An Approximate Method for the Optimization of Long-Horizon Tank Blending and Scheduling Operations

Benjamin Beach∗, Robert Hildebrand†, Kimberly Ellis† and Baptiste Lebreton‡

June 16, 2020

Abstract

We address a challenging tank blending and scheduling problem regarding operations for a chemical plant. We model the problem as a nonconvex MIQCP, then approximate this model as a MILP using a discretization-based approach. We combine a rolling horizon approach with the discretization of individual chemical property specifications to deal with long scheduling horizons, time-varying quality specifications, and multiple suppliers with discrete arrival times. This approach has been evaluated using industry-representative data sets from the specialty chemical industry. We demonstrate that the proposed approach supports fast planning cycles.

1 Introduction

For speciality chemical manufacturing, companies purchase and blend supply streams from a variety of suppliers. Because the supply streams are derivatives of refinery operations, the chemical composition (specifications) of the streams may vary significantly by supplier as well as by season. Supply streams may have both desirable properties and less desirable properties that prevent their exclusive use during the production process. Furthermore, some streams might have limited available quantities or intermittent availability. Thus, supply streams of differing composition are often combined into intermediate storage tanks. The resulting compounds are then blended to meet demand for production feed (see Figure 1).

For our motivating application, supply streams typically arrive in barges from suppliers and may remain at the inbound docks for several days before unloading into one or multiple storage tanks. The time windows for the arrival and departure of barges are committed several months in advance. As the supply barges are unloaded into the tanks, the volume and chemical composition of the compounds in the storage tanks are changed. The compounds from one or multiple storage tanks are then blended to create the desired

∗Grado Department of Industrial and Systems Engineering, Virginia Tech. Corresponding author, e-mail: bben6@vt.edu
†Grado Department of Industrial and Systems Engineering, Virginia Tech
‡Operations Research Team Leader, Eastman Chemical Company, Kingsport, TN, email: blebreton@eastman.com
quantity and specification of production feed. The production feed is planned in product campaigns (runs), typically spanning 3-14 days per product. Approximately 3 to 12 months of product campaigns are planned in advance due to the wide variety of spec requirements over time for very different non-concurrent production runs, the non-periodic nature of the supply acquisition process, and the limited capacity of the storage tanks.

Given the volume and characteristics of the compounds on the inbound barges, the time windows for the availability of barges, and the desired production feed (both quantity and specifications), the company must determine an assignment of supply barges to tanks, an unloading plan from barges to tanks, and a blending plan from tanks to production feed. Although similar problems are often inherently multi-objective, the most important objective by far is to satisfy production requirements and to unload supply. Hence, the objective is to determine an operational schedule for the given barge arrivals and production plan. Rapid solutions are required for long scheduling horizons to support effective planning decisions. Thus, we propose an efficient optimization-based approach to provide solutions for this assignment, mixing, and blending problem within ten minutes for a planning horizon of up to a year.

Within the context of supply chain management, this problem encompasses short-term decisions as highlighted in the Supply Chain Planning Matrix (Figure 2, adapted from [24]).
2 PROBLEM DESCRIPTION

In this work, we employ a discrete-time mixed-integer linear program (MILP) approximate formulation of the core nonconvex mixed-integer quadratically-constrained program (MIQCP), formulated via direct tracking of specs, using single-day scheduling periods with scheduling horizons of 90 to 368 days. The ‘binary’ version of this discretization combines aspects of multi-parametric disaggregation (MDT) with normalized MDT (NMDT), while avoiding the explicit tracking of spec qualities. The demand spec and spec ratio requirements are tightened to help ensure that the approximate representation of the specs does not lead to violations of the demand composition requirements. To satisfy the long-horizon scheduling goals within a reasonable computation time, we use a rolling-horizon solution approach. After optimization of the model approximation, the scheduling solution is simulated to obtain the correct tank inventory and demand feed specs.

In the following sections, we provide a formal problem description, highlight the related research, formulate a mixed integer non-linear representation of the problem, propose approaches for solving the model, and compare the approaches with computational studies based on industry-representative data sets.

2 Problem Description

For a given a set \( S \) of supply nodes (barges), assume each supply node \( s \) contains a given volume \( y_s \) of raw material, with a level for chemical property \( q \) of \( f_{s,q} \), where \( q \in Q \). The levels of these chemical properties are often specified as a percentage of the volume for a material component of interest. The percentages may not necessarily sum to 100% of the material, because some components may overlap and others may not be of interest. Each supply node \( s \) is available during a set \( T_s \) of consecutive time periods, from \( t_s^{\text{min}} \) to \( t_s^{\text{max}} \). With the availability of all supply nodes known, the set of supply nodes available each
time period \( t \) is denoted as \( S_t \). The number of barges that are unloaded each day is limited to \( ul_{\text{max}} \). If a barge is unloaded in a given day, the minimum percentage of inventory that can be unloaded is \( ulp_{\text{min}} \). If a supply barge has less than \( ulp_{\text{min}} \) remaining, then the inventory is not unloaded and the barge departs.

A set \( K \) of intermediate storage tanks is available to hold raw material or a blend of raw materials. Each tank \( k \) has an initial inventory volume \( v_{k,0} \), minimum inventory level \( v_{k,\text{min}} \), and maximum inventory level \( v_{k,\text{max}} \). The initial quality level for chemical property \( q \) is denoted by \( f_{k,q,0} \). Each tank can be supplied by the supply nodes in \( K_s \). If the mixture in a tank \( k \) is used to fulfill production feed, the minimum percentage of feed that can originate from the tank is \( tkp_{k,\text{min}} \). The demand for production feed is known for each time period \( t \in T \). This demand is characterized by a volume \( d_t \), bounds \( f_{\text{min}} \) and \( f_{\text{max}} \) on the quality level of the chemical properties, and bounds \( l_{t,q,1} \) and \( l_{t,q,2} \) on certain ratios of chemical properties. Production feed is planned in campaigns or runs of consecutive days. During a run, \( f_{\text{min}} \) and \( f_{\text{max}} \) remain constant, and we require the daily flow plan for this feed to remain constant.

In order to meet production feed \( d_t \), each time period, the volume of material transferred needs to be determined:

- volume to unload from supply node \( s \) to tank \( k \) during time \( t \), \( y_{s,k,t}^{\text{in}} \), and
- volume to unload from tank \( k \) to meet production feed in time \( t \), \( y_{k,t}^{\text{out}} \).

In addition, the following assignment variables on these flows are needed to address various operational constraints:

- if supply barge \( s \) is unloaded during time \( t \), then \( \gamma_{s,t} = 1 \).
- If tank \( k \) provides mixture to meet production feed in time \( t \), then \( \sigma_{k,t} = 1 \).

To model our objective of meeting demand and unloading requirements, we maximize a weighted sum of the amount of demand met and product unloaded.

Additional operational constraints include the following. The number of times a barge can be unloaded is limited to two, and all unload operations for the barge must be performed within a range of \( ul_{\text{max}} \) time periods (e.g., seven days). Flow from tanks to meet production feeds must be continuous for each campaign (or demand run).

Conservation of flow must be maintained, such that the sum of the initial tank inventory level and the volume of supply that flows into the tank less the volume that flows out for production feed equals the final tank inventory level. The resulting levels of chemical properties from this flow must meet the specified composition requirements. Due the relatively small sizes of the storage tanks in this work, we allow storage tanks to both receive supply and provide material for production feed simultaneously. If this occurs in a given time period, for the sake of simplicity, we model that complete mixing of inflows occurs before any outflows within that time period. The problem and constraints are formally specified in Section 4.
3 Literature Review

The blending and scheduling problems, also known as multi-period pooling problems, studied in the literature vary widely in network size, scope, objective, and detail. Such problems, similar to the standing pooling problem, are typically characterized as bilinear nonconvex MIQCP’s due to the balance of specs (or properties) in mixing and splitting of product. This is referred to as ‘linear blending’, and is often an approximation of what occurs in practice (as e.g. the mixing process may not finish completely before material is split into multiple streams). However, similar problems can also involve linear representation for spec balance, or a more highly nonlinear representations for balance of some specs (‘nonlinear blending’, as in the extended pooling problem), depending on the nature of the operational network or the specifications to track.

The underlying resource networks can be small and separated into discrete layers, and can represent front-end acquisition, as in e.g. [45, 50, 11, 6] and the current work; back-end operations and sales, as in e.g. [13]; or the full process from acquisition to sales, as in [45, 64]. The networks can also span larger, more complex supply chains, as in [52, 2]. The scope of the problem can include scheduling only, as in [44, 41, 30, 6] and the current work, but can also extend to include more high-level acquisition [50], production planning decisions [38, 37, 61], or more detailed equipment control decisions [15]. Some works, such as [22, 23], integrate all three levels of planning, scheduling, and control in a single model.

Common examples of objective functions used in similar problems include maximization of profits [53, 54, 29, 9, 3, 36, 2, 64] or gross margins [45, 50, 7, 16, 6], or minimization of operating costs [45, 33, 35, 32, 7, 6]. The objective function used in [14] in particular is similar to that in this work, but includes additional penalties for performing mixing operations, missing the demand composition requirements, and for not respecting the maintenance requirements. [13] maximizes the yields of final products. Many problems have only a single type of supply source, which can be characterized either by discrete arrivals [45, 50, 11, 7, 6] or continuous supply streams e.g. [33, 32, 13, 3, 16, 36]. Some works require multiple types of supply sources. For example, in [53, 54, 34, 9, 64], there are both small barges with a single type of inventory and larger barges, referred to as very large crude carriers (VLCCs), that carry multiple packages with varied compositions. Some works explicitly include vessel scheduling decisions to be made at unloading points, such as docks [53, 34]. Most works assume known supply and demand quantities or bounds, but some, such as [48, 38, 13, 19, 12, 15, 49], explicitly take into account stochastic supply and demand.

Muti-period pooling problems classically come in the forms of P, Q, or PQ formulations. The Q formulation is source-based, the P formulation is spec-based, and the PQ is a hybrid of the two. That is, the Q formulation tracks compositions based on origin fractions or volumes, while the P formulation tracks compositions explicitly based on the concentrations of the physical qualities to track. See [19] for a discussion of these models. Some examples of spec-based formulations can be found in [13, 11, 6], while examples of source-based formulations can be found in [53, 45, 16, 9]. Comparing to the traditional pooling problem, a spec-based formulation mirrors the traditional P-formulation while a source-based formulation mirrors the Q or PQ-formulations. This choice of representation affects both the required number of bilinear terms and the MILP tightness of the formulation. Overall, the number of bilinear terms for a source-based for-
mulation scales with the number of mixing or splitting nodes in the network and the number of distinct source compositions, while the number of bilinear terms for a spec-based formulation scales instead with the number of specs that require tracking. As such, this choice of representation can have a significant effect on both the tightness of the MIP and the number of required bilinear terms. While source-based formulations can be tighter [36], they are only viable in situations where the number of distinct sources is relatively small. In this work, as many as 35 distinct source compositions can be represented within the scheduling horizon, rendering source-based formulations computationally expensive.

3.1 Comparison to Similar Research

We highlight and compare three research streams that are similar to this current work. Note that many differences between this work and others stem from the specialty chemical nature of our problem, combined with the properties of the production site for the application of interest.

Castro [6] studies a similar problem motivated by crude oil operations in refineries. This problem is also addressed in [31, 46, 45, 7], among others. The problem involves barge-based supply where the barges carry a single type of crude, arrive within given time intervals, and unload according to the arrival sequence. The barges unload to storage tanks. The material in the storage tanks are mixed into charging tanks that supply crude oil into charging tanks that supply crude oil distillation units (CDUs).

The similarities of the core problem include a similar network structure, a barge-based supply stream, and the capacity of each storage tank (∼1-2 barges worth). Both works solve front-end operational scheduling problems and determine feed plans for downstream operations. In [6], such feeds are routed through crude oil distillation units (CDUs). In this current work, the chemicals feed a single production feed.

However, the differences between the problems affect the modeling and solution approaches. In [6], the barges arrive at regular intervals and are unloaded in the order of arrival, with only one barge unloaded at a time. In the current work, multiple barges (up to a limit) may be unloaded simultaneously and the order of unloading needs to be determined. The shorter scheduling horizon in [6] (∼15 days, ∼3 barges) allows for solution over the entire scheduling horizon at once, while the long scheduling horizon (up to 365 days, ∼30 barges) in the current work necessitates a heuristic such as a rolling horizon scheme. At the same time, the larger network in [6] reduces the length of the scheduling horizon over which the problem can be solved within a reasonable time. A significant observation of [6] is that, for the problem considered, the choice of objective function significantly impacted the relative efficacy of the discrete-time and continuous-time formulations. Another key difference is that the production schedule for each CDU is not fixed in [6], but is chosen based on gross margins per crude or on operating costs. On the other hand, a CDU can be sourced from only a single charging tank at a time.

In Castro’s work, the source-based choice is convenient because there are few composition types to track. In the current work, the source-based formulation is prohibitive due to the large amount of different possible composition types, and a spec-based formulation is used instead. As mentioned in [6], when viable, source-based formulations yield tighter MILP relaxations [36], and can yield better computational
3.2 Related Solution Methodologies

Castro [3] addresses a similar multi-period pooling problem with composition constraints on the flows to demand stream (tested on both per-flow and a mixture bases). Similar to the solution approach used in [6], Castro employs an iterated MILP-NLP approach, where the MILP stage is represented using base-10 NMDT. However, the utilized objective functions and formulations differ. In [3], a discrete-time formulation is used, and both spec-based and source-based formulations are presented, and compared with commercial global solvers. The objective function is profit maximization as opposed to gross margin maximization or cost minimization. The problem instance tested has two source streams, up to four blending/storage tanks for which tank-to-tank transfer operations are allowed, and two demand streams, and are solved with a time horizon of up to four periods.

