or postponed, expect for the demands of the first or current day, which must stay the same). The closed-loop scheduling is simulated five times (using distinct sets of random numbers) and the average mean closed-loop objective function (i.e., the average mean among the five simulations) for each value of d_5 is represented in Figure 5.16.

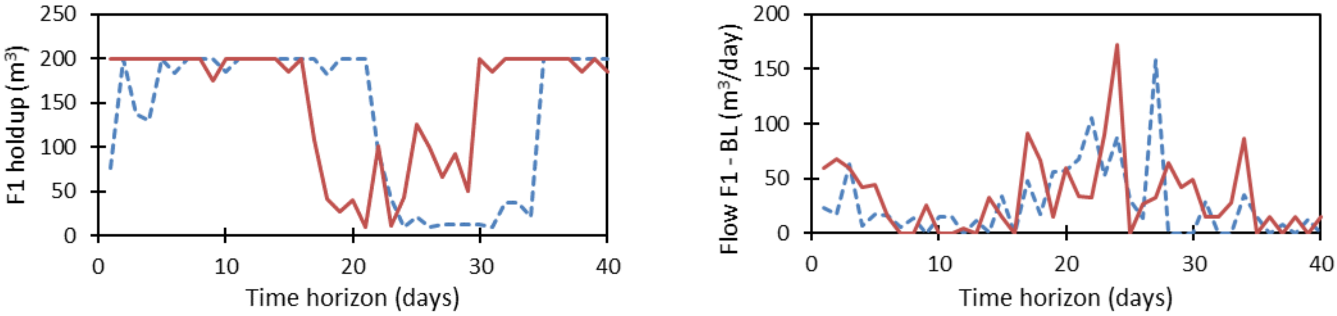
Figure 5.16 : Impact of the demand due date disturbance in the closed-loop scheduling.



Source: Author (2021).

The demand of each product remains unchanged in these closed-loop simulations, so a non-increase behavior in the closed-loop objective function is expected as the triggering rate of the respective disturbance increases, as shown in Figure 5.16. The line plots representing the operational schedules for the cases with disturbance 5 using $d_5 = 0.075$ (red solid line), and without disturbances (blue dashed line) are presented in Figure 5.17.

Figure 5.17: Closed-loop operational schedule without (blue dashed line) and with (red solid line) disturbance 5.

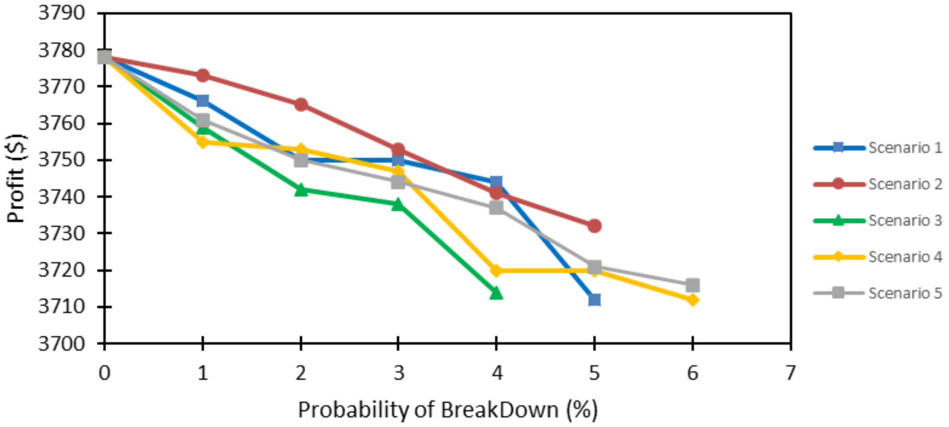


Source: Author (2021).

5.4.3.5 A Deeper Analysis on Blender Breakdown Disturbance

Scenario 5.12 investigates the impact of breakdowns in blender units on the process operations. At each day, there is an assumed probability $d_6 \in \{0.00, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07\}$ that a blender breaks down (the probability d_6 is for each blender unit). It was assumed, however, that simultaneous breakdowns of both blenders do not happen). Whenever there is a breakdown, the blender cannot be used for a (randomly and uniformly chosen) period of one to four days. The closed-loop scheduling is simulated five times (using distinct sets of random numbers) for each value of d_6, and the respective closed-loop objective functions are represented by a dot in Figure 5.18. The end of each curve indicates that the closed-loop scheduling solution became infeasible at a certain probability of breakdown disturbance (e.g.., depending on how often the blender breaks down, the scheduling operations cannot meet the product demands).

Figure 5.18: Impact of the blender breakdown disturbance in the closed-loop scheduling.

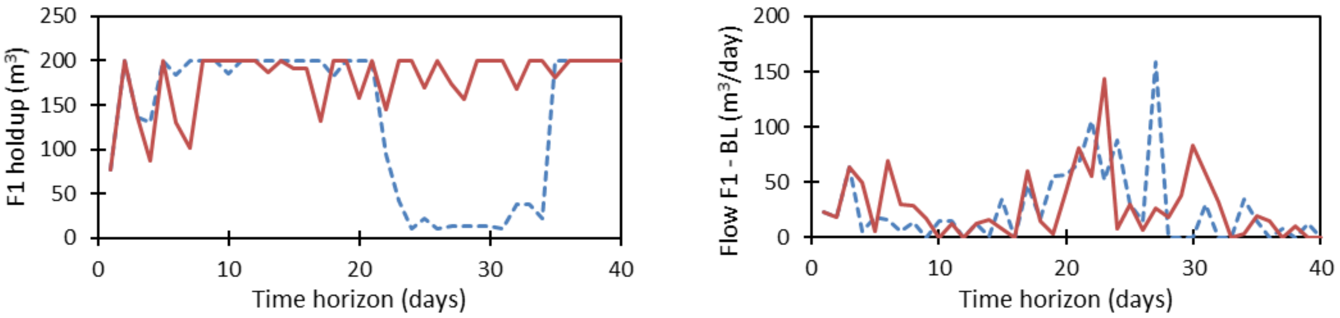


Source: Author (2021).

Breakdowns typically result in negative impacts for the operations, in which a decrease in the profit is expected. Furthermore, the higher the probability of breakdown (i.e., higher average number of breakdowns), the higher the risk of infeasibilities in both the optimization and scheduling operations. The five closed-loop solutions simulated in Scenario 5.13 become infeasible when the probability of breakdown becomes around 5% to 7%, in which the closed-loop objective function is no longer plotted. It is worth noting, however, that typical breakdown probabilities on industrial operations are expected to be much lower than the worst-case probabilities simulated herein.

The line plots representing the operational schedules for the cases with disturbance 4 (red solid line) using $d_6 = 3.0$, and without disturbances (blue dashed line) are presented in Figure 5.19. As in the previous comparisons, there are significant differences between the schedules.

Figure 5.19: Closed-loop operational schedule without (blue dashed line) and with (red solid line) disturbance 6.



Source: Author (2021).

5.5 Conclusions

For improved industrial operations, proper mathematical formulation and optimization are fundamental. That includes minimizing the plant-model mismatches so that the optimal solution is coherent with the actual process conditions and operations as much as possible. Because of the high nonlinear and uncertain nature of most industrial problems, unforeseen events and uncertainties are likely to happen, motivating a continuous optimization cycle, in which the current state of the system is updated, and a closed-loop rescheduling is performed in a moving horizon approach.

The online closed-loop rescheduling approach proposed herein is based on dynamic scheduling and relies on a systematic bi-layer framework that simulates the closed-loop scheduling solution for continuous nonlinear processes within a moving horizon strategy, in which noises, disturbances, disruptions, and other unforeseen events are assumed to happen with respective triggering probabilities. The framework is employed to test multiple scenarios for a MINLP blend scheduling problem in order to investigate the impact of the disturbances in the closed-loop scheduling regarding both the economic value and the scheduling operations. Several common types of disturbance are considered and further analyses on the issues caused by neglecting them are addressed.

Although most unexpected disturbances negatively affect the scheduling operations, an efficient rescheduling mechanism might mitigate or minimize issues on the incumbent or official schedule. Moreover, not considering noises and disturbances (assuming that they happen) does not only limit the economic value and increase the risk of schedule infeasibilities, but may also lead to the implementation of a completely different schedule rather than the real optimal schedule (considering the real state of the process, which includes all the changes resulted from disturbances and unexpected events), which reinforces the importance of online scheduling strategies for process optimization and control purposes. Some of the most relevant issues detected herein when neglecting disturbances on schedule operations are related to process infeasibilities, difficulties in tracking amounts and properties throughout the process, unspecified products, product giveaways, and inventory limits.

The proposed framework simulates an entire closed-loop scheduling solution, and effectively handles the triggered disturbances (either to mitigate/reduce their effects or to take advantage of new information), reduces inaccuracies and plant-model mismatches by maintaining and updating the state of the system, and provides a systematic approach to improve the scheduling implementation. We believe this approach is suitable for a large variety of problems as it is robust and reliable enough to handle several types of disturbances, considers both continuous and binary variables as well as nonlinear constraints (in an MINLP formulation) and can perform quick re-optimizations.

5.6 Supplementary Material

In the following we present the mathematical formulation for the blend scheduling problem defined in Figure 5.4. The objective function in Equation (5.1) maximizes the gross margin from product revenues by subtracting feedstocks costs. Variables for flows in process-units $x_{m,t}$ and arrows (between units) $x_{j,i,t}$, holdups for tanks $xh_{m,t}$, and properties $p_{j,t}$ are considered in the model. The indices i and j represent flows for inlet and outlet ports, respectively, whereas t represents the time steps. The port connections establish the material flows throughout the process. The sets I and J represent in- and out-port, respectively, while the set JI defines connecting flows between out- and in-ports. The sets M_R, M_{TK}, M_{BL}, and M_D are used for raw materials (feedstocks), tanks, blenders, and product demands, respectively. For $x \in \mathbb{R}^+$ and $y \in \{0,1\}$:

$$Max\, Z = \sum_{t}\left(\sum_{m\in M_D} price_{m,t}\, x_{m,t} - \sum_{m\in M_R} cost_{m,t}\, x_{m,t}\right) \qquad (5.1)$$

s.t.

$$\bar{x}^{L}_{j,i,t}\, y_{j,i,t} \le x_{j,i,t} \le \bar{x}^{U}_{j,i,t}\, y_{j,i,t} \quad \forall\ (j,i) \in JI, t \qquad (5.2)$$

$$\bar{x}^{L}_{m,t}\, y_{m,t} \le x_{m,t} \le \bar{x}^{U}_{m,t}\, y_{m,t} \quad \forall\ m \notin M_{TK}, t \qquad (5.3)$$

$$\overline{xh}^{L}_{m,t}\, y_{m,t} \le xh_{m,t} \le \overline{xh}^{U}_{m,t}\, y_{m,t} \quad \forall\ m \in M_{TK}, t \qquad (5.4)$$

$$\sum_{j\in J_S} x_{j,i,t} \ge \bar{x}^{L}_{m,t}\, y_{m,t} \quad \forall\, (i,m) \in M_D\, , t \qquad (5.5)$$

$$\sum_{j\in J_S} x_{j,i,t} \le \bar{x}^{U}_{m,t}\, y_{m,t} \quad \forall\, (i,m) \in M_D\, , t \qquad (5.6)$$

$$\sum_{i\in I_F} x_{j,i,t} \ge \bar{x}^{L}_{m,t}\, y_{m,t} \quad \forall\, (m,j) \in M_R, t \qquad (5.7)$$

$$\sum_{i\in I_F} x_{j,i,t} \le \bar{x}^{U}_{m,t}\, y_{m,t} \quad \forall\, (m,j) \in M_R, t \qquad (5.8)$$

$$\sum_{i\in I_S} x_{j,i,t} \ge \bar{x}^{L}_{m,t}\, y_{m,t} \quad \forall\, (m,j) \in M_{BL}\, , t \qquad (5.9)$$

$$\sum_{i\in I_S} x_{j,i,t} \le \bar{x}^{U}_{m,t}\, y_{m,t} \quad \forall\, (m,j) \in M_{BL}, t \qquad (5.10)$$

$$\sum_{j\in J_F} x_{j,i,t} \ge \bar{r}^{L}_{i,t}\, x_{m,t} \quad \forall\, (i,m) \in M_{BL}, t \qquad (5.11)$$

$$\sum_{j\in J_F} x_{j,i,t} \le \bar{r}^{U}_{i,t}\, x_{m,t} \quad \forall\, (i,m) \in M_{BL}, t \qquad (5.12)$$

$$\sum_{i\in S} x_{j,i,t} \ge \bar{r}^{L}_{j,t}\, x_{m,t} \quad \forall\, (m,j) \in M_{BL}, t \qquad (5.13)$$

$$\sum_{i \in I_S} x_{j,i,t} \leq \bar{r}_{j,t}^{U}\, x_{m,t} \quad \forall\, (m,j) \in M_{BL}, t \tag{5.14}$$

$$xh_{m,t} = xh_{m,t-1} + \sum_{j_{up} \in J_{TK}} x_{j_{up},i,t} - \sum_{i_{do} \in I_{TK}} x_{j,i_{do},t} \quad \forall\, (i,m,j) \in M_{TK}, t \tag{5.15}$$

$$\sum_{i \in I_{BL}} \sum_{j_{up} \in J_F} x_{j_{up},i,t} = \sum_{i_{do} \in I_S} x_{j,i_{do},t} \quad \forall\, m \in M_{BL}, t \tag{5.16}$$

$$x_{m,t}, x_{j,i,t}, xh_{m,t} \geq 0;\; y_{j,i,t}, y_{m,t} \in \{0,1\} \tag{5.17}$$

Equations (5.2) to (5.4) control the flows $x_{j,i,t}$, the yields of units $x_{m,t}$ (except tanks), and the inventory of tanks $xh_{m,t}$. For instance, if the binary variable $y_{j,i,t}$ in Equation (5.2) is true, then the flow of the arrow $x_{j,i,t}$ vary between its bounds ($\bar{x}_{j,i,t}^{L}$ and $\bar{x}_{j,i,t}^{U}$). It is similar for the binary variables of unit-operations $y_{m,t}$ ($m \notin M_{TK}$) with respect to their bounds ($\bar{x}_{m,t}^{L}$ and $\bar{x}_{m,t}^{U}$) in Equation (5.3) and for the binary variables $y_{m,t}$ ($m \in M_{TK}$) with their respective bounds ($\overline{xh}_{m,t}^{L}$ and $\overline{xh}_{m,t}^{U}$) in Equation (5.4).

Equations (5.5) to (5.9) establish bounds ($\bar{x}_{m,t}^{L}$ and $\bar{x}_{m,t}^{U}$) for the sum of the flows incoming to and outgoing from port whenever the respective binary variable $y_{m,t}$ is active. Equations (5.5) and (5.6) are specific for flows from storage tanks ($j \in J_S$) to demand sinks $(i,m) \in M_D$, Equations (5.7) and (5.8) define flows from raw material sources $(m,j) \in M_R$ to feed tanks ($i \in I_F$), and Equations (5.9) and (5.10) represent from the blenders ($m \in M_{BL}$) to the downstream storage tanks ($i \in I_S$).

Equations (5.11) and (5.12) impose bounds for the incoming flows i of unit-operation m ($m \notin$ MTK) by using their respective lower and upper inverse yields ($r_{i,t}^{L}$ and $r_{i,t}^{U}$). Similarly, Equations (5.13) and (5.14) impose bounds for outgoing flows to the unit-operation m ($m \notin M_{BL}$) by their lower and upper yields ($r_{j,t}^{L}$ and $r_{j,t}^{U}$). Besides, although a blender can be connected to several sinks, only one outlet flow is typically allowed in industrial operations.

The material balance for tanks ($m\in$MTK) in Equation (5.15) considers previous holdups $xh_{m,t-1}$ and the material inlets and outlets of tanks. Blenders MBL are treated as continuous units so that Equation (5.16) are material balances to enforce no accumulation in these units.

5.6.1 Logistics Problem: MILP Blend Scheduling

The logistics problem includes Equations (5.1 to 5.17) for the UOPSS flowsheet formulation (Figure 5.3), plus Equations (5.18 to 5.32) that involves: a) multi-use of objects; b) minimum operational time and zero downtime of units; c) transition constraints and selection of operational modes; d) temporal transitions of sequence-dependent cycles for unit-operations; and e) factor-flow balance (KELLY, MENEZES, and GROSSMANN, 2018).

Equation (5.18) implies that a connecting binary variable $y_{j,i,t}$ is active when the binaries of their connected unit-operations m_{up} and m are also active. Equation (5.19) enforces that at most one operational mode for a physical unit-operation m (as $y_{m,t}$ for operational modes) is allowed simultaneously, in which U_m is the set for distinct unit-operations m within the same physical unit.

$$y_{m_{up},t} + y_{m,t} \geq 2y_{j_{up},i,t} \;\; \forall \left(m_{up}\;, j_{up}, i, m\right), t \tag{5.18}$$

$$\sum_{m \in U_m} y_{m,t} \leq 1 \;\; \forall\, t \tag{5.19}$$

The temporal transition in Equations (5.20) and (5.21) control the operations for semi-continuous blenders. The binary variable $y_{m,t}$ manages the start-up ($zsu_{m,t}$) switch-over ($zsw_{m,t}$) and shut-down variables ($zsd_{m,t}$), which are relaxed in the interval [0,1]. Equation (5.22) guarantees the integrality of the relaxed variables.

$$y_{m,t} - y_{m,t-1} - zsu_{m,t} + zsd_{m,t} = 0 \;\; \forall\, m \in M_{BL}, t \tag{5.20}$$

$$y_{m,t} + y_{m,t-1} - zsu_{m,t} - zsd_{m,t} \; - 2zsw_{m,t} = 0 \;\; \forall\, m \in M_{BL}, t \tag{5.21}$$

$$zsu_{m,t} + zsd_{m,t} + zsw_{m,t} \leq 1 \;\; \forall\, m \in M_{BL}, t \tag{5.22}$$

In Equations (5.23) and (5.24), the parameters $USE^{L}_{j,t}$ e $USE^{U}_{j,t}$ manage the out-ports ($j \in J_{USE}$) from their respective in-ports i_{do} to avoid simultaneous flow from a blender to two or more distinct tanks in the same time step. Equations (5.25) and (5.26) limit $USE^{L}_{i,t}$ e $USE^{U}_{i,t}$ to in-ports ($i \in I_{USE}$) from their respective out-ports j_{up}, controlling the number of simultaneous flows to the same unit.

$$\sum_{i_{do}} y_{j,i_{do},t} \geq USE^{L}_{j,t}\, y_{m,t} \;\; \forall\, j \in J_{USE}, t \tag{5.23}$$

$$\sum_{i_{do}} y_{j,i_{do},t} \leq USE^{U}_{j,t}\, y_{m,t} \;\forall\, j \in J_{USE}, t \tag{5.24}$$

$$\sum_{j_{up}} y_{j_{up},i,t} \geq USE^{L}_{i,t}\, y_{m,t} \;\forall\, i \in I_{USE}, t \tag{5.25}$$

$$\sum_{j_{up}} y_{j_{up},i,t} \leq USE^{U}_{i,t}\, y_{m,t} \forall\, i \in I_{USE}, t \tag{5.26}$$

Equations (5.27) to (5.29) model the uptime considering UPT^{U}as the upper bound, Δt as time step, t_{end} as the end of the horizon, and n_p as the number of periods (KELLY and ZYNGIER, 2007).

$$\sum_{tt=1}^{UPT^{L}-1} zsu_{m,t-\mathrm{tt}} \leq y_{m,t} \;\forall\, m \in M_{BL}, t > 1 \mid t - tt \geq n_p \tag{5.27}$$

$$\sum_{tt=t}^{\frac{UPT^{U}}{\Delta t}} y_{m,tt} \leq \frac{UPT^{U}}{\Delta t} \;\forall\, m \in M_{BL}, t < t_{end} - UPT^{U} \tag{5.28}$$

$$\Delta t \sum_{t} zsu_{m,t} \leq n_p \;\forall\, m \in M_{BL} \tag{5.29}$$

To reduce the MILP-NLP gap in the proposed decomposition and to avoid poor NLP results or even infeasibilities in this stage, Equation (5.30) has been used to linearize non-convex NLP blending equations in the MILP model by considering the property nonlinear variables as parameters referred to as factors $f_{i,p,t}$. For further details please see Kelly, Menezes, and Grossmann (2018).

$$\sum_{i \in I_{BL}} f_{i,p,t} \sum_{j_{up} \in J_{TK}} x_{j_{up},i,t} = f_{j,p,t} \sum_{i \in I_{TK}} x_{j,i,t} + x_{j^F,p,t} \quad \forall\, j, j^F, p, t \tag{5.30}$$

5.6.2 Quality Problem: NLP Blend Scheduling

The NLP mathematical model includes Equations (5.1) to (5.30) from the UOPSS flowsheet formulation (by fixing the binary variables from the MILP solution) and Equations (5.31) to (5.36) for the blending constraints. Considering p as property (specific gravity, sulfur content) and v and w as volume- and weight-based properties, respectively, Equation (5.31) defines the volume-based properties ($p \in P_v$) of in-ports i for $i \notin I_{TK}$. Equation (5.32) is used for mass-based properties ($p \in P_w$) such as sulfur concentration.

$$v_{i,p,t} \sum_{j_{up}} x_{j_{up},i,t} = \sum_{j_{up}} v_{j_{up},p,t} x_{j_{up},i,t} \quad \forall\, i \notin I_{TK}, p \in P_v, t \tag{5.31}$$

$$w_{i,p,t} \sum_{j_{up}} v_{j_{up},p=sg,t}\, x_{j\prime,i,t} = \sum_{j_{up}} w_{j_{up},p,t}\, v_{j_{up},p=sg,t}\, x_{j_{up},i,t} \quad \forall\, i \notin I_{TK}, p \in P_w, t \tag{5.32}$$

Equations (5.33) and (5.34) represent the quality balances of volume- and mass-based properties for tanks, respectively. The quality variable for the out-port of a tank unit-operation ($m \in M_{TK}$) is the quality in the tank itself as defined by Equations (5.35) and (5.36).

$$v_{m,p,t}\, xh_{m,t} = v_{m,p,t-1}\, xh_{m,t-1} + \sum_{j_{up}} v_{j_{up},p,t}\, x_{j_{up},i,t} - v_{m,p,t} \sum_{i_{do}} x_{j,i_{do},t} \quad \forall\, (i,m,j) \in M_{TK}, p \in P_v, t \tag{5.33}$$

$$
\begin{aligned}
& w_{m,p,t}\ v_{m,p=sg,t}\ xh_{m,t} \\
&\qquad = w_{m,p,t-1}\ v_{m,p=sg,t-1}\ xh_{m,t-1} + \sum_{j_{up}} w_{j_{up},p,t}\ v_{j_{up},p=sg,t}\ x_{j_{up},i,t} \\
&\qquad - w_{m,p,t}\ v_{m,p=sg,t} \sum_{i_{do}} x_{j,i_{do},t}\ \ \forall\ (i,m,j) \in M_{TK}, p \in P_w, t \qquad (5.34)
\end{aligned}
$$

$$v_{j,p,t} \ = v_{m,p,t}\ \forall\ (m,j) \in M_{TK}, p \in P_v, t \qquad (5.35)$$

$$w_{j,p,t} \ = w_{m,p,t}\ \forall\ (m,j) \in M_{TK}, p \in P_w, t \qquad (5.36)$$

5.7 Nomenclature

Subscripts

i: in-port

i_{do}: in-port i downstream to out-port *j*

j: out-port

j_{up}: out-port upstream to in-port *i*

m: unit-operations

p: properties

t: time periods

u: units

Superscripts

L: lower bound

U: upper bound

Sets

I_{TK}: in-port of tanks

I_{USE}: in-port with multi-use constraint

J_{TK}: out-port of tanks

J: out-port with multi-use constraint

M_{BL}: unit-operation of blenders

M_{FD}: unit-operation of feedstocks supply

M_{FP}: unit-operation of demands (final products)

M_{TK}: unit-operation of tanks

P_v: volume-base property

P_w: mass-base property

U_m: unit of multiple operations

Parameters

Δt: time step

$f_{j,p,t}$: factor parameters

n_p: total number of time periods

$r_{i,t}^L$: inverse yield lower bound in the in-port i of unit m at time t

$r_{i,t}^U$: inverse yield upper bound in the in-port i of unit m at time t

$r_{j,t}^L$: yield lower bound in the out-port j of unit m at time t

$r_{j,t}^U$: yield upper bound in the out-port j of unit m at time t

t_{end}: end of time horizon

$USE_{i,t}^L$: lower bound for number of out-ports i from the blender at time t

$USE_{i,t}^U$: upper bound for number of the out-ports i from the blender at time t

$USE_{j,t}^L$: lower bound for number of the out-ports j to the blender at time t

$USE_{j,t}^U$: upper bound for number of the out-ports j to the blender at time t

UPT^L: uptime lower bound for blender

UPT^U: uptime upper bound for blender

$\bar{v}_{i,p,t}^L$: lower bound for property p (volume-based) in the in-port i at time t

$\bar{v}^{U}_{i,p,t}$: upper bound for property p (volume-based) in the in-port i at time t

$\bar{v}^{L}_{j,p,t}$: lower bound for property p (volume-based) in the out-port j at time t

$\bar{v}^{U}_{j,p,t}$: upper bound for property p (volume-based) in the out-port j at time t

$\bar{w}^{L}_{i,p,t}$: lower bound for property p (mass-based) in the in-port i at time t

$\bar{w}^{U}_{i,p,t}$: upper bound for property p (mass-based) in the in-port i at time t

$\bar{w}^{L}_{j,p,t}$: lower bound for property p (mass-based) in the out-port j at time t

$\bar{w}^{U}_{j,p,t}$: upper bound for property p (mass-based) in the out-port j at time t

$\bar{x}^{L}_{m,t}$: lower bound for flow of unit-operation m at time t

$\bar{x}^{U}_{m,t}$: upper bound for flow of unit-operation m at time t

$\overline{xh}^{L}_{m,t}$: lower bound for flow of unit-operation m at time t

$\overline{xh}^{U}_{m,t}$: upper bound for flow of unit-operation m at time t

$xh^{EMPTY}_{m,t}$: lower bound for flow of unit-operation m at time t

$xh^{FULL}_{m,t}$: upper bound for flow of unit-operation m at time t

$\bar{x}^{L}_{j,i,t}$: lower bound for flow from j to i at time t

$\bar{x}^{U}_{j,i,t}$: upper bound for flow from j to i at time t

Binary Variables

$y_{m,t}$: unit-operation setup of m at time t

$y_{j,i,t}$: unit-operation-port-unit-operation-port setup between j and i at time t

Continuous Variables

$p_{i,p,t}$: generic property p in the out-port i of m at time t

$p_{j,p,t}$: generic property p in the out-port j of m at time t

$p_{m,p,t}$: generic property p in the unit-operation m of tanks at time t

$v_{i,p,t}$: property p (volume-based) in the out-port i of m at time t

$v_{j,p,t}$: property p (volume-based) in the out-port j of m at time t

$v_{m,p,t}$: property p (volume-based) in the unit-operation m of tanks at time t

$w_{i,p,t}$: property p (mass-based) in the out-port i of m at time t

$w_{j,p,t}$: property p (mass-based) in the out-port j of m at time t

$w_{m,p,t}$: property p (mass-based) in the unit-operation m of tanks at time t

$x_{m,t}$: flow of unit-operation m at time t

$xh_{m,t}$: holdup of unit-operation m at time t

$x_{j,i,t}$: unit-operation-port-unit-operation-port flow between j and i at time t

$zsu_{m,t}$: start-up of unit-operation m at time t

$zsw_{m,t^{t}}$: switchover-to-itself of unit-operation m at time t

$zsd_{m,t}$: shutdown of unit-operation m at time t

Acronyms

BL: continuous blender unit

D1 to D3: final products pool

F1 to F4: feed tanks

LP: linear programming

MILP: mixed-integer linear programming

MINLP: mixed-integer nonlinear programming

NLP: nonlinear programming

R1 to R4: feedstock resources

SG: specific gravity

S: sulfur

S1 to S3: storage tanks

UOPSS: unit-operation-port superstructure

5.8 References

ADHITYA, A.; SRINIVASAN, R.; KARIMI, I. A. A model-based rescheduling framework for managing abnormal supply chain events. Computers & Chemical Engineering, v. 31, n. 5-6, p. 496-518, 2007.

BRUNAUD, B.; PEREZ, H. D.; AMARAN, S.; BURY, S.; WASSICK, J.; GROSSMANN, I. E. Batch scheduling with quality-based changeovers. Computers & Chemical Engineering, v. 132, p. 106617, 2020.

