I&EC research
Industrial & Engineering Chemistry Research

Article

pubs.acs.org/IECR

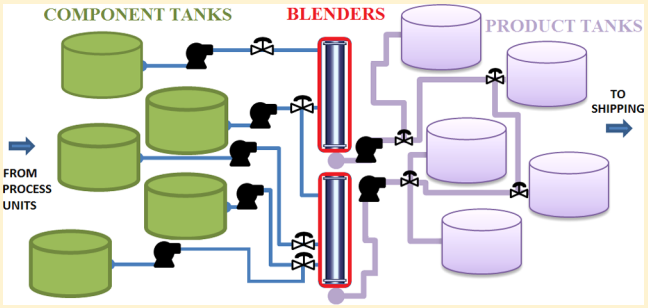
Optimizing Gasoline Recipes and Blending Operations Using Nonlinear Blend Models

Jaime Cerdá,* Pedro C. Pautasso, and Diego C. Cafaro

INTEC (UNL-CONICET), Güemes 3450, 3000 Santa Fe, Argentina

Supporting Information

ABSTRACT: Gasoline is one of the largest-volume products of the oil industry that yields 60%−70% of the total refinery revenues. This work presents a novel continuous-time mixed-integer nonlinear programming (MINLP) formulation for the gasoline blend scheduling problem. It incorporates nonlinear blending correlations for an improved prediction of key blend properties, and nonlinear constraints for precisely tracking the inventory level in product tanks when multiple blenders are operated. The approach handles nonidentical blenders, multipurpose tanks, sequence-dependent changeovers, limited amounts of gasoline components, and multiperiod scenarios with component flow rates changing with the period. Operating rules for blenders and product/component tanks are also considered. A special model feature is the use of floating slots dynamically allocated to time periods while solving the problem. An approximate mixed-integer linear programming (MILP) formulation assuming ideal mixing provides a good initial point. By fixing the integer variables, the resulting nonlinear programming (NLP) is then solved to find a near-optimal MINLP solution. Alternatively, a MINLP solver can be directly applied to the original MINLP formulation. Eleven benchmark examples have been successfully solved using the two solution strategies at rather low computational cost.



1. INTRODUCTION

Gasoline is one of the largest-volume products of the oil industry that yields 60%−70% of the total refinery revenues.[1] There are several important properties used to characterize the gasoline quality. Such properties are specified to guarantee an acceptable engine performance and the fulfillment of environmental regulations. They include the octane numbers (ONs), volatility, boiling range, sulfur and aromatics contents, and viscosity. ONs measure the antiknock properties of transportation fuels. Knocking occurs when a fuel/air mixture ignites prematurely in the combustion chamber producing a knocking sound, reducing the engine's power, and causing mechanical stress on the engine parts.[2] The ON of a fuel is defined as the percentage of iso-octane in a binary mixture with *n*-heptane that exhibits a resistance to detonation similar to the fuel being tested in a standard engine under standard conditions. On the octane scale, iso-octane is assigned an ON of 100, and for *n*-heptane, ON = 0. Two standard test procedures are used to characterize the antiknock properties of fuels for spark engines: the ASTM Standard D-2699 test, which gives the research octane number (RON), and the ASTM Standard D-2700 test, which provides the motor octane number (MON). The RON represents the antiknock property under conditions of low speed and frequent accelerations, while the MON measures the engine performance under more-severe high-speed conditions.[3]

Other major properties that affect the engine performance are the volatility and the boiling range, given in terms of the Reid vapor pressure (RVP) and the ASTM distillation points, respectively. A vapor pressure that is too high leads to vapor stalling and motor locking, while very low RVP will bring difficulties in the engine startup.[2] The RVP of a fuel is its absolute vapor pressure at 100 °F, as determined by the ASTM Standard D-323 test method. In turn, the boiling range also affects the engine during startup and driving, and is particularly important for good performance during quick acceleration and high-speed driving.[2] Government regulations place a maximum limit on the RVP to reduce the emission of volatile organic compounds into the atmosphere. Other environmental restrictions include maximum concentrations of aromatics, olefin, sulfur, and oxygenates.

Gasoline is indeed a complex mixture of hydrocarbons, additives, and blending agents. The feedstocks for the gasoline pool come from various intermediate production units such as catalytic reformers, alkylation and isomerization units, fluidized catalytic cracking (FCC) units, and hydrocrackers. The low-cost *n*-butane having a RON of 94 can also be a gasoline component, but its volume fraction is limited by the maximum RVP limit. Additives and blending agents are also incorporated in the hydrocarbon mixture to improve the antiknock

Received: April 22, 2016
Revised: June 15, 2016
Accepted: June 23, 2016
Published: June 23, 2016

DOI: 10.1021/acs.iecr.6b01566
Ind. Eng. Chem. Res. 2016, 55, 7782−7800

performance and the stability of the gasoline. These oxygenated compounds include octane enhancers, such as methyl *tert*-butyl ether (MTBE) and *tert*-butyl alcohol (TBA), and alternative fuels, such as ethanol and methanol, for economic and environmental reasons.

Gasoline blending is the process of combining several refinery streams of different compositions to make proper amounts of on-spec blends to fulfill product demands. Several different gasoline grades are usually produced in the gasoline blending unit (GBU) of a refinery, with each grade meeting some specific quality requirements. A schematic diagram of the GBU is shown in Figure 1. The gasoline components are

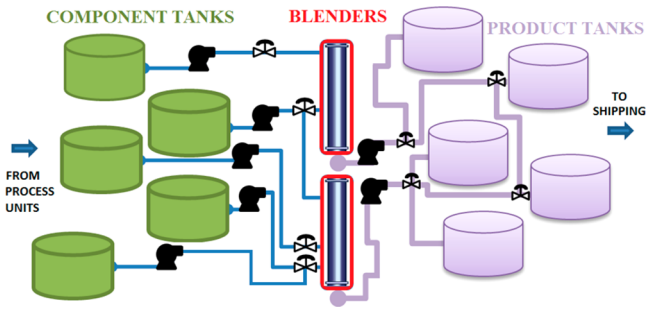


Figure 1. Schematic of the gasoline blending unit.

usually stored in a set of dedicated storage tanks and supplied to blenders at constant flow rates. At the same time, the component tanks can be receiving additional amounts of components from upstream production units.

Gasoline components can be blended in two different ways: (a) the traditional batch-blending process where the feedstocks are mixed in a blend tank, and (b) the tankless, inline blenders mixing the intermediate streams using flowmeters and control valves to obtain on-spec products.[4] Usually, several blenders are operated in a semicontinuous mode with each one making only one product at a time. Inline blending allows one to save money by reducing the blend time and avoiding the need for mixing tanks. The blending task is particularly difficult because some critical properties of the gasoline components like the ONs and the Reid vapor pressure do not mix in an ideal fashion, thus requiring nonlinear correlations for the accurate prediction of the gasoline quality. The finished products obtained in the blenders are discharged into a farm of nondedicated product tanks from which customer orders should be delivered to the market within specific time windows.

A large refinery can have more than 20 gasoline components. Some of them are more expensive but meet many quality specifications, while others are less expensive and fail to reach most of the required properties by themselves. One of the problem goals then is to choose the best proportion of each one in the gasoline blends, to satisfy their quality specifications and, at the same time, minimize the total feedstock cost (i.e., the optimal blend recipes). Quality giveaways arise when the refinery makes products of higher quality than required, thus reducing the profit margin. An octane giveaway of 0.1 can cost several millions of dollars per year.[5] In turn, reblending is performed when the production fails to reach the quality specifications. It may cause a significant cost in feedstocks, tank space, and blending time, and, consequently, a reduction in the overall capacity of the GBU.

The blending process should be performed, to maximize the total profit by simultaneously minimizing feedstock costs, quality giveaways, reblending operations, product changeovers in blenders and product tanks, and penalties for tardy orders. In other words, product recipes and the scheduling of blending and delivery operations are to be optimized. This work introduces a novel continuous-time mixed-integer nonlinear programming (MINLP) formulation for the gasoline blending problem. The proposed MINLP model is an extension of the MILP approach introduced by Cerdá et al.,[6] that now includes nonlinear blending equations for an improved prediction of key blend properties. Moreover, additional nonlinear constraints are included to (i) precisely track the inventory level in finished product tanks and, (ii) if necessary, force using the same recipe at all production runs of each finished product. The MINLP formulation is solved by using either a very efficient two-stage MILP-NLP strategy or a MINLP solver. In both cases, an approximate MILP problem representation is first solved to get a good initial point for either the NLP or the MINLP solver. The two solution strategies are able to determine near-optimal solutions at rather low computational cost.