Oddsdottir [50] addresses a similar network, with the same structure (barges to storage tanks to demand (CDUs)). The main differences in this respect involve barge availability (less restrictive time windows), the number of storage tanks (6), and the number of demand streams (2). Another common feature is the requirement of a much longer scheduling horizon (3 months) and in the flexibility in barge arrivals. However, the goal in [50] is related to procurement planning instead of operational scheduling. As such, the choice of which barges to order and when is a problem decision, and a courser time-grid is used (three days per period), resulting in a total of 30 time periods to model. Additionally, in both works, multiple barges might be unloaded within a single time period (up to a limit), possibly contributing to the choice of a discrete-time formulation. Further, a given barge may unload to multiple storage tanks in a given time period. As with [6], however, the limited number of crude types in the system in [50] allow for a reasonably compact source-based formulation, which would be unwieldy in the current work.

3.2 Related Solution Methodologies

Rolling planning aside, the solution approach selected in [50] is comparable to the approach in [6], but with some key differences. As in [6], a source-based formulation is used and composition constraints are formulated with respect to crude volumes instead of crude volume fractions. However, instead of discretizing, [50] applies a MILP approximation to the problem in which the constraints enforcing correct proportions for the outflow composition are reformulated in a manner that allows composition discrepancy in the outflows. A nonconvex MIQCP stage is then applied to resolve the composition discrepancy, yielding near-optimal feasible solutions in a much shorter time frame than the full nonconvex MIQCP: Running in GAMS with BARON, over 20 datasets, the average solution gap for this PRONODIS method (compared to the full nonconvex MIQCP) was 1.1%, with an average computation time of 20s (compared to over 13000s for the full nonconvex MIQCP).

A wide variety of formulations and approaches have been developed to tackle these difficult scheduling problems, with varying time-scheduling formulations, spec representations, and algorithms to deal with the problem nonlinearity, which usually incorporate some form of MILP approximation to the basic nonconvex MIQCP.
3.2 Related Solution Methodologies

A number of continuous-time and discrete-time scheduling formulations have been presented. The choice of time-formulation can have a significant impact on problem performance, as showcased in works such as [54] and [46, 16]. The preferred scheduling formulation can depend on the type of objective function used, as observed in [6], as well as on the types of operational constraints to be considered. Compared to continuous-time formulations, discrete-time formulations frequently yield simpler representations, with a trade-off of more integer variables. Even so, problems with certain types of operational constraints, such as variable flow rates and constraints on the number of simultaneous operations (such as barge unloads), can be significantly easier to model and solve in discrete-time (as seen in e.g. [6]). A discrete-time formulation is also used in our work.

A number of solution methodologies have emerged to deal with the handling of the nonlinear nature of these scheduling problems, particularly over long scheduling horizons. Such methods include single-step [63, 45, 50, 7, 3, 6] and iterated MILP-NLP [27, 26, 9, 14, 64] approaches, which have emerged as some of the most efficient methods to ensure small optimality gaps for relatively short scheduling horizons. Other methods include MILP-(smaller MINLP) iterations [50, 36] and iterative MILP-refinement methods [53, 53, 41, 34, 30].

For longer scheduling horizons, rolling planning methods are often used to limit the number of integer variables. Such models typically yield good feasible solutions far faster than the full model, with far better scaling of computation times with the scheduling horizon. In general, a rolling horizon model operates on a discrete time grid by representing in detail only the ‘present’ segment \( \{t^h, \ldots, t^h + H^{roll} - 1\} \) of the scheduling horizon \( \{1, \ldots, H\} \) at each iteration, then taking a step forwards in time of size \( t^{step} \leq t^h \) for the next iteration, freezing the decisions made in the interval \( \{t^h, \ldots, t^h + t^{step} - 1\} \) and proceeding until the full horizon is reached.

In some rolling planning models, such as those in [57, 13, 50, 59, 37] and heuristic H2 in [61], all aspects of the ‘past’ decisions in \( \{1, \ldots, t^h - 1\} \) are frozen; in others, such as [9] and heuristics H1, H3, and H4 in [61], only certain variables are frozen (most commonly the binary/integer decisions). The treatment of the ‘future’ portion of the scheduling horizon \( \{t^h + H^{roll}, \ldots, H\} \) varies considerably between models: Some models exclude the ‘future’ from the current model entirely, as in [57, 13, 59, 39, 9], whereas some relax complicating constraints [38, 61] and/or treat integer variables as continuous [37, 61]. Some works, such as [13, 50, 10, 56], deal with uncertain events by updating uncertain data or events at each (or certain) rolling planning step(s). For bi-level integrated planning and scheduling models, it is common to solve the scheduling subproblem only for the ‘present’ segment, while including (a subset of) the future segments in the planning subproblem, as is done in [35, 66, 57, 56]. Some works use an alternate framework based on supply arrivals to determine the time steps for each rolling iteration, as in [53, 54, 34, 9]; [34] in particular allows for some backtracking to correct spec quality discrepancies in the current plan.

Finally, many ideas for MILP approximations to deal with bilinear terms have emerged in recent years, especially for the pooling problem. Some methods are either binary-free, or add a number of binary variables that scales linearly with the desired level of precision. These include various linearization techniques [25, 44], piecewise-linear approximation [51, 16], outer-approximation methods [64], and McCormick or piecewise McCormick envelopes [40, 44, 51, 14]. [4] in particular gives a numerical comparison for fifteen different piecewise McCormick-related schemes, split into three classes of schema. For other methods,
often utilizing digit-based representations for a single variable, the number of binaries scales logarithmically with the desired precision. Such methods include multi-parametric disaggregation \([60, 28, 7, 8, 4, 6]\) and other techniques \([62, 13, 20, 80, 21]\). Approximation via generalized disjunctive programming (GDP) is another, far more expressive, approach that has been used to tackle pooling-type problems. Castro and Grossmann et. al. have derived a MILP-based formulation for variants of multi-parametric disaggregation as a disjunctive program \([18, 28, 7, 3, 4]\), and have also applied it separately \([55, 6, 36]\). In this work, we choose to directly discretize the inventory specs of each tank using a formulation based on normalized multi-parametric disaggregation (NMDT), as set forth in \([4]\), yielding a relatively simple and highly efficient model in conjunction with a rolling planning approach.

We were unable to find any other work in the literature which combines a version of the multi-parametric disaggregation technique with a version of rolling planning; this combination is a key novelty of this work. The primary results of this work include the speed of the method achieved while yielding near-optimal solutions without compromising the qualities of the specs in the demand feeds, combined with the incorporation of constraints on the ratios of specs in the demand feeds.

## 4 Mathematical Model

We describe a nonconvex MIQCP model in this section that introduces variables to handle the constraints. Data in the problem is written with an underscore, continuous variables are written with lower-case letters, binary variables are written with Greek letters, and sets are written with capital letters.

### 4.1 Terminology

- **Run** Production run, consisting of a number of consecutive days for which the production feed must be constant in terms of volume.
- **Feed** Flow from a storage tank to the downstream production processes.
- **Mixing** The mixing of material from a number of different sources to a single destination.
- **Splitting** The splitting of material from a single source to a number of different destinations.
- **Spec** Concentration of a certain type of chemical property.
4.2 Sets and Parameters

Sets

\( S \) : Set of supply nodes (barges with raw material), indexed by \( s \).
\( S_k \) : Set of supply barges allowed to unload to tank \( k \in K \).
\( S_t \) : Set of supply barges available during time period \( t \in T \).
\( K \) : Set of tanks, indexed by \( k \).
\( K_s \) : Set of tanks to which supply barge \( s \in S \) is allowed to unload.
\( Q \) : Set of chemical properties, indexed by \( q \).
\( Q_{rat} \) : Pairs of specification for which ratio bounds are to be enforced.
\( R \) : Set of non-overlapping demand runs (consecutive days for which the production feed is constant).
\( T \) : Set of time periods (days), such that \( T := \{1, 2, \ldots, H\} \).
\( T_r \) : Time periods corresponding to run \( r \) where runs do not overlap, such that \( T_r = \{t_r^i, t_r^i + 1, \ldots, t_f^r\} \subseteq T \).
\( T_s \) : Time periods during which supply \( s \in S \) can be unloaded, such that \( T_s = \{t_s^{min}, t_s^{min} + 1, \ldots, t_s^{max}\} \).
\( T_d \) : Time periods where there is some demand, such that \( T_d = \bigsqcup_{r \in R} T_r \).

Parameters

\( c_{inb}^s \) : Penalty cost per volume of not unloading supply from supply node \( s \).
\( c_{dmd}^d \) : Penalty cost per volume of not meeting production feed volume for time period \( t \).
\( d_t \) : Demand to meet in time step \( t \in T \). Must be constant throughout each run.

Thus, \( d_t = d_{t'} \) for all \( t, t' \in T_r \) for \( r \in R \). Also, \( d_t = 0 \) for \( t \not\in T_d \).
\( f_{s,q} \) : Value of chemical characteristic \( q \in Q \) in supply source \( s \in S \).
\( f_{k,q,0} \) : Initial value of chemical characteristic \( q \in Q \) in tank \( k \in K \).
\( f_{min}^{q,t}, f_{max}^{q,t} \) : Bounds on the value of chemical characteristic \( q \in Q \) for the demand to be filled in time step \( t \in T \).
\( H \) : Scheduling time horizon (implicit in the definition of \( T \)).
\( f_{min}^{q_1, q_2, t}, f_{max}^{q_1, q_2, t} \) : Bounds on the ratio of chemical characteristics \( f_{q_1}, f_{q_2}, (q_1, q_2) \in Q_{rat} \) for the demand to be filled in time step \( t \in T \).
\( t_{kp}^{min} \) : Minimum percentage of feed each day that can originate from tank \( k \) (if used).
\( ul^{max} \) : Maximum number of supply barges that can be unloaded each day.
\( ulp_s^{min} \) : Minimum percentage of inventory from supply barge \( s \) that can be unloaded each day that unloading occurs for \( s \).
\( ult^{max} \) : Maximum permissible gap between the first and last unloads for each supply barge.
\( v_s \) : Volume of supply available for unloading from supply node \( s \in S \).
\( v_k, 0 \) : Initial inventory volume of tank \( k \in K \).
\( v_{k, min}^{min}, v_{k, max}^{max} \) : Bounds on permissible inventory for each tank \( k \in K \).

Variables (All variables are non-negative)
4.3 Constraints

Assignment: \( \gamma_{s,t} : = \begin{cases} 1 : \text{supply barge } s \text{ is unloaded in time step } t \\ 0 : \text{otherwise} \end{cases} \quad s \in S, t \in T \)

\( \sigma_{k,t} : = \begin{cases} 1 : \text{tank } k \text{ provides material for production feed in time step } t \\ 0 : \text{otherwise} \end{cases} \quad k \in K, t \in T \)

Transfer: \( y_{in}^{s,k,t} : \text{Volume of raw material to transfer from supply } s \in S \text{ to tank } k \in K \text{ in time step } t \in T \)

\( y_{out}^{k,t} : \text{Volume of material to feed to demand from node } k \in K \text{ in time step } t \in T \)

Resource: \( v_{end}^{k,t} : \text{Volume of inventory in tank } k \in K \text{ at the end of time step } t \in T \)

\( v_{mid}^{k,t} : \text{Volume of inventory in tank } k \in K \text{ after material is supplied from supply barges, but before feed operations, in time step } t \in T \)

Demand: \( v_{prod}^{t} : \text{Total volume fed to production in time step } t \in T \)

Tank specs: \( f_{k,q,t}^{prod} : \text{Value of chemical characteristic } q \in Q \text{ in tank } k \in K \text{ at the end of time step } t \in T \)

Final specs: \( f_{prod}^{q,t} : \text{Value of chemical characteristic } q \in Q \text{ at production at the end of time step } t \in T \)

Unload time: \( t_{first}^{s} : \text{Earliest unload date for supply barge } s \)

\( t_{last}^{s} : \text{Latest unload date for supply barge } s \)

Slack: \( v_{s\text{unused}}^{t} : \text{Volume of material that is not unloaded from supply barge } s \in S \text{ in time step } t \in T \)

\( d_{t\text{unmet}} : \text{Volume of production demand missed in time step } t \in T \)

Objective: Our objective is to minimize the amount of material that is not unloaded from barges \( (v_{s\text{unused}}) \) and the demand that is unmet \( (d_{t\text{unmet}}) \). To accomplish this, we maximize the sum of the value of the raw material unloaded from the barges and the value of the production feed demand that is met. The amount of material unloaded for a barge \( s \) is exactly \( \gamma_{s} - v_{s\text{unused}}^{s} \), and the amount of demand met at time \( t \) is \( d_{t} - d_{t\text{unmet}}^{t} \). Hence, we have

\[
\max \left( \sum_{s \in S} \xi_{s}^{in} \cdot (\gamma_{s} - v_{s\text{unused}}^{s}) + \sum_{t \in T} \xi_{t}^{d\text{md}} \cdot (d_{t} - d_{t\text{unmet}}^{t}) \right) \tag{1}
\]

4.3 Constraints

The following constraints limit the solution space of the assignment, blending, and mixing problem. In the following discussion, note that variables \( y^{in}, y^{out} \) correspond to flows, while the variables \( v \) correspond to stationary volumes at certain times including volumes on barges \( v_{s} \), volumes in tanks \( v_{k,t} \), and volumes required for production feed \( v_{t} \).

