AIChE

Continuous-Time Modeling and Global Optimization Approach for Scheduling of Crude Oil Operations

Jie Li, Ruth Misener and Christodoulos A. Floudas
Dept. of Chemical and Biological Engineering, Princeton University, Princeton, NJ 08544

DOI 10.1002/aic.12623
Published online May 4, 2011 in Wiley Online Library (wileyonlinelibrary.com).

Scheduling of crude oil operations is a critical and complicated component of overall refinery operations, because crude oil costs account for about 80% of the refinery turnover. Moreover, blending with less expensive crudes can significantly increase profit margins. The mathematical modeling of blending different crudes in storage tanks results in many bilinear terms, which transforms the problem into a challenging, nonconvex, and mixed-integer nonlinear programming (MINLP) optimization model. Two primary contributions have been made. First, the authors developed a novel unit-specific event-based continuous-time MINLP formulation for this problem. Then they incorporated realistic operational features such as single buoy mooring (SBM), multiple jetties, multiparcel vessels, single-parcel vessels, crude blending, brine settling, crude segregation, and multiple tanks feeding one crude distillation unit at one time and vice versa. *In addition, 15 important volume-based or weight-based crude property indices are also considered. Second, they exploited recent advances in piecewise-linear underestimation of bilinear terms within a branch-and-bound algorithm to globally optimize the MINLP problem. It is shown that the continuous-time model results in substantially fewer bilinear terms. Several examples taken from the work of Li et al. are used to illustrate that (1) better solutions are obtained and (2) ε-global optimality can be attained using the proposed branch-and-bound global optimization algorithm with piecewise-linear underestimations of the bilinear terms.* © 2011 American Institute of Chemical Engineers *AIChE J,* 58: 205–226, 2012
Keywords: refinery, crude oil scheduling, mixed-integer nonlinear programming, nonconvex, global optimization, piecewise linear, branch and bound

Introduction

In recent years, the petroleum industry faces intense competition compounded by fluctuating product demands, ever-changing crude prices, and strict environmental regulations. As a result, the refiners have to exploit all potential cost-saving opportunities to survive successfully. Since 1950s, oil refineries have used optimization techniques for planning refinery operations.

Scheduling of crude oil operations is of major importance in refinery operations,[1,2] because crude oil costs account for about 80% of the refinery turnover.[3] Crude oils vary significantly in composition, product yields, properties, and prices. Premium crudes sell roughly $15 per barrel higher than low-quality crudes.[4] Therefore, most refineries blend premium crudes with low-quality crudes over time to exploit the higher profit margins of low-quality crudes. However, the low-cost crudes contain some less-than-desirable properties

Correspondence concerning this article should be addressed to C. A. Floudas at floudas@titan.princeton.edu.

with high composition, such as sulfur, aromatics, and residue, and can lead to processing problems in crude distillation units (CDUs) and downstream units. Therefore, a key issue is to identify and process optimal blends of low-cost crudes and premium crudes to minimize the operational problems while maximizing profit margins. Scheduling of crude oil operations using advanced optimization techniques such as mixed-integer linear programming (MILP) can increase profits by using cheaper crudes, minimizing crude changeovers, avoiding ship demurrage, and managing crude inventory optimally. However, the mathematical modeling of the blending of different crudes in storage tanks results in many bilinear terms, which transforms the problem into a difficult, nonconvex, and mixed-integer nonlinear programming (MINLP) problem.

The crude oil scheduling problem has received considerable attention with researchers developing different models based on discrete-time[5–8] and continuous-time representations.[9–14] Although most models[6,9,11,12,14] only considered a single jetty, Li et al.[7] incorporated multiple jetties. All of these models[5–7,9,11,12,14] allowed one tank feeding one CDU at a time. Pan et al.[13] and Reddy et al.[8,10] included a single buoy mooring (SBM) station, multiple-parcel vessels, brine settling, crude segregation, and multiple tanks feeding one CDU at a time and *vice versa*. Besides all operational features of Reddy et al.[10] and Pan et al.,[13] Reddy et al.[8] incorporated multiple jetties. All of the above models considered at most two key components that are linearly additive. Li et al.[4] extended the model of Reddy et al.[8] for 15 important volume-based and weight-based crude property indices. Therefore, it is important to note that there are no continuous-time models that incorporate all of the aforementioned realistic features, especially multiple jetties.

The presence of crude blending gives rise to bilinear terms in the mathematical formulation for scheduling, whereas discrete scheduling decisions such as selecting a tank to unload and the often complex nonlinear nature of crude properties and qualities make such a model challenging, nonlinear, and nonconvex MINLP. As shown by Li et al.,[4] even the best existing commercial solvers are unable to solve these scheduling problems of practical industrial size.[4] Hence, several researchers have developed some special algorithms such as linearization approach,[6] the iterative decomposition algorithm,[7] discretization procedure,[11] rolling horizon algorithm,[8,10] and the improved rolling horizon algorithm with intelligent backtracking and partial relaxation strategies.[4] The linearization approach[6] led to composition discrepancy, in which the amount of individual crudes delivered from a tank to CDU are not proportional to the crude composition in the tank as shown by Li et al.[7] and Reddy et al.[8] The iterative decomposition algorithm[7] and rolling horizon algorithm[8] may fail to find feasible schedules although feasible solutions do exist as claimed by Reddy et al.[8] and Li et al.,[4] respectively. The discretization procedure[11] not only gets approximate optimal solutions but also increases problem size to an extent that makes it almost impossible to solve reasonably sized problems. Although the improved rolling algorithm with intelligent backtracking and partial relaxation strategies[4] solved all tested 20 examples, it cannot guarantee global optimality. Karuppiah et al.[12] developed an outer approximation algorithm to globally optimize crude oil scheduling operations. They generated cutting planes from spatial decomposition of the crude oil network and added them to the MILP relaxation from McCormick convex and concave envelope[15] to reduce computational time. However, their outer approximation algorithm is proposed for inland refineries, which have both storage and charging tanks, and are difficult to extend to marine-access refineries, which is addressed in this article. In future work, we plan to extend the proposed continuous-time model for inland refineries and then compare the proposed branch-and-bound global optimization algorithm with the outer approximation algorithm.

Recently, global optimization algorithms using piecewise-linear relaxations have been greatly advanced. Meyer and Floudas[16] and Karuppiah and Grossmann[17] noticed that the piecewise-linear relaxation of convex envelopes can tighten the linear relaxation, thus usually leading to fewer nodes needed in a branch-and-bound tree. They used the piecewise-linear relaxation to tightly underestimate large-scale wastewater treatment and water networks problems. Wicaksono and Karimi[18] developed 15 mathematically equivalent alternative formulations and compared their performance on several test cases. Gounaris et al.[19] recently proposed five additional piecewise-linear formulations and comprehensively compared their computational performance using benchmark pooling problems. Bergamini et al.,[20] Saif et al.,[21] Pham et al.,[22] Misener and Floudas,[23,24] Hasan and Karimi,[25] and Misener et al.[26] also used piecewise-linear underestimators to tighten the relaxation of bilinear items for process systems applications.

In this article, we address the crude oil scheduling problem described by Li et al.[4] for a typical marine-access refinery. The contributions are threefold: (1) a new continuous time model; (2) piecewise linear underestimation of the bilinear terms; and (3) a global optimization algorithm. We developed a novel unit-specific event-based continuous-time MINLP formulation for this problem. We incorporated all aforementioned realistic operational features such as SBM, multiple jetties, multiparcel vessels, single-parcel vessels, crude blending, brine settling, crude segregation, and multiple tanks feeding one CDU at one time and *vice versa*. In addition, we also considered 15 important volume-based or weight-based crude property indices. We exploited recent advances in piecewise-linear underestimation of bilinear terms[16–26] within a branch-and-bound algorithm for global optimization. Several examples from Li et al.[4] are used to illustrate the capability to reach global optimality of the proposed branch-and-bound global optimization algorithm with piecewise-linear underestimation.

Problem Statement

Consider Figure 1 in which a schematic of crude oil unloading, storage, and processing in a typical marine-access refinery is shown. It involves offshore facilities for crude unloading such as a SBM station, onshore facilities for crude unloading such as B jetties, tank farm consisting of I ($i = 1, 2, 3, \ldots, I$) crude storage, and/or charging tanks, and U ($u = 1, 2, 3, \ldots, U$) CDUs. Different types of crudes C ($c = 1, 2, 3, \ldots, C$) can be allowed to blend in these crude storage tanks. Thus, the crude compositions in crude tanks vary with time. In this work, we assume that the refinery has no

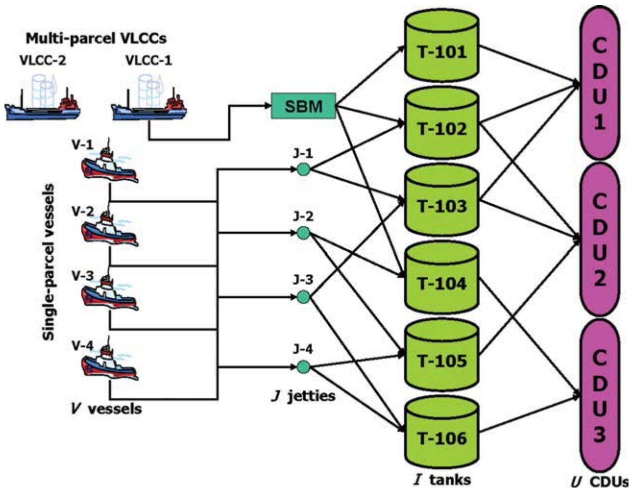

Figure 1. Schematic of crude oil unloading, blending, and processing.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

separate charging tanks, and hence crude storage tanks also act as charging tanks. Pipelines are employed to connect unloading facilities to storage tanks. The pipeline connecting the SBM station with crude tanks is called the SBM line, and it normally has a substantial holdup.

Crudes are supplied to the refinery in either large multi-parcel tankers or small single-parcel vessels ($v = 1, 2, 3, \ldots, V$). Very large crude carriers (VLCCs) or ultra large crude carriers carry multiple parcels of several crudes and dock at the SBM station offshore. Because of their huge size, they cannot dock at jetties. Small vessels carry single crudes and berth at the jetties. The entire operation involves unloading crudes from ships into various storage tanks at various times, blending different crudes in storage tanks, and feeding CDUs from one or more storage tanks at various rates over time. The entire problem can be stated as follows:

Given:

1. V ships, their expected arrival times, their crude parcels, and parcel sizes.
2. B Jetties, jetty-tank and SBM-tank connections, crude unloading transfer rates, and SBM pipeline holdup volume and its resident crude.
3. I storage tanks, their capacities, their initial crude volumes and compositions, and crude quality specifications or limits.
4. U CDUs, their processing rates, and crude quality specifications or limits.
5. Scheduling horizon H and product demands.
6. Economic data: crude margins, demurrage, crude changeover costs, and safety stock penalties.

Determine:

1. Unloading schedule for each ship including the timings, rates, and tanks for all parcel transfers.
2. Inventory and crude concentration profiles of all storage tanks.
3. Charging schedule for each CDU including the feed tanks, feed rates, and timings.

Subject to the operating practices:

1. Only one VLCC can dock at the SBM station at any time.
2. The sequence in which a VLCC unloads its parcels is known *a priori*. This is normally fixed when the VLCC loads its parcels and the refinery needs to specify that at the time of shipping.
3. A parcel can unload to only one storage tank at any moment, but may unload to multiple tanks over time.
4. A storage tank cannot receive and output crude at the same time.
5. Each tank needs 8 h to settle and remove brine after each crude receipt.
6. Multiple tanks can feed a CDU simultaneously and *vice versa*.

Assumptions:

1. All parameters are deterministic.
2. Holdup of the SBM line is far smaller than a typical parcel size. Thus, only one crude resides in the SBM line at the end of each parcel transfer. Crude flow is plug flow in the SBM.
3. Holdup of the jetty pipeline is negligible.
4. Crude mixing is perfect in each storage tank.
5. Crude changeover times are negligible.
6. All jetties are identical.
7. During operation, CDUs never shut down.

The objective of the crude oil scheduling problem is to maximize the gross profit, which is the revenue computed in terms of crude margins minus the operating costs such as demurrage and safety stock penalties.

To address this class of problems, we developed a unit-specific event-based continuous-time MILP formulation. The advantages of using unit-specific event-based approaches are well established in the literature.[27–46] Most importantly, it can significantly reduce the bilinear terms, which determine the complexity of the nonconvex model.

Mathematical Formulation

We define jetties, storage tanks (i), and CDUs (u) as units (m). As all jetties are identical, we follow the approach of Li et al.[47] and treat them as one single resource. For each unit m, we divide the scheduling horizon [0, H] into N ($n = 1, 2, \ldots, N$) event points (Figure 2). Let T_M^{start} (m,n) and T_M^{end} (m,n) [T_M^{end} $(m,n) \geq T_M^{\text{start}}$ (m,n)] denote the start and end times of event point n on unit m, where m becomes i for a storage tank, and u for a CDU. The location of event points are different for different units, and therefore, T_M^{start} (m,n) and

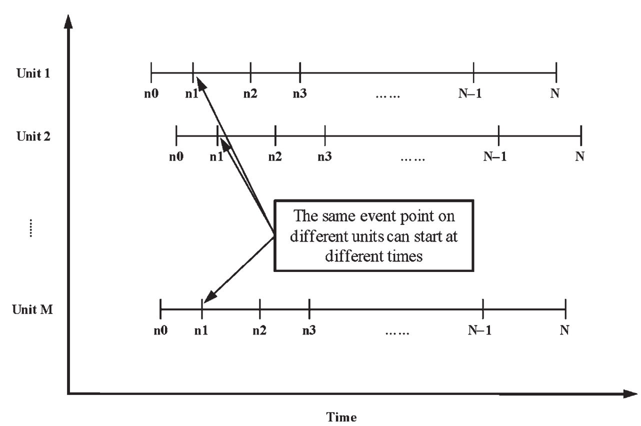

Figure 2. Event points definition for each unit.

T_M^{end} (m,n) may vary with unit and are fully or partially independent from each other.

The event point $(n + 1)$ on unit m must start after the event point n on this unit m ends. Thus

$$T_B^{\text{start}}(n+1) \geq T_B^{\text{end}}(n) \qquad \forall n \qquad (1a)$$

$$T_I^{\text{start}}(i,n+1) \geq T_I^{\text{end}}(i,n) \qquad \forall i,n \qquad (1b)$$

$$T_U^{\text{start}}(u,n+1) \geq T_U^{\text{end}}(u,n) \qquad \forall u,n \qquad (1c)$$

It should be noted that we follow the same approach of parcel creation from Reddy et al.,[8] in which all arriving multiparcel VLCCs were divided into individual single-crude parcels and the significant holdup of the SBM line was treated as a distinct single-crude parcel. Thus, we have P ($p = 1, 2, 3, \ldots, P$) parcels. The parcels unloaded via SBM station are called as VLCC parcels and those via jetties are called as jetty parcels. We use sets **VP** and **JP** to denote all VLCC parcels and jetty parcels, respectively.

Because a parcel p is allowed to be completed in a single event point or several consecutive event points, we use a parameter Δn as follows to denote if a parcel is unloaded in a single event point.

$$\Delta n = \begin{cases} 1 & \text{if a parcel is unloaded in multipe event points} \\ 0 & \text{otherwise} \end{cases}$$

Parcel unloading

To model a parcel p to a tank i connection, we define a binary variable $X(p, i, n)$ as follows

$$X(p,i,n) = \begin{cases} 1 & \text{if parcel } p \text{ is unloaded to tank } i \text{ during event point } n \\ 0 & \text{otherwise} \end{cases} \qquad \forall (p,i) \in \mathbf{S}_{P,I}$$

where $\mathbf{S}_{P,I} = \{(p, i) \mid \text{parcel } p \text{ that can be unloaded to tank } i\}$.

If a parcel p must be completed in a single event point, that is, $\Delta n = 0$, then it must be unloaded exactly one time during the scheduling horizon.

$$\sum_{i:(p,i)\in \mathbf{S}_{PI}} \sum_n X(p,i,n) = 1 \qquad \forall p, \Delta n = 0 \qquad (2a)$$

For the case in which a parcel p is allowed to be unloaded in multiple event points, that is $\Delta n = 1$, we impose that this parcel p must be unloaded into at most one storage tank at each event point.

$$\sum_{i:(p,i)\in \mathbf{S}_{P,I}} X(p,i,n) \leq 1 \qquad \forall p, n, \Delta n = 1 \qquad (2b)$$

At each time, a tank i can receive at most B ($B > 1$) jetty parcels.

$$\sum_{\substack{p:(p,i)\in \mathbf{S}_{P,I} \\ p \in \mathbf{JP}}} X(p,i,n) \leq B \qquad \forall i, n, B > 1 \qquad (3)$$

Note that Eq. 3 is only applied when there is more than one jetty in a refinery. From Eqs. 2 and 3, it can be concluded that we do not enforce a tank i to receive at most one VLCC parcel at each event point. In other words, a tank i is allowed to receive more than one VLCC parcel at each event point, but at different times. Later, we impose some constraints (i.e., Eqs. 9 and 10) to ensure that different VLCC parcels are unloaded to the same tank one after another, although they may be unloaded at the same event point. This strategy is also used when only one jetty is involved. This can further reduce the number of event points and hence binary variables and bilinear terms.

