A Cost-Effective Model for the Gasoline Blend Optimization Problem

Jaime Cerdá, Pedro C. Pautasso, and Diego C. Cafaro
INTEC (UNL—CONICET), Güemes 3450, 3000 Santa Fe, Argentina

DOI 10.1002/aic.15208
Published online in Wiley Online Library (wileyonlinelibrary.com)

Gasoline blending is a critical process with a significant impact on the total revenues of oil refineries. It consists of mixing several feedstocks coming from various upstream processes and small amounts of additives to make different blends with some specified quality properties. The major goal is to minimize operating costs by optimizing blend recipes, while meeting product demands on time and quality specifications. This work introduces a novel continuous-time mixed-integer linear programming (MILP) formulation based on floating time slots to simultaneously optimize blend recipes and the scheduling of blending and distribution operations. The model can handle non-identical blenders, multipurpose product tanks, sequence-dependent changeover costs, limited amounts of gasoline components, and multi-period scenarios. Because it features an integrality gap close to zero, the proposed MILP approach is able to find optimal solutions at much lower computational cost than previous contributions when applied to large gasoline blend problems. *AIChE J*, 00: 000–000, 2016
Keywords: gasoline recipe, blending operations, short-term scheduling, MILP approach, property indices

Introduction

Gasoline is the dominant refined product that accounts for a large portion of the total refinery profit[1]. It is a mixture of hydrocarbons, additives and blending agents. Straight-run gasoline separated from crude oil via distillation does not meet the required quality for modern engines, especially the octane number (ON) and vapor pressure (VP) specifications. Such properties can be improved through reforming and isomerization processes. Nonetheless, the straight-run gasoline is used as a blending component, but in a rather low proportion. The other feedstocks blended to make gasoline come from various refinery processing units such as catalytic reformers, alkylation units, isomerization units, fluid catalytic crackers, and hydrocrackers. The low-value *n*-butane is also blended in the gasoline mixture, but its amount is limited by the VP-specification. Additives and blending agents are added to the hydrocarbon mixture to improve the anti-knock performance and stability of gasoline. These oxygenated compounds include octane enhancers such as methyl tert-butyl ether and tert-butyl alcohol, as well as alternative fuels such as ethanol and methanol for economic and environmental reasons.

A large refinery can have more than 20 gasoline components that are blended into several gasoline grades. None of the individual components meet the specifications of commercial gasoline. Then, it is necessary to determine the best proportion of each one in the finished gasoline, called the blend recipe, to meet the particular properties of the blend required by the customer. A number of different gasoline grades are usually produced in the gasoline blending unit (GBU) of a refinery, with each grade meeting certain quality requirements. Quality specifications include the octane rating, given in terms of the Research Octane Number (RON) and the Motor Octane Number (MON), the Reid Vapor Pressure, ASTM distillation points, the flash point, the aromatic and sulfur content, etc. These properties are monitored during production to ensure the required qualities of the final product. RON and MON are considered to be the most important ones because the increase in the engine's compression ratio led to higher requirements in the octane rating.

Several blenders with each one making only one final product at a time are operated in a semi-continuous mode in the GBU. Gasoline components stored in a set of dedicated storage tanks are supplied to the blenders at constant feed flow rates according to the blend recipe selected for the final product. At the same time, the component tanks can be receiving additional amounts of components from upstream production units. The finished products obtained in the blenders are discharged into a farm of product tanks which afterwards deliver orders to the market within specified time windows. When using on-line certification, product properties are tested as the blend is being produced with an on-line multi-property blend analyzer, and the product qualities are certified without the need of additional laboratory testing. By eliminating the waiting time for additional testing at the end of the blend run, the on-line certification has an important advantage in terms of the GBU total throughput.

The operational management of a GBU requires making a number of key decisions such as the choice of the blend recipe for every gasoline grade, the assignment of final products to

Additional Supporting Information may be found in the online version of this article.

Correspondence concerning this article should be addressed to J. Cerdá at jcerda@intec.unl.edu.ar.

blending runs, the allocation sequence of production runs to blenders, the assignment of product tanks to gasoline grades and the allocation sequence of blending runs and order deliveries to product tanks. In addition, the problem involves the scheduling of blending and distribution operations and the tracking of the inventory level in component and product tanks. The goal is to produce the required amounts of on-spec blended products while minimizing quality giveaway and product changeovers in blenders and product tanks, and maximizing the use of low-cost components in the blends. In short, the gasoline blending problem aims to simultaneously determine the product recipes and the short-term scheduling of blending and order delivery operations. It is a very complex problem because it usually involves a large number of orders, final products, blend headers, gasoline components, and storage tanks for components and final products.

The blending technology used by refiners to optimize the gasoline recipe typically works on three levels: off-line optimization, on-line optimization, and regulatory control. Given the component properties, an off-line optimizer generates the initial blend recipes to get on-spec products and the short-term planning of blending operations. This information is subsequently downloaded to the on-line optimizer that determines the set points for the controller. Both off-line and on-line optimizers are typically conducted using either linear programming or successive LPs to solve nonlinear optimization models. Recently, an off-line multi-blend optimizer using event-based, multi-period nonlinear models has been made commercially available.[2] Based on blend quality measurements, an on-line optimizer can modify the initial recipes during the blend and determine the final blend recipes to be executed by the controller. To maintain the accuracy of the models used by the blend optimizer, a correction term called bias updating is usually employed to account for the nonlinearity of some gasoline properties. Bias updating involves comparing measured blend properties with those predicted by the model in the on-line optimizer. The difference between the two is added as an error term to the linear blending models. In this way, the on-line optimizer is usually formulated as a linear programming with bias updating.

This work introduces a novel continuous-time mixed-integer linear programming (MILP) formulation based on floating time-slots to simultaneously optimize gasoline blend recipes and the short-term planning of blending and distribution operations using linear blending indices. Floating slots are not pre-allocated to time periods but such assignment decisions are model variables. The proposed off-line optimizer can handle non-identical blenders, multipurpose product tanks, sequence-dependent changeover costs, limited amounts of gasoline components, and multi-period scenarios with feed flow rates to component tanks changing with the period. Moreover, all the operational rules for the management of GBUs are considered. Because it features an integrality gap close to zero, the proposed MILP approach is able to find optimal solutions at much lower computational cost than previous contributions when applied to large gasoline blend scheduling problems.

Previous Contributions

The gasoline components can be blended using either the traditional batch-blending process where they are mixed in a blend tank, or tankless, inline blenders continuously mixing the feedstocks. Several papers have already studied the multi-period mixing of crude oil or refined petroleum products in blending tanks and considering the issue of (non-)simultaneous input and output flows.[3–6]

Other important contributions assumed that the blending process is carried out in tankless inline blenders, and focused on the simultaneous optimization of gasoline recipes and the scheduling of blending and distribution operations[7] developed a multi-level integrated approach to coordinate the short-term scheduling of blending operations with nonlinear recipe optimization. At the upper level, a nonlinear problem is solved to determine blending recipes and product volumes for the scheduling level. The lower-level scheduling problem was modeled through an MILP formulation based on a Resource-Task Network representation that allows recipe changeovers between alternate product recipes determined at the upper level[8] decomposed the overall refinery system into three major sections: (a) crude oil unloading, blending and processing, (b) scheduling of production units yielding intermediate streams, and (c) gasoline blending and delivery of final products. Then, they proposed a continuous-time event-based MILP formulation for the simultaneous scheduling of gasoline blending and distribution operations. The model assumed fixed product recipes and a single blend header that can concurrently feed various product tanks. An interesting model feature is the handling of multipurpose product tanks that can deliver multiple orders of a given product at the same time. Reciprocally, multiple tanks can deliver the same order[9] proposed an iterative method that consists of solving a sequence of MILP formulations based on either a discrete-time or a slot-based continuous-time domain representation. Since variable product recipes and nonlinear gasoline properties are considered, the iterative procedure aims to preserve the model's linearity. Thus, the solution of a large nonlinear, non-convex mixed-integer programming (MINLP) model is replaced by sequential MILP approximations. When the slot-based approach is adopted, the scheduling horizon is divided into a number of time intervals using the order due dates, and a set of process time slots with unknown durations is postulated for each interval. Every product demand can be satisfied by making multiple discharges from one or more product tanks during one or more time slots. Besides, the MILP model assumes the operation of parallel identical blenders, dedicated storage tanks for components and final products, a constant feed component flow rate all along the scheduling horizon, and simultaneous loading and unloading operations at every product tank. The problem objective is to maximize the production profit while satisfying process constraints, final product demands and quality specifications. Since feasible solutions are difficult to find through the sequential MILP procedure, additional constraints penalizing out-spec products and component shortages were included.

Li et al.[10] developed a continuous-time MILP formulation based on process slots for the simultaneous treatment of variable recipes, blending campaigns, component/product storage, and order scheduling. Compared with the work of Méndez et al.,[9] the approach incorporates several new operational features such as the handling of parallel non-identical blenders, multi-purpose product tanks, and blending and storage transitions. In addition, a blender can at most charge one product tank at a time, the blending rate should remain constant during a production run, and a product tank cannot receive product from a blender and deliver customer orders at the same time. Although some blend quality specifications are predicted using

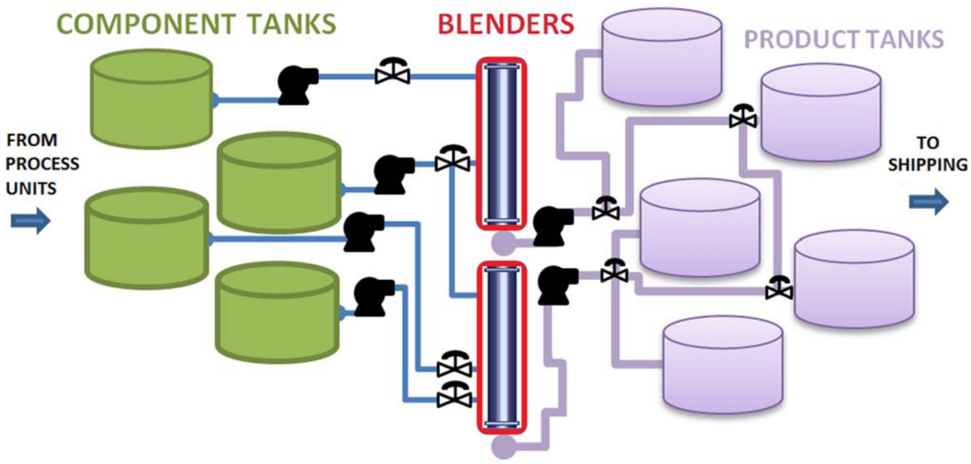

Figure 1. A sketch of the gasoline blending process.

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

highly nonlinear correlations, Li et al.[11] noted that a linear blending index exists for almost every hydrocarbon property with nonlinear mixing correlations. Using such property indices, Li et al.[10] were able to model the problem through an MILP that ensures a constant blend rate per time slot. Because a production run may be extended over two or more slots, the authors developed a schedule adjustment procedure to get a constant blend rate during a blending run. The procedure consists of successive MILPs instead of solving a non-convex MINLP. The selected problem goal was to minimize the total operating cost, including component, order tardiness, transition and backorder costs. The MILP model has also been extended to consider multi-period scenarios with each period featuring a different constant component feed rate. Later, Li and Karimi[12] presented a multigrid continuous-time MILP formulation to improve the efficiency of the approach proposed by Li et al.[10] To this end, the new MILP formulation uses unit slots instead of process slots to get a higher RMIP bound and a faster improvement of the lower bound. Besides, it accounts for limited component inventories that force to change the product recipes along the time horizon. By including additional constraints, the model was extended to allow for simultaneous receipt/delivery operations in product tanks.

Kolodziej et al.[13] introduced a generalized non-convex MINLP formulation for the multi-period blend scheduling problem. The primary difficulties to be faced were the presence of bilinear terms together with binary decision variables that are defined to impose operational constraints. A radix-based discretization technique was applied to reformulate the model as an approximate MILP that is incorporated either in a heuristic procedure or in two rigorous global optimization methods. Flows are not allowed to enter and exit a blending tank in the same time period. The resulting approaches require much less computational time than existing global optimization solvers. More recently, Castillo-Castillo and Mahalec[14] presented a continuous-time MILP model that modifies to some extent the formulation of Li and Karimi[12] by incorporating new operational constraints, lower bounds on the objective function, and additional equations transforming binary variables into continuous ones. Moreover, the use of demand information allows lowering the number of binary variables. The proposed formulation accounts for product-dependent setup times in blenders, minimum production of blend runs, and penalties for fulfilling the same order by product discharges from multiple tanks. When nonlinear correlations are considered to estimate RON and MON properties of the gasoline blend, satisfactory results were obtained by using global MINLP solvers.

One of the major shortcomings of previous contributions is the very high computational cost they require to find near-optimal solutions for real-world gasoline blend scheduling problems. This work presents a new tight MILP formulation that solves large problems at much lower CPU time.

Problem Statement

The GBU of an oil refinery comprises a set of dedicated tanks for gasoline components $s \in \boldsymbol{S}$, a pumping station, a number of blend headers $b \in \boldsymbol{B}$ working in parallel and a set of multipurpose product tanks $j \in \boldsymbol{J}$ that can hold different final products $p \in \boldsymbol{P}$ along the time horizon (see Figure 1).

Several production runs $i \in \boldsymbol{I}$ are sequentially performed in each blender to produce different gasoline blends or final products $p \in \boldsymbol{P}$. These products are temporarily stored in product tanks and subsequently delivered to meet a set of customer orders or requests $r \in \boldsymbol{R}$. A blend header b is usually assigned to the production of a group of final products $p \in \boldsymbol{P_b}$ and its processing rate can vary within some given product-dependent range $[\,rb_{b,p}^{\min},\ \ rb_{b,p}^{\max}]$. The production runs should have a minimum length whose value depends on both the product and the blender ($lb_{b,p}^{\min}$). A product changeover in the blender b has a sequence-dependent transition time ($\tau_{b,p,p'}$) and transition cost ($ctrb_{b,p,p'}$).