Initial Conditions: The volume of material in the tanks and the chemical characteristics of the material are initialized. Specifically, \( v_{end}^{k,0} = \gamma_{k,0} \) enforces that the initial volumes in each storage tank are correctly represented, and \( f_{k,q,0} = \bar{f}_{k,q,0} \) enforces that the initial chemical characteristics in each storage tank are specified.

\[
\begin{align*}
v_{end}^{k,0} &= \gamma_{k,0} \quad &k \in K \\
f_{k,q,0} &= \bar{f}_{k,q,0} \quad &k \in K, q \in Q
\end{align*}
\]
4.3 Constraints

**Flow**: For these flow constraints, we assume that inbound supply material is unloaded and blended in the tanks before any material is drawn from the demand tanks to meet production feed on any given day. The model assumes that in the first half of a time period, barges unload to tanks (3a), adding to the volume previously in the tank and creating an intermediate volume $v_{k,t}^{\text{mid}}$. In the second half of the time period, material is unloaded from the tanks (3b) to support production (3c), while within required volume bounds at production (3d) and (3e).

\[
\sum_{s \in S_k} y_{s,k,t}^{\text{in}} + y_{k,t-1}^{\text{end}} = v_{k,t}^{\text{mid}} \\
v_{k,t}^{\text{mid}} - y_{k,t}^{\text{out}} = v_{k,t}^{\text{end}} \\
\sum_{k \in K} y_{k,t}^{\text{out}} = v_{t}^{\text{prod}} \\
v_{k,t}^{\text{end}} \geq \Sigma_{k}^{\text{min}} \\
v_{k,t}^{\text{mid}} \leq \Sigma_{k}^{\text{max}}
\]

**Chemical Characteristics**: The following constraints capture the blending and mixing of chemical characteristics. Specifically, (4a) ensures the characteristics of the material in a tank at the end of the previous time period are combined with the material from supply barges to create a mix. Expression (4b) determines the characteristics of the material provided from tanks to meet production feed.

\[
f_{k,q,t} \cdot v_{k,t}^{\text{mid}} = \sum_{s \in S_k} f_{s,q} \cdot y_{s,k,t}^{\text{in}} + f_{k,q,t} - 1 \cdot v_{k,t}^{\text{end}} \\
f_{q,t}^{\text{prod}} \cdot v_{t}^{\text{prod}} = \sum_{k \in K} f_{k,q,t} \cdot y_{k,t}^{\text{out}}
\]

Thus, while supply unloads and production feeds can occur in the same time period (day), they are not considered to occur simultaneously.

**Demand**: The following constraint set captures the production feed volume that is not met each time period, $d_t^{\text{unmet}}$.

\[
v_{t}^{\text{prod}} = d_t - d_t^{\text{unmet}}
\]

**Feed Characteristics**: Both the chemical characteristics (6a) and ratios of characteristics (6b) need to be within the required bounds. Note that in practice for (6b) the denominator would be multiplied through, as numerical issues can arise when the denominator is small.

\[
z_{q,t}^{\text{min}} \leq \frac{f_{q,t}^{\text{prod}}}{f_{q,t}^{\text{prod}}} \leq z_{q,t}^{\text{max}} \\
z_{q_1,q_2,t}^{\text{min}} \leq \frac{f_{q_1,t}^{\text{prod}}}{f_{q_2,t}^{\text{prod}}} \leq z_{q_1,q_2,t}^{\text{max}}
\]

**Total Inflow**: Ideally, the volume of supply on each barge is fully unloaded. Expression (7a) captures the volume of supply that is not unloaded ($v_s^{\text{unloaded}}$). In addition, (7b) ensures no material is unloaded from
4.3 Constraints

MATHEMATICAL MODEL

a supply barge when the barge is not available.

\[
\sum_{k \in K} \sum_{t \in T_s} y_{s,k,t}^{in} = y_s - v_s^{unused} \quad s \in S
\]

(7a)

\[
y_{s,k,t} = 0 \quad s \in S, k \in K_s, t \in T \setminus T_s
\]

(7b)

Continuous Feed During Runs: The volume of supply provided by each tank to meet production feed needs to remain constant during a production run.

\[
y_{out}^{k,t} = y_{out}^{k,t-1} \quad r \in R, k \in K, t \in T_r \setminus \{t_r^i\}
\]

(8)

Tank Feed Share Constraints: If production feed is supplied by tank \( k \) during time period \( t \) (such that \( \sigma_{k,t} = 1 \)), then the volume of production feed \( (y_{k,t}^{out}) \) must be at least \( t/kp_k^{min} \) of the demand \( d_t \) at time \( t \), as enforced by (9a). If \( \sigma_{k,t} = 0 \), then (9b) ensures no feed is provided by tank \( k \) for time \( t \).

\[
y_{out}^{k,t} \geq \left( t/kp_k^{min} \cdot d_t \right) \cdot \sigma_{k,t} \quad k \in K, t \in T
\]

(9a)

\[
y_{out}^{k,t} \leq d_t \cdot \sigma_{k,t} \quad k \in K, t \in T
\]

(9b)

Supply Limitations: Although it is preferred that barges are completely unloaded in one day, due to tank storage capacity, this may need to be done over several days. Hence, we constrain the number of unloads to be bounded by \( bul_{max} \), which are 2 in all of our instances (10a). The number of times any barge is unloaded within a time period is also limited (10c). Expression (10b) serves as a tightening constraint to ensure that barges are not unloaded when they are not available. If a supply barge is not unloaded during a time period (such that \( \gamma_{s,t} \) is zero), then no volume is unloaded from the barge (10d).

\[
\sum_{r \in T_s} \gamma_{s,t} \leq bul_{max} \quad s \in S
\]

(10a)

\[
\gamma_{s,t} = 0 \quad s \in S, t \in T \setminus T_s
\]

(10b)

\[
\sum_{s \in S} \gamma_{s,t} \leq ul_{max} \quad t \in T
\]

(10c)

\[
y_{s,k,t}^{in} \leq v_s \cdot \gamma_{s,t} \quad s \in S, t \in T, k \in K_s
\]

(10d)

Unload Inventory Percentage Constraints: (11) enforces that the percent of inventory unloaded per supply barge remains above required minimum \( ulp_{s}^{min} \) each day, if unloading is to occur that day.

\[
\sum_{k \in K_s} y_{s,k,t}^{in} \geq (v_s \cdot ulp_{s}^{min}) \cdot \gamma_{s,t} \quad s \in S, t \in T
\]

(11)

Unload Time Gap Constraints: In an effort to minimize the cost associated with barges remaining in the area for an extended time, these constraints limit the time duration for a supply barge to remain in the unloading area. To accomplish this, constraints are included to limit the interval \([t_{s}^{first}, t_{s}^{last}]\). Expression (12a) ensures that \( t_{s}^{first} \) is no later than the first period that any inventory is unloaded for barge \( s \). Expression (12b) ensures that \( t_{s}^{last} \) is no earlier than the last period that any inventory is unloaded for
4.4 Reformulation without spec variables $f_{q,t}^{\text{prod}}$  

For the implementation of this model, we adjust the model to eliminate terms consisting of only a single $f^{\text{prod}}$ variable from the model, restricting the model to product terms $f \cdot v$ or $f^{\text{prod}} \cdot v$, referred to as spec-volume terms. In this way, after discretization, we no longer have need to explicitly keep track of these specs at all, relying solely on these spec-volume product terms. In addition, we eliminate the $f_{q,t}^{\text{prod}}$ and $v_{t}^{\text{prod}}$ variables.

The initial constraint (2b) is replaced with

$$f_{k,q,0} \cdot v_{k,0}^{\text{end}} = f_{k,0}^{\text{end}} \sum_{k,0}$$  \hspace{1cm} k \in K, q \in Q \hspace{1cm} (2b\text{-prod})$$

By multiplying (3b) by $f_{k,q,t}$, we obtain

$$f_{k,q,t} \cdot v_{k,t}^{\text{mid}} - f_{k,q,t} \cdot v_{k,t}^{\text{out}} = f_{k,q,t} \cdot v_{k,t}^{\text{end}} \hspace{1cm} k \in K, t \in T \hspace{1cm} (3b\text{-prod})$$

Note that the $v_{k,t}^{\text{mid}}$ variables could also be eliminated via the constraint (3b). However, as this elimination choice can impact the quality of the MILP approximation, we defer this discussion to a later section.

We reformulate the constraints (6a), (6b) by multiplying through by volume (and any denominators) to obtain the following:

$$f_{q,t}^{\text{min}} \cdot v_{t}^{\text{prod}} \leq f_{q,t}^{\text{prod}} \cdot v_{t}^{\text{prod}} \leq f_{q,t}^{\text{max}} \cdot v_{t}^{\text{prod}} \hspace{1cm} t \in T, q \in Q \hspace{1cm} (6a\text{-prod})$$

$$f_{q1,q2,t}^{\text{prod}} \cdot v_{t}^{\text{prod}} \leq f_{q1,t}^{\text{prod}} \cdot v_{t}^{\text{prod}} \leq f_{q2,t}^{\text{prod}} \cdot v_{t}^{\text{prod}} \hspace{1cm} t \in T, (q1,q2) \in Q_{\text{rat}} \hspace{1cm} (6b\text{-prod})$$

Finally, we substitute the definitions of $v_{t}$ and $f_{k,q,t} \cdot v_{t}$ in (3c) and (4b) into constraints (5), (6a-prod), (6b-prod)
To obtain:
\[
\sum_{k \in K} y_{k,t}^{\text{out}} = d_i - d_i^{\text{inlet}} \quad t \in T
\]
(5-sub)

\[
\min_{f_{k,q,t}} \sum_{k \in K} y_{k,t}^{\text{out}} \leq \sum_{k \in K} f_{k,q,t} \cdot y_{k,t}^{\text{out}} \leq \max_{f_{k,q,t}} \sum_{k \in K} y_{k,t}^{\text{out}} \quad t \in T, q \in Q
\]
(14a)

\[
\min_{f_{q,t}} \sum_{k \in K} f_{k,q,t} \cdot y_{k,t}^{\text{out}} \leq \sum_{k \in K} f_{k,q,t} \cdot y_{k,t}^{\text{out}} \leq \max_{f_{q,t}} \sum_{k \in K} f_{k,q,t} \cdot y_{k,t}^{\text{out}} \quad t \in T, (q_1, q_2) \in Q_{\text{rat}}
\]
(14b)

To summarize, the MINLP-Mixing reformulation is the initial model, but with (2b), (3b), (3c), (4b), (5), (6a), and (6b) replaced by (2b-prod), (3b-prod), (5-sub), (14a), (14b).

### 4.5 Model MINLP-SPLIT: Volume-based reformulation

The model presented above can be reformulated to keep track of volumes of chemical characteristics instead of percentages \( f_{k,q,t} \), removing \( f_{k,q,t} \) from the model. This reformulation results in a linear representation for the mixing constraints in (14a), (14b), (4a) and modified flow balance constraint in (3b-prod). However, there is now a nonlinear representation for the splitting process, in which some volume is split into multiple parts (e.g., future inventory and demand feed). In the original model, the definition and usage of \( f_{k,q,t} \) ensures that the outflow characteristics are correct. However, after reformulating without \( f_{k,q,t} \), this is not longer the case, so we must ensure that the outflow characteristics are correct.

To enact the reformulation, we make the following substitutions:

\[
f_{k,q,t} \cdot v_{k,t}^{\text{end}} = v_{k,q,t}^{\text{end}} \quad k \in K, q \in Q, t \in T
\]
(15a)

\[
f_{k,q,t} \cdot v_{k,t}^{\text{mid}} = v_{k,q,t}^{\text{mid}} \quad k \in K, q \in Q, t \in T
\]
(15b)

\[
f_{k,q,t} \cdot y_{k,t}^{\text{out}} = y_{k,q,t}^{\text{out}} \quad k \in K, q \in Q, t \in T
\]
(15c)

Thus, (3b-prod), (4a), (14a) and (14b) are written as

\[
v_{k,q,t}^{\text{end}} = v_{k,q,t}^{\text{mid}} - y_{k,q,t}^{\text{out}} \quad k \in K, t \in T
\]
(16a)

\[
v_{k,q,t}^{\text{mid}} = \sum_{s \in S_k} f_{k,s,t} \cdot y_{k,s,t}^{\text{in}} + v_{k,q,t}^{\text{end}} - v_{k,q,t}^{\text{mid}} \quad k \in K, q \in Q, t \in T
\]
(16b)

\[
\min_{f_{k,q,t}} \sum_{k \in K} y_{k,t}^{\text{out}} \leq \sum_{k \in K} f_{k,q,t} \cdot y_{k,t}^{\text{out}} \leq \max_{f_{k,q,t}} \sum_{k \in K} y_{k,t}^{\text{out}} \quad t \in T, q \in Q
\]
(16b)

\[
\min_{f_{q,t}} \sum_{k \in K} f_{k,q,t} \cdot y_{k,t}^{\text{out}} \leq \sum_{k \in K} f_{k,q,t} \cdot y_{k,t}^{\text{out}} \leq \max_{f_{q,t}} \sum_{k \in K} f_{k,q,t} \cdot y_{k,t}^{\text{out}} \quad t \in T, (q_1, q_2) \in Q_{\text{rat}}
\]
(16c)

In addition, the initial condition (2b-prod) is replaced with

\[
v_{k,q,0}^{\text{end}} = f_{k,q,0} \cdot y_{k,q,0} \quad k \in K, q \in Q
\]
(17)

Now, to enforce the requirement that the chemical characteristics sent to production feed match the chemical characteristics in the tank, that is:

\[
f_{k,q,t} = \frac{v_{k,q,t}^{\text{mid}}}{v_{k,t}^{\text{mid}}} = \frac{y_{k,q,t}^{\text{out}}}{y_{k,t}^{\text{out}}}
\]
(18a)
we add the following nonlinear constraint
\[
\begin{align*}
\nu^{f,\text{mid}}_{k,q,t} \cdot y^{\text{out}}_{k,q,t} &= \nu^{f,\text{out}}_{k,q,t} \cdot \nu^{\text{mid}}_{k,t} \\
k \in K, q \in Q, t \in T
\end{align*}
\] (19a)

To summarize, the MINLP-Splitting model is the initial model, but with (2b), (3b), (3c), (4a), (4b), (5), (6a), (6b) replaced by (17), (16a), (5-sub), (16b), (16c), and (19a).