CHARITOPOULOS, V. M.; PAPAGEORGIOU, L. G.; DUA, V. Closed-loop integration of planning, scheduling and multi-parametric nonlinear control. Computers & Chemical Engineering, v. 122, p. 172-192, 2019.

COTT, B.; MACCHIETTO, S. Minimizing the effects of batch process variability using online schedule modification. Computers & Chemical Engineering, v. 13, n. 1, p. 105–113, 1989.

DU, J.; PARK, J.; HARJUNKOSKI, I.; BALDEA, M. A time scale-bridging approach for integrating production scheduling and process control. Computers & Chemical Engineering, v. 79, p. 59–69, 2015.

FRANZOI, R. E.; MENEZES, B. C.; KELLY, J. D.; GUT, J. W. Effective scheduling of complex process-shops using online parameter feedback in crude oil refineries. Computer Aided Chemical Engineering, v. 44, p. 1279-1284, 2018.

GUPTA, D.; MARAVELIAS, C. T. On Deterministic Online Scheduling: Major Considerations, Paradoxes and Remedies. Computers & Chemical Engineering, v. 94, p. 312-330, 2016.

GUPTA, D.; MARAVELIAS, C. T.; WASSICK, J. M. From Rescheduling to Online Scheduling. Chemical Engineering Research and Design, v. 116, p. 83-97, 2016.

GUPTA, D.; MARAVELIAS, C. T. A General State-Space Formulation for Online Scheduling. Processes, v. 5, n. 4, p. 69, 2017a.

GUPTA, D.; MARAVELIAS, C. T. On the Design of an Online Scheduling Algorithm. AICHE Meeting, Minneapolis, MN, 2017b.

HONKOMP, S.; MOCKUS, L.; REKLAITIS, G. A framework for schedule evaluation with processing uncertainty. Computers & Chemical Engineering, v. 23, n. 4, p. 595–609, 1999.

HUERCIO, A.; ESPUNA, A.; PUIGJANER, L. Incorporating on-line scheduling strategies in integrated batch production control. Computers & Chemical Engineering, v. 19, p. 609–614, 1995.

JANAK, S. L.; FLOUDAS, C. A.; KALLRATH, J.; VORMBROCK, N. Production scheduling of a large-scale industrial batch plant. II. Reactive scheduling. Industrial & Engineering Chemistry Research, v. 45, n. 25, p. 8253–8269, 2006.

KANAKAMEDALA, K. B.; REKLAITIS, G. V.; VENKATASUBRAMANIAN, V. Reactiveschedule modification in multipurpose batch chemical plants. Industrial & Engineering Chemistry Research, v. 33, n. 1, p. 77–90, 1994.

KATRAGJINI, K.; VALLADA, E.; RUIZ, R. Flow shop rescheduling under different types of disruption. International Journal of Production Research, v. 51, n. 3, p. 780-797, 2013.

KELLY, J. D. The Unit-Operation-Stock Superstructure (UOSS) and the Quantity-Logic-Quality Paradigm (QLQP) for Production Scheduling in The Process Industries. In Multidisciplinary International Scheduling Conference Proceedings: New York, United States, p. 327-333, 2005.

KELLY, J. D.; ZYNGIER, D. An Improved MILP Modeling of Sequence-Dependent Switchovers for Discrete-Time Scheduling Problems. Industrial & Engineering Chemistry Research, v. 46, p. 4964, 2007.

KELLY, J. D.; MENEZES, B. C.; ENGINEER, F.; GROSSMANN, I. E. Crude oil Blend Scheduling Optimization of an Industrial-Sized Refinery: A Discrete-Time Benchmark. In Foundations of Computer Aided Process Operations, FOCAPO, Tucson, AR, United States, 2017.

KELLY, J. D.; ZYNGIER, D. Continuously improve the performance of planning and scheduling models with parameter feedback. In Foundations of Computer Aided Process Operations, FOCAPO, Boston, MA, United States, 2008.

KELLY, J. D.; MENEZES, B. C.; GROSSMANN, I. E. Successive LP approximation for non-convex blending in MILP scheduling optimization using factors for qualities in the process industry. Industrial & Engineering Chemistry Research, v. 57, n. 32, p. 11076-11093, 2018.

KELLY, J. D.; MENEZES, B. C. Industrial Modeling and Programming Language (IMPL) for off- and on-line optimization and estimation applications. In *Optimization in Large Scale Problems*, p. 75-96, 2019.

KOPANOS, G. M.; PISTIKOPOULOS, E. N. Reactive scheduling by a multiparametric programming moving horizon framework: a case of a network of combined heat and power units. Industrial & Engineering Chemistry Research, v. 53, n. 11, p. 4366–4386, 2014.

LARSEN, R.; PRANZO, M. A framework for dynamic rescheduling problems. International Journal of Production Research, p. 1-18, 2018.

LI, Z.; IERAPETRITOU, M. G. Process scheduling under uncertainty: review and challenges. Computers & Chemical Engineering, v. 32, n. 4–5, p. 715–727, 2008a.

MCKAY, K. N.; WIERS V. C. S. Unifying the Theory and Practice of Production Scheduling. Journal of Manufacturing Systems, v. 18, n. 4, p. 241–255, 1999.

MÉNDEZ, C.A.; CERDÁ, J. Dynamic scheduling in multiproduct batch plants. Computers & Chemical Engineering, v. 27, n. 8, p. 1247–1259, 2003.

MENEZES, B.C.; KELLY, J.D.; GROSSMANN, I.E. Phenomenological decomposition heuristic for process design synthesis of oil-refinery Units. Computer Aided Chemical Engineering; v. 37, p. 1877-1882, 2015.

MENEZES, B. C.; KELLY, J. D.; LEAL, A. G. Identification and Design of Industry 4.0 Opportunities in Manufacturing: Examples from Mature Industries to Laboratory Level Systems. IFAC-PapersOnLine, v. 52, n. 13, p. 2494-2500, 2019.

MENEZES, B. C.; KELLY, J. D, LEAL, A. G.; LE ROUX, G. C. Predictive, prescriptive and detective analytics for smart manufacturing in the information age. IFAC-PapersOnLine, v. 52, n. 1, p. 568-573, 2019.

NIE, Y.; BIEGLER, L. T.; WASSICK, J. M.; VILLA, C. M. Extended discrete–time resource task network formulation for the reactive scheduling of a mixed batch/continuous process. Industrial & Engineering Chemistry Research; v. 53, n. 44, p, 17112–17123, 2014.

RODRIGUES, M.; GIMENO, L.; PASSOS, C.; CAMPOS, M. Reactive scheduling approach for multipurpose chemical batch plants. Computers & Chemical Engineering, v. 20, S1215–S1220, 1996.

STEVENSON, Z.; FUKASAWA, R.; RICARDEZ-SANDOVAL, L. Evaluating periodic rescheduling policies using a rolling horizon framework in an industrial-scale multipurpose plant. Journal of Scheduling, v. 23, n. 3, 397-410, 2020.

SUBRAMANIAN, K.; MARAVELIAS, C.T.; RAWLINGS, J.B. A state-space model for chemical production scheduling. Computers & Chemical Engineering, v. 47, p. 97-110, 2012.

VIEIRA, G. E.; HERRMANN, J. W.; LIN, E. Rescheduling manufacturing systems: a framework of strategies, policies, and methods. Journal of scheduling, v. 6, n. 1, p. 39-62, 2003.

VIN, J. P.; IERAPETRITOU, M. G. A new approach for efficient rescheduling of multiproduct batch plants. Industrial & Engineering Chemistry Research 39, 4228–4238, 2000.

VIN, J. P.; IERAPETRITOU, M. G. Robust short-term scheduling of multiproduct batch plants under demand uncertainty. Industrial & Engineering Chemistry Research; v. 40, p. 4543–4554, 2001.

ZHUGE, J.; IERAPETRITOU, M. G. Integration of scheduling and control with closed loop implementation. Industrial & Engineering Chemistry Research; v. 51, n. 25, p. 8550–8565, 2012.

6

Cutpoint Temperature Surrogate Modeling for Refinery Applications[5]

For high-performance operations in crude oil refinery processing, it is important to properly determine yields and properties of output streams from distillation units. To address such complex representation, a cutpoint temperature modeling framework is proposed in this Chapter, which uses a coefficient setup MIQP (mixed-integer quadratic programming) technique to determine optimizable surrogate models to correlate independent X variables (crude oil compositions, temperatures) to dependent Y variables (yields and properties of distillates). The X inputs are randomly generated by Latin Hypercube Sampling (LHS) and the experiments to obtain the synthetic Y outputs are simulated using the well-known conventional and improved swing-cut methods. By using these optimizable surrogate models (which are suitable to handle continuous data from the process) with measurement feedback (for adjustments and improvements), distillation outputs can be continuously updated in an online fashion. The proposed approach successfully builds accurate surrogates for the distillation unit, which can be embedded into complex planning and scheduling environments. Moreover, this MIQP surrogate identification technique may also be applied to other types of downstream process optimization problems such as reacting and blending unit-operations, as well as other separating processes.

6.1 Introduction

Crude oil distillation units (CDUs) are complex sets of towers designed, operated, and controlled to separate liquid hydrocarbon feedstocks or crude oil raw materials into intermediate fractions or distillates according to boiling range temperatures (RIAZI, 2005). As these distilled streams are processed in downstream unit-operations and

[5] This chapter is based on the following manuscript:
FRANZOI, R. E.; MENEZES, B. C.; KELLY, J. D.; GUT, J. A. W.; GROSSMANN, I. E. Cutpoint Temperature Surrogate Modeling for Distillation Yields and Properties. Industrial and Engineering Chemistry Research, v. 59, n. 41, p. 18616-18628, 2020.

blended into final products, for an overall high-performance operation of the refinery, it is important to precisely calculate the distillation unit product yields and properties as a function of feed quality and operating conditions (FU and MAHALEC, 2015). Both rigorous and surrogate models can be used to predict product amounts and properties of distillation processes for planning, scheduling, multi-unit coordinating, and real-time optimization (RTO) environments. The rigorous, mechanistic, physics-based, first principles, white-box or engineering-based modeling typically consider molar, mass, energy, separation, and equilibrium balances in the distillation columns. Compositions, flows and processing conditions may be accurately determined, but at a high computational cost, and with convergence issues for their application in large-scale integrated problems. Conversely, non-rigorous, black-box or empirical-based modeling can use surrogate or simplified shortcut correlations based on measured and/or synthetic data using regression techniques. Due to their simplicity, effectiveness and acceptable accuracy within a localized region, surrogate modeling is commonly used for process optimization in crude oil refineries (LI, HUI, and LI, 2005).

In order to calculate CDU yields and cold-flow properties using non-rigorous models, one can use the temperature distribution from the crude oil true boiling point (TBP) curve, which represents how the crude oil yields and properties (such as specific gravity, sulfur content, etc.) vary with the distillation temperature (FU and MAHALEC, 2015; KELLY, MENEZES, and GROSSMANN, 2014). The TBP curve is related to the crude oil assay, which provides data on the quantities and qualities of each discretized temperature cut or micro-cut range through their distillation temperature distribution (MENEZES, KELLY, and GROSSMANN). Due to operational limitations and inefficiencies regarding reflux and re-circulation rates, number of stages, etc., there is a well-known overlap in the TBP boiling ranges of adjacent fractions or compounds in any physical distillation column (LI, HUI, and LI, 2005). Therefore, this non-sharp fractionation between adjacent distillates should be considered to properly formulate cutpoint optimization methods.

By observational evidence in the oil refinery, the bulk quality of raw material composition (assay) of crude oils to be processed typically determines around 80 to 90% of the amounts and properties of the distillates, whereas the remaining part is determined by its operational variables, such as internal reflux rates, system pressure profile, steam flows of side-strippers, pump-around re-circulation rates, parallel split ratios of pre-heat exchanger trains, feed and location temperatures in the furnace and tower, tray and/or packing characteristics, etc. Moreover, lower and upper bounds of quality specifications (circa 30 distinct types of properties, such as specific gravity, sulfur concentration, acidity, carbon residue content, etc.) can be considered in the cutpoint temperature model. Due to uncertainties in the transformation equations, feed compositions and other processing data, high-fidelity modeling becomes unrealistic to be included in problems such as the blend scheduling and processing optimization of crude oils.

In addition to the above considerations, we are currently moving towards a more complex process optimization within the well-known Industry 4.0 paradigm (JOLY et al., 2018), driven by advancements in decision-making modeling, computer-aided capabilities, connectivity and solution algorithms. In this direction, big data (CHEN et al., 2014; CHIANG, LU, and CASTILLO, 2017; MAKTOUBIAN, GHASEMPOUR-MOUZIRAJI, AND NOORI, 2020; WU et al., 2013), data-driven models (BOUKOUVALA et al., 2016; LI et al., 2016; YU, 2019; AHMAD et al., 2020; MCBRIDE, SANCHES MEDINA, and SUNDMACHER, 2020), and machine learning techniques (ANDERSON, 2017; BECK et al., 2016; WILSON and SAHINIDIS, 2017) have been used in a wide range of engineering problems: a) handling large, complex, and unreliable data sets; b) as a better or more efficient alternative to solve particular problems; and c) solving problems which are either intractable or that require faster solutions for specific applications. Production and process optimization for the crude oil refinery industry typically handle large, complex, nonlinear, and non-convex models. Thus, modeling and optimizing a fully integrated petroleum refinery problem is still not yet attainable in terms of complexity and uncertainty. Data from the plant is not always accurate and contains various noise levels, and model-plant mismatches can often impose issues regarding model infeasibilities and solution implementation. Employing online measurements in the entire plant can help to mitigate errors and provide better fidelity; however, that is typically not as effective as identifying and estimating better local models. Hence, error propagation becomes significant, especially because of the nonlinearities associated with crude oil refinery problems.

The framework thus proposed herein establishes surrogate models based on process analytics and machine learning techniques to reduce uncertainties around the determination of cutpoints in distillation units when incrementally optimizing and controlling the stream flow, yield and properties drawn from the CDU at a certain section in the tower. The main contributions of the proposed model are: a) it is as representative and accurate as typical models used for planning and scheduling environments such as the swing-cut methods (LI, HUI, and LI, 2005; MENEZES, KELLY, and GROSSMANN, 2014); b) real data from the plant can be embedded into the model to give it self-adjustability and self-improvability over time, minimizing the impact of uncertainties and disturbances in the process by correlated parameter updating (with re-estimation of coefficients); c) it is small in size, with fewer equations and degrees of freedom than the swing-cut models, and can be properly integrated into planning, scheduling, coordinating and RTO environments with minimal increase in the simulation and optimization effort.

The novelty of our approach relies on using data available from any reliable source (e.g., process measurements, rigorous simulator) to predict distillation unit outputs. This is especially helpful to handle plant-model mismatches and uncertainties in the process, as the accuracy of the predictions can be continuously improved by using new data, so that our method depends on the quality and volume of data available. Ideally, it would be as accurate as the original source of data. If a rigorous simulator is used to generate high fidelity data, simple yet accurate surrogates built with this methodology can be embedded in decision-making control, modeling, and optimization environments (which would be computationally too expensive when using the rigorous simulator itself). In this work, instead of comparing other methods from the cutpoint optimization literature regarding their accuracy, we demonstrate how to build surrogate models for the distillation unit that can accurately predict its processing behavior. Hence, if that data accurately represents the distillation behavior, so does our model. That might be useful for the integration of proxy or surrogate models into large scale applications, such as the refinery planning and scheduling problems.

The outline of this chapter is as follows. Section 6.2 presents an overview of the cutpoint optimization approaches reported in the literature. The problem statement is described in Section 6.3. The proposed algorithm for identifying surrogate process analytics formulae, known as coefficient-setup technique, is presented in Section 6.4, whereby the swing-cut modeling (conventional and improved) as well as the inputs-outputs or X-Y data blocks or sets are given in the Supplementary Material. Examples using the proposed algorithm for linear and interaction terms of bilinear correlations of the surrogate model are compared in Section 6.5. The conclusions and future work on the topic are discussed in Section 6.6.

6.2 Previous Shortcut Distillation Methods

The simplest approach to model a distillation unit is to use fixed yield and property values for its outputs, specifying the increments of the discretized CDU fractions and using crude oil assay data to calculate these fractions (BROOKS, VAN WALSEM, and DRURY, 1999). As this approach uses sharp fractionation to calculate the CDU yields, it does not rigorously compute the molecular behavior considering the non-perfect or non-sharp separation. A variation in the fixed yield model, known as multiple fixed yields, allows multiple and usually hypothetical operational modes in the process, whereby each mode is related to a distinct crude oil assay and hence, different yields and properties for the final cuts. This method introduces an additional degree of freedom, which is represented by binary variables. Such improvement allows only a small number of different solutions (equal to the number of operational modes), despite the feasibility of intermediate solutions that lie between the range of pre-defined modes. Furthermore, crude oil refinery optimization problems are typically highly nonlinear and non-convex due to the blending of streams and inventories throughout the processing network of unit-operations and intermediate tanks. The introduction of binary variables would lead to a non-convex mixed-integer nonlinear programming (MINLP) problem, which is difficult to solve for medium to large-scale cases. Brooks et al.19 optimized product yields for a crude oil distillation unit by introducing eight pre-defined modes of operation. Each mode had a distinct choice of cutpoints, and their approach allowed blending the outputs of distinct modes to achieve required yields and properties of the final distillates. To handle the complex nature of the problem, the authors employed tabulated values of yields and properties of intermediate products as a linear model for the distillation unit.

To better predict crude oil distillation unit outputs, there are models that consider non-sharp fractionation by including volume and/or mass variations in the cutpoints. A traditional empirical approach is known as delta-based or shift-vector modeling and uses small increments for product deviations in the TBP curve representing only first-order or linear effects. The swing-cut methods (LI, HUI, and LI, 2005; MENEZES, KELLY, and GROSSMANN, 2013; ZHANG, ZU, and TOWLER, 2001) are examples that require both the TBP range for each product and estimating the size of each swing-cut. This information can be combined with the crude oil quality in the respective TBP range to calculate the properties of each distillate (FU and MAHALEC, 2015). In addition, the swing-cut method creates additional degrees of freedom by optimizing the amount of each swing-cut to the lighter and heavier final distillates. The swing-cut methods are commonly used for distillation unit modeling due to their simplicity (GUERRA and LE ROUX, 2011) and improvements when compared to the fixed yields method.

Zhang et al. (2001) proposed a swing-cut method that considers operational conditions and feed properties in the distillation unit as variables in the optimization. However, this method considers the properties of adjacent distilled products fixed regardless of the amounts and properties of the swing-cut splits added to the distillates, failing to represent the high nonlinearity of the distillation process. The weight transfer ratio method (WTR), proposed by Li et al. (2005), considers crude oil characteristics and product yields and qualities in simplified empirical nonlinear models. The authors used an empirical procedure to calculate the mass transfer rates of each product in the CDU and to determine the size of each swing-cut. In addition, the authors used regression models based on the properties of the feed load to consider the variation of properties in each swing-cut. However, due to the possibility of processing more than one type of crude oil simultaneously, additional procedures are required to calculate the TBP curve of the crude oil mixture.

To efficiently deal with the variation of properties within the swing-cut, Menezes et al. (2013) improved the traditional or conventional swing-cut (CSW) method by dividing each swing-cut into light and heavy fractions with different qualities. The model uses arbitrary 10 °C increments for micro-cuts or pseudo-components of crude oil assay distributions and given initial and final boiling point temperature ranges of internal sections of the distillation columns to calculate the transformation of crude oils micro-cuts to the tower internal fractions or macro-cuts formed by pre-defined sections (fuel gas, LPG, naphtha, swing-cut, etc.). The properties of each light and heavy swing-cuts are calculated individually using interpolated quality information regarding their respective splits and the hypothetical light and heavy interfaces (blend of the micro-cuts in neighbor internal fractions or macro-cuts). This method adds property information on both light and heavy split fractions, in addition to the flow variables and nonlinear balance constraints to determine the final cuts (final products). The improved swing-cut (ISW) method accurately predicts the volumes and properties of distillation unit outputs, and can improve the selection of both crude oils and swing-cuts to provide specified final products more efficiently.

Alattas, Grossmann, and Palou-Rivera (2011) proposed a nonlinear model for the distillation unit to be used in a production planning environment. The distillation unit is represented as two-phase separation flash towers operating in series, using the Heaviside step function for modeling the gas-liquid equilibrium in the stripping and rectifying sections of the flashes. Fractionation indices (FI) are introduced for both gas and liquid layers of the flashes to be calculated with crude oil assay micro-cuts or pseudo-component distribution, as well as characteristics of the columns such as temperature ranges of the products (as initial and final boiling temperatures of the distillates in the bottom stream of the flashes). The model uses the FI values and molar balances to determine distillation tower operations and outputs more accurately than conventional swing-cut models. Alattas, Grossmann, and Palou-Rivera (2012) improved their previous methodology by formulating the disjunction of the fractionation index with mixed-integer constraints by formulating the selection of the stripping and rectifying sections considering a binary variable for each FI of the flashes, leading to a faster and more robust model. The authors also extended the application of their model to a multiperiod refinery planning problem based on an MINLP formulation. The nonlinearities are related to the blend of streams for crude oil selection and distilled product destinations, as well as to the vapor pressure calculation using reduced temperature .

Mahalec and Sanchez (2012) proposed a hybrid model to optimize distillation unit towers based on first principles with traditional mass and energy balances per section of the towers (THIELE and GEDDES, 1933) and statistical models based on data. In this approach, operational variables are correlated to product distillation curves by using partial least-square (PLS) models with datasets generated in rigorous simulation tools. The results of this procedure are used to estimate the deviation between initial and final boiling temperatures of the evaporation curves of the distillates. By partially relying on a statistical modeling, the method manages to reduce the prediction error of the fractions in the distillation unit. However, the increment used to set the cutpoints (between 14 and 67 °C) may not be small enough to give sufficiently good accuracy.

With measured values of final distillates, hydraulic amounts and evaporation curves in ASTM, TPB (true boiling point), or simulated distillation (SD) distributions, Kelly et al. (2014) used a monotonic interpolation method that avoids Runge’s phenomenon (oscillation at the edges of an interval when constructing a polynomial interpolant of high degree) to define cutpoint optimization by variations in yields and evaporation curves of the distillates. First, an initial procedure uses analytical expressions to convert experimental methods of ASTM or SD into TBP temperatures. Second, monotonic interpolation converts the TBP temperatures into cumulative evaporations. Third, a cutpoint temperature optimization is performed to adjust the front and back end of each distillation curve component using measured flow and laboratory ASTM distillation data. Four case studies were provided, in which good agreement is shown between predicted and real blending properties. This distillation blending and cutpoint temperature optimization can be used to integrate blend-shops and crude oil distillation units whereby interpolated distillation curves of the CDU streams are mixed with other streams that blend linearly in TBP by mass (e.g., sulfur concentration) or volume (e.g., specific gravity). The blend is then converted back to ASTM using another monotonic interpolation to find the initial and final boiling points of each distillate.

Fu, Sanchez, and Mahalec (2015) proposed a hybrid model to optimize a three-tower distillation unit. Partial least squares from the feed TBP curve and operational conditions were used to predict product TBP curves. Combined with volumetric and energy balances, their approach enables predictions with small discrepancies when compared to rigorous simulation. In their model, operating variables are used to compute the distances from the middle line of a product TBP curve, instead of relying on the internal reflux on selected trays, since tray temperatures are not required for monitoring or optimization. This hybrid model is of small size and good convergence, being suitable for planning, scheduling, coordinating and RTO environments.

For better predictions of process-shop yields and properties, Franzoi et al. (2018) use predictive analytics techniques by doing constrained and weighted least squares to better fit base plus delta or shift-vector sub-models using data reconciliation and regression techniques. The reconciliation strengthens the consistency of yields and the regression fits base and delta coefficients simultaneously across all yields. The proposed hybrid cutpoint optimization approach can be applied to the online optimization of crude oil blend scheduling operations in complex industrial-sized refineries to determine the composition-quality feed demands for the volumes and properties of distillates in towers in cascade (as a real process equipment design). According to Kelly and Zyngier (2008), a continuous cycle of improvements can use process measured feedback, which leads to higher accuracy and reliability, and aims to reduce the gap between the model predictions and the actual plant values.

Cutpoint temperature modeling has also been used for energy efficient operations of crude oil distillation units. Durrani et al. (2018) proposed a hybrid artificial neural network model based on the Taguchi method and genetic algorithm to handle uncertainties in the crude oil feed compositions to reduce energy costs. Atmospheric distillation units may represent more than 25% of the potential of energy savings in crude oil refineries. Therefore, an efficient cutpoint temperature modeling saves a substantial amount of energy typically lost during the process.

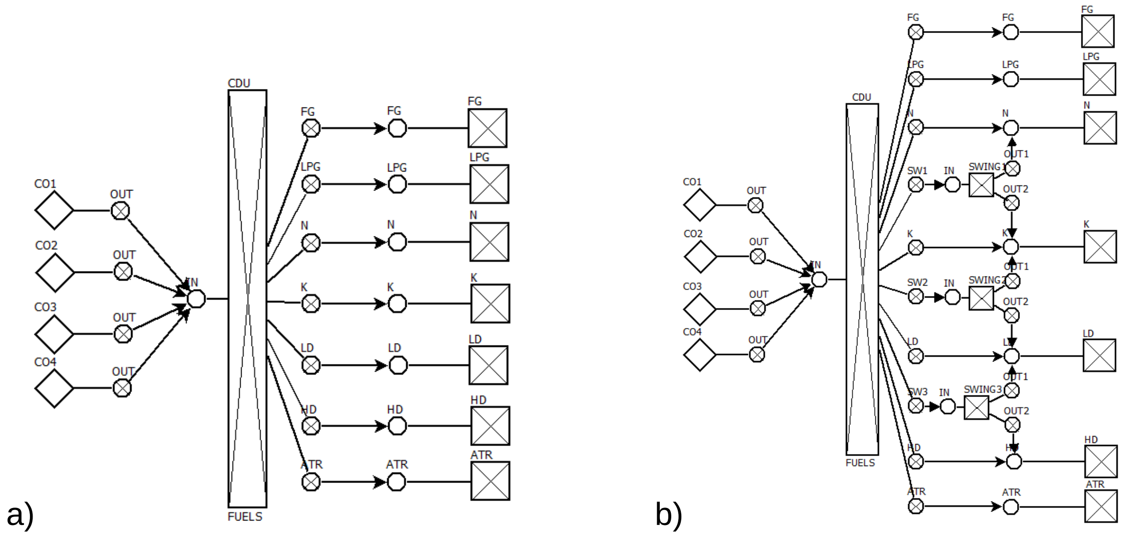
Most cutpoint temperature methods presented in the literature do not take into account the highly dynamic and uncertain real process environment or the typical plant versus model mismatches. In this context, the methodology proposed herein focuses on building an accurate cutpoint temperature model, suitable for planning, scheduling, coordinating and real time applications, and that uses continuous and real data from any reliable sources, such as the production plant or rigorous simulation to improve the predictions of crude oil distillation units.

6.3 Problem Statement

For the sake of simplicity in the presentation, and to motivate the ideas behind the proposed method, we consider a specific case in which there are four crude oils (CO1 to CO4) feeding a crude distillation unit, which produces seven final cuts: fuel gas (FG), liquefied petroleum gas (LPG), naphtha (N), kerosene (K), light diesel (LD), heavy diesel (HD), and atmospheric residue (ATR). The yields and properties of the distillates are calculated using different methods: a) fixed yield (FY); b) conventional swing-cut (CSW); and c) improved swing-cut (ISW). Figure 6.1a shows the process flowsheet within a UOPSS (unit-operation-port-state superstructure) representation (KELLY,

2005) for the distillation example in which swing-cuts are not considered, while Figure 6.1b represents a scenario with three swing-cuts between naphtha and kerosene, kerosene and light diesel, light diesel and heavy diesel for CSW and ISW. The capacities for the crude oil pools and final product pools are 100 Mbbl. The maximum flowrate for the distillation unit is 100 Mbbl/day. The crude oil assay data is embedded in the optimization problem as well.

Figure 6.1: Crude oil distillation unit flowsheet a) without swing-cuts and b) with swing-cuts.



Source: Author (2021).

The swing-cuts shown in Figure 6.1b represent hypothetical flows used for modeling and optimization purposes only. In real distillation unit operations, variables such as flow, temperature and pressure are adjusted in the plant to control the production of fuels. For example, decreasing or increasing the trays temperature profile changes the production of each distillate which has an important role in the refinery economics. In this work, we do not directly model these process variables, but the swing-cuts represent the same purpose or degree-of-freedom as an outcome of the operational variations within the distillation column.