Conversely, every final product p is characterized by specifying the allowable range of some critical properties $g \in \boldsymbol{G}$, and the limiting proportions of the gasoline components in the blend, given by $[ppr_{g,p}^{\min},\ ppr_{g,p}^{\max}]$ and $[\ vc_{s,p}^{\min},\ \ vc_{s,p}^{\max}]$, respectively. Besides, the values of the critical properties for the gasoline components ($spr_{g,s}$) are problem data. Each gasoline blend has its own recipe consisting on the group of constituent components and their relative proportions. The optimal recipe of a final product is the one featuring the lowest component cost, while satisfying both property requirements and limiting proportions of gasoline components. If these components are available in unlimited amounts, the recipe of a particular product is independent of the other products to be blended during the time horizon. Under limited amounts of components, the recipe will depend on the other products to be obtained and the amounts of them required by the customers. Thus, the optimization of the recipes should be done simultaneously for all the required products.

The feed rate to each component tank will be regarded as a piecewise constant function of time. In other words, the scheduling horizon will be viewed as composed by a number of time periods $k \in \boldsymbol{K}$ with each one featuring a different constant component feed rate $svr_{s,k}$. The time limits for period k are given by [$llim_k$, $ulim_k$] and its length is $hk_k = ulim_k - llim_k$. The overall length of the scheduling horizon $h = \sum_{k \in \boldsymbol{K}} hk_k$ is also a problem data. The unit cost of component s is provided by the parameter $scost_s$. Customer orders demanding product p are grouped into the set $\boldsymbol{R_p}$. In this way, each request is defined by specifying the order group $\boldsymbol{R_p}$ to which it belongs, the order size q_r, the delivery time window [atw_r, btw_r] and the order delivery rate rdr_r. Orders that are delivered beyond their due dates btw_r should pay a demurrage penalty per unit time (ctd). Data for component tanks include the tank capacity ($scap_s$), the initial inventory (iis_s) and the constant feed flow rate during each period k ($svr_{s,k}$). For every product tank $j \in \boldsymbol{J}$, it is given the group of products that can be stored ($\boldsymbol{P_j}$), the tank capacity ($pcap_j$), the product currently stored, the initial inventory ($iij_{p,j}$), and the maximum delivery rate (pdr_j).

In addition to the gasoline property specifications and the limiting component proportions in the blends, the management of a GBU also involves a series of operational rules for blenders and component/product tanks.[5] Those rules are the following:

1. Every blender can process several products over the time horizon, but one after another.
2. After starting the processing of a gasoline product, a blender should operate for some minimum time before stopping or switching to another product.
3. A component tank may feed multiple blenders at the same time.
4. A component tank can receive flows from upstream processes and feed blenders at the same time.
5. A blender can at most feed a single product tank at any time instant.
6. Different products can be sequentially stored in the same product tank.
7. A product tank cannot receive a lot of final product from a blender and simultaneously deliver a customer order.
8. A customer order can be satisfied by delivering the requested product from different product tanks.
9. A product tank can deliver several orders at the same time.

The gasoline recipe and blend scheduling problem aims to determine: (a) the allocation of production runs to blenders; (b) the gasoline grade and the amount yielded by every production run; (c) the product recipe for every production run; (d) the short-term schedule of blending operations in every blender; (e) the sequence of gasoline grades that are stored in each product tank; (f) the allocation of blending runs and customer orders to product tanks, (g) the schedule of delivery operations from each product tank; and (h) the inventory profiles for component and product tanks, in such a way that all customer orders are fully satisfied while minimizing quality giveaway, off-spec products, component cost, order delivery tardiness, and changeover costs in blenders and product tanks.

Model Assumptions

In addition to the operational rules, a series of assumptions already proposed in previous works have been used to model the problem.[10] They are:

1. The scheduling horizon is composed of a number of time periods of known lengths.
2. The feed rate to every component tank is a piecewise constant function of time, i.e., it can have a different value for each period.
3. Multiple component tanks may simultaneously feed a particular blender.
4. Several parallel non-identical blenders can be operated.
5. Every blending run should occur within a single time slot.
6. Mixing in every blender is perfect.
7. The product changeover time and cost in blenders are sequence-dependent.
8. Every product tank can at most receive the production from a single run during a time slot.
9. The changeover time in product tanks is negligible.
10. Each order involves a single product and must be delivered within the time horizon. Then, multiproduct requests are handled as a set of single-product orders with the same delivery window.
11. Loading and delivery operations in product tanks should occur in different time slots.
12. Delivery of a customer order from a product tank should occur within a single time slot and begin at the start of the slot. Moreover, the amount delivered should be greater than a threshold value.
13. No additional time is needed for product certification.

The Mathematical Model

The proposed continuous-time MILP formulation for the simultaneous optimization of blend recipes and the scheduling of blending and delivery operations is based on the use of ordered sets of production runs ($\boldsymbol{I}$) and floating time slots ($\boldsymbol{T}$) with variable length. Such model features allow to sequencing blend operations in blenders ($\boldsymbol{B}$) and manage non-simultaneous receipt and delivery tasks in product tanks, respectively. As shown in Figure 2, the elements of $\boldsymbol{I}$ are said to be chronologically ordered because the production run $(i + 1)$ never begins before starting run i. If both campaigns are performed in the same blender, then run $(i + 1)$ must begin after finishing run i. Similarly, the time slot $t \in \boldsymbol{T}$ starts at the completion time of slot $(t - 1)$. In addition, the scheduling horizon of known length is composed of a number of time intervals $k \in \boldsymbol{K}$, with the value of each component feed rate ($svr_{s,k}$) varying from one to another period, i.e., a multi-period scenario. In contrast to previous works, floating time slots $t \in \boldsymbol{T}$ and production campaigns $i \in \boldsymbol{I}$ are not pre-assigned to time periods because it is somewhat difficult to know *a priori* the number of them required in each period. They can be viewed as floating elements that can move from one to another period during the solution procedure. Pre-assignment of slots and production runs to time periods may lead to non-optimal solutions. Instead, unique sets of time slots and productions runs are defined, and the assignment of them to time periods is made by the model in an optimal way. Another difference with previous approaches is the handling of sequence-dependent transition costs in blenders and product tanks. $\boldsymbol{R}$ is another important set involving the customer orders to be satisfied during the current scheduling horizon. It is forbidden for a product tank to receive a flow of product from some blender and deliver a customer order of the same product within the same time slot. However, a simple change allows the proposed

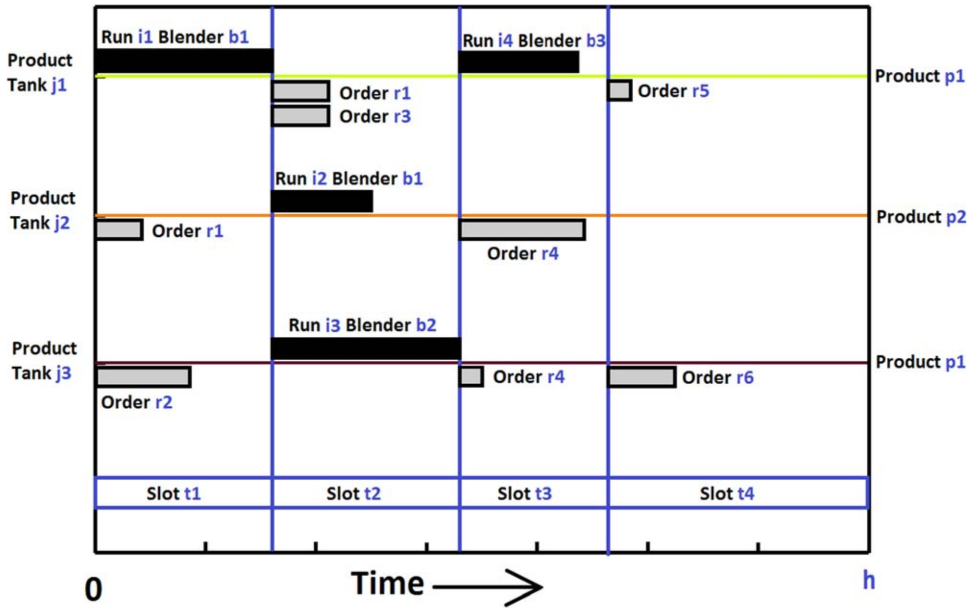

Figure 2. Illustrating the proposed slot-based approach.

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

approach to also manage simultaneous receipt/delivery operations in product tanks.

Model variables

Model decision variables can be gathered into three groups according to their purpose. The arrangement of blending operations in the blend headers is handled through three different sets of binary variables. Variables $WI_{i,k}$ assign production runs to time periods $k \in \boldsymbol{K}$, variables $WB_{i,b}$ allocate production runs to blenders $b \in \boldsymbol{B}$, and $YB_{i,p}$ select the final product $p \in \boldsymbol{P}$ generated by every production run. The second group of 0–1 variables is defined to allocate production runs to time slots $t \in \boldsymbol{T}$ and product tanks $j \in \boldsymbol{J}$. $XIJ_{i,j,t}$ indicates that the production run i is discharged into the product tank j during the time slot t, and $XPJ_{p,j,t}$ identifies the final product stored in product tank j during the slot t. In combination with $YB_{i,p}$, the assignment variables $XIJ_{i,j,t}$ and $XPJ_{p,j,t}$ choose the destination for the volume of product generated by run i during the time slot t. In fact, $XIJ_{i,j,t}$ can be equal to one only if $YB_{i,p} = XPJ_{p,j,t} = 1$. The third group involves the 0–1 variables $XRJ_{r,j,t}$ and $XDJ_{p,j,t}$ to assign customer orders to product tanks and time slots. If $XRJ_{r,j,t} = 1$, then the product tank j delivers a portion or the whole order r within the time slot t. Let us assume that order r demands product p. In that case, $XRJ_{r,j,t}$ can be equal to one only if a volume of product p is discharged from tank j during the time slot t, i.e., $XDJ_{p,j,t} = 1$.

The continuous variables associated to a production run i are: $QB_{i,p}$ representing the production of final product p, (SB_i, CB_i) denoting the initial and final times, and $LB_{i,b,p}$ standing for the length of run i. The value of $LB_{i,b,p}$ is restricted to the range $[lb^{\min}_{b,p},\ lb^{\max}_{b,p}]$ which depends on the selected blender and the product to be obtained. If run i is allocated to time slot t and (ST_t, CT_t) represent the initial and final times of slot t, then the following conditions must hold: $SB_i \geq ST_t$ and $CB_i \leq CT_t$. To track the inventory levels in component tanks, the proposed model includes the continuous variables ($SINI_{s,i}$, $SINC_{s,i}$) standing for the inventory of component s at the start and end times of run i, $SINF_s$ denoting the inventory of component s at the end of the scheduling horizon, and $US_{s,i,}$ indicating the amount of component $s \in \boldsymbol{S}$ assigned to run i. Similarly, the inventory levels in product tanks are controlled by the variables: $PINV_{p,j,t}$ standing for the inventory of final product p in tank $j \in \boldsymbol{J_p}$ at the end of slot t, $QPJ_{i,p,j,t}$ denoting the amount of product p coming from run i that is discharged into tank $j \in J_p$ during the slot t, and $UP_{p,j,t}$ representing the amount of product p unloaded from product tank j during the slot t. A product changeover in tank j can occur at the start of a time slot only if tank j is empty at that time. It is assumed that an order delivery assigned to the slot t always begins at the initial time ST_t. Associated to an order delivery r there are two continuous variables: $UR_{r,j,t}$ representing the amount of product delivered from tank j within the time slot t for order r, and $CR_{r,j,t}$ denoting the final time of that delivery.

Model constraints

Different groups of constraints are defined for: (a) scheduling production runs in the blend headers, (b) fulfilling the demand and quality specifications of final products, (c) tracking inventory levels in component and product tanks, and (d) scheduling receipt and delivery operations in multipurpose product tanks. All the model constraints are given below.