2. PREVIOUS CONTRIBUTIONS

Numerous contributions on the gasoline blend scheduling problem have already been published. Some key differentiating features can be used to characterize them, such as (a) blending tanks or inline blenders for the mixing process; (b) single or multiple identical/nonidentical blenders; (c) dedicated or multipurpose product tanks; (d) fixed or variable product recipes: (d) linear or nonlinear correlations relating ONs and RVP with the blend composition; (e) nonsimultaneous or simultaneous input and output flows in product tanks; (f) constant or variable feedstock flow-rate from upstream units to component tanks along the time horizon; (g) ignoring or considering transition times and costs in blenders and product tanks; and (h) discrete or continuous-time formulations.

Previous works that assume inline blenders are first reviewed. Grismann and Gruhnn[7] developed a two-level integrated approach to coordinate the short-term scheduling of blending operations with nonlinear recipe optimization. At the top level, a nonlinear problem allows to determine blending recipes and product volumes. After fixing them, the blend scheduling problem modeled as an MILP is solved at the lower level. Jia and Ierapetritou[8] decomposed the overall refinery system into three major domains: (a) crude oil unloading, blending and processing; (b) production units yielding several feedstocks; and (c) gasoline blending and delivery of final products. By assuming fixed product recipes and a single blend header, they developed a continuous-time event-based MILP formulation for the simultaneous scheduling of gasoline blending and distribution operations. Méndez et al.[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 representation. The iterative procedure aims to preserve the model's linearity even though variable recipes and nonlinear mixing correlations are considered. The approach assumes the operation of parallel identical blenders, dedicated storage tanks, a constant feed component flow rate, and simultaneous loading and unloading operations at every product tank. Li et al.[10] developed a continuous-time MILP formulation based on process slots that incorporates new features, such as the operation of nonidentical blenders and multipurpose product tanks, multiperiod scenarios with component flow-rates varying as piecewise constant functions of time, and nonsimultaneous input and output flows in

DOI: 10.1021/acs.iecr.6b01566
Ind. Eng. Chem. Res. 2016, 55, 7782−7800

product tanks. By using linear mixing correlations, the problem was modeled through an MILP that ensures a constant blend rate per time slot. Because a production run may be extended over two or more slots, a schedule adjustment procedure is subsequently applied to get a constant blend rate during the entire run. The approach of Li et al.[10] was later improved by Li and Karimi,[11] who presented a multigrid continuous-time MILP formulation based on unit slots instead of process slots to get a better relaxed solution and a faster improvement of the lower bound. More importantly, the MILP model explicitly accounts for limited component inventories forcing the product recipes to change along the time horizon. By including additional constraints, simultaneous receipt/delivery operations in product tanks can also be considered. More recently, Castillo-Castillo and Mahalec[12] modified to some extent the MILP formulation of Li and Karimi[11] by incorporating new operational constraints, lower bounds on the objective function, and additional equations transforming binary variables into continuous ones. The proposed formulation accounts for product-dependent setup times in blenders, minimum production of blend runs and penalties for fulfilling the same order from multiple tanks. When nonlinear correlations are considered to estimate blend octane numbers, satisfactory results were obtained by using global MINLP solvers. Castillo-Castillo and Mahalec[13] also developed a new three-level approach that decomposes the blend scheduling problem into three stages: blend recipe optimization, approximate scheduling, and detailed scheduling. The first level determines the blend recipes by solving a discrete time LP (or NLP) model, the second level assumes fixed blend recipes to compute an approximate schedule via a discrete-time MILP model, and the third level uses a continuous-time MILP to exactly determine the start/stop time of each operation. In this way, the solution time is reduced by ∼3 orders of magnitude. Recently, Cerdá et al.[6] developed an efficient MILP continuous-time approach based on the use of floating slots and linear mixing correlations for predicting blend properties.

Some other contributions assumed the use of blend tanks for the mixing process. Kolodziej et al.[14] addressed the multiperiod blend scheduling problem, which is an extension of the pooling problem that accounts for time-varying feedstock supply and product demand. The blending infrastructure consists of a network of supply, blending, and demand tanks operating over a time horizon divided into a set of time periods. Flows cannot enter and exit a blending tank during the same time period. The problem goal is to determine the optimal flows between tanks to maximize the total profit while meeting the product demands within the specified composition limits. By using a radix-based discretization technique that was introduced by Teles et al.,[15] the MINLP formulation reduces to an approximate MILP that is incorporated into a heuristic solution procedure and in two rigorous global optimization methods. Castro[16] developed a new MINLP formulation for the multiperiod pooling problem that uses individual flows and split fractions to generate nonconvex bilinear terms better handled by a decomposition algorithm. In this way, global optimization solvers for MINLP can provide the best solutions of pooling problems. More recently, Lotero et al.[17] proposed an alternative formulation based on generalized disjunctive programming (GDP) that includes redundant constraints to improve the linear relaxation of the problem. In addition, an efficient solution procedure decomposes the MINLP model into two levels. The first level or master problem is an MILP relaxation of the original MINLP, and the second level is a smaller MINLP in which some of the binary variables have been fixed. The lower-level subproblem provides a rigorous lower bound when a feasible solution of the MINLP is found. The two subproblems are iteratively solved until the gap between the upper and the lower bound is closed.

3. ESTIMATING BLEND PROPERTIES USING NONLINEAR CORRELATIONS

Key blend properties that mostly determine the optimal product recipes are the octane numbers (ONs), i.e., the RON and MON properties. Assuming an ideal mixing, the octane number of the gasoline components will blend linearly on a volumetric basis, as stated by eq 1. **S** is the set of available gasoline components, w_s is the volumetric fraction of component s in the blend, and ON_s is the octane number of component s. Ideal mixing implies no interaction between gasoline components.

$$ON_{blend} = \sum_{s \in \mathbf{S}} w_s ON_s \tag{1}$$

However, ONs do not blend linearly. Positive or negative deviations from the value predicted by eq 1 usually arise. As stated by Rusin et al.,[18] the mixing of an olefin and a paraffin, both featuring RON = 90, may result in a blend with RON = 95. The mixture of an olefin and a paraffin usually presents positive deviations from the linear blending rule, while aromatic and olefin blends show slightly negative deviations. As a result, several empirical blending models that account for the nonlinear dependence of the octane numbers with the blend composition have been proposed in the literature. There are several characteristics that are desirable in a blending model. The most important are predictive accuracy, simplicity, and the model ability to remain accurate over a quality range beyond the one used to estimate the model parameters. Moreover, a good blending model should not require a large dataset to be periodically updated. Through better predictions, gasoline reblending can be avoided and millions of dollars per year can be saved.

The simplest nonideal blending model is based on fictitious octane numbers (BONs) of gasoline components that blend linearly on a volumetric basis.[3] Such BON values are derived from regression analysis of a small blend dataset. The approach has limited utility, because it remains accurate only for blends with similar components and quality range of the data used to estimate the BONs. Values of the BONs for some typical gasoline components can be found in Gary and Handwerk.[3]

$$ON_{blend} = \sum_{s \in \mathbf{S}} w_s BON_s \tag{2}$$

A more reliable method is the nonlinear Ethyl RT-70 scheme proposed by Healy et al.[19] It is one of the pioneer models available in the literature that is usually adopted as a benchmark to evaluate the performance of new gasoline blend correlations. Nonlinearities are modeled as functions of (i) the olefin (O_s) and aromatics (A_s) contents of every pure component s, and (ii) the blend component sensitivity, given by the difference ($RON_s - MON_s$). Each of these factors appears in different nonlinear terms, with O_s and A_s given on a volumetric basis. The Ethyl model comprises a total of six empirical parameters (a_1−a_6) for computing the RON and MON values of each blend. For blends outside the dataset used to estimate the

DOI: 10.1021/acs.iecr.6b01566
Ind. Eng. Chem. Res. 2016, 55, 7782−7800

parameters, the standard deviations of the prediction errors were found to be 0.92 for RON and 0.61 for MON. Other accurate methods comprise a higher number of parameters and require a larger experimental cost to update the blending model by least-squares analysis. Accounting for its accuracy and simplicity, the nonlinear Ethyl model is one of the methods that has been adopted to evaluate the RON and MON properties in the proposed MINLP formulation. The Ethyl model equations are given in Section 5.2.