If a parcel p is allowed to be unloaded in multiple event points, then it must be unloaded in consecutive event points. We define a 0–1 continuous variable $xe(p,n)$ as follows

$$xe(p,n) = \begin{cases} 1 & \text{if parcel } p \text{ is completed at the end of event point } n \\ 0 & \text{otherwise} \end{cases} \qquad \forall p, n$$

If a parcel p is unloaded at events n and $n + 1$, it cannot be completed at the end of event n.

$$xe(p,n) \leq 2 - \sum_{i:(p,i)\in \mathbf{S}_{P,I}} X(p,i,n) - \sum_{i:(p,i)\in \mathbf{S}_{P,I}} X(p,i,n+1) \qquad \forall p, n<N, \Delta n = 1 \qquad (4a)$$

Similarly, when a parcel p is unloaded at event point n, but not at event point $n + 1$, then it must be completed at the end of event point n.

$$xe(p,n) \geq \sum_{i:(p,i)\in \mathbf{S}_{P,I}} X(p,i,n) - \sum_{i:(p,i)\in \mathbf{S}_{P,I}} X(p,i,n+1) \qquad \forall p, n<N, \Delta n = 1 \qquad (4b)$$

All parcels must be completed within the scheduling horizon. Thus, if a parcel p is unloaded during the last event point N, then it must be completed at the end of this last event point.

$$xe(p,n) \geq \sum_{i:(p,i)\in \mathbf{S}_{P,I}} X(p,i,n) \qquad \forall p, n = N, \Delta n = 1 \qquad (4c)$$

Each parcel must end exactly once within the scheduling horizon.

$$\sum_n xe(p,n) = 1 \qquad \forall p, \Delta n = 1 \qquad (5)$$

We define a positive variable $V_{P,I}(p,i,n)$ to denote the crude volume unloaded from parcel p to tank i during event point n. $T_{P,I}^{\text{start}}$ (p,i,n) and $T_{P,I}^{\text{end}}$ (p,i,n) are used to denote the start and end times of parcel p unloaded to tank i during event point n. Then, the crude volume must meet its minimum [$F_{P,I}^{\min}$ (p,i)] and maximum [$F_{P,I}^{\max}$ (p,i)] unloading rates.

$$V_{P,I}(p,i,n) \geq F_{P,I}^{\min}(p,i) \cdot \left[T_{P,I}^{\text{end}}(p,i,n) - T_{P,I}^{\text{start}}(p,i,n)\right] \qquad \forall (p,i) \in \mathbf{S}_{P,I}, n \qquad (6a)$$

$$V_{P,I}(p,i,n) \le F_{P,I}^{\max}(p,i) \cdot \left[T_{P,I}^{\text{end}}(p,i,n) - T_{P,I}^{\text{start}}(p,i,n)\right] \quad \forall (p,i) \in \mathbf{S}_{P,I}, n \quad (6b)$$

If a parcel p is not unloaded into a tank i during event point n, then its crude volume unloaded must be zero.

$$V_{P,I}(p,i,n) = V_P^{\text{init}}(p) \cdot X(p,i,n) \quad \forall (p,i) \in \mathbf{S}_{P,I}, \Delta n = 0 \quad (7a)$$

$$V_{P,I}(p,i,n) \le V_{P,I}^{\max}(p,i,n) \cdot X(p,i,n) \quad \forall (p,i) \in \mathbf{S}_{P,I}, \Delta n = 1 \quad (7b)$$

where $V_{P,I}^{\max}$ (p,i,n) is the maximum crude volume of parcel p that can be unloaded to tank i during event point n, that is, it is the upper bound (UB) of $V_{P,I}$ (p,i,n).

Total crude volume unloaded for each parcel should be equal to its initial volume [V_P^{init} (p)].

$$\sum_{i:(p,i)\in \mathbf{S}_{P,I}} \sum_n V_{P,I}(p,i,n) = V_P^{\text{init}}(p) \quad \forall p, \Delta n = 1 \quad (8)$$

We define two positive variables T_P^{start} (p) and T_P^{end} (p) to denote the start and end unloading time of parcel p. Then, the relationships among T_P^{start} (p), T_P^{end} (p), $T_{P,I}^{\text{start}}$ (p,i,n), and $T_{P,I}^{\text{end}}$ (p,i,n) are presented by

$$T_P^{\text{start}}(p) \le T_{P,I}^{\text{start}}(p,i,n) + H[1 - X(p,i,n)] \quad \forall (p,i) \in \mathbf{S}_{P,I}, n \quad (9a)$$

$$T_P^{\text{end}}(p) \ge T_{P,I}^{\text{end}}(p,i,n) - H[1 - X(p,i,n)] \quad \forall (p,i) \in \mathbf{S}_{P,I}, n \quad (9b)$$

For VLCC parcels, parcel $(p + 1)$ must start unloading after its previous parcel p is completed.

$$T_P^{\text{start}}(p+1) \ge T_P^{\text{end}}(p) \quad \forall p \in \mathbf{VP} \quad (10)$$

Similarly, if there is only one jetty, then the jetty parcels must be unloaded based on their arrival time. We define $T_{\text{ARR}}(p)$ as the arrival time of parcel p.

$$T_P^{\text{start}}(p') \ge T_P^{\text{end}}(p) \quad \forall p, p' \in \mathbf{JP}, T_{\text{ARR}}(p) < T_{\text{ARR}}(p'), B = 1 \quad (11)$$

We define V_I (i,n) as the crude volume in tank i at the end of event point n and $V_{I,C}$ (i,c,n) as the volume of crude c in tank i at the end of event point n.

$$\sum_{c:(i,c)\in \mathbf{S}_{I,C}} V_{I,C}(i,c,n) = V_I(i,n) \quad \forall i, n \quad (12)$$

where $\mathbf{S}_{I,C} = \{(i,c) \mid \text{tank } i \text{ that can hold crude } c\}$.

At any time, the crude c in tank i must meet its lower [$E_I^{\min}$ (i,c)] and upper [$E_I^{\max}$ (i,c)] fractions in this tank.

$$V_I(i,n) \cdot E_I^{\min}(i,c) \le V_{I,C}(i,c,n) \quad \forall (i,c) \in \mathbf{S}_{I,C}, n \quad (13a)$$

$$V_{I,C}(i,c,n) \le V_I(i,n) \cdot E_I^{\max}(i,c) \quad \forall (i,c) \in \mathbf{S}_{I,C}, n \quad (13b)$$

For each storage tank i, we identify several possible event points during which the concentration of this tank i is the same as its initial composition. These possible event points on each tank is defined as $\mathbf{S}_{F,I}$. During these possible event points

$$V_{I,U,C}(i,u,c,n) = E_I^{\text{init}}(i,c) \cdot V_{I,U}(i,u,n) \quad \forall (i,u) \in \mathbf{S}_{I,U}, (i,c) \in \mathbf{S}_{I,C}, (i,n) \in \mathbf{S}_{F,I}, n \quad (14)$$

At other event points on each tank i [i.e., $(i, n) \notin \mathbf{S}_{F,I}$], we define a positive variable $E_{I,C}(i,c,n)$ to denote the unknown fraction of crude c in tank i at the end of event point n.

$$V_{I,U,C}(i,u,c,n) = E_{I,C}(i,c,n-1) \cdot V_{I,U}(i,u,n) \quad \forall (i,u) \in \mathbf{S}_{I,U}, (i,c) \in \mathbf{S}_{I,C}, (i,n) \notin \mathbf{S}_{F,I}, n \quad (15)$$

At the end of each event point n

$$V_{I,C}(i,c,n) = E_{I,C}(i,c,n) \cdot V_I(i,n) \quad \forall (i,c) \in \mathbf{S}_{I,C}, n \quad (16)$$

Note that Eqs. 15 and 16 involve bilinear terms.

Tank loading and charging

To model tank feeding CDU operations, we define a binary variable $Y(i,u,n)$ as follows

$$Y(i,u,n) = \begin{cases} 1 & \text{if tank } i \text{ feeds CDU } u \text{ during event point } n \\ 0 & \text{otherwise} \end{cases} \quad \forall (i,u) \in \mathbf{S}_{I,U}$$

where $\mathbf{S}_{I,U} = \{(i,u) \mid \text{tank } i \text{ that can feed CDU } u\}$.

A storage tank cannot receive and charge during the same event point n. Thus

$$X(p,i,n) + Y(i,u,n) \le 1 \quad \forall (p,i) \in \mathbf{S}_{P,I}, (i,u) \in \mathbf{S}_{I,U}, n \quad (17)$$

At any time, one storage tank can feed at most two CDUs simultaneously and *vice versa*.

$$\sum_{u:(i,u)\in \mathbf{S}_{I,U}} Y(i,u,n) \le 2 \quad \forall i, n \quad (18a)$$

$$\sum_{i:(i,u)\in \mathbf{S}_{I,U}} Y(i,u,n) \le 2 \quad \forall u, n \quad (18b)$$

We define a positive variable $V_{I,U}(i,u,n)$ to denote the total amount of crudes charged from tank i to CDU u during event point n. Then, the total amount of crudes from storage tank i to CDU u during each event point n must meet its minimum [$F_{I,U}^{\min}$ (i,u)] and maximum [$F_{I,U}^{\max}$ (i,u)] feed rates.

$$V_{I,U}(i,u,n) \le F_{I,U}^{\max}(i,u) \cdot \left[T_{I,U}^{\text{end}}(i,u,n) - T_{I,U}^{\text{start}}(i,u,n)\right] \quad \forall (i,n) \in \mathbf{S}_{I,U} \quad (19a)$$

$$F_{I,U}^{\min}(i,u) \cdot \left[T_{I,U}^{\text{end}}(i,u,n) - T_{I,U}^{\text{start}}(i,u,n)\right] \le V_{I,U}(i,u,n) \quad \forall (i,n) \in \mathbf{S}_{I,U} \quad (19b)$$

Table 1. Crude Properties, Their Relevance, and Corresponding Indexes and Correlations

Crude Property	Blending Index	Addition Basis	Relevance to (Important for)	Index Correlation
Specific gravity (SG)	DNI	Volume	Crudes, all products	1/SG
Sulfur	SULI	Weight	Crudes, all products	Weighted average
Nitrogen	NITI	Weight	Crudes, residue streams (>550 °C), vacuum gas oil (370–550 °C)	Weighted average
Carbon residue	CRI	Weight	Crudes, residue streams (550 °C), vacuum gas oil (370–550 °C)	Weighted average
Pour point (PP, °C)	PIndex	Volume	Crudes, all products	316,200× Exp(12.5×Log (0.001(1.8×PP+491.67)))
Freeze point (°C)	FreezeIndex	Volume	Kerosene (150–280 °C)	3,162,000× Exp(12.5×Log (0.001(1.8×Freeze point+491.67)))
Flash point (FLP, °C)	FPIndex	Volume	All products	Exp((−6.1184+(2414/ (FLP+230.56)))×Log(10))
Smoke point (SMP, mm)	SMI	Volume	Kerosene (150–280 °C)	−362+3200/Log(SMP)
Ni	NiIndex	Weight	Crudes, residue streams (>550 °C), vacuum gas oil (370–550 °C)	Weighted average
Reid vapor pressure (RVP, bar)	RVI	Volume	Crudes, products up to naphtha range boiling below 200 °C	Exp(1.14×Log(100×RVP))
Asphaltenes	ASPI	Weight	Crudes, residue streams (>550 °C), vacuum gas oil (370–550 °C)	Weighted average
Aromatics	AROI	Volume	Naphtha range boiling below 200 °C	Volumetric average
Paraffins	PARI	Volume	Naphtha range boiling below 200 °C	Volumetric average
Naphthenes	NAPHI	Volume	Naphtha range boiling below 200 °C	Volumetric average
Viscosity at 50 °C (Visc_cst)	ViscIndex	Weight	Crudes, residue streams (>550 °C), vacuum gas oil (370–550 °C)	79.1+33.47×(Log(Log (Visc_cst+0.8)/Log(10))/Log(10))

Index corrections from Reddy (Singapore Petroleum Company).

If tank i does not feed CDU u at event point n, then the total amount [$V_{I,U}(i,u,n)$] charged should be zero.

$$V_{I,U}(i,u,n) \le V_{I,U}^{\max}(i,u,n) \cdot Y(i,u,n) \qquad \forall (i,u) \in \mathbf{S}_{I,U} \tag{20}$$

We define a positive variable $V_{I,U,C}(i,u,c,n)$ to denote the amount of crude c fed from tank i to CDU u during event point n. Then, we can write

$$V_{I,U}(i,u,n) = \sum_{c:(i,c)\in \mathbf{S}_{I,C}} V_{I,U,C}(i,u,c,n) \qquad \forall (i,u) \in \mathbf{S}_{I,U} \tag{21}$$

The total amount of crudes [$V_U(u,n)$] fed to each CDU u during event point n is given by

$$\sum_{i:(i,u)\in \mathbf{S}_{I,U}} V_{I,U}(i,u,n) = V_U(u,n) \qquad \forall u,n \tag{22}$$

During each event point n, the total amount of crudes fed to each CDU u must meet its minimum [$D_U^{\min}(u)$] and maximum [$D_U^{\max}(u)$] processing rates.

$$D_U^{\min}(u)\left[T_U^{\text{end}}(u,n) - T_U^{\text{start}}(u,n)\right] \le V_U(u,n) \qquad \forall u,n \tag{23a}$$

$$V_U(u,n) \le D_U^{\max}(u)\left[T_U^{\text{end}}(u,n) - T_U^{\text{start}}(u,n)\right] \qquad \forall u,n \tag{23b}$$

In plant operation, CDUs cannot process crude mixtures with some extreme fractions of crudes. Therefore, the crude fraction in the feed to any CDU u must also meet its minimum [$E_U^{\min}(u, c)$] and maximum [$E_U^{\max}(u, c)$] fractions.

$$V_U(u,n) \cdot E_U^{\min}(u,c) \le \sum_{i:(i,u)\in \mathbf{S}_{I,U}} V_{I,U,C}(i,u,c,n) \qquad \forall (u,c) \in \mathbf{S}_{U,C} \tag{24a}$$

$$\sum_{i:(i,u)\in \mathbf{S}_{I,U}} V_{I,U,C}(i,u,c,n) \le V_U(u,n) \cdot E_U^{\max}(u,c) \qquad \forall (u,c) \in \mathbf{S}_{U,C} \tag{24b}$$

Product quality

In practice, the refiners must ensure acceptable qualities of feeds to CDUs, as a feed with poor quality can seriously disrupt the operation of a CDU and even downstream units. Various crude properties are used in practice, such as specific gravity, sulfur, nitrogen, oxygen, carbon residue, pour point, flash point, nickel, Reid vapor pressure, asphaltene, aromatics, paraffins, naphthene, wax, and viscosity. Many of these properties (e.g., Reid vapor pressure and pour point) involve highly nonlinear mixing rules. However, as noted by Li et al.,[4] a linear blending index usually exists and is used for almost every hydrocarbon property with nonlinear mixing correlations. These blending indices are linearly additive on either volume or weight basis. Table 1 lists 15 crude properties, indices, and their additive bases. Let $e_C(c,k)$ be the known blending index for a property k of crude c, ρ_c be the density of crude c, and $e_U^{\min}(u, k)$ and $e_U^{\max}(u, k)$ are the lower and upper acceptable limits, respectively, on property k that feed to CDU u. Then, the following ensure the desired crude qualities feeding to CDUs.

$$e_U^{\min}(u,k) \cdot \sum_{i:(i,u)\in \mathbf{S}_{I,U}} V_{I,U}(i,u,n) \le \sum_{i:(i,u)\in \mathbf{S}_{I,U}} \sum_{c:(i,c)\in \mathbf{S}_{I,C}} e_C(c,k) \cdot V_{I,U,C}(i,u,c,n) \qquad \forall (i,u) \in \mathbf{S}_{I,U} \tag{25a}$$

$$\sum_{i:(i,u)\in \mathbf{S}_{I,U}} \sum_{c:(i,c)\in \mathbf{S}_{I,C}} e_C(c,k)\cdot V_{I,U,C}(i,u,c,n) \le e_U^{\max}(u,k) \cdot \sum_{i:(i,u)\in \mathbf{S}_{I,U}} V_{I,U}(i,u,n) \qquad \forall (i,u)\in \mathbf{S}_{I,U} \quad (25b)$$

$$e_U^{\min}(u,k)\cdot\left(\sum_{i:(i,u)\in \mathbf{S}_{I,U}} \sum_{c:(i,c)\in \mathbf{S}_{I,C}} \rho_c \cdot V_{I,U,C}(i,u,c,n)\right) \le \sum_{i:(i,u)\in \mathbf{S}_{I,U}} \sum_{c:(i,c)\in \mathbf{S}_{I,C}} e_C(c,k)\cdot \rho_c \cdot V_{I,U,C}(i,u,c,n) \quad \forall (i,u)\in \mathbf{S}_{I,U} \quad (26a)$$

$$\sum_{i:(i,u)\in \mathbf{S}_{I,U}} \sum_{c:(i,c)\in \mathbf{S}_{I,C}} e_C(c,k)\cdot \rho_c \cdot V_{I,U,C}(i,u,c,n) \le e_U^{\max}(u,k) \cdot\left(\sum_{i:(i,u)\in \mathbf{S}_{I,U}} \sum_{c:(i,c)\in \mathbf{S}_{I,C}} \rho_c \cdot V_{I,U,C}(i,u,c,n)\right) \qquad \forall (i,u)\in \mathbf{S}_{I,U} \quad (26b)$$

Note that Eq. 25 is for volume-based indices, and Eq. 26 is for weight-based indices.