5 MILP Approximation Methods

We consider approaches of changing the nonlinear portions of the model in Section 4.4 to an MILP to more efficiently solve the problem and more easily interact with a rolling planning approach. The core change is that we will replace all products \(y_{in} \cdot f\) and \(y_{out} \cdot f\) with approximations. The new models, given sufficiently accurate approximations, will produce feasible flows \(y_{in}\) and \(y_{out}\) that completely determine the actions of the system.

For a spec variable \(f \in [l, u]\) we describe it by a discrete part \(\hat{f} \in \{l, l + \epsilon, l + 2\epsilon, \ldots, l + \zeta \epsilon\}\) and a continuous part \(\Delta f \in [0, \epsilon]\), where we choose \(\epsilon\) such that \(u - \epsilon < 1 + \zeta \epsilon \leq u\). The discrete part \(\hat{f}\) will be modeled with binary variables in a way known as Normalized Multi-Parametric Disaggregation Techniques (NMDT).

For the products with \(\Delta f\), we will consider two options - restricting the value to a midpoint or using the McCormick relaxation. That is, the two main ideas considered are the models:

\[
\begin{align*}
\text{(Center)} & \quad x \cdot \hat{f} = \underbrace{x \cdot \hat{f}}_{\text{NMDT}} + \underbrace{x \cdot \Delta f}_{\text{Center}} \\
\text{(McCormick)} & \quad x \cdot \tilde{f} = \underbrace{x \cdot \hat{f}}_{\text{NMDT}} + \underbrace{x \cdot \Delta f}_{\text{McCormick}}
\end{align*}
\]

When using (Center), we relax (4a) to maintain flexibility on inflow volumes. We do this by replacing the left-hand side with its upper and lower bounds with respect to \(\Delta f\). These choices, along with various relaxations of redundant constraints are described in detail in this section.

We will first describe common linearizations that we will consider, then describe our two competing models, and then have a discussion of rolling planning approaches.

5.1 Linearization

We define here two types of linearizations that we will consider. First, the McCormick envelope is a 3-dimensional polytope \(M(x, \beta)\) that is the smallest convex set containing the equation \(x \beta = x \cdot \beta\) with upper and lower bounds on \(\beta\) and \(x\). We will assume \(0 \leq \beta \leq 1\) and \(\underline{x} \leq x \leq \overline{x}\). The McCormick envelope is

\[
M(x, \beta) = \left\{ (x, \beta, x^\beta) \in [\underline{x}, \overline{x}] \times [0, 1] \times \mathbb{R} \colon \begin{array}{c}
\underline{x} \cdot \beta \leq x^\beta \leq \overline{x} \cdot \beta \\
x - \overline{x} \cdot (1 - \beta) \leq x^\beta \leq x - \underline{x} \cdot (1 - \beta)
\end{array} \right\}.
\] (20)
More generally, suppose we have $\beta_0, \ldots, \beta_m \in \{0, 1\}$ such that $\sum_{j=0}^{m} \beta_j = 1$ hence $\sum_{j=0}^{m} x_j \beta_j = x$. Defining $x^\beta_j = x \cdot \beta_j$, we can then model the convex hull of all such solutions as

$$
M^m(x, \beta) = \left\{ (x, \beta, x^\beta) \in [x^{\min}, x^{\max}] \times [0, 1]^{m+1} \times \mathbb{R}^{m+1} : \begin{align*}
\sum_{j=0}^{m} \beta_j &= 1, \\
\sum_{j=0}^{m} x^\beta_j &= x \\
x - x^{\max} \cdot (1 - \beta_j) &\leq x^\beta_j \leq x - x^{\min} \cdot (1 - \beta_j)
\end{align*} \right\}.
$$

In the special case where $m = 1$, the set $M^1(x, \beta)$ is a bijective lifting of $M(x, \beta)$.

### 5.2 Discretization

We discuss two methods of linearizing or relaxing bilinear products: Normalized Multiparametric Discretization Technique (NMDT) \cite{4} and McCormick envelopes. The discrete variable $\dot{f}$, contained in the interval $[l, u]$, is converted into binary variables as

$$
\dot{f} = \lambda_0 + \varepsilon \sum_{i=1}^{n} \sum_{j=0}^{m} (j \lambda_i) \cdot \alpha_{ij}, \quad \sum_{j=0}^{m} \alpha_{ij} = 1 \text{ for all } i = 1, \ldots, n,
$$

where $\alpha_{ij} \in \{0, 1\}$ for all $i, j$ and $\lambda_i$ are appropriately chosen scalars. Given a desired level of precision $0 < \hat{\varepsilon} \leq u - l$, the choices of $\lambda_i$ (and resulting choices for $\varepsilon$, $m$, and $n$) will affect the number of binary variables $\alpha_{ij}$ that are needed to reach the desired precision.

1. Multiparametric Disaggregation Technique with base $b$ provided $l \geq 0$:

$$
\lambda_0 = 0, \lambda_i = b^{i-1}, m = b - 1; \rightarrow \varepsilon = b^{\lceil \log_b(\hat{\varepsilon}) \rceil}, n = \lceil \log_b(\frac{u}{\hat{\varepsilon}}) \rceil.
$$

(MDT)

2. Normalized Multiparametric Disaggregation Technique, with base $b$:

$$
\lambda_0 = l, \lambda_i = b^{i-1}, m = b - 1; n = \lceil \log_b \left( \frac{u-l}{\hat{\varepsilon}} \right) \rceil, \varepsilon = (u - l)b^{-n}.
$$

(NMDT)

3. Monolithic Disaggregation:

$$
\lambda_0 = l, n = 1, \lambda_1 = 1; m = \lceil \frac{u-l}{\varepsilon} \rceil, \varepsilon = (u - l)/m.
$$

(Mono)

(MDT) discretizes $\dot{f}$ independent of the lower bound $l$, while (NMDT) and (Mono) use the lower bound and can be interpreted as discretizing $\dot{f}$ after normalizing the bounds $[l, u]$ to $[0, 1]$. Incorporating the lower bound is crucial for reducing the number of binary variables needed (and hence the complexity of the model) to obtain a certain precision $\varepsilon$. Furthermore, the exponential approach of (NMDT) is vastly preferred over the unary approach of (Mono), allowing only $O(\log(1/\varepsilon))$ as opposed to $O(1/\varepsilon)$.

In the originating work for (NMDT) \cite{4}, the same number of digits $n$ (and hence, precision $\varepsilon$) is applied to all discretized variables (to maintain the same precision in the normalized space), while in the version...
5.2 Discretization

5 MILP APPROXIMATION METHODS

presented here, \( n \) is computed based on the desired precision \( \hat{\varepsilon} \) for each individual discretized variable (to maintain similar precision in the original space).

\[
D(x, f, \alpha) = \begin{cases} 
  f = \hat{f} + \Delta f \\
  \hat{f} = \lambda_0 + \varepsilon \sum_{i=1}^{n} \sum_{j=1}^{m} (j \lambda_i) \cdot \alpha_{ij} \\
  \sum_{j=0}^{m} \alpha_{ij} = 1 & i = 1, \ldots, n \\
  x^f = x \cdot f = \lambda_0 x + x \cdot \Delta f + \varepsilon \sum_{i=1}^{n} \sum_{j=0}^{m} (j \lambda_i) \cdot x^{\alpha_{ij}} \\
  x^{\alpha_{ij}} = x \cdot \alpha_{ij} & i = 1, \ldots, n, j = 0, \ldots, m \\
  x^\text{min} \leq x \leq x^\text{max} \\
  \alpha_{ij} \in \{0, 1\} & i = 1, \ldots, n, j = 0, \ldots, m 
\end{cases}
\]

(23)

Most of the nonlinear products are now linearized and we drop the variables \( \dot{f} \) and \( f \). In this way, \( f \) is now only recorded implicitly (approximately) through product terms \( x^{\alpha_{ij}}, x^f \), and \( x^\Delta f \). This produces the set

\[
\hat{D}(x, f, \alpha) = \left\{ (x, f, \dot{f}, \Delta f, x^f, x^{\alpha}) : \begin{align*} 
  x^f &= \lambda_0 \cdot x + x^\Delta f + \varepsilon \sum_{i=1}^{n} \sum_{j=0}^{m} (j \lambda_i) \cdot x^{\alpha_{ij}} \\
  (x, \alpha_i, x^{\alpha_i}) &\in M^n(x, \alpha_i) \\
  \alpha_{ij} &\in \{0, 1\} \\
  i = 1, \ldots, n, j = 0, \ldots, m 
\end{align*} \right\},
\]

(24)

where \( \lambda_0, n, \) and \( \varepsilon \) are computed from a specified \( \hat{\varepsilon}, x^\text{min} \) and \( x^\text{max} \).

The remaining nonlinear product \( x \cdot \Delta f \), approximated by \( x^\Delta f \), will be dealt with in two different ways.

The binarization in [21] is similar to (NMDT) with \( b = 2 \) after projecting out the \( \alpha_{i0} \) variables using the equation \( \sum_{j=0}^{1} \alpha_{ij} = 1 \), i.e., \( \alpha_{i0} + \alpha_{i1} = 1 \).

5.2.1 Choosing base \( b = 2 \) reduces variables.

We suggest the best model uses \( b = 2 \) since it uses asymptotically fewer binary variables than any other choice \( b > 2 \) (as \( \hat{\varepsilon} \to 0^+ \)).

**Proposition 1.** Let \( z_{\hat{\varepsilon}}(b) := (b - 1) \cdot n = (b - 1) \left\lfloor \log_b \left( \frac{u - l}{\hat{\varepsilon}} \right) \right\rfloor \) be the number of binary variables used to represent \( \dot{f} \), and let \( b_1, b_2 \) be two different bases. Then \( z'(b_1, b_2) := \lim_{\hat{\varepsilon} \to 0^+} \frac{z_{\hat{\varepsilon}}(b_1)}{z_{\hat{\varepsilon}}(b_2)} = \frac{(b_1 - 1) \ln(b_2)}{(b_2 - 1) \ln(b_1)} \).
5.2 Discretization

Proof. Asymptotically, as \( \varepsilon \to 0^+ \), we can remove the ceiling function in the definition of \( z_\varepsilon \). Hence

\[
z'(b_1, b_2) = \lim_{\varepsilon \to 0^+} \frac{(b_1 - 1) \log_{b_1} \left( \frac{u - l}{\varepsilon} \right)}{(b_2 - 1) \log_{b_2} \left( \frac{u - l}{\varepsilon} \right)} = \frac{(b_1 - 1) \log_{b_1}(b_2)}{(b_2 - 1) \ln(b_1)}
\]

Because \( z_\varepsilon(b) \) is an increasing function of \( b \), \( z'(b, 2) > 1 \), for all \( b > 2 \). Using the proposition, we compute \( z'(b, 2) \) for various values of \( b \) in [3]. In [4], Castro uses \( b = 10 \). As the table shows, this binarization will use nearly three times as many binary variables.

| \( b \) | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|
| \( z'(b, 2) \) | 1.26186 | 1.5 | 1.72271 | 1.93426 | 2.13724 | 2.33333 | 2.52372 | 2.70927 |

Table 3: Asymptotic number of binary variables required for base \( b \), as a multiple of the number required for base 2.

In summary, we suggest to use NMDT with a base 2 because it will require the fewest number of binary variables to achieve a desired accuracy.

5.2.2 Implied and redundant equations

Constraint (3b) (and subsequently (3b-prod)) couples the variables \( v^\text{mid}_{k,t}, y^\text{out}_{k,t}, \) and \( v^\text{end}_{k,t} \). For convenience, let us write \( x_1 = v^\text{mid}_{k,t}, x_2 = y^\text{out}_{k,t}, x_3 = v^\text{end}_{k,t}, \) and \( f = f_{k,q,t} \). That is,

\[
x_1 = x_2 + x_3.
\]

As a consequence of (25), by multiplying through by \( f, \Delta f, \) and \( \alpha_i \), we obtain three sets of valid equations

\[
\begin{align*}
    x_1^f &= x_2^f + x_3^f, \quad (26a) \\
    x_1^{\Delta f} &= x_2^{\Delta f} + x_3^{\Delta f}, \quad (26b) \\
    x_1^{\alpha_i} &= x_2^{\alpha_i} + x_3^{\alpha_i} & i = 1, \ldots, n. \quad (26c)
\end{align*}
\]

Consider the approximation of \( f \) and the resulting product form of the equations

\[
\begin{align*}
    f &= l + \Delta f + \varepsilon \sum_{i=1}^{n} 2^{i-1} \cdot \alpha_i \quad (27a) \\
    x_\kappa^f &= l \cdot x_\kappa + x_\kappa^{\Delta f} + \varepsilon \sum_{i=1}^{n} 2^{i-1} \cdot x_\kappa^{\alpha_i} & \kappa = 1, 2, 3 \quad (27b) \\
    x_\kappa^{\alpha_i} &= x_\kappa \cdot \alpha_i & i = 1, \ldots, n, \kappa = 1, 2, 3 \quad (27c)
\end{align*}
\]
5.3 MILP Models

Of key importance here for model reduction and a stronger formulation are the observations: (1) Bilinear terms can be eliminated using the above equations (2) Applying McCormick relaxations to these bilinear terms before variable elimination produces a tighter formulation.