For instance, let us consider the hypothetical swing-cut between naphtha and kerosene. If this swing-cut splits equally between naphtha and kerosene in the optimization problem, the operational conditions in a real process should be chosen to meet this condition. In that case, the naphtha/kerosene swing-cut split fraction sw_1 is a proxy for the naphtha endpoint cutpoint temperature (final boiling point) and relates the swing-cut naphtha (light-key) flow Q_N^{sw} to the swing-cut kerosene (heavy-key) flow Q_K^{sw}. Depending on the value of sw_1 chosen by the optimization, the swing naphtha/kerosene cutpoint temperature $T_{N/K}$ (to be adjusted in the actual unit in the tower) can be calculated as shown in Equations (6.1) and (6.2), in which T_N and T_K are the final cutpoint temperature of naphtha and the initial cutpoint temperature of kerosene, respectively. Although the swing-cut splits (light and heavy streams) go from internal swing-cut fractions to their respective neighbor final cut or distillate (lighter swing-cut goes to the upper distillate, and the heavier to the lower), due to the assumption of perfect- or sharp-separation between internal (cuts) and external (final-cuts) fractions of the towers in the ISW model, the final naphtha cutpoint is equivalent to the initial kerosene cutpoint.

$$sw_1 = \frac{Q_N^{sw}}{(Q_N^{sw} + Q_K^{sw})} = \frac{(T_{N/K} - T_N)}{(T_K - T_N)} \tag{6.1}$$

$$(1 - sw_1) = \frac{Q_K^{sw}}{(Q_N^{sw} + Q_K^{sw})} = \frac{(T_K - T_{N/K})}{(T_K - T_N)} \tag{6.2}$$

The final flows for naphtha and kerosene are their outlet CDU flows summed to their respective swing-cut parts, as shown in Equations (6.3) and (6.4).

$$Q_N^{final} = Q_N^{cdu} + Q_N^{sw} \tag{6.3}$$

$$Q_K^{final} = Q_K^{cdu} + Q_K^{sw} \tag{6.4}$$

Substituting Equations (6.1) and (6.2) into Equations (6.3) and (6.4), the naphtha and kerosene final flows can be rewritten as a function of the naphtha/kerosene cutpoint temperature $T_{N/K}$:

$$Q_N^{final} = Q_N^{cdu} + \frac{(T_{N/K} - T_N)}{(T_K - T_N)}(Q_N^{sw} + Q_K^{sw}) \tag{6.5}$$

$$Q_K^{final} = Q_K^{cdu} + \frac{(T_K - T_{N/K})}{(T_K - T_N)}(Q_N^{sw} + Q_K^{sw}) \tag{6.6}$$

6.4 Proposed distillation cutpoint modeling

Three crude oil distillation unit models from the literature are reproduced in this work, and described in the Supplementary Material (Appendix A) as: a) fixed yield (FY); b) conventional swing-cut (CSW); and c) improved swing-cut (ISW). Moreover, a novel distillation model, based on data analytics identification and estimation using MIQP coefficient setup techniques to determine optimizable surrogate models, is proposed in this section. For the proposed distillation cutpoint model, the outputs (yields and properties) of the Y dataset are determined using the CSW and ISW models.

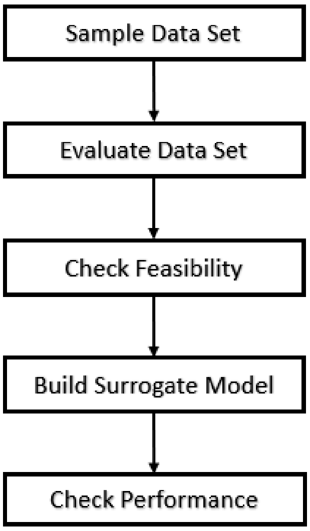
We propose a cutpoint temperature modeling framework using a data-driven coefficient setup MIQP technique to determine optimizable surrogate models to correlate variations in independent or X variables to dependent or Y variables inferred from X-Y datasets. This machine learning linear regression methodology focuses on establishing simple yet reliable correlations to estimate the yields and properties outputs from actual distillation units. These surrogate models are built from experiments, process simulations or any other reliable source of data, and can accurately predict the outputs of processes in which there is missing/uncertain data, and that are typically very complex and require high effort to be simulated or obtained. They can reduce the impact of variations in the crude oil assay and uncertainties in the process since they are predicted from a data-driven methodology (that implicitly accounts for crude oil assay information), rather than directly based on crude oil assay from a very expensive TPB experiment (which can be out-of-date). If experimental or actual data from the field is used in real processes, the proposed data-driven machine learning approach eliminates the need of distillation curves or distribution of yields and properties of the crude oil assays. Then, once the X-Y or input-output correlations are identified, the estimation of the parameters or coefficients may be updated using active or passive historical data. Moreover, when building the surrogates, all the crudes available are considered so that changing the feed of the distillation unit does not impact the performance of the method. Besides, the distillation unit feed is assumed to have constant properties (i.e., typically a tank feeding the CDU cannot be simultaneously filled; therefore, the crude blend composition within this tank can be reasonably assumed to be constant).

The surrogates take the functional form of low-degree polynomials to keep the model simple. Extensive testing showed that high accuracy could be achieved using this type of model. Several polynomial-based models are built to test which terms are necessary to provide good accuracy while keeping the model as small and simple as possible. These terms include intercept, linear terms for the crude oils, linear terms for swing-cuts, and bilinear terms for the product of each crude oil for each swing-cut. Some of the proposed models are linear, which may be more suitable for problems such as crude oil planning, that are typically large and may not afford nonlinearities (although that has changed recently given the technological improvements in computational power, solution algorithms, linearization strategies, etc.). Yet some models are bilinear and can provide better accuracy to nonlinear problems such as the crude oil scheduling. In order to define the structure of the surrogates, the following assumptions are adopted: a) linear terms for crude oils must be used due to the direct correlation between feed and distillates; b) bilinear terms between two crudes or two swing-cuts are not included to limit the total number of terms, and to avoid terms that are not statistically significant; c) the intercept term, the linear term for the crude oils and all the bilinear terms may or may not improve accuracy; d) terms that do not provide additional accuracy to the model should be avoided (i.e., over two similarly accurate models, the simplest or smaller model is recommended); e) terms that do not have a physical relation are not used in the surrogates. For instance, the liquefied petroleum gas and the atmospheric residue are not related to any swing-cut. Therefore, to estimate these variables, we do not use any term related to swing-cuts. Naphtha is solely related to the first swing-cut; the variables related to swing-cuts 2 and 3 are therefore not used to estimate the naphtha variables. A similar analysis was applied to all the variables in the surrogate model estimation.

The proposed methodology is implemented and tested using the crude oil distillation example presented in Figure 6.1b, in which four crude oils feed a distillation unit to produce seven final products. The framework for the proposed model, shown in Figure 6.2, is implemented in Python 3 using the Microsoft Visual Studio 2015 environment, and is integrated to Microsoft Excel to provide a more user-friendly approach for data manipulation and for better visualization of results. The modeling platform used is IMPL (Industrial Modeling & Programming Language), and the MIQP optimizations are carried out through the commercial solvers GUROBI 8.1.0 and CPLEX 12.8.0

connected to IMPL. The machine used was an Intel Core i7 with 2.90 GHz and 16 GB RAM.

Figure 6.2: Framework for the proposed strategy



Source: Author (2021).

In summary, this methodology: a) builds a data set using randomly generated data; b) calculates or simulates the process variables of interest; in this case, they are the outputs of the distillation unit, using this data set; c) checks the feasibility status for the optimal solutions previously found; d) builds or identifies a surrogate model that properly fits the data; and e) checks the performance of the surrogate model found by calculating the average mean square error between the surrogate model and the data set. Each step of the framework shown in Figure 6.2 is explained in detail as follows.

Sample Data Set: The proposed framework uses the well-established Latin Hypercube Sampling (LHS) technique to randomly sample 200 points for the independent variables for building a data set to be used to train the surrogate, which is divided into a training set and a testing set, each containing 100 points. For providing a better insight into the reliability and robustness of our methodology, the surrogates are tested over both the training (Data Set 1) and the testing data set (Data Set 2). As an example, Data Set 1 generated for the independent variables is presented in the Supplementary Material (Table 6.4). Each data set is used in both the CSW and the ISW models:

- Case 1a: Training data set using the CSW method;
- Case 1b: Testing data set using the CSW method;

- Case 2a: Training data set using the ISW method;
- Case 2b: Testing data set using the ISW method.

There are seven independent variables, related to the compositions of the four crude oils (x_1 to x_4) and the three swing-cut splits (x_5 to x_7). Variables x_5, x_6 and x_7 represent the light fractions for each swing cut, so that information on the heavy fractions is not accounted as the light and heavy fractions of each swing-cut, which are complementary to the unity. These seven independent variables are randomly generated for each sample point, respecting the composition consistency represented by Equation (6.7), in which the sum of compositions of all crude oils found in set CR must be equal to the unity.

$$\sum_{j \in CR} x_j = 1 \qquad (6.7)$$

Evaluate Data Set: The initial data sets are evaluated to calculate the yields and properties of the final distillation cuts. For that, Equations (6.18) to (6.32), see Appendix A in Supplementary Material, are employed in an optimization problem respecting the quality constraints (product specifications). The randomly generated independent variables, and the crude oil assay data are known. Although we do not need to optimize the problem (instead, we could calculate the final variables directly using Equations (6.18) to (6.32)), the optimization is useful to detect and to avoid poor selection of feedstocks that would eventually result in infeasible solutions, potentially with unspecified final products. As an example, the values of the dependent variables calculated using the ISW method based on Data Set 1 are shown in the Supplementary Material (Table 6.5 for yields, Table 6.6 for specific gravity, and Table 6.7 for sulfur content).

Check Feasibility: A feasibility check is performed over all the optimal solutions from the previous step. Infeasible or inconsistent solutions and low-quality sub-optimal solutions with objective functions 30% worse than the best solution found, are removed.

Build Surrogate Model: The final pool of solutions is used to train or to build a surrogate model for the yields and properties of each distillate stream. Each model is a function of the independent variables of crude oils and of its respective swing-cuts. For example, the yields and properties of naphtha stream vary with the crude oil composition and with the naphtha-kerosene swing-cut (i.e., swing-cut 1 in Figure 6.1b). These surrogate models are intended to be a simple yet accurate correlation to replace the swing-cut models (or any other distillation unit model). For that, X basis are introduced in the problem to account for the independent variables. For each distillation component $i \in DC$ (i.e., any yield or property of a distillation cut), there are four linear basis l_{ji} to account for each crude oil $j \in CR$ (x_1, x_2, x_3, x_4), and three linear basis l_{ki} to account for each swing cut $k \in SW$ (x_5, x_6, x_7). There are twelve bilinear basis b_{jki} for the relations between them, i.e., for the products of coefficients of each crude oil by each swing-cut: x_1x_5, x_1x_6, x_1x_7, x_2x_5, x_2x_6, x_2x_7, x_3x_5, x_3x_6, x_3x_7, x_4x_5, x_4x_6, x_4x_7, also known as interaction second-order effect terms. Quadratic terms are not considered in the model to limit the total number of terms. Other bilinear basis in which both coefficients represent crude oils or in which both represent swing-cuts (e.g., x_1x_2, x_5x_6) are not considered since these have no physical or relatable meaning, although they may prove beneficial when regressing with other data sets. An intercept coefficient (I_i) has also been used to account for any possible behavior not accounted for the other terms.

For each case, ten distinct models are proposed, in which three types of coefficients are employed. There are intercept coefficients, which are not associated to any basis; linear coefficients, which are multiplied for a linear basis (related to either a crude oil or a swing-cut); the bilinear or interaction coefficients, which are multiplied by two basis, becoming a second-order term in the equation. We believe these three types of coefficient-basis (intercept, linear and bilinear) are enough to accurately represent the interactions for the crude oil distillation process addressed herein.

An intelligent pre-choice or pre-elimination of coefficients is performed so as to be representative of real operations. For example, for naphtha components, the linear/bilinear coefficients used are related to all the four crude oils, but only to the first swing-cut, as there is no relation between naphtha components and swing-cuts 2 and 3 (as shown in Figure 6.1b). Similarly, kerosene relates to swing-cuts 1 and 2, light diesel relates to swing-cuts 2 and 3, and heavy diesel relates to swing-cut 3. Fuel gas, liquefied petroleum gas and atmospheric residue are not related to any swing-cut. Moreover, one of the independent variables related to the crude oils (x_4) is not included in some models that use the intercept term. That helps to avoid large variances and unobservable terms due to multicollinearity issues. We propose four linear models and six nonlinear models as follows:

- Model 1: linear coefficients for the crude oils (including x_4);
- Model 2: linear coefficients for the crude oils + linear coefficients for the swing-cuts (including x_4);
- Model 3: linear coefficients for the crude oils + bilinear coefficients for the products between one crude oil and one swing-cut (including x_4);
- Model 4: linear coefficients for the crude oils + linear coefficients for the swing-cuts + bilinear coefficients for the products between one crude oil and one swing-cut (including x_4);
- Model 5: intercept coefficient + linear coefficients for the crude oils (not including x_4);
- Model 6: intercept coefficient + linear coefficients for the crude oils + linear coefficients for the swing-cuts (not including x_4);
- Model 7: intercept coefficient + linear coefficients for the crude oils + bilinear coefficients for the products between one crude oil and one swing-cut (not including x_4);
- Model 8: intercept coefficient + linear coefficients for the crude oils + linear coefficients for the swing-cuts + bilinear coefficients for the products between one crude oil and one swing-cut (not including x_4);
- Model 9: intercept coefficient + linear coefficients for the crude oils + bilinear coefficients for the products between one crude oil and one swing-cut (including x_4);
- Model 10: intercept coefficient + linear coefficients for the crude oils + linear coefficients for the swing-cuts + bilinear coefficients for the products between one crude oil and one swing-cut (including x_4).

As an example, Surrogate Model 10 for a dependent variable (Y_{ip}) can be mathematically written as shown in Equation (6.8), in which CR and SW are the sets for crude oils and swing-cuts, respectively.

$$Y_{ip} = I_i + \sum_{j \in CR} l_{ji}\ X_{jp} + \sum_{j \in SW} l_{ki}\ X_{kp} + \sum_{j \in CR} \sum_{k \in SW} b_{jki}\, X_{jp} X_{kp} \qquad \forall\ i\ \in DC,\ \forall\ p \in P \qquad (6.8)$$

For each point p in data set P of independent variables (X), and for each dependent variable i (yields and properties of each output distillate from the distillation unit), Y_{ip} are the estimated outputs or dependent variable values, I_i are the intercept coefficients, l_{ji} and l_{ki} are the linear basis and b_{jki} are the bilinear basis with respect to crude oils j and swing-cuts k, respectively, X_{jp} are the crude oil compositions, and X_{kp} are the swing-cut yields or split fractions.

The surrogate models are built using MIQP optimizations for the stream yields of each final cut (except fuel gas), for the specific gravity of each final cut (except fuel gas), and for the sulfur content of each final cut (except fuel gas and LPG). Therefore, for each scenario, a surrogate is built for each one of the 17 variables and hence, a total of 17 optimization problems are formulated and optimized, each one with at most 20 coefficients (one intercept, seven linear and twelve bilinear basis). Each optimal solution contains the active basis (binaries equal to one) and its respective coefficients. Each optimization leads to a surrogate related to a specific process variable (i.e., yields or properties of distillation cuts), which are the dependent variables in our model. As an example of process variables of interest in our problem, there is the yield of naphtha (YLD_N), specific gravity of light diesel (SG_{LD}), sulfur content of atmospheric residue (S_{ATR}), etc. The MIQP optimization problems are formulated to minimize the least square error in Equation (6.9) subject to Equation (6.8) and to Equations (6.10) to (6.14) that limit or bound the values of the coefficients and impose a maximum specified number of basis:

$$\text{Minimize } E_i = \frac{1}{n} \sum_{p=1}^{n} (y_{ip} - Y_{ip})^2 \qquad (6.9)$$

$$-Mz_0 \le I_i \le Mz_0 \qquad \forall\ i\ \in DC \qquad (6.10)$$

$$-Mz_j \le l_{ji} \le Mz_j \qquad \forall\ j \in CR, i\ \in DC \qquad (6.11)$$

$$-Mz_k \le l_{ki} \le Mz_k \qquad \forall\ k \in SW, i\ \in DC \qquad (6.12)$$

In $$-Mz_{jk} \leq b_{jki} \leq Mz_{jk} \qquad \forall\ j\ \in CR, k\ \in SW, i\ \in DC \tag{6.13}$$

$$\sum_{j\,\in\,CR} z_j + \sum_{k\,\in\,SW} z_k + \sum_{j\,\in\,CR,k\,\in\,SW} z_{jk} + z_0 \leq B \qquad z_j, z_{jk}, z_{jk}, z_0 \in \{0,1\} \tag{6.14}$$

Equations (6.9) to (6.14), the number of points in the data set is $n = 100$, M is a sufficiently large number ($M = 1000$), z_j and z_k are binary variables that correspond to the linear basis, z_{jk} are binary variables for the bilinear basis, z_0 is the binary variable for the intercept, and B is the maximum number of basis. The real, physical or actual values for the dependent variables (y_{ip}) are calculated using either the conventional or the improved swing-cut method to provide acurate approximations. For building a surrogate model, we need to identify which basis l_{ji}, l_{ji}, l_{ki}, and b_{jki} should be used and which their respective coefficients are. For that, parameter B must be given in the optimization problem. In this work, the value of B is not limited as the problem to be optimized is small in size.

Check Performance: The surrogate models built for each dependent variable are compared to the original data set to calculate the error in the predictions (squared difference between the real values and the estimated values, i.e., $(y_{ip} - Y_{ip})^2$). To allow an easier comparison between distinct models, Equation (6.9) takes the average least square error among all 100 points from the data set, and Equations (6.15) to (6.17) take the final average error, in which DV_YLD, DV_SG, and DV_S are the sets for the yield, specific gravity, and sulfur content dependent variables, and $dv_{yld} = 6$, $dv_{sg} = 6$, and $dv_s = 5$ are the number of dependent variables within each category.

$$Yield\ Error = \frac{1}{dv\_yld} \sum_{i\,\in\,DV\_YLD} E_i \tag{6.15}$$

$$Specific\ Gravity\ Error = \frac{1}{dv\_sg} \sum_{i\,\in\,DV\_SG} E_i \tag{6.16}$$

$$Sulfur\ Content\ Error = \frac{1}{dv\_s} \sum_{i\,\in\,DV\_S} E_i \tag{6.17}$$

6.5 Results and Discussion

In this section, the results for each case are presented by using both the CSW and the ISW methods. Table 6.1 presents the least square errors for Yields, Specific Gravity and Sulfur Content from each surrogate model for each proposed case. Each of them is the average least square error among all data points for their respective dependent variables, obtained from Equations (6.15) to (6.17). Detailed results for each distillate can be found in the Supplementary Material (Tables 6.8 to 6.11). Also note that the solution of the MIQPs requires small computational times (less than 3 seconds per MIQP).

Table 6.1: Yield, Specific Gravity and Sulfur Content least square errors for each model and case.

		Model 1	Model 2	**Model 3**	Model 4	Model 5	Model 6	Model 7	Model 8	Model 9	Model 10
Data Set 1 CSW	Yield Error	1.57E+00	2.22E-03	**5.65E-16**	5.65E-16	2.55E-02	1.59E+00	2.30E-02	6.67E-02	5.65E-16	5.65E-16
	Specific Gravity Error	1.22E-05	2.48E-07	**1.90E-07**	1.90E-07	2.57E-07	1.22E-05	1.99E-07	4.98E-07	1.90E-07	1.90E-07
	Sulfur Content Error	1.59E-04	3.02E-06	**2.09E-06**	2.09E-06	9.36E-06	1.68E-04	8.36E-06	1.77E-05	2.09E-06	2.09E-06
Data Set 2 CSW	Yield Error	1.81E+00	1.77E-03	**8.33E-16**	8.33E-16	2.26E-02	1.83E+00	2.07E-02	6.75E-02	8.33E-16	8.33E-16
	Specific Gravity Error	1.44E-05	2.87E-07	**2.78E-07**	2.78E-07	2.94E-07	1.44E-05	2.85E-07	5.37E-07	2.78E-07	2.78E-07
	Sulfur Content Error	1.55E-04	3.30E-06	**3.05E-06**	3.05E-06	8.66E-06	1.59E-04	8.54E-06	1.74E-05	3.05E-06	3.05E-06
Data Set 1 ISW	Yield Error	1.57E+00	2.22E-03	**5.65E-16**	5.65E-16	2.55E-02	1.59E+00	2.30E-02	6.67E-02	5.65E-16	5.65E-16
	Specific Gravity Error	1.19E-05	7.38E-08	**3.09E-08**	3.09E-08	8.25E-08	1.19E-05	3.95E-08	3.30E-07	3.09E-08	3.09E-08
	Sulfur Content Error	1.54E-04	8.55E-07	**2.93E-07**	2.93E-07	7.15E-06	1.64E-04	6.56E-06	1.60E-05	2.93E-07	2.93E-07
Data Set 2 ISW	Yield Error	1.81E+00	1.77E-03	**8.33E-16**	8.33E-16	2.26E-02	1.83E+00	2.07E-02	6.75E-02	8.33E-16	8.33E-16
	Specific Gravity Error	1.43E-05	6.53E-08	**4.61E-08**	4.61E-08	7.25E-08	1.43E-05	5.33E-08	3.51E-07	4.61E-08	4.61E-08
	Sulfur Content Error	1.54E-04	1.09E-06	**2.75E-07**	2.75E-07	6.54E-06	1.59E-04	5.77E-06	1.54E-05	2.75E-07	2.75E-07

Source: Author (2021).

The lower the average error between the swing-cut approach and the surrogate model, the better the performance of the surrogate is. The main findings from the results in Table 6.1 are summarized as follows:

a) As expected, the performance of the linear models (Models 1, 2, 5 and 6) is not as good as their nonlinear counterparts. Among them, Model 2 performs better, which indicates that including variable x_4 is more important than the intercept term.

b) Including linear variables for swing-cuts improves the predictions in some models (e.g., Model 2 performs better than Model 1), but may not be necessary for some nonlinear models (e.g., Models 3 and 4 have the same performance).

c) Not including x_4 in the nonlinear models reduces the accuracy even when an intercept term is used (e.g., Models 3 and 4 performs better than Models 7 and 8).

d) Models 3, 4, 9 and 10 achieve the same accuracy in all the cases tested. Thus, neither the intercept term nor the linear swing-cuts terms were significantly useful to improve the performance. We chose the best surrogate as being the simplest surrogate with the highest accuracy. Therefore, Model 3 achieves the best performance.

e) Although Models 9 and 10 provide an excellent performance, they use both independent variable x_4 and the intercept term, which leads to a multicollinear model. That may not be advisable because even if the optimization solver can handle the multicollinearities and find an optimal solution, there might be unobservable coefficients in the model as well as large variances in the error.

Note that Model 2 performs fairly well considering it is linear and small in size, it has a good accuracy, and it can be easily integrated into large-scale planning environments. The trade-off between the accuracy and the increase in the computational effort must be evaluated upon the desired purpose or application. It should also be mentioned that all of the candidate models require the crude oil compositions to be tracked and traced prior to the overall crude oil mixture being charged to the crude distillation unit in order to achieve the predicted results presented here.

For the case studies considered, the yields and properties from our surrogates very accurately match the values from both the conventional and improved swing-cut model, so that our model is as representative as the swing-cut models. As there is only one equation needed to calculate each yield/property of each final cut, our data-driven machine learning coefficient-setup model requires fewer equations and fewer degrees of freedom than other methods. It can be easily integrated into any planning/scheduling environment without significantly increasing the simulation/optimization effort. As an example, we take the results from Model 3 using the ISW model to show the active basis and its respective coefficients being used to establish correlations to calculate the yields of the final cuts:

$$YLD_{LPG} = 0.9029x_{CO1} + 0.9762x_{CO2} + 1.2367x_{CO3} + 1.0680x_{CO4}$$

$$\begin{aligned} YLD_N = {} & 7.1934x_{CO1} + 6.4813x_{CO2} + 7.0187x_{CO3} + 4.7761x_{CO4} + 3.4999x_{CO1}x_{SW1} \\ & + 3.4137x_{CO2}x_{SW1} + 2.7201x_{CO3}x_{SW1} + 2.1239x_{CO4}x_{SW1} \end{aligned}$$

$$\begin{aligned} YLD_K = {} & 9.7963x_{CO1} + 9.8719x_{CO2} + 8.1131x_{CO3} + 6.7589x_{CO4} - 3.4999x_{CO1}x_{SW1} \\ & - 3.4137x_{CO2}x_{SW1} - 2.7201x_{CO3}x_{SW1} - 2.1239x_{CO4}x_{SW1} \\ & + 5.5787x_{CO1}x_{SW2} + 6.3726x_{CO2}x_{SW2} + 5.1732x_{CO3}x_{SW2} \\ & + 4.7251x_{CO4}x_{SW2} \end{aligned}$$

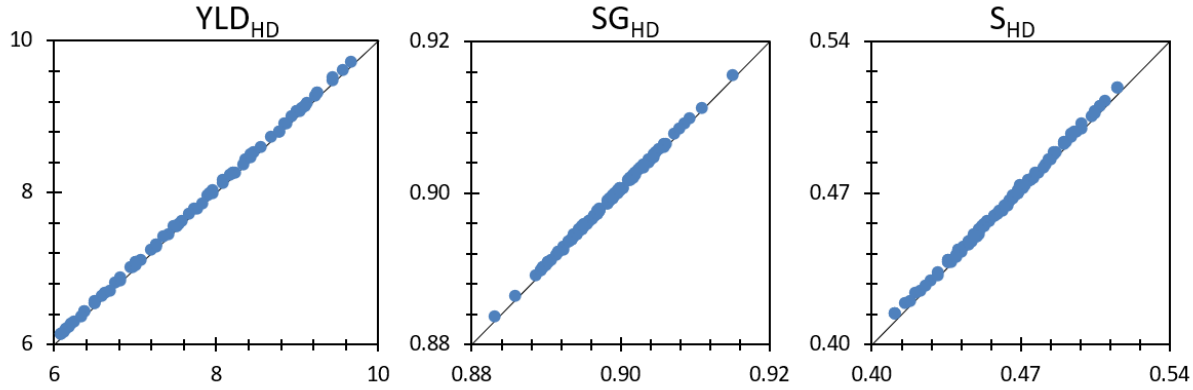
$$\begin{aligned} YLD_{LD} = {} & 12.9190x_{CO1} + 14.5937x_{CO2} + 12.4719x_{CO3} + 11.9013x_{CO4} - 5.5787x_{CO1}x_{SW2} \\ & - 6.3726x_{CO2}x_{SW2} - 5.1732x_{CO3}x_{SW2} - 4.7251x_{CO4}x_{SW2} \\ & + 4.7096x_{CO1}x_{SW3} + 4.7427x_{CO2}x_{SW3} + 4.5683x_{CO3}x_{SW3} \\ & + 4.8972x_{CO4}x_{SW3} \end{aligned}$$

$$\begin{aligned} YLD_{HD} = {} & 9.4868x_{CO1} + 9.4351x_{CO2} + 9.3082x_{CO3} + 10.0838x_{CO4} - 4.7096x_{CO1}x_{SW3} \\ & - 4.7427x_{CO2}x_{SW3} - 4.5683x_{CO3}x_{SW3} - 4.8972x_{CO4}x_{SW3} \end{aligned}$$

$$YLD_{ATR} = 59.5791x_{CO1} + 58.5091x_{CO2} + 61.7528x_{CO3} + 65.3256x_{CO4}$$

Figure 6.3 displays the cross plots (parity charts) for the $Y_{estimated}$ (y axis) versus the Y_{real} (x axis) for the yield, specific gravity, and sulfur content of heavy diesel in Surrogate Model 3 using the ISW method. The black line represents the $y = x$ function, so that the closer to the line, the better the model adjusts to the real function. The other cross plots for the Surrogate Model 3 are presented in the Supplementary Material (Figure 6.6).

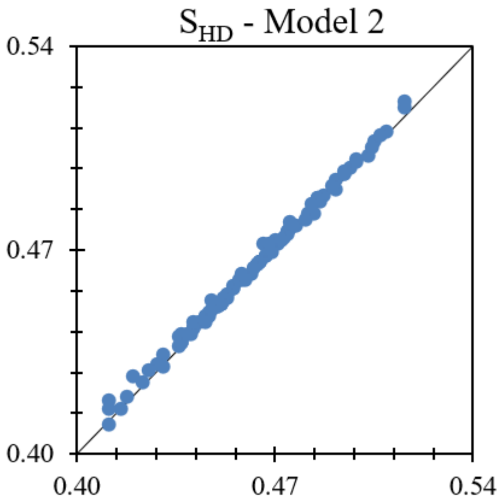
Figure 6.3: Cross plots for the heavy diesel dependent variables in Surrogate Model 3



Source: Author (2021).