At approximately the same time as the Ethyl method, simple nonlinear correlations for predicting the octane numbers of gasoline blends were also proposed by Stewart.[20] Both models attributed most of the blending nonlinearity to the olefin content of the feedstocks. In addition to the octane numbers and the olefin content of the feedstocks, the Stewart's correlations also include the olefin content of the blend (O_{blend}), which is computed as the volumetric average of olefin contents of the selected gasoline components:

$$O_{\text{blend}} = \sum_{s \in \mathbf{S}} w_s O_s$$

Besides, V_s is the volume of component s in the blend. The Stewart model includes four parameters that were determined by using least-squares analysis on 102 blends. The proposed MINLP formulation was also tested using the Stewart correlations (eqs 3−5). Values of the parameters $\overline{a}$ and $\overline{c}$ for estimating $\text{RON}_{\text{blend}}$ and $\text{MON}_{\text{blend}}$ are (0.0414, 0.01994) for $\text{RON}_{\text{blend}}$ and (0.130, 0.0970) for $\text{MON}_{\text{blend}}$.

$$\text{RON}_{\text{blend}} = \frac{\sum_{s \in \mathbf{S}} V_s D_s [\text{RON}_s + \overline{c}(O_s - O_{\text{blend}})]}{\sum_{s \in \mathbf{S}} V_s D_s} \tag{3}$$

$$\text{MON}_{\text{blend}} = \frac{\sum_{s \in \mathbf{S}} V_s D_s [\text{MON}_s + \overline{c}(O_s - O_{\text{blend}})]}{\sum_{s \in \mathbf{S}} V_s D_s} \tag{4}$$

$$D_s = \frac{\overline{a}(O_s - O_{\text{blend}})}{1 - \exp[\overline{a}(O_s - O_{\text{blend}})]} \tag{5}$$

Besides the octane numbers, the other critical property that blends nonlinearly is the Reid vapor pressure (RVP), which is defined by the ASTM Standard D-323-56 method. The RVP gives an indication of the volatility of a gasoline blend. When determining the optimal blend recipe, the maximum RVP limit restricts the amount of *n*-butane (a relatively inexpensive source of octane rating) that can be added to the gasoline blend.[3] The maximum RVP is especially important under limited amounts of the preferred gasoline components, such as the fluidized catalytic cracking (FCC) gasoline, the reformate stream (REF), and the light straight run (LSR) naphtha. In those cases, the optimal blend recipe has a tendency to include increasing amounts of *n*-butane until reaching the RVP upper bound. The blending index method is the simplest empirical nonlinear model developed by the Chevron Research Company to estimate the RVP of gasoline blends.[3] It is given by eq 6, which only requires knowing the RVP values of the pure components and the blend composition on a volumetric basis.

$$\text{RVP}_{\text{blend}}^{1.25} = \sum_{s \in \mathbf{S}} w_s \text{RVP}_s^{1.25} \tag{6}$$

Other approaches for predicting the RVP of gasoline blends include the nonlinear correlations of Stewart[21] and the interaction method of Morris,[22] which can also be applied to RVP blending by replacing octane numbers by RVPs.

4. PROBLEM STATEMENT

The gasoline recipe and blend scheduling problem can be defined as follows.

Given: (a) a set of dedicated tanks for gasoline components, together with their maximum capacities (scap_s) and initial inventories (iis_s) for each one; (b) a set of blend headers $b \in \mathbf{B}$ working in parallel, their processing rate limits $[\text{rb}_{b,p}^{\min}, \text{rb}_{b,p}^{\max}]$ and the final products $p \in \mathbf{P}$ that each one can process; (c) a set of gasoline components $s \in \mathbf{S}$, and their critical properties ($\text{spr}_{g,s}$) and unit costs (scost_s); (d) a set of demanded gasoline grades $p \in \mathbf{P}$ and their quality specifications given by the allowable range $[\text{ppr}_{g,p}^{\min}, \text{ppr}_{g,p}^{\max}]$ for the critical properties $g \in \mathbf{G}$, and the limiting proportions $[\text{vc}_{s,p}^{\min}, \text{vc}_{s,p}^{\max}]$ of the gasoline components in the finished products; (e) a set of production runs $i \in \mathbf{I}$ performing in the blend headers with a minimum length $\text{lb}_{b,p}^{\min}$; (f) a set of customer orders $r \in \mathbf{R}$, the requested product, the order size q_r, the delivery time window $[\text{atw}_r, \text{btw}_r]$ and the order delivery rate rdr_r for each one; (g) a set of multipurpose product tanks $j \in \mathbf{J}$, the tank capacity (pcap_j), the subset of final products that can store ($\mathbf{P}_j$), the current stored product, the initial inventory ($\text{iij}_{p,j}$) and the maximum delivery rate (pdr_j); (h) a time horizon composed of several time periods $k \in \mathbf{K}$ with different constant component feed rates $\text{svr}_{s,k}$ over each period.

Determine: (1) the allocation of production runs to blenders; (2) the gasoline grade, the selected recipe and the amount yielded by each production run; (3) the short-term schedule of blending operations in every blender; (4) the sequence of gasoline grades that are stored in each product tank; (5) the allocation of production runs to product tanks; (6) the schedule of delivery operations from each product tank; and (7) the inventory profiles in component and product tanks over the time horizon, in such a way that all customer orders are fully satisfied while minimizing quality giveaway, off-spec products, total component cost, penalties on tardy orders, and changeover costs in blenders and product tanks.

In this work, the management of the gasoline blending unit also accounts for a series of operational rules for blenders and tanks. Such important rules are as follows:

(i) every blender can process several products over the time horizon, but one after another;
(ii) after starting the processing of a gasoline product, a blender should operate for some minimum time before stopping or switching to another product;
(iii) a component tank may feed multiple blenders at the same time;
(iv) a component tank can receive flows from upstream processes and feed blenders at the same time;
(v) a blender can, at most, feed a single product tank at any time instant;
(vi) a product tank cannot receive some final product from a blender and simultaneously deliver a customer order;
(vii) a customer order can be satisfied by delivering the requested product from different product tanks; and
(viii) a product tank can deliver several orders at the same time.

In addition to the operational rules, a series of assumptions already proposed in previous works have been used to model the problem. They can be summarized as follows:

DOI: 10.1021/acs.iecr.6b01566
Ind. Eng. Chem. Res. 2016, 55, 7782−7800

(a) every blending run should occur within a single time slot;
(b) mixing in every blender is perfect;
(c) the product changeover time and cost in blenders are sequence-dependent;
(d) during a time slot, every product tank can, at most, receive the production from a single blender;
(e) the product changeover time in product tanks is negligible;
(f) each order involves a single product and must be delivered within the time horizon;
(g) loading and delivery operations in product tanks should occur in different time slots;
(h) delivery of a customer order from a product tank should be greater than a threshold value, and it must occur within a single time slot and begin at the starting time of the assigned slot.

5. THE MINLP MATHEMATICAL MODEL

The proposed MINLP formulation is based on the use of ordered sets of production runs ($\mathbf{I}$) and floating time slots ($\mathbf{T}$) with variable length. The elements of the set $\mathbf{I}$ are chronologically ordered with the production run $(i + 1)$ never beginning before starting run i. Similarly, the time slot $t \in \mathbf{T}$ starts at the completion time of slot $(t - 1)$. Floating time slots $t \in \boldsymbol{T}$ and production campaigns $i \in \boldsymbol{I}$ are not preassigned 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 period to another during the solution procedure. Therefore, unique sets of time slots and productions runs are defined, and the assignment of them to time periods is optimally made by the model.