First, although other model choices are possible, we project out the variables associated with $\kappa = 1$. Conveniently, equations (27b) and (27c), for $\kappa = 1$, are both implied by all other equations. In fact, variables $x^f_1$, $x^\Delta f_1$, and $x^{\alpha_1}$ by substitution, can be projected out of the model using equations (25), (26b), and (26c), then the equations (27b) and (27c) become tautologies. This observation provides convenient reduction choices in the model in terms of both variables and constraints.

Second, applying the projection/linear relaxation in the right order can improve the model. In particular, let $\mathcal{F}$ represent the feasible set given by the above equations, let $M^{123}(x, \alpha_\kappa)$ be the McCormick relaxation in the space of all variables (similarly, $M^{23}(x, \alpha_\kappa)$ in the space of variables associated with $\kappa = 2, 3$), $\text{proj}_{23}$ be the projecting into the space of variables with $\kappa = 2, 3$, and denote the removal of bilinear equations (27c).

Then we have the strict inclusion

$$\text{relax} \left( \text{proj}_{23} \left( \mathcal{F} \cap \bigcap_{\kappa=1,2,3} M^{123}(x, \alpha_\kappa) \right) \right) \subset \text{relax} \left( \text{proj}_{23}(\mathcal{F}) \cap \bigcap_{\kappa=2,3} M^{23}(x, \alpha_\kappa) \right). \quad (28)$$

That is, using the inequalities from the McCormick relaxation from the $\kappa = 1$ variables improves the formulation.

5.3 MILP Models

We will now outline the two models we propose to consider and then discuss how to integrate them with rolling planning.

5.3.1 Center Approximation

We create a mixed integer linear approximation to the main problem that will only approximately track constraints on specifications. To check feasibility of the resulting flows $y^{\text{in}}$ and $y^{\text{out}}$, these solutions must be plugged into the main model to recover spec values and ensure the constraints are met. The approximation error will be guided by how fine the discrete approximation of the spec variables is. For the model, a prevision $\hat{\epsilon}_q$ is chosen such that, given an accurate discretization, we should have $|f_{k,q,t} - \hat{f}_{k,q,t}| \leq \hat{\epsilon}$. The choice of $\hat{\epsilon}_q$, as discussed earlier, defines the choices of $\epsilon_q, n_q$ and $m_q$ which are implied parameters in the sets $\mathcal{D}(x, f_{k,q,t}, \alpha)$. Thus, for each spec $q$, there are (potentially) unique $\epsilon, n$ and $m$ that are used for the set $\mathcal{D}(x, f_{k,q,t}, \alpha)$. We write all products in terms of single variables and we relax (4a) to increase the number of feasible solutions to the approximate model.
5.3 MILP Models 5 MILP APPROXIMATION METHODS

We thus model \((3b_{\text{prod}}), (4a), (14a), (14b)\) as

\[
\begin{align*}
\nu_{k,q,t}^{\text{mid}} - \nu_{k,q,t}^{\text{out}} & = \nu_{k,q,t}^{\text{end}} & k \in K, t \in T, \\
\nu_{k,q,t}^{\text{mid}} - \frac{\epsilon_k}{2} \nu_{k,q,t}^{\text{mid}} & \leq \sum_{s \in S_k} \nu_{s,k,t}^{\text{in}} - \nu_{k,q,t}^{\text{end}} & k \in K, q \in Q, t \in T, \\
\nu_{k,q,t}^{\text{mid}} + \frac{\epsilon_k}{2} \nu_{k,q,t}^{\text{mid}} & \geq \sum_{s \in S_k} \nu_{s,k,t}^{\text{in}} + \nu_{k,q,t}^{\text{end}} & k \in K, q \in Q, t \in T, \\
\nu_{k,q,t}^{\text{min}} \cdot \sum_{k \in K} \nu_{k,q,t}^{\text{out}} & \leq \sum_{k \in K} \nu_{k,q,t}^{\text{out}} \leq \nu_{k,q,t}^{\text{max}} \cdot \sum_{k \in K} \nu_{k,q,t}^{\text{out}} & t \in T, q \in Q, \\
\nu_{q,t}^{\text{min}} \cdot \sum_{k \in K} \nu_{k,q,t}^{\text{out}} & \leq \sum_{k \in K} \nu_{k,q,t}^{\text{out}} \leq \nu_{q,t}^{\text{max}} \cdot \sum_{k \in K} \nu_{k,q,t}^{\text{out}} & t \in T, (q_1, q_2) \in Q_{\text{rat}},
\end{align*}
\]

The remaining constraints are generated using \(\tilde{D}\) with \(\lambda_0 = \nu_{k,q}^{\text{min}}\) and \(\Delta f_{k,q,t} = \frac{\epsilon_k}{2}\) and \(b = 2\),

\[
\begin{align*}
(\nu_{k,q,t}^{\text{mid}}, \alpha_{k,q,t}, \nu_{k,q,t}^{\text{mid}}, \Delta f_{k,q,t}, \nu_{k,q,t}^{\text{mid}}) & \in \tilde{D}(\nu_{k,q,t}^{\text{mid}}, f_{k,q,t}, \alpha_{k,q,t}) & k \in K, q \in Q, t \in T, \\
(\nu_{k,q,t}^{\text{end}}, \alpha_{k,q,t}, \nu_{k,q,t}^{\text{end}}, \nu_{k,q,t}^{\text{min}}, \nu_{k,q,t}^{\text{end}}) & \in \tilde{D}(\nu_{k,q,t}^{\text{end}}, f_{k,q,t}, \alpha_{k,q,t}) & k \in K, q \in Q, t \in T, \\
(\nu_{k,q,t}^{\text{out}}, \alpha_{k,q,t}, \nu_{k,q,t}^{\text{out}}, \nu_{k,q,t}^{\text{out}}, \nu_{k,q,t}^{\text{out}}) & \in \tilde{D}(\nu_{k,q,t}^{\text{out}}, f_{k,q,t}, \alpha_{k,q,t}) & k \in K, q \in Q, t \in T.
\end{align*}
\]

Note that each \(\alpha_{k,q,t}\) is a vector of \(n_{k,q,t}\) binary variables, which may vary in number depending on the accuracy used to approximate \(f_{k,q,t}\).

5.3.2 Redundant equations

Defining both sets of constraints in \((30a), (30b)\) creates some redundant inequalities. In particular, after omitting the \(k, q, t\) indices for convenience, the inequalities

\[
\begin{align*}
\nu_{i,j}^{\text{mid}} & \geq \nu_{i,j}^{\text{min}} \cdot \alpha_{i,j} & i \in I, j \in \{0, 1\} \\
\nu_{i,j}^{\text{end}} & \leq \nu_{i,j}^{\text{max}} \cdot \alpha_{i,j} & i \in I, j \in \{0, 1\}
\end{align*}
\]

are redundant due to \((3b)\) and the non-negativity of all volume variables.

5.3.3 Tightening feasible region to produce feasible solutions

Because the proposed model is an approximation of the original problem, the model might produce a schedule with flows that result in infeasible specs - violating either the spec bounds or spec ratio bounds. However, in most cases, the magnitude of these violations we observed to be within \(\epsilon/2\) for the proposed discretization methods.

Thus, to mitigate this error, we add a buffer of \(\epsilon/2\) to each spec requirement, so that the interval \([\nu_{k,q}^{\text{min}}, \nu_{k,q}^{\text{max}}]\) is replaced with \([\nu_{k,q}^{\text{min}} + \epsilon/2, \nu_{k,q}^{\text{max}} - \epsilon/2]\).
As for the spec ratios, we used differentials of the ratio to obtain an upper approximation for the magnitude of the buffer to use: If the spec ratio to control is $f_{q_1}/f_{q_2}$, where $f_{q_1}$ and $f_{q_2}$ are spec qualities to control, the buffer to use is given via the differential:

$$\Delta \left( \frac{f_{q_1}}{f_{q_2}} \right) = \frac{1}{f_{q_2}^2} \Delta f_{q_1} + \frac{f_{q_2}^\text{max}}{(f_{q_2}^\text{min})^2} \Delta f_{q_1},$$  \hspace{1cm} (32)$$

where $\Delta f_{q_1}$ and $\Delta f_{q_2}$ are the computed buffers $\hat{\varepsilon}/2$ for $f_{q_1}$ and $f_{q_2}$, respectively. Thus, the interval $[f_{q_1}^\text{min}, f_{q_2}^\text{max}]$ enforced for $\left( \frac{f_{q_1}}{f_{q_2}} \right)$ is replaced by $[f_{q_1}^\text{min}, f_{q_2}^\text{max} + \Delta \left( \frac{f_{q_1}}{f_{q_2}} \right), f_{q_1}^\text{max}, f_{q_1}^\text{max} - \Delta \left( \frac{f_{q_1}}{f_{q_2}} \right)]$. These buffers proved effective in largely resolving the issue of incorrect feed specs and spec ratios for the problems tested. The buffers could be reduced if they would result in possible ranges for specs or spec ratios that are smaller than the range provided by a single discretization bin (based on initial tank and supply inventories and the parameter $\hat{\varepsilon}_q$ specified by the user). However, for the purposes of this work, we will assume that $\hat{\varepsilon}_q$ is small enough that the resulting ranges for the demand specs are sufficiently large (with width $\geq \hat{\varepsilon}$ after tightening) that feasibility issues do not become problematic for the proposed discretization scheme.

These buffers result in a replacement of the parameters $f_{q_1}^\text{max}, f_{q_1}^\text{min}, f_{q_2}^\text{max}, f_{q_2}^\text{min}$ in Section 4 with their ‘buffered’ counterparts, tightened further by the possible feed specs based on initial tank inventory and inbound supply specs.

### 5.3.4 McCormick Based Approximation

In a similar way, the final model for the original NMDT method (with McCormick envelopes around the continuous product terms) is given as follows:

$$v_{k,q,t}^{f,\text{mid}} - v_{k,q,t}^{f,\text{out}} = v_{k,q,t}^{f,\text{end}}, \hspace{1cm} k \in K, t \in T \hspace{1cm} (33a)$$

$$v_{k,q,t}^{f,\text{mid}} = \sum_{s \in S_k} f_{s,k} \cdot y_{s,k,t}^{\text{in}} + v_{k,q,t}^{f,\text{end}}, \hspace{1cm} k \in K, q \in Q, t \in T \hspace{1cm} (33a)-\text{relax}$$

$$y_{k,q,t}^{\min} \cdot \sum_{k \in K} y_{k,q,t}^{\text{out}} \leq \sum_{k \in K} y_{k,q,t}^{\text{out}} \leq y_{k,q,t}^{\max} \cdot \sum_{k \in K} y_{k,q,t}^{\text{out}}, \hspace{1cm} q \in Q, t \in T \hspace{1cm} (33b)$$

$$\sum_{k \in K} y_{k,q,t}^{\text{out}} \cdot y_{k,q,t}^{\min} \leq \sum_{k \in K} y_{k,q,t}^{\text{out}} \leq \sum_{k \in K} y_{k,q,t}^{\text{out}} \cdot y_{k,q,t}^{\max}, \hspace{1cm} t \in T, \ (q_1, q_2) \in Q_{\text{rat}} \hspace{1cm} (33c)$$

Where, for all $k \in K, q \in Q, t \in T$,

$$v_{k,q,t}^{f,\text{mid}} \in [0, \varepsilon \cdot y_{k,q,t}^{\max}] \hspace{1cm} (34a)$$

$$v_{k,q,t}^{f,\text{end}} \in [0, \varepsilon \cdot y_{k,q,t}^{\max}] \hspace{1cm} (34b)$$

$$y_{k,q,t}^{\text{out}} \in [0, \varepsilon \cdot y_{k,q,t}^{\max}] \hspace{1cm} (34c)$$

$$\Delta v_{k,q,t}^{f,\text{mid}} \in [0, \varepsilon \cdot y_{k,q,t}^{\max}] \hspace{1cm} (34d)$$

$$\Delta v_{k,q,t}^{f,\text{end}} \in [0, \varepsilon \cdot y_{k,q,t}^{\max}] \hspace{1cm} (34e)$$

$$\Delta y_{k,q,t}^{\text{out}} \in [0, \varepsilon \cdot y_{k,q,t}^{\max}] \hspace{1cm} (34f)$$
Finally, the continuous product terms $\Delta v_{f,k,q,t}^{\text{mid}}$, $\Delta v_{f,k,q,t}^{\text{end}}$, and $\Delta y_{f,k,q,t}^{\text{out}}$ are modeled via their McCormick envelopes, for all $k \in K$, $q \in Q$, $t \in T$,

$$\Delta f_{k,q,t} \in [0, \varepsilon]$$  \hspace{1cm} (35a)

$$\left(\Delta f_{k,q,t}/\varepsilon, v_{k,q,t}^{\text{mid}}\right) \in M(\Delta f_{k,q,t}, v_{k,q,t}^{\text{mid}})$$ \hspace{1cm} (35b)

$$\left(\Delta f_{k,q,t}/\varepsilon, v_{k,q,t}^{\text{end}}\right) \in M(\Delta f_{k,q,t}, v_{k,q,t}^{\text{end}})$$ \hspace{1cm} (35c)

$$\left(\Delta f_{k,q,t}/\varepsilon, y_{k,q,t}^{\text{out}}\right) \in M(\Delta f_{k,q,t}, y_{k,q,t}^{\text{out}})$$ \hspace{1cm} (35d)

### 5.3.5 Error propagation in time horizon

From a perspective of numerical error propagation, the McCormick-based approach results in an exactly-represented tank-inflow blending process (constraint (4a)) for our problem: since only two ‘spec · volume’ terms contain variable specs, the volumes of the specs are coupled correctly. On the other hand, the splitting processes in (6a) are inexact, since the McCormick envelopes around the $\Delta f$ product terms effectively allow the specs leaving a tank to differ slightly from the specs propagated to the next time period. However, due to the tightness of the McCormick envelopes, once a post-blending value $\tilde{f}$ is chosen for a tank, the represented specs will remain in the interval $[\tilde{f}, \tilde{f} + \varepsilon]$ until the next blending operation in the tank, allowing a maximum possible spec error in the interval $[\varepsilon/2, \varepsilon]$ to accrue between blending operations.