Production runs performed in blenders

A Production Run i Should at Most Be Performed Within a Single Time Period k. The ordered set $\boldsymbol{K}$ includes the time periods into which the scheduling horizon has been divided, and the binary variable WI_{ik} assigns the production run $i \in \boldsymbol{I}$ to time period $k \in \boldsymbol{K}$. According to Eq. 1, a production run should at most be performed within a single time period. Therefore, $\sum_{k \in \boldsymbol{K}} WI_{i,k} = 0$ characterizes a fictitious run i. Moreover, the feed rate of any gasoline blending component s ($vc_{s,k}$) can change with the period $k \in \boldsymbol{K}$. For single-period blend scheduling problems, the set $\boldsymbol{K}$ just includes only one element.

$$\sum_{k \in \boldsymbol{K}} WI_{i,k} \leq 1 \ \forall \ i \in \boldsymbol{I} \tag{1}$$

A Production Run Can at Most Be Assigned to a Single Blender. The set $\boldsymbol{B}$ comprises the blenders available for the mixing process, while the 0–1 variable WB_{ib} allocates production runs to blenders. By Eq. 2, a production run should at most be assigned to a single blender.

$$\sum_{b\in\boldsymbol{B}} WB_{i,b} = \sum_{k\in\boldsymbol{K}} WI_{i,k} \quad \forall\ i\in\boldsymbol{I}. \tag{2}$$

A Production Run Can at Most Yield a Single Product. The binary variable $YB_{i,p}$ allocates production runs to final products. By Eq. 3, a production run can at most yield a single final product.

$$\sum_{p\in\boldsymbol{P}} YB_{i,p} = \sum_{b\in\boldsymbol{B}} WB_{i,b} \quad \forall\ i\in\boldsymbol{I} \tag{3}$$

Definition of the Continuous Variable $WBP_{i,b,p}$. The continuous variable $WBP_{i,b,p}$ identifies the final product yielded by production run i in blender b. Its value becomes determined by Eqs. 4–6.

$$WBP_{i,b,p} \geq WB_{i,b}+YB_{i,p}-1 \quad \forall\ i\in\boldsymbol{I},\ p\in\boldsymbol{P},\ b\in\boldsymbol{B_p} \tag{4}$$

$$\sum_{b\in\boldsymbol{B_p}} WBP_{i,b,p} \leq YB_{i,p} \quad \forall\ i\in\boldsymbol{I},\ p\in\boldsymbol{P} \tag{5}$$

$$\sum_{p\in\boldsymbol{P_b}} WBP_{i,b,p} \leq WB_{i,b} \quad \forall\ i\in\boldsymbol{I},\ b\in\boldsymbol{B} \tag{6}$$

Length of a Production Run. The continuous variable $LB_{i,b,p}$ whose value is restricted to the interval $[lb^{\min}_{b,p},\ lb^{\max}_{b,p}]$ denotes the length of the production run i. By Eq. 7, it has a finite value only if $WBP_{i,b,p}=1$, i.e., just for the blender b and the final product p assigned to run i.

$$lb^{\min}_{b,p}\ WBP_{i,b,p} \leq LB_{i,b,p} \leq lb^{\max}_{b,p}\ WBP_{i,b,p} \quad \forall\ i\in\boldsymbol{I},\ p\in\boldsymbol{P},\ b\in\boldsymbol{B_p}. \tag{7}$$

Volume of Final Product Generated by a Production Run. The continuous variable $QB_{i,p}$ stands for the amount of final product p yielded by run i. Its value is given by Eq. 8, where the interval $[rb^{\min}_{b,p},\ rb^{\max}_{b,p}]$ denotes the processing rate limits for producing the final product p in blender b.

$$\sum_{b\in B_p} rb^{\min}_{b,p} LB_{i,b,p} \leq QB_{i,p} \leq \sum_{b\in B_p} rb^{\max}_{b,p} LB_{i,b,p} \quad \forall\ i\in\boldsymbol{I},\ p\in\boldsymbol{P} \tag{8}$$

Ordered Execution of Production Runs in Blenders. By Eq. 9, a generic run i can be assigned to a blender only if the precedence campaign $(i-1)$ is really performed, i.e., $\sum_{b\in\boldsymbol{B}} WB_{(i-1),b}=1$. Therefore, the last elements of the set I are reserved for dummy or fictitious runs.

$$\sum_{b\in\boldsymbol{B}} WB_{i,b} \leq \sum_{b\in\boldsymbol{B}} WB_{(i-1),b} \quad \forall\ (i-1),i\in\boldsymbol{I},\ b\in\boldsymbol{B} \tag{9}$$

Sequencing Production Runs in Every Blender b. The continuous variables SB_i and CB_i stand for the starting and completion times of run i. By Eq. 10a, a production run i' can never start before completing run i if $i<i'$ and both runs have been assigned to the same blender. Moreover, if run i is assigned to time period k, it should be performed within the time limits of period k, i.e., ($llim_k$, $ulim_k$). Then, the starting and completion times of the production run i are determined by the set of Eqs. 10a–10d. The parameter $\tau_{b,p,p'}$ is the sequence-dependent changeover time in blender b. It is equal to zero for $p=p'$, i.e. $\tau_{b,p,p}=0$.

$$CB_i \leq SB_{i'}-\tau_{b,p,p'}+h\ (2-WBP_{i,b,p}-WBP_{i',b,p'}) \quad \forall\ i,i'\in\boldsymbol{I}\ (i'>i),\ p,p'\in P,\ b\in\boldsymbol{B_p}\cap\boldsymbol{B_{p'}} \tag{10a}$$

$$CB_i \leq \sum_{k\in\boldsymbol{K}} ulim_k\ WI_{i,k}\ \forall\ i\in\boldsymbol{I} \tag{10b}$$

$$SB_i \geq \sum_{k\in\boldsymbol{K}} llim_k\ WI_{i,k}\ \forall\ i\in\boldsymbol{I} \tag{10c}$$

$$CB_i = SB_i+\sum_{p\in\boldsymbol{P}}\sum_{b\in\boldsymbol{B_p}} LB_{i,b,p}\ \forall\ i\in\boldsymbol{I} \tag{10d}$$

Since the elements of the set I stand for generic runs, Eq. 11a states that the model should activate the production runs in the same order that they are listed in the set I. Then, a production run $(i+1)$ can never begin before starting the preceding run i whatever are the blenders assigned to both runs. In this way, it is reduced the size of the solution space but no feasible alternative is cut off by the model. Moreover, fictitious runs arise last. In addition, it is also imposed that a production run i can never finish after completing the run $i'>i$ to facilitate the tracking of component/final product inventories. Eq. 11b just disallows solutions where run i finishes after completing the campaign $i'>i$. However, equivalent solutions can still be considered by performing a succeeding run i'' making the same product than run i in the same blender and starting at the end of campaign i. As a result, additional runs may be needed in the set I because of Eq. 11b.

$$SB_i \leq SB_{i+1} \quad \forall\ i\in\boldsymbol{I} \tag{11a}$$

$$CB_i \leq CB_{i+1} \quad \forall\ i\in\boldsymbol{I} \tag{11b}$$

If the changeover time is not sequence-dependent, Eq. 10a should be replaced by the following constraint:

$$CB_i \leq SB_{i'}-\tau_b+h\ \left(2-WB_{i,b}-WB_{i',b}\right)\ \forall\ i,i'\in\boldsymbol{I}\ (i'>i),\ b\in\boldsymbol{B}$$

Transition Cost in Blenders. The continuous variable $TRB_{i,b}$ given by Eqs. 12a and 12b denotes the cumulative transition cost at blender $b\in B$ after performing the production run i. In Eq. 12a, the parameter $ctrb_{b,p,p'}$ denotes the sequence-dependent transition cost in blender b and M_B is a relatively large number. When $p=p'$, no changeover occurs and $ctrb_{b,p,p}=0$. As stated by Eq. 12c, the maximum value of $TRB_{i,b}$ provides a lower bound for the total transition cost ($TTRB_b$) in blender b.

$$TRB_{i',b} \geq TRB_{i,b}+ctrb_{b,p,p'}-M_B\ \left(2-WBP_{i,b,p}-WBP_{i',b,p'}\right) \quad \forall\ (i,i')\in\boldsymbol{I}\ (i<i'),\ b\in\boldsymbol{B_p}\cap\boldsymbol{B_{p'}},\ p,p'\in\boldsymbol{P} \tag{12a}$$

$$TRB_{i,b} \leq M_B\ WB_{i,b} \quad \forall\ i\in\boldsymbol{I},\ b\in\boldsymbol{B} \tag{12b}$$

$$TTRB_b \geq TRB_{i,b} \quad \forall\ i\in\boldsymbol{I},\ b\in\boldsymbol{B} \tag{12c}$$

If the transition cost is not sequence-dependent and independent of the blender ($ctrb$), a lower bound on the total transition cost in blenders is given by Eq. 13. Such a lower bound assumes that a number of production runs equal to the number of products to be made has been chosen, i.e. one run per product.

$$\sum_{b\in \boldsymbol{B}} TTRB_b \;\geq\; ctrb\;[card(\boldsymbol{P})-card(B)] \tag{13}$$

Overall fulfillment of final product demands

The Net Demand of a Final Product Should Be Satisfied with New Production from Blenders. The parameter dem_p denotes the total demand of final product p to be satisfied during the scheduling horizon. It is given by Eq. 14, where q_r is the size of order r. Part of that demand can be fulfilled by making use of the initial inventory of product p. Equation 15 states that the production of product p should be large enough to cover its net demand over the time horizon. In Eq. 15, the set J_p includes the allowable storage tanks for product p, and the parameter $iij_{p,j}$ represents the initial inventory of product p in the tank $j\in \boldsymbol{J_p}$.

$$dem_p=\sum_{r\in \boldsymbol{R_p}} q_r \;\;\forall\; p\in \boldsymbol{P} \tag{14}$$

$$\sum_{i\in \boldsymbol{I}} QB_{i,p} \geq \; dem_p-\sum_{j\in \boldsymbol{J_p}} iij_{p,j} \;\;\forall\; p\in \boldsymbol{P} \tag{15}$$

Monitoring the inventory of component s at the start/end times of a production run

Total Amount of Gasoline Blending Components Assigned to a Production Run. The continuous variable $QS_{s,i,p}$ represents the amount of component $s\in S$ assigned to run i producing product p. According to Eq. 16, the total amount of components assigned to run i should be equal to the production of the final product yielded by run i.

$$\sum_{s\in S} QS_{s,i,p}=\; QB_{i,p} \quad \forall\; i\in \boldsymbol{I},\;\; p\in \boldsymbol{P} \tag{16}$$

Amounts of Components Assigned to a Production Run. Equation 17 provides the value of $QS_{s,i,p}$. The parameters $\{\; vc^{\min}_{s,p},\;\; vc^{\max}_{s,p}\}$ denote the limiting volume fractions of component s in the final product p.

$$vc^{\min}_{s,p}\; QB_{i,p} \;\leq\; QS_{s,i,p} \;\leq\; vc^{\max}_{s,p}\; QB_{i,p} \;\;\forall\; s\in \boldsymbol{S},\;\; i\in \boldsymbol{I},\;\; p\in \boldsymbol{P} \tag{17}$$

Total Amount of Component s Consumed by Production Runs up to the Completion Time of Run i. The total amount of component s consumed up to time CB_i given by the continuous variable $US_{s,i}$ is obtained by summing up the amounts of component s assigned to production runs $i'\leq i$.

$$US_{s,i}=\sum_{\substack{i'\in I\\ i'\leq i}}\;\sum_{p\in \boldsymbol{P}} QS_{s,i',p} \quad \forall\; i\in \boldsymbol{I},\;\; s\in \boldsymbol{S} \tag{18}$$

Inventories of Component s at the Start/End Times of a Production Run. The inventory level of every component tank should be monitored to avoid overloading and running-out conditions at the start/end time of every production run. The continuous variables $SINI_{s,i}$ and $SINC_{s,i}$ represent the inventory levels of component s at the start and completion times of run i. Equation 19a provides the inventory of component s at the start of run i. Its value must be lower than the capacity of the tank assigned to component s. Assuming that the production run i is assigned to time period k, then Eq. 19a accounts for: (a) the initial inventory of component s; (b) the amount of component s loaded into the dedicated tank over the periods $k'<k$; (c) the amount of component s loaded during the period k up to the start time of run i; (d) the amount of component s discharged from the assigned tank and allocated to production runs $i'<i$. In Eq. 19a, the continuous variable $LKS_{i,k}$ denotes the length of the time interval between the beginning of period k and the start time of run i. Its value is given by Eqs. 19b and 19c. Obviously, $LKS_{i,k}$ is equal to zero if run i is not assigned to period k.

$$SINI_{s,i}=iis_s+\sum_{k\in \boldsymbol{K}}\left[\left(\sum_{\substack{\boldsymbol{k'}\in K\\ \boldsymbol{k'}<k}} svr_{s,k'}\;\; hk_{k'}\right)WI_{i,k}+svr_{s,k}\;\; LKS_{i,k}\right]-\left(US_{s,i}-\sum_{p\in \boldsymbol{P}} QS_{s,i,p}\right)\leq scap_s \;\;\forall\; s\in \boldsymbol{S},\;\; i\in \boldsymbol{I} \tag{19a}$$

$$LKS_{i,k}\geq SB_i-llim_k+hk_k\;\left(1-WI_{i,k}\right) \quad \forall\; i\in \boldsymbol{I},\;\; k\in \boldsymbol{K} \tag{19b}$$

$$LKS_{i,k}\leq hk_k\;\; WI_{i,k} \quad \forall\; i\in \boldsymbol{I},\;\; k\in \boldsymbol{K} \tag{19c}$$

Similarly, Eq. 20a provides the inventory of component s at the completion of run i given by the continuous variable $SINC_{s,i}$. In this case, $LKF_{i,k}$ denotes the length of the time interval between the beginning of period k and the completion of run i. Its value is given by Eqs. 20b and 20c. The inventory of component s at the horizon end ($SINF_s$) is determined by Eq. 20d. To avoid running-out conditions for component s, the condition $SINC_{s,i}\geq 0$ is to be satisfied for any run i. In turn, $SINF_s$ must never exceed the capacity of the assigned tank ($scap_s$).

$$SINC_{s,i}=iis_s+\sum_{k\in \boldsymbol{K}}\left[\left(\sum_{\substack{\boldsymbol{k'}\in K\\ \boldsymbol{k'}<k}} svr_{s,k'}\;\; hk_{k'}\right)WI_{i,k}+\; svr_{s,k}\;\; LKF_{i,k}\right]-\; US_{s,i} \;\;\forall\; s\in \boldsymbol{S},\;\; i\in \boldsymbol{I} \tag{20a}$$

$$LKF_{i,k}\leq CB_i-\; llim_k\;\; WI_{i,k} \;\;\forall\; i\in \boldsymbol{I},\;\; k\in \boldsymbol{K} \tag{20b}$$

$$LKF_{i,k}\leq hk_k\;\; WI_{i,k} \;\;\forall\; i\in \boldsymbol{I},\;\; k\in \boldsymbol{K} \tag{20c}$$

$$SINF_s = iis_s + \sum_{k\in K} svr_{k,s}\ hk_k - \sum_{i\in I}\sum_{p\in P} QS_{s,i,p} \le scap_s \quad \forall\ s\in S \tag{20d}$$