5.1. Model Variables. Model decision variables can be gathered according to their purpose. They are defined for sequencing operations in the blenders, and allocating blending runs and order deliveries to product tanks and time slots. The arrangement of blending operations in the blend headers is handled through three different sets of binary variables: $WI_{i,k}$, assigning production runs to time periods; $WB_{i,b}$, allocating production runs to blenders $b \in \mathbf{B}$; and $YB_{i,p}$, selecting the final product $p \in \mathbf{P}$ yielded by each production run. The second group of 0−1 variables ($XIJ_{i,j,t}$, $XPJ_{p,j,t}$) is defined to allocate production runs and final products to product tanks and time slots. In combination with $YB_{i,p}$, the assignment variables $XIJ_{i,j,t}$ and $XPJ_{p,j,t}$ select the destination for the volume of product generated by run i during the time slot t. The third group involves the 0−1 variables $XRJ_{r,j,t}$ and $XDJ_{p,j,t}$ that assign customer orders to product tanks and time slots.

The continuous variables associated with every production run i include $QB_{i,p}$ representing the volume 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. 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 $s \in \mathbf{S}$ 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 $j \in \mathbf{J}_p$ at the end of slot t, $QPJ_{i,p,j,t}$ denoting the amount of product p 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 time slot t. It is assumed that an order delivery assigned to time slot t always begins at its initial time ST_t. Relative to an order delivery r, there are two continuous variables: $UR_{r,j,t}$, representing the amount of product delivered from tank j for order r within the time slot t, and $CR_{r,j,t}$, denoting the final time of that delivery.

5.2. Model Constraints. The gasoline blend optimization problem comprises four types of constraints related to (i) the scheduling of production runs in the blend headers, (ii) the fulfillment of quality specifications and demands of the final products, (iii) the tracking of inventory levels in component and product tanks to avoid overloading and running-out conditions, and (iv) the scheduling of nonsimultaneous receipt and delivery operations at multipurpose product tanks. Most of the constraints are linear. Nonlinearities just appear in equations of type (ii) and (iii). Nonlinear correlations are used to better estimate key gasoline blend properties such as the RON, the MON, and RVP. When multiple blenders are operated, nonlinear terms are to be included in constraints of type (iii) to exactly monitor the inventory level in component tanks. In some cases, the same recipe is to be adopted in all production campaigns of a final product. Additional nonlinear equations are necessary to guarantee that condition.

5.2.1. Production Runs Performed in Blenders. 5.2.1.1. A Production Run i Should, At Most, Be Performed within a Single Time Period k. According to eq 7, a production run should, at most, be performed within a single time period. Therefore, $\sum_{k\in\mathbf{K}} WI_{i,k} = 0$ characterizes a fictitious run i that was never performed:

$$\sum_{k\in\mathbf{K}} WI_{i,k} \leq 1 \qquad \forall\, i \in \mathbf{I} \tag{7}$$

5.2.1.2. A Production Run i Can, At Most, Be Assigned to a Single Blender. Using eq 8, a production run should, at most, be assigned to a single blender:

$$\sum_{b\in\mathbf{B}} WB_{i,b} = \sum_{k\in\mathbf{K}} WI_{i,k} \qquad \forall\, i \in \mathbf{I} \tag{8}$$

5.2.1.3. A Production Run Can, At Most, Yield a Single Product. Using eq 9, a production run yields, at most, a single final product:

$$\sum_{p\in\mathbf{P}} YB_{i,p} = \sum_{b\in\mathbf{B}} WB_{i,b} \qquad \forall\, i \in \mathbf{I} \tag{9}$$

5.2.1.4. Definition of the Continuous Variable $WBP_{i,b,p}$. The continuous variable $WBP_{i,b,p} \in [0,1]$ identifies the final product p yielded by production run i in blender b. Its value is determined by eqs 10−12.

$$WBP_{i,b,p} \geq WB_{i,b} + YB_{i,p} - 1 \qquad \forall\, i \in \mathbf{I},\ p \in \mathbf{P},\ b \in \mathbf{B}_p \tag{10}$$

$$\sum_{b\in\mathbf{B}_p} WBP_{i,b,p} \leq YB_{i,p} \qquad \forall\, i \in \mathbf{I},\ p \in \mathbf{P} \tag{11}$$

$$\sum_{p\in\mathbf{P}_b} WBP_{i,b,p} \leq WB_{i,b} \qquad \forall\, i \in \mathbf{I},\ b \in \mathbf{B} \tag{12}$$

DOI: 10.1021/acs.iecr.6b01566
Ind. Eng. Chem. Res. 2016, 55, 7782−7800

5.2.1.5. Length of a Production Run. Using eq 13, the length of a nonfictitious production run yielding product p in blender b should have a value within the range $[\mathrm{lb}_{b,p}^{\min}, \mathrm{lb}_{b,p}^{\max}]$:

$$\mathrm{lb}_{b,p}^{\min}\mathrm{WBP}_{i,b,p} \le \mathrm{LB}_{i,b,p} \le \mathrm{lb}_{b,p}^{\max}\mathrm{WBP}_{i,b,p} \qquad \forall\, i \in \mathbf{I},\, p \in \mathbf{P},\, b \in \mathbf{B}_p \tag{13}$$

5.2.1.6. Volume of Final Product Yielded by a Production Run. The continuous variable $\mathrm{QB}_{i,p}$ stands for the amount of final product p yielded by run i, and its value is given by eq 14:

$$\sum_{b\in\mathbf{B}_p} \mathrm{rb}_{b,p}^{\min}\mathrm{LB}_{i,b,p} \le \mathrm{QB}_{i,p} \le \sum_{b\in\mathbf{B}_p} \mathrm{rb}_{b,p}^{\max}\mathrm{LB}_{i,b,p} \qquad \forall\, i \in \mathbf{I},\, p \in \mathbf{P} \tag{14}$$

5.2.1.7. Ordered Execution of Production Runs in Blenders. Using eq 15, a generic run i can be performed only if the preceding run $(i-1)$ has already been assigned to a blend header, i.e., $\sum_{b\in\mathbf{B}} \mathrm{WB}_{(i-1),b} = 1$. Therefore, the last elements of $\mathbf{I}$ are reserved for fictitious runs.

$$\sum_{b\in\mathbf{B}} \mathrm{WB}_{i,b} \le \sum_{b\in\mathbf{B}} \mathrm{WB}_{(i-1),b} \qquad \forall\, (i-1),\, i \in \mathbf{I},\, b \in \mathbf{B} \tag{15}$$

5.2.1.8. Sequencing Production Runs in Each Blender. Using eq 16a, 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., $(\mathrm{llim}_k, \mathrm{ulim}_k)$. The starting and completion times of production run i then are determined by the set of equations 16a−16d. The parameter $\tau_{b,p,p'}$ is the sequence-dependent changeover time in blender b.

$$\mathrm{CB}_i \le \mathrm{SB}_{i'} - \tau_{b,p,p'} + h(2 - \mathrm{WBP}_{i,b,p} - \mathrm{WBP}_{i',b,p'}) \qquad \forall\, i, i' \in \mathbf{I}(i' > i),\, p, p' \in P,\, b \in \mathbf{B}_p \cap \mathbf{B}_{p'} \tag{16a}$$

$$\mathrm{CB}_i \le \sum_{k\in\mathbf{K}} \mathrm{ulim}_k \mathrm{WI}_{i,k} \qquad \forall\, i \in \mathbf{I} \tag{16b}$$

$$\mathrm{SB}_i \ge \sum_{k\in\mathbf{K}} \mathrm{llim}_k \mathrm{WI}_{i,k} \qquad \forall\, i \in \mathbf{I} \tag{16c}$$

$$\mathrm{CB}_i = \mathrm{SB}_i + \sum_{p\in\mathbf{P}} \sum_{b\in\mathbf{B}_p} \mathrm{LB}_{i,b,p} \qquad \forall\, i \in \mathbf{I} \tag{16d}$$

To facilitate the tracking of component and final product inventories, a production run $(i+1)$ can never begin before starting the preceding run i, regardless of the blenders assigned to both runs. In addition, a production run i can never finish after completing run $(i+1)$. Both conditions are redundant if runs i and $(i+1)$ are assigned to the same blender.