Surrogate Model 3 provides an excellent fit to the data, as shown in the plots from Figure 6.3. Furthermore, even if the bilinear coefficients are not used, the method is able to find a fairly good fit. As an example, Figure 6.4 shows the cross plots for the sulfur content dependent variable to illustrate the difference $Y_{estimated}$ versus the Y_{real} for Surrogate Model 2 (which is linear) using the ISW method.

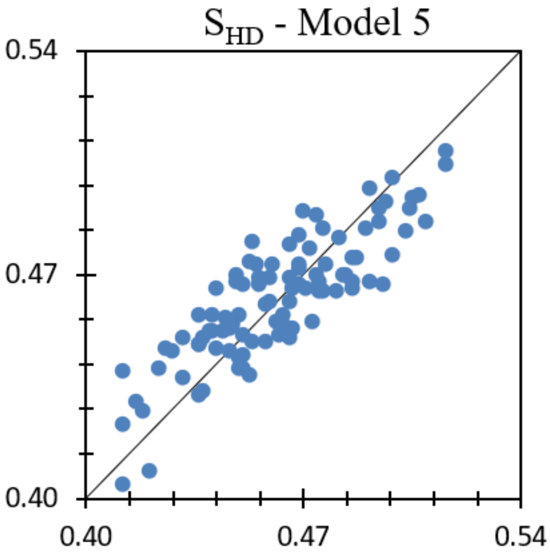
Figure 6.4: Cross plot for the heavy diesel dependent variables in Surrogate Model 2.



Source: Author (2021).

If the fit to the data is not good, there may be large errors in the predictions. For example, Figure 6.5 presents a similar cross plot for Surrogate Model 5, which shows a more inaccurate fit.

Figure 6.5: Cross plot for the heavy diesel dependent variables in Surrogate Model 5.



Source: Author (2021).

Discrepancies can be observed between the ISW method and the correlation using Surrogate Model 5 (Figure 6.5), which might have a significant impact depending on the application of the method. Poor predictions in the sulfur content of distillates may underspecify the product, which can potentially imply costs for the refinery, to reprocess or blend the product with overspecified streams. As an alternative to mitigate off-spec products, the final fuels can be sold at reduced prices and waivers can be requested from regulatory agencies, which is not a good practice or sustainable procedure. Conversely, overspecifications by poor predictions can produce distillates with product giveaways, which typically imply losses at selling higher quality products at regular prices. To reduce both under and overspecification in production, a better prediction of the inputs and outputs of the distillation process is required, whereby the surrogate model methodology proposed herein is an alternative to improve the accuracy of predictions under simulation/optimization environments. For the predictions of yields and properties of distillation processes using the addressed coefficient setup MIQP technique, the use of bilinear terms, in addition to the intercept and linear terms (without the need of the linear term for the swing-cuts), is an efficient fashion to mathematically represent this type of process.

The method proposed could potentially be applied to distillation units in problems such as crude oil refinery planning and scheduling. Regardless of the application, the CDU outputs must be somehow estimated in any planning or scheduling model. This research provides an alternative approach to methods presented in the literature and currently used by the industry. The data required to train and to build the surrogates can be provided by rigorous simulators to achieve high accuracy and the surrogates could be further embedded into planning and scheduling environments. We believe these surrogates can be easily integrated into planning, scheduling, or coordinating environments for the following reasons:

i. The surrogates are shown to be very accurate, which means they can simulate the behavior of the distillation process fairly well.
ii. The surrogates are built considering crude oil composition, which indirectly accounts for crude assay information. Therefore, embedding the surrogates into refinery planning and scheduling environments does not require the complex crude assay data.
iii. The surrogates should be estimated prior to their integration into other environments. They are built through MIQP optimizations that require small computational times (around 3 seconds). Therefore, they can be re-estimated whenever needed (e.g., if new data points are available, if new crudes arrive at the refinery, etc.).
iv. They are simple and small as they use a small number of terms (at most bilinear ones). In any case, linear models, such as Model 2, can be used if nonlinearities are not afforded in the main optimization environment.

6.6 Conclusions

We are moving towards a more complex and detailed process optimization age, mainly due to the advancements in decision-making modeling, computer-aided resources, and solution algorithms. Big data is becoming a reality and leads to opportunities of cost reduction in industrial processes. The surrogate modeling proposed herein to estimate compositions and properties of distillates may be considered machine-learning and predictive analytics techniques, and can use either real (and uncertain) data from the plant or rigorous simulated data to improve the predictions of distillation units by using measurement feedback. In other words, the proposed surrogate model can be self-adjustable and self-improvable, despite requiring engineering supervision.

The results show that our model provides accurate predictions when compared to both the conventional and the improved swing-cut methods. Due to the small number of equations required, our shortcut sub-models can be easily integrated into any planning, scheduling, and coordinating environment with minimal increase in the simulation and optimization effort and data requirements.

Future work will focus on gathering real data from crude oil refineries with various crude oil compositions, as well as varying distillation tower operating variables considering input and output datasets from both physical experiments (taken and estimated using field measurements) and synthetical trials (using rigorous process simulation).

6.7 Supplementary Material

The supplementary material for the cutpoint temperature surrogate modeling research includes a review on distillation unit modeling (fixed yield, swing-cut, and improved swing-cut models), the datasets used to build the surrogates, the least squares errors calculated by using the surrogates, and the cross plots for the best surrogate model.

6.7.1 Appendix A: Review on Distillation Unit Modeling

In this section we present a review on distillation unit modeling, including the fixed yield, swing-cut, and improved swing-cut models.

6.7.1.1 Fixed yield modeling (FY)

A straightforward method to model a crude distillation unit is by using single or multiple fixed yield values for the CDU product stream outputs. This approach is largely used for strategic, tactical, and operational planning in industry, and consists in specifying the final-cuts or product distillates that divide the CDU fractions considering crude oil assay information to calculate these fractions. As this method uses perfect (sharp) fractionation to calculate the CDU yields, it does not rigorously compute the distillation column fractionation. When the fixed-yield formulation is addressed, the crude oil assay data and fixed boiling point temperature ranges without overlapping (sharp fractionation) are used to calculate the yields and properties for the final cuts. Hence, the distillates in the proposed problem were assigned to a certain range of micro-cuts or pseudo-components, as shown in Table 6.2.

Table 6.2: Micro-cut range for the fixed yield model.

Distillate	Initial micro-cut	Final micro-cut	Number of micro-cuts
FG – Fuel gas	CH4	C2H5	02
LPG – Liquefied petroleum gas	C3H8	NC4H10	03
N – Naphtha	IC5H12	CUT150	14
K – Kerosene	CUT160	CUT240	09
LD – Light diesel	CUT250	CUT320	08
HD – Heavy diesel	CUT330	CUT350	03
ATR – Atmospheric residue	CUT360	CUT850	50

Source: Author (2021).

In Equation (6.18), the inlet volumetric flows $Q_{c,cdu}$ for each crude oil c incoming to the CDU are summed to calculate the overall volumetric feed flow Q_{cdu} for the CDU.

$$Q_{cdu} = \sum_{c} Q_{c,cdu} \tag{6.18}$$

The volumetric flow rate Q_{cut}^{fc} for each distillate or final cut fc can be calculated as the summation of yields of micro-cuts YLD_{mc} that flow to the pre-defined boiling point temperature ranges, resulting in the product of the overall volumetric flow in the CDU and the summation of pre-defined micro-cuts yields to each cut fc in Equation (6.19). The set MC_{fc} can be determined from Table A1 and represents the micro cuts related to final cuts of distillates.

$$Q_{cut}^{fc} = Q_{cdu} \sum_{mc \in MC_{fc}} YLD_{mc} \qquad \forall \; cut \in FC \tag{6.19}$$

Volume- and mass-based mixing rules are used to calculate the properties of each final cut fc. Equation (6.20) represents the volumetric rules for specific gravity, and Equation (6.21) represents the mass rules for sulfur content. Thus, yields and properties (volume- and mass-based) for the final cuts can be calculated from the crude oil assay data using Equations (6.20) and (6.21).

$$VP_{cut}^{fc} = \frac{\sum_{mc \in MC_{fc}} VP_{mc}^{fc} Q_{mc}}{\sum_{mc \in MC_{fc}} Q_{mc}} \qquad \forall\, fc \tag{6.20}$$

$$MP_{cut}^{fc} = \frac{\sum_{mc \in MC_{fc}} MP_{mc}^{fc} VP_{mc}^{fc} Q_{mc}}{\sum_{mc \in MC_{fc}} VP_{mc}^{fc} Q_{mc}} \qquad \forall\, fc \tag{6.21}$$

In order to improve the prediction of the crude oil distillation unit outputs, there are models that consider non-sharp fractionation by including volume and/or mass variations for the CDU products of external distillates by creating hypothetical internal streams that can flow to the respective upper and lower final distillates, the so-called light and heavy swing-cuts, as shown in the swing-cut modeling as follows.

6.7.1.2 Conventional swing-cut modeling (CSW)

The swing-cut method creates additional degrees of freedom by allowing the optimization of each swing-cut amount flowing to the lighter and heavier final distillates. The swing-cuts are hypothetical internal modeling constructs, and do not physically exist in the tower. In this work, three swing-cuts sc are considered. Table 6.3 shows the initial and final micro cuts, and the respective number of micro-cuts related to each distillate.

Table 6.3: Micro-cuts for the swing-cut model.

Distillate	Initial Micro-cut	Final Micro-cut	Number of Micro-cuts
FG – Fuel gas	CH4	C2H5	02
LPG – Liquefied petroleum gas	C3H8	NC4H10	03
N – Naphtha	IC5H12	CUT120	11
SW1 – Swing-cut 1	CUT130	CUT150	03
K – Kerosene	CUT160	CUT200	05
SW2 – Swing-cut 2	CUT210	CUT240	04
LD – Light diesel	CUT250	CUT290	05
SW3 – Swing-cut 3	CUT300	CUT320	03
HD – Heavy diesel	CUT330	CUT360	04
ATR – Atmospheric residue	CUT370	CUT850	49

Source: Author (2021).

The mathematical model for the conventional swing-cut method is given as follows. Equations (6.18) to (6.21) are the same as in the fixed yield model, which represent a mass balance for the CDU, the flow calculation for each distillate, and the volume- and mass-based balances, respectively. However, some of the final cuts in the fixed yield model are intermediate cuts in the swing-cut model, which will be further blended with the swing-cuts to create the final cuts. Thus, Equation (6.22) represents the mass balances for the intermediate cuts (naphtha, kerosene, light diesel, heavy diesel, and the three swing-cuts), with Q_{cut}^{ic} as their flows.

$$Q_{cut}^{ic} = Q_{cdu} \sum_{mc \in MC_{ic}} YLD_{mc} \qquad \forall\, cut \in IC \qquad (6.22)$$

By the definition of swing-cut methods, Equation (6.23) represents the mass balance when the swing-cut splits to its light and heavy fractions, which are the additional decision variables included in the optimization problem.

$$Q_{sw} = Q_{sw}^{light} + Q_{sw}^{heavy} \qquad \forall\ sw \in SW \qquad (6.23)$$

When intermediate cuts are mixed with their respective swing-cuts, we use material and property balance constraints to calculate the flows and properties for the final cuts. Equations (6.24), (6.25) and (6.26) calculate the flows, the volume-based properties, and the mass-based properties, respectively, for the final cuts of each distillate related to swing-cuts.

$$Q_N^{fc} = Q_N^{ic} + Q_{sw1}^{light} \qquad (6.24a)$$

$$Q_K^{fc} = Q_K^{ic} + Q_{sw1}^{heavy} + Q_{sw2}^{light} \qquad (6.24b)$$

$$Q_{LD}^{fc} = Q_{LD}^{ic} + Q_{sw2}^{heavy} + Q_{sw3}^{light} \qquad (6.24c)$$

$$Q_{HD}^{fc} = Q_{HD}^{ic} + Q_{sw3}^{heavy} \qquad (6.24d)$$

$$VP_N^{fc} = \frac{Q_N^{ic} VP_N^{ic} + Q_{sw1}^{light} VP_{sw1}}{Q_N^{ic} + Q_{sw1}^{light}} \qquad (6.25a)$$

$$VP_K^{fc} = \frac{Q_K^{ic} VP_K^{ic} + Q_{sw1}^{heavy} VP_{sw1} + Q_{sw2}^{light} VP_{sw2}}{Q_K^{ic} + Q_{sw1}^{heavy} + Q_{sw2}^{light}} \qquad (6.25b)$$

$$VP_{LD}^{fc} = \frac{Q_{LD}^{ic} VP_{LD}^{ic} + Q_{sw2}^{heavy} VP_{sw2} + Q_{sw3}^{light} VP_{sw3}}{Q_{LD}^{ic} + Q_{sw2}^{heavy} + Q_{sw3}^{light}} \tag{6.25c}$$

$$VP_{HD}^{fc} = \frac{Q_{HD}^{ic} VP_{HD}^{ic} + Q_{sw3}^{heavy} VP_{sw3}}{Q_{HD}^{ic} + Q_{sw3}^{heavy}} \tag{6.25d}$$

$$MP_{N}^{fc} = \frac{Q_{N}^{ic} VP_{N}^{ic} MP_{N}^{ic} + Q_{sw1}^{light} VP_{sw1} MP_{sw1}}{Q_{N}^{ic} VP_{N}^{ic} + Q_{sw1}^{light} VP_{sw1}} \tag{6.26a}$$

$$MP_{K}^{fc} = \frac{Q_{K}^{ic} VP_{K}^{ic} MP_{K}^{ic} + Q_{sw1}^{heavy} VP_{sw1} MP_{sw1} + Q_{sw2}^{light} VP_{sw2} MP_{sw2}}{Q_{K}^{ic} VP_{K}^{ic} + Q_{sw1}^{heavy} VP_{sw1} + Q_{sw2}^{light} VP_{sw2}} \tag{6.26b}$$

$$MP_{LD}^{fc} = \frac{Q_{N}^{ic} VP_{N}^{ic} MP_{N}^{ic} + Q_{sw2}^{heavy} VP_{sw2} MP_{sw2} + Q_{sw3}^{light} VP_{sw3} MP_{sw3}}{Q_{N}^{ic} VP_{N}^{ic} + Q_{sw2}^{heavy} VP_{sw2} + Q_{sw3}^{light} VP_{sw3}} \tag{6.26c}$$

$$MP_{HD}^{fc} = \frac{Q_{N}^{ic} VP_{N}^{ic} MP_{N}^{ic} + Q_{sw3}^{heavy} VP_{sw3} MP_{sw3}}{Q_{N}^{ic} VP_{N}^{ic} + Q_{sw3}^{heavy} VP_{sw3}} \tag{6.26d}$$

The conventional swing-cut method creates additional degrees of freedom by allowing the optimization of each swing-cut amount flowing to the lighter and heavier final distillates. Therefore, the search space for optimization becomes larger and an equivalent or better solution is expected when compared to the (multiple) fixed yield method.

6.7.1.3 Improved swing-cut modeling (ISW)

The conventional swing-cut modeling considers fixed properties for the hypothetical cuts that swing between adjacent light and heavy distillates, although that may lead to inaccuracies in the quantity and quality predictions of the final distillates (MENEZES, KELLY, and GROSSMANN, 2013). In the improved swing-cut modeling, each swing-cut is split into two internal light and heavy streams, similarly to the CSW modeling. However, the light and heavy streams are not assumed to have the bulk quality. The ISW method proposed by these authors adds a set of interpolations to improve the prediction of distillates by considering quality variations for the light and heavy fractions of each swing-cut. If the whole swing-cut flows to a specific fraction (either light or heavy), the properties of this fraction will be the bulk properties of the swing-cut.

However, whenever the swing-cut splits to both fractions, there is a difference among their properties, which are also different from the bulk. Any light swing-cut stream that flows to the upper cut will have a lighter property than the swing-cut bulk (lower value in the vast majority of the properties, such as density and sulfur concentration), and heavier than the interface between the bulk and the upper cut. Similarly, a heavy swing-cut stream that flows to the lower cut will have a heavier property than the bulk swing-cut (typically higher value when compared with the bulk) and lighter than the interface between the bulk and the lower cut.

The mathematical model for the improved swing-cut method and a brief explanation about how this method mathematically works is given as follows. A complete explanation and additional details can be found in Menezes, Kelly, and Grossmann (2013). The model for the improved swing-cut uses Equations (6.18) to (6.26) from the conventional swing-cut method. The main contribution to the improved swing-cut method regards the swing-cut properties calculation. Each swing-cut may be split into two internal streams with different qualities, whereas the conventional swing-cut uses the bulk quality for both streams. Interpolations are performed to better predict the qualities of final distillates, which vary linearly between the properties at their adjacent hypothetical interfaces. Besides, new variables are created to represent these interfaces between adjacent cuts, VPI for volume-based properties and MPI for mass-based properties, and their calculation is performed considering the blending of micro-cut streams, such as Equations (6.27) and (6.28), in which $mc1$ and $mc2$ are the lighter and heavier adjacent micro cuts to the respective hypothetical interface of the swing-cut.

$$VPI_{sw} = \frac{VP_{mc1}Q_{mc1} + VP_{mc2}Q_{mc2}}{Q_{mc1} + Q_{mc2}} \qquad \forall \ sw \in SW \qquad (6.27)$$

$$MPI_{sw} = \frac{VP_{mc1}MP_{mc1}Q_{mc1} + VP_{mc2}MP_{mc2}Q_{mc2}}{VP_{mc1}Q_{mc1} + VP_{mc2}Q_{mc2}} \qquad \forall \ sw \in SW \qquad (6.28)$$

Equations (6.29) and (6.30) use the interface variables and calculate the light and heavy swing-cut volume-based properties using linear interpolation around the bulk and hypothetical light and heavy interfaces of the swing-cuts.

$$VP_{sw}^{light} = VPI_{sw}^{light} + \frac{VP_{sw} - VPI_{sw}^{light}}{Q_{sw}} \, Q_{sw}^{light} \qquad \forall \; sw \in SW \qquad (6.29)$$

$$VP_{sw}^{heavy} = VPI_{sw}^{heavy} + \frac{VP_{sw} - VPI_{sw}^{heavy}}{Q_{sw}} \, Q_{sw}^{heavy} \qquad \forall \; sw \in SW \qquad (6.30)$$

Similarly, Equations (6.31) and (6.32) applies linear interpolation to calculate mass-based properties.

$$MP_{sw}^{light} = MPI_{sw}^{light} + \frac{MP_{sw} - MPI_{sw}^{light}}{VP_{sw}Q_{sw}} \, VP_{sw}^{light}Q_{sw}^{light} \qquad \forall \; sw \in SW \qquad (6.31)$$

$$MP_{sw}^{heavy} = MPI_{sw}^{heavy} + \frac{MP_{sw} - MPI_{sw}^{heavy}}{VP_{sw}Q_{sw}} \, VP_{sw}^{heavy}Q_{sw}^{heavy} \qquad \forall \; sw \in SW \qquad (6.32)$$

These quality interpolation constraints are used for improvements in the predictions of the final distillates properties and provide a more accurate modeling for the swing-cut method.

6.7.2 Appendix B: Data Set 1: Independent and Dependent Variables

Table 6.4 shows data set 1 used to test the proposed methodology, generated by the Latin Hypercube Sampling technique. There are seven independent variables: x1 to x4 account for the four crude oils whereas x5 to x7 account for the three swing-cuts.

Table 6.4: Independent data set 1 generated by Latin Hypercube Sampling.

x1	x2	x3	x4	x5	x6	x7
0.115977805	0.033017147	0.398073556	0.452931491	0.150655345	0.244379184	0.250940892
0.17063455	0.304731891	0.356932792	0.167700766	0.680388327	0.626619307	0.690691442
0.437511765	0.000262226	0.345845439	0.21638057	0.877668313	0.891336845	0.948349719
0.06653081	0.297344795	0.275739844	0.360384551	0.410538576	0.111839298	0.983078379
0.152055211	0.399795334	0.291271843	0.156877611	0.01524906	0.15094929	0.094721638
0.393300459	0.333894058	0.144234342	0.128571141	0.607306773	0.674940778	0.659651006
0.010596159	0.165004684	0.442102947	0.38229621	0.712632694	0.327694143	0.735454199
0.038434607	0.38827486	0.190178029	0.383112504	0.595498137	0.6371853	0.323330144
0.217176484	0.280328505	0.314511862	0.187983149	0.568093062	0.490090458	0.810325839
0.109132126	0.104175936	0.499419653	0.287272285	0.391814658	0.818955285	0.791474982
0.08831644	0.417232906	0.16525176	0.329198894	0.160594948	0.279288916	0.728819599
0.059161441	0.555996951	0.192046255	0.192795352	0.242507308	0.380317354	0.203769106
0.631079851	0.241088534	0.114511387	0.013320228	0.999989158	0.217771444	0.924663898
0.047267418	0.252543914	0.383151728	0.31703694	0.086919631	0.978241615	0.596830213
0.301732708	0.041790985	0.422001624	0.234474683	0.355744247	0.053566554	0.6430744
0.162560335	0.356109954	0.083705345	0.397624366	0.429969602	0.859107215	0.934965853
0.556992807	0.332220876	0.057359512	0.053426805	0.02319535	0.12425503	0.994294199
0.574136288	0.095049313	0.148851646	0.181962754	0.628285865	0.340537526	0.164196392
0.092649922	0.312104146	0.381446414	0.213799518	0.537739813	0.953041174	0.573006701
0.236260597	0.524800342	0.108995724	0.129943337	0.89295066	0.781743732	0.756911513
0.427723623	0.153049757	0.187589264	0.231637356	0.259666102	0.478175334	0.672836395
0.31169184	0.26910479	0.212191229	0.207012141	0.889221903	0.732520616	0.523995144
0.117632609	0.38795601	0.053727691	0.44068369	0.382432681	0.59606146	0.374956835
0.322007089	0.258974325	0.307061522	0.111957064	0.64803347	0.568529948	0.292840421
0.273043214	0.220208361	0.473145391	0.033603034	0.971843473	0.373243971	0.621421271
0.237883639	0.215175385	0.209803636	0.33713734	0.049106146	0.586990483	0.559046003
0.340726865	0.345626102	0.249014753	0.06463228	0.233591847	0.228178438	0.225002512
0.477200204	0.202741417	0.246580943	0.073477436	0.58635123	0.105116871	0.2392449
0.204828615	0.156521522	0.196053062	0.442596801	0.203779618	0.924044133	0.63974291
0.30480881	0.127817029	0.365191246	0.202182916	0.830288693	0.872068679	0.44828057
0.225846078	0.13774912	0.289218145	0.347186657	0.077385275	0.904312847	0.036014407
0.28250372	0.339768573	0.114112391	0.263615315	0.900499748	0.179266455	0.101303877
0.002539167	0.410328186	0.52547982	0.061652826	0.673991095	0.338332678	0.397317653
0.557116869	0.402118027	0.002191236	0.038573868	0.48978725	0.83431064	0.218920469
0.281609622	0.399572796	0.049989996	0.268827586	0.449350636	0.096132998	0.487303161
0.196046454	0.433687785	0.134227674	0.236038088	0.812327007	0.160011901	0.38001758
0.316425842	0.33662573	0.050049906	0.296898521	0.213649885	0.50361124	0.114105238
0.319104695	0.15994204	0.046221621	0.474731643	0.611634752	0.195280097	0.462591437
0.132824617	0.230353351	0.618761274	0.018060758	0.529043193	0.538950901	0.047797072
0.071724645	0.095630889	0.710389988	0.122254478	0.745677304	0.578449196	0.605643118
0.299835468	0.253401121	0.189506118	0.257257293	0.634881085	0.396394522	0.342729099
0.308821689	0.187756368	0.167864923	0.33555702	0.433154624	0.71975874	0.085143373
0.0362072	0.406903808	0.149448948	0.407440044	0.703920146	0.455895396	0.853924856
0.413609157	0.308696295	0.261500574	0.016193975	0.131964333	0.984540154	0.806375558
0.261570354	0.536580972	0.135339552	0.066509123	0.361078227	0.141759778	0.076593328
0.075749103	0.276913337	0.375837866	0.271499694	0.177204514	0.606265314	0.711136716

0.217669165	0.253620523	0.291287758	0.237422554	0.296214777	0.521762382	0.280361672
0.240834669	0.594453974	0.018403926	0.146307431	0.108937319	0.869517924	0.897756624
0.320045522	0.189543543	0.161998148	0.328412787	0.003435683	0.423524738	0.560278727
0.225603917	0.14154767	0.300618144	0.332230269	0.828803262	0.946431202	0.549208683
0.230464911	0.190838634	0.140540782	0.438155673	0.519730053	0.073007878	0.173401968
0.27597789	0.268306593	0.173932071	0.281783447	0.910299011	0.265484195	0.242859578
0.261702128	0.116741071	0.293340567	0.328216234	0.794934663	0.236879507	0.123880473
0.130062949	0.254380426	0.464180228	0.151376397	0.194138999	0.434156526	0.268732769
0.263131768	0.015260785	0.134008205	0.587599241	0.755590235	0.025077255	0.773423579
0.309993093	0.306674423	0.098405294	0.28492719	0.848066024	0.01565984	0.584933583
0.075578937	0.159223085	0.287535682	0.477662296	0.181336995	0.290880454	0.667794628
0.267825028	0.396272588	0.24328753	0.092614854	0.284186397	0.650773226	0.865605778
0.102212235	0.275506549	0.415182877	0.207098339	0.86018695	0.062075946	0.270419201
0.29143001	0.368151307	0.096860351	0.243558332	0.953205478	0.464475275	0.065940442
0.463527356	0.088446431	0.181834004	0.266192209	0.348929792	0.933006231	0.685718848
0.152557545	0.069329974	0.429506921	0.34860556	0.056623499	0.201149505	0.02973766
0.385629502	0.441622049	0.008044685	0.164703764	0.577551008	0.139793034	0.361523459
0.240648997	0.20826862	0.164422086	0.386660297	0.452450138	0.619467113	0.433765516
0.250463778	0.333763157	0.149350382	0.266422682	0.461057243	0.04908057	0.9071385
0.367264685	0.131106095	0.045361441	0.45626778	0.038486717	0.88626096	0.147958285
0.233267688	0.202880328	0.272083179	0.291768806	0.125272405	0.963310069	0.912331421
0.344916173	0.237902907	0.314987275	0.102193645	0.473519485	0.761987186	0.318638166
0.269318229	0.038744906	0.408123023	0.283813841	0.772472747	0.809466924	0.130592706
0.120934507	0.208724113	0.343478556	0.326862824	0.305003105	0.407193157	0.615382281
0.217846446	0.257689014	0.310000455	0.214464085	0.505135404	0.030727352	0.764611438
0.270065621	0.409460865	0.18311659	0.137356924	0.337429315	0.824496024	0.47467383
0.01162043	0.349374611	0.345123337	0.293881622	0.320952772	0.08323987	0.822498898
0.407398326	0.009558082	0.360509075	0.222534517	0.69035889	0.661870913	0.704437433
0.145080225	0.307262722	0.442527866	0.105129188	0.805438883	0.701307125	0.158855807
0.265121486	0.283910615	0.044177925	0.406789975	0.724865479	0.484106826	0.003427587
0.202870721	0.121065449	0.317291614	0.358772215	0.261406704	0.442476975	0.300526757
0.293687025	0.284408811	0.374459719	0.047444445	0.657371243	0.51847781	0.356544551
0.219150784	0.15235821	0.026964449	0.601526558	0.406731957	0.183995822	0.197781205
0.303318171	0.254919324	0.213692313	0.228070192	0.769101448	0.998033844	0.83480881
0.268282681	0.309261801	0.41827218	0.004183338	0.942099383	0.644327723	0.506040373
0.389320368	0.193811382	0.061390148	0.355478103	0.924231289	0.745273259	0.51765643
0.210047903	0.23487911	0.347124409	0.207948579	0.859826862	0.690064576	0.455322383
0.041159738	0.593529257	0.157349342	0.207961663	0.542680783	0.285917113	0.878920891
0.328196369	0.074580152	0.350210192	0.247013286	0.73584936	0.550193165	0.53651673
0.343546437	0.236163906	0.098295602	0.321994056	0.372635265	0.250906814	0.180828409
0.32936563	0.233719957	0.205107654	0.231806759	0.938582996	0.364410311	0.784091258
0.200587377	0.298082185	0.233605482	0.267724956	0.31523557	0.724231425	0.977567324
0.154625684	0.046087334	0.287853094	0.511433888	0.147223749	0.689544766	0.428890384
0.137319112	0.453727219	0.326253934	0.082699735	0.091831764	0.006457821	0.966591577
0.025812305	0.075056895	0.744091365	0.155039435	0.069746495	0.549338193	0.847010612
0.197615367	0.498781179	0.261374731	0.042228723	0.662790466	0.779361495	0.408638294
0.201585499	0.307412926	0.090767574	0.400234001	0.223184322	0.318905005	0.746426674
0.284005445	0.094593526	0.341205532	0.280195497	0.961930765	0.350375465	0.490448101
0.360018284	0.134038875	0.42028751	0.085655331	0.274743498	0.753969207	0.410455991
0.314523455	0.032074291	0.286350436	0.367051817	0.78281421	0.30842235	0.014433129
0.487925509	0.077371378	0.281812535	0.152890578	0.553822529	0.845246878	0.334273906
0.258761573	0.145742821	0.357453076	0.238042531	0.49860892	0.410119392	0.05662512
0.283075476	0.111305208	0.089537145	0.516082172	0.98981634	0.913228388	0.959522311
0.218288194	0.20228937	0.368913784	0.210508652	0.113340986	0.791948651	0.880974726

Source: Author (2021).