Equation 19a assumes that the blending runs preceding run i are all completed before starting run i. Moreover, Eq. 20a supposes that the blending runs succeeding run i do not start before ending run i. In other words, blending runs do not overlap. Therefore, Eqs. 19a and 20a are strictly valid only if a single blender is available. When multiple blenders are operated, overlapping events can occur and the equations controlling the inventory in component tanks should be modified. Let us define the binary variable $ZO_{i',i}$ denoting the overlapping of blending runs (i', i) with $i' < i$ whenever $ZO_{i',i} = 1$. In contrast, no overlapping of runs (i', i) will occur if $ZO_{i',i} = 0$. Then, the binary variable $ZO_{i',i}$ is defined by Eqs. 21a and 21b with the parameter ε standing for a very small number. The subset IP_i comprises the preceding runs ($i' < i$) performed in other blenders that can overlap in time with run i. It is given by: $IP_i = \{i' \in I|\ i' < i \cap i' > i - card(B)\}$.

$$SB_i \le CB_{i'} - \varepsilon + H\ (1 - ZO_{i',i}) \quad \forall\ i\in I,\ i' \in IP_i \tag{21a}$$

$$SB_i \ge CB_{i'} - H\ ZO_{i',i} \quad \forall\ i\in I,\ i' \in IP_i \tag{21b}$$

To account for overlapping of blending runs when multiple blenders are operated, Eqs. 19a and 20a will be generalized but preserving the model linearity. Because Eq. 19a intends to avoid the overloading of component tanks, the amount of component s consumed by run i' up to time SB_i will be ignored in the new expression of that constraint if $ZO_{i',i} = 1$. This is a conservative assumption based on the worst case at which both runs (i', i) with $i' < i$ start at the same time, i.e., $SB_{i'} = SB_i$. In turn, Eq. 20a has been included in the model to avoid running-out conditions in component tanks. If runs (i, i') with $i < i'$ overlap in time, the worst case occurs when they finish at the same time, i.e., $CB_i = CB_{i'}$. Then, the new expression for Eq. 20a will assume that the total amount of component s assigned to run i' (with $i' > i$) has been totally consumed at time CB_i when $ZO_{i,i'} = 1$. For the multi-blender case, Eqs. 19a–20a should be replaced by Eqs. 22a and 22b. In Eq. 22b, the subset IS_i is given by $IS_i = \{i\prime \in I|\ i' > i\ \cap\ i' < i + card(B)\}$.

$$SINI_{s,i} = iis_s + \sum\nolimits_{k\in K}\left[\left(\sum_{\substack{k' \in K\\ k' < k}} svr_{s,k'}\ hk_{k'}\right) WI_{i,k} + svr_{s,k}\ LKS_{i,k}\right] - \left(US_{s,i} - \sum_{p\in P} QS_{s,i,p}\right) + \sum\nolimits_{i' \in IP_i} QSO_{s,i',i} \le scap_s\ \ \forall\ s\in S,\ i\in I \tag{22a}$$

$$SINC_{s,i} = iis_s + \sum\nolimits_{k\in K}\left[\left(\sum_{\substack{k' \in K\\ k' < k}} svr_{s,k'}\ hk_{k'}\right) WI_{i,k} + svr_{s,k}\ LKF_{i,k}\right] - US_{s,i} - \sum_{i' \in IS_i} QSO_{s,i,i'} \ \ge\ 0\ \ \forall\ s\in S,\ i\in I \tag{22b}$$

The continuous variable $QSO_{s,i',i}$ arising in constraints 22a and 22b are defined by Eqs. 22c and 22d. $QSO_{s,i',i}$ is equal to $QS_{s,i'}$ in Eq. 22a providing the value of $SINI_{s,i}$ if $ZO_{i',i} = 1$. The parameter MS_s is a relatively large number.

$$QSO_{s,i',i} \le MS_s\ ZO_{i',i} \tag{22c}$$

$$QSO_{s,i',i} \ \ge\ \sum_{p\in P} QS_{s,i',p} - MS_s\ (1 - ZO_{i',i}) \quad \forall\ s\in S,\ i\in I,\ i' \in IP_i \tag{22d}$$

Fulfillment of the gasoline quality specifications

Controlling the Value of Every Critical Property g in the Final Product p. Equations 23a and 23b seek to make on-spec final products within the desired limits { $ppr^{\min}_{g,p}$, $ppr^{\max}_{g,p}$}, using linear blending indices, on volume or weight additive base. The parameter $spr_{g,s}$ represents the blending index of property g for the component s. In Eq. 23b, ρ_s represents the density of component s.

$$ppr^{\min}_{g,p}\ QB_{i,p} \le \sum_{s\in S} spr_{g,s}\ QS_{s,i,p} \le ppr^{\max}_{g,p}\ QB_{i,p} \quad \forall\ i\in I, p\in P,\ g\in G \tag{23a}$$

$$ppr^{\min}_{g,p}\left(\sum_{s\in S} \rho_s\ QS_{s,i,p}\right) \le \sum_{s\in S} spr_{g,s}\ \rho_s\ QS_{s,i,p} \le\ ppr^{\max}_{g,p}\left(\sum_{s\in S} \rho_s\ QS_{s,i,p}\right) \quad \forall\ i\in I,\ p\in P,\ g\in G \tag{23b}$$

Allocating the production from blenders to final product tanks

Assigning Production Runs to Storage Tanks. The 0–1 variable $XIJ_{i,j,t}$ denotes the assignment of the production from run i to the final product tank j during the time slot t whenever $XIJ_{i,j,t}$ is equal to one. By Eq. 24, a production run should at most be assigned to a single product tank and performed within a single time slot.

$$\sum_{t\in T}\sum_{j\in J} XIJ_{i,j,t} = \sum_{b\in B} WB_{i,b} \quad \forall\ i\in I \tag{24}$$

In turn, Eq. 25 states that a product tank can at most receive a single production run during a time slot.

$$\sum_{i \in I} XIJ_{i,j,t} \leq 1 \quad \forall\ j \in \boldsymbol{J},\ t \in \boldsymbol{T} \tag{25}$$

Assigning Storage Tanks to Final Products During a Time Slot. The binary variable $XPJ_{p,j,t}$ denotes the assignment of the storage tank j to the final product p during the time slot t whenever $XPJ_{p,j,t}=1$. By Eq. 26, a storage tank should be allocated to a single product over a time slot. The set P_j includes the final products that can be stored in tank j.

$$\sum_{p \in \boldsymbol{P_j}} XPJ_{p,j,t} = 1 \quad \forall\ t \in \boldsymbol{T},\ j \in \boldsymbol{J} \tag{26}$$

Moreover, the production from a campaign of product p (i.e., $YB_{i,p}=1$) must be discharged into a storage tank assigned to that product. Assuming that run i yields product p, then the variable $XIJ_{i,j,t}$ can be equal to one only if $XPJ_{p,j,t}=1$ by Eq. 27.

$$XIJ_{i,j,t}+YB_{i,p} \leq 1+XPJ_{p,j,t} \quad \forall\ i \in I,\ j \in J,\ p \in \boldsymbol{P_j},\ t \in \boldsymbol{T} \tag{27}$$

Amount of Product Loaded into a Storage Tank During a Time Slot. The continuous variable $QPJ_{i,p,j,t}$ stands for the amount of product p from run i discharged into the storage tank $j \in J_p$ during the time slot t. As specified by Eqs. 28a and 28b, its value is zero if either the production run i does not yield product p (i.e., $YB_{i,p}=0$) or the run i is not assigned to tank j during the slot t (i.e. $XIJ_{i,j,t}=0$). Otherwise, it is equal to $QB_{i,p}$ by Eq. 28c.

$$QPJ_{i,p,j,t} \leq dem_p\ XIJ_{i,j,t} \quad \forall\ i \in I,\ j \in J,\ p \in P_j,\ t \in \boldsymbol{T} \tag{28a}$$

$$\sum_{t \in \boldsymbol{T}} \sum_{j \in \boldsymbol{J_p}} QPJ_{i,p,j,t} \leq dem_p\ YB_{i,p}\ \forall\ p \in \boldsymbol{P},\ i \in \boldsymbol{I} \tag{28b}$$

$$\sum_{t \in \boldsymbol{T}} \sum_{j \in \boldsymbol{J_p}} QPJ_{i,p,j,t} = QB_{i,p}\ \forall\ p \in \boldsymbol{P},\ i \in \boldsymbol{I} \tag{28c}$$

Product Transition Cost in Storage Tanks. The positive variable $TRJ_{p,j,t}$ represents the changeover cost in the product tank j when product p stored during the slot t is replaced by another product $p' \neq p$ in the next slot $(t+1)$. The value of $TRJ_{p,j,t}$ is given by Eq. 29a. The parameter $ctrj_j$ denotes the product changeover cost in tank j per instance.

$$TRJ_{p,j,t} \geq ctrj_j \left(XPJ_{p,j,t} + \sum_{\substack{\boldsymbol{p'} \in \boldsymbol{P_j} \\ \boldsymbol{p'} \neq p}} XPJ_{p',j,t+1} - 1 \right) \quad \forall\ p \in \boldsymbol{P},\ j \in \boldsymbol{J_p},\ t \in \boldsymbol{T} \tag{29a}$$

If sequence-dependent transition costs are to be handled, Eq. 29a is to be replaced by Eq. 29b.

$$TRJ_{p,j,t} \geq ctrj_{j,p,p'} \left(XPJ_{p,j,t}+XPJ_{p',j,t+1}-1 \right) \quad \forall\ p,p' \in \boldsymbol{P}\ (p \neq p'),\ j \in \boldsymbol{J_p} \cap \boldsymbol{J_{p'}},\ t \in \boldsymbol{T} \tag{29b}$$

Sequencing time slots

Ordered Set of Time Slots. A common set of process time slots T for all product tanks is defined. It is said that T is an ordered set because the time slot t must begin just after finishing the precedence slot $(t-1)$ as specified by Eq. 30.

$$ST_t \geq CT_{t-1} \quad \forall\ (t-1), \quad t \in \boldsymbol{T} \tag{30}$$

Initial and Final Times of Production Runs Assigned to Slot t. By Eqs. 31a and 31b, a production run assigned to the time slot t must be performed within the interval $\{ST_t, CT_t\}$. The length of slot t is given by Eq. 32.

$$ST_t \leq SB_i + h \left(1 - \sum_{j \in \boldsymbol{J}} XIJ_{i,j,t} \right) \quad \forall\ i \in \boldsymbol{I},\ t \in \boldsymbol{T} \tag{31a}$$

$$CB_i \leq CT_t + h \left(1 - \sum_{j \in \boldsymbol{J}} XIJ_{i,j,t} \right) \quad \forall\ i \in \boldsymbol{I},\ t \in \boldsymbol{T} \tag{31b}$$

$$LT_t = CT_t - ST_t \quad \forall\ t \in \boldsymbol{T} \tag{32}$$

Total Length of the Time Slots. Equation 33 states that the total length of the time slots should be equal to the horizon length h.

$$\sum_{t \in \boldsymbol{T}} LT_t = h \tag{33}$$

Unloading final products from the storage tanks to satisfy customer demands

Non-Simultaneous Loading and Unloading Operations in Storage Tanks. The binary variable $XDJ_{p,j,t}$ denotes the delivery of the final product p from tank j during the time slot t whenever $XDJ_{p,j,t}$ is equal to one. Equation 34 imposes that receipt and delivery operations cannot be performed in a product tank within the same time slot.

$$\sum_{p \in \boldsymbol{P_j}} XDJ_{p,j,t} + \sum_{i \in \boldsymbol{I}} XIJ_{\boldsymbol{i,j,t}} \leq 1 \quad \forall\ t \in \boldsymbol{T},\ j \in \boldsymbol{J} \tag{34}$$

Amount of Product p Unloaded from a Storage Tank During a Time Slot. The continuous variable $UP_{p,j,t}$ represents the amount of product p unloaded from the product tank j during the slot t to satisfy customer orders for product p. Its value should be equal to zero in case: (a) the tank j receives the production from a blender during the slot t, (b) the tank j has not been assigned to product p during the slot t, or (c) no delivery of product p from tank j is planned during the slot t. Such conditions are expressed by Eqs. 35–37, respectively. If none of those conditions holds, the unloaded volume of product p can never exceed the available inventory of p in tank j at the end of the preceding slot $(t-1)$ as established by Eq. 38.

$$\sum_{p \in P_j} UP_{p,j,t} \leq pcap_j \left(1 - \sum_{i \in I} XIJ_{i,j,t} \right) \quad \forall\ j \in \boldsymbol{J},\ t \in \boldsymbol{T} \tag{35}$$

$$XDJ_{p,j,t} \leq XPJ_{p,j,t} \quad \forall\ j \in J,\ p \in P_j,\ t \in \boldsymbol{T} \tag{36}$$

$$UP_{p,j,t} \leq pcap_j\ XDJ_{p,j,t} \quad \forall\ j \in J,\ p \in P_j,\ t \in \boldsymbol{T} \tag{37}$$

$$UP_{p,j,t} \leq PINV_{p,j,t-1} \quad \forall\ j \in J,\ p \in P_j,\ t \in \boldsymbol{T} \tag{38}$$

Meeting the Total Demand of Product p. By Eq. 39, the total volume of product p unloaded from all the storage tanks over the scheduling horizon should be large enough to meet all the orders requiring product p.

$$\sum_{t \in \boldsymbol{T}} \sum_{j \in J_p} UP_{p,j,t} \geq \sum_{r \in R_p} q_r \quad \forall\ p \in \boldsymbol{P} \tag{39}$$

Controlling the inventory of final products in storage tanks

Inventory Level of Final Products in Storage Tanks Along the Scheduling Horizon. The positive variable $PINV_{p,j,t}$ denotes the inventory of product p in tank $j \in J_p$ at the end of slot t. Its value, given by Eq. 40, accounts for: (a) the initial inventory of product p in tank j ($iij_{p,j}$), (b) the volume of product p discharged from a blender into the tank j during the slot t, and (c) the amount of p unloaded from tank $j \in J_p$ to meet customer orders over the slot t. By Eq. 41, the value of $PINV_{p,j,t}$ should never exceed the capacity of tank j.