Demand requirement

The demand for the crude mixture from each CDU u is equal to the sum of CDU volumes over all event points:

$$\sum_n V_U(u,n) = D(u) \qquad \forall u \quad (27)$$

Sequencing constraints on storage tanks and CDUs

Any parcel p unloading to tank i during event point $n + 1$ must start after its unloading to this tank i at event point n.

$$T_{P,I}^{\text{start}}(p,i,n+1) \ge T_{P,I}^{\text{end}}(p,i,n) \qquad \forall (p,i)\in \mathbf{S}_{P,I}, n<N \quad (28)$$

If a parcel p is allowed to unload in multiple event points (i.e., $\Delta n = 1$), then this parcel unloading during event point $n + 1$ must start after it completes unloading during event point n.

$$T_{P,I}^{\text{start}}(p,i',n+1) \ge T_{P,I}^{\text{end}}(p,i,n) - H[1 - X(p,i,n)] \qquad \forall (p,i)\in \mathbf{S}_{P,I}, (p,i')\in \mathbf{S}_{P,I}, n<N, \Delta n = 1 \quad (29)$$

If a tank i receives a parcel p during event point n, then this tank i must start to receive any other parcel p' at event point $n + 1$ after it completes receiving p at event point n.

$$T_{P,I}^{\text{start}}(p',i,n+1) \ge T_{P,I}^{\text{end}}(p,i,n) - H[1 - X(p,i,n)] \qquad \forall (p,i)\in \mathbf{S}_{P,I}, (p',i)\in \mathbf{S}_{P,I}, p\ne p', n<N \quad (30)$$

If a parcel p is allowed to be unloaded in multiple event points (i.e., $\Delta n = 1$), then it must be unloaded consecutively. In other words, if this parcel is unloaded during event points n and $n + 1$, then this parcel unloaded during event point $n + 1$ must start immediately after it completes during event point n.

$$T_{P,I}^{\text{start}}(p,i',n+1) \le T_{P,I}^{\text{end}}(p,i,n) + H[2 - X(p,i,n) - X(p,i',n+1)] \qquad \forall (p,i)\in \mathbf{S}_{P,I}, (p,i')\in \mathbf{S}_{P,I}, n<N, \Delta n = 1 \quad (31)$$

If a parcel p is unloaded via a jetty during event point n, then the start and end times of this parcel p unloaded to any tank must be within those of jetties. Thus

$$T_B^{\text{start}}(n) \le T_{P,I}^{\text{start}}(p,i,n) + H[1 - X(p,i,n)] \qquad \forall p\in \mathbf{JP}, (p,i)\in \mathbf{S}_{P,I}, n<N, B>1 \quad (32a)$$

$$T_B^{\text{end}}(n) \ge T_{P,I}^{\text{start}}(p,i,n) - H[1 - X(p,i,n)] \qquad \forall p\in \mathbf{JP}, (p,i)\in \mathbf{S}_{P,I}, n<N, B>1 \quad (32b)$$

A tank i must start to feed CDU u at event point $n + 1$ after it completes feeding this CDU at event point n.

$$T_{I,U}^{\text{start}}(i,u,n+1) \ge T_{I,U}^{\text{end}}(i,u,n) \qquad \forall (i,u)\in \mathbf{S}_{I,U}, n<N \quad (33)$$

If a tank i feeds any CDU u during event point n, then this tank i must start to receive any parcel p at event point $n + 1$ after it completes feeding at event point n.

$$T_{P,I}^{\text{start}}(p,i,n+1) \ge T_{I,U}^{\text{end}}(i,u,n) - H[1 - Y(i,u,n)] \qquad \forall (p,i)\in \mathbf{S}_{P,I}, (i,u)\in \mathbf{S}_{I,U}, n<N \quad (34)$$

Similarly, if a tank i receives any parcel p during event point n, then this tank must start to feed any CDU u during event point $n + 1$ after it completes receiving at event point n.

$$T_{I,U}^{\text{start}}(i,u,n+1) \ge T_{P,I}^{\text{end}}(p,i,n) - H[1 - X(p,i,n)] \qquad \forall (p,i)\in \mathbf{S}_{P,I}, (i,u)\in \mathbf{S}_{I,U}, n<N \quad (35)$$

If a tank i feeds a CDU u during event point n, then it must start feeding any CDU u during event point $(n + 1)$ after it completes feeding at event point n.

$$T_{I,U}^{\text{start}}(i,u',n+1) \ge T_{I,U}^{\text{end}}(i,u,n) - H[1 - Y(i,u,n)] \qquad \forall (i,u)\in \mathbf{S}_{I,U}, (i,u')\in \mathbf{S}_{I,U}, u\ne u', n<N \quad (36)$$

As multiple tanks can feed a CDU u at the same time, we must impose the same start and end times for each tank feeding to avoid discrepancy.[45] In other words

$$T_U^{\text{start}}(u,n) \ge T_{I,U}^{\text{start}}(i,u,n) - H[1 - Y(i,u,n)] \qquad \forall (i,u)\in \mathbf{S}_{I,U}, n \quad (37a)$$

$$T_U^{\text{start}}(u,n) \le T_{I,U}^{\text{start}}(i,u,n) + H[1 - Y(i,u,n)] \qquad \forall (i,u)\in \mathbf{S}_{I,U}, n \quad (37b)$$

$$T_U^{\text{end}}(u,n) \ge T_{I,U}^{\text{end}}(i,u,n) - H[1 - Y(i,u,n)] \qquad \forall (i,u)\in \mathbf{S}_{I,U}, n \quad (38a)$$

$$T_U^{\text{end}}(u,n) \le T_{I,U}^{\text{end}}(i,u,n) + H[1 - Y(i,u,n)] \qquad \forall (i,u)\in \mathbf{S}_{I,U}, n \quad (38b)$$

Brine settling

After a tank i receives crudes from vessels, it needs some time to settle and remove brine before it can feed any CDU. Thus, if it completes receiving crudes at the end of event point n, then it must feed any CDU u after some required settling time from the end of event point n. Let ST denote this required settling time. Then

$$T_{I,U}^{\text{start}}(i,u,n+1) \geq T_{P,I}^{\text{end}}(p,i,n) + ST \cdot X(p,i,n) - H[1 - X(p,i,n)] \quad \forall (p,i) \in \mathbf{S}_{P,I}, (i,u) \in \mathbf{S}_{I,U}, n<N \qquad (39)$$

Note that Eq. 39 is more general than Eq. 35. We can remove Eq. 35 from the model.

Mass balance

The sequencing constraints (Eqs. 28–39) make us write the correct inventory balance for each storage tank i at the end of event point n.

$$V_{I,C}(i,c,n) = V_{I,C}(i,c,n-1) + \sum_{p:(p,c)\in \mathbf{S}_{P,C}} V_{P,I}(p,i,n) \cdot E_P(p,c) - \sum_{u:(i,u)\in \mathbf{S}_{I,U}} V_{I,U,C}(i,u,c,n) \qquad \forall (i,c) \in \mathbf{S}_{I,C}, n > 1 \qquad (40a)$$

$$V_{I,C}(i,c,n) = V_{I,C}^{\text{init}}(i,c) + \sum_{p:(p,c)\in \mathbf{S}_{P,C}} V_{P,I}(p,i,n) \cdot E_P(p,c) - \sum_{u:(i,u)\in \mathbf{S}_{I,U}} V_{I,U,C}(i,u,c,n) \qquad \forall (i,c) \in \mathbf{S}_{I,C}, n = 1 \qquad (40b)$$

where $E_P(p,c)$ denotes fraction of crude c in parcel p, and $V_{I,C}^{\text{init}}(i,c)$ is the initial volume of crude c in tank i.

CDU operation continuity

During the scheduling horizon, each CDU u must operate continually.

$$\sum_n \left[T_U^{\text{end}}(u,n) - T_U^{\text{start}}(u,n)\right] = H \qquad \forall u \qquad (41)$$

Changeovers

Changeover is defined as a change in the feed composition that upset the steady operation of a CDU.[8] As pointed out by Reddy et al.,[8] a changeover can perturb the processing unit operation and lead to calling off special products, generation of off-spec products, and additional work resulting in lost productivity. Thus, every changeover incurs some cost to the refinery and is undesirable. In real operation, refiners strive to minimize the changeovers. To detect such changes, we define a 0–1 continuous variable $z(u,n)$ as follows

$$z(u,n) = \begin{cases} 1 & \text{if a tank switch on CDU } u \text{ takes place at the end of event } n \\ 0 & \text{otherwise} \end{cases} \qquad \forall u, n$$

$$z(u,n) \geq Y(i,u,n) - Y(i,u,n+1) \qquad \forall (i,u) \in \mathbf{S}_{I,U}, n<N \qquad (42a)$$

$$z(u,n) \geq Y(i,u,n+1) - Y(i,u,n) \qquad \forall (i,u) \in \mathbf{S}_{I,U}, n<N \qquad (42b)$$

Demurrage

A key operating cost in crude oil scheduling is the demurrage or sea-waiting cost. The logistics contract with each vessel stipulates an acceptable sea-waiting period. If the vessel harbors beyond this stipulated period, then the demurrage (or sea waiting cost) incurs. Let $T_{\text{ULD}}^{\min}(v)$ be the stipulated time of departure in the logistics contract of vessel v. The demurrage incurs, if the last parcel of vessel v remains connected to the SBM/jetty line beyond $T_{\text{ULD}}^{\min}(v)$. If vessel v arrives at time $T_{\text{ARR}}(p)$ and its demurrage or sea-waiting cost is C_{SEA} (k\$ per unit time), then the demurrage incurred is given by

$$T_{\text{CW}}(v) \geq C_{\text{SEA}} \cdot \left[T_P^{\text{end}}(p) - T_{\text{ARR}}(p) - T_{\text{ULD}}^{\min}(v)\right] \quad \forall (v,p) \in \mathbf{S}_{V,P}^L \qquad (43)$$

where set $\mathbf{S}_{V,P}^L = \{(v,p) \mid \text{parcel } p \text{ is the last parcel in vessel } v\}$.

Safety stock

In real operations, one inventory-related decision does fall under the scheduling activity. The desire of most refiners is to maintain a minimum stock of crude to guard against uncertainty. As we only know the inventory level in each storage tank at each event point, the total inventory levels of storage tanks at each time are approximated by $\frac{\sum_i \left(\sum_n V_I(i,n) + V_I^{\text{init}}(i)\right)}{N+1}$. Let SS be the desired safety stock of crude, and SSP be the average safety stock at the end of each event point.

$$SSP \geq SS - \frac{\sum_i \left(\sum_n V_I(i,n) + V_I^{\text{init}}(i)\right)}{N+1} \qquad (44)$$

Objective

We use the total gross profit as the scheduling objective. This is defined as the total netback from crudes minus the operating cost. The netback from crudes is the value of products minus the purchase cost of crude. As the product yields vary with crudes and CDUs, we define $C_{\text{PROF}}(c)$ as the netback (\$ per unit volume) for crude c processed in CDU u, C_{PEN} as the penalty (\$ per unit volume per hour) for under-running the crude safety stock, and C_{SET} as changeover cost (k\$ per changeover). Note that the netback does not include any operating costs. The operating costs include the demurrage, the penalty for under-running the crude safety stock, and the changeover costs.

$$\text{PROFIT} = \sum_i \sum_u \sum_c \sum_n C_{\text{PROF}}(c) \cdot V_{I,U,C}(i,u,c,n) - \sum_v T_{\text{CW}}(v) - C_{\text{PEN}} \cdot SSP \cdot H - \sum_u \sum_n C_{\text{SET}} \cdot z(u,n) \qquad (45)$$

Hard bounds

$$0 \le xe(p,n) \le 1 \qquad \forall\, p,n \tag{46}$$

$$0 \le z(u,n) \le 1 \qquad \forall\, u,n \tag{47}$$

$$SSP \le SS \tag{48}$$

$$E_{I,C}^{\min}(i,c,n) = 0, E_{I,C}^{\max}(i,c,n) = 1 \qquad \forall (i,c) \in \mathbf{S}_{I,C}, n \tag{49}$$

$$V_I^{\min}(i,n) = V_I^{\min}(i), V_I^{\max}(i,n) = V_I^{\max}(i) \quad \forall\, i,n \tag{50}$$

$$T_{\text{ARR}}(p) \le T_P^{\text{start}}(p) \le H \qquad \forall\, p \tag{51a}$$

$$T_{\text{ARR}}(p) \le T_P^{\text{end}}(p) \le H \qquad \forall\, p \tag{51b}$$

$$T_{P,I}^{\text{start}}(p,i,n) \le H \qquad \forall (p,i) \in \mathbf{S}_{P,I}, n \tag{52a}$$

$$T_{P,I}^{\text{end}}(p,i,n) \le H \qquad \forall (p,i) \in \mathbf{S}_{P,I}, n \tag{52b}$$

$$T_B^{\text{start}}(n) \le H \qquad \forall\, n \tag{53a}$$

$$T_B^{\text{end}}(n) \le H \qquad \forall\, n \tag{53b}$$

$$T_{I,U}^{\text{start}}(i,u,n) \le H \qquad \forall (i,u) \in \mathbf{S}_{I,U}, n \tag{54a}$$

$$T_{I,U}^{\text{end}}(i,u,n) \le H \qquad \forall (i,u) \in \mathbf{S}_{I,U}, n \tag{54b}$$

$$T_U^{\text{start}}(u,n) \le H \qquad \forall\, u,n \tag{55a}$$

$$T_U^{\text{end}}(u,n) \le H \qquad \forall\, u,n \tag{55b}$$

As noted by Androulakis et al.,[48] tight bounds on variables participating in bilinear terms are critically important such as on the variables $E_{I,C}(i,c,n)$. Besides the hard bounds (i.e., Eqs. 46–55), we update the values of some parameters based on detailed analysis. The updated values for some parameters are presented in Appendix.

We complete our novel unit-specific event-based formulation. The entire nonconvex MINLP problem denoted as **MP** is presented as follows:

$$\begin{aligned} (\mathbf{MP}) \quad \mathbf{Min} \quad & -\text{PROFIT} \\ s.t. \quad & \text{eqs. } 1-44 \\ & \text{eqs. } 46-55 \text{ and eqs. A1} - \text{A7 in Appendix} \end{aligned}$$

It should also be noted that there are some differences between the proposed unit-specific event-based continuous-time model and the discrete-time models proposed and enhanced by Reddy et al.[8] and Li et al.[4] These differences include the demurrage calculation, the violation of minimum safety stock, and the changeover caused by a change in the tank-to-CDU flow, which have been already discussed in detail by Reddy et al.[10] Moreover, the proposed unit-specific event-based continuous-time model does not impose constraints to control period-to-period changes in crude feed rates as Li et al.[4] did as a comparison of flow rates in consecutive event points results in nonlinearities. The resulting mathematical model is a nonconvex MINLP, and the sources of nonconvexities are the distinct bilinear terms (i.e., Eqs. 15 and 16) presented as follows

$$V_{I,U,C}(i,u,c,n) = E_{I,C}(i,c,n-1) \cdot V_{I,U}(i,u,n) \qquad \forall (i,u) \in \mathbf{S}_{I,U}, (i,c) \in \mathbf{S}_{I,C}, (i,n) \notin \mathbf{S}_{F,I}, n \tag{15$'$}$$

$$V_{I,C}(i,c,n) = E_{I,C}(i,c,n) \cdot V_I(i,n) \qquad \forall (i,c) \in \mathbf{S}_{I,C}, n \tag{16$'$}$$

From Eqs. 15 and 16, we can calculate the total number of bilinear terms for the problem **MP** as follows

$$\text{No. bilinear terms} = N \cdot \left(\sum_{i=1}^{I} \sum_{c:(i,c)\in \mathbf{S}_{I,C}} 1 \right) + \left(\sum_{i=1}^{I} \sum_{u:(i,u)\in \mathbf{S}_{I,U}} \sum_{c:(i,c)\in \mathbf{S}_{I,C}} \sum_{n:(i,n)\notin \mathbf{S}_{F,I}} 1 \right) \tag{56}$$

Branch-and-bound global optimization algorithm

In this section, we discuss the branch-and-bound global optimization algorithm that we use to solve this complex nonconvex MINLP problem. At each node in the branch-and-bound tree, we minimize a piecewise-linear relaxation of the node using the solver CPLEX[49] 11.0.0/GAMS 22.6 and branch the node to create two child nodes. From solving this piecewise-linear relaxation, we obtain a pool of feasible solutions including the final solve, which is the best or optimal integer solution for this relaxation. We define this pool as pool-1. Note that each solution from pool-1 is not feasible for the problem **MP**. If the final solve is greater than the current lower bound (LB), the LB is updated with this final solve. Then, we use each solution from the pool-1 to fix the current values of the binary variables, to initialize the continuous variables using their current values, and to locally minimize the resulting NLP using CONOPT3 (Ref. 50)/GAMS 22.6. With this, we obtain another pool of feasible local optimal solutions and define this pool to be pool-2. If the smallest objective value in the pool-2 is less than the current UB, then the UB is updated with this value. At each step, nodes with relaxations within a predetermined tolerance (e.g., $\varepsilon = 0.02$) of the current UB are eliminated. In other words, nodes such that [UB $\le$ LB $(1 - \varepsilon)$] are eliminated. The algorithm terminates with ε-convergence when there are no remaining nodes in the branch-and-bound tree. Comprehensive coverage of branch-and-bound algorithms can be found in the textbooks written by Floudas.[51,52]

Next, we discuss in detail some strategies that we use in the global optimization algorithm, which include piecewise-linear underestimators, branching strategy, solution improvement strategy, optimality-based tightening LBs and UBs, and so on.