Table 6.5 shows the dependent variables for yields calculated using the Improved Swing-Cut Method from data set 1.

Table 6.5: Dependent variables for yields of distillates from data set 1.

Y_{LPG}	Y_{N}	Y_{K}	Y_{LD}	Y_{HD}	Y_{ATR}
1.112979447	6.391693324	8.602618808	12.28891701	8.495072666	63.01181936
1.072063092	8.52572334	10.15903877	12.87892558	6.260683754	60.99258915
1.054074163	9.183486314	10.66630271	12.32009492	5.095686541	61.57402297
1.07623991	7.196822774	7.728944012	16.99127767	4.969574266	61.9312527
1.055327277	6.524707024	9.665623045	12.89293887	9.061538651	60.68594921
0.996734221	8.551955198	11.08659676	12.56866125	6.404093078	60.27420124
1.125690764	7.938020818	7.745318478	14.37779963	6.15301811	62.56038903
1.058098703	7.61963125	10.17338087	11.14224673	8.120014715	61.77857741
1.059461984	8.172713296	9.730837796	14.1709648	5.711972845	61.04304586
1.124670153	7.397891996	11.30042447	12.0251454	5.846658475	62.20401675
1.042996947	6.534520036	9.640285212	15.12597619	6.161733789	61.38359995
1.039589608	7.034375682	10.39031521	12.33598285	8.573609003	60.50952396
0.960971704	10.34100461	7.453845096	16.3635732	5.112323854	59.64662111
1.101652578	6.418704612	13.20576879	10.43029734	6.775577193	61.96357198
1.08553106	7.535092684	7.645737283	15.30044854	6.529798652	61.79909665
1.022593064	7.192380609	11.96102912	12.83153224	5.217778601	61.66495477
0.955199589	6.895456193	10.20221321	17.37354873	4.795714958	59.65534694
0.989578574	8.623437466	8.888704505	11.75043539	8.787969081	60.846625
1.088400601	7.937137017	12.21053192	10.51870197	6.833535099	61.30288683
0.99919678	9.335847523	10.97405095	12.63607316	5.933095033	60.00121078
1.014967721	7.276365559	10.59868669	13.44456443	6.400065269	61.15421383
1.027624121	9.155450587	10.19009454	11.49072715	7.081466414	60.94199736
1.022032553	6.92028075	10.5904758	11.62347288	7.921359896	61.81315452
1.042852655	8.683317152	10.12380696	11.3131919	8.110306858	60.61282375
1.08251272	9.852127911	8.000234557	13.87045511	6.517449485	60.56505481
1.044363394	6.32872795	11.45129764	12.34207674	6.983817804	61.74226927
1.022013327	7.492017266	9.762434831	13.07786072	8.40600499	60.12198138
1.012183628	8.698254874	7.889744298	13.61061238	8.352844519	60.32041348
1.052889202	6.533033405	12.42578254	10.84751199	6.656259231	62.38115796
1.067541396	8.979218688	10.82166926	10.19718137	7.428687763	61.39799736
1.066856113	6.421055817	12.8195656	8.068005148	9.464642001	62.05548482
1.009412928	9.012821496	7.110468593	12.65054505	9.124621232	60.97846203
1.118558003	8.662090572	8.661621862	13.25242387	7.556871853	60.63655793
0.939455482	8.483847362	12.92793867	9.695169841	8.453207453	59.37528662
0.99325345	7.62379369	8.095118599	15.07204295	7.295121014	60.80504191
1.018460729	8.754408764	7.329153478	14.24308449	7.776520241	60.7632162
0.993293508	6.873214154	10.99597719	10.89621722	9.09349848	61.03383632
1.00842928	7.634543497	7.609387652	13.87092432	7.534421349	62.23648023
1.129300535	8.450302457	10.10635619	10.26954345	9.153735301	60.78140154
1.167217653	8.770823394	9.215253899	12.4005736	6.620636444	61.72346971
1.02719793	8.247600964	8.999080984	12.4476089	7.969082747	61.19822196
1.028091343	7.471550689	11.12386276	9.354329016	9.242577734	61.67136871
1.04988613	7.854940948	8.880014559	14.69499899	5.601912634	61.80990984
1.01547338	7.31726776	14.51751985	11.46546974	5.655224767	59.91029087
0.998378404	7.802788981	9.066539629	13.21559778	9.113041663	59.68134677
1.093477271	6.772108753	11.14467114	13.0100217	6.213303915	61.65992154
1.057915816	7.254202798	10.58626509	11.44715604	8.239718295	61.30524393
0.976765134	6.765418868	14.1596824	12.87943257	5.27208502	59.82379968
1.025083982	6.24626421	10.81033635	13.21380644	6.980222913	61.61565037
1.068470165	8.554457461	10.97686061	10.28116189	7.024098738	61.99025382

1.036140673	7.413979839	7.190032242	13.17154688	8.884843037	62.19825311
1.027142583	8.979312144	7.434455899	12.70159097	8.457292774	61.28935773
1.063558629	8.498998067	7.329992068	11.98686144	9.037918264	61.97791313
1.101476902	7.129759189	10.38181537	11.86944255	8.223241191	61.18577734
1.045766035	7.69237437	5.959747772	15.87394637	6.099314121	63.23069724
1.005262382	8.817260963	6.32697735	15.79577489	6.839592259	61.10220017
1.089418159	6.347488091	8.908143119	14.2452995	6.531084284	62.77861753
1.028436956	7.539640271	11.96019817	13.73337341	5.404873619	60.21613773
1.095872271	8.890606902	6.361174021	13.91623109	8.251379678	61.37687098
1.00242252	9.240019091	8.618974087	10.94573912	9.282412782	60.79533556
1.014015914	7.496494068	12.63679079	10.97337412	6.360566234	61.40939982
1.108905555	6.378048599	8.906360938	11.59024978	9.474361726	62.44178634
0.965137688	8.344275429	8.259750897	14.39729182	7.842192946	60.07053401
1.036886299	7.35839801	10.39663499	11.55555871	7.610252995	61.93559703
1.02119965	7.663310877	7.655903139	17.17586544	5.293440686	61.07761478
1.002978085	6.097955708	12.91017701	8.688207138	9.035115583	62.15936538
1.056759543	6.655506809	13.29102632	11.97495408	5.284190519	61.63009152
1.042340787	8.186617946	11.74004399	10.33668338	7.984174087	60.59649422
1.088823315	8.56196082	10.30733901	8.916633311	8.966714833	62.05571281
1.086817937	7.037782097	9.566742935	13.52392911	6.699472605	61.98069044
1.060669253	7.922869518	7.325259047	16.4303006	5.941343908	61.20963583
1.016706756	7.594923196	12.75443988	10.91525537	7.273034267	60.32834308
1.09223416	7.071411188	7.906600289	16.48418815	5.694156821	61.64423809
1.060667361	8.595859501	9.979065434	12.38383237	6.243405402	61.63131238
1.09048865	9.046348284	10.24684007	9.97703616	8.711698186	60.81637027
1.005616894	8.088718103	9.052308162	10.34413049	9.690667343	61.70896346
1.076917829	6.902969424	9.781734405	11.7195776	8.214735713	62.20093406
1.056562573	8.860536347	9.903427003	11.93668466	7.766716475	60.36138943
1.022385609	6.699139449	7.814576274	12.55175983	8.879161303	62.93135445
1.030562857	8.728609655	11.94669897	11.47854215	5.622440551	61.08144782
1.065866087	9.849445405	9.784622465	11.96346426	7.039714844	60.18143086
0.996281131	8.908356793	9.932541886	11.2980744	7.206715947	61.54792782
1.070313514	8.97598003	9.843726711	11.33019023	7.393457302	61.27729651
1.033263932	7.890259767	8.955704296	16.14161404	5.380789187	60.48109618
1.066040356	8.601639737	9.250377252	12.25458692	7.041447152	61.68001465
1.006176917	7.332667535	8.935538204	12.43652726	8.787808268	61.39041364
1.026759573	9.247362252	7.94622405	14.69488613	5.86627534	61.10694431
1.04692494	7.214817472	11.668321	13.69421085	4.959066795	61.30641808
1.08680523	6.252067351	10.87426511	10.90082523	7.694049976	63.09443462
1.05870926	6.897405909	8.783295169	17.9490028	4.917252109	60.2780199
1.18237407	6.823501195	10.74918317	13.62214099	5.515837982	62.00713547
1.033670553	8.808097534	11.69390639	10.9681638	7.520151985	59.85622293
1.021810574	6.627677984	9.550778071	14.81917937	6.124179191	61.74740994
1.069982242	9.12115061	7.494731696	13.10455019	7.274760225	61.82970546
1.067145032	7.65346854	12.10564878	10.6827651	7.538834432	60.84148183
1.061430281	8.400732028	7.630416791	10.94356447	9.5847253	62.27648283
1.027866911	8.41593033	11.72740532	9.778042623	7.952320941	60.98748799
1.072195178	7.888292967	9.238707596	10.8350439	9.290918311	61.56806037
1.026151928	8.533766301	10.14352991	12.40361524	5.169856053	62.62029371
1.075620368	6.805329762	12.50641063	12.73798389	5.392326207	61.37424256

Source: Author (2021).

Table 6.6 shows the dependent variables for specific gravity of distillates calculated using the Improved Swing-Cut Method from data set 1.

Table 6.6: Dependent variables for specific gravity of distillates from data set 1.

SG_{LPG}	SG_{N}	SG_{K}	SG_{LD}	SG_{HD}	SG_{ATR}
0.54796633	0.735040789	0.815308475	0.863090005	0.901993516	0.986544204
0.547453858	0.735038641	0.817945003	0.866821417	0.903237105	0.982389511
0.545854201	0.73492886	0.823117631	0.867585969	0.900425871	0.980323092
0.548496209	0.734931687	0.809006285	0.866094934	0.907466353	0.984194701
0.5477253	0.72259269	0.799885584	0.856787041	0.896259566	0.981421099
0.546498227	0.724667215	0.810593766	0.860981203	0.894060578	0.976554614
0.548407103	0.745323105	0.822417424	0.869289582	0.909539017	0.987420366
0.548929994	0.736673172	0.817894339	0.864776536	0.900025624	0.983396924
0.547290726	0.731903043	0.8137004	0.865490586	0.902734354	0.981531182
0.547596734	0.737524474	0.824793092	0.872923513	0.908821582	0.986399229
0.548640405	0.726991934	0.804508601	0.863398798	0.90229476	0.981972713
0.548615333	0.726665675	0.804982185	0.860022528	0.897522995	0.981133767
0.544770604	0.725550471	0.804154603	0.854239613	0.890557673	0.972485391
0.54828401	0.731785364	0.821400923	0.871846392	0.905993952	0.985530013
0.546561389	0.730339553	0.808461412	0.860725282	0.901776395	0.982921325
0.548469986	0.729434971	0.8163856	0.868633429	0.901619468	0.98080165
0.545475297	0.707755579	0.789664605	0.854341093	0.891860883	0.972733599
0.54532699	0.723306845	0.80615171	0.852249719	0.886176307	0.975606651
0.547916108	0.735689358	0.823208482	0.87027003	0.904331538	0.983986773
0.547638619	0.730676452	0.815482194	0.864462937	0.897441168	0.977457299
0.546289533	0.720830397	0.806597614	0.859936967	0.894961009	0.978127885
0.546962125	0.733507994	0.818207174	0.862842097	0.896150658	0.979288425
0.548887323	0.729727549	0.811910794	0.862404101	0.896710057	0.981225691
0.54655931	0.729869992	0.813577029	0.860159734	0.894976158	0.979593646
0.546401851	0.738173968	0.817274334	0.8631616	0.902118814	0.981756683
0.547583612	0.723303242	0.809615444	0.863538902	0.898599832	0.981383833
0.546519641	0.720279644	0.800282925	0.855089327	0.892569117	0.977955392
0.5456325	0.724401714	0.80232196	0.852295225	0.890007769	0.97676243
0.547937209	0.728675157	0.81871446	0.867811645	0.90031967	0.982623675
0.546635616	0.736258621	0.823321178	0.865391362	0.898482292	0.981669907
0.547467475	0.726275804	0.817348852	0.863490218	0.8951283	0.982813033
0.547466888	0.732807687	0.808174555	0.854048443	0.891080554	0.978668769
0.547942657	0.739369156	0.816280304	0.865218165	0.906423467	0.985383914
0.545580426	0.715496607	0.806117016	0.855048964	0.883419079	0.971642445
0.547651264	0.724156308	0.799323004	0.855301642	0.89358317	0.977730297
0.547946891	0.732699447	0.80692514	0.857024682	0.895681204	0.979459197
0.547450107	0.719918	0.804331705	0.856232468	0.88955681	0.97773908
0.547639107	0.730221679	0.807848393	0.85700626	0.893692172	0.979674029
0.546909348	0.736156883	0.819098963	0.864001682	0.901520959	0.985137791
0.547164066	0.745018866	0.827559158	0.871522563	0.911148194	0.9881885
0.547151916	0.730038672	0.81070598	0.858594386	0.894447509	0.979614405
0.547235238	0.727559044	0.814530138	0.859676142	0.891769117	0.980079586
0.549084989	0.737820878	0.815567683	0.867133698	0.904626392	0.983061024
0.545951664	0.716590272	0.810543801	0.865456313	0.896749215	0.976908812
0.547329423	0.721309961	0.797216995	0.852665967	0.890584025	0.976892264
0.548088939	0.731423985	0.815380651	0.86899503	0.90621069	0.984676612
0.54739806	0.727874184	0.811551317	0.861524756	0.897265673	0.98171722
0.54787807	0.715721458	0.805931458	0.865190636	0.89701702	0.976157061
0.547169495	0.719399329	0.804467428	0.860314507	0.896029773	0.97980743
0.547424882	0.7386593	0.825763114	0.867728051	0.900475451	0.982826267

0.547919184	0.731926922	0.806868298	0.855830919	0.894179495	0.981503942
0.547373022	0.734909057	0.811922238	0.856885599	0.893775774	0.979867168
0.547213709	0.737262808	0.814967171	0.857845611	0.895312068	0.982345467
0.5474217	0.730078962	0.812359872	0.863420578	0.90173145	0.984233344
0.547883582	0.738444938	0.812638135	0.860827337	0.900059039	0.982725455
0.547350909	0.731624017	0.804523353	0.855660339	0.894988669	0.978417044
0.54851952	0.734231309	0.813196919	0.866010263	0.905047625	0.985526596
0.54704078	0.722816675	0.808755073	0.865219367	0.900325025	0.97886485
0.547766088	0.74114883	0.813871964	0.859843782	0.901635446	0.98434192
0.54743158	0.732664506	0.812642011	0.85612669	0.890150672	0.97808528
0.546107375	0.722321351	0.814785199	0.863472661	0.894467335	0.978109912
0.547568967	0.730974051	0.811314607	0.860258986	0.899013684	0.985601287
0.546938072	0.721760616	0.798157978	0.852294795	0.888954911	0.974875315
0.547736903	0.730112478	0.814584468	0.862507497	0.896705639	0.981210642
0.547585762	0.727238402	0.801604746	0.859831368	0.900179525	0.97954367
0.54730344	0.718740746	0.811557921	0.859733068	0.889673481	0.979022905
0.547405903	0.725321692	0.816932223	0.870286917	0.90309608	0.981879924
0.546382372	0.726574878	0.814418517	0.861835323	0.894880276	0.979395543
0.546846032	0.738634094	0.825045565	0.863976019	0.897115877	0.983537433
0.54795332	0.733068978	0.814017444	0.865685515	0.904000052	0.984366444
0.547325887	0.731301659	0.805687753	0.860771865	0.902328068	0.981749706
0.547228414	0.723136696	0.811540365	0.863152584	0.895545119	0.978410753
0.548576958	0.734772581	0.808009524	0.865533067	0.908070999	0.985035704
0.546021994	0.732612237	0.818956062	0.864308346	0.899017923	0.980890532
0.547335813	0.738104278	0.821436075	0.864357503	0.899626972	0.983257663
0.547925822	0.731495747	0.812454535	0.856467458	0.889993031	0.979367464
0.54755123	0.730810431	0.813863732	0.862269617	0.898860052	0.983570134
0.546510376	0.730786096	0.813461547	0.860997561	0.897140726	0.980176563
0.548473402	0.731078065	0.808114301	0.856861931	0.893407158	0.981664976
0.547036288	0.732165447	0.821216232	0.86777956	0.899483231	0.979628047
0.546521739	0.735906549	0.818842649	0.863989065	0.89980577	0.980607918
0.546999969	0.732135728	0.817519795	0.860698575	0.892476856	0.977929524
0.547275985	0.737841881	0.821254302	0.865017399	0.900109779	0.982321047
0.548843988	0.73119776	0.806316561	0.864339829	0.903915895	0.980928479
0.546573321	0.734890714	0.818503213	0.863106473	0.898851461	0.981721914
0.547171162	0.723929457	0.803800358	0.854748765	0.890677402	0.978534824
0.546892866	0.734356246	0.813472028	0.861355264	0.898328598	0.979299057
0.547685017	0.727761366	0.814049069	0.868649877	0.903488401	0.981320031
0.548073535	0.732800186	0.819646272	0.866531532	0.901292872	0.985174263
0.547650491	0.723558613	0.797361529	0.862386104	0.905503559	0.981268589
0.547380104	0.736674454	0.822309805	0.874995588	0.915243171	0.989445949
0.547392338	0.729682133	0.815000672	0.863701752	0.897358428	0.979218314
0.54820186	0.725449125	0.805162525	0.861923347	0.899162557	0.980585743
0.546907489	0.739791786	0.818752857	0.861656246	0.899257802	0.982348929
0.546032727	0.724943874	0.814450916	0.863447369	0.897458911	0.980716315
0.546941517	0.736884339	0.816604	0.857266921	0.893240995	0.982184967
0.545585637	0.725835554	0.815968149	0.860507508	0.891849157	0.978148085
0.546980519	0.732079824	0.81404134	0.85924871	0.895391102	0.982365572
0.547800964	0.738956123	0.824990645	0.868165692	0.900267509	0.981110555
0.547174623	0.726035122	0.815209843	0.869703672	0.904571041	0.982608627

Source: Author (2021).

Table 6.7 shows the dependent variables for sulfur content of distillates calculated using the Improved Swing-Cut Method from data set 1.

Table 6.7: Dependent variables for sulfur content of distillates from data set 1.

S_N	S_K	S_{LD}	S_{HD}	S_{ATR}
0.004411258	0.078810801	0.249719462	0.491839154	0.78147386
0.007436213	0.087007962	0.268075388	0.475345327	0.739370276
0.008048212	0.094645078	0.296470277	0.466698609	0.762040219
0.005636645	0.071346755	0.270279363	0.51015702	0.752821246
0.00300331	0.059605458	0.208251784	0.442352556	0.729528048
0.004822081	0.074093413	0.243294514	0.437838811	0.724537368
0.009459914	0.092198101	0.283068562	0.517050133	0.769621917
0.006594057	0.088880619	0.253403561	0.476643144	0.74549309
0.006235428	0.079281248	0.265361142	0.475686143	0.74080486
0.006372269	0.099945239	0.30846614	0.505994318	0.767923818
0.003547224	0.06692128	0.25143622	0.486861917	0.737981524
0.003985141	0.068756695	0.221470184	0.448987819	0.71837724
0.005649054	0.059581977	0.219636098	0.412095453	0.718610188
0.003927444	0.09736525	0.297966857	0.497585627	0.75647206
0.005195229	0.067238545	0.244611793	0.475545133	0.764181853
0.004536571	0.087195172	0.290413537	0.49255209	0.744140835
0.001865701	0.043420762	0.219253192	0.421404348	0.7137375
0.004525268	0.063803765	0.20115814	0.412227188	0.744672514
0.006820974	0.099037768	0.285646359	0.48252207	0.743968491
0.006725762	0.083037516	0.254782948	0.451629071	0.710800511
0.003386272	0.067247057	0.245136454	0.450240929	0.747027769
0.007474034	0.08663270	0.251917593	0.451742497	0.738672366
0.00428800	0.07916523	0.246072326	0.469892773	0.744560861
0.006277626	0.07845601	0.231553829	0.438170273	0.735156002
0.009979598	0.082568147	0.248121276	0.459108473	0.738078325
0.002958545	0.074420166	0.258083378	0.472073887	0.753130184
0.003506506	0.058632268	0.204638711	0.424019278	0.722782489
0.005013108	0.058731433	0.196902022	0.417117603	0.732599819
0.003794354	0.091121588	0.288675912	0.487973019	0.76495521
0.008293995	0.096127027	0.267038856	0.460223571	0.753740444
0.003350586	0.088572345	0.252395751	0.455592627	0.762092904
0.006968032	0.068955924	0.201154186	0.432080309	0.734853605
0.008789816	0.083757888	0.245721846	0.475302009	0.730887669
0.002947031	0.066439936	0.206604477	0.38689441	0.705178995
0.003968902	0.057104217	0.212639315	0.446438002	0.728679575
0.006817306	0.068126282	0.215710114	0.449812684	0.727384901
0.002904671	0.065865366	0.212989233	0.429059719	0.735139079
0.004999775	0.067974849	0.229354272	0.464377758	0.761169682
0.007551604	0.087926283	0.24005179	0.453962382	0.742491421
0.010679277	0.101501097	0.288542436	0.499767102	0.763595827
0.005772092	0.073541495	0.227814943	0.448085942	0.743038418
0.00454715	0.081591506	0.233217023	0.442091859	0.75290697
0.007172629	0.08404237	0.273393838	0.504055442	0.744667522
0.002951201	0.076720014	0.268118764	0.437367825	0.721317403
0.00378422	0.055243157	0.188679311	0.412163138	0.705381603
0.004357964	0.084226576	0.280096125	0.495462802	0.750832532
0.004433617	0.076632007	0.238719474	0.456280839	0.745737692
0.002425886	0.071192668	0.260531749	0.453347387	0.703581018
0.002540394	0.065276423	0.243767953	0.461973849	0.751938626
0.008429633	0.101977641	0.283786107	0.478653056	0.761093183

0.005276866	0.066604655	0.214324156	0.459353963	0.760177564
0.007631097	0.074650888	0.217610221	0.446244075	0.743549355
0.007949957	0.078490526	0.22239071	0.456173947	0.761979954
0.004537416	0.077687155	0.242873805	0.465835902	0.745778934
0.007044075	0.073415447	0.256719956	0.500031888	0.782700088
0.006448346	0.06295575	0.218186354	0.454424672	0.738110261
0.004309746	0.077617174	0.270543463	0.508380865	0.771400686
0.003895143	0.07327659	0.261979925	0.45858863	0.721818937
0.009740735	0.077527917	0.225707335	0.469537324	0.747125911
0.007066731	0.076751845	0.209515228	0.426203465	0.730521252
0.003733428	0.082020124	0.270636335	0.451371761	0.753786105
0.003764614	0.072890603	0.229983651	0.471056539	0.772178641
0.00390263	0.054705033	0.196036932	0.41881579	0.714657101
0.004892805	0.081603519	0.252508821	0.467890181	0.75580581
0.004651979	0.059921034	0.241225045	0.474335714	0.737035716
0.002343201	0.07843971	0.242134457	0.444554848	0.761370191
0.003465849	0.088406424	0.301675111	0.486276841	0.752704892
0.005129148	0.081079863	0.241692108	0.437424075	0.735980065
0.008504782	0.098836595	0.256514181	0.460862403	0.767872303
0.005045393	0.079815843	0.264900852	0.491406442	0.758315537
0.005820659	0.065159505	0.241673295	0.476398439	0.744318062
0.003950513	0.078304687	0.247234299	0.442591679	0.722279712
0.00536381	0.070376248	0.260428971	0.505465164	0.74694303
0.006886098	0.086244515	0.270666938	0.46416546	0.762597679
0.00899898	0.093041836	0.245412813	0.453020585	0.737514305
0.005692476	0.077230865	0.214317447	0.437589037	0.747702767
0.004518355	0.078938238	0.247061935	0.472974757	0.765241571
0.006729771	0.078146946	0.233668601	0.441101826	0.730831201
0.004240159	0.068986401	0.223783154	0.469035025	0.771238522
0.006745433	0.093621538	0.286470249	0.46729828	0.741467258
0.00919452	0.087000522	0.248634396	0.447699232	0.727354144
0.006455914	0.085073804	0.249534382	0.450706046	0.74966158
0.008763006	0.092182416	0.259101877	0.466035902	0.746789831
0.005666673	0.069896976	0.251004157	0.484702506	0.715783254
0.007474459	0.085438725	0.257448111	0.465785954	0.760189341
0.003773305	0.062873295	0.209367078	0.437462805	0.745740191
0.007755376	0.076830488	0.249609824	0.46316404	0.742746213
0.00436595	0.082171006	0.287660197	0.486474161	0.743071962
0.004000269	0.090149009	0.277549769	0.49532812	0.780936581
0.003420822	0.056017058	0.24455896	0.477424401	0.721045488
0.004690323	0.092669264	0.309598181	0.516645074	0.768947238
0.006323091	0.083135416	0.242379316	0.440084123	0.711476177
0.003423798	0.066737085	0.251670286	0.481468008	0.747687768
0.009493388	0.084694984	0.247588223	0.4697824	0.761349578
0.004366868	0.081086762	0.253036151	0.446804182	0.745730034
0.007706498	0.080716698	0.221039614	0.451779734	0.770087019
0.005129455	0.082458119	0.244223766	0.431892253	0.749077181
0.005969182	0.078463256	0.226075189	0.448970163	0.755335413
0.008067145	0.100215821	0.303547812	0.496050792	0.769344319
0.003661800	0.084293562	0.29233006	0.483998843	0.74997722

Source: Author (2021).

6.7.3 Appendix C: Average Errors for each Surrogate Model

The least squares regression errors of each distillation component for each surrogate model are shown in Tables 6.8 to 6.11. Each error corresponds to the average error from the 100 points in the data set.

Table 6.8: Least Squares errors for Data Set 1 using the CSW model.