On the other hand, the Center approximation presented here incurs an error of up to $\varepsilon/2$ immediately after blending into the tank, but then accrues no additional numerical error within that tank until the next blending operation. This allows the model to remain closer to feasibility for cases in which each tank does not receive supply from barges very frequently.

### 5.4 Rolling Horizon Approach

As mentioned in Section 1, we need a blending and scheduling optimizer that can efficiently optimize over a scheduling horizon of at least 3 months and up to 12 months. For the instances of interest, the target time for this optimization is typically preferred to be no more than five minutes (with a ten minute maximum). However, even the linear discrete approximation above is extremely slow even for a scheduling horizon of 120 days, requiring several hours of computation time in Gurobi 8.1, even for extremely coarse spec discretization (with a total of roughly six binary variables per tank per day for only three tanks). As such, we employ a rolling planning approach.

We present three major alternatives for rolling planning. These alternatives are characterized treatment of the ‘past’, ‘present’, and ‘future’, coupled with the choice of time periods (‘present’) to use for each rolling planning step. The ‘past’ is the portion of the planning horizon before the current rolling horizon window, the ‘present’ is the current horizon window, and the ‘future’ is the portion of the planning horizon after the current rolling horizon window. For the considered schemes, the ‘future’ is split into two partitions:
the ‘near future’ and the ‘far future’. We consider two alternatives for the treatment of the ‘past’ and ‘future’, and two major schemes for the choice of time periods.

We will compare two schedules for rolling planning - fixed-length periods and run-based periods. In both schemes, periods are chosen to be disjoint since previous performed test with overlapping periods performed slower.

Our fixed-length periods are simply equal length periods of some length $\Delta t$, except the last period may be truncated to not exceed the horizon length $H$. See Figure 3.

Figure 3: This figure shows specified demand runs, the barge time windows, and the periods that we use in the rolling horizon calculation for the fixed-length periods scheme (and considering a horizon $H = 30$ days).

Although this period schedule is consistent, it subdivides runs. This makes modeling consistent volume feed throughout a run complicated in the rolling planning process and can lead to poor choices early in the rolling planning. To alleviate this issue, we propose a ‘run-based’ schedule.

The run-based periods either

- Continue in a run to the end of the run or,
- Encompass all periods and gaps (days between runs) that are completely contained within a time change of $\Delta t$ from the beginning of period.

This choice makes it so that periods and gaps are never subdivided. This is depicted in Figure 4, where a sample partition resulting from this procedure, using $\Delta t = 7$, is illustrated. Formally, our choice of periods is described by Algorithm 1.
5.4 Rolling Horizon Approach

Algorithm 1 Rolling Planning with Run Based Sub-periods.

**Input:** Desired approximate length $\Delta t$ for a single rolling planning sub-period

**Output:** Sub-periods $P_i$ for rolling planning.

$t_0 \leftarrow 0$, $i \leftarrow 0$  \hspace{1cm} $\triangleright$ Initialize beginning of current rolling planning step and index.

while $t_i < H$ do

$t_{\text{end}} \leftarrow$ first ending time period of a run that is $\geq t_i$

$t_{\text{change}} \leftarrow$ latest ending time period of a run or gap between runs that is $\leq t_i + \Delta t$

$t_{i+1} \leftarrow \max(t_{\text{end}}, t_{\text{change}}) + 1$

$P_i \leftarrow [t_i, t_{i+1})$

$i \leftarrow i + 1$

end

Figure 4: This figure shows specified demand runs, the barge time windows, and the periods that we use in the rolling horizon calculation for the ‘run-based periods’ scheme (and considering $H = 30$ days).

Next, we will discuss the major schemes for the treatment of the past, present, near future, and future segments of the scheduling horizon. These schemes include a full-horizon scheme and a partial-horizon scheme.

**Full-Horizon Scheme** Given the current rolling horizon period $P_i = [t_i, t_i + 1)$ and a near future horizon $H^{NF}$, let $t_i^{NF} = \min(H, t_i + H^{NF} - 1)$. In our examples, we use $H^{NF} = 90$.

1. In the past, we freeze the binary decisions made, but allow all continuous variables to continue to vary. This allows the model some additional ability to compensate for poor early decisions, which (as we will see) helps to improve the optimality gap substantially.

2. In the present, the full model is used.

3. In the near future, the binary variables related to demand decisions and discretized tank compositions are relaxed to the interval $[0, 1]$, while the binary variables related to supply decisions are enforced to be binary. All other constraints are included.

4. In the far future, all binary variables are relaxed to the interval $[0, 1]$. All other constraints are included.
The decision not to relax supply decision binaries in the ‘near future’ is due to the inclusion of the ‘limited unloading window’ constraints for barges. These constraints rendered the inclusion of small objective function terms based on supply unloading decisions computationally prohibitive, due to poor branching performance of the solver. Before these were included, small objective function terms had been added to slightly favor earlier unloading times (all else equal), encouraging barges to be unloaded more quickly (and allowing for a model that more consistently avoided small unnecessary supply misses). After the inclusion of these constraints, the solver performance with these small objective terms became extremely poor, requiring at least an order of magnitude more computational time to solve to the same optimality gap in preliminary tests via Gurobi 8.1. When these terms were removed, the model began to exhibit very large extraneous supply misses, due in part to the poor linear relaxation of the ‘limited unloading window’ constraints, and in part to relaxations of the constraints limiting the number of unloads per barge. These poor relaxations caused the rolling planning model to favor unloading during the relaxed-binary periods, a decision which the model could not recover from. However, by including the supply-based binaries in the ‘near future’ (defined to end 90 days after the start of the ‘present’ period), preliminary testing showed that these unnecessary misses became small enough that their cost was considered to be acceptable, while the inclusion of these extra binary variables had only a moderate impact on computational cost (close to a 50% increase).

For all rolling horizon methods, we use 0.5% as the MIP gap. For the Full-Horizon scheme, though experimental results we learned that increasing this, on average, does not significantly reduce computation time, but may have a large effect on resulting objective values. In particular, we want to optimize as much as we can in each step since, due to the way we relax variables, the objective value in each subsequent step can only decrease. In particular, since we are targeting the best objective value possible, we attempt to make optimal decisions at every step.

**Partial-Horizon Scheme**

1. In the past, we fix all decisions made, and use them to re-compute accurate values for spec compositions to use for the next rolling planning period. Possible \([l, u]\) for the specs in each tank are re-computed, and the number of remaining unloads (and remaining volumes) for barges that have been unloaded in the past are reduced accordingly.

2. In the present, the full model is enforced.

3. In the near future, the binary variables related to demand decisions and discretized tank compositions are relaxed to the interval \([0, 1]\), while the binary variables related to supply decisions are enforced to be binary. All other constraints are included.

4. In the far future, all variables and constraints are omitted.
5.5 Post-processing and Error Mitigation

Table 4: Description of how variables are either fixed, fully active, relaxed to be continuous, or omitted in the two versions of the rolling planning horizon schemes.

| Vars | Past $[0, t_i)$ | Present $P_i$ | Near $(t_{i+1}, t_{i}^{NF}]$ | Far $(t_{i}^{NF}, H]$ |
|------|----------------|--------------|----------------------------|---------------------|
| $y, v$ | active | active | active | active |
| $y, v$ | active | active | active | active |
| $d$ | active | active | active | active |
| $t_s$ | active | active | active | relaxed |
| $\gamma$ | fixed | active | active | relaxed |
| $\sigma$ | fixed | active | relaxed | relaxed |
| $\alpha$ | fixed | active | relaxed | relaxed |

In the partial-horizon scheme, we note that it is possible that only a small part of the arrival window, even just a single period, for a single barge overlaps with the ‘present’ and ‘near future’. To compensate for this, we pro-rate the inventory to be unloaded from such barges: If only 10% of the volume on a barge overlaps with the currently-considered period, then we enforce that only 10% of the volume from the barge is to be unloaded in the current rolling planning step. In this way, as the model rolls forwards, more of the barge’s unloading window enters the visible horizon window, allowing for improved decisions regarding the volume on the barge.

5.5 Post-processing and Error Mitigation

After solving the approximation model, we check feasibility of the scheduled flows in terms of required spec quality bounds. To do so, using the flow variables $y^\text{in}, y^\text{out}$, we plug their computed values into the original model from Section 4 to uniquely determine the spec values $f_{k,q,t}$.

After the optimization of the discretized model with rolling planning, we compute specs $f_{k,q,t}$ from simulate the scheduling solution to reveal the true specs throughout the scheduling horizon, and report the true tank and feed volumes and compositions throughout the horizon in addition to the basic scheduling plan.

During preliminary computational testing, after simulating the scheduling solution to reveal the true specs throughout the scheduling horizon, we found that, without tightening the spec requirements, the MILP discretization techniques for the model often resulted in solutions for which the actual spec and spec ratios for the demand feeds were not within the specified ranges. However, this was largely remedied after tightening the feasible region as described in Section 5.3.3.
Table 5: The capacities of the three tanks we consider. Minimums are about 10% of total volumes. Note that barges hold between 1200 - 1400 metric tons. Hence a barge is comparable to our small tanks, and our large tank can hold slightly more than 2 barges.

| Barge Type | Type 1 | Type 2 | Type 3 | Type 4 |
|------------|--------|--------|--------|--------|
| Volume     | 1240   | 1360   | 1182   | 1360   |
| $c_{inb}$  | 1000   | 800    | 800    | 1000   |

Table 6: Example data for barge types, their capacities, and related objective values.

6 Numerical Results

To evaluate the performance of the proposed methods, we constructed a realistic problem with a 368-day scheduling horizon from real operational data with a 119-day scheduling horizon, extended to the desired horizon in a periodic fashion. The resulting problem had a total of 33 supply arrivals over the scheduling horizon, three available storage tanks, two specs ($S_1$ and $S_2$) to track, and one spec ratio to control. Note that the range of possible spec qualities was relatively narrow for some tanks, resulting in relatively few binary variables used to approximate the specs for each tank. See (5) for example data on the storage tanks. Other relevant parameters include $u_l^{max} = 2$, $u_l^{max} = 7$, $tkp_k^{min} = 10\%$ for each tank, and $ulb_k^{min} = 10\%$ for each supply barge. For the objective coefficients, we globally use $c_{dmd}^{t} = 3000$. $c_{inb}^{t}$ depends on the barge type, and is either $800$ or $1000$ for each type, as specified in Table 6. Figure 6 showcases the unloading and arrival windows for a sample instance with $H = 150$, while Figures 5,7, showcase sample solution schedules with $H = 40$. Note that the high variability in bounds for the $S1/S2$ ratio is due to the substantially different requirements for product produced in different runs.

Most computational experiments are performed on a Dell desktop with an Intel Xeon 3.2GHz CPU with 8 cores and 16 threads, and 32 GB of RAM, using Gurobi 8.1 in Python 3 on a Windows 10 operations system. The computational experiments in Section 6.2 are performed on a Dell desktop with an Intel Core i7-7820X 3.6GHz CPU with 8 cores and 16 threads, and 32 GB of RAM, using Gurobi 9.0.1 in Python 3 on a Windows 10 operations system.

6.1 Comparison with McCormick

To perform our computational tests, we obtained three years of demand data for the same system, and generated randomized supply data for those three years. We generated ten different such randomizations, and performed each computational test with five different starting dates (separated by half a year), obtaining a total of fifty test problems for each scheduling horizon.
Figure 5: Volume profiles with some bounds describing allowable ranges of spec and spec ratios.
6.1 Comparison with McCormick

NUMERICAL RESULTS

Figure 6: Example of time windows for barge arrivals and dates of runs on a 150 day time horizon.