Piecewise-linear underestimators

Following the suggestion of recent studies by Wicaksono and Karimi,[18] Gounaris et al.,[19] Misener and Floudas,[24] and

Misener et al.,[26] we choose to piecewise-underestimate each of the bilinear terms in the model using the **nf4r** formulation.[19] As $E_{I,C}(i,c,n)$ occurs in both bilinear terms, we choose to partition the $E_{I,C}(i,c,n)$ based on our computational experience. We uniformly partitioned the $E_{I,C}(i,c,n)$ variables because equal partitioning often gives the tightest relaxation.[21] In the following, we briefly introduce the **nf4r** formulation. The details about **nf4r** formulation can be found in the articles written by Gounaris et al.[19] and Misener and Floudas.[24]

First, each $E_{I,C}(i,c,n)$ variable is *ab initio* partitioned into GR (gr = 0, 1, 2, …, GR) segments with (GR + 1) grid points as follows

$$E_{I,C}(i,c,n,gr) = E_I^{\min}(i,c) + \frac{gr}{GR}\left[E_I^{\max}(i,c) - E_I^{\min}(i,c)\right] \quad \forall (i,c) \in \mathbf{S}_{I,C}, n, gr \quad (57)$$

Second, an **SOS1** variable $\lambda_{I,C}(i,c,n,gr)$ is defined as follows to activate one and only one segment

$$\lambda_{I,C}(i,c,n,gr) = \begin{cases} 1 & \text{if} E_{I,C}(i,c,n,gr-1) \le E_{I,C}(i,c,n) \le E_{I,C}(i,c,n,gr) \\ 0 & \text{otherwise} \end{cases}$$

$$\forall (i,\mathrm{c}) \in \mathbf{S}_{I,C}, n, gr > 0 \sum_{gr=1}^{GR} \lambda_{I,C}(i,c,n,gr) = 1 \qquad \forall (i,c) \in \mathbf{S}_{I,C}, n \quad (58)$$

$$\sum_{gr=1}^{GR} E_{I,C}(i,c,n,gr-1)\cdot\lambda_{I,C}(i,c,n,gr) \le E_{I,C}(i,c,n) \le \sum_{gr=1}^{GR} E_{I,C}(i,c,n,gr)\cdot\lambda_{I,C}(i,c,n,gr) \qquad \forall (i,c) \in \mathbf{S}_{I,C}, n \quad (59)$$

Third, two continuous variables $\Delta V_{I,U}(i,u,c,n,gr)$ and $\Delta V_I(i,c,n,gr)$ are defined as follows to place holders for $V_{I,U}(i,u,n)$ and $V_I(i,n)$

$$V_{I,U}(i,u,n) = V_{I,U}^{\min}(i,u,n) + \sum_{gr=1}^{GR} \Delta V_{I,U}(i,u,c,n,gr) \quad \forall (i,u) \in \mathbf{S}_{I,U}, (i,c) \in \mathbf{S}_{I,C}, n \quad (60)$$

$$V_I(i,n) = V_I^{\min}(i,n) + \sum_{gr=1}^{GR} \Delta V_I(i,c,n,gr) \qquad \forall (i,c) \in \mathbf{S}_{I,C}, n \quad (61)$$

$$\Delta V_{I,U}(i,u,c,n,gr) \le \left[V_{I,U}^{\max}(i,u,n) - V_{I,U}^{\min}(i,u,n)\right] \cdot \lambda_{I,C}(i,c,n-1,gr) \qquad \forall (i,u) \in \mathbf{S}_{I,U}, (i,c) \in \mathbf{S}_{I,C}, n, gr > 0 \quad (62)$$

$$\Delta V_I(i,c,n,gr) \le \left[V_I^{\max}(i,n) - V_I^{\min}(i,n)\right] \cdot \lambda_{I,C}(i,c,n,gr) \quad \forall (i,c) \in \mathbf{S}_{I,C}, n, gr > 0 \quad (63)$$

With Eqs. 57–63, the relaxation of the bilinear terms (Eqs. 15 and 16 or 15′ and 16′) is given by

$$\begin{cases} V_{I,U,C}(i,u,c,n) \ge E_{I,C}(i,c,n-1)\cdot V_{I,U}^{\min}(i,u,n) + \sum_{gr=1}^{GR} E_{I,C}(i,c,n-1,gr-1)\cdot \Delta V_{I,U}(i,u,c,n,gr) \\ V_{I,U,C}(i,u,c,n) \ge E_{I,C}(i,c,n-1)\cdot V_{I,U}^{\max}(i,u,n) + \\ \sum_{gr=1}^{GR} \left\{ E_{I,C}(i,c,n-1,gr)\cdot\left[\Delta V_{I,U}(i,u,c,n,gr) - (V_{I,U}^{\max}(i,u,n) - V_{I,U}^{\min}(i,u,n))\cdot\lambda_{I,C}(i,c,n-1,gr)\right]\right\} \\ V_{I,U,C}(i,u,c,n) \le E_{I,C}(i,c,n-1)\cdot V_{I,U}^{\min}(i,u,n) + \sum_{gr=1}^{GR} E_{I,C}(i,c,n-1,gr)\cdot \Delta V_{I,U}(i,u,c,n,gr) \\ V_{I,U,C}(i,u,c,n) \le E_{I,C}(i,c,n-1)\cdot V_{I,U}^{\max}(i,u,n) + \\ \sum_{gr=1}^{GR} \left\{ E_{I,C}(i,c,n-1,gr-1)\cdot\left[\Delta V_{I,U}(i,u,c,n,gr) - (V_{I,U}^{\max}(i,u,n) - V_{I,U}^{\min}(i,u,n))\cdot\lambda_{I,C}(i,c,n-1,gr)\right]\right\} \end{cases}$$

$$\forall (i,u) \in \mathbf{S}_{I,U}, (i,c) \in \mathbf{S}_{I,C}, n \quad (64)$$

$$\begin{cases} V_{I,C}(i,c,n) \ge E_{I,C}(i,c,n)\cdot V_I^{\min}(i,n) + \sum_{gr=1}^{GR} E_{I,C}(i,c,n,gr-1)\cdot\Delta V_I(i,c,n,gr) \\ V_{I,C}(i,c,n) \ge E_{I,C}(i,c,n)\cdot V_I^{\max}(i,n) + \\ \sum_{gr=1}^{GR} \left\{E_{I,C}(i,c,n,gr)\cdot\left[\Delta V_I(i,c,n,gr) - (V_I^{\max}(i,n) - V_I^{\min}(i,n))\cdot\lambda_{I,C}(i,c,n,gr)\right]\right\} \\ V_{I,C}(i,c,n) \le E_{I,C}(i,c,n)\cdot V_I^{\min}(i,n) + \sum_{gr=1}^{GR} E_{I,C}(i,c,n,gr)\cdot\Delta V_I(i,c,n,gr) \\ V_{I,C}(i,c,n) \le E_{I,C}(i,c,n)\cdot V_I^{\max}(i,n) + \\ \sum_{gr=1}^{GR} \left\{E_{I,C}(i,c,n,gr-1)\cdot\left[\Delta V_I(i,c,n,gr) - (V_I^{\max}(i,n) - V_I^{\min}(i,n))\cdot\lambda_{I,C}(i,c,n,gr)\right]\right\} \end{cases} \quad \forall (i,c) \in \mathbf{S}_{I,C}, n \quad (65)$$

As noted by Misener and Floudas,[24] there exists a trade-off between few partitions, which solve quickly, and many partitions, which may close the gap in a single node. In other words, it is not clear how many partitions are appropriate. Here, we follow the approach used by Misener and Floudas,[24] in which they solved each example with several partitioning levels and reported the discretization generating the best results. It should also be noted that when the number of grid points is equal to 2, the **nf4r** formulation is reduced to McCormick convex and concave envelopes[15] as follows

$$V_{I,U,C}(i,u,c,n) \begin{cases} \geq E_{I,C}(i,c,n-1)\cdot V_{I,U}^{\min}(i,u,n)+E_{I,C}^{\min}(i,c,n-1)\cdot V_{I,U}(i,u,n)-E_{I,C}^{\min}(i,c,n-1)\cdot V_{I,U}^{\min}(i,u,n) \\ \leq E_{I,C}(i,c,n-1)\cdot V_{I,U}^{\min}(i,u,n)+E_{I,C}^{\max}(i,c,n-1)\cdot V_{I,U}(i,u,n)-E_{I,C}^{\max}(i,c,n-1)\cdot V_{I,U}^{\min}(i,u,n) \\ \leq E_{I,C}(i,c,n-1)\cdot V_{I,U}^{\max}(i,u,n)+E_{I,C}^{\min}(i,c,n-1)\cdot V_{I,U}(i,u,n)-E_{I,C}^{\min}(i,c,n-1)\cdot V_{I,U}^{\max}(i,u,n) \\ \geq E_{I,C}(i,c,n-1)\cdot V_{I,U}^{\max}(i,u,n)+E_{I,C}^{\max}(i,c,n-1)\cdot V_{I,U}(i,u,n)-E_{I,C}^{\max}(i,c,n-1)\cdot V_{I,U}^{\max}(i,u,n) \end{cases} \quad \forall (i,u)\in \mathbf{S}_{I,U},(i,c)\in \mathbf{S}_{I,C},n \tag{66}$$

$$V_{I,C}(i,c,n) \begin{cases} \geq E_{I,C}(i,c,n)\cdot V_{I}^{\min}(i,n)+E_{I,C}^{\min}(i,c,n)\cdot V_{I}(i,n)-E_{I,C}^{\min}(i,c,n)\cdot V_{I}^{\min}(i,n) \\ \leq E_{I,C}(i,c,n)\cdot V_{I}^{\min}(i,n)+E_{I,C}^{\max}(i,c,n)\cdot V_{I}(i,n)-E_{I,C}^{\max}(i,c,n)\cdot V_{I}^{\min}(i,n) \\ \leq E_{I,C}(i,c,n)\cdot V_{I}^{\max}(i,n)+E_{I,C}^{\min}(i,c,n)\cdot V_{I}(i,n)-E_{I,C}^{\min}(i,c,n)\cdot V_{I}^{\max}(i,n) \\ \geq E_{I,C}(i,c,n)\cdot V_{I}^{\max}(i,n)+E_{I,C}^{\max}(i,c,n)\cdot V_{I}(i,n)-E_{I,C}^{\max}(i,c,n)\cdot V_{I}^{\max}(i,n) \end{cases} \quad \forall (i,c)\in \mathbf{S}_{I,U},n \tag{67}$$

For convenience, we denote the piecewise linear relaxation of problem (**MP**) as problem (**RMP**), which is defined as follows

$$\begin{aligned} (\mathbf{RMP})\ \ \mathbf{Min}\quad & -\text{PROFIT} \\ s.t.\quad & \text{eqs. } 1-14,\ 17-44,\ \text{and } 57-65 \text{ or } 66 \text{ and } 67 \\ & \text{eqs. } 46-55 \text{ and eqs. } A1-A7 \text{ in Appendix} \end{aligned}$$

Branching strategy

After solving a relaxation of each node using the piecewise-linear underestimator, we have to select a variable for branching and the branching point. As we use the MILP solver CPLEX[49] 11.0.0 to optimize the relaxation of each node, we only need to branch on variables that participate in the nonconvex bilinear terms. In other words, we choose a variable for branching from $E_{I,C}(i,c,n)$, $V_I(i,n)$, and $V_{I,U}(i,u,n)$. Although the variable $E_{I,C}(i,c,n)$ occurs in both bilinear terms, our computational studies suggest that branching on $V_I(i,n)$ always perform better than $E_{I,C}(i,c,n)$ and $V_{I,U}(i,u,n)$. The possible reason is that the number of variables $V_I(i,n)$ is fewer than that of $E_{I,C}(i,c,n)$ and $V_{I,U}(i,u,n)$. Moreover, the LBs and UBs of $E_{I,C}(i,c,n)$ and $V_{I,U}(i,u,n)$ is greatly affected by those of $V_I(i,n)$ given in Appendix.

After generating the optimal solution ($\hat{X}$, $\hat{Y}$, $\hat{V}_{P,I}$ $\hat{V}_I$ $\hat{V}_{I,C}$ $\hat{E}_{I,C}$ $\hat{V}_{I,U}$) from piecewise-linear relaxation for a given node, we follow common practice by branching on the variable $V_I(i,n)$, where $(i,\ n)$ contributes to the greatest discrepancy between the auxiliary and original problem variables

$$\max_{i,n}\left\{\sum_{c:(i,c)\in \mathbf{S}_{I,C}}\left|\hat{V}_{I,C}\,(i,c,n)-\hat{E}_{I,C}(i,c,n)\cdot \hat{V}_{I}(i,n)\right|\right\}\quad \forall i,n \tag{68}$$

After choosing the specific variable $V_I(i,n)$ for branching from eq. 68, we need to decide where to branch it in order to divide the single node into two nodes. As discussed on branching selection by Foudas[52] and Adjiman et al.,[53] there are three possible options: (1) branching halfway between the bounds, (2) branching at the optimal solution $\hat{V}_I(i,n)$ of the node, and (3) branching 10% away from the optimal solution $\hat{V}_I(i,n)$. We follow the suggestion of Misener and Floudas[24] that branching 10% away from the optimal solution always performs at least as good as branching at the optimal solution and better than branching halfway between the bounds. Branching 10% away from the optimal solution $\hat{V}_I(i,n)$ is presented as follows

$$V_I^{\substack{\text{branching}\\ \text{point}}}(i,n)=\begin{cases} 1.1\cdot V_I(i,n) & \text{if } 1.1\cdot V_I(i,n)<\frac{1}{2}\left[V_I^{\min}(i,n)+V_I^{\max}(i,n)\right] \\ 0.9\cdot V_I(i,n) & \text{if } 0.9\cdot V_I(i,n)>\frac{1}{2}\left[V_I^{\min}(i,n)+V_I^{\max}(i,n)\right] \\ \frac{1}{2}\left[V_I^{\min}(i,n)+V_I^{\max}(i,n)\right] & \text{otherwise} \end{cases} \tag{69}$$

Solution improvement strategy

Note that any solution from pool-2 is a feasible solution (i.e., with no composition discrepancy) for the problem **MP**, and hence it provides an UB for **MP**. Li et al.[4] developed a refinement strategy to improve solution quality, and we follow their refinement strategy to improve the quality of each solution from pool-2. Let **S** denote any solution from pool-2.

Then, we can obtain a feasible schedule from **S** by solving MILPs and NLPs repeatedly. The refinement strategy is illustrated in Figure 3. First, we extract the compositions of tanks from **S** and fix them in our MINLP to get an MILP. Clearly, a solution **S**1 to this MILP is a schedule with no composition discrepancy. We take the values of the binary variables from **S**1 and fix the binary variables in our exact MINLP (involving Eqs. 15 and 16) to those values to get a NLP. We continue this alternating series of MILP and NLP until the solutions of successive NLPs converge. All improved solutions for pool-2 form another solution pool, denoted as pool-3. The best solution in pool-3 is chosen as the final UB.

Optimality-based tightening LBs and UBs

With Eqs. 46–55 and Eqs. A1–A7, we can obtain tight LBs and UBs for variables *xe* (*p*,*n*), *z* (*u*,*n*), $E_{I,C}$ (*i*,*c*,*n*), V_I (*i*,*n*), T_P^{start} (*p*), T_P^{end} (*p*), $T_{P,I}^{\text{start}}$, $T_{P,I}^{\text{end}}$, T_B^{start} (*n*), T_B^{end} (*n*), $T_{I,U}^{\text{start}}$ (*i*,*u*,*n*), $T_{I,U}^{\text{end}}$ (*i*,*u*,*n*), T_U^{start} (*u*,*n*), T_U^{end} (*u*,*n*), $V_{P,I}$ (*p*,*i*,*n*), $V_{I,U}$ (*i*,*u*,*n*), and V_U (*u*,*n*) especially for $E_{I,C}$(*i*,*c*,*n*), V_I(*i*,*n*), and $V_{I,U}$ (*i*,*u*,*n*). However, we may further tighten the LBs and UBs for variables $E_{I,C}$ (*i*,*c*,*n*), V_I (*i*,*n*), and $V_{I,U}$ (*i*,*u*,*n*) by solving minimization and maximization problems with Eqs. 1–44 and McCormick convex and concave envelopes[15] (i.e., Eqs. 66 and 67) to relax the bilinear terms (i.e., Eqs. 15 and 16 or 15′ and 16′).

The minimization and maximization problems can be stated as follows

$$\begin{aligned}
(\mathbf{MPL})\ & \text{Min } z_\text{obbt} \\
& s.t.\quad \text{eqs.}1-14,\ 17-44,\ 46-55,\ 66-67 \text{ and} \\
& \qquad \text{A1}-\text{A7}\ 0 \le X(p,i,n), Y(i,u,n) \le 1 \\
(\mathbf{MPU})\ & \text{Max } z_\text{obbt} \\
& s.t.\quad \text{eqs.}1-14, 17-44, 46-55, 66-67 \text{ and} \\
& \qquad \text{A1}-\text{A7}\ 0 \le X(p,i,n), Y(i,u,n) \le 1
\end{aligned}$$

where, *z*_obbt becomes V_I(*i*,*n*), $E_{I,C}$(*i*,*c*,*n*), and $V_{I,U}$(*i*,*u*,*n*) to update $V_I^{\min}$ (*i*,*n*), $V_I^{\max}$ (*i*,*n*), $E_{I,C}^{\min}$ (*i*,*c*,*n*), $E_{I,C}^{\max}$ (*i*,*c*,*n*), $V_{I,U}^{\min}$ (*i*,*u*,*n*), and $V_{I,U}^{\max}$ (*i*,*u*,*n*) accordingly. As $V_{I,U}^{\min}$ (*i*,*u*,*n*) can be zero at each event point in the problem addressed in this article, it is not necessary to solve the problem **MPL** for updating $V_{I,U}^{\min}$ (*i*,*u*,*n*). It is also important to note that the problems (**MPL**) and (**MPU**) are LP problems. Hence, we do not need too much computational time to solve the problems (**MPL**) and (**MPU**).