$$PINV_{p,j,t}=iij_{j,p}+\sum_{i\in \boldsymbol{I}}\sum_{\substack{\boldsymbol{t}'\in T\\ t'\le t}} QPJ_{i,p,j,t'} - \sum_{\substack{\boldsymbol{t}'\in T\\ t'\le t}} UP_{p,j,t'} \quad \forall\ j\in \boldsymbol{J},\ p\in \boldsymbol{P_j},\ t\in \boldsymbol{T} \tag{40}$$

$$PINV_{p,j,t} \le pcap_j\ XPJ_{p,j,t}\ \forall\ j\in \boldsymbol{J},\ p\in \boldsymbol{P_j},\ t\in \boldsymbol{T} \tag{41}$$

Conditions for the Assignment of Tank $j \in J_p$ to Product p During the Time Slot t. Eq. 42 specifies that a storage tank $j \in J_p$ allocated to product p during the time slot t cannot contain another product at the end of slots ($t-1$) and t.

$$\begin{gathered} PINV_{p',j,(t-1)} \le pcap_j\ \left(1-XPJ_{p,j,t}\right) \\ \forall\ (t-1),t\in \boldsymbol{T},\ p,p'\in \boldsymbol{P_j}\ (p'\ne p),\ j\in \boldsymbol{J_p}\cap \boldsymbol{J_{p'}} \end{gathered} \tag{42}$$

Assigning volumes of final products to customer orders

Allocating Deliveries of Final Products from Storage Tanks to Customer Orders. The binary variable $XRJ_{r,j,t}$ denotes that a portion or the whole amount of final product discharged from tank j during the time slot t has been allocated to customer order r. By Eq. 43, no assignment of product to orders requiring product p can be made if there is no discharge of p from tank j during the slot t, i.e. $XDJ_{p,j,t}=0$. The parameter nr_p represents the number of customer orders for the final product p.

$$\sum_{r\in \boldsymbol{R_p}} XRJ_{r,j,t} \le nr_p\ XDJ_{p,j,t} \quad \forall\ j\in J,\ p\in P_j,\ t\in \boldsymbol{T} \tag{43}$$

The continuous variable $UR_{r,j,t}$ stands for the volume of product assigned to order r from tank j within the slot t. Equation 44 requires that the whole amount of product discharged from a storage tank j should be entirely allocated to one or several customer orders requiring that product. By Eq. 45, the total amount of product assigned to customer order r from one or several tanks at the same or different time slots should be exactly equal to the order size.

$$UP_{p,j,t} = \sum_{r\in \boldsymbol{R_p}} UR_{r,j,t} \quad \forall\ j\in \boldsymbol{J},\ p\in \boldsymbol{P_j},\ t\in \boldsymbol{T} \tag{44}$$

$$\sum_{t\in \boldsymbol{T}}\sum_{j\in \boldsymbol{J}_r} UR_{r,j,t}= q_r \quad \forall\ r\in \boldsymbol{R} \tag{45}$$

Limiting the Amount of Product Delivered from a Storage Tank for Any Customer Order. Equations 46a and 46b restrict the value of $UR_{r,j,t}$ to the interval ($sr_{\min}$, q_r), where $sr_{\min}$ is the minimum amount of product coming from a storage tank that can be assigned to a customer order. If the parameter rdr_r stands for the delivery rate of order r demanding product p and $pdr_{p,j}$ denotes the maximum delivery rate of product p from tank j, then the value of $UR_{r,j,t}$ is also limited by Eqs. 47a and 47b.

$$UR_{r,j,t} \ge sr_{\min}\ XRJ_{r,j,t} \quad \forall\ j\in \boldsymbol{J},\ r\in \boldsymbol{R},\ t\in \boldsymbol{T} \tag{46a}$$

$$UR_{r,j,t} \le q_r\ \ XRJ_{r,j,t} \quad \forall\ j\in \boldsymbol{J},\ r\in \boldsymbol{R},\ t\in \boldsymbol{T} \tag{46b}$$

$$UR_{r,j,t} \le rdr_r\ (CT_t- ST_t) \quad \forall\ p\in \boldsymbol{P},\ j\in \boldsymbol{J_p},\ t\in \boldsymbol{T} \tag{47a}$$

$$\sum_{r\in \boldsymbol{R}_p} rdr_r\ XRJ_{r,j,t} \le pdr_{p,j} \quad \forall\ j\in \boldsymbol{J_p},\ \boldsymbol{p}\in \boldsymbol{P},\ t\in \boldsymbol{T} \tag{47b}$$

Starting and Completion Times of Product Deliveries from Storage Tanks. The positive variable $CR_{r,j,t}$ represents the completion time for the delivery of order r from tank j during the time slot t. Assuming that every unloading operation begins at the start of time slot t and ST_t satisfies the constraint (48), then the value of $CR_{r,j,t}$ is given by Eq. 49. Besides, the delivery of a customer order should occur within a single slot. Then, $CR_{r,j,t}$ should not be greater than CT_t if $XRJ_{r,j,t}=1$ by Eqs. 47a and 49.

$$ST_t \ge atw_r\ XRJ_{r,j,t} \quad \forall\ r\in \boldsymbol{R},\ j\in \boldsymbol{J_r},\ t\in \boldsymbol{T} \tag{48}$$

$$\begin{gathered} CR_{r,j,t} \ge ST_t+ \left(UR_{r,j,t}/rdr_r\right)-h\ \left(1- XRJ_{r,j,t}\right) \\ \forall\ r\in \boldsymbol{R},\ j\in \boldsymbol{J_r},\ t\in \boldsymbol{T} \end{gathered} \tag{49}$$

Tardiness of Customer Order r. The continuous variable CRF_r is the time at which the customer order r is completely satisfied. Its value is given by Eq. 50. Then, the tardiness of order r (TD_r) can be determined by Eq. 51.

$$CRF_r \ge CR_{r,j,t}\ \forall\ r\in \boldsymbol{R},\ j\in \boldsymbol{J_r},\ t\in \boldsymbol{T} \tag{50}$$

$$TD_r \ge CRF_r- btw_r\ \forall\ r\in \boldsymbol{R} \tag{51}$$

Objective function

The problem goal (52) is the minimization of the total operating cost, including the component cost, the transition costs in blenders and product tanks, and the tardiness costs. In Eq. 52, the parameter ctd stands for the tardiness cost per unit time, and $scost_s$ is the unit cost of component s.

$$\min\ Z=\left(\sum_{s\in \boldsymbol{S}}\sum_{i\in \boldsymbol{I}}\sum_{p\in P} scost_s\ QS_{s,i,p} + \sum_{r\in \boldsymbol{R}} ctd\ TD_r + \sum_{b\in \boldsymbol{B}} TTRB_b + \sum_{p\in \boldsymbol{P}}\sum_{j\in \boldsymbol{J}_p}\sum_{\substack{t\in T\\ t<|\boldsymbol{T}|}} TRJ_{j,p,t}\right) \tag{52}$$

Simultaneous receipt/delivery operations in product tanks

A few changes should be introduced in the proposed formulation to handle simultaneous receipt/delivery operations in product tanks. Let us suppose that the order deliveries assigned to time slot t finish at time CT_t instead of assuming that they start at time ST_t. In this way, running-out conditions in product tanks can be avoided by monitoring their inventory levels at the end time of every slot. The required changes in the mathematical formulation consist on removing the constraints 34 and 35 and replacing Eqs. 48 and 49 by constraints 53–54b.

$$ST_t \geq atw_r \; XRJ_{r,j,t} \quad \forall \; r \in \boldsymbol{R}, \; j \in \boldsymbol{J_r}, \; t \in \boldsymbol{T} \tag{53}$$

$$SR_{r,j,t} \geq CT_t - \left(\frac{UR_{r,j,t}}{rdr_r}\right) + h \; \left(1 - XRJ_{r,j,t}\right) \quad \forall \; r \in \boldsymbol{R}, \; j \in \boldsymbol{J_r}, \; t \in \boldsymbol{T} \tag{54a}$$

$$CR_{r,j,t} \leq CT_t + h \; \left(1 - XRJ_{r,j,t}\right) \quad \forall \; r \in \boldsymbol{R}, \; j \in \boldsymbol{J_r}, \; t \in \boldsymbol{T} \tag{54b}$$

Simple rules for selecting the number of production runs and time slots

Two cases will be considered to develop simple rules for choosing the number of production runs. When the gasoline components are available in enough quantities, the optimal recipe of a particular product is independent of the other products to be blended during the time horizon (Case 1). Such optimal product recipes are usually called the *preferred product recipes*. Under limited amounts of components, the preferred recipes for all products are no longer obtained. The optimal *constrained* recipe of a blend will depend on the other products to be made and the amounts of them required by the customers (Case 2). Assuming that the available amount of each component is given, blending problems with relative low product demands belong to Case 1. As the product demands rise, blending problems can switch to Case 2.

Let us assume that $\mu^*_{s,p}$ stands for the fraction of component s in the preferred recipe of product p. The preferred product recipes are usually achieved (Case I) when the following condition holds:

$$iis_s + \sum_{k \in \boldsymbol{K}} svr_{s,k} \; hk_k \gg \sum_{p \in \boldsymbol{P}} \mu^*_{s,p} \; dem_p \tag{55}$$

The symbol ($\gg$) in Eq. 55 means that the total availability of each component is sufficiently greater than the optimal component demand to avoid temporal running-out conditions in component tanks along the time horizon. For problems belonging to Case 1, the rule for choosing the number of production runs is given by:

$$|I| = |\hat{\boldsymbol{P}}| \tag{56}$$

where $\hat{\boldsymbol{P}}$ is the set of products to be blended. Generally $\hat{\boldsymbol{P}} = \boldsymbol{P}$, but in some cases the demands of some products can be fully satisfied using the inventories initially available in product tanks.

For blending problems belonging to Case 2, the required quantities of some components to achieve the preferred product recipes exceed the available amounts and usually more runs are to be performed to minimize the total operating costs. Additional runs allow overcoming some temporal component shortages. For Case 2,

$$|I| = |\hat{P}| + n \;\; (\text{with } n \geq 1) \tag{57}$$

The value of n should be increased by one as the component shortages grow. For problem with sizable component shortages, it is recommended $n = 3$. In turn, the criterion proposed for choosing the number of time slots for Cases 1 and 2 is given by:

$$|T| = |I| + 1 \tag{58}$$

In few instances, the MILP model may become infeasible using the number of time slots given by Eq. 58. In such cases, the value of |T| should be increased by one. As shown in the Supporting Information, the number of time slots has usually no impact on the optimal solution and a minor effect on the CPU time.

Computational Results and Discussion

The proposed approach has been applied to single-period (SP) and multi-period (MP) instances of 14 examples first introduced by Li et al.[10] and later studied by Li and Karimi[12] and Castillo-Castillo and Mahalec[14]. Data for the two instances of Examples 1–14 are given as Supporting Information. They can also be found in Li et al.[10] For both instances of every example, it is given: (a) the list of customer orders to be satisfied together with the required product, order size, delivery rate, and delivery time window; (b) the product tanks, tank capacities, the storable products, the current stored product, initial inventories, and maximum delivery rates; (c) the component tanks and their capacities, initial stocks and maximum delivery rates; (d) the limiting proportions of the gasoline components in the final products; (5) the limiting values of the critical properties for the different gasoline grades demanded by the customers; (6) the available blenders together with the products that can be processed, the allowable processing rates and the minimum length of the production runs; and (7) economic data including the component and demurrage unit costs and transition costs (in blenders and product tanks) per instance. Example 1 is the simplest case study involving five orders for only two final products, one blender, nine gasoline components, one critical property, and a time horizon of 72 h. In turn, Examples 2–14 comprise 10–45 orders, 3–5 final products, 1–3 blend headers, 9 gasoline components, 9 gasoline critical properties, 11 product tanks, and a time horizon of 192 h. At the initial time, the blenders are idle except for Example 5. In Example 5, the unique blender is processing the final product P_1 at time zero and the production run has already a length of 10 h and a production volume of 150 kbbl. Multiperiod scenarios are also defined for Examples 1–14 involving 2–4 time periods. All the examples have been solved using GAMS/CPLEX 24.2 on an Intel(R) Core i7 3632QM 2.20 GHz one-processor PC with 12 GB RAM and four cores. The relative optimality gap tolerance has been fixed to 0.001 for all examples. Besides, a maximum CPU time of 3600 s was allowed.

Computational results are shown in Tables 1 and 2. In addition, the best solutions found for the SP and MP-instances of Examples 1–14 are all illustrated in a series of 26 figures presented as Supporting Information. Such figures contain the Gantt chart showing the sequence of operations in blenders and product tanks, and the inventory profiles in component

Table 1. Computational Results for the SP and MP-Instances of Examples 1–14.