$$\mathrm{SB}_i \le \mathrm{SB}_{i+1} \qquad \forall\, i \in \mathbf{I} \tag{17a}$$

$$\mathrm{CB}_i \le \mathrm{CB}_{i+1} \qquad \forall\, i \in \mathbf{I} \tag{17b}$$

If the changeover time is not sequence-dependent, eq 16a can be replaced by the simpler eq 16a′:

$$\mathrm{CB}_i \le \mathrm{SB}_{i'} - \tau_b + h(2 - \mathrm{WB}_{i,b} - \mathrm{WB}_{i',b}) \qquad \forall\, i, i' \in \mathbf{I}(i' > i),\, b \in \mathbf{B} \tag{16a′}$$

5.2.1.9. Transition Costs in Blenders. The continuous variable $\mathrm{TRB}_{i,b}$, which is given by eqs 18a and 18b, denotes the cumulative transition cost at blender $b \in \mathbf{B}$ after performing production run i. In eq 18a, the parameter $\mathrm{ctrb}_{b,p,p'}$ denotes the sequence-dependent transition cost in blender b and M_{B} is a relatively large number. As stated by eq 18c, the maximum value of $\mathrm{TRB}_{i,b}$ for all $i \in \mathbf{I}$ provides a lower bound for the total transition cost (TTRB_b) in blender b.

$$\mathrm{TRB}_{i',b} \ge \mathrm{TRB}_{i,b} + \mathrm{ctrb}_{b,p,p'} - M_{\mathrm{B}}(2 - \mathrm{WBP}_{i,b,p} - \mathrm{WBP}_{i',b,p'}) \qquad \forall\, (i, i') \in \mathbf{I}(i < i'),\, b \in \mathbf{B}_p \cap \mathbf{B}_{p'},\, p, p' \in \mathbf{P}(p \ne p') \tag{18a}$$

$$\mathrm{TRB}_{i,b} \le M_{\mathrm{B}}\mathrm{WB}_{i,b} \qquad \forall\, i \in \mathbf{I},\, b \in \mathbf{B} \tag{18b}$$

$$\mathrm{TTRB}_b \ge \mathrm{TRB}_{i,b} \qquad \forall\, i \in \mathbf{I},\, b \in \mathbf{B} \tag{18c}$$

If the transition cost has a constant value (ctrb), a lower bound on the total transition cost in blenders is given by eq 19, which assumes a single run per product.

$$\sum_{b\in\mathbf{B}} \mathrm{TTRB}_b \ge \mathrm{ctrb}[\mathrm{card}(\mathbf{P}) - \mathrm{card}(B)] \tag{19}$$

5.2.2. Overall Fulfillment of Final Product Demands. The parameter dem_p denotes the total demand of final product p to be satisfied during the scheduling horizon, given by eq 20. Because part of that demand can be fulfilled by making use of the initial inventory of product p, then eq 21 states that the production of product p should be large enough to cover its net demand over the time horizon.

$$\mathrm{dem}_p = \sum_{r\in\mathbf{R}_p} q_r \qquad \forall\, p \in \mathbf{P} \tag{20}$$

$$\sum_{i\in\mathbf{I}} \mathrm{QB}_{i,p} \ge \mathrm{dem}_p - \sum_{j\in\mathbf{J}_p} \mathrm{iij}_{p,j} \qquad \forall\, p \in \mathbf{P} \tag{21}$$

5.2.3. Monitoring the Inventory of Component s at the Start/End Times of a Production Run. 5.2.3.1. Total Amount of Gasoline Components Assigned to Each Production Run. The continuous variable $\mathrm{QS}_{s,i,p}$ represents the amount of component $s \in \mathbf{S}$ assigned to run i producing product p. According to eq 22, the total amount of gasoline components assigned to run i should be equal to the production of the final product yielded by run i.

$$\sum_{s\in S} \mathrm{QS}_{s,i,p} = \mathrm{QB}_{i,p} \qquad \forall\, i \in \mathbf{I},\, p \in \mathbf{P} \tag{22}$$

5.2.3.2. Amount of Each Individual Gasoline Component Assigned to a Production Run. Equation 23 provides the amount of component s allocated to run i yielding product p, i.e., $\mathrm{QS}_{s,i,p}$. Then, gasoline recipes are strictly determined by continuous variables. There is no 0−1 variable performing the selection of the blend components.

$$\mathrm{vc}_{s,p}^{\min}\mathrm{QB}_{i,p} \le \mathrm{QS}_{s,i,p} \le \mathrm{vc}_{s,p}^{\max}\mathrm{QB}_{i,p} \qquad \forall\, s \in \mathbf{S},\, i \in \mathbf{I},\, p \in \mathbf{P} \tag{23}$$

5.2.3.3. Total Amount of Component s Consumed by Blending Runs up to Time SB_i. By the definition of the set $\mathbf{I}$, those production runs i' starting before run i and belonging to the subset $\mathrm{INP}_i = \{i' \in \mathbf{I} | i' < i - \mathrm{card}(\mathbf{B}) + 1\}$ are completed before beginning run i. They cannot be executed in parallel with run i. Therefore, the total amount of component s assigned to those production campaigns $i' \in \mathbf{INP}_i$ was already consumed at time SB_i. Instead, the preceding runs i' belonging to the subset $\mathbf{IP}_i = \{i' \in \mathbf{I} | i' > i - \mathrm{card}(\mathbf{B})\}$ can be executed in

DOI: 10.1021/acs.iecr.6b01566
Ind. Eng. Chem. Res. 2016, 55, 7782−7800

parallel with run i if more than one blender is operated, i.e., card(**B**) > 1. Then, they can be totally or partially overlapped with run i. Consequently, the amounts of components allocated to runs $i' \in \mathbf{IP}_i$ may be partially consumed (or not consumed at all) at time SB_i. Assuming a constant production rate over the entire run i, the consumption rate of component s during campaign i ($RS_{i,s}$) is given by the nonlinear constraint given as eq 24. In turn, eq 25 provides a lower bound for the continuous variable $QSP_{s,i',i}$ representing the amount of component s consumed by run $i' \in \mathbf{IP}_i$ after time SB_i. The difference ($CB_{i'} - SB_i$) is the overlap in time between runs i' and i. If the continuous variable $USP_{s,i}$ denotes the total amount of component s consumed up to time SB_i, then its value given by eq 26 can be obtained by simply subtracting ($\sum_{i' \in \mathbf{IP}_i} QSP_{s,i',i}$) from the total amount of component s assigned to the preceding runs $i' < i$. When a single blender is available, $\mathbf{IP}_i$ is an empty set.

$$\sum_{p \in \mathbf{P}} QS_{s,i,p} = RS_{s,i}\left(\sum_{p \in \mathbf{P}} \sum_{b \in \mathbf{B}_p} LB_{i,b,p}\right) \qquad \forall\, i \in \mathbf{I},\ s \in \mathbf{S} \tag{24}$$

$$QSP_{s,i',i} \geq RS_{s,i'}(CB_{i'} - SB_i) \qquad \forall\, i \in \mathbf{I},\ i' \in \mathbf{IP_i},\ s \in \mathbf{S} \tag{25}$$

$$USP_{s,i} = \sum_{\substack{i' \in \mathbf{I} \\ i' < i}} \sum_{p \in \mathbf{P}} QS_{s,i',p} - \sum_{i' \in \mathbf{IP_i}} QSP_{s,i',i} \qquad \forall\, i \in \mathbf{I},\ s \in \mathbf{S} \tag{26}$$

5.2.3.4. Total Amount of Component s Consumed by the Blending Runs up to Time CB_i. Because of the definition of the set **I**, all the production runs i' that precede run i ($i' < i$) have already finished at the completion time of run i. The amounts of gasoline components assigned to them then have been totally consumed at time CB_i. However, succeeding runs belonging to the subset $IS_i = \{i' \in I |\, i < i' < i + \text{card}(\mathbf{B})\}$ can be performed in parallel with run i if multiple blenders are available. They can be partially or completely overlapped in time with run i. As a result, the amount of component s allocated to runs $i' \in IS_i$ can be either partially or totally consumed at time CB_i. Equation 27 provides a lower bound for the value of the continuous variable $QSS_{s,i,i'}$, denoting the amount of component s consumed by run $i' \in IS_i$ up to time CB_i. The total amount of component s consumed up to time CB_i, given by the continuous variable $USS_{s,i}$, is obtained through eq 28 by summing up those amounts consumed by production runs $i' \leq i$ and by all campaigns $i' \in IS_i$.