	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	Model 8	Model 9	Model 10
YLD$_{LPG}$	4.83E-30	4.83E-30	4.83E-30	4.83E-30	1.68E-29	1.68E-29	1.68E-29	1.68E-29	4.43E-30	4.43E-30
YLD$_{N}$	6.90E-01	2.85E-03	3.21E-16	3.21E-16	2.85E-03	6.90E-01	3.21E-16	1.14E-02	3.21E-16	3.21E-16
YLD$_{K}$	3.25E+00	6.61E-03	1.62E-15	1.62E-15	6.61E-03	3.25E+00	1.62E-15	7.66E-02	1.62E-15	1.62E-15
YLD$_{LD}$	3.65E+00	3.75E-03	1.09E-15	1.09E-15	3.75E-03	3.65E+00	1.09E-15	1.14E-01	1.09E-15	1.09E-15
YLD$_{HD}$	1.84E+00	1.13E-04	3.38E-16	3.38E-16	2.95E-03	1.84E+00	1.19E-03	6.13E-02	3.38E-16	3.38E-16
YLD$_{ATR}$	2.62E-17	2.62E-17	2.62E-17	2.62E-17	1.37E-01	1.37E-01	1.37E-01	1.37E-01	2.62E-17	2.62E-17
SG$_{LPG}$	1.29E-09	1.29E-09	1.29E-09	1.29E-09	5.32E-08	5.32E-08	5.32E-08	5.32E-08	1.29E-09	1.29E-09
SG$_{N}$	2.14E-05	3.09E-07	2.82E-07	2.82E-07	3.09E-07	2.14E-05	2.82E-07	8.35E-07	2.82E-07	2.82E-07
SG$_{K}$	3.01E-05	8.00E-07	5.23E-07	5.23E-07	8.00E-07	3.01E-05	5.23E-07	1.25E-06	5.23E-07	5.23E-07
SG$_{LD}$	1.42E-05	1.55E-07	1.17E-07	1.17E-07	1.55E-07	1.42E-05	1.17E-07	4.41E-07	1.17E-07	1.17E-07
SG$_{HD}$	7.40E-06	2.21E-07	2.16E-07	2.16E-07	2.21E-07	7.40E-06	2.16E-07	4.01E-07	2.16E-07	2.16E-07
SG$_{ATR}$	2.70E-09	2.70E-09	2.70E-09	2.70E-09	2.70E-09	2.70E-09	2.70E-09	2.70E-09	2.70E-09	2.70E-09
S$_{N}$	3.17E-06	1.82E-07	3.39E-08	3.39E-08	1.82E-07	3.17E-06	3.39E-08	1.61E-07	3.39E-08	3.39E-08
S$_{K}$	8.98E-05	2.10E-06	1.76E-06	1.76E-06	2.10E-06	8.98E-05	1.76E-06	5.43E-06	1.76E-06	1.76E-06
S$_{LD}$	5.55E-04	6.37E-06	4.49E-06	4.49E-06	6.37E-06	5.55E-04	4.49E-06	2.44E-05	4.49E-06	4.49E-06
S$_{HD}$	1.46E-04	6.31E-06	4.05E-06	4.05E-06	6.69E-06	1.62E-04	4.05E-06	2.71E-05	4.05E-06	4.05E-06
S$_{ATR}$	1.23E-07	1.23E-07	1.23E-07	1.23E-07	3.15E-05	3.15E-05	3.15E-05	3.15E-05	1.23E-07	1.23E-07

Source: Author (2021).

Table 6.9: Least Squares errors for Data Set 2 using the CSW model.

	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	Model 8	Model 9	Model 10
YLD_{LPG}	1.50E-29	1.50E-29	1.50E-29	1.50E-29	2.39E-29	2.39E-29	2.39E-29	2.39E-29	1.50E-29	1.50E-29
YLD_{N}	8.20E-01	2.18E-03	7.03E-16	7.03E-16	2.18E-03	8.20E-01	7.03E-16	9.63E-03	7.03E-16	7.03E-16
YLD_{K}	3.31E+00	5.76E-03	1.60E-15	1.60E-15	5.76E-03	3.31E+00	1.60E-15	7.38E-02	1.60E-15	1.60E-15
YLD_{LD}	4.82E+00	2.56E-03	1.46E-15	1.46E-15	2.56E-03	4.82E+00	1.46E-15	1.23E-01	1.46E-15	1.46E-15
YLD_{HD}	1.92E+00	1.44E-04	4.04E-16	4.04E-16	3.14E-03	1.92E+00	1.88E-03	7.59E-02	4.04E-16	4.04E-16
YLD_{ATR}	8.32E-16	8.32E-16	8.32E-16	8.32E-16	1.22E-01	1.22E-01	1.22E-01	1.22E-01	8.32E-16	8.32E-16
SG_{LPG}	1.73E-09	1.73E-09	1.73E-09	1.73E-09	4.47E-08	4.47E-08	4.47E-08	4.47E-08	1.73E-09	1.73E-09
SG_{N}	2.60E-05	4.47E-07	3.65E-07	3.65E-07	4.47E-07	2.60E-05	3.65E-07	6.43E-07	3.65E-07	3.65E-07
SG_{K}	3.95E-05	8.18E-07	8.42E-07	8.42E-07	8.18E-07	3.95E-05	8.42E-07	1.45E-06	8.42E-07	8.42E-07
SG_{LD}	1.27E-05	1.78E-07	1.85E-07	1.85E-07	1.78E-07	1.27E-05	1.85E-07	6.34E-07	1.85E-07	1.85E-07
SG_{HD}	8.37E-06	2.75E-07	2.74E-07	2.74E-07	2.75E-07	8.37E-06	2.74E-07	4.52E-07	2.74E-07	2.74E-07
SG_{ATR}	1.93E-09	1.93E-09	1.93E-09	1.93E-09	1.93E-09	1.93E-09	1.93E-09	1.93E-09	1.93E-09	1.93E-09
S_{N}	3.69E-06	1.66E-07	4.42E-08	4.42E-08	1.66E-07	3.69E-06	4.42E-08	1.22E-07	4.42E-08	4.42E-08
S_{K}	1.19E-04	2.76E-06	2.82E-06	2.82E-06	2.76E-06	1.19E-04	2.82E-06	6.11E-06	2.82E-06	2.82E-06
S_{LD}	4.88E-04	6.79E-06	7.06E-06	7.06E-06	6.79E-06	4.88E-04	7.06E-06	3.50E-05	7.06E-06	7.06E-06
S_{HD}	1.63E-04	6.67E-06	5.19E-06	5.19E-06	5.98E-06	1.57E-04	5.19E-06	1.82E-05	5.19E-06	5.19E-06
S_{ATR}	1.13E-07	1.13E-07	1.13E-07	1.13E-07	2.76E-05	2.76E-05	2.76E-05	2.76E-05	1.13E-07	1.13E-07

Source: Author (2021).

Table 6.10: Least Squares errors for Data Set 1 using the ISW model.

	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	Model 8	Model 9	Model 10
YLD_{LPG}	4.83E-30	4.83E-30	4.83E-30	4.83E-30	1.68E-29	1.68E-29	1.68E-29	1.68E-29	4.43E-30	4.43E-30
YLD_{N}	6.90E-01	2.85E-03	3.21E-16	3.21E-16	2.85E-03	6.90E-01	3.21E-16	1.14E-02	3.21E-16	3.21E-16
YLD_{K}	3.25E+00	6.61E-03	1.62E-15	1.62E-15	6.61E-03	3.25E+00	1.62E-15	7.66E-02	1.62E-15	1.62E-15
YLD_{LD}	3.65E+00	3.75E-03	1.09E-15	1.09E-15	3.75E-03	3.65E+00	1.09E-15	1.14E-01	1.09E-15	1.09E-15
YLD_{HD}	1.84E+00	1.13E-04	3.38E-16	3.38E-16	2.95E-03	1.84E+00	1.19E-03	6.13E-02	3.38E-16	3.38E-16
YLD_{ATR}	2.62E-17	2.62E-17	2.62E-17	2.62E-17	1.37E-01	1.37E-01	1.37E-01	1.37E-01	2.62E-17	2.62E-17
SG_{LPG}	1.29E-09	1.29E-09	1.29E-09	1.29E-09	5.32E-08	5.32E-08	5.32E-08	5.32E-08	1.29E-09	1.29E-09
SG_{N}	2.17E-05	1.36E-07	7.89E-08	7.89E-08	1.36E-07	2.17E-05	7.89E-08	5.94E-07	7.89E-08	7.89E-08
SG_{K}	2.87E-05	2.47E-07	8.51E-08	8.51E-08	2.47E-07	2.87E-05	8.51E-08	7.61E-07	8.51E-08	8.51E-08
SG_{LD}	1.32E-05	4.52E-08	1.37E-08	1.37E-08	4.52E-08	1.32E-05	1.37E-08	3.22E-07	1.37E-08	1.37E-08
SG_{HD}	7.94E-06	1.15E-08	3.44E-09	3.44E-09	1.15E-08	7.94E-06	3.44E-09	2.47E-07	3.44E-09	3.44E-09
SG_{ATR}	2.70E-09	2.70E-09	2.70E-09	2.70E-09	2.70E-09	2.70E-09	2.70E-09	2.70E-09	2.70E-09	2.70E-09
S_{N}	3.24E-06	1.30E-07	1.91E-08	1.91E-08	1.30E-07	3.24E-06	1.91E-08	1.34E-07	1.91E-08	1.91E-08
S_{K}	8.80E-05	3.04E-07	1.18E-07	1.18E-07	3.04E-07	8.80E-05	1.18E-07	3.89E-06	1.18E-07	1.18E-07
S_{LD}	5.21E-04	1.61E-06	7.40E-07	7.40E-07	1.61E-06	5.21E-04	7.40E-07	1.98E-05	7.40E-07	7.40E-07
S_{HD}	1.58E-04	2.11E-06	4.66E-07	4.66E-07	2.22E-06	1.74E-04	4.66E-07	2.49E-05	4.66E-07	4.66E-07
S_{ATR}	1.23E-07	1.23E-07	1.23E-07	1.23E-07	3.15E-05	3.15E-05	3.15E-05	3.15E-05	1.23E-07	1.23E-07

Source: Author (2021).

Table 6.11: Least Squares errors for Data Set 2 using the ISW model.

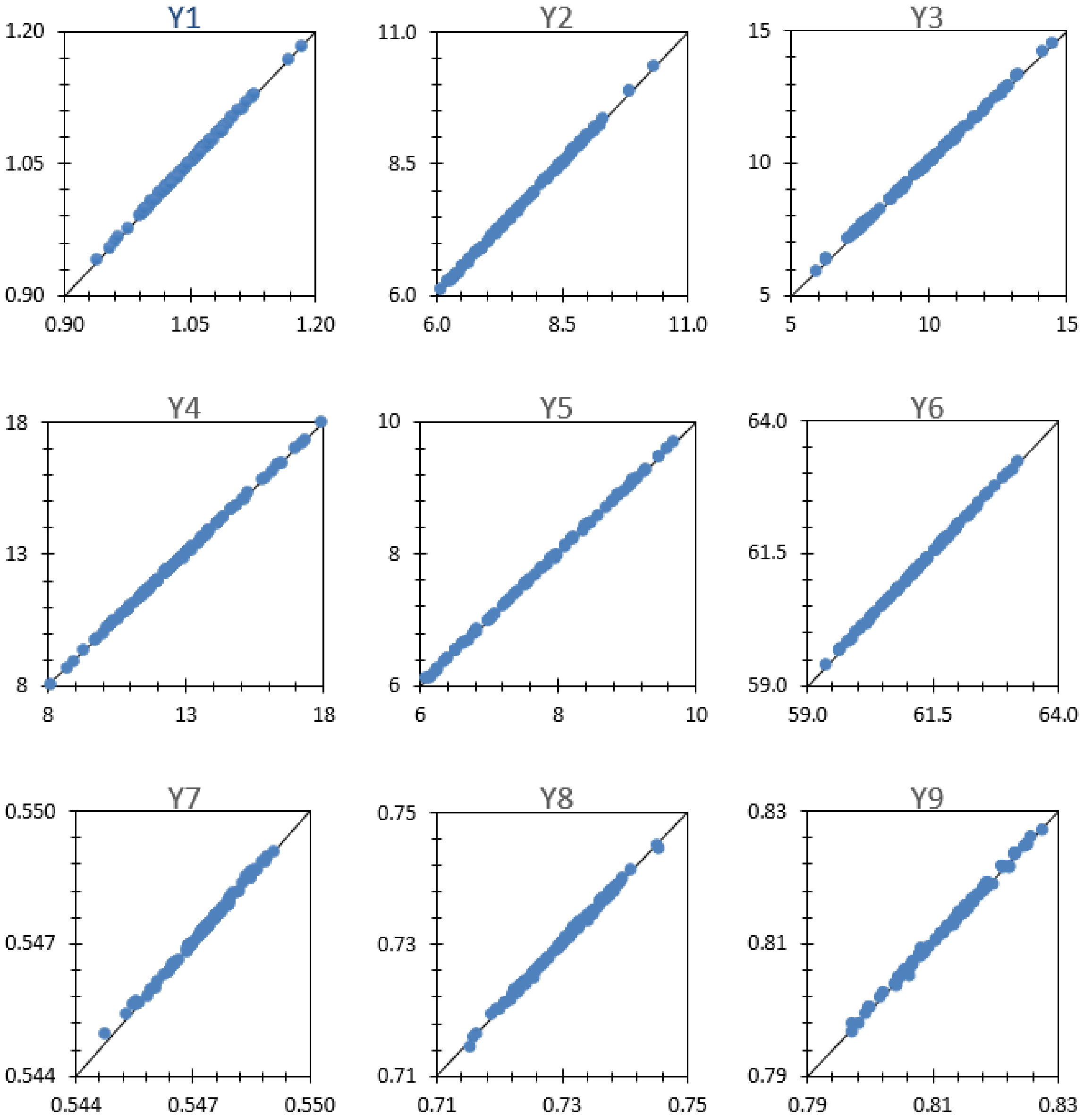
	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	Model 8	Model 9	Model 10
YLD$_{LPG}$	1.50E-29	1.50E-29	1.50E-29	1.50E-29	2.39E-29	2.39E-29	2.39E-29	2.39E-29	1.50E-29	1.50E-29
YLD$_{N}$	8.20E-01	2.18E-03	7.03E-16	7.03E-16	2.18E-03	8.20E-01	7.03E-16	9.63E-03	7.03E-16	7.03E-16
YLD$_{K}$	3.31E+00	5.76E-03	1.60E-15	1.60E-15	5.76E-03	3.31E+00	1.60E-15	7.38E-02	1.60E-15	1.60E-15
YLD$_{LD}$	4.82E+00	2.56E-03	1.46E-15	1.46E-15	2.56E-03	4.82E+00	1.46E-15	1.23E-01	1.46E-15	1.46E-15
YLD$_{HD}$	1.92E+00	1.44E-04	4.04E-16	4.04E-16	3.14E-03	1.92E+00	1.88E-03	7.59E-02	4.04E-16	4.04E-16
YLD$_{ATR}$	8.32E-16	8.32E-16	8.32E-16	8.32E-16	1.22E-01	1.22E-01	1.22E-01	1.22E-01	8.32E-16	8.32E-16
SG$_{LPG}$	1.73E-09	1.73E-09	1.73E-09	1.73E-09	4.47E-08	4.47E-08	4.47E-08	4.47E-08	1.73E-09	1.73E-09
SG$_{N}$	2.62E-05	1.61E-07	1.04E-07	1.04E-07	1.61E-07	2.62E-05	1.04E-07	5.16E-07	1.04E-07	1.04E-07
SG$_{K}$	3.85E-05	1.67E-07	1.48E-07	1.48E-07	1.67E-07	3.85E-05	1.48E-07	8.09E-07	1.48E-07	1.48E-07
SG$_{LD}$	1.27E-05	5.11E-08	1.73E-08	1.73E-08	5.11E-08	1.27E-05	1.73E-08	4.10E-07	1.73E-08	1.73E-08
SG$_{HD}$	8.40E-06	9.54E-09	3.99E-09	3.99E-09	9.54E-09	8.40E-06	3.99E-09	3.23E-07	3.99E-09	3.99E-09
SG$_{ATR}$	1.93E-09	1.93E-09	1.93E-09	1.93E-09	1.93E-09	1.93E-09	1.93E-09	1.93E-09	1.93E-09	1.93E-09
S$_{N}$	3.89E-06	1.59E-07	2.72E-08	2.72E-08	1.59E-07	3.89E-06	2.72E-08	1.56E-07	2.72E-08	2.72E-08
S$_{K}$	1.17E-04	3.19E-07	1.34E-07	1.34E-07	3.19E-07	1.17E-04	1.34E-07	4.56E-06	1.34E-07	1.34E-07
S$_{LD}$	4.83E-04	9.96E-07	5.90E-07	5.90E-07	9.96E-07	4.83E-04	5.90E-07	2.63E-05	5.90E-07	5.90E-07
S$_{HD}$	1.68E-04	3.87E-06	5.11E-07	5.11E-07	3.62E-06	1.63E-04	5.11E-07	1.86E-05	5.11E-07	5.11E-07
S$_{ATR}$	1.13E-07	1.13E-07	1.13E-07	1.13E-07	2.76E-05	2.76E-05	2.76E-05	2.76E-05	1.13E-07	1.13E-07

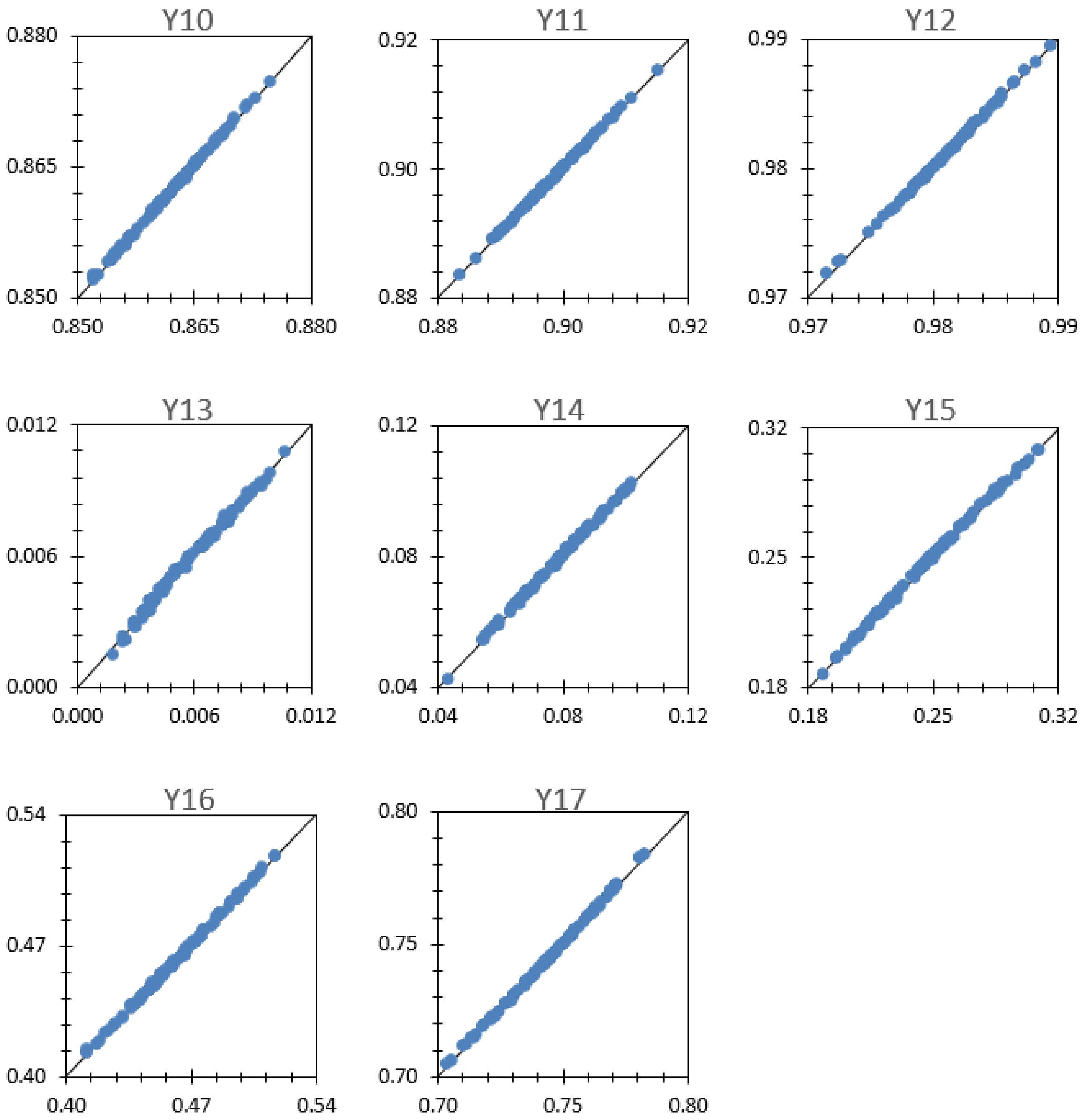
Source: Author (2021).

6.7.4 Appendix D: Cross plots for the best Surrogate Model

Figure 6.6 presents the cross plots for the $Y_{estimated}$ (y axis) versus the Y_{real} (x axis) for the yield, specific gravity, and sulfur content of all dependent variables in Surrogate Model 3 using the ISW method. The black line represents the $y = x$ function.

Figure 6.6: Plots for Surrogate Model 3 using the ISW method.





Source: Author (2021).

6.8 Nomenclature

Continuous Variables

B: maximum number allowed of basis

b_{jki}: bilinear basis of crude oil j and swing-cut k for dependent variable i

dv: total number of dependent variables

dv_s: number of sulfur content dependent variables

dv_sg: number of specific gravity dependent variables

dv_yld: number of yield dependent variables

E: objective function to be minimized

I_i: intercept basis for dependent variable i

l_{ji}: linear basis of crude oil j for dependent variable i

l_{ki}: linear basis of swing-cut k for dependent variable i

MP_{cut}^{fc}: mass-based property of final cut

MP_{cut}^{ic}: mass-based property of intermediate cut

MPI_{sw}: mass-based property of interface of swing-cut

MP_{mc1}: mass-based property of adjacent light micro cut

MP_{mc2}: mass-based property of adjacent heavy micro cut

MP_{sw}: mass-based property of swing-cut (bulk)

MP_{sw}^{light}: mass-based property of light fraction of swing-cut

MP_{sw}^{heavy}: mass-based property of heavy fraction of swing-cut

MPI_{sw}^{light}: mass-based property of interface of light fraction of swing-cut

MPI_{sw}^{heavy}: mass-based property of interface of heavy fraction of swing-cut

n: total number of points in the data set

$Q_{c,cdu}$: inlet volumetric feed flow for crude c incoming to cdu

Q_{cdu}: overall volumetric feed flow for cdu

Q_{cut}^{fc}: volumetric flow of final distillate cut

Q_{mc}: volumetric flow of micro cut mc (pseudo-components)

Q_{cut}^{ic}: volumetric flow of intermediate distillate cut

Q_{sw1} to Q_{sw3}: volumetric flow of swing-cuts (bulk)

Q_{sw1}^{light} to Q_{sw3}^{light}: volumetric flow of light fraction of swing-cuts

Q_{sw1}^{heavy} to Q_{sw3}^{heavy}: volumetric flow of heavy fraction of swing-cuts

Q_{sw}: volumetric flow of swing-cut

Q_{mc1}: volumetric flow of adjacent light micro cut

Q_{mc2}: volumetric flow of adjacent heavy micro cut

Q_{sw}^{light}: volumetric flow of light fraction of swing-cut

Q_{sw}^{heavy}: volumetric flow of heavy fraction of swing-cut

Q_K: final kerosene flow

Q_N: final naphtha flow

$Specific\ Gravity\ Error$: average error for the specific gravity dependent variables

$Sulfur\ Content\ Error$: average error for the sulfur content dependent variables

VP_{cut}^{fc}: volume-based property of final cut

VP_{cut}^{ic}: volume-based property of intermediate cut

VPI_{sw}: volume-based property of interface of swing-cut

VP_{mc1}: volume-based property of adjacent light micro cut

VP_{mc2}: volume-based property of adjacent heavy micro cut

VP_{sw}: volume-based property of swing-cut (bulk)

VP_{sw}^{light}: volume-based property of light fraction of swing-cut

VP_{sw}^{heavy}: volume-based property of heavy fraction of swing-cut

VPI_{sw}^{light}: volume-based property of interface of light fraction of swing-cut

VPI_{sw}^{heavy}: volume-based property of interface of heavy fraction of swing-cut

$T_{N/K}$: naphtha/kerosene cutpoint temperature

T_K: end cutpoint temperature of kerosene

T_N: initial cutpoint temperature of naphtha

x_j: generic independent variable for crude oil j

x_1 to x_4: independent variables for the four crude oils

x_5 to x_7: independent variables for the three swing-cuts

X_{jp}: value of independent decision variable of crude oil j at point p

X_{kp}: value of independent decision variable of swing-cut k at point p

Y_{ip}: value of dependent variable i at point p calculated by the surrogate model

y_{ip}: value of dependent variable i at point p calculated by the swing-cut methods

Y_{mc}: yields of micro cut mc

$Yield\ Error$: average error for the yield dependent variables

YLD_{mc}: yields of micro cut mc

Binary Variables

z_0: binary variable for intercept coefficient

z_j: binary variable for linear coefficient of crude oil j

z_{jk}: binary variable for bilinear coefficient of crude oil j and swing-cut k

z_k: binary variable for linear coefficient of swing-cut k

Parameters

M: big M parameter

Sets

CR: crude oils

DC: distillation component

DV: dependent variables

DV_S: sulfur content dependent variables

DV_SG: specific gravity dependent variables

DV_YLD: yield dependent variables

FC: final cuts of distillates

IC: intermediate cuts of distillates

MC_{fc}: micro cuts related to final cuts of distillates

MC_{ic}: micro cuts related to intermediate cuts of distillates

P: points in the data set

SW: swing-cuts

Subscripts

ATR: atmospheric residue

c: crude oils

CDU: crude distillation unit

cut: final distillate cut

HD: heavy diesel

$heavy$: heavy fraction of swing-cut

i: dependent variable in the identification model

j: crude oil in the identification model

k: swing-cut in the identification model

K: kerosene

LD: light diesel

$light$: light fraction of swing-cut

LPG: liquefied petroleum gas

N: naphtha

N/K: naphtha/kerosene

p: point of data set

sw: swing-cut

$sw1$: swing-cut 1, between naphtha and kerosene

$sw2$: swing-cut 2, between kerosene and light diesel

$sw3$: swing-cut 3, between light diesel and heavy diesel

Superscripts

cdu: crude distillation unit

fc: final cut

$final$: final distillate

$heavy$: heavy fraction of swing-cut

ic: intermediate cut

$light$: light fraction of swing-cut

sw: swing-cut

Abbreviations

ATR: atmospheric residue

CDU: crude distillation unit

CO1 to CO4: crude oils

CSW: conventional swing-cut

FG: fuel gas

FI: fractionation indices

FY: fixed yields

HD: heavy diesel

IMPL: Industrial modeling and programming language

ISW: improved swing-cut

K: kerosene

LD: light diesel

LHS: Latin hypercube sampling

LPG: liquefied petroleum gas

MINLP: mixed-integer nonlinear programming

MIQP: mixed-integer quadratic programming

N: naphtha

RTO: real time optimization

S: sulfur content

SG: specific gravity

SW1 to SW3: swing-cuts

TBP: true boiling point

UOPSS: unit-operation-port-state superstructure

WTR: weight transfer ratio

x_1 to x_4: independent variables for the four crude oils in the surrogate model

x_5 to x_7: independent variables for the three swing-cuts in the surrogate model

Y: dependent variable in the surrogate model

YLD: volume yield percent

6.9 References

AHMAD, I.; AHSAN, A.; MANABU, K.; IZZAT, I. C. Gray-box Soft Sensors in Process Industry: Current Practice, and Future Prospects in Era of Big Data. Processes, v. 8, n. 2, p. 243, 2020.

ALATTAS, A. M.; GROSSMANN, I. E.; PALOU-RIVERA, I. Integration of nonlinear crude distillation unit models in refinery planning optimization. Industrial & Engineering Chemistry Research, v. 50, n. 11, p. 6860-6870, 2011.

ALATTAS, A. M.; GROSSMANN, I. E.; PALOU-RIVERA, I. Refinery production planning: multiperiod MINLP with nonlinear CDU model. Industrial & engineering chemistry research, v. 51, n. 39, p. 12852-12861, 2012.

ANDERSON, R. N. Petroleum Analytics Learning Machine for optimizing the Internet of Things of today's digital oil field-to-refinery petroleum system. In 2017 IEEE International Conference on Big Data (Big Data), p. 4542-4545, 2017.

BECK, D. A. C.; CAROTHERS, J. M.; SUBRAMANIAN, V. R.; PFAENDTNER, J. Data science: Accelerating innovation and discovery in chemical engineering. AIChE Journal, v. 62, n. 5, p. 1402-1416, 2016.

BOUKOUVALA, F.; LI, J.; XIAO, X.; FLOUDAS, C. A. Data-Driven Modeling and Global optimization of industrial-scale petrochemical planning operations. In 2016 American Control Conference (ACC), p. 3340-3345, 2016.

BROOKS, R. W.; VAN WALSEM, F. D.; DRURY, J. Choosing cutpoints to optimize product yields. Hydrocarbon Processing, v. 78, n. 11, p. 53-60, 1999.

CHEN, M.; MAO, S.; LIU, Y. Big data: A survey. Mobile networks and applications, v. 19, n. 2, p. 171-209, 2014.

CHIANG, L.; LU, B.; CASTILLO, I. Big Data Analytics in Chemical Engineering. Annual Review of chemical and biomolecular engineering, v. 8, p. 63-85, 2017.

DURRANI, M. A.; AHMAD, I.; KANO, M.; HASEBE, S. An Artificial Intelligence Method for Energy Efficient Operation of Crude Distillation Units under Uncertain Feed Composition. Energies, v. 11, n. 11, p. 2993, 2018.