Example		Our approach			Li and Karimi (2011)		
		Cost ($\times 10^3$ \$)	CPU time (s)	Switching operations	Cost ($\times 10^3$ \$)	CPU time (s)	Switching operations
1	SP	5139.93	0.34	1	5149.73	0.89	2
	MP	5139.93	0.34	1	5149.73	1.25	2
2	SP	3658.11	0.55	1	3658.11	2.12	1
	MP	3658.11	0.58	1	3658.11	0.69	1
3	SP	3159.12	0.39	1	3159.12	2.62	1
	MP	3159.12	0.36	1	3159.12	1.16	1
4–5	SP	4556.67	0.27	1	4556.67	4.20	1
	MP	4556.67	0.27	1	4556.67	1.97	1
6	SP	5213.88	0.50	2	5213.88	2.41	2
	MP	5213.88	0.95	2	5213.88	1037	2
7	SP	8100.35	1.9	3	8100.35	3024	3
	MP	8100.35	1.7	3	8100.35	10,814	3
8	SP	8080.35	2.7	2	8080.35	10,802	2
	MP	8080.35	1.4	2	8082.85	14,170	2
9	SP	10,573.65	8.8	4	10,573.65	10,819	4
	MP	10,576.74	2.7	4	10,573.65	10,817	4
10	SP	11,289.80	8.3	4	11,286.10	10,814	4
	MP	11,313.23	7.0	4	11,306.10	11,735	5
11	SP	13,248.58	818.8	4	13,248.58	14,400	4
	MP	13,282.98	1987.0	4	13,248.58	36,075	4
12	SP	14,774.14	641.1	4	14,764.86	46,800	5
	MP	15,221.74	23.0	4	14,809.36$^{(1)}$	46,800	7
13	SP	17,986.65	1003.9	3	15,646.15$^{(2)}$	118,800	6
	MP	18,609.73	47.4	3	16,187.58$^{(3)}$	118,800	5
14	SP	20,352.22	1544.6	3	17,737.39$^{(4)}$	118,800	9
	MP	21,101.43	134.0	3	18,678.48$^{(5)}$	118,800	9

LP relaxation bound = {15,147.24$^{(1)}$, 17,918.26$^{(2)}$, 18,555.23$^{(3)}$, 20,286.52$^{(4)}$, 21,046.93$^{(5)}$}.

and product tanks. Table 1 presents the optimal objective values for the SP and MP instances of Examples 1–14, and the CPU solution times. It also includes the results reported by Li and Karimi[12]. It is observed that both instances of Examples 4 and 5 share the same solution. In turn, Table 2 shows the optimal values for the different cost items (component, transition, and demurrage costs) at the best solution, the selected number of production runs and the LP relaxation bound for every example. Values for the tardiness costs are not included because orders are delivered on time at all examples. Although Examples 3 and 4 both involve three products, two production runs have been selected to solve such examples (see Table 2). This is so because the demand of P_3 is satisfied using the initial inventory of that product. Then, no production of P_3 is needed and $|\hat{\boldsymbol{P}}| = |I| = 2$. The number of time slots chosen for each example is reported in the Supporting Information. In

Table 2. Values of the Different Cost Items for Examples 1–14 (SP and MP Instances)

Example		Componentcost (10^3 \$)	Switching cost (10^3 \$)		Number of runs	Total cost (10^3 \$)	RMIP bound (10^3 \$)
			Blenders	Product tanks			
1	SP	5119.93	20	–	3	5139.93	5119.93
	MP	5119.93	20	–	3	5139.93	5119.93
2	SP	3638.11	20	–	2	3658.11	3638.11
	MP	3638.11	20	–	2	3658.11	3638.11
3	SP	3139.12	20	–	2	3159.12	3139.12
	MP	3139.12	20	–	2	3159.12	3139.12
4–5	SP	4536.67	20	–	2	4556.67	4536.67
	MP	4536.67	20	–	2	4556.67	4536.67
6	SP	5173.88	40	–	3	5213.88	5173.88
	MP	5173.88	40	–	3	5213.88	5173.88
7	SP	8040.35	60	–	4	8100.35	8040.35
	MP	8040.35	60	–	4	8100.35	8040.35
8	SP	8040.35	40	–	4	8080.35	8040.35
	MP	8040.35	40	–	4	8080.35	8040.35
9	SP	10,499.15	60	14.5	5	10,573.65	10,499.15
	MP	10,502.24	60	14.5	5	10,576.74	10,502.24
10	SP	11,215.30	60	14.5	5	11,289.80	11,211.60
	MP	11,238.73	60	14.5	5	11,313.23	11,235.04
11	SP	13,174.08	60	14.5	7	13,248.58	13,174.08
	MP	13,208.98	60	14.5	7	13,282.98	13,205.71
12	SP	14,699.64	60	14.5	8	14,774.14	14,689.92
	MP	15,147.24	60	14.5	8	15,221.74	15,147.24
13	SP	17,932.15	40	14.5	8	17,986.65	17,918.26
	MP	18,555.23	40	14.5	8	18,609.73	18,555.23
14	SP	20,297.72	40	14.5	9	20,352.22	20,286.52
	MP	21,046.93	40	14.5	9	21,101.43	21,046.93

Table 3. Optimal Blend Recipes for the SP Instance of Examples 1–12

		Component fraction (%)								
Ex.	Product	C1	C2	C3	C4	C5	C6	C7	C8	C9
1	P1	**22.00**	**10.00**	–	–	**25.00**	–	0.39	**30.00**	12.61
	P2	**24.00**	**10.00**	–	–	**25.00**	–	3.50	**30.00**	7.50
2–5	P1	1.27	38.73	–	**40.00**	–	**20.00**	–	–	–
	P4	**24.00**	**10.00**	–	**44.00**	16.95	5.05	–	–	–
6	P1	1.27	38.73	–	**40.00**	–	**20.00**	–	–	–
	P3	10.00	29.00	–	**43.00**	–	**18.00**	–	–	–
	P4	**24.00**	**10.00**	–	**44.00**	16.95	5.05	–	–	–
7–8	P1	1.27	38.73	–	**40.00**	–	**20.00**	–	–	–
	P2	–	31.98	–	**45.00**	–	**22.00**	–	1.02	–
	P3	10.00	29.00	–	**43.00**	–	**18.00**	–	–	–
	P4	**24.00**	**10.00**	–	**44.00**	16.95	5.05	–	–	–
9 &11	P1	1.27	38.73	–	**40.00**	–	**20.00**	–	–	–
	P2	–	31.98	–	**45.00**	–	**22.00**	–	1.02	–
	P3	10.00	29.00	–	**43.00**	–	**18.00**	–	–	–
	P4	24.00	10.00	–	**44.00**	16.95	5.05	–	–	–
	P5	–	30.76	–	**40.00**	–	**20.00**	–	9.24	–
10	P1	1.27	38.73	–	**40.00**	–	**20.00**	–	–	–
	P2	–	31.98	–	**45.00**	–	**22.00**	–	1.02	–
	P3	10.00	29.00	–	**43.00**	–	**18.00**	–	–	–
	P4	**24.00**	**10.00**	–	**44.00**	16.95	5.05	–	–	–
	P5	–	30.55	–	**40.00**	–	**20.00**	1.69	7.76	–
12	P1	1.27	38.73	–	**40.00**	–	**20.00**	–	–	–
	P2	–	31.98	–	**45.00**	–	**22.00**	–	1.02	–
	P3	10.00	29.00	–	**43.00**	–	**18.00**	–	–	–
		6.92	30.23	–	**41.15**	3.70	**18.00**	–	–	–
	P4	**24.00**	**10.00**	–	29.70	19.51	16.79	–	–	–
	P5	–	30.76	–	**40.00**	–	**20.00**	–	9.24	–

Table 4. Optimal Blend Recipes for the MP Instance of Examples 1–12

		Component fraction (%)								
Ex.	Product	C1	C2	C3	C4	C5	C6	C7	C8	C9
1	P1	**22.00**	**10.00**	–	–	**25.00**	–	0.39	**30.00**	12.61
	P2	**24.00**	**10.00**	–	–	**25.00**	–	3.50	**30.00**	7.50
2–5	P1	1.27	38.73	–	**40.00**	–	**20.00**	–	–	–
	P4	**24.00**	**10.00**	–	**44.00**	16.95	5.05	–	–	–
6	P1	1.27	38.73	–	**40.00**	–	**20.00**	–	–	–
	P3	10.00	29.00	–	**43.00**	–	**18.00**	–	–	–
	P4	**24.00**	**10.00**	–	**44.00**	16.95	5.05	–	–	–
7–8	P1	1.27	38.73	–	**40.00**	–	**20.00**	–	–	–
	P2	–	31.98	–	**45.00**	–	**22.00**	–	1.02	–
	P3	10.00	29.00	–	**43.00**	–	**18.00**	–	–	–
	P4	**24.00**	**10.00**	–	**44.00**	16.95	5.05	–	–	–
9	P1	1.27	38.73	–	**40.00**	–	**20.00**	–	–	–
	P2	–	31.98	–	**45.00**	–	**22.00**	–	1.02	–
	P3	10.00	29.00	–	**43.00**	–	**18.00**	–	–	–
	P4	**24.00**	**10.00**	–	42.27	17.26	6.47	–	–	–
	P5	–	30.76	–	**40.00**	–	**20.00**	–	9.24	–
10	P1	1.27	38.73	–	**40.00**	–	**20.00**	–	–	–
	P2	–	31.98	–	**45.00**	–	**22.00**	–	1.02	–
	P3	10.00	29.00	–	**43.00**	–	**18.00**	–	–	–
	P4	**24.00**	**10.00**	–	30.84	17.30	15.86	–	–	–
	P5	–	30.55	–	**40.00**	–	**20.00**	1.69	7.76	–
11	P1	1.27	38.73	–	**40.00**	–	**20.00**	–	–	–
	P2	–	31.98	–	**45.00**	–	**22.00**	–	1.02	–
	P3	10.00	29.00	–	**43.00**	–	**18.00**	–	–	–
	P4	**24.00**	**10.00**	-	26.40	20.10	19.50	–	–	–
		24.00	**10.00**	–	32.79	18.95	14.25	–	–	–
	P5	–	30.76	–	**40.00**	–	**20.00**	–	9.24	–
12	P1	–	**10.00**	32.02	29.27	12.01	16.70	–	–	–
	P2	–	**10.00**	16.17	**45.00**	7.42	21.41	–	–	–
		–	26.78	6.22	**45.00**	–	**22.00**	–	–	–
	P3	–	19.98	0.16	**43.00**	18.87	**18.00**	–	–	–
		–	**10.00**	55.16	–	16.84	**18.00**	–	–	–
	P4	–	75.00	11.10	–	13.90	–	–	–	–
		–	62.10	–	–	**25.00**	12.90	–	–	–
	P5	–	**15.00**	18.84	**40.00**	–	**20.00**	–	6.16	–

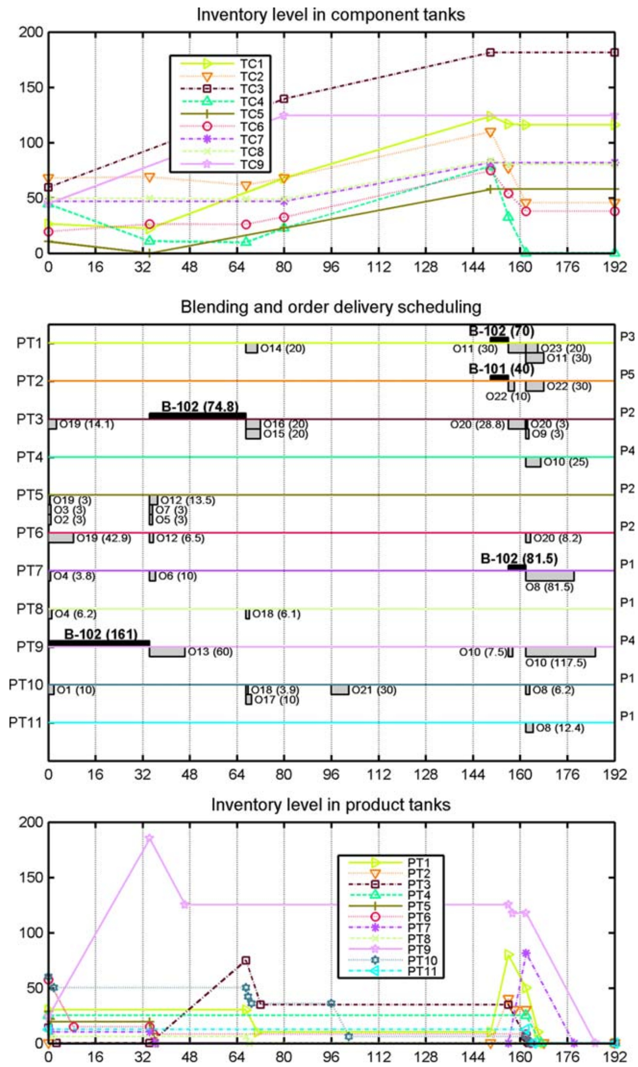

Figure 3. Best solution found for the MP-instance of Example 9.

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

addition, Tables 3 and 4 include the product recipes selected at the best solutions found for the SP and MP instances of Examples 1–12, respectively.

From Table 1, it can be said that the proposed formulation is able to find the best solutions for both instances of Examples 1–8 in less than 5 s of CPU time. When our results are compared with the ones found by Li and Karimi[12], it is observed a sharp reduction in the computational cost, especially for the SP and MP instances of Examples 7–14. Indeed, the two approaches mostly discover the same solutions for the SP and MP versions of Examples 1–8. Then, we will focus on those cases where we produce different results. Using our approach, better solutions have been found for the SP and MP instances of Example 1, and the MP-instance of Example 8. Figures illustrating them are presented in the Supporting Information. At Example 1, the proposed solutions avoid changeovers in product tanks by receiving the new production of final products P_1 and P_2 in product tanks initially containing those gasoline grades. In this way, the transition cost in product tanks each amounting to \$9800 is saved and the total operating cost decreases from \$5,149,730 to \$5,139,930 (see Table 1). In fact, only one changeover in the blender occurs along the time horizon. Conversely, the cost saving for the MP instance of Example 8 amounts to \$2500, i.e., 1-hour demurrage cost. The improvement was achieved by finding a solution where all customer orders are delivered on time. In turn, Li and Karimi[12] discovered lower-cost solutions for the MP instance of Examples 9 and 11. Both solutions feature optimal values similar to the single-period instances of those examples. The common operating cost reported by Li and Karimi[12] for both variants of Example 9 amounts to \$10,573,650. However, such results for the MP instance of Examples 9 and 11 may be questionable as explained next.