$$QSS_{s,i,i'} \geq RS_{s,i'}(CB_i - SB_{i'}) \qquad \forall\, i \in \mathbf{I},\ i' \in IS_i,\ s \in \mathbf{S} \tag{27}$$

$$USS_{s,i} = \sum_{\substack{i' \in \mathbf{I} \\ i' \leq i}} \sum_{p \in \mathbf{P}} QS_{s,i',p} + \sum_{i' \in \mathbf{IS_i}} QSS_{s,i,i'} \qquad \forall\, i \in \mathbf{I},\ s \in \mathbf{S} \tag{28}$$

5.2.3.5. Inventories of Component s at the Start/End Times of a Production Run. The inventory level of every component 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, respectively. Equation 29a 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 ($scap_s$) to avoid overloading. Assuming that production run i is assigned to time period k (i.e., $WI_{ik} = 1$), then eq 29a accounts for (i) the initial inventory of component s; (ii) the amount of component s loaded into the assigned tank over the periods $k' < k$; (iii) the amount of component s loaded during the period k up to the start time of run i; and (iv) the cumulative amount of component s ($USP_{s,i}$) delivered from the assigned tank to the blenders up to time SB_i. In eq 29a, 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 29b and 29c. Evidently, $LKS_{i,k}$ is equal to zero if run i is not assigned to period k.

$$SINI_{s,i} = iis_s + \sum_{k \in \mathbf{K}} \left[\left(\sum_{\substack{k' \in \mathbf{K} \\ k' < k}} svr_{s,k'} hk_{k'}\right) WI_{i,k} + svr_{s,k} LKS_{i,k}\right] - USP_{s,i} \leq scap_s \qquad \forall\, s \in \mathbf{S},\ i \in \mathbf{I} \tag{29a}$$

$$LKS_{i,k} \geq SB_i - llim_k + hk_k(1 - WI_{i,k}) \qquad \forall\, i \in \mathbf{I},\ k \in \mathbf{K} \tag{29b}$$

$$LKS_{i,k} \leq hk_k WI_{i,k} \qquad \forall\, i \in \mathbf{I},\ k \in \mathbf{K} \tag{29c}$$

Similarly, eq 30a 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 the assigned period k and the completion of run i and its value is given by eqs 30b and 30c. To avoid running-out conditions for component s, the condition $SINC_{s,i} \geq 0$ is to be satisfied at the completion of any run i.

$$SINC_{s,i} = iis_s + \sum_{k \in \mathbf{K}} \left[\left(\sum_{\substack{k' \in \mathbf{K} \\ k' < k}} svr_{s,k'} hk_{k'}\right) WI_{i,k} + svr_{s,k} LKF_{i,k}\right] - USS_{s,i} \qquad \forall\, s \in \mathbf{S},\ i \in \mathbf{I} \tag{30a}$$

$$LKF_{i,k} \leq CB_i - llim_k WI_{i,k} \qquad \forall\, i \in \mathbf{I},\ k \in \mathbf{K} \tag{30b}$$

$$LKF_{i,k} \leq hk_k WI_{i,k} \qquad \forall\, i \in \mathbf{I},\ k \in \mathbf{K} \tag{30c}$$

Besides, the inventory of component s at the horizon end ($SINF_s$) is determined by eq 30d. $SINF_s$ must never exceed the capacity of the assigned tank ($scap_s$).

$$SINF_s = iis_s + \sum_{k \in \mathbf{K}} svr_{k,s} hk_k - \sum_{i \in \mathbf{I}} \sum_{p \in \mathbf{P}} QS_{s,i,p} \leq scap_s \qquad \forall\, s \in \mathbf{S} \tag{30d}$$

5.2.4. Fulfillment of the Gasoline Quality Specifications. 5.2.4.1. Controlling the Value of Every Key Property g in the Final Product p. Let us first consider the gasoline properties that blend linearly. Equations 31a and 31b seek to make on-spec final products within the desired limits $\{ppr_{g,p}^{min}, ppr_{g,p}^{max}\}$ of any property g that blends linearly on either volume or weight base. In eq 31b, ρ_s represents the density of component s. The set G_L^V comprises the properties that blend linearly on a volume basis, while G_L^W is the set of properties linearly blended on a weight basis.

$$ppr_{g,p}^{min} QB_{i,p} \leq \sum_{s \in \mathbf{S}} spr_{g.s} QS_{s,i,p} \leq ppr_{g,p}^{max} QB_{i,p} \qquad \forall\, i \in \mathbf{I},\ p \in \mathbf{P},\ g \in \mathbf{G_L^V} \tag{31a}$$

DOI: 10.1021/acs.iecr.6b01566
Ind. Eng. Chem. Res. 2016, 55, 7782−7800

$$\mathrm{ppr}_{g,p}^{\min}\Big(\sum_{s\in\mathbf{S}}\rho_s \mathrm{QS}_{s,i,p}\Big) \le \sum_{s\in\mathbf{S}}\mathrm{spr}_{g,s}\rho_s \mathrm{QS}_{s,i,p} \le \mathrm{ppr}_{g,p}^{\max}\Big(\sum_{s\in\mathbf{S}}\rho_s \mathrm{QS}_{s,i,p}\Big) \qquad \forall\, i\in\mathbf{I},\, p\in\mathbf{P},\, g\in\mathbf{G}_{\mathbf{L}}^{\mathbf{W}} \tag{31b}$$

5.2.4.2. Model Equations for Predicting and Monitoring Nonlinear Properties of Gasoline Blends. As explained previously, the research and motor octane numbers (RON, MON) are not accurately estimated by the volumetric average of the component octane numbers because they blend in nonideal fashion. Nonlinear models are needed to improve prediction of octane numbers. Let us introduce the continuous variables $\mathrm{PRON}_{i,p}$ and $\mathrm{PMON}_{i,p}$ to represent the values of the RON and MON properties of the final product p yielded by the blending run i. In turn, $\mathrm{VFR}_{s,i,p}$ represents the volumetric fraction of gasoline component s in blend p provided by run i. Moreover, the parameters sron_s and smon_s denote the research and motor octane numbers of component s, while O_s and A_s represent its olefin and aromatic content. Using the Ethyl RT-70 method introduced by Healy et al.,[19] eqs 32−35 provide the values of $\mathrm{PRON}_{i,p}$ and $\mathrm{PMON}_{i,p}$. In eqs 32−35, nonlinear terms that are functions of the sensitivity $(\mathrm{SRON}_s - \mathrm{SMON}_s)$, and the olefin and the aromatic content of the components are added to the volumetric average of the RON/MON values of the gasoline components. The continuous variable $\mathrm{VFR}_{s,i,p}$ represents the volume fraction of component s in blend p produced by run i.