Additional strategy

When solving an MILP relaxation using the piecewise-linear underestimators at each node, we adjust the relative convergence of this MILP relaxation based on the difference between the best LBs and UBs in the tree. The MILP nodes are solved to greater accuracy as the LBs and UBs of the overall branch-and-bound tree converge. In the implementation of CPLEX[49] 11.0.0/GAMS 22.6, we require that the relative gap between the "best estimate" and the "best integer" be less than some ratio (e.g., 0.05) of the current relative gap between the LBs and UBs. We define γ to denote that ratio. In other words

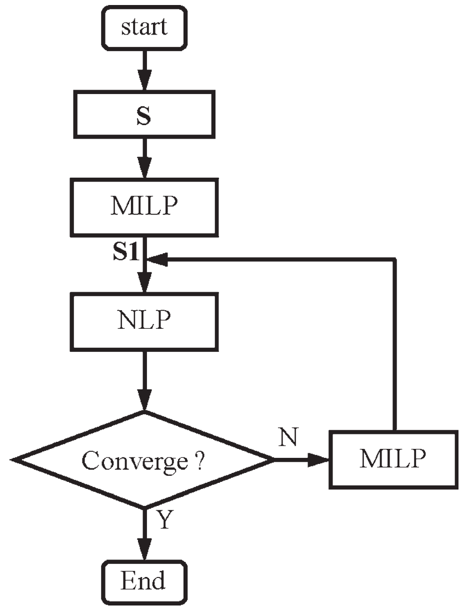

Figure 3. Flow chart for the refinement strategy.

$$\text{Relative gap} = \gamma \cdot \left| \frac{\text{UB} - \text{LB}}{\text{LB}} \right| \tag{70}$$

Note that we may not solve the MILP relaxation at each node to optimality. To avoid cutting off a feasible solution of this MILP relaxation, we always consider the best estimate (which is higher than the best integer) as the results of that node in the branch-and-bound tree. The entire procedure for the proposed branch-and-bound global optimization algorithm is illustrated in Figure 4.

Computational Studies

We take seven examples from Li et al.[4] to evaluate the performance of the proposed branch-and-bound global optimization algorithm (i.e., Examples 1, 2, 3, 5, 6, 11, and 21). Tables 2–8 provide the complete data of these examples. Table 9 presents specific sizes of all seven examples evaluated in this article. From Table 9, Example 1 is a very small example involving one VLCC, one SBM pipeline, four storage tanks, and two CDUs, with 72-h scheduling horizon. Example 2 features 17 single-parcel vessels, seven storage tanks, and two CDUs, with 56-h scheduling horizon. Examples 3, 5, and 6 have two VLCCs, eight tanks, and three CDUs, with 160-h scheduling horizon. Example 11 has three VLCCs, eight tanks, and three CDUs, with 336-h scheduling horizon. Example 21 has three jetties, two VLCCs, 15 parcels, eight tanks, and three CDUs, with 336-h scheduling horizon. Examples 1, 3, 5, 6, and 11 involve only one specification, whereas Example 21 uses 15 specifications on crude feed quality. Therefore, the selected test examples vary widely in structure, size, scale, and complexity and are representative of industrial relevant scenarios. It is important to note that the proposed formulation significantly reduces the number of bilinear terms when compared with the discrete-time model of Reddy et al.[8] and Li et al.[4] (see Table 9). For

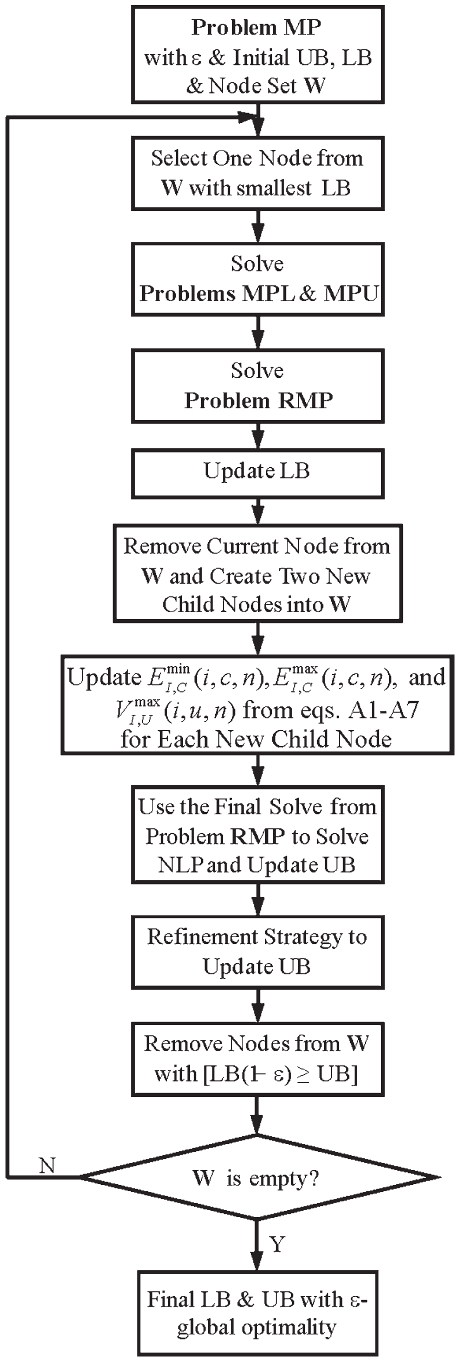

Figure 4. Flow chart for the proposed branch-and-bound global optimization algorithm.

instance, the proposed formulation involves 192 bilinear terms for Example 11, whereas it is 3280 from the discrete-time model.

We solved all examples using GAMS 22.6/CPLEX[49] 11.0.0 on Dell OPTIPLEX 960 of Intel® Xeon™ CPU 3.0 GHz with 2 GB RAM running Linux. The computational performance of the proposed branch-and-bound global optimization algorithm is presented in Table 10. It should be noted that there is no constraint for safety stock counting for Example 2, and the objective is expressed as follows

$$\text{PROFIT} = \sum_i \sum_u \sum_c \sum_n C_{\text{PROF}}(c) \cdot V_{I,U,C}(i,u,c,n) - \sum_p C_{\text{ULD}} \left[T_P^{\text{end}}(p) - T_P^{\text{start}}(p)\right] - \sum_p C_{\text{SEA}} \left[T_P^{\text{start}}(p) - T_{\text{ARR}}(p)\right] - \sum_i \sum_{u:(i,u)\in \mathbf{S}_{I,U}} \sum_{n=1}^{N-1} C_{\text{SET}} \cdot z(u,n) - C_{IVS} \cdot \frac{\sum_i \left(\sum_n V_I(i,n) + V_I^{\text{init}}(i)\right)}{N+1} \cdot H \quad (71)$$

where C_{ULD} is the unloading cost ($/h), C_{IVS} is the inventory cost ($/unit/h) and V_I^{init} (i) is the initial crude volume in tank i.

As shown in Table 10, the proposed branch-and-bound global optimization algorithm can identify the best integer feasible solution for all seven examples, which is denoted as the UB listed in Column 7. For instance, the best integer solution for Example 1 is −5122.565 k$, whereas it is −4670.103 k$ for Example 11. For Example 21, we obtain the best integer solution of −$4795.037 k$. The LB provided from the proposed branch-and-bound global optimization algorithm is listed in Column 6. The LB for Example 1 is −5193.234 k$ from the proposed branch-and-bound global optimization algorithm, whereas it is −4763.167 k$ for Example 11. The gap between the LBs and UBs is smaller than 2% for all examples, which is given in Column 8. The total CPU time for all examples from the proposed branch-and-bound global optimization algorithm is listed in Column 9. For instance, we reach a 1.361% optimality gap in 64.8 CPU seconds for Example 1; 1.941% optimality gap in 22,500 CPU seconds (i.e., 6.25 h) for Example 5; and 1.967% optimality gap in 4069 CPU seconds for Example 21. For illustration, the operational schedules for Examples 5, 11, and 21 are illustrated in Figures 5–7, respectively.

Table 2. Vessel Arrival Data for Examples 1–3, 5, 6, 11 and 21

Example	Arrival Time (h)	Vessel (Crude-Parcel Size, kbbl or kton*)
1	0	VLCC-1 (C2-10, C1-300, C4-300, C3-340)
2	1	V1 (C1-3), V2(C1-3)
	2	V3(C1-3), V4(C1-3)
	3	V5 (C1-5), V6 (C1-5), V7 (C1-3), V8 (C1-3)
	4	V9(C1-3)
	5	V10 (C2-5), V11 (C6-5), V12 (C2-3.5), V13 (C4-3.5)
	6	V14 (C1-3), V15 (C1-3)
	7	V16 (C4-3), V17 (C2-1.5, C6-1.5)
3	0	VLCC-1 (C2-10, C3-250, C4-300, C5-190)
	104	VLCC-2 (C5-10, C6-250, C3-250, C8-240)
5 and 6	0	VLCC-1 (C2-10, C3-250, C4-300, C5-190)
	104	VLCC-2 (C5-10, C6-250, C3-250, C8-240)
11	0	VLCC-1 (C2-10, C3-350, C4-200, C5-300)
	120	VLCC-2 (C5-10, C6-200, C8-250, C3-240)
	216	VLCC-3 (C3-10, C6-250, C2-250, C7-190)
21	0	VLCC-1 (C2-10, C6-100, C8-100, C4-90)
	16	V1 (C2-125)
	24	V2(C5-125), V3 (C3-100)
	32	V4(C7-120)
	160	VLCC-2 (C4-10, C8-130, C3-120, C2-100)
	176	V5(C6-100), V6(C1-90)
	184	V7(C7-125)

*kton for Example 2.

Table 3. Tank Capacities, Heels, and Initial Inventories for Examples 1–3, 5, 6, 11 and 21

	Capacity (kbbl or kton*)						Heel (kbbl or kton*)			
Tank	Ex 1	Ex 2	Ex 3	Ex 5 and 6	Ex 11	Ex 21	Ex 1	Ex 2	Ex 3, 5 and 6, 11	Ex 21
T1	700	25	570	570	570	570	50	0	60	50
T2	700	25	570	570	570	570	50	0	60	50
T3	900	25	570	570	570	570	50	0	60	50
T4	700	25	980	980	980	980	50	0	110	50
T5	–	40	980	980	980	570	–	0	110	50
T6	–	40	980	570	570	570	–	0	60	50
T7	–	40	570	570	570	570	–	0	60	50
T8	–	0	570	570	980	980	–	0	60	50

	Allowable Crude (Class)				Initial Inventory (kbbl or kton*)					
	Ex 1	Ex 2	Ex 3, 5 and 6	Ex 11, 21	Ex 1	Ex 2	Ex 3	Ex 5 and 6	Ex 11	Ex 21
T1	C1-C2(1)	C1(1)	C1-C4(1)	C1-C4(1)	300	5	350	350	400	350
T2	C3-C4 (2)	C2(1)	C5-C8 (2)	C5-C8 (2)	300	6	400	400	400	400
T3	C3-C4 (2)	C3(1)	C5-C8 (2)	C5-C8 (2)	250	7	350	350	350	350
T4	C1-C2(1)	C1(1)	C5-C8 (2)	C5-C8 (2)	300	8	950	950	950	950
T5	–	C1(1)	C5-C8 (2)	C5-C8 (2)	–	8	300	300	300	300
T6	–	C2, C4, C6 (1)	C1-C4(1)	C1-C4(1)	–	20	80	80	80	80
T7	–	C3(1)	C1-C4(1)	C1-C4(1)	–	10	80	80	80	80
T8	–	–	C1-C4(1)	C1-C4(1)	–	0	450	450	450	450

*kton for Example 2

From Figure 6, although VLCC parcels P5, P6, and P12 are unloaded into Tank 5 at event point N2, they are unloaded one after another. If we enforce a tank to receive at most one VLCC parcel at each event point, then we need five event points to obtain the same schedule. The same observation can be made from Figure 7. Tank 5 receives VLCC parcels P2 and P3 at the same event point N1, but at different times. If we impose a tank to receive at most one VLCC parcel at each event point, then we need at least four event points to obtain the same schedule. Thus, allowing a tank to receive multiple VLCC parcels at each event point can significantly reduce the number of event points.

As discussed in the section "Mathematical Formulation," there are some differences between the proposed unit-specific event-based continuous-time model and the discrete-time models proposed and enhanced by Reddy et al.[8] and Li et al.[4] To compare our proposed model with the model of Li et al.[4] fairly.

First, we remove the constraints (i.e., Eqs. 72 and 73) from the model of Li et al.[4] that force the flow rates from respective tanks to be constant in contiguous periods to avoid changeovers caused by the tank-to-CDU flow changes and to control period-to-period changes in crude feed flow rate. These constraints are presented as follows

$$M\left[2-\sum_i YY_{iut}\right]+\mathrm{FTU}_{iut}\geq \mathrm{FTU}_{iu(t+1)} \qquad (i,u)\in \mathbf{IU} \tag{72a}$$

$$M\left[2-\sum_i YY_{iut}\right]+\mathrm{FTU}_{iut(t+1)}\leq \mathrm{FTU}_{iut} \qquad (i,u)\in \mathbf{IU} \tag{72b}$$

$$\gamma_u^L \mathrm{FU}_{ut}\leq \mathrm{FU}_{u(t+1)}\leq \gamma_u^U \mathrm{FU}_{ut} \tag{73}$$

where YY_{iut} is a 0–1 continuous variable to denote if a tank i is connected to CDU u during both periods t and $(t+1)$; FTU_{iut} denotes total amount of crude fed from tank i to CDU u during period t; M is a large number; FU_{ut} denotes total amount of crude fed to CDU u during period t; and γ_u^L and γ_u^U denote control period-to-period changes in crude feed flows.

Second, the changeover defined in this article is also applied to the model of Li et al.[4] Thus, the following constraints (i.e., Eqs. 74 and 75) from Reddy et al.[8] and Li et al.[4] change to Eqs. 76a,b

$$YY_{iut}\geq Y_{iut}+Y_{iu(t+1)}-1 \qquad (i,u)\in \mathbf{IU}, t<T \tag{74a}$$

Table 4. Initial Crude Amounts (kbbl or kton*) for Examples 1–3, 5, 6, 11, and 21

	Ex 1		Ex 2				Ex 3, 5, 6, and 21				Ex 11			
Tank	C1 or C3	C2 or C4	C1 or C5	C2	C3 or C6	C4	C1 or C5	C2 or C6	C3 or C7	C4 or C8	C1 or C5	C2 or C6	C3 or C7	C4 or C8
T1	200	100	5	–	–	–	50	100	100	100	100	100	100	100
T2	100	200	–	6	–	–	100	100	100	100	100	100	100	100
T3	50	200	–	–	7	–	100	100	50	100	100	100	50	100
T4	130	170	8	–	–	–	200	250	200	300	200	250	200	300
T5	–	–	8	–	–	–	100	100	50	50	100	100	50	50
T6	–	–	–	10	5	5	20	20	20	20	20	20	20	20
T7	–	–	–	–	10	–	20	20	20	20	20	20	20	20
T8	–	–	–	–	–	–	100	100	100	150	100	100	100	150

*kton for Example 2.

Table 5. Crude Concentration Ranges in Tanks and CDUs for Examples 1–3, 5, 6, 11 and 21

Example	Crude	Concentration Range (Min–Max)	Tank and CDU
1	C1-C2	0.20–0.80	T1, T4
	C3-C4	0.20–0.80	T2
	C1-C2	0.30–0.70	CDU1
	C3-C4	0.00–1.00	T3, CDU2
2	C1-C3	1.00–1.00	T1, T2, T3 and T4-T5
	C2, C4, C6	0.00–1.00	T6
3	C1-C8	0.00–1.00	T1-T8
	C5-C8	0.00–1.00	CDU1 and CDU2
	C1	0.15–0.85	CDU3
	C2-C4	0.00–1.00	CDU3
5, 6, and 21	C1-C4	0.00–1.00	T1, T6-T8 and CDU3
	C5-C8	0.00–1.00	T2-T5, CDU1 and CDU2
11	C1-C8	0.00–1.00	T1-T8
	C1-C4	0.10–0.90	CDU3
	C5-C8	0.10–0.90	CDU1 and CDU2

$$YY_{iut} \le Y_{iu(t+1)} \qquad (i,u) \in \mathbf{IU}, t<T \qquad (74b)$$

$$YY_{iut} \le Y_{iut} \qquad (i,u) \in \mathbf{IU}, \qquad (74c)$$

$$CO_{ut} \ge Y_{iut} + Y_{iu(t+1)} - 2YY_{iut} \qquad (i,n) \in \mathbf{IU}, t<T \qquad (75)$$

where Y_{iut} is a binary variable to denote if a tank i feeds CDU u during period t, and CO_{ut} is a 0–1 continuous variable to denote if a CDU u has a changeover during period t

$$CO_{ut} \ge Y_{iut} - Y_{iu(t+1)} \qquad (i,u) \in \mathbf{IU}, t<T \qquad (76a)$$

$$CO_{ut} \ge Y_{iu(t+1)} - Y_{iut} \qquad (i,u) \in \mathbf{IU}, t<T \qquad (76b)$$

Third, the safety stock is also calculated in a similar approach as in this article, not checked at the end of each time period.

$$SC \ge SSP \cdot \left\{ SS - \frac{\sum_i \left(\sum_t VS_{it} + V0_i \right)}{T+1} \right\} \cdot T \qquad (77)$$

where SC denotes the safety stock penalty; SSP is safety stock penalty ($ per unit volume per period below desired safety stock); VS_{it} is crude level in tank i at the end of period t; and $V0_i$ is initial crude amount in tank i.