Figure 3 illustrates the best solution provided by our approach for Example 9(MP) that includes a Gantt chart displaying the sequence of operations in blenders and product tanks, and the inventory profiles for component and product tanks. It was found in 2.7 s with an absolute gap equal to zero when the solution algorithm stops. As shown in the upper part of Figure 3, Example 9(MP) presents some limitation in the availability of component C_4. At Example 9(SP), the feed flow rate of C_4 into the assigned storage tank remains equal to 1.0 (kbbl/h) during the entire planning horizon. In contrast, it

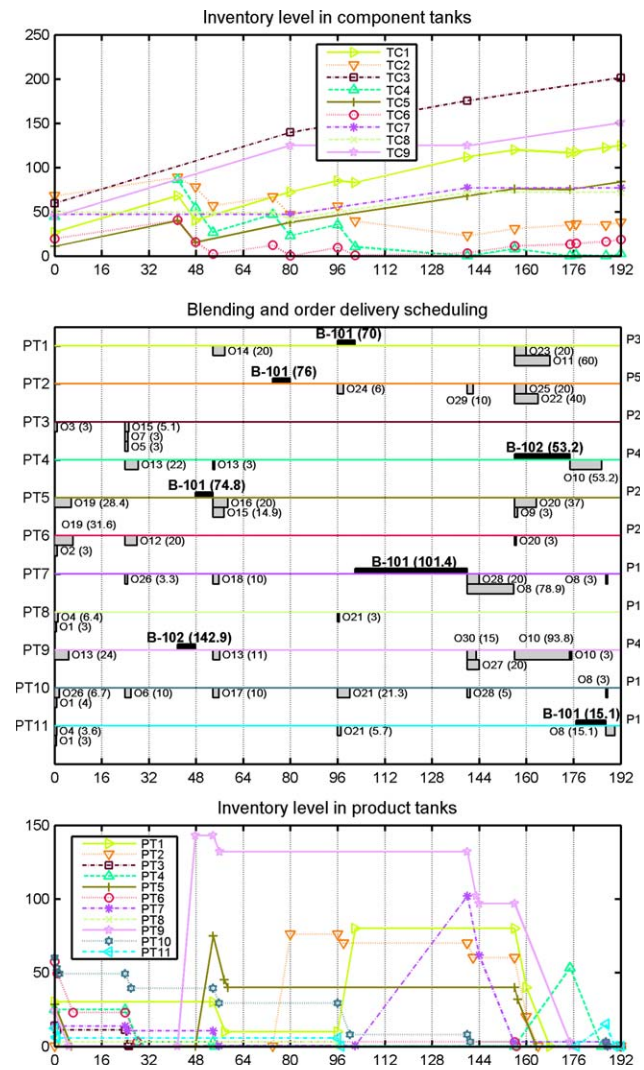

Figure 4. Best solution found for the MP-instance of Example 11.

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

Table 5. Comparison with Previous Results Reported by Castillo-Castillo and Mahalec[14]

	Our approach			Castillo-Castillo et al. (2015)		
Example MP-instance	Cost (10^3 \$)	CPU time (s)	Transition in product tanks	Cost (10^3 \$)	CPU time (s)	Transition in product tanks
3	3159.12	0.36	0	3159.1	1.9	0
4	4556.67	0.27	0	4556.7	2.02	0
7	8100.35	13.0	0	8100.3	36.9	0
8	8080.35	39.5	0	8080.3	45.8	0
9	10,576.74	3.2	1	10,778.8	210.0	1
12	15,221.74	18.8	1	15,221.7	1342.0	1
14	21,101.43	119.0	1	21,101.4	2877.0	1

decreases from 1.0 to 0.8 (kbbl/h) at time = 80 h and drops to zero at time = 162 h for Example 9(MP). Looking at Tables 3 and 4, we see that C_4 is the dominant component in the preferred recipes of all final products systematically selected at both instances of Examples 2–8. To still stick with the preferred product recipes and satisfy the blend demands in Example 9(MP), we need to have available 183.23 (kbbl) of component C_4. However, the total volume of C_4 obtained by adding the receiving flows from upstream units to the initial inventory is: $(44.44 + 80.00 + 56.00) = 180.44$ (kbbl) < 183.23 (kbbl) at Example 9(MP). This shortage of C_4 produces a slight deviation from the preferred recipe of product P_4 (see Table 4). That change produces an increase of \$3090 in both the LP relaxation bound and the total component cost (see Table 2). Consequently, the optimal value for Example 9(MP) rises to \$10,576,740 slightly larger than the one for Example 9(SP). Then, the best solution reported by Li and Karimi[12] for Example 9(MP) cannot be feasible. From Table 4, the preferred recipes are still adopted for the other finished products $\{P_1, P_2, P_3, P_5\}$. For Example 9(SP), the available amount of C_4 given by: $44.44 + 192.00 = 236.44$ (kbbl) is large enough to still adopt the preferred recipes (see Table 3). A figure illustrating the optimal solution for Example 9(SP) is included in the Supporting Information.

Similar comments can be made on the result found by Li and Karimi[12] for Example 11(MP). Again, they reported the same optimal value for the SP and MP instances of Example 11. However, a shortage of gasoline component C_4 arises in Example 11(MP) because of a decreasing feed flow rate from upstream units along the planning horizon. To select the preferred product recipes, it is needed an amount of component C_4 equal to 227.03 kbbl, but the total volume available is: $44.44 + 80.00 + 48.00 + 26.00 = 198.44$ kbbl. As a result, a deviation from the preferred recipe of product P_4 occurs (see Table 4). The shortage of C_4 produces an increase of the operating cost with regards to the SP-instance from \$13,248,580 to \$13,282,980, i.e., an increment of \$34,400. Therefore, the solution reported by Li and Karimi[12] for Example 11(MP) may be infeasible. In contrast, no shortage of C_4 occurs in Example 11(SP) and the preferred recipes for all products are still chosen at the best solution (see Table 3). In this case, the total available volume of C_4 is: $44.44 + 192 = 236.44$ (kbbl). Figure 4 illustrates the best solution found with our approach for Example 11(MP). Similar differences arise in the SP and MP instances of Example 10. It should be emphasized that the best solutions discovered by our approach for Examples 10(SP) and 10(MP) both feature an absolute gap equal to zero.

The proposed MILP formulation presents a remarkable computational efficiency simply because it has a very small integrality gap. Accounting for the values of the LP relaxation bound (at the root node) given by the last column of Table 2, the ratio between the best integer solution and the initial LP relaxation bound for the SP and MP instances of Examples 1–14 is always below 1.0075. Moreover, the starting LP bound is quite close or equal to the optimal component cost (see Table 2). Therefore, the difference between the integer solution and the LP relaxation bound is mostly due to the product transition costs in blenders and product tanks. For the purpose of comparison, Tables 1 and 2 include the number of switching operations performed on the best solutions found with our formulation and those reported by Li and Karimi.[12] In every example, the number of changeovers using our formulation is

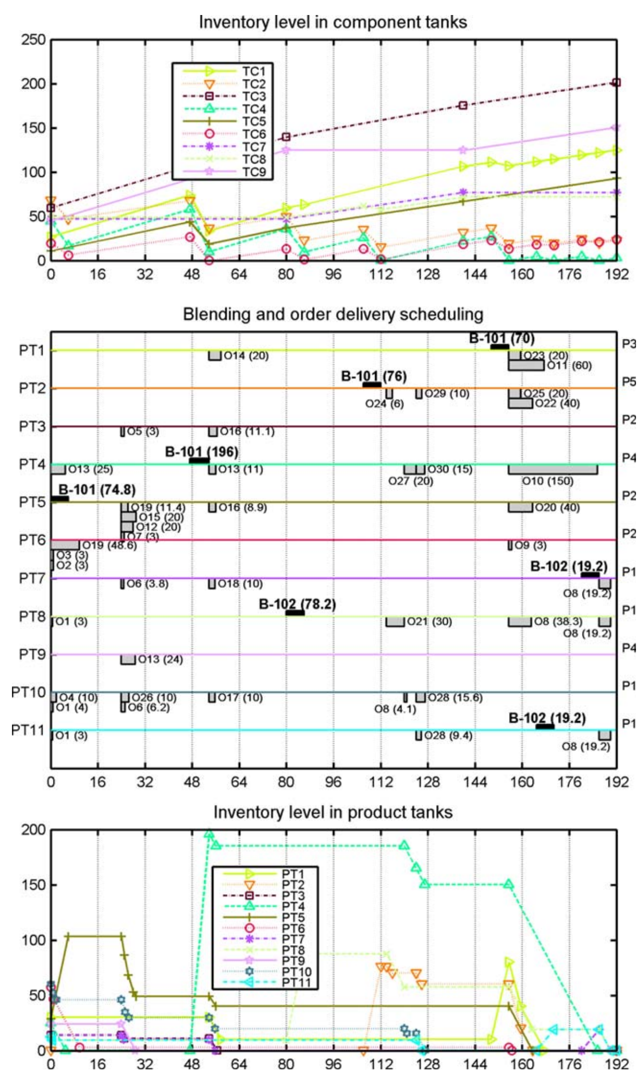

Figure 5. Best solution found for the MP-instance of Example 12.

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

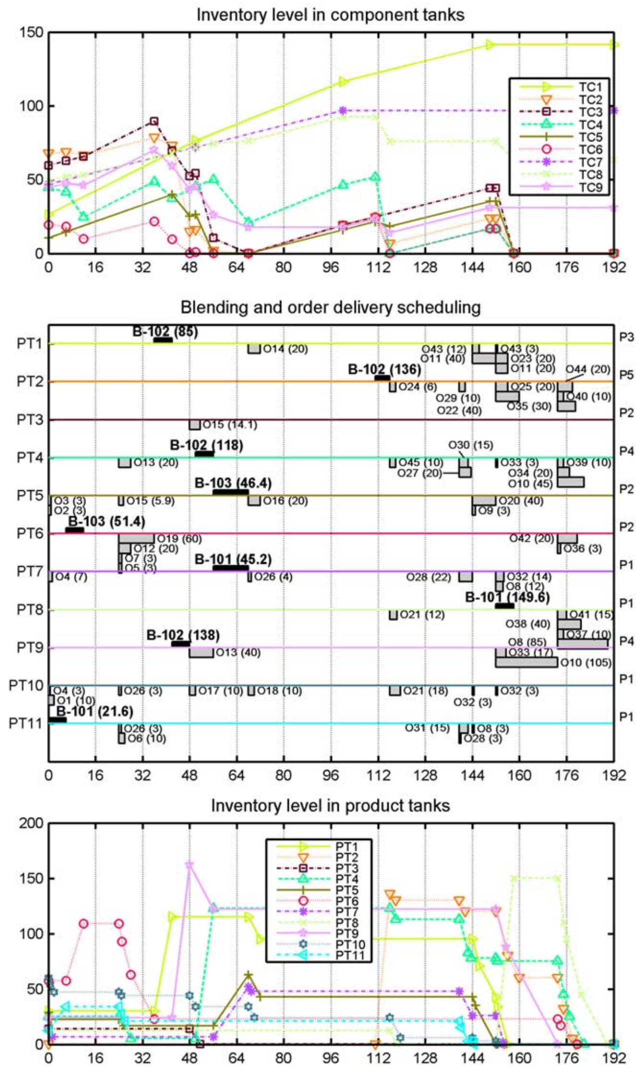

Figure 6. Best solution found for the MP-instance of Example 14.

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

always equal or smaller (especially for larger examples) than the ones performed in Li and Karimi.[12] In fact, the best solutions found for the two instances of Examples 1–14 always exhibit the least possible number of changeovers in blenders and product tanks. Just for Examples 9–14, a single changeover P_3–P_5 is additionally required in the product tank PT-2 simply because no product tank is initially devoted to product P_5. The other product tanks are devoted to a unique product all over the scheduling horizon and no changeover is needed. Orders are always delivered on time to avoid the payment of penalties for tardy orders in every example. Moreover, no inventory remains in product tanks at the end of the scheduling horizon. Thus, it is produced only what is needed to satisfy customer orders. In this way, the total operating cost is minimized. As shown in the Supporting Information, the number of time slots has usually no impact on the optimal solution found and a weak influence on the solution time.

Table 5 shows a comparison with the results recently published by Castillo-Castillo and Mahalec[14] for the MP-instances of Examples {3, 4, 7, 8, 9, 12, and 14}. Solutions similar to those reported by Castillo-Castillo and Mahalec[14] were found for the examples {3, 4, 7, 8, 12, and 14} at smaller CPU times, but our approach leads to a better solution of Example 9(MP). To reduce the computational cost, Castillo-Castillo and Mahalec[14] estimated a tight lower bound for the component and switching costs by solving an aggregate optimization model. Solutions and computational cost shown in Table 5 were obtained by Castillo-Castillo and Mahalec[14] through applying that bound. To solve examples {3, 4, 7, 8, 9}, they selected a number of blending runs similar to those used in our work and shown in Table 2. Let us have another look at Table 1. We can see that the optimal values reported by Li and Karimi[12] for Example 12(MP) and both instances of Examples 13 and 14 are smaller than the LP relaxation bound at the root node. We believe that they correspond to infeasible solutions. Results obtained for Castillo-Castillo and Mahalec[14] for the MP-instance of Examples 12 and 14 seem to confirm that presumption. Figures 5 and 6 illustrate the best solutions provided by our approach for the MP-instance of Examples 12 and 14, respectively. They reveal shortages of components C_2, C_4, and C_6 for Example 12(MP) and insufficient amounts of components C_2–C_6 for Example 14(MP). Such shortages produce a substantial increase in the values of both the LP bound and the optimal integer solution as the number of orders rises. In addition, deviations from the preferred recipes are observed for all products at those examples (see Tables 3 and 4).

The best solutions for both variants of Examples 1–8 present the same optimal objective value (see Table 2). As shown in Tables 3 and 4, they all systematically select the same recipes for the final products P_1–P_5. Such recipes can be regarded as the preferred product recipes. Every final product p is always produced using the preferred recipe even if multiple production runs are performed along the time horizon. Then, Examples 2–8 can be viewed as blend scheduling problems with unlimited component inventories. As discussed before, the preferred recipes are still chosen for the gasoline grades produced in the two instances of Example 9 with only one exception. This exception is caused by a slight shortage of component C_4 in the MP-instance of Example 9. Consequently, the recipe used for product P_4 undergoes a minor change to overcome that restraint in the amount of C_4 (see Table 4). In fact, the selected feed flow rates of components from upstream units for the MP-instance of Examples 9–14 produce increasing component shortages as the number of orders grows. The earlier lack of components in the MP-examples gives rise to a growing discrepancy between the least operating costs for the SP and MP versions of Examples 9–14. The rising shortage of gasoline components first cause some limited changes on the recipe of the lowest-quality product P_4 and then on the recipes of all products, especially at Examples 12–14. Contrarily, such component shortages for the SP-variant just appear at Example 12 and become more important for Examples 13 and 14.