Figure 7: Example of solution for schedules of loading to tanks and unloading (feed) from tanks to production for a 40 day time horizon.
In this section, we compare the effectiveness of two different methods for handling the product terms $v \cdot \Delta f$: McCormick and the proposed Center method, each using a base-2 binarization of the specs. The tests were repeated using several different values of $H$, and with several values for $\hat{\epsilon}$ (chosen to be the same for all specs). For $\hat{\epsilon} = 1$, we used $H \in \{10, 20, 30, 45, 70, 90\}$, and for $\hat{\epsilon} = 0.25$, we used $H \in \{10, 20, 30, 45, 60\}$ (since McCormick became computationally prohibitive over longer scheduling horizons). A total time limit of 600 seconds was enforced for all computations in this section. The average-case results are shown in Figure 8. In this section, we investigate performance in terms of computation time and percentage of value lost (%loss) in the objective, with respect to solution values $v_{\text{unused}}$ and $d_{\text{unmet}}$, computed via

\begin{align}
\text{val}_{\text{target}} &= \sum_{s \in S} c_{\text{inh}}^s \cdot y_s + \sum_{t \in T} c_{\text{dmd}}^t \cdot d_t \quad \text{(36a)} \\
\text{val}_{\text{missed}} &= \sum_{s \in S} c_{\text{inh}}^s \cdot v_{\text{unused}}^s + \sum_{t \in T} c_{\text{dmd}}^t \cdot d_{\text{unmet}}^t \quad \text{(36b)} \\
\%\text{loss} &= \frac{\text{val}_{\text{missed}}}{\text{val}_{\text{target}}} \cdot 100\% \quad \text{(36c)}
\end{align}

In all cases, we see that the proposed Center method performs much faster than McCormick, with a difference of more than an order of magnitude for longer scheduling horizons. However, this performance improvement came at a cost: as shown in Figure 8(b), the average optimality gap for $\hat{\epsilon} = 1$ yielded by the Center method is consistently quite a bit worse that that yielded by McCormick. As showcased in Figure 9(c), the majority of this difference seems to be caused by significantly worse solutions in roughly a quarter of the test cases. However, this disadvantage disappears at higher fidelity; with $\hat{\epsilon} = 0.25$, the optimality results are very close (well within Gurobi’s default optimality gap of $10^{-4}$), with the Center method taking a slight edge over McCormick. At the same time, the Center method proved to yield similar feasibility results as McCormick, with each becoming infeasible for no more than one out of the 50 trials in all test cases.

Figure 9 gives another perspective on the performance advantage of the proposed Center method, based on the number of instances solved by a given time for the longest scheduling horizons tested: for $\hat{\epsilon} = 1$ and $H = 90$, the advantage is overwhelming: almost all of the 50 test instances have converged for the Center method before any have converged for McCormick. The results for $\hat{\epsilon} = 0.25$, $H = 60$ also show a strong advantage for the proposed method, though this advantage is somewhat less pronounced due to the shorter scheduling horizon.

These results suggest that, for longer scheduling horizons, the new Center idea fares far better computationally than does the McCormick-style discretization in NMDT for the same level of precision.
6.1 Comparison with McCormick

Figure 8: Comparison of computational performance of the Center approximation vs. the McCormick relaxation. Averaged over 50 similar instances with different randomized supply parameters. (a) and (c): Computational time using step sizes of $\hat{\varepsilon} = 1$ and $0.25$. (b) and (d): % Loss computed as fraction of upper bound on objective value not obtained (see (36c)), where model accuracy is set to $\hat{\varepsilon} = 1$ and $0.25$. 
6.1 Comparison with McCormick

Figure 9: Time-completion (a,b) and % value missed (c) profiles for the comparison of computational performance for McCormick vs. the proposed approximate method, using the largest values of $H$ tested for (a,c) $\hat{\varepsilon} = 1$, and (b) $\hat{\varepsilon} = 0.25$.

Without rolling planning, the difference is far less pronounced: using 30-day and 40-day scheduling horizons with fidelity of $\hat{\varepsilon} = 1$, we find little difference between the performance of the two methods, as showcased in figure Figure 10. As a significant performance difference between the methods had already manifested by such small scheduling horizons in the cases with rolling planning, we find that, for these problem instances, the primary advantage with the Approx method lies in its interaction with rolling planning.
6.2 Comparison to Nonconvex MIQCP in Gurobi 9.0

In this section, we compare the computational performance of the methods presented, using $\varepsilon = 1$ and without rolling planning, to solving the models MINLP-Mix and MINLP-Split defined in Section ?? with Gurobi 9.0, using scheduling horizons of $H = 20$, $H = 30$, and $H = 40$. The MINLP-Mix model timed out (5-minute limit) for $H = 30$ on all instances but one, so we omit it from our comparisons beyond the first plot with $H = 20$. The results are shown in Figure 11. Note that the MINLP-Mix results are omitted from (b), as many instances also didn’t finish for $H = 20$.

The performance of MINLP-Mix is far inferior to that of MINLP-Split: For $H = 20$, $\sim 44$ instances finished for MINLP-Split within 50s, while only 10 had finished by that point for MINLP-Mix, with only 19 finishing within 5 minutes. For $H = 30$, no more than a single instance finish within 5 minutes for MINLP-Mix.

It is interesting to note that, in quite a few instances, Gurobi is able to solve the MINLP-Split model very quickly ($\sim 26$ for $H = 20$, $\sim 21$ for $H = 30$, and $\sim 15$ for $H = 40$). The results show that more instances get stuck with long solution times than with the NMDT-based options, with the slowest 40-60% of solutions slower than the NMDT-based methods. However, the solutions it returns are precise in terms of spec feasibility, while the level of solution precision $\varepsilon = 1$ for the discrete options is quite coarse. It seems likely that, using to $\varepsilon = 0.5$ or $\varepsilon = 0.25$, the Gurobi-based solver may overtake the NMDT-based variants in both solution time and quality for these methods. As a result, one might imagine that a rolling planning scheme based on this MINLP-split model, e.g. relaxing the bilinear constraint (19a) for the ‘future’ portion of the horizon, could be quite fruitful. We leave this exploration as future work.

The %loss results show that, even at such course fidelity, the approximation scheme without rolling planning consistently yields near-optimal results. Indeed, the relative difference between the solutions is consistently within the MIP-gap, 0.5%, used for the optimization.
Figure 11: Time-completion (a,c,e) and percent-loss (b,d,f) profiles for the comparison of computational performance for Gurobi 9.0.1 vs. the proposed NMDT-based methods, using $\varepsilon = 1$ tested for (a,b) $H = 20$, (c,d) $H = 30$, and (e,f) $H = 40$. 
6.3 Performance Enhancement

We explore how to enhance the performance of the method via two simple changes to the Center model that preserve the mixed-integer feasible solutions. The first of these is to add the coupling constraints described in Section 5.2.2

\[ \nu_{i,j}^{\alpha,\text{mid}} = \nu_{i,j}^{\alpha,\text{end}} + \gamma_{i,j}^{\alpha,\text{out}} \quad i \in I_{k,q,t}, j \in \{0, 1\} \]  

(37a)

where \( I_{k,q,t} \) is defined as in Section 5.3.1. We then remove constraints (31) and (31b), since they are now made redundant in the associated LP by (37a) combined with the nonnegativity of \( \nu_{p,i,1}^{\alpha,\text{mid}}, \nu_{p,i,1}^{\alpha,\text{end}}, \) and \( \gamma_{p,i,1}^{\alpha,\text{out}} \).

The second of these is to relax the MILP approximation of the \( \nu_{p,i,j}^{\alpha,\text{mid}} \) terms. This corresponds then removing constraints (31a), (31b), and the constraints

\[ \nu_{i,j}^{\alpha,\text{end}} \geq \Sigma_k \min \cdot \alpha_{i,j} \quad i \in I_{k,q,t}, j \in \{0, 1\}, \]  

(38a)

\[ \gamma_{i,j}^{\alpha,\text{out}} \leq \delta_t \cdot \alpha_{i,j} \quad i \in I_{k,q,t}, j \in \{0, 1\}, \]  

(38b)

which are part of the representation of (30a) and (30c). The viability of the removal of (38b) is shown in Section 5.2.2 since any one of the three associated variables \( \nu_{p,i,1}^{\alpha,\text{mid}}, \nu_{p,i,1}^{\alpha,\text{end}}, \) and \( \gamma_{p,i,1}^{\alpha,\text{out}} \) can be eliminated completely without compromising the MILP, a valid model is still obtained when relaxing any number of constraints related to the definition of \( \gamma_{p,i,1}^{\alpha,\text{out}} \). Constraint (38a) is always safe to remove, as it is implied by the fact that only one \( \nu_{i,j}^{\alpha,\text{mid}} \) variable can nonzero for each \( i \), combined with the fact that \( \nu_{i,j}^{\alpha,\text{mid}} \) sum to \( \nu_{k,t}^{\alpha,\text{mid}} \), which is bounded below by \( \Sigma_k \min \). Note that either (31b) or (38b) must be included if the coupling constraints (37a) are not included. In this case, we choose to remove (38b).

The results, displayed in Figure 12, show that while both modeling changes make a large difference in computational cost, the simple relaxation of the MILP yields a larger improvement in computational time. This improvement is increased further when the coupling constraints are added, yielding a method in which the majority of instances finish within the desired 10 minutes when using a scheduling horizon of a full year with \( \hat{\varepsilon} = 1 \).
6.4 Comparison of Rolling Planning Ideas

In this section, we compare the computational performance of different rolling planning regimes. We restrict ourselves here to the full-horizon scheme, and compare the performance obtained using two different ideas. For the first, we use fixed-length periods with $H^{\text{int}} = 7$. For the second, we use run-based periods using $\tilde{H}^{\text{int}} = 4$ to generate the periods, with $n^H = \Delta n = 2$ as our stepping parameters. We compare the methods using a time limit of 1800s using $\hat{\varepsilon} = 1$ and scheduling horizons of $H \in \{60, 75, 90, 129, 180, 365\}$ (in days).

The results, as shown in Figure 13, indicate that the regime with a fixed step size yields a more consistent, and often lower, optimality gap on average while incurring a higher computational cost. This outcome is emphasized in Figure 14: when $H = 365$, the 7-1-1 fixed-step scheme results in nearly all test cases finishing within about a 5-10 minute time window, while incurring high optimality gaps in a handful of instances. On the other hand, for the 4-2-2 run-dependent-step scheme, most instances finish within about a 10-20 minute time window, but optimality gap for the worst instances is far more controlled.

Figure 12: Time-completion profiles using different combinations of relaxing (constraints) and adding (coupling constraint), with $H = 365$ and a 7-1-1 fixed-step-size rolling planning scheme.
Figure 13: Comparison of fixed-step-length and run-dependent-step rolling planning methods. Averaged over 50 similar instances with different randomized supply parameters, using spec step size $\hat{\epsilon} = 1$. (a): Computational cost, (b) Optimality gap.

Figure 14: Time-completion profiles of fixed-step-length and run-dependent-step rolling planning methods, using $H = 365$ days and $\hat{\epsilon} = 1$. Run using both the relaxation and coupling things. (a) Computational cost, (b) Optimality gap.
7 Conclusion

In this work, we have developed an approximation method for the tank mixing and scheduling problem that combines rolling horizon planning with two different discretization schemes, yielding reasonably high-quality solutions very quickly over long scheduling horizons, while with high probability ensuring that demand specs remain within the required ranges. Due to this performance, the model could be especially useful for obtaining rough, fast plan feasibility results during the planning of supply acquisitions and product production runs in an industry setting (counting solutions with no (or only small) supply and demand inventory misses as ‘feasible’).

Comparing the in-house “Center” discretization scheme to the original “McCormick”-based NMDT scheme [4], we find that the in-house method yields much faster performance when combined with a rolling horizon scheme, while sometimes yielding worse solutions. However, without a rolling horizon scheme, the methodologies yield comparable performance, suggesting that the “Center” scheme may yield relaxations more compatible with a rolling horizon scheme. Further, when comparing rolling horizon schemes, we find that a scheme using run-based rolling horizon periods yields better solutions than a fixed-duration scheme, at some cost to performance. To extend this work, we plan to explicitly incorporate acquisition planning of barges, and possibly planning of production runs.

References

[1] Maria Lorena Bergamini, Pio Aguirre, and Ignacio Grossmann. Logic-based outer approximation for globally optimal synthesis of process networks. Computers & Chemical Engineering, 29(9):1914–1933, aug 2005.

[2] Pedro Castillo Castillo, Pedro M. Castro, and Vladimir Mahalec. Global optimization algorithm for large-scale refinery planning models with bilinear terms. Industrial & Engineering Chemistry Research, 56(2):530–548, January 2017.

[3] Pedro M. Castro. New MINLP formulation for the multiperiod pooling problem. AIChe Journal, 61(11):3728–3738, sep 2015.

[4] Pedro M. Castro. Normalized multiparametric disaggregation: an efficient relaxation for mixed-integer bilinear problems. Journal of Global Optimization, 64(4):765–784, jul 2015.

[5] Pedro M. Castro. Tightening piecewise McCormick relaxations for bilinear problems. Computers & Chemical Engineering, 72:300–311, jan 2015.

[6] Pedro M. Castro. Source-based discrete and continuous-time formulations for the crude oil pooling problem. Computers & Chemical Engineering, 93:382–401, oct 2016.
[7] Pedro M. Castro and Ignacio E. Grossmann. Global optimal scheduling of crude oil blending operations with RTN continuous-time and multiparametric disaggregation. *Industrial & Engineering Chemistry Research*, 53(39):15127–15145, sep 2014.