Finally, the objective of Li et al.[4] changes to

$$\text{Profit} = \sum_i \sum_u \sum_c \sum_t FCTU_{iuct} CP_{cu} - \sum_v DC_v - COC \sum_u \sum_t CO_{ut} - SC \qquad (78)$$

where $FCTU_{iuct}$ denotes the amount of crude c fed from tank i to CDU u during period t; CP_{cu} is profit margin ($/unit volume) for crude c in CDU u; DC_v denotes demurrage cost for vessel v; COC is cost (k$) per changeover; and CO_{ut} denotes if a CDU u has a changeover during period t. Note that the objective function for Example 2 is Eq. 71. The calculation

Table 6. Transfer Rates, Processing Limits, Operating Costs, Crude Margins, and Demand for Examples 1–3, 5 and 6, 11, and 21

Example	Flow Rate Limits (kbbl or kton* per 8 h): Parcel-Tank Min Max	Flow Rate Limits (kbbl or kton* per 8 h): Tank-CDU Min-Max	Demurrage (k$ per 8 h) or Sea Waiting Cost* (kYuan per 8 h)	Changeover Loss (k$ or k Yuan* per instance)	Inventory Penalty ($/bbl per 8 h)	Inventory Cost (Yuan/ton per 8 h)	Unloading Cost (k Yuan per 8 h)	Desired Safety Stock (kbbl)	Crude	Margin ($/bbl or kYuan/ton*): Ex 1	Ex 2	Ex 3 and11	Ex 5	Ex 6	Ex 21
1	10–400	0–100	100	5	0.2	–	–	1200	C1	3.0	1	1.50	1.50	1.50	1.50
2	0–5	0–5	5	1	–	0.05	7	–	C2	4.5	1	1.70	1.70	1.50	1.75
3	10–400	20–45	25	10	0.20	–	–	1500	C3	5.0	1	1.50	1.50	1.50	1.85
5-6	10–400	20–45	25	10	0.20	–	–	1500	C4	6.0	1	1.60	1.60	1.50	1.25
11	10–400	20–40	25	10	0.02	–	–	1500	C5	–	1	1.45	1.45	1.50	1.45
21	10–250	20–250	15	5	0.20	–	–	1200	C6	–	1	1.60	1.60	1.50	1.65
–	–	–	–	–	–	–	–	–	C7	–	–	1.55	1.55	1.50	1.55
–	–	–	–	–	–	–	–		C8	–	–	1.60	0.50	1.50	1.60

CDU	Processing Limits (kbbl or kton* per 8 h): Ex 1	Ex 2	Ex 3-6	Ex 11	Ex 21	Total Demand (kbbl): Ex 1	Ex 3	Ex 5 and 6	Ex 11	Ex 21					
											–	–	–	–	–
CDU 1	50–100	2–8	20–45	20–40	20–40	550	750	750	1000	1000	–	–	–	–	–
CDU 2	50–100	1–3	20–45	20–40	20–40	550	750	750	1000	1000	–	–	–	–	–
CDU 3	–	–	20–45	20–40	20–40	–	750	750	1000	1000	–	–	–	–	–

*For Example 2.

Table 7. Specific Gravities, Sulfur Contents, Nitrogen Contents, Carbon Residues, Pour Point, Freeze Point, and Flash Point for Crudes and Acceptable Ranges for Feeds to CDUs for Examples 1–3, 5, 6, 11 and 21

Crude and CDU	Specific Gravity	Sulfur				Nitrogen	Carbon Residue	Pour Point	Freeze Point	Flash Point
	Ex 21	Ex 1	Ex 3, 5 and 6	Ex 11	Ex 21	Ex 21	Ex 21	Ex 21	Ex 21	Ex 21
C1	1.2057	0.01	0.0020	0.0020	0.0095	55.00	0.0450	58.0549	270.2996	207.7017
C2	1.2339	0.01	0.0025	0.0025	0.0085	45.00	0.0420	12.0466	211.3251	551.5897
C3	1.2113	0.02	0.0015	0.0015	0.0080	50.00	0.0436	21.8409	248.0304	311.3055
C4	1.2749	0.01	0.0060	0.0060	0.0090	40.00	0.0350	10.3347	168.4381	661.2327
C5	1.0375	–	0.0120	0.0120	0.0250	93.00	0.1880	5.1896	1412.5240	16.5062
C6	1.0615	–	0.0130	0.0130	0.0235	88.00	0.1730	4.6626	1286.6348	21.3079
C7	1.0664	–	0.0090	0.0090	0.0225	84.00	0.1540	48.4716	1015.0334	29.5074
C8	1.0968	–	0.0150	0.0150	0.0210	78.00	0.1260	7.5624	768.6957	39.4486
CDU1										
Min	1.0000	0.00	0.0010	0.0010	0.0200	75.00	0.1000	4.0000	700.0000	15.0000
Max	1.0920	0.01	0.0130	0.0135	0.0242	92.00	0.1800	45.0000	1405.0000	39.0000
CDU2										
Min	1.0000	0.01	0.0010	0.0010	0.0200	75.00	0.1000	4.0000	700. 0000	15.0000
Max	1.0900	0.02	0.0125	0.0130	0.0245	91.50	0.1850	48.0000	1410.0000	39.2000
CDU3										
Min	1.2000	–	0.0010	0.0010	0.0060	10.00	0.0100	10.0000	150.0000	200.0000
Max	1.2700	–	0.0035	0.0040	0.0092	54.00	0.0440	58.0000	270.0000	650.0000

Table 8. Smoke Points, Ni Contents, and Reid Vapor Pressures for Crudes and Acceptable Ranges for Feeds to CDUs for Example 21

	Smoke Point	Ni	Reid Vapor Pressure	Asphaltenes	Aromatics	Paraffins	Naphthenes	Viscosity
Crude & CDU	Ex 21	Ex 21	Ex 21	Ex 21	Ex 21	Ex 21	Ex 21	Ex 21
C1	548.1218	0.075	153.6366	0.0850	0.2972	0.3844	0.3414	76.8625
C2	588.8047	0.062	120.4380	0.0650	0.2793	0.4222	0.3203	76.2073
C3	567.6004	0.050	144.8884	0.0500	0.2756	0.3614	0.3022	75.7175
C4	626.5365	0.035	113.5842	0.0700	0.2713	0.4004	0.3443	76.5457
C5	431.4538	19.000	24.1774	0.2000	0.5216	0.2400	0.2384	82.6218
C6	455.4485	18.300	22.5324	0.1890	0.4942	0.3244	0.2302	81.5636
C7	477.3611	17.500	21.1838	0.1750	0.4577	0.2756	0.2407	81.1988
C8	503.5443	16.700	13.8983	0.1500	0.4317	0.3016	0.2439	80.3514
CDU1								
Min	400.0000	15.000	10.0000	0.1500	0.4000	0.2400	0.2000	80.0000
Max	475.0000	18.800	24.0000	0.1960	0.5000	0.3200	0.2420	82.5000
CDU2								
Min	400.0000	15.000	10.0000	0.1500	0.4000	0.2400	0.2000	80.0000
Max	470.0000	18.600	23.9000	0.1950	0.5200	0.3200	0.2410	82.6000
CDU3								
Min	500.0000	0.010	100.0000	0.0500	0.2500	0.3500	0.3000	70.0000
Max	600.0000	0.072	150.0000	0.0800	0.2950	0.4200	0.3440	76.8000

Table 9. Sizes of Examples 1–3, 5, 6, 11 and 21

Example	SBM	Jetties	VLCCs	Single-Parcel Vessels	Parcels	Tanks	CDUs	Crude Properties	Discrete Variables	Continuous Variables	Constraints	Bilinear Terms	
												Proposed Model	Discrete-Time Model
1	1	0	1	0	4	4	2	1	48	913	1408	56	132
2	0	4	0	17	17	7	2	–	105	1628	5596	57	126
3	1	0	2	0	8	8	3	1	116	1425	3799	192	1488
5	1	0	2	0	8	8	3	1	116	2265	3849	192	1488
6	1	0	2	0	8	8	3	1	116	1401	3273	192	1488
11	1	0	3	0	12	8	3	1	132	1622	4946	192	3280
21	1	3	2	7	15	8	3	15	160	1765	6122	192	3312

Discrete-time model from Reddy et al.[8] and Li et al.[4] (Scheduling horizon for Example 1: 72 h; Example 2: 56 h, Examples 3, 5 and 6: 160 h; Examples 11 and 21: 336 h.

Table 10. Performance of the Proposed Branch and Bound Global Optimization Algorithm for Examples 1–3, 5, 6, 11, and 21

Example	GR	Event Points	γ	$\varepsilon = 0.02$				
				Nodes	LB (k$)	UB (k$)	Gap (%)	CPU Time (s)
1	7	4	0.05	3	−5193.234	−5122.565	1.361	64.8
2	4	5	0.05	1	−1.0251E+08	−1.0156 E +08	0.920	11.3
3	2	3	0.05	13	−3553.582	−3483.648	1.968	4012
5	7	3	0.05	3	−3194.658	−3132.658	1.941	22500
6	2	3	0.05	13	−3375.000	−3355.000	0.593	445
11	2	3	0.50	3	−4763.167	−4670.103	1.954	212
21	2	3	0.50	15	−4891.242	−4795.037	1.967	4069

Branch-and-bound tree limited to 360,000 CPU seconds (100 h). The relative gap for root node is 0.03.

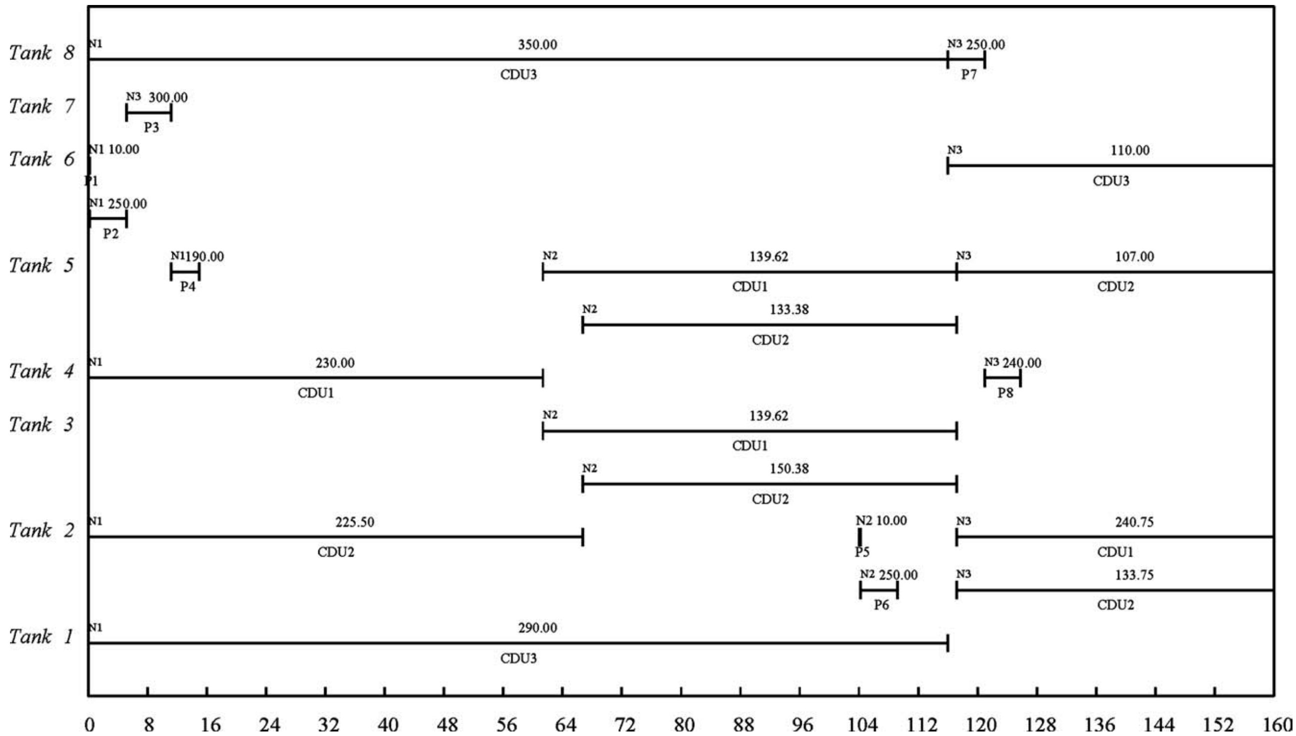

Figure 5. Operation schedule for Example 5 from the proposed branch-and-bound global optimization algorithm.

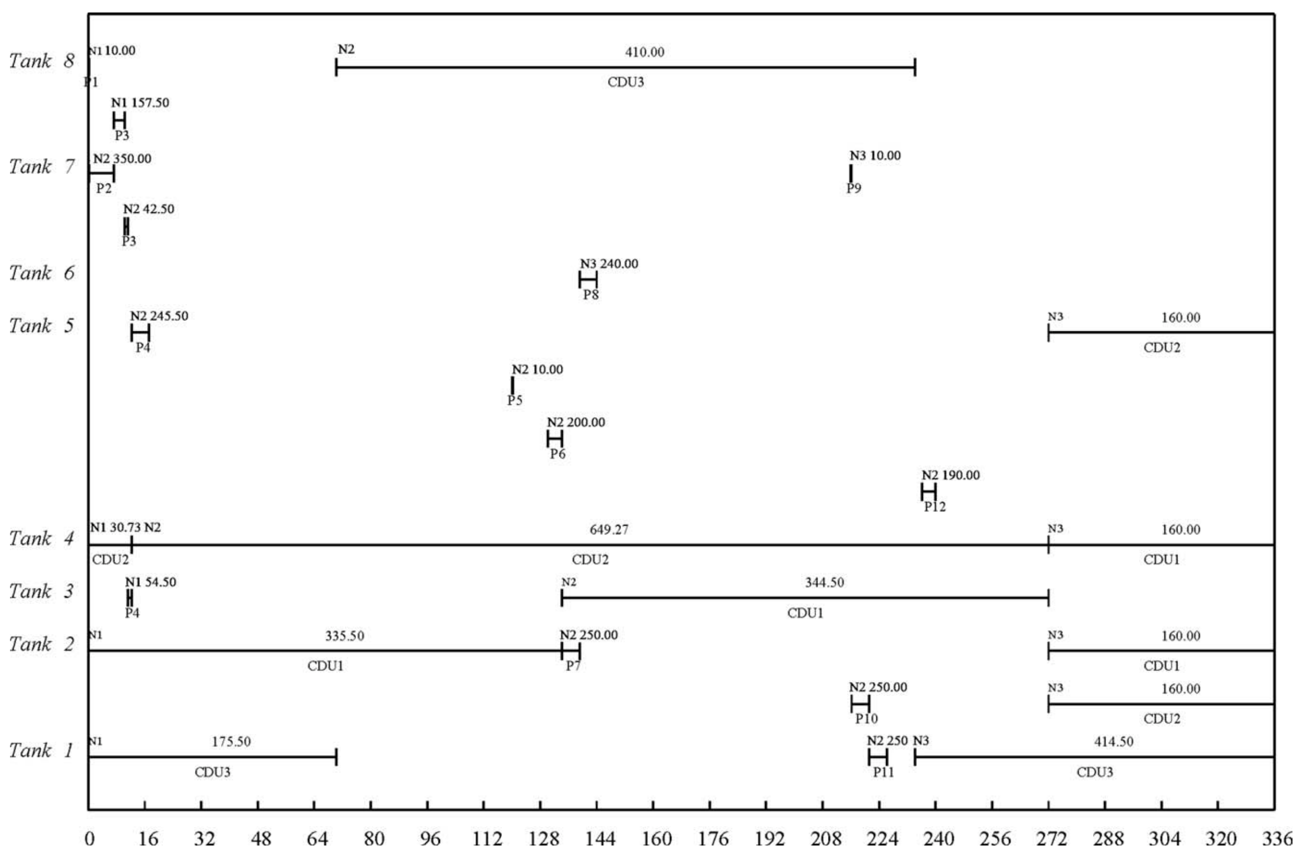

Figure 6. Operation schedule for Example 11 from the proposed branch-and-bound global optimization algorithm