FRANZOI, R. E.; KELLY, J. D.; MENEZES, B. C.; GUT, J. W. Advanced Data Analytics for Process-Shop Base+Delta Sub-Model Estimation in Planning and Scheduling Decision-Making. In: AICHE Annual Meeting, Pittsburgh, PA, United States. 2018.

FU, G.; MAHALEC, V. Comparison of Methods for Computing Crude Distillation Product Properties in Production Planning and Scheduling. Industrial & Engineering Chemistry Research, v. 54, n. 45, p. 11371-11382, 2015.

FU, G.; SANCHEZ, Y.; MAHALEC, V. Hybrid model for optimization of crude oil distillation units. AIChE Journal, v. 62, n. 4, p. 1065-1078, 2015.

GUERRA, O. J.; LE ROUX, G. A. C. Improvements in petroleum refinery planning: 1. Formulation of process models. Industrial & Engineering Chemistry Research, v. 50, n. 23, p. 13403-13418, 2011.

JOLY, M.; ODLOAK, D.; MIYAKE, M.; MENEZES, B. C.; KELLY, J. D. Refinery production scheduling toward Industry 4.0. Frontiers of Engineering Management, v. 5, n. 2, p. 202-213, 2018.

KELLY, J. D. The Unit-Operation-Stock Superstructure (UOSS) and the Quantity-Logic-Quality Paradigm (QLQP) for Production Scheduling in The Process Industries. In Multidisciplinary International Scheduling Conference Proceedings: New York, United States, p. 327-333, 2005.

KELLY, J. D.; MENEZES, B. C.; GROSSMANN, I. E. Distillation blending and cutpoint temperature optimization using monotonic interpolation. Industrial & Engineering Chemistry Research, v. 53, n. 39, p. 15146-15156, 2014.

KELLY J. D.; ZYNGIER D. Continuously improve the performance of planning and scheduling models with parameter feedback. In: Proceedings of the foundations of computer-aided process operations, 2008.

LI, J.; XIAO, X.; BOUKOUVALA, F.; FLOUDAS, C. A.; ZHAO, B.; DU, G.; SU, X.; LIU, H. Data-driven mathematical modeling and global optimization framework for entire petrochemical planning operations. AIChE Journal, v. 62, n. 9, p. 3020-3040, 2016.

LI, W.; HUI, C.; LI, A. Integrating CDU, FCC and product blending models into refinery planning. Computers & chemical engineering, v. 29, n. 9, p. 2010-2028, 2005.

MAHALEC, V.; SANCHEZ, Y. Inferential monitoring and optimization of crude separation units via hybrid models. Computers & Chemical Engineering, v. 45, p. 15-26, 2012.

MAKTOUBIAN, J.; GHASEMPOUR-MOUZIRAJI, M.; NOORI, M. Oil and Gas supply chain optimization using Agent-based modelling (ABM) integration with Big Data technology. EAI Endorsed Transactions on Smart Cities, v. 4, n. 9, 2020.

MCBRIDE, K., SANCHEZ MEDINA, E. I.; SUNDMACHER, K. Hybrid Semi-parametric Modeling in Separation Processes: A Review. Chemie Ingenieur Technik, 2020.

MENEZES, B. C.; KELLY, J. D.; GROSSMANN, I. E. Improved swing-cut modeling for planning and scheduling of oil-refinery distillation units. Industrial & Engineering Chemistry Research, v. 52, n. 51, p. 18324-18333, 2013.

RIAZI, M. R. Characterization and properties of petroleum fractions. ASTM international, 2005.

THIELE, E. W.; GEDDES, R. L. Computation of distillation apparatus for hydrocarbon mixtures. Industrial & Engineering Chemistry, v. 25, n. 3, p. 289-295, 1933.

WILSON, Z. T.; SAHINIDIS, N. V. The ALAMO approach to machine learning. Computers & Chemical Engineering, v. 106, p. 785-795, 2017.

WU, X.; ZHU, X.; WU, G.; DING, W. Data mining with big data. IEEE transactions on knowledge and data engineering, v. 26, n. 1, p. 97-107, 2013.

YU, W.; MORALES, A. Data driven fast real-time optimization with application to crude oil blending. In 2019 1st International Conference on Industrial Artificial Intelligence (IAI), p. 1-6, 2019.

ZHANG, J.; ZHU, X. X.; TOWLER, G. P. A level-by-level debottlenecking approach in refinery operation. Industrial & engineering chemistry research, v. 40, n. 6, p. 1528-1540, 2001.

7

Large-Scale Online Refinery Scheduling Optimization with Decompositions, Heuristics, and Surrogate Approximations

Optimization of large-scale discrete-time scheduling problems is challenging due to the combinatorial complexity of binary or discrete decisions being made. When including networks of unit-operations and inventory-tanks to fulfill both the logistics and quality balances as found in complex-scope process industries, the decomposition of the quantity-logic-quality phenomena (QLQP) paradigm in mixed-integer linear programming (MILP) and nonlinear programming (NLP) has been often used to find solutions of industrial-sized problems in reasonable computing processing times. Furthermore, as these industries manage with procurement of resources and production of products, solving a scheduling-type of problem within a time-horizon of a planning solution potentially increases the economic value of the process through market opportunities and more efficient operations. However, there are open challenges to automatically solve complex large-scale discrete-time problems in acceptable computing times using time steps within the shift of the operators (e.g., 8 hours) or even in smaller windows such as 1, 2, or 4 hours. In this direction, this chapter discusses strategies for handling and solving large-scale scheduling problems of process industry networks considering the unit-operation-port-state superstructure (UOPSS) constructs and the semantics of the QLQP concepts in a discrete-time formulation. That includes the use of phenomenological decompositions within a two-step solving procedure of mixed-integer nonlinear programming (MINLP) models, in addition to heuristic procedures, including exclusions to reduce the scale of the optimization search space in constructive rolling horizon strategies, relaxations to employ relax-and-fix iterations for a large-scale MILP formulation, linear reformulations to approximate nonlinear blending equations, and surrogate models embedded in refinery scheduling environments within a scheduling formulation towards online applications.

7.1 Introduction

The optimization of large-scale industrial scheduling problems is challenging because of their complexity and size. To build the mathematical formulation for these problems, three types of variables and constraints are used, namely, 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.). Process industry scheduling networks manage the integration of these three types of operations by considering the simultaneous optimization of the quantity, logistic, and quality (qualogistics) problems. Moreover, industrial problems typically involve large-scale models with dozens or hundreds of thousands of variables and constraints, including non-convex terms. Thus, a large-scale mixed-integer nonlinear programming (MINLP) model arises, imposing difficulties for the solution of complex industrial cases primarily due to the discrete and nonlinear combination of logic and quality phenomena. In this full space approach, due to the relaxation of binary variables in the first stage of the algorithm, the optimization of nonlinear problems is initially highly degenerate (many undifferentiated solutions) and strongly dependent on good initial points to converge. Even pure mixed-integer linear programming (MILP) models using linear programming (LP) approximations or reformulations of processing transformations and blending of streams and inventories are by themselves challenging problems to be solved.

Specific-domain modeling and heuristic approaches suitable for operations research (OR) and process system engineering (PSE) problems can handle the complexities of process industry scheduling via ad hoc representation of the models, whereby both the configuration and solution procedures can be constructed iteratively for large-scale and complex-scope applications. Even though the advances in computational processing and commercial solvers (e.g., CPLEX, GUROBI, XPRESS) have allowed automated decision-making of large-scale scheduling problems with much lower computational effort (HARJUNKOSKI et al., 2014), academia and industry have mostly addressed scheduling optimization within continuous-time approaches to avoid NP-hard discrete-time formulations. However, to coordinate operational activities for future prescriptions of decisions and executions in the production field, most tasks and procedures are still carried out manually by operators within discrete-time windows.

Aiming to develop large-scale discrete-time formulations and simultaneously overcome these overly complex modeling and solving difficulties for decision-making applications, Kelly et al. (2017b) effectively solved an industrial-sized refinery scheduling formulation by employing the unit-operation-port-state superstructure (UOPSS) (KELLY, 2005) constructs and the quantity-logic-quality phenomena (QLQP) concepts, in addition to the phenomenological decomposition heuristic (MENEZES et al., 2015) and the feedstock assignment strategy (Kelly et al., 2017a). The problem was formulated for the future 7-day time horizon discretized in 2-hour intervals (in a total of 84 time steps), with dozens of thousands of variables, constraints, and degrees of freedom. This breakthrough on discrete-time refinery scheduling optimization was achieved due to the recent research and technological advancements in the modeling and optimization approaches, computing processing power, and solution algorithms, and indicates that there are opportunities for further improvements on the topic, especially towards industrial applications.

However, despite the recent improvements in the decision-making modeling and solving capabilities for the process industry, there are still open challenges in industries with diverse quality raw materials and products, such as in the crude oil refining, manufacturing of petrochemicals, and liquefied natural gas production and processing. In such industries, the transformation of raw materials into products is typically performed within integrated solutions of the following networks: a) feedstock procurement, shipping, unloading, storage, dieting and charging, b) combined operations of job-shops, process-shops and blend-shops, c) management of intermediate and final product inventories, d) marketing, sales and distribution of the final products. Thus, a proper formulation should include the supply, production, and demand chains (or value chain) in a decision-making framework considering multiple scope as well as multiple scale with discrete-time steps of days or hours (MENEZES, GROSSMANN, and KELLY, 2017). In value chains from raw materials to products, the planning and scheduling integration from the midstream to the downstream operations can be performed by optimizing a scheduling problem within the operational planning time-horizon (e.g., weeks to months). From a product standpoint, such a complex planning-scheduling solution is considered as spotting, because of the potential profit increase by selling in spot markets rather than solely through contract markets. For the producer, allowable product delivery service levels offered by the spotting solution is achieved by considering the time-horizon of a month with prescriptions to be executed in a scheduling level of detail. By solving such complex planning horizon with scheduling complexities, the commercialization teams can sell products in spot markets since this requires a quarter to a whole month in advance for time-sensitive contracts. From planning and scheduling perspectives to the spotting level of service, the benefits of using such optimization-based decision-making for sizing, selecting, sequencing, slotting, and spotting are as follows (MENEZES and KELLY, 2019):

a) Optimized (instead of simulated) schedules where profit and performance are maximized within time steps of hours;

b) Enhanced marketing coordination by the optimized and achievable blend production schedules for all future product (and saleable component) shipments;

c) Improved stewardship or management of the feedstock determination with the supply and demand of the processing site production;

d) Increased capability to explore spot market opportunities after the fulfilment of the contracted market;

e) Improved matching of product quality specifications to avoid property giveaway (over-specification) and to reduce off-specification products.

Several recent works in the literature address modeling and solving strategies to handle complex industrial problems. Nevertheless, there are still challenges to solve such enterprise-wide optimization problems in a proper and efficient fashion, aiming to provide a formulation that is both computationally tractable and coherent with the real process or application. In this direction, large-scale industrial scheduling applications can be tackled by simultaneously addressing the UOPSS representation within a discrete-time formulation, in addition to decompositions and heuristic strategies to overcome the current solving barriers that limit an efficient optimization of large-scale problems with dozens of thousands of variables and constraints, including: a) decomposition strategies to enable the optimization of large scale problems; b) linearization procedures to include nonlinear quality information for blending of streams in an mixed-integer linear programming model; c) improved processing and blending design to provide a proper mathematical representation of the problem; d) exclusion heuristics using rolling horizon strategies; e) relaxation heuristics considering relax-

and-fix iterations to construct the complete MILP problem by the integration of the solution of the subproblems; f) surrogate correlations to provide simple yet accurate approximations of complex formulas; and g) online scheduling concepts to mitigate plant-model mismatches.

7.2 Modeling and Optimization Strategies for Crude Oil Refinery Scheduling Applications: Decompositions, Heuristics, and Surrogate Approximations

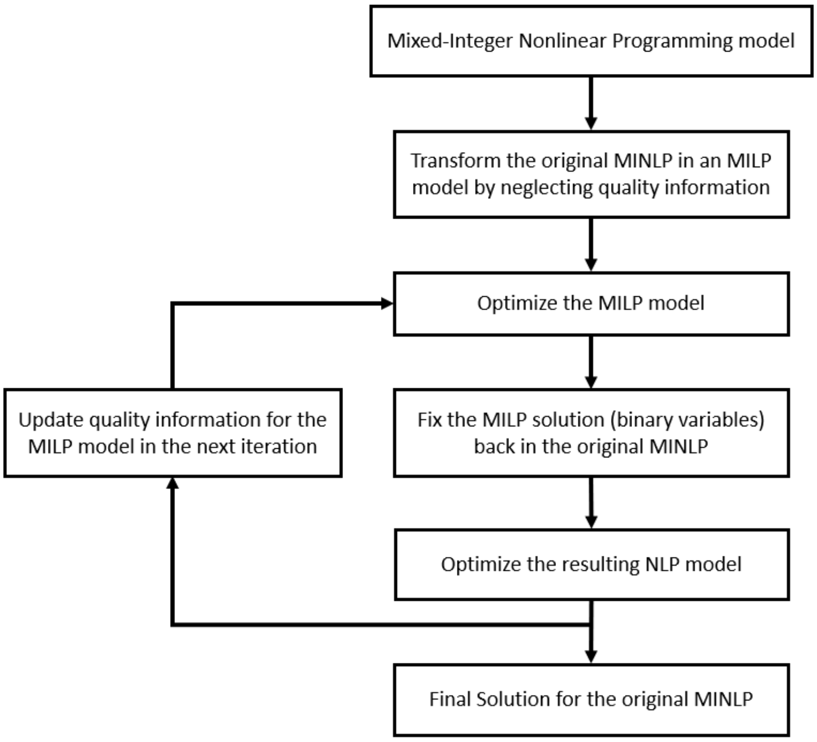
Heuristic approaches and decomposition algorithms have been widely used to handle intractable problems. For industrial-sized scheduling production of complex phenomenological (separating, converting, blending) and procedural (sequences, setups, startups) optimizations, the mathematical formulations may be difficult to be solved as full space MINLP problems. In order to properly determine the optimal schedule operations, it is required an accurate modeling including the logistics and quality information throughout the process, in addition to utilize efficient problem solving techniques via software optimization, which typically uses commercial solvers (e.g., CPLEX). However, there are modeling and optimization limitations to handle large-scale MINLP formulations, including applications on crude oil refinery scheduling.

Therefore, several works in the literature have been developed strategies to break down the MINLP formulations into smaller, simples, and less complex subproblems (MOURET, GROSSMANN, and PESTIAUX, 2009; LOTERO et al., 2016; ASSIS et al., 2019) to handle such complicated models that vary in a three-dimensional MINLP quantity-logic-quality (QLQ) relationship space. Typical decompositions lead to MILP-NLP formulations to be sequentially solved until a convergence criterion is met (WENKAI et al., 2002; MOURET et al., 2009; CASTRO and GROSSMANN, 2014; CAFARO et al., 2015; KELLY et al., 2017a).

7.2.1 Phenomenological Decomposition Heuristic

Menezes et al. (2015b) propose a phenomenological decomposition heuristic (PDH) approach for a strategic planning problem to determine the refinery configuration and the process unit dimensions. The problem is formulated as an MINLP, which is decomposed into an MILP and NLP sub-models. First, the MILP model is optimized considering only logistic and quantity information (i.e., by neglecting the quality information in the MINLP model). Then, the binary variables from the optimal MILP 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 the new coefficients or parameters in the MILP problem in the next iteration, with logic and quantity variables under consideration. Kelly et al. (2017b) applied the same method for scheduling applications, in which multiple MILP solution are generated for improve performance and convergence of the algorithm. An iterative procedure is performed, in which the best NLP solution (i.e., yields and quality information) is retro-fed to the MILP in the next iteration, until a convergence criterion is met. Figure 7.1 illustrates the phenomenological decomposition heuristic procedure.

Figure 7.1: Phenomenological decomposition heuristic.

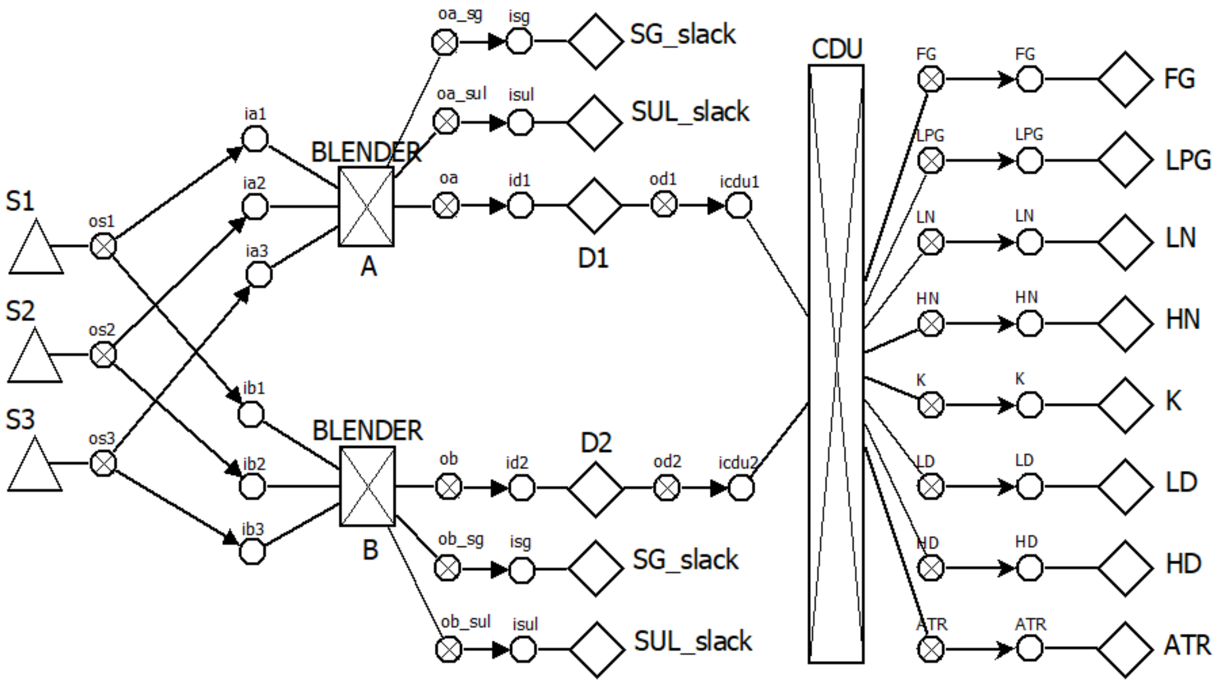

Mixed-Integer Nonlinear Programming model
Transform the original MINLP in an MILP model by neglecting quality information
Optimize the MILP model
Update quality information for the MILP model in the next iteration
Fix the MILP solution (binary variables) back in the original MINLP
Optimize the resulting NLP model
Final Solution for the original MINLP

Source: Author (2021).

7.2.2 Linear Approximation of Blending Equations: MILP Factor Blending

Aiming to reduce the MILP-NLP decomposition gap from the phenomenological decomposition heuristic, Kelly et al. (2018) developed a linear programming (LP) reformulation for nonlinear blending equations to approximate non-convex quality constraints in linear formulas valid exclusively for a mixing point of streams, such as a blender unit. Figure 7.2 illustrates a blend scheduling problem network in which the LP reformulation is incorporated to the blender unit.

Figure 7.2: Blend scheduling problem network including the factors approximation.



Source: Author (2021).

This strategy considers property variables as invariant coefficients of qualities (referred to as factors) in quality material balances of factor-flows that matches the product specification by including slack or surplus variables in an equality constraint. The extended quality amount of the property p is considered in the constraint as an in-out quantity and quality product or factors f multiplied by volume flows x around the blender unit-operations. To enforce the quantity-quality balance in the blender, the factor-flows $x_{j^F,p,t}$ outgoing from the slack or surplus out-port-states j^F $(\mathrm{oa\_sg}, \mathrm{oa\_sul})$ in Figure 7.2 are considered within the factor-flow balance in Equation (7.1).

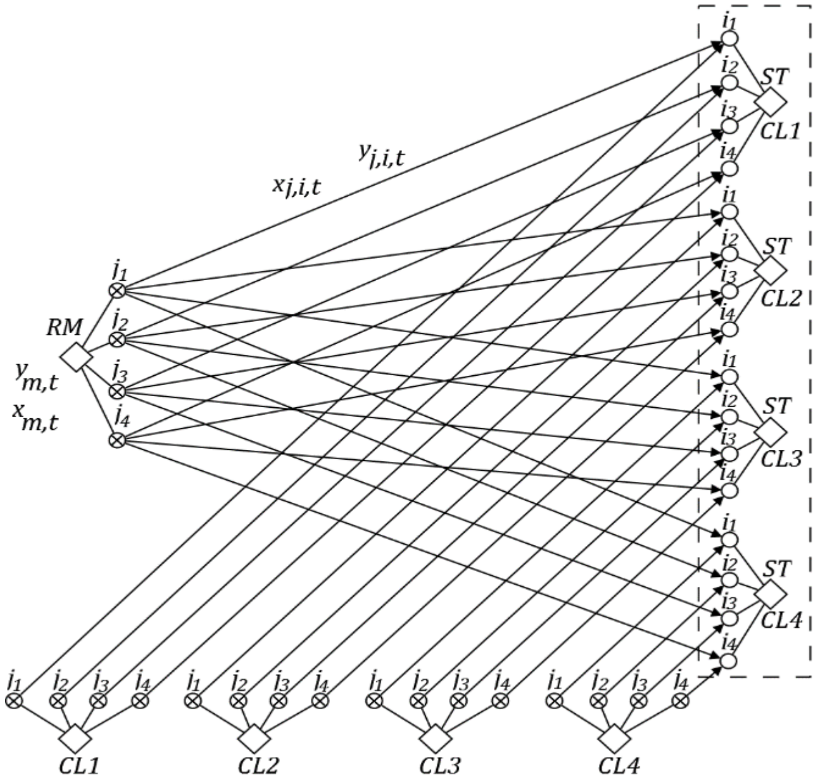
$$\sum_{i} f_{i,p,t} \sum_{j_{up}} x_{j_{up},i,t} = f_{j,p,t} \sum_{i_{do}} x_{j,i_{do},t} + x_{j^F,p,t} \quad \forall\, j, j^F, p, t \qquad (7.1)$$

In this LP approximation included in the MILP formulation, for each property p considered to be calculated in the blender $m \in M_{BL}$, amounts of raw materials $\sum_{j_{up}} x_{j_{up},i,t}$ incoming to multiple in-port-states i with factors for qualities $f_{i,p,t}$, in the left side of Equation (7.1) counterbalance the total amount $\sum_{i_{do}} x_{j,i_{do},t}$ of the blended material factor or property specification $f_{j,p,t}$ added to slacks or surpluses of the factor-flow variables $x_{j^F,p,t}$, in the right side of Equation (7.1). From the summation $\sum_{i_{do}} x_{j,i_{do},t}$, although there are multiple outlet ports considered for the blender, operationally there is only one outlet stream. The value of the slack or surplus factor-flows $x_{j^F,p,t}$ represents the insufficient or exceeded amount of qualities for the LP factor flow of each respective property p. The factor-flow variable $x_{j^F,p,t}$ closes the balance in Equation (7.1) considering the blended material amounts and the product factor $f_{j,p,t}$. For an upper bound of property specification, a slack or negative value is needed, so that $x_{j^F,p,t} \leq 0$. Similarly, for a lower bound, a positive factor-flow or surplus ($x_{j^F,p,t} \geq 0$) applies. Also, as transformation from property to property index may change the signal of the number, to avoid infeasibilities, the factor-flow is modeled as $x_{j^F,p,t} \leq 0$ and $x_{j^F,p,t} \geq 0$ for property indices.

7.2.3 Feedstock Storage Assignment

Kelly, Menezes, and Grossmann (2017a) develop a mixed-integer linear programming model to design pre-assignments of distinct feedstocks with different qualities when moving them from supply sources to shared storages. Their method assigns individual units or sources (e.g., crude oils, tanks, feedstocks) to a limited number of storage or sinks, and specifies the variables to be clustered (i.e., compound-properties). This clustering strategy is similar to the concept found in many sequence-dependent changeover heuristics (e.g., using product-wheels and blocking), in which the individuals are grouped in families given some common criteria. This is especially helpful when the storage is limited as well as for large-scale applications typically found in the crude oil, metal, and food processing industries. The storage assignment formulation minimizes the quality deviation in the clustering of a larger number of feedstocks into a smaller number of pools. This method designs simple and straightforward segregation rules in order to achieve better crude oil management, crude oil blend control, and improved blend scheduling optimizations. Figure 7.3 illustrates the feedstock storage assignment method, in which the raw materials RM are assigned to the storage sinks ST under distinct clustering groups CL.

Figure 7.3: Feedstock storage assignment flowsheet.

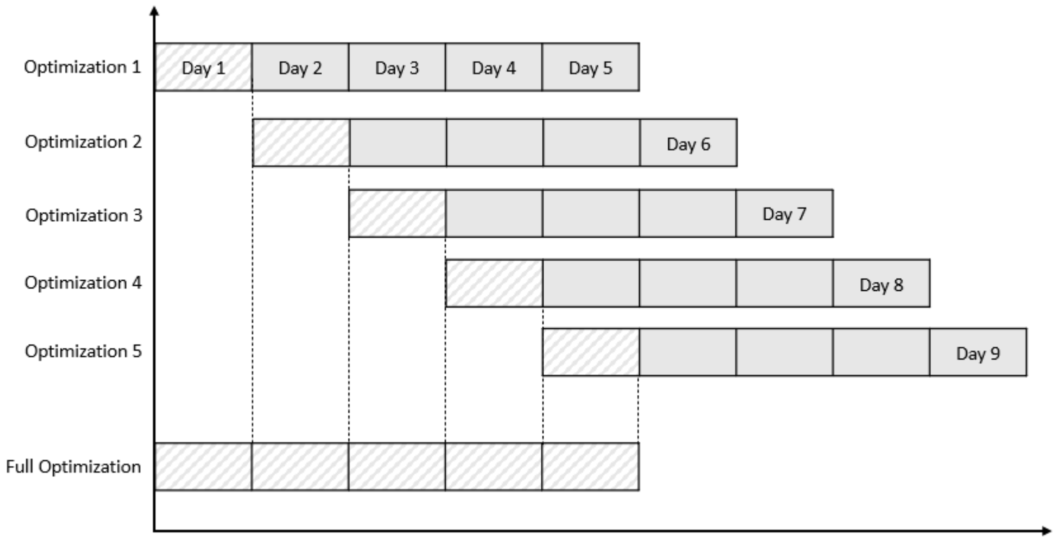


Source: Kelly et al. (2017a).

7.2.4 Chronological Decomposition Heuristic

Kelly (2002) developed the chronological decomposition heuristic (CDH), a straightforward time-based strategy used to find integer-feasible solutions. The CDH is specifically designed for discrete-time production scheduling optimization problems typically found in the petrochemical, chemical, and pharmaceutical industries. This heuristic decomposes the model regarding its time dimension (i.e., the full time horizon is discretized or decomposed into smaller steps). Each sub-model is then solved using mixed-integer linear programming (MILP) techniques starting from the first model onwards. Thus, instead of optimizing one large problem over the entire time horizon, multiple sequential time-discretized models are solved. Figure 7.4 illustrates the concept of the chronological decomposition heuristic over a rolling horizon.

Figure 7.4: Rolling horizon chronological decomposition strategy.

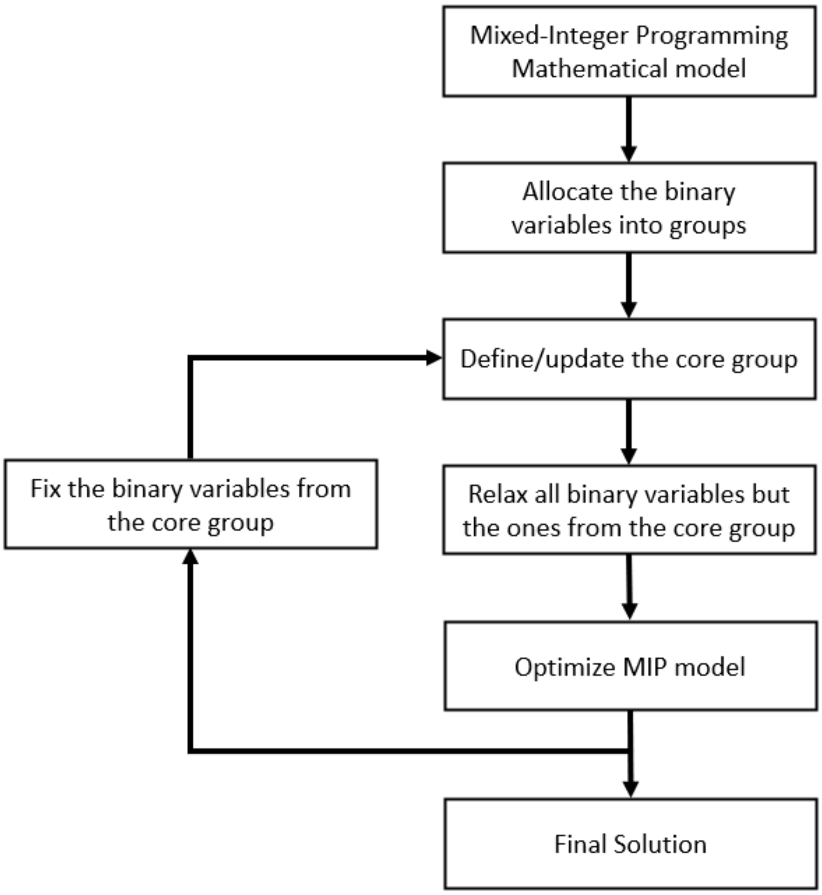


Source: Author (2021).