[8] Pedro M. Castro and Ignacio E. Grossmann. Optimality-based bound contraction with multiparametric disaggregation for the global optimization of mixed-integer bilinear problems. *Journal of Global Optimization*, 59(2-3):277–306, mar 2014.

[9] Jaime Cerdá, Pedro C. Pautasso, and Diego C. Cafaro. Efficient approach for scheduling crude oil operations in marine-access refineries. *Industrial & Engineering Chemistry Research*, 54(33):8219–8238, aug 2015.

[10] Billy R. Champion and Steven A. Gabriel. A multistage stochastic energy model with endogenous probabilities and a rolling horizon. *Energy and Buildings*, 135:338–349, jan 2017.

[11] Xuan Chen, Simin Huang, Dejing Chen, Zhihai Zhang, Li Zheng, Ignacio Grossmann, and Shihui Chen. Hierarchical decomposition approach for crude oil scheduling: A SINOPEC case. *Interfaces*, 44(3):269–285, jun 2014.

[12] Yunfei Chu, Fengqi You, John M. Wassick, and Anshul Agarwal. Integrated planning and scheduling under production uncertainties: Bi-level model formulation and hybrid solution method. *Computers & Chemical Engineering*, 72:255–272, January 2015.

[13] Cao Cuiwen, Gu Xingsheng, and Xin Zhong. A data-driven rolling-horizon online scheduling model for diesel production of a real-world refinery. *AIChE Journal*, 59(4):1160–1174, sep 2012.

[14] Leonardo Salsano de Assis, Eduardo Camponogara, Bernardo Zimberg, Enrique Ferreira, and Ignacio E. Grossmann. A piecewise McCormick relaxation-based strategy for scheduling operations in a crude oil terminal. *Computers & Chemical Engineering*, 106:309–321, nov 2017.

[15] Lisia S. Dias and Marianthi G. Ierapetritou. Integration of scheduling and control under uncertainties: Review and challenges. *Chemical Engineering Research and Design*, 116:98–113, December 2016.

[16] Xiaoyong Gao, Yongheng Jiang, Tao Chen, and Dexian Huang. Optimizing scheduling of refinery operations based on piecewise linear models. *Computers & Chemical Engineering*, 75:105–119, apr 2015.

[17] Chrysanthos E. Goumaris, Ruth Misener, and Christodoulos A. Floudas. Computational comparison of piecewise-linear relaxations for pooling problems. *Industrial & Engineering Chemistry Research*, 48(12):5742–5766, June 2009.

[18] Ignacio E. Grossmann and Juan P. Ruiz. Generalized disjunctive programming: A framework for formulation and alternative algorithms for MINLP optimization. In *Mixed Integer Nonlinear Programming*, pages 93–115. Springer New York, November 2011.

[19] Vijay Gupta and Ignacio E. Grossmann. Multistage stochastic programming approach for offshore oilfield infrastructure planning under production sharing agreements and endogenous uncertainties. *Journal of Petroleum Science and Engineering*, 124:180–197, December 2014.
[20] Akshay Gupte, Shabbir Ahmed, Myun Seok Cheon, and Santanu Dey. Solving mixed integer bilinear problems using MILP formulations. *SIAM Journal on Optimization*, 23(2):721–744, January 2013.

[21] Akshay Gupte, Shabbir Ahmed, Santanu S. Dey, and Myun Seok Cheon. Relaxations and discretizations for the pooling problem. *Journal of Global Optimization*, 67(3):631–669, apr 2016.

[22] Miguel Angel Gutiérrez-Limón, Antonio Flores-Tlacuahuac, and Ignacio E. Grossmann. MINLP formulation for simultaneous planning, scheduling, and control of short-period single-unit processing systems. *Industrial & Engineering Chemistry Research*, 53(38):14679–14694, September 2014.

[23] Miguel Angel Gutiérrez-Limón, Antonio Flores-Tlacuahuac, and Ignacio E. Grossmann. A reactive optimization strategy for the simultaneous planning, scheduling and control of short-period continuous reactors. *Computers & Chemical Engineering*, 84:507–515, January 2016.

[24] Michael Wagner J. Rohde, Herbert Meyr. Die supply chain planning matrix. *PPS Management*, 5(1), jan 2000.

[25] M. Joly and J.M. Pinto. Mixed-integer programming techniques for the scheduling of fuel oil and asphalt production. *Chemical Engineering Research and Design*, 81(4):427–447, apr 2003.

[26] Ramkumar Karuppiah, Kevin C. Furman, and Ignacio E. Grossmann. Global optimization for scheduling refinery crude oil operations. *Computers & Chemical Engineering*, 32(11):2745–2766, nov 2008.

[27] J. D. Kelly and J. L. Mann. Crude oil blend scheduling optimization: an application with multimillion dollar benefits - part 1. *Hydrocarbon Processing, International Edition*, 82(6):47–53, jun 2003.

[28] Scott Kolodziej, Pedro M. Castro, and Ignacio E. Grossmann. Global optimization of bilinear programs with a multiparametric disaggregation technique. *Journal of Global Optimization*, 57(4):1039–1063, jan 2013.

[29] Scott P. Kolodziej, Ignacio E. Grossmann, Kevin C. Furman, and Nicolas W. Sawaya. A novel global optimization approach to the multiperiod blending problem. In *Computer Aided Chemical Engineering*, pages 1492–1496. Elsevier, 2012.

[30] Scott P. Kolodziej, Ignacio E. Grossmann, Kevin C. Furman, and Nicolas W. Sawaya. A discretization-based approach for the optimization of the multiperiod blend scheduling problem. *Computers & Chemical Engineering*, 53:122–142, jun 2013.

[31] Heeman Lee, Jose M. Pinto, Ignacio E. Grossmann, and Sunwon Park. Mixed-integer linear programming model for refinery short-term scheduling of crude oil unloading with inventory management. *Industrial & Engineering Chemistry Research*, 35(5):1630–1641, January 1996.

[32] Jie Li and I. A. Karimi. Scheduling gasoline blending operations from recipe determination to shipping using unit slots. *Industrial & Engineering Chemistry Research*, 50(15):9156–9174, August 2011.

[33] Jie Li, I. A. Karimi, and Rajagopalan Srinivasan. Recipe determination and scheduling of gasoline blending operations. *AIChE Journal*, pages NA–NA, 2009.
[34] Jie Li, Wenkai Li, I. A. Karimi, and Rajagopalan Srinivasan. Improving the robustness and efficiency of crude scheduling algorithms. *AIChE Journal*, 53(10):2659–2680, 2007.

[35] Zukui Li and Marianthi G. Ierapetritou. Rolling horizon based planning and scheduling integration with production capacity consideration. *Chemical Engineering Science*, 65(22):5887–5900, Nov 2010.

[36] Irene Lotero, Francisco Trespalacios, Ignacio E. Grossmann, Dimitri J. Papageorgiou, and Myun-Seok Cheon. An MILP-MINLP decomposition method for the global optimization of a source based model of the multiperiod blending problem. *Computers & Chemical Engineering*, 87:13–35, Apr 2016.

[37] Shan Lu, Hongye Su, Yue Wang, Lei Xie, and Quanling Zhang. Multi-product multi-stage production planning with lead time on a rolling horizon basis. *IFAC-PapersOnLine*, 48(8):1162–1167, 2015.

[38] Chunpeng Luo and Gang Rong. A strategy for the integration of production planning and scheduling in refineries under uncertainty. *Chinese Journal of Chemical Engineering*, 17(1):113–127, Feb 2009.

[39] Julien F. Marquant, Ralph Evins, and Jan Carmeliet. Reducing computation time with a rolling horizon approach applied to a MILP formulation of multiple urban energy hub system. *Procedia Computer Science*, 51:2137–2146, 2015.

[40] Garth P. McCormick. Computability of global solutions to factorable nonconvex programs: Part i — convex underestimating problems. *Mathematical Programming*, 10(1):147–175, Dec 1976.

[41] Carlos A. Méndez, Ignacio E. Grossmann, Iiro Harjunkoski, and Pousga Kaboré. A simultaneous optimization approach for off-line blending and scheduling of oil-refinery operations. *Computers & Chemical Engineering*, 30(4):614–634, Feb 2006.

[42] Ruth Misener and Christodoulos A Floudas. Advances for the pooling problem: Modeling, global optimization, and computational studies. *Applied and Computational Mathematics*, 8(1):3–22, 2009.

[43] Ruth Misener, Jeffrey P. Thompson, and Christodoulos A. Floudas. APOGEE: Global optimization of standard, generalized, and extended pooling problems via linear and logarithmic partitioning schemes. *Computers & Chemical Engineering*, 35(5):876–892, May 2011.

[44] Lincoln F. L. Moro and José M. Pinto. Mixed-integer programming approach for short-term crude oil scheduling. *Industrial & Engineering Chemistry Research*, 43(1):85–94, Jan 2004.

[45] Sylvain Mouret. *Optimal Scheduling of Refinery Crude-Oil Operations*. PhD thesis, Carnegie Mellon University, Mar 2010.

[46] Sylvain Mouret, Ignacio E. Grossmann, and Pierre Pestiaux. A novel priority-slot based continuous-time formulation for crude-oil scheduling problems. *Industrial & Engineering Chemistry Research*, 48(18):8515–8528, Sep 2009.

[47] Sylvain Mouret, Ignacio E. Grossmann, and Pierre Pestiaux. Tightening the linear relaxation of a mixed integer nonlinear program using constraint programming. In *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pages 208–222. Springer Berlin Heidelberg, 2009.
Sergio Neiro and Jose Pinto. Lagrangean decomposition applied to multiperiod planning of petroleum refineries under uncertainty. *Latin American Applied Research*, 36:213–220, 10 2006.

Chao Ning and Fengqi You. A data-driven multistage adaptive robust optimization framework for planning and scheduling under uncertainty. *AIChE Journal*, 63(10):4343–4369, May 2017.

Thordis Anna Oddsdottir, Martin Grunow, and Renzo Akkerman. Procurement planning in oil refining industries considering blending operations. *Computers & Chemical Engineering*, 58:1–13, nov 2013.

Viet Pham, Carl Laird, and Mahmoud El-Halwagi. Convex hull discretization approach to the global optimization of pooling problems. *Industrial & Engineering Chemistry Research*, 48(4):1973–1979, February 2009.

J.M. Pinto and L.F. L. Moro. A planning model for petroleum refineries. *Brazilian Journal of Chemical Engineering*, 17(4-7):575–586, December 2000.

P. Chandra Prakash Reddy, I. A. Karimi, and R. Srinivasan. Novel solution approach for optimizing crude oil operations. *AIChE Journal*, 50(6):1177–1197, 2004.

P. Chandra Prakash Reddy, I.A. Karimi, and R. Srinivasan. A new continuous-time formulation for scheduling crude oil operations. *Chemical Engineering Science*, 59(6):1325–1341, mar 2004.

Juan P. Ruiz and Ignacio E. Grossmann. Strengthening of lower bounds in the global optimization of bilinear and concave generalized disjunctive programs. *Computers & Chemical Engineering*, 34(6):914–930, June 2010.

Javier Silvente, Georgios M. Kopanos, Vivek Dua, and Lazaros G. Papageorgiou. A rolling horizon approach for optimal management of microgrids under stochastic uncertainty. *Chemical Engineering Research and Design*, 131:293–317, mar 2018.

Javier Silvente, Georgios M. Kopanos, Efstratios N. Pistikopoulos, and Antonio Espuña. A rolling horizon optimization framework for the simultaneous energy supply and demand planning in microgrids. *Applied Energy*, 155:485–501, oct 2015.

A. Singh, J.F. Forbes, P.J. Vermeer, and S.S. Woo. Model-based real-time optimization of automotive gasoline blending operations. *Journal of Process Control*, 10(1):43–58, February 2000.

Raik Stolletz and Emilio Zamorano de Acha. A rolling planning horizon heuristic for scheduling agents with different qualifications. *SSRN Electronic Journal*, 2014.

João P. Teles, Pedro M. Castro, and Henrique A. Matos. Multi-parametric disaggregation technique for global optimization of polynomial programming problems. *Journal of Global Optimization*, 55(2):227–251, nov 2011.

S. Torkaman, S.M.T Fatemi Ghomi, and B. Karimi. Multi-stage multi-product multi-period production planning with sequence-dependent setups in closed-loop supply chain. *Computers & Industrial Engineering*, 113:602–613, nov 2017.
[62] Juan Pablo Vielma and George L. Nemhauser. Modeling disjunctive constraints with a logarithmic number of binary variables and constraints. *Mathematical Programming*, 128(1-2):49–72, July 2009.

[63] Li Wenkai, Chi-Wai Hui, Ben Hua, and Zhongxuan Tong. Scheduling crude oil unloading, storage, and processing. *Industrial & Engineering Chemistry Research*, 41(26):6723–6734, Dec 2002.

[64] Jialin Xu, Shujing Zhang, Jian Zhang, Sujing Wang, and Qiang Xu. Simultaneous scheduling of front-end crude transfer and refinery processing. *Computers & Chemical Engineering*, 96:212–236, Jan 2017.

[65] Yale Zhang, Dayadeep Monder, and J. Fraser Forbes. Real-time optimization under parametric uncertainty: a probability constrained approach. *Journal of Process Control*, 12(3):373–389, April 2002.

[66] Edwin Zondervan, Michiel Kaland, Martijn A.H. van Elzakker, Jan C. Fransoo, and Jan Meuldijk. Integrating planning and scheduling in an oil refinery with a rolling horizon approach. In *Computer Aided Chemical Engineering*, volume 33, pages 439–444. Elsevier, 2014.