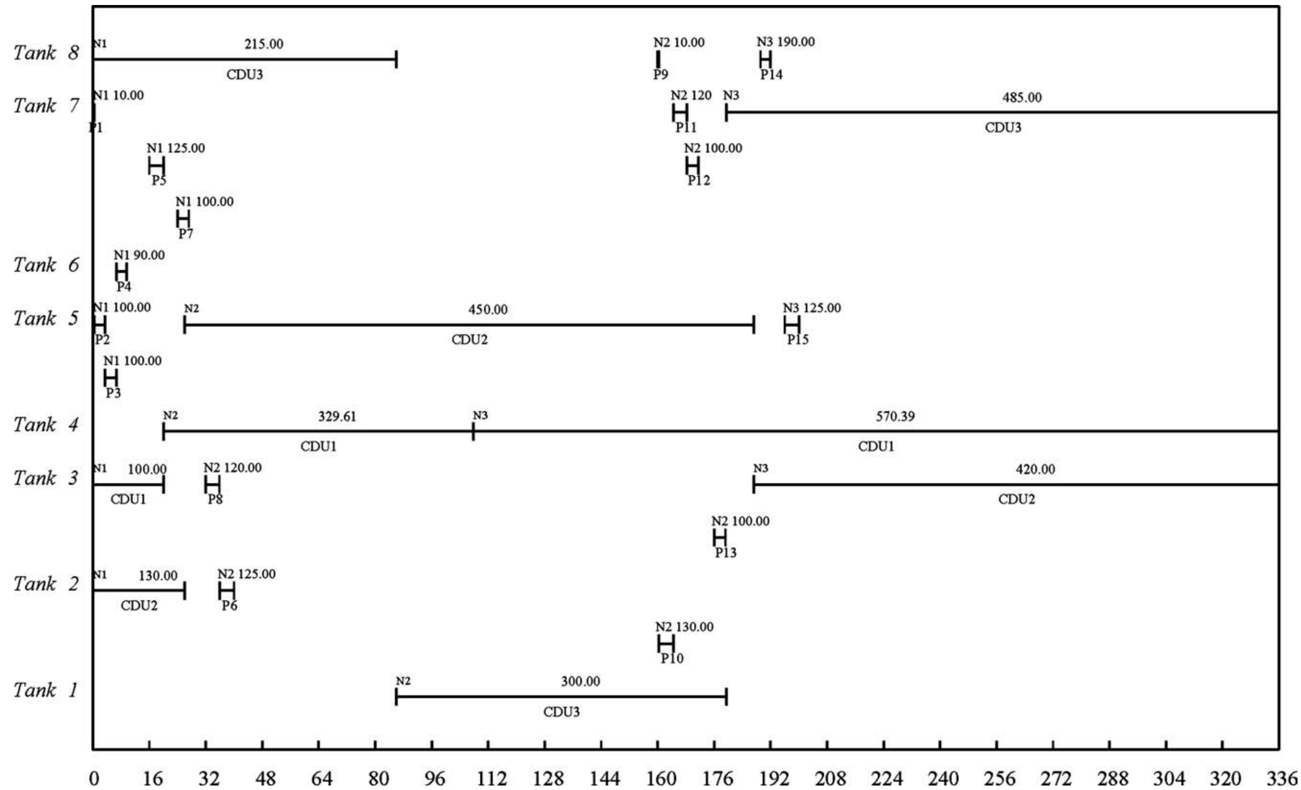

Figure 7. Operation schedule for Example 21 from the proposed branch-and-bound global optimization algorithm.

of demurrage cost is on a per unit time basis and, thus, more accurate than that of Reddy et al.[8] and Li et al.[4]

We solve those seven examples using the revised model of Li et al.[4] following their approach. In their approach, we set upper limits on CPU times for all examples, which are 30,000 CPU seconds for UB, and 172,800 CPU seconds for LB. The comparative results are presented in Table 11. We obtain the same or better integer solutions (i.e., UB) for all seven examples from the proposed branch-and-bound global optimization algorithm when compared with those using the approach of Li et al.[4] For instance, we obtain an integer feasible solution of −5122.565 k$ for Example 1 from the proposed branch-and-bound global optimization algorithm, which is better compared with that of −5117.454 k$ from the approach of Li et al.[4] Moreover, we identify a better solution of −4795.037 k$ for Example 21 from the proposed branch-and-bound global optimization algorithm than that of −4765.258 k$ from the approach of Li et al.[4] We also obtain better LBs for Examples 1, 5, 11, and 21 than those of Li et al.[4] regardless of larger CPU times for the approach of Li et al.[4] A reduction in the gap is achieved for Examples 1, 5, 11, and 21. The gap for Example 5 is reduced from 6.842% from the approach of Li et al.[4] to 1.941%, which is a significant reduction. Although the gap for Example 3 is reduced to below 2% using both approaches, the approach of Li et al.[4] needs 202,800 CPU seconds (i.e., 2.35 days), which is much larger than 4012 CPU seconds from the proposed branch-and-bound global optimization algorithm. Most importantly, the total CPU times for Examples 2, 3, 5, 6, 11, and 21 are reduced significantly from the proposed models and branch-and-bound global optimization algorithm. For instance, we need 22,500 CPU seconds for Example 5 to achieve the gap of 1.941% when compared with 202,800 CPU seconds to achieve the gap of 6.842% from the approach of Li et al.[4] The approach of Li et al.[4] needs huge CPU time to obtain tighter LB. In addition, both approaches

Table 11. Comparative Results of Lower and Upper Bounds from the Proposed Approach and Revised Model of Li et al.[4]

	Branch-and-Bound Global Optimization Algorithm				Revised Model of Li et al.[4]			
Example	LB (k$)	UB (k$)	Gap (%)	CPU Time (s)	LB (k$)	UB (k$)	Gap (%)	CPU Time (s)
1	−5193.234	−5122.565	1.361	64.8	−5193.667	−5117.454	1.468	19.3
2	−1.0251 E +08	−1.0156 E +08	0.720	11.3	−1.0162 E +08	−1.0146 E +08	0.158	30,005
3	−3553.582	−3483.648	1.968	4012	−3531.063	−3465.280	1.863	202,800
5	−3194.658	−3132.658	1.941	22500	−3361.191	−3131.235	6.842	202,800
6	−3375.000	−3355.000	0.593	445	−3355.000	−3355.000	0.000	44,588
11	−4763.167	−4670.103	1.954	212	−4766.004	−4654.743	2.335	202,800
21	−4891.242	−4795.037	1.967	4069	−4937.748	−4765.258	3.494	202,800

CPU time limits for UB and LB in revised model of Li et al.[4] are 30,000 and 172,800 CPU seconds, respectively.

can find the global optimal solution for Example 6, which is −3355.00 k$.

Conclusions

In this article, we developed a novel unit-specific event-based continuous-time MINLP formulation for crude oil scheduling problems. We incorporated many realistic operational features such as SBM, multiple jetties, multiparcel vessels, single-parcel vessels, crude blending, brine settling, crude segregation, and multiple tanks feeding one CDU at one time and *vice versa*. The proposed model significantly reduced the number of bilinear terms and problem size when compared with the discrete-time formulation of Reddy et al.[8] and Li et al.[4] The computational results show that the developed branch-and-bound global optimization algorithm with piecewise-linear underestimation[16–26] was effective to address all tested examples and resulted in better integer feasible solutions. These integer feasible solutions were guaranteed to be within 2% of global optimality.

Acknowledgments

The authors gratefully acknowledge the support from the National Science Foundation (CMMI-08856021). Ruth Misener is further thankful for her National Science Foundation Graduate Research Fellowship.

Notation

Sets

VP = VLCC parcels
JP = jetty parcels
$\mathbf{S}_{F,I}$ = set of pairs (tank i, event point n) that the concentration of tank i is the same as its initial composition during the possible event point n
$\mathbf{S}_{P,I}$ = set of pairs (parcel p, tank i) that tank i can receive parcel p
$\mathbf{S}_{I,C}$ = set of pairs (tank i, crude c) that tank i can hold crude c
$\mathbf{S}_{I,U}$ = set of pairs (tank i, CDU u) that tank i can feed CDU u
$\mathbf{S}_{U,C}$ = set of pairs (CDU u, crude c) that CDU u can process crude c

Parameters

Δn = 1 if a parcel is unloaded in multiple event points
C_{IVS} = inventory cost ($/unit/h)
C_{PEN} = safety stock penalty ($/unit/h)
$C_{\mathrm{PROF}}(c)$ = marginal profit ($/unit volume) from crude c
C_{SEA} = demurrage or sea-waiting cost
C_{SET} = cost (k$) per changeover
C_{ULD} = unloading cost ($/hr)
$D_U^{\min}(u)$ = minimum allowable crude processing rate of CDU u
$D_U^{\max}(u)$ = maximum allowable crude processing rate of CDU u
$D(u)$ = demand of each CDU u
$E_{I,C}^{\min}(i,c)$ = lower limit on the composition of crude c in tank i
$E_{I,C}^{\min}(i,c,n)$ = lower limit on the composition of crude c in tank i at event point n
$E_{I,C}^{\max}(i,c)$ = upper limit on the composition of crude c in tank i
$E_{I,C}^{\max}(i,c,n)$ = upper limit on the composition of crude c in tank i at event point n
$E_P(p,c)$ = fraction of crude c in parcel p
$E_U^{\min}(u,c)$ = minimum allowable composition of crude c in feed to CDU u
$E_U^{\max}(u,c)$ = maximum allowable composition of crude c in feed to CDU u
$e_C(c,k)$ = index of property k in crude c
$e_U^{\min}(u,k)$ = minimum allowable index of property k in CDU u
$e_U^{\max}(u,k)$ = maximum allowable index of property k in CDU u
$F_{I,U}^{\min}(i,u)$ = minimum feeding rate of crude from tank i to CDU u
$V_{I,U}^{\min}(i,u,n)$ = minimum feeding amount of crudes from tank i to CDU u at event point n
$F_{I,U}^{\max}(i,u)$ = maximum feeding rate of crude from tank i to CDU u
$V_{I,U}^{\max}(i,u,n)$ = maximum feeding amount of crudes from tank i to CDU u at event point n
$F_{P,I}^{\min}(p,i)$ = minimum unloading rate of crude from parcel p to tank i
$F_{P,I}^{\max}(p,i)$ = maximum unloading rate of crude from parcel p to tank i
$V_{P,I}^{\max}(p,i,n)$ = maximum unloading amount of crude from parcel p to tank i at event point n
H = scheduling horizon
SS = desired safety stock
ST = minimum time for crude settling and brine removal
$T_{\mathrm{ARR}}(p)$ = expected arrival time of parcel p
$V_I^{\mathrm{init}}(i)$ = initial crude volume in tank i
$V_{I,C}^{\mathrm{init}}(i,c)$ = initial amount of crude c in tank i
$V_P^{\mathrm{init}}(p)$ = initial crude volume of parcel p
$T_{\mathrm{ULD}}^{\min}(v)$ = stipulated departure time in the logistics contract for each vessel v
$V_I^{\min}(i)$ = minimum allowable crude inventory in tank i
$V_I^{\min}(i,n)$ = minimum allowable crude inventory in tank i at event point n
$V_I^{\max}(i)$ = maximum allowable crude inventory in tank i
$V_I^{\max}(i,n)$ = maximum allowable crude inventory in tank i at event point n

Binary variables

$X(p,i,n)$ = 1 if parcel p is unloaded to tank i during event point n
$Y(i,u,n)$ = 1 if tank i feeds CDU u during event point n

0–1 Continuous variables

$xe(p,n)$ = 1 if parcel p is completed at the end of event point n
$z(u,n)$ = 1 if a tank switch on CDU u takes place at the end of event n

Positive variables

$E_{I,C}(i,c,n)$ = composition of crude c in tank i at the end of event point n
SSP = average safety stock at the end of each event point
$T_B^{\mathrm{start}}(n)$ = start time of event point n on jetties
$T_B^{\mathrm{end}}(n)$ = end time of event point n on jetties
$T_{\mathrm{CW}}(v)$ = demurrage cost of vessel v
$T_{I,U}^{\mathrm{start}}(i,u,n)$ = start time that tank i feeds CDU u during event point n
$T_{I,U}^{\mathrm{end}}(i,u,n)$ = end time that tank i feeds CDU u during event point n
$T_P^{\mathrm{start}}(p)$ = start time for parcel p unloading
$T_P^{\mathrm{end}}(p)$ = end time for parcel p unloading
$T_{P,I}^{\mathrm{start}}(p,i,n)$ = start time that parcel p is unloaded to tank i during event point n
$T_{P,I}^{\mathrm{end}}(p,i,n)$ = end time that parcel p is unloaded to tank i during event point n
$T_U^{\mathrm{start}}(u,n)$ = start time of event point n on CDU u
$T_U^{\mathrm{end}}(u,n)$ = end time of event point n on CDU u
$V_{P,I}(p,i,n)$ = crude amount transferred from parcel p to tank i during event point n
$V_I(i,n)$ = crude volume in tank i at the end of event point n
$V_{I,C}(i,c,n)$ = volume of crude c in tank i at the end of event point n
$V_{I,U,C}(i,u,c,n)$ = amount of crude c fed from tank i to CDU u during event point n
$V_{I,U}(i,u,n)$ = amount of crude that tank i feeds to CDU u during event point n
$V_U(u,n)$ = total amount of crudes fed to CDU u during event point n

Literature Cited

1. Pinto JM, Joly M, Moro L. Planning and scheduling models for refinery operations. *Comput Chem Eng*. 2000;24:2259–2276.
2. Li J, Karimi IA, Srinivasan R. Recipe determination and scheduling of gasoline blending operations. *AIChE J*. 2010;56:441–465.

3. Kelly JD, Mann JL. Crude-oil blend scheduling optimization: an application with multi-million dollar benefits—Part 1. *Hydrocarbon Process*. 2003;82:47–53.
4. Li J, Li WK, Karimi IA, Srinivasan R. Improving the robustness and efficiency of crude scheduling algorithms. *AIChE J*. 2007;53: 2659–2680.
5. Shah N. Mathematical programming techniques for crude oil scheduling. *Comput Chem Eng*. 1996;20:S1227–S1232.
6. Lee H, Pinto JM, Grossmann IE, Park S. Mixed-integer linear programming model for refinery short-term scheduling of crude oil unloading with inventory management. *Ind Eng Chem Res*. 1996;35:1630–1641.
7. Li W, Hui CW, Hua B, Zhongxuan T. Scheduling crude oil unloading, storage, and processing. *Ind Eng Chem Res*. 2002;41:6723–6734.
8. Reddy PCP, Karimi IA, Srinivasan R. Novel solution approach for optimizing crude oil operations. *AIChE J*. 2004;50:1177–1197.
9. Jia ZY, Ierapetritou M, Kelly JD. Refinery short-term scheduling using continuous-time formulation: crude-oil operations. *Ind Eng Chem Res*. 2003;42:3085–3097.
10. Reddy PCP, Karimi IA, Srinivasan R. A new continuous-time formulation for scheduling crude oil operations. *Chem Eng Sci*. 2004;59:1325–1341.
11. Moro LFL, Pinto JM. Mixed-integer programming approach for short-term crude oil scheduling. *Ind Eng Chem Res*. 2004;43:85–94.
12. Karuppiah R, Furman KC, Grossmann IE. Global optimization for scheduling refinery crude oil operations. *Comput Chem Eng*. 2008;32:2745–2766.
13. Pan M, Li XX, Qian Y. New approach for scheduling crude oil operations. *Chem Eng Sci*. 2009;64:965–983.
14. Mouret S, Grossmann IE, Pestiaux P. A novel priority-slot based continuous-time formulation for crude-oil scheduling problems. *Ind Eng Chem Res*. 2009;48:8515–8528.
15. McCormick GP. Computability of global solutions to factorable nonconvex programs. I. Convex underestimating problems. *Math Program*. 1976;10:146–175.
16. Meyer CA, Floudas CA. Global optimization of a combinatorially complex generalized pooling problem. *AIChE J*. 2006;52:1027–1037.
17. Karuppiah R, Grossmann IE. Global optimization for the synthesis of integrated water systems in chemical processes. *Comput Chem Eng*. 2006;30:650–673.
18. Wicaksono DS, Karimi IA. Piecewise MILP under- and overestimators for global optimization of bilinear programs. *AIChE J*. 2008;54:991–1008.
19. Gounaris CE, Misener R, Floudas CA. Computational comparison of piecewise-linear relaxations for pooling problems. *Ind Eng Chem Res*. 2009;48:5742–5766.
20. Bergamini ML, Grossmann IE, Scenna N, Aguirre P. An improved piecewise outer-approximation algorithm for the global optimization of MINLP models involving concave and bilinear terms. *Comput Chem Eng*. 2008;32:477–493.
21. Saif Y, Elkamel A, Pritzker M. Global optimization for reverse osmosis network for wastewater treatment and minimization. *Ind Eng Chem Res*. 2008;47:3060–3070.
22. Pham V, Laird C, El-Halwagi M. Convex hull discretization approach to the global optimization of pooling problems. *Ind Eng Chem Res*. 2009;48:1973–1979.
23. Misener R, Floudas CA. Advances for the pooling problem: modeling, global optimization, and computational studies. *Appl Comput Math*. 2009;8:3–22.
24. Misener R, Floudas CA. Global optimization of large-scale generalized pooling problems: quadratically constrained MINLP models. *Ind Eng Chem Res*. 2010;49:5424–5438.
25. Hasan MMF, Karimi IA. Piecewise linear relaxation of bilinear programs using bivariate partitioning. *AIChE J*. 2010;56:1880–1893.
26. Misener R, Gounaris CE, Floudas CA. Mathematical modeling and global optimization of large-scale extended pooling problems with the (EPA) complex emissions constraints. *Comput Chem Eng*. 2010;34:1432–1456.
27. Floudas CA, Lin X. Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. *Comput Chem Eng*. 2004;8:2109–2129.
28. Floudas CA, Lin X. Mixed integer linear programming in process scheduling: modeling, algorithms, and applications. *Ann Oper Res*. 2005;139:131–162.
29. Ierapetritou MG, Floudas CA. Effective continuous-time formulation for short-term scheduling. I. Multipurpose batch processes. *Ind Eng Chem Res*. 1998;37:4341–4359.
30. Ierapetritou MG, Floudas CA. Effective continuous-time formulation for short-term scheduling. II. Continuous and semi-continuous process. *Ind Eng Chem Res*. 1998;37:4360–4374.
31. Ierapetritou MG, Hene TS, Floudas CA. Effective continuous-time formulation for short-term scheduling. III. Multiple intermediate due dates. *Ind Eng Chem Res*. 1999;38:3446–3461.
32. Lin X, Floudas CA. Design, synthesis and scheduling of multipurpose batch plants via an effective continuous-time formulation. *Comput Chem Eng*. 2001;25:665–674.
33. Lin X, Floudas CA, Modi S, Juhasz NM. Continuous-time optimization approach for medium-range production scheduling of a multiproduct batch plant. *Ind Eng Chem Res*. 2002;41:3884–3906.
34. Lin X, Chajakis ED, Floudas CA. Scheduling of tanker lightering via a novel continuous-time optimization framework. *Ind Eng Chem Res*. 2003;42:4441–4451.
35. Lin X, Janak SL, Floudas CA. A new robust optimization approach for scheduling under uncertainty. I. Bounded uncertainty. *Comput Chem Eng*. 2004;28:1069–1085.
36. Janak SL, Lin X, Floudas CA. Enhanced continuous-time unit-specific event-based formulation for short-term scheduling of multipurpose batch processes: resource constraints and mixed storage policies. *Ind Eng Chem Res*. 2004;43:2516–2533.
37. Janak SL, Lin X, Floudas CA. Comments on "Enhanced continuous-time unit-specific event-based formulation for short-term scheduling of multipurpose batch processes: resource constraints and mixed storage policies". *Ind Eng Chem Res*. 2005;44:426.
38. Janak SL, Floudas CA, Kallrath J, Vormbrock N. Production scheduling of a large-scale industrial batch plant. I. Short-term and medium-term scheduling. *Ind Eng Chem Res*. 2006;45:8234–8252.
39. Janak SL, Floudas CA, Kallrath J, Vormbrock N. Production scheduling of a large-scale industrial batch plant. II. Reactive scheduling. *Ind Eng Chem Res*. 2006;45:8253–8269.
40. Janak SL, Lin X, Floudas CA. A new robust optimization approach for scheduling under uncertainty. II. Uncertainty with known probability distribution. *Comput Chem Eng*. 2007;31:171–195.
41. Janak SL, Floudas CA. Improving unit-specific event based continuous-time approaches for batch processes: integrality gap and task splitting. *Comput Chem Eng*. 2008;32:913–955.
42. Shaik MA, Janak SL, Floudas CA. Continuous-time models for short-term scheduling of multipurpose batch plants: a comparative study. *Ind Eng Chem Res*. 2006;45:6190–6209.
43. Shaik MA, Floudas CA. Improved unit-specific event-based model continuous-time model for short-term scheduling of continuous processes: rigorous treatment of storage requirements. *Ind Eng Chem Res*. 2007;46:1764–1779.
44. Shaik MA, Floudas CA. Unit-specific event-based continuous-time approach for short-term scheduling of batch plants using RTN framework. *Comput Chem Eng*. 2008;32:260–274.
45. Shaik MA, Floudas CA. Novel unified modeling approach for short-term scheduling. *Ind Eng Chem Res*. 2009;48:2947–2964.
46. Li J, Floudas CA. Optimal event point determination for short-term scheduling of multipurpose batch plants via unit-specific event-based continuous-time approaches. *Ind Eng Chem Res*. 2010;49:7446–7469.
47. Li J, Karimi IA, Srinivasan R. Efficient bulk maritime logistics for the supply and delivery of multiple chemicals. *Comput Chem Eng*. 2010;34:2118–2128.
48. Androulakis IP, Maranas CD, Floudas CA. αBB: a global optimization method for general constrained nonconvex problems. *J Global Optim*. 1995;7:337–363.
49. IBM. ILOG, CPLEX, version 12.1.0.; 2009. Available at http://www.ilog.com/products/cplex/.
50. Drud A. CONOPT, version 3, 2007. Available at http://www.gams.-com/dd/docs/solvers/conopt.pdf.
51. Floudas CA. *Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications*. New York: Oxford University Press, 1995.
52. Floudas CA. *Deterministic Global Optimization: Theory, Methods, and Applications; Nonconvex Optimization and It's Applications*. Dordrecht, The Netherlands: Kluwer Academic Publishers, 2000.
53. Adjiman CS, Dallwig S, Floudas CA, Neumaier A. A global optimization method, α BB, for general twice differentiable NLPs. I. Theoretical advances. *Comput Chem Eng*. 1998;22:1137–1158.