$$\mathrm{PRON}_{i,p} = \bar{r}_{i,p} + a_1(\overline{\mathrm{rs}}_{i,p} - \bar{r}_{i,p}\bar{s}_{i,p}) + a_2(\overline{O^2}_{i,p} - \bar{O}_{i,p}{}^2) + a_3(\overline{A^2}_{i,p} - \bar{A}_{i,p}{}^2) \qquad \forall\, i\in\mathbf{I},\, p\in\mathbf{P} \tag{32}$$

$$\mathrm{PMON}_{i,p} = \bar{m}_{i,p} + a_4(\overline{\mathrm{ms}}_{i,p} - \bar{m}_{i,p}\bar{s}_{i,p}) + a_5(\overline{O^2}_{i,p} - \bar{O}_{i,p}{}^2) + a_6\left(\frac{\overline{A^2}_{i,p} - \bar{A}_{i,p}{}^2}{100}\right)^2 \qquad \forall\, i\in\mathbf{I},\, p\in\mathbf{P} \tag{33}$$

$$\mathrm{QS}_{s,i,p} = \mathrm{VFR}_{s,i,p}\Big(\sum_{s'\in\mathbf{S}}\mathrm{QS}_{s',i,p}\Big) \qquad \forall\, s\in\mathbf{S},\, i\in\mathbf{I},\, p\in\mathbf{P} \tag{34}$$

$$\bar{r}_{i,p} = \sum_{s\in\mathbf{S}}\mathrm{VFR}_{s,i,p}\mathrm{SRON}_s \qquad \forall\, i\in\mathbf{I},\, p\in\mathbf{P} \tag{35a}$$

$$\bar{m}_{i,p} = \sum_{s\in\mathbf{S}}\mathrm{VFR}_{s,i,p}\mathrm{SMON}_s \qquad \forall\, i\in\mathbf{I},\, p\in\mathbf{P} \tag{35b}$$

$$\bar{s}_{i,p} = \sum_{s\in\mathbf{S}}\mathrm{VFR}_{s,i,p}(\mathrm{SRON}_s - \mathrm{SMON}_s) \qquad \forall\, i\in\mathbf{I},\, p\in\mathbf{P} \tag{35c}$$

$$\overline{\mathrm{rs}}_{i,p} = \sum_{s\in\mathbf{S}}\mathrm{VFR}_{s,i,p}(\mathrm{SRON}_s - \mathrm{SMON}_s)\mathrm{SRON}_s \qquad \forall\, i\in\mathbf{I},\, p\in\mathbf{P} \tag{35d}$$

$$\overline{\mathrm{ms}}_{i,p} = \sum_{s\in\mathbf{S}}\mathrm{VFR}_{s,i,p}(\mathrm{SRON}_s - \mathrm{SMON}_s)\mathrm{SMON}_s \qquad \forall\, i\in\mathbf{I},\, p\in\mathbf{P} \tag{35e}$$

$$\bar{O}_{i,p} = \sum_{s\in\mathbf{S}}\mathrm{VFR}_{s,i,p}O_s \qquad \forall\, i\in\mathbf{I},\, p\in\mathbf{P} \tag{35f}$$

$$\bar{O}_{i,p}{}^2 = \sum_{s\in\mathbf{S}}\mathrm{VFR}_{s,i,p}O_s^2 \qquad \forall\, i\in\mathbf{I},\, p\in\mathbf{P} \tag{35g}$$

$$\bar{A}_{i,p} = \sum_{s\in\mathbf{S}}\mathrm{VFR}_{s,i,p}A_s \qquad \forall\, i\in\mathbf{I},\, p\in\mathbf{P} \tag{35h}$$

$$\bar{A}_{i,p}{}^2 = \sum_{s\in\mathbf{S}}\mathrm{VFR}_{s,i,p}A_s{}^2 \qquad \forall\, i\in\mathbf{I},\, p\in\mathbf{P} \tag{35i}$$

Values for the model coefficients a_1−a_6 reported by Healy et al.[19] are $a_1 = 0.03224$, $a_2 = 0.00101$, $a_3 = 0$, $a_4 = 0.04450$, $a_5 = 0.00081$, and $a_6 = -0.0645$. In this work, we use the Chevron blending index method to predict the Reid vapor pressure (RVP).[3] The RVP value for any product p yielded by blending run i ($\mathrm{PRVP}_{i,p}$) is estimated by a nonlinear mixing rule given by eq 36. The parameter srvp_s is the RVP of the gasoline component s.

$$\mathrm{PRVP}_{i,p}{}^{1.25} = \sum_{s\in\mathbf{S}}\mathrm{VFR}_{s,i,p}(\mathrm{srvp})_s^{1.25} \qquad \forall\, i\in\mathbf{I},\, p\in\mathbf{P} \tag{36}$$

Note that if all the production runs for product p should use the same recipe given by $\mathrm{VFR}_{s,p}$, then eq 34 should be replaced by the nonlinear eq 37. In addition, $\mathrm{VFR}_{s,i,p}$ must be substituted by $\mathrm{VFR}_{s,p}$ in eqs 35a−35i.

$$\mathrm{QS}_{s,i,p} = \mathrm{VFR}_{s,p}\Big(\sum_{s'\in\mathbf{S}}\mathrm{QS}_{s',i,p}\Big) \qquad \forall\, s\in\mathbf{S},\, i\in\mathbf{I},\, p\in\mathbf{P} \tag{37}$$

To obtain on-spec gasoline blends, the values of the nonlinear properties $\mathrm{PRON}_{i,p}$, $\mathrm{PMON}_{i,p}$, and $\mathrm{PRVP}_{i,p}$ should satisfy the constraints described by eqs 38a−38c.

$$\mathrm{pron}_p^{\min}\mathrm{YB}_{i,p} \le \mathrm{PRON}_{i,p} \le \mathrm{pron}_p^{\max}\mathrm{YB}_{i,p} \qquad \forall\, i\in\mathbf{I},\, p\in\mathbf{P} \tag{38a}$$

$$\mathrm{pmon}_p^{\min}\mathrm{YB}_{i,p} \le \mathrm{PMON}_{i,p} \le \mathrm{pmon}_p^{\max}\mathrm{YB}_{i,p} \qquad \forall\, i\in\mathbf{I},\, p\in\mathbf{P} \tag{38b}$$

$$\mathrm{prvp}_p^{\min}\mathrm{YB}_{i,p} \le \mathrm{PRVP}_{i,p} \le \mathrm{prvp}_p^{\max}\mathrm{YB}_{i,p} \qquad \forall\, i\in\mathbf{I},\, p\in\mathbf{P} \tag{38c}$$

The intervals $(\mathrm{pron}_p^{\min}, \mathrm{pron}_p^{\max})$, $(\mathrm{pmon}_p^{\min}, \mathrm{pmon}_p^{\max})$, and $\mathrm{prvp}_p^{\min}, \mathrm{prvp}_p^{\max})$ stand for the specified allowable ranges of variables $\mathrm{PRON}_{i,p}$, $\mathrm{PMOM}_{i,p}$, and $\mathrm{PRVP}_{i,p}$, respectively.

5.2.5. Allocating the Blend Production to Final Product Tanks and Time Slots. 5.2.5.1. Assigning Production Runs to Storage Tanks. By eq 39, a production run should, at most, be allocated to a single product tank and performed within a single time slot.

$$\sum_{t\in\mathbf{T}}\sum_{j\in\mathbf{J}}\mathrm{XIJ}_{i,j,t} = \sum_{b\in\mathbf{B}}\mathrm{WB}_{i,b} \qquad \forall\, i\in\mathbf{I} \tag{39}$$

In turn, eq 40 states that a product tank can, at most, receive a single production run within every time slot.

$$\sum_{i\in\mathbf{I}}\mathrm{XIJ}_{i,j,t} \le 1 \qquad \forall\, j\in\mathbf{J},\, t\in\mathbf{T} \tag{40}$$

5.2.5.2. Assigning Storage Tanks to Final Products during Any Time Slot. Using eq 41, a storage tank should be allocated to a single product at every time slot.

DOI: 10.1021/acs.iecr.6b01566
Ind. Eng. Chem. Res. 2016, 55, 7782−7800

$$\sum_{p\in\mathbf{P}_j} \mathrm{XPJ}_{p,j,t} = 1 \qquad \forall\, t \in \mathbf{T},\, j \in \mathbf{J} \tag{41}$$

Moreover, a campaign of product p (i.e., $\mathrm{YB}_{i,p} = 1$) must be discharged into a storage tank assigned to that product. Assuming that run i yields product p, then the variable $\mathrm{XIJ}_{i,j,t}$ can be equal to one only if $\mathrm{XPJ}_{p,j,t} = 1$ by eq 42.

$$\mathrm{XIJ}_{i,j,t} + \mathrm{YB}_{i,p} \le 1 + \mathrm{XPJ}_{p,j,t} \qquad \forall\, i \in \mathbf{I},\, j \in \mathbf{J},\, p \in \mathbf{P}_j,\, t \in \mathbf{T} \tag{42}$$

5.2.5.3. Amount of Product Loaded into a Storage Tank during a Time Slot. The continuous variable $\mathrm{QPJ}_{i,p,j,t}$ stands for the amount of product p from run i discharged into storage tank $j \in J_p$ during the time slot t. As specified by eqs 43a and 43b, its value is zero if either production run i does not yield product p (i.e., $\mathrm{YB}_{i,p} = 0$) or run i is not assigned to tank j during the slot t (i.e., $\mathrm{XIJ}_{i,j,t} = 0$). Otherwise, it is equal to $\mathrm{QB}_{i,p}$ by eq 43c.