Gasoline components that reach the allowed minimum/maximum fractions in the optimal product recipes for Examples 1–12 appear in bold type in Tables 3 and 4. The fraction of C_4 is at its maximum value in the recipes of the final products P_1–P_5, and is the largest one for every product. This explains why it is the first component presenting shortages at Example 9(MP). The other preferred component C_6 reaches its maximum allowed proportion in the recipes of products {P_1, P_2, P_3, P_5} but compared with C_4 is relatively less required. Similarly to what happen with the product recipes, the values of

Table 6. Optimal Product Property Indices for the MP Instance of Examples 1–12(MP)

		Property								
Ex.	Product	RBN	RVI	SULI	BI	AROI	OI	DNI	FLI	OXI
1	P1	**110.45**	–	–	–	–	–	–	–	–
	P2	**111.95**	–	–	–	–	–	–	–	–
2–5	P1	**110.45**	53.046	13.574	0.390	12.595	9.374	1.455	4.618	0.536
	P4	**103.24**	94.017	17.323	0.302	9.170	2.724	1.511	3.536	0.454
6	P1	**110.45**	53.046	13.574	0.390	12.595	9.374	1.455	4.618	0.536
	P3	**108.97**	59.749	15.454	0.363	11.693	7.131	1.473	4.345	0.520
	P4	**103.24**	94.017	17.323	0.302	9.170	2.724	1.511	3.536	0.454
7–8	P1	**110.45**	53.046	13.574	0.390	12.595	9.374	1.455	4.618	0.536
	P2	**111.95**	50.708	11.041	0.317	10.334	7.772	1.467	4.459	0.673
	P3	**108.97**	59.749	15.454	0.363	11.693	7.131	1.473	4.345	0.520
	P4	**103.24**	94.017	17.323	0.302	9.170	2.724	1.511	3.536	0.454
9	P1	**110.45**	53.046	13.574	0.390	12.595	9.374	1.455	4.618	0.536
	P2	**111.95**	50.708	11.041	0.317	10.334	7.772	1.467	4.459	0.673
	P3	**108.97**	59.749	15.454	0.363	11.693	7.131	1.473	4.345	0.520
	P4	**103.24**	93.952	17.269	0.302	9.170	2.736	1.509	3.532	0.440
	P5	**115.01**	55.987	10.638	0.326	11.525	7.479	1.454	4.581	1.737
10	P1	**110.45**	53.046	13.574	0.390	12.595	9.374	1.455	4.618	0.536
	P2	**111.95**	50.708	11.041	0.317	10.334	7.772	1.467	4.459	0.673
	P3	**108.97**	59.749	15.454	0.363	11.693	7.131	1.473	4.345	0.520
	P4	**103.24**	93.527	16.913	0.305	9.170	2.811	1.494	3.508	0.821
	P5	**115.01**	54.419	10.662	0.320	11.174	7.426	1.451	4.537	3.000
11	P1	**110.45**	53.046	13.574	0.390	12.595	9.374	1.455	4.618	0.536
	P2	**111.95**	50.708	11.041	0.317	10.334	7.772	1.467	4.459	0.673
	P3	**108.97**	59.749	15.454	0.363	11.693	7.131	1.473	4.345	0.520
	P4	**103.24**	93.600	16.973	0.304	9.170	2.798	1.497	3.512	0.366
		103.24	93.362	16.774	0.305	9.170	2.841	1.488	3.499	0.316
	P5	**115.01**	55.987	10.638	0.326	11.525	7.479	1.454	4.581	1.737
12	P1	**110.45**	54.676	12.557	0.372	11.926	9.107	1.457	4.552	0.529
	P2	**111.95**	50.708	11.041	0.317	10.334	7.772	1.467	4.459	0.673
	P3	**108.97**	72.578	7.447	0.217	6.424	5.028	1.487	3.886	0.471
		108.97	70.998	8.433	0.235	7.073	5.287	1.490	3.448	0.497
	P4	**103.24**	93.339	16.755	0.306	9.170	2.845	1.487	3.497	0.311
		103.24	80.534	25.039	0.559	17.926	9.272	1.449	4.502	0.460
	P5	**115.01**	55.987	10.638	0.326	11.525	7.479	1.454	4.581	1.737

the gasoline blend properties also remain the same at the best solution of both instances of Examples 1–8. The optimal blend properties for the MP-instance of Examples 1–12 are listed in Table 6. At all examples, the best solutions have a common, constant feature. None of the gasoline properties are at their limiting values with the exception of the RBN index associated to the RON that is always set to its lowest limit, and the OXI index that reaches the upper limit for product P_5 at Example 10(SP).

Conclusions

Based on the notion of floating time slots, a new MILP approach for the simultaneous optimization of the gasoline blend recipes and the short-term planning of blending and distribution operations has been developed. By relying on a very tight MILP formulation with an integrality gap close to zero, the approach is able to discover the best solutions for large benchmark problems at much lower computational time than previous contributions. All the typical operations rules for the management of gasoline blending processes have been considered. Moreover, the proposed model handles several usual features of the gasoline blend unit such as the presence of multiple non-identical blenders, multipurpose product tanks, sequence-dependent changeover costs in blenders and product tanks, limited amounts of gasoline components, and multiperiod scenarios with component feed flow rates from upstream units changing with the period. In contrast to previous works, the assignment of floating slots and production runs to time periods is made by the model. By so doing, we reduce the model size, preserve the optimality of the solution and substantially close the integrality gap.

The approach was successfully applied to a significant number of gasoline blend optimization problems. Single-period and multi-period scenarios were considered in all cases. When compared with the results reported in previous works, it is observed a sharp reduction in the computational cost, especially for the larger problems. In addition, the best solutions provided by the proposed MILP formulation present some interesting features: (a) they exhibit the least number of changeovers in blenders and product tanks; (b) orders are all delivered on time to avoid penalties for tardy orders; (c) no inventory remains in product tanks at the end of the scheduling horizon, i.e., it is produced what is needed to satisfy customer orders; and (d) the RON property for all final products is at the allowed minimum value, while the quality constraints for the other properties of every product are almost always redundant at the optimum. Results also show the impact of component shortages on the optimal recipes of the final products. Under unlimited stocks of components, the optimal product recipe can be individually established before determining the short-term planning of blending operations. This solution scheme can be done for Examples 1–8. When the number of orders rises and shortages of some preferred components appear, the recipe of a final product will depend on the other gasoline grades and their demands. As a result, the product recipes and the scheduling of blend operations are to be simultaneously determined.

Supporting Information

The Supporting Information file contains a set of tables with all the problem data, the model sizes and the selected number of time slots for Examples 1–14. It also includes the figures showing the best solutions for the single-period and multi-period instances of all examples.

Acknowledgment

The authors acknowledge financial support from FONCYT-ANPCyT under Grant PICT 2010-1073, from CONICET under Grant PIP-2221, and from Universidad Nacional del Litoral under CAI+D 2011-256.

Notation

Sets

B = blender headers
K = ordered set of time periods
I = ordered set of production runs in blenders
P = final products
$\mathbf{P_j}$ = final products that can be stored in tank j
$\boldsymbol{T}$ = process time slots

Parameters

$ctrb_{b,p,p'}$ = sequence-dependent changeover cost in blender b
ctd = penalty cost per unit time for tardy orders
dem_p = total demand of final product p to be satisfied during the scheduling horizon
$iij_{p,j}$ = initial inventory of product p in the tank $j \in \boldsymbol{J_p}$
iis_s = initial inventory of gasoline component s
$llim_k$, $ulim_k$ = time limits for period k
M_B = a large number
$pcap_j$ = capacity of the product tank j
$pdr_{p,j}$ = delivery rate of product p from product tank j
$ppr_{g,p}^{\min}, ppr_{g,p}^{\max}$ = limiting values of property g per unit amount of product p
q_r = size of order r
$rb_{b,p}^{\min}, rb_{b,p}^{\max}$ = processing rate limits for final product p in blender b
rdr_r = delivery rate of product for order r
$scap_s$ = capacity of the dedicated tank for component s
$scost_s$ = unit cost of component s
$spr_{g,s}$ = value of property g per unit amount of component s
$sr_{\min}$ = minimum amount of a customer order that can be delivered by a product tank
$svr_{s,k}$ = feed flow rate of component s during period k
$vc_{s,p}^{\min}, vc_{s,p}^{\max}$ = limiting proportions of component s in the final product p
$vc_{s,k}$ = feed rate of blending component s during time period k
$\tau_{b,p,p'}$ = sequence-dependent changeover time in blender b

Binary variables

$XDJ_{p,j,t}$ = denotes the discharge of final product p from tank j during the time slot s
$XIJ_{i,j,t}$ = assigns production run i to product tank j during the time slot t
$XPJ_{,p,j,t}$ = assigns the storage tank j to the final product p during the time slot t
$XRJ_{r,j,t}$ = denotes the discharge of order r to product tank j during the time slot t
$YB_{i,p}$ = allocates production runs to final products
WB_{ib} = allocates production runs to blenders
WI_{ik} = assigns production runs to time periods
$ZO_{i',i}$ = identifies overlapping of runs (i',i) with $i' < i$

Positive continuous variables

CB_i = completion time of run i
$CR_{r,j,t}$ = completion time for the delivery of order r from product tank j during the slot t
CRF_r = time at which the customer order r is completely satisfied
CT_t = final time of slot t
$LB_{i,b,p}$ = length of the production run i within the range $[lb_{b,p}^{\min},\ lb_{b,p}^{\max}]$
$LKF_{i,k}$ = length of time between the beginning of period k and the completion of run i
$LKS_{i,k}$ = length of time between the beginning of period k and the start of run i
L_t = length of time slot t
$PINV_{p,j,t}$ = inventory of final product p in tank $j \in J_p$ at the end of slot t
$QB_{i,p}$ = amount of final product p yielded by run i
$QPJ_{i,p,j,t}$ = amount of product p from run i discharged into tank $j \in J_p$ during the slot t
$QS_{s,i,p}$ = amount of component $s \in S$ assigned to run i producing product p
$QSO_{s,i',i}$ = amount of component s assigned to run i' that overlaps with run i
SB_i = starting time of run i
$SINC_{s,i}$ = inventory level of component s available at the completion time of run i
$SINF_s$ = inventory level of component s at the end of the scheduling horizon
$SINI_{s,i}$ = inventory level of component s available at the start of run i
ST_t = starting time of slot t
$SR_{r,j,t}$ = starting time for the delivery of order r from product tank j during the slot t
$TRB_{i,b}$ = cumulative transition cost in blender b up to run i
$TRJ_{p,j,t}$ = product transition cost in the product tank j
$TTRB_b$ = total transition cost in blender b
$UP_{p,j,t}$ = amount of product p unloaded from product tank j during the slot t
$UR_{r,j,t}$ = amount delivered for order r from tank j during the time slot t
$US_{s,i}$ = total amount of component s consumed up to time CB_i
$WBP_{i,b,p}$ = identifies the final product yielded by production run i in blender b

Literature Cited

1. Kelly JD, Mann H. Crude oil blend scheduling optimization: an application with multimillion dollar benefits. *Hydrocarb Proces.* 2003;82:47–51.
2. Aspen Technology Inc. (2016). Aspen refinery multi-blend optimizer. Available at: http://www.aspentech.com/petroleum-supply-chain/aspen-refinery-multi-blend-optimizer/. Last accessed January 2016.
3. Yadav S, Shaik MA. Short-term scheduling of refinery crude oil operations. *Ind Eng Chem Res.* 2012;51:9287–9299.
4. Castro P, Grossmann IE. Global optimal scheduling of crude oil blending operations with RTN continuous-time and multiparametric disaggregation. *Ind Eng Chem Res.* 2014;53:15127–15145.
5. Castro PM. New MINLP formulation for the multiperiod pooling problem. *AIChE J.* 2015;61:3728–3738.
6. Lotero I, Trespalacios, F, Grossmann IE, Papageorgiou DJ, Cheon M-S. An MILP-MINLP decomposition method for the global optimization of a source based model of the multiperiod blending problem. *Comp Chem Eng.* 2016;87:13–35.
7. Glismann K, Gruhn G. Short-term scheduling and recipe optimization of blending processes. *Comp Chem Eng.* 2001;25:627–634.
8. Jia Z, Ierapetritou M. Efficient short-term scheduling of refinery operations based on a continuous time formulation. *Comp Chem Eng.* 2004;28:101–1019.
9. Méndez CA, Grossmann IE, Harjunkoski I, Kabore P. A simultaneous optimization approach for off-line blending and scheduling of oil-refinery operations. *Comp Chem Eng.* 2006;30:614–634.
10. Li J, Karimi IA, Srinivasan R. Recipe determination and scheduling of gasoline blending operations. *AIChE J.* 2010;56:441–465.
11. Li J, Li WK, Karimi IA, Srinivasan R. Improving the robustness and efficiency of crude scheduling algorithms. *AIChE J.* 2007;53:2659–2680.
12. Li J, Karimi IA. Scheduling gasoline blending operations from recipe determination to shipping using unit slots. *Ind Eng Chem Res.* 2011;50:9156–9174.
13. Kolodziej S, Grossmann IE, Furman KC, Sawaya NW. A discretization-based approach for the optimization of the multiperiod blend scheduling problem. *Comp Chem Eng.* 2013;53:122–142.
14. Castillo-Castillo PA, Mahalec V. Improved continuous-time model for gasoline blend scheduling. *Comp Chem Eng.* 2016;84:627–646.

Manuscript received Oct. 28, 2015, and revision received Feb. 11, 2016.