Appendix: Bound Updates Based On Topology

First, the value of $V_I^{\max}$ (i,n) is given by

$$V_I^{\max}(i,n)=\begin{cases} V_I^{\text{init}}(i)+\sum\limits_{\substack{p:(p,i)\in\mathbf{S}_{P,I}\\ V_P^{\text{init}}(p)+V_I^{\text{init}}(i)\le V_I^{\max}(i)}} V_P^{\text{init}}(p) & \text{if } \Delta n=0 \text{ and } n=1\\ \min\left\{V_I^{\max}(i), V_I^{\text{init}}(i)+\sum\limits_{p:(p,i)\in\mathbf{S}_{P,I}} V_P^{\text{init}}(p)\right\} & \text{if } \Delta n=1 \text{ and } n=1\\ V_I^{\max}(i) & \text{if } n>1 \end{cases} \quad \forall i \tag{A1}$$

Initialization: $E_{I,C}^{\min}$ $(i,c,n)=0$ and $E_{I,C}^{\max}$ $(i,c,n)=1$

$$E_{I,C}^{\min}(i,c,n)=\begin{cases} \max\left\{\dfrac{V_{I,C}^{\text{init}}(i,c)}{V_I^{\text{init}}(i)+\sum\limits_{\substack{p:(p,i)\in\mathbf{S}_{P,I},p:(p,c)\notin\mathbf{S}_{P,C}\\ V_P^{\text{init}}(p)+V_I^{\text{init}}(i)\le V_I^{\max}(i)}} V_P^{\text{init}}(p)}, \dfrac{V_{I,C}^{\text{init}}(i,c)}{V_I^{\max}(i,n)}\right\} & \text{if } V_I^{\text{init}}(i)>0 \text{ and } \Delta n=0\\ \max\left\{\dfrac{V_{I,C}^{\text{init}}(i,c)}{V_I^{\text{init}}(i)+\sum\limits_{p:(p,i)\in\mathbf{S}_{P,I},p:(p,c)\notin\mathbf{S}_{P,C}} V_P^{\text{init}}(p)}, \dfrac{V_{I,C}^{\text{init}}(i,c)}{V_I^{\max}(i,n)}\right\} & \text{if } V_I^{\text{init}}(i)>0 \text{ and } \Delta n=1\\ 0 & \text{if} V_I^{\text{init}}(i)=0 \end{cases} \quad \forall (i,c)\in\mathbf{S}_{I,C}, n=1 \tag{A2a}$$

$$E_{I,C}^{\min}(i,c,n)=\begin{cases} \dfrac{E_{I,C}^{\min}(i,c,n-1)\cdot V_I^{\min}(i,n-1)}{V_I^{\max}(i,n)} & \text{if } V_I^{\max}(i,n)>0\\ 0 & \text{if } V_I^{\max}(i,n)=0 \end{cases} \quad \forall (i,c)\in\mathbf{S}_{I,C}, n>1 \tag{A2b}$$

$$E_{I,C}^{\max}(i,c,n)=\begin{cases} \dfrac{\left[E_{I,C}^{\text{init}}(i,c)-1\right]\cdot\left[V_I^{\min}(i)\right]^{\sigma}}{\left[V_I^{\min}(i)+V_P^{\text{init}}(p)\right]^{\sigma}}+1 & \text{if } V_I^{\min}(i)>0\\ E_{I,C}^{\text{init}}(i,c) & \text{if } V_I^{\min}(i)=0 \text{ and } \max\limits_{\substack{p:(p,i)\in\mathbf{S}_{P,I},p:(p,c)\in\mathbf{S}_{P,C}\\ V_P^{\text{init}}(p)+V_I^{\text{init}}(i)\le V_I^{\max}(i,n)}}\left\{V_P^{\text{init}}(p)\right\}=0\\ 1 & \text{if } V_I^{\min}(i)=0 \text{ and } \max\limits_{\substack{p:(p,i)\in\mathbf{S}_{P,I},p:(p,c)\in\mathbf{S}_{P,C}\\ V_P^{\text{init}}(p)+V_I^{\text{init}}(i)\le V_I^{\max}(i,n)}}\left\{V_P^{\text{init}}(p)\right\}>0 \end{cases} \quad \forall (i,c)\in\mathbf{S}_{I,C}, n, \sigma=\sum_{\substack{p:(p,i)\in\mathbf{S}_{P,I}\\ p:(p,c)\in\mathbf{S}_{P,C}}} 1 \tag{A3}$$

$$E_{I,C}^{\max}(i,c,n)=\begin{cases} \min\left\{\dfrac{V_{I,C}^{\text{init}}(i,c)+\sum\limits_{\substack{p:(p,i)\in\mathbf{S}_{P,I},p:(p,c)\in\mathbf{S}_{P,C}\\ V_P^{\text{init}}(p)+V_I^{\text{init}}(i)\le V_I^{\max}(i,n)}} V_P^{\text{init}}(p)}{V_I^{\text{init}}(i)+\sum\limits_{\substack{p:(p,i)\in\mathbf{S}_{P,I},p:(p,c)\in\mathbf{S}_{P,C}\\ V_P^{\text{init}}(p)+V_I^{\text{init}}(i)\le V_I^{\max}(i,n)}} V_P^{\text{init}}(p)}, \dfrac{V_{I,C}^{\text{init}}(i,c)+V_I^{\max}(i,n)-V_I^{\text{init}}(i)}{V_I^{\max}(i,n)}\right\} & \text{if } V_I^{\text{init}}(i)>0\\ 1 & \text{if } V_I^{\text{init}}(i)=0 \text{ and } \max\limits_{\substack{p:(p,i)\in\mathbf{S}_{P,I},p:(p,c)\in\mathbf{S}_{P,C}\\ V_P^{\text{init}}(p)+V_I^{\text{init}}(i)\le V_I^{\max}(i,n)}}\left\{V_P^{\text{init}}(p)\right\}>0\\ 0 & V_I^{\text{init}}(i)=0 \text{ and } \max\limits_{\substack{p:(p,i)\in\mathbf{S}_{P,I},p:(p,c)\in\mathbf{S}_{P,C}\\ V_P^{\text{init}}(p)+V_I^{\text{init}}(i)\le V_I^{\max}(i,n)}}\left\{V_P^{\text{init}}(p)\right\}=0 \end{cases} \quad \forall (i,c)\in\mathbf{S}_{I,C}, n=1, \Delta n=0 \tag{A4a}$$

$$E_{I,C}^{\max}(i,c,n)=\begin{cases}\min\left\{\dfrac{V_{I,C}^{\text{init}}(i,c)+\sum\limits_{p:(p,i)\in\mathbf{S}_{P,I},p:(p,c)\in\mathbf{S}_{P,C}}V_P^{\text{init}}(p)}{V_I^{\text{init}}(i)+\sum\limits_{p:(p,i)\in\mathbf{S}_{P,I},p:(p,c)\in\mathbf{S}_{P,C}}V_P^{\text{init}}(p)},\dfrac{V_{I,C}^{\text{init}}(i,c)+V_I^{\max}(i,n)-V_I^{\text{init}}(i)}{V_I^{\max}(i,n)}\right\} & \text{if} V_I^{\text{init}}(i)>0\\ 1 & \text{if} V_I^{\text{init}}(i)=0 \text{ and } \max\limits_{p:(p,i)\in\mathbf{S}_{P,I},p:(p,c)\in\mathbf{S}_{P,C}}\left\{V_P^{\text{init}}(p)\right\}>0\\ 0 & \text{if} V_I^{\text{init}}(i)=0 \text{ and } \max\limits_{p:(p,i)\in\mathbf{S}_{P,I},p:(p,c)\in\mathbf{S}_{P,C}}\left\{V_P^{\text{init}}(p)\right\}=0\end{cases}$$
$$\forall (i,c)\in\mathbf{S}_{I,C}, n=1, \Delta n=1 \quad \text{(A4b)}$$

$$E_{I,C}^{\max}(i,c,n)=\begin{cases}\min\left\{\begin{array}{l}\dfrac{E_{I,C}^{\max}(i,c,n-1)\cdot V_I^{\min}(i,n-1)+\sum\limits_{\substack{p:(p,i)\in\mathbf{S}_{P,I},p:(p,c)\in\mathbf{S}_{P,C}\\ V_P^{\text{init}}(p)+V_I^{\min}(i,n-1)\le V_I^{\max}(i,n)}}V_P^{\text{init}}(p)}{V_I^{\min}(i,n-1)+\sum\limits_{\substack{p:(p,i)\in\mathbf{S}_{P,I},p:(p,c)\in\mathbf{S}_{P,C}\\ V_P^{\text{init}}(p)+V_I^{\min}(i,n-1)\le V_I^{\max}(i,n)}}V_P^{\text{init}}(p)},\\ \dfrac{E_{I,C}^{\max}(i,c,n-1)\cdot V_I^{\min}(i,n-1)+V_I^{\max}(i,n)-V_I^{\min}(i,n-1)}{V_I^{\max}(i,n)}\end{array}\right\} & \text{if} V_I^{\min}(i,n-1)>0\\ 1 & \text{if} V_I^{\min}(i,n-1)=0 \text{ and } \max\limits_{\substack{p:(p,i)\in\mathbf{S}_{P,I},p:(p,c)\in\mathbf{S}_{P,C}\\ V_P^{\text{init}}(p)+V_I^{\min}(i,n-1)\le V_I^{\max}(i,n)}}\left\{V_P^{\text{init}}(p)\right\}>0\\ E_{I,C}^{\max}(i,c,n-1) & \text{if} V_I^{\min}(i,n-1)=0 \text{ and } \max\limits_{\substack{p:(p,i)\in\mathbf{S}_{P,I},p:(p,c)\in\mathbf{S}_{P,C}\\ V_P^{\text{init}}(p)+V_I^{\min}(i,n-1)\le V_I^{\max}(i,n)}}\left\{V_P^{\text{init}}(p)\right\}=0\end{cases}$$
$$\forall(i,c)\in\mathbf{S}_{I,C}, n>1, \Delta n=0 \quad \text{(A5a)}$$

$$E_{I,C}^{\max}(i,c,n)=\begin{cases}\min\left\{\begin{array}{l}\dfrac{E_{I,C}^{\max}(i,c,n-1)\cdot V_I^{\min}(i,n-1)+\sum\limits_{p:(p,i)\in\mathbf{S}_{P,I},p:(p,c)\in\mathbf{S}_{P,C}}V_P^{\text{init}}(p)}{V_I^{\min}(i,n-1)+\sum\limits_{p:(p,i)\in\mathbf{S}_{P,I},p:(p,c)\in\mathbf{S}_{P,C}}V_P^{\text{init}}(p)},\\ \dfrac{E_{I,C}^{\max}(i,c,n-1)\cdot V_I^{\min}(i,n-1)+V_I^{\max}(i,n)-V_I^{\min}(i,n-1)}{V_I^{\max}(i,n)}\end{array}\right\} & \text{if } V_I^{\min}(i,n-1)>0\\ 1 & \text{if } V_I^{\min}(i,n-1)=0 \text{ and } \max\limits_{p:(p,i)\in\mathbf{S}_{P,I},p:(p,c)\in\mathbf{S}_{P,C}}\left\{V_P^{\text{init}}(p)\right\}>0\\ E_{I,C}^{\max}(i,c,n-1) & \text{if } V_I^{\min}(i,n-1)=0 \text{ and } \max\limits_{p:(p,i)\in\mathbf{S}_{P,I},p:(p,c)\in\mathbf{S}_{P,C}}\left\{V_P^{\text{init}}(p)\right\}=0\end{cases}$$
$$\forall (i,c)\in\mathbf{S}_{I,C}, n>1, \Delta n=1 \quad \text{(A5b)}$$

We take the maximum value among Eqs. A3–A5 as final $E_{I,C}^{\max}$ (i,c,n).
The value of $V_{P,I}^{\max}$ (p,i,n) is computed by

$$V_{P,I}^{\max}(p,i,n)=\min\left\{V_{P,I}^{\max}(p,i)\cdot H, V_p^{\text{init}}\right\} \qquad \forall (p,i)\in\mathbf{S}_{P,I}, n \quad \text{(A6)}$$

The values of $V_{I,U}^{\min}$ (i,u,n) and $V_{I,U}^{\max}$ (i,u,n) are given by

$$\begin{cases}V_{I,U}^{\min}(i,u,n)=0 & \forall n\\ V_{I,U}^{\max}(i,u,n)=\min\left\{F_{I,U}^{\max}(i,u)\cdot H, V_I^{\text{init}}-V_I^{\min}(i,n), V_I^{\max}(i)-V_I^{\min}(i)\right\} & \text{if } n=1\\ V_{I,U}^{\max}(i,u,n)=\min\left\{\begin{array}{l}F_{I,U}^{\max}(i,u)\cdot H, V_I^{\max}(i,n-1)-V_I^{\min}(i,n)\\ V_I^{\max}(i)-V_I^{\min}(i)\end{array}\right\} & \text{if } n>1\end{cases} \qquad \forall (i,u)\in\mathbf{S}_{I,U} \quad \text{(A7)}$$

Manuscript received Dec. 24, 2010, and revision received Feb. 28, 2011.