7.2.5 Relax-and-fix Decomposition

Aiming to tackle large-scale MILP problems, Kelly and Mann (2004) developed the flowsheet decomposition heuristic (FDH) to reduce the computational time needed to find good integer-feasible solutions for industrial applications. This method is a sort of relax-and-fix heuristic, which allocates individual units or sources (e.g., feedstocks, tanks) into groups, and solves mixed-integer linear programming problems (as many as there are groups), given a pre-specified order of importance. In each MILP subproblem, a group of units is defined as the core (most important) group according to the pre-definition. All the binary variables but the ones in this group are relaxed, and the MILP is optimized. The core group binaries are fixed to their optimal values (from the optimal solution), and the algorithm moves to the next iteration until the convergence is met (after optimizing as many MILPs as there are groups). At this point, all binaries will have been fixed, so that the solution for the last MILP optimization is also a feasible solution of the original MILP. Figure 7.5 illustrates the flowsheet decomposition heuristic relax-and-fix approach.

Figure 7.5: Relax-and-fix decomposition approach.


Mixed-Integer Programming Mathematical model
Allocate the binary variables into groups
Define/update the core group
Fix the binary variables from the core group
Relax all binary variables but the ones from the core group
Optimize MIP model
Final Solution

Source: Author (2021).

The phenomenological decomposition heuristic, the feedstock assignment strategy, and the Unit-Operation-Port-State Superstructure (UOPSS) (KELLY, 2005) were simultaneously applied for the modeling and optimization of an industrial size problem for the future 7-day time horizon discretized in 2-hour intervals. The formulation includes 5 atmospheric distillation units in 9 operational modes and 35 storage and feed tanks. The logistics problem (MILP) had around 30 thousand continuous variables and 30 thousand binary variables, 6.5 thousand equations, 80 thousand inequalities, and 53802 degrees of freedom, and was solved in 128.8 seconds using 8 threads in the solver CPLEX 12.6 (International Business Machine IBM, USA). The quality problem (NLP) had over 102 thousand continuous variables, 58 equality constraints, 768 inequality constraints, and 44520 degrees of freedom, and was solved in 10.3 minutes in a sequential linear programming (SLP) approach using the solver CPLEX 12.6. The MILP-NLP gap between the two solutions was below 3.5% after two iterations of the algorithm.

7.2.6 Surrogate Modeling for Refinery Unit-Operations

For high-performance operations in crude oil refinery processing, it is important to properly estimate the yields and properties of the output streams from unit-operations. However, effectively and accurately modeling the refinery processing units involve complex formulations and is typically assisted by rigorous simulation tools. Although these tools provide highly accurate solutions, they may not be suitable for large scale problems and for optimization applications. Looking for alternatives to complex formulations that often lead to convergence issues and to time consuming solutions, surrogate modeling approaches have been increasingly developed.

One of the most important processing units in the crude oil refinery network due to its economic and operational impact is the crude distillation unit (CDU), which is typically compound by a complex set of towers designed, operated, and controlled to separate liquid hydrocarbon feedstocks into intermediate distillates according to boiling range temperatures (RIAZI, 2005). The CDU operations are controlled by rigorous simulation tools, which are time consuming and not suitable for environments such as refinery planning and scheduling. Therefore, when solving planning and scheduling problems, the CDU calculations are typically estimated by using simplified models or correlations (e.g., fixed yield, swing-cuts). Aiming to improve these predictions, the use of surrogate formulations built based on data from either the real plant or generated from rigorous simulation presents several benefits. Franzoi et al. (2020) proposed a cutpoint temperature modeling framework to model the distillation unit through a coefficient setup MIQP (mixed-integer quadratic programming) technique to determine optimizable surrogate models to correlate independent X variables (crude oil compositions, temperatures) to dependent Y variables (yields and properties of distillates). These optimizable surrogate models are suitable to handle continuous data from the process with measurement feedback for adjustments and improvements, and the distillation outputs can be continuously updated in an online fashion by using updated data from the process.

Other refinery unit-operations (e.g., naphtha reformer, fluid catalytic cracker, delayed coker) can also be properly and effectively modeled by surrogate formulations, although specific methodologies might have to be developed for each type of unit, given that they are highly complex and have their own specificities regarding which are the most important output variables to be predicted, which are the most impactful variables that affect the predictions, what is the availability and accuracy of the data to be used to build or train the surrogates, etc.

7.2.7 Online Applications within the Crude Oil Refinery Scheduling

Designing the crude oil refinery scheduling for real applications includes building the mathematical formulation (modeling), solving the formulation to find feasible solutions (simulation or optimization), and implementing the solution found in the plant operations (implementation). As the schedule is implemented in the real process, uncertainties and disturbances are likely to arise, and new information becomes available and should be used as soon as possible for improved operations (GUPTA et al., 2016a), especially because changes as disruptions, delays, and market fluctuations, may result in sub-optimality and infeasibilities for the incumbent schedule (GUPTA et al., 2016b). The operational data used in the crude oil scheduling are typically out of date or not integrated with the production, which leads to inconsistencies in the prediction throughout the process (MENEZES et al., 2015c). Thus, a continuous cycle of improvement is required to reduce the deviation between the model predictions and the actual values in the plant, in which an efficient online scheduling (rescheduling) procedure facilitates the adaptation of the schedule under uncertainties and unforeseen events, in addition to considering new information as soon as possible, which aims to enhance the economic value of the refinery operations (GUPTA et al., 2016a).

This cycle of improvement should synchronize the process operations and the mathematical formulation (i.e., modeling and optimization). On one hand, feedback from the plant should be directly embedded in the modeling capabilities for new rescheduling cycles. On the other hand, continuous rescheduling should be performed in a systematic fashion to maintain the state of the system up-to-date and to handle changes and disturbances in the process. Towards online applications for refinery scheduling problems, efficient modeling and solving capabilities are required to provide accurate high-quality scheduling solutions. In this context, it is essential to properly handle complex formulations within online applications, in which fast and accurate scheduling solutions are obtained and embedded in a systematic rescheduling implementation strategy.

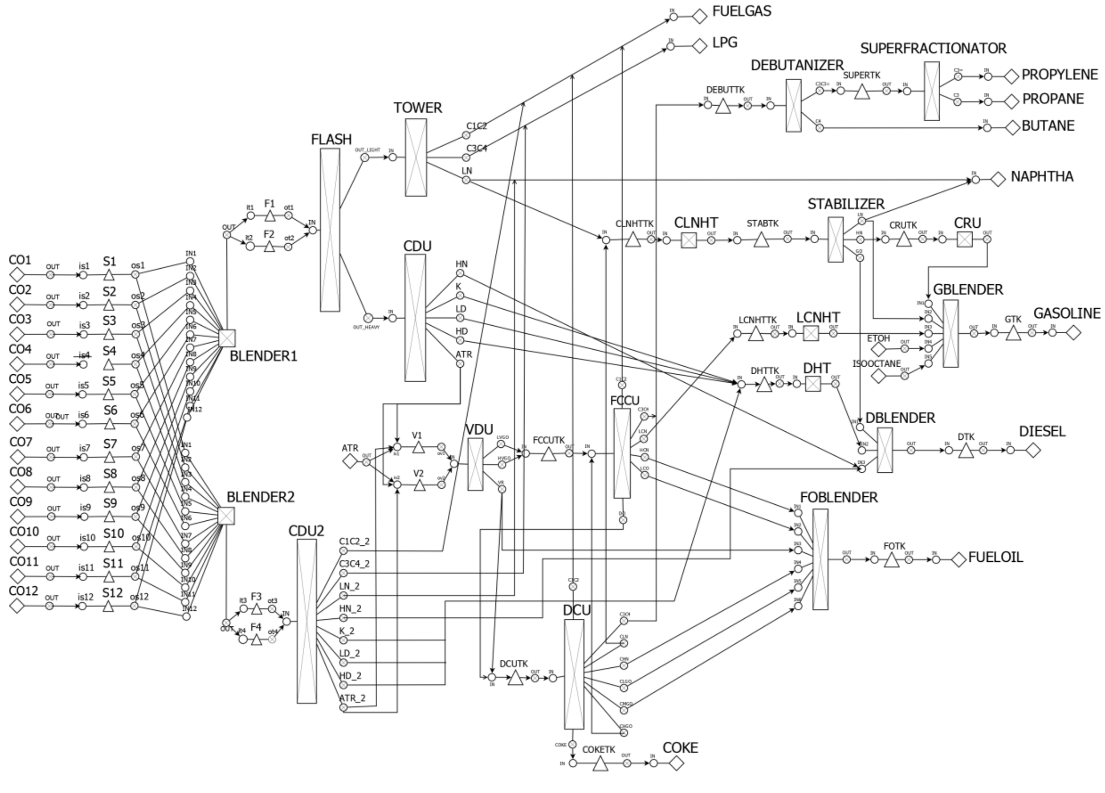
7.2.8 Future Outlook on Crude Oil Refinery Scheduling Applications

Industrial scheduling operations are typically based on simulation of discrete production scenarios for the sequence, selection, and setups of unit-operations and tanks considering a complex continuous-process network within a time-horizon of days, weeks, or months. In the past decades, several works in the literature have addressed continuous-time modeling for the optimization of scheduling decision-making. However, the operational activities in the production field are by nature discretized in time windows or shifts. Therefore, to coordinate the schedules, which are typically carried out by human beings, practitioners in the industry have to adjust or adapt the continuous solution to the real discrete problem in the plant. However, discrete-time formulations still pose difficulties due to their complexity especially for large-scale applications. These open challenges include how to automatically solve complex large-scale discrete-time problems in acceptable computing time using time-steps within the shift of the operators (8 hours) or even in smaller windows such as 1, 2, or 4 hours. In this direction, this work addresses strategies to handle and solve large-scale scheduling problems considering the unit-operation-port-state superstructure (UOPSS) constructs and the semantics of the QLQP concepts in a discrete-time formulation. The examples highlight decompositions as phenomenological heuristics, which consist in a two-step solving procedure of mixed-integer nonlinear programming (MINLP) models; as factorizing, to approximate nonlinear blending terms in a linear programming (LP) model, in which the factor-flows of qualities are modeled explicitly as slack or surplus variables; as exclusions, to reduce the scale of the optimization search space in constructive rolling horizon strategies; and as relaxations, when MILP programs construct a full problem by relax-and-fix iterations. Moreover, surrogate models are built and included in the refinery scheduling formulation, and scheduling optimizations towards online applications are carried out.

7.3 Example and Discussion

To illustrate some of the decomposition methods and heuristic strategies previously introduced, we present a large-scale crude oil refinery scheduling problem that considers the blending of twelve crude oil feedstocks, eight storage tanks, two crude oil blenders, the cascaded distillation unit with five towers (two atmospheric distillation units, a vacuum distillation unit, a pre-flash tower and a debutanizer), and the whole process-shops, which includes additional blenders, catalytic cracker, hydrotreaters, delayed coker, debutanizers, superfractionator, and reformer. The distillation unit is modeled as a series of towers in cascade, and surrogates are embedded in the formulation to replace the model for the two atmospheric distillation units. Figure 7.6 presents the flowsheet of the full crude oil blend scheduling problem.

Figure 7.6: Crude oil refinery scheduling flowsheet.



Source: Author (2021).

The PDH strategy is employed to break down the MINLP formulation in a sequential MILP-NLP optimization procedure. The chronological decomposition heuristic splits the time horizon in time chunks and optimize the problem in successive integrated subproblems with or without crossover between time windows, i.e., with recalculation of the time periods in the neighborhood. The UOPSS formulation is used for building the mathematical model, and the example is built within the modeling and solving platform IMPL. GUROBI 12.7.1 is used as optimization solver in an Intel Core i7 with 2.7 GHz and 16 GB RAM.

A total of 9 scenarios are proposed to formulate the crude oil refinery scheduling problem presented in Figure 7.6, in which distinct model sizes (in terms of time horizon length and time step) are tested. The chronological decomposition heuristic is used in some scenarios (b, c, d), in which the time horizon is decomposed in time segments of 1, 3, or 5 days. The proposed scenarios are presented in Table 7.1.

The mathematical model for each scenario is formulated in discrete time. The largest model is from Scenario 7.9a, which is formulated for the future 30 days with time steps of 2 hours, in a total of 460 time periods. In the base case (not using rolling horizon), there are around 70,000 continuous and 50,000 binary variables, 180,000 constraints and 90,000 degrees of freedom.

The MINLP formulation is highly complex because of the multiple nonlinearities and nonconvexities form the processing units, and it is large in size because of the large number of variables and constraints required to represent such complex system. The large scenario proposed considers hundreds of thousands of variables and constraints in an MINLP formulation, which is successfully solved in reasonable computational time by employing the PDH strategy.

When increasing the time horizon length or reducing the time step, the mathematical model increases in size and becomes more time consuming to solve. However, this typically results in better solutions and should be employed aiming to improved operations. The increase in the computational time and profitability are noticed when comparing Scenarios 7.1a to 7.9a, Scenarios 7.1b to 7.9b, etc.

There are significant improvements when using the chronological decomposition heuristic, which can be noticed by comparing Scenarios 7.1a and 7.1b, 7.2a and 7.2b, etc. In Scenario 7.9b, the computational time is reduced from 36,000 to around 5,000 seconds with a gap of only 0.75% from the best solution. Promising results are also achieved when coupling the CDH with smaller time steps in the optimization. For example, Scenarios 7.5a and 7.9b have similar computational cost but the latter has an increase in the profit of 16.4%, which is possible by the utilization of the chronological decomposition heuristic.

Table 7.1: Scenarios proposed for the crude oil refinery scheduling problem.

Scenario	Time Horizon (days)	Time Step (hours)	CDH Heuristic	Horizon Decomposition	Profit (US$)	Gap (%)	CPU (s)
7.1a	10	24	No	-	100,079	-	23
7.1b	10	24	Yes	5 days (2x)	98,351	1.72	14
7.2a	15	24	No	-	130,969	-	551
7.2b	15	24	Yes	5 days (3x)	123,707	5.54	22
7.3a	20	24	No	-	157,786	-	1,346
7.3b	20	24	Yes	5 days (4x)	150,718	4.48	30
7.4a	25	24	No	-	188,103	-	2678
7.4b	25	24	Yes	5 days (5x)	175,698	6.60	41
7.5a	30	24	No	-	218,270	-	4,498
7.5b	30	24	Yes	5 days (6x)	206,143	5.56	58
7.6a	30	12	No	-	233,760	-	36,000+
7.6b	30	12	Yes	5 days (6x)	223,060	4.57	851
7.7a	30	8	No	-	244,615	-	36,000+
7.7b	30	8	Yes	5 days (6x)	239,152	2.23	1,806
7.7c	30	8	Yes	3 days (10x)	235,800	3.60	730
7.8a	30	4	No	-	252,754	-	36,000+
7.8b	30	4	Yes	5 days (6x)	245,002	3.07	3,265
7.8c	30	4	Yes	3 days (10x)	241,780	4.34	2,285
7.8d	30	4	Yes	1 day (30x)	225,488	10.79	254
7.9a	30	2	No	-	255,899	-	36,000+
7.9b	30	2	Yes	5 days (6x)	253,990	0.75	5,108
7.9c	30	2	Yes	3 days (10x)	252,568	1.30	3,734
7.9d	30	2	Yes	1 day (30x)	234,217	8.4	760

Source: Author (2021).

7.4 Conclusions

Decomposition and heuristic approaches for large-scale complex applications have been increasingly used due to the recent technological advances, which allows the modeling and optimization of problems that were previously intractable, or that require fast solution for time-limited applications. In this new reality, the open challenges become open opportunities for the development of strategies such as the ones addressed in this chapter.

To handle complex formulations such as the ones found in large-scale discrete-time scheduling, modeling methods, decomposition approaches, and heuristic strategies are often used to design, model, and solve large-scale problems. That includes: a) Mathematical modeling of large-scale scheduling problems using the UOPSS superstructure; b) Phenomenological decomposition of the quantity-logic-quality phenomena paradigm, which addresses a two-step solving procedure of MINLP formulations as MILP and NLP sub-models; c) Iterative procedure within the PDH approach to reduce the MILP-NLP gap; d) Exclusions employed to reduce the scale of the optimization search space in constructive rolling horizon strategies; e) Relaxations in MILP models to by using relax-and-fix iterations; f) Factorizing strategy to approximate nonlinear blending terms in a linear programming model, in which the factor-flows of qualities are modeled explicitly as slack or surplus variables.

The example shown in this chapter illustrates the application of decomposition and heuristic approaches and highlights their respective improvements in terms of the modeling, solving, and solution aspects. These methods are successfully applied to reduce the complexity of intractable formulations, and provide much faster solutions as well. From Table 7.1 it is shown that MINLP problems with large complexity and size can be efficiently solved by employing decomposition and heuristic strategies, and the need of properly addressing formulations with larger time horizons and smaller time steps towards improved solutions is illustrated as well.

Future work on the topic will focus on addressing multiple heuristic strategies within a systematic framework aiming to achieve improved solutions by solving formulations that are even more complex and that address additional processes and distinct applications.

7.5 References

HARJUNKOSKI I.; MARAVELIAS C. T.; BONGERS P.; CASTRO P. M.; ENGELL S.; GROSSMANN I. E.; HOOKER J.; MENDEZ C.; SAND G.; WASSICK J. Scope for industrial applications of production scheduling models and solution methods. Computers & Chemical Engineering, v. 62, p. 161-193, 2014.

KELLY, J. D. The unit-operation-stock superstructure (UOSS) and the quantity-logic-quality paradigm (QLQP) for production scheduling in the process industries. In Multidisciplinary International Scheduling Conference Proceedings: New York, United States, p. 327-333, 2005.

KELLY, J. D.; MENEZES, B. C.; ENGINEER, F.; GROSSMANN, I. E. Crude oil blend scheduling optimization of an industrial-sized refinery: a discrete-Time benchmark. In: Foundations of Computer Aided Process Operations, Tucson, United States, 2017b.

KELLY, J. D.; MENEZES, B. C.; GROSSMANN, I. E.; ENGINEER F. Feedstock storage assignment in process industry quality problems. Foundations of Computer Aided Process Operations: Tucson, United States, 2017a.

MENEZES, B. C.; GROSSMANN, I. E; KELLY, J. D. Enterprise-wide optimization for operations of crude-oil refineries: closing the procurement and scheduling gap, Computer Aided Chemical Engineering, v. 40, 1249-1254, 2017.

MENEZES, B. C.; KELLY, J. D. High-quality blend scheduling solution for sizing, selecting, sequencing, slotting and spotting in the processing industries. Computer Aided Chemical Engineering, v. 46, 1813-1818, 2019.

MENEZES, B. C.; KELLY, J. D.; GROSSMANN, I. E. Phenomenological decomposition heuristic for process design synthesis of oil-refinery Units. Computer Aided Chemical Engineering, v. 37, p. 1877-1882, 2015.

8

General Conclusions and Future Outlook

The crude oil refinery scheduling optimization is a complex and challenging problem due to the high number of continuous and binary variables, in addition to nonlinear and non-convex terms, which results in a non-convex large-scale MINLP formulation. Hence, three main strategies have been adopted in both industry and academia to handle this complex problem. First, modeling and solving crude oil refinery scheduling problems often considers a simplified formulation that does not include all the processing units, tanks, flows, and variables from the real industrial problem. Second, the refinery scheduling model is broken down into sub-problems to be hierarchically solved. Third, several industrial processes still use simulation-based instead of optimization-based approaches due to the intractability of such formulation.

However, we are moving towards the Industry 4.0 age, with more complex, detailed, accurate, and efficient tools and resources on process optimization. Solving challenging industrial problems in a near online fashion is becoming reality mainly due to the advancements in decision-making modeling, computer-aided resources, and solution algorithms. Improved modeling, solving, and implementation approaches, as well as machine learning and big data strategies, increasingly lead to opportunities for cost reduction, increased economic value of the process, and improved operations. Commercial optimization solvers have become increasingly robust and efficient, in addition to the enhancements of computational processing, which have reduced the computational time and effort (for both simulation and optimization approaches) in over two orders of magnitude in the last decades. Many problem solving and decision-making strategies have been developed and improved, with an emphasis on decomposition and heuristic approaches for large-scale industrial applications, which allows the modeling and optimization of previously intractable problems, and provides resources for novel real-time industrial applications. In the Industry 4.0 age, the current challenges open opportunities for the development of novel and improved modeling and optimization strategies.

Aiming to study and contribute to the state-of-the-art on the chemical engineering optimization literature, the research topics addressed herein are focused on handling complex formulations typically found in the chemical engineering industry through the implementation and development of modeling and optimization approaches, decomposition and heuristic strategies, and machine learning techniques for chemical processes with an emphasis on crude oil refinery scheduling applications. This includes: a) Mathematical modeling of large-scale discrete-time crude oil refinery scheduling problems using the UOPSS representation; b) Phenomenological decomposition of the quantity-logic-quality phenomena, which addresses an iterative two-step solving procedure of MINLP formulations as sequential MILP and NLP sub-models; c) Efficient process design regarding both the mathematical formulation and the operations in the plant; d) Linearization strategies to approximate nonlinear blending terms in a linear programming model, in which the factor-flows of qualities are modeled explicitly as slack or surplus variables; e) Exclusions employed to reduce the scale of the optimization search space in constructive rolling horizon strategies; f) Relaxations in MILP models by using relax-and-fix iterations; g) Parameter feedback and rescheduling strategies aiming to minimize plant-model mismatches, to handle process uncertainties and disturbances, and to improve the reliability and accuracy of the scheduling solution and implementation; h) Surrogate modeling approaches as an alternative to complex or rigorous models that often lead to intractable or infeasible formulations, and that typically impose limitations for integrated optimization environments.

The crude oil refinery scheduling problems addressed are formulated using the unit-operation-port-state superstructure and the quantity-logic-quality phenomena (QLQP) concepts. That allows the modeling and solving of complex-scope industrial-sized scheduling problems using a discrete-time formulation. A phenomenological decomposition heuristic is applied to handle the complex MINLP refinery scheduling formulations by breaking down the problem to significantly reduce the computational burden within optimization approaches.

A proper design plays a key role on industrial operations, including the methodology used to build the mathematical formulation to be solved, as well as the process operations in the real plant. On one hand, the mathematical formulation is expected to accurately represent the real process, especially aiming to mitigate plant-model mismatches over time. On the other hand, the process design directly impacts the economic, technical, and operational conditions. For improved operations, especially when addressing online strategies toward smart processing with real-time feedback from the plant, both an improved blend design and a complex process design are the recommendable networks to be constructed, modeled, and solved, aiming to achieve improved solutions in terms of economic value and production flexibility. The blending formulation considering a continuous blender unit instead of batch mixtures (without the blender), and the processing formulation considering a complex cascaded distillation network instead of a simplified one-tower network, are examples of improved designs that provide more accurate predictions, production flexibility, and increased economic value for the process. This is especially beneficial when the formulation simultaneously includes the blending and the processing improved designs.

Modeling strategies such as the linearization of blending constraints within linear problems improves the mathematical formulation by considering additional information for more accurate predictions, which results in better optimized solutions. This strategy includes proxied information on the qualities of streams by using a linear programming factor reformulation for the blending operations of crude oil. The benefits include improved solutions, with better economic value, and better convergence within the optimization procedure. As the formulation remains unchanged, except for the blending constraints, the computational effort increase is expected not to be a limiting factor for the utilization of this method.

Modeling, solving, and heuristic strategies are employed for handling complex industrial-sized refinery scheduling problems within a discrete-time formulation, including decompositions to reduce the optimization search space in constructive rolling horizon strategies, and relaxations on mixed-integer linear programming problems to construct the problem by an ad-hoc relax-and-fix approach. Both heuristics imply in an expected slight reduction in the objective function, but with great benefits in terms of reduced computational effort. Moreover, this reduced effort is expected to scale with the size of the problem; therefore, it might be especially useful for large-scale applications.

For improved industrial operations, efficient mathematical formulation and optimization procedures are fundamental. That includes minimizing the plant-model mismatches so that the optimal solution is as coherent as possible with the actual process conditions and operations. Because of the high nonlinear and uncertain nature of most industrial problems, unforeseen events and uncertainties are likely to happen, motivating a continuous optimization cycle, in which the current state of the system should be updated, and a closed-loop rescheduling should be performed in a moving horizon approach. This type of closed-loop rescheduling-based approach can effectively handle disturbances (either to mitigate/reduce their effects or to take advantage of new information), reduce inaccuracies and plant-model mismatches by maintaining and updating the state of the system, and provide a systematic approach for improved scheduling implementation. Moreover, parameter updating may also be required to reduce the offsets or inaccuracies over the operations life-time as an effective way to reduce the plant-model mismatches.

Surrogate modeling has been increasingly used as an alternative to rigorous or complex models that often lead to time consuming solutions, difficulties within optimization environments, and infeasibility or convergence issues. Seeking alternatives to such complex formulations, especially found in industrial operations such as in the crude oil refinery scheduling, the use of surrogates is addressed to build relatively simple correlations that significantly reduce the computational burden within optimization applications while attempting to maintain a high degree of accuracy or fidelity required to provide high quality solutions. A surrogate model building methodology is developed to model the distillation unit, which is the most important processing unit in crude oil refinery operations. For the distillation unit surrogate modeling, the surrogates estimate the yields and properties of distillates using the crude oil assay and the hypothetical swing-cuts as input training variables. They are built through measurement feedback by using simulated data based on the swing-cut methods, although real (and uncertain) data from the plant or rigorous simulated data could be used for improved predictions. The results indicate that the surrogates built using the proposed methodology provide accurate predictions for the distillation process, and due to the small number of equations required, these shortcut sub-models can be properly integrated into any planning, scheduling, and coordinating environment with minimal increase in the simulation and optimization effort and data requirements. Hence, the surrogates are potentially suitable to replace complex or rigorous distillation models for several applications, including crude oil refinery planning and scheduling. It is worth noting that the proposed surrogate model building methodology is self-adjustable and self-improvable by using reliable and up-to-date measurements from the plant, so that the surrogates are expected to achieve a good and robust performance over time. Future work on the topic includes utilizing either real data from crude oil refineries or data simulated via a rigorous process simulator for improved predictions, and embedding the surrogates in crude oil refinery applications such as planning and scheduling.

The research developed herein contributes to the state-of-the-art on the crude oil refinery scheduling optimization literature by implementing and developing design strategies, decomposition and other heuristic approaches, and machine learning techniques, in a complete crude oil refinery scheduling optimization problem. The formulation proposed is coherent with large-scale industrial applications in terms of operational constraints, refinery economics, and problem complexity and size. The strategies and approaches addressed have been either applied to the refinery scheduling problem or could potentially be applied in future work. The results indicate that complex non-convex MINLP refinery scheduling formulations can be efficiently solved by utilizing decomposition, heuristic, and machine learning strategies, which would potentially provide improved modeling and optimization capabilities for real industrial applications.

The novelty of this research consists of modeling and optimizing a complete crude oil refinery scheduling problem, including the investigation of design features for blending and processing operations, decomposition approaches for handling intractable formulations, rescheduling strategies for mitigating plant-model mismatches and handling uncertainties, noises, and disturbances in the process, and surrogate modeling approaches to effectively replace complex formulations in order to allow the integration of unit-operation models within refinery scheduling environments.

Future work on the topic includes: a) modeling and optimization of a discrete-time refinery scheduling formulation for longer time horizon (e.g., 1-month), with smaller time steps (e.g., 1-hour), aiming to achieve improved solutions towards online scheduling applications; b) implementation of parameter feedback and rescheduling strategies in the crude oil refinery scheduling optimization for mitigating plant-model mismatches and handling uncertainties and disturbances in the process; c) development of surrogate-based models for other unit-operations in crude oil refineries and their integration to environments such as the refinery planning, scheduling, and real-time optimization.