$$\mathrm{QPJ}_{i,p,j,t} \le \mathrm{dem}_p \mathrm{XIJ}_{i,j,t} \qquad \forall\, i \in \mathbf{I},\, j \in \mathbf{J},\, p \in \mathbf{P}_j,\, t \in \mathbf{T} \tag{43a}$$

$$\sum_{t\in\mathbf{T}} \sum_{j\in\mathbf{J_p}} \mathrm{QPJ}_{i,p,j,t} \le \mathrm{dem}_p \mathrm{YB}_{i,p} \qquad \forall\, p \in \mathbf{P},\, i \in \mathbf{I} \tag{43b}$$

$$\sum_{t\in\mathbf{T}} \sum_{j\in\mathbf{J_p}} \mathrm{QPJ}_{i,p,j,t} = \mathrm{QB}_{i,p} \qquad \forall\, p \in \mathbf{P},\, i \in \mathbf{I} \tag{43c}$$

5.2.5.4. Product Transition Cost in Storage Tanks. The positive variable $\mathrm{TRJ}_{p,j,t}$ 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 $\mathrm{TRJ}_{p,j,t}$ is given by eq 44a, where the parameter ctrj_j denotes the product changeover cost in tank j.

$$\mathrm{TRJ}_{p,j,t} \ge \mathrm{ctrj}_j\Big(\mathrm{XPJ}_{p,j,t} + \sum_{\substack{p'\in\mathbf{P}_j \\ p'\neq p}} \mathrm{XPJ}_{p',j,t+1} - 1\Big) \qquad \forall\, p \in \mathbf{P},\, j \in \mathbf{J}_p,\, t \in \mathbf{T} \tag{44a}$$

If sequence-dependent transition costs are to be handled, eq 44a is to be replaced by eq 44b.

$$\mathrm{TRJ}_{p,j,t} \ge \mathrm{ctrj}_{j,p,p'}\big(\mathrm{XPJ}_{p,j,t} + \mathrm{XPJ}_{p',j,t+1} - 1\big) \qquad \forall\, p, p' \in \mathbf{P}(p \neq p'),\, j \in \mathbf{J}_p \cap \mathbf{J}_{p'},\, t \in \mathbf{T} \tag{44b}$$

5.2.6. Sequencing Time Slots. 5.2.6.1. Ordered Set of Time Slots. It is said that $\mathbf{T}$ is an ordered set because time slot t must begin just after finishing the preceding slot $(t-1)$, as specified by eq 45.

$$\mathrm{ST}_t \ge \mathrm{CT}_{t-1} \qquad \forall\, (t-1),\, t \in \mathbf{T} \tag{45}$$

5.2.6.2. Initial and Final Times of Production Runs Assigned to Time Slot t. Using eqs 46a and 46b, a production run assigned to time slot t must be performed within the range $\{\mathrm{ST}_t, \mathrm{CT}_t\}$. The length of slot t is given by eq 47. Equation 48 states that the total length of the time slots should be equal to the horizon length h.

$$\mathrm{ST}_t \le \mathrm{SB}_i + h\Big(1 - \sum_{j\in\mathbf{J}} \mathrm{XIJ}_{i,j,t}\Big) \qquad \forall\, i \in \mathbf{I},\, t \in \mathbf{T} \tag{46a}$$

$$\mathrm{CB}_i \le \mathrm{CT}_t + h\Big(1 - \sum_{j\in\mathbf{J}} \mathrm{XIJ}_{i,j,t}\Big) \qquad \forall\, i \in \mathbf{I},\, t \in \mathbf{T} \tag{46b}$$

$$\mathrm{LT}_t = \mathrm{CT}_t - \mathrm{ST}_t \qquad \forall\, t \in \mathbf{T} \tag{47}$$

$$\sum_{t\in\mathbf{T}} \mathrm{LT}_t = h \tag{48}$$

5.2.7. Unloading Products from the Storage Tanks To Satisfy Customer Demands. 5.2.7.1. Nonsimultaneous Loading and Unloading Operations in Storage Tanks. Equation 49 imposes that simultaneous receipt and delivery operations in product tanks cannot be performed within the same time slot. The binary variable $\mathrm{XDJ}_{p,j,t}$ denotes the discharge of product p from product tank j during time slot t whenever $\mathrm{XDJ}_{p,j,t}$ is equal to 1.

$$\sum_{p\in\mathbf{P}_j} \mathrm{XDJ}_{p,j,t} + \sum_{i\in\mathbf{I}} \mathrm{XIJ}_{i,j,t} \le 1 \qquad \forall\, t \in \mathbf{T},\, j \in \mathbf{J} \tag{49}$$

5.2.7.2. Amount of Product Unloaded from a Storage Tank during a Time Slot. The continuous variable $\mathrm{UP}_{p,j,t}$ represents the amount of product p unloaded from product tank j during slot t to satisfy customer orders for product p. Its value should be equal to zero if (a) product tank j receives production from a blender during time slot t, (b) product tank j has not been assigned to product p during time slot t, or (c) no delivery of product p from tank j is planned during time slot t. Such conditions are expressed by eqs 50, 51, and 52, respectively. If none of the above conditions holds, the unloaded volume of p can never exceed the available inventory of p in tank j at the end of the preceding time slot $(t-1)$, as established by eq 53.

$$\mathrm{UP}_{p,j,t} \le \mathrm{pcap}_j \mathrm{XDJ}_{p,j,t} \qquad \forall\, j \in \mathbf{J},\, p \in \mathbf{P}_j,\, t \in \mathbf{T} \tag{50}$$

$$\mathrm{XDJ}_{p,j,t} \le \mathrm{XPJ}_{p,j,t} \qquad \forall\, j \in \mathbf{J},\, p \in \mathbf{P}_j,\, t \in \mathbf{T} \tag{51}$$

$$\mathrm{UP}_{p,j,t} \le \mathrm{pcap}_j \mathrm{XDJ}_{p,j,t} \qquad \forall\, j \in \mathbf{J},\, p \in \mathbf{P}_j,\, t \in \mathbf{T} \tag{52}$$

$$\mathrm{UP}_{p,j,t} \le \mathrm{PINV}_{p,j,t-1} \qquad \forall\, j \in \mathbf{J},\, p \in \mathbf{P}_j,\, t \in \mathbf{T} \tag{53}$$

5.2.7.3. Meeting the Total Demand of Product p. By eq 54, the total volume of product p unloaded from all the storage tanks over the entire scheduling horizon should be large enough to meet the requirements of all of the orders demanding product p.

$$\sum_{t\in\mathbf{T}} \sum_{j\in\mathbf{J}_p} \mathrm{UP}_{p,j,t} \ge \sum_{r\in\mathbf{R}_p} q_r \qquad \forall\, p \in \mathbf{P} \tag{54}$$

5.2.8. Controlling the Inventory of Final Products in Storage Tanks. 5.2.8.1. Inventory Level of Final Products in Storage Tanks along the Scheduling Horizon. The nonnegative variable $\mathrm{PINV}_{p,j,t}$ denotes the inventory of product p in tank $j \in \mathbf{J}_p$ at the end of time slot t. Its value, given by eq 55, accounts for (i) the initial inventory of product p in tank j ($\mathrm{iij}_{p,j}$), (ii) the total volume of product p from blend headers discharged into tank j up to time CT_t, and (iii) the amount of product p unloaded from tank $j \in \mathbf{J}_p$ to meet customer orders from the horizon start up to time CT_t. By eq 56, the value of $\mathrm{PINV}_{p,j,t}$ should never exceed the capacity of tank j.

DOI: 10.1021/acs.iecr.6b01566
Ind. Eng. Chem. Res. 2016, 55, 7782−7800

$$\mathrm{PINV}_{p,j,t} = \mathrm{iij}_{j,p} + \sum_{i\in\mathbf{I}} \sum_{\substack{t'\in\mathbf{T}\\ t'\le t}} \mathrm{QPJ}_{i,p,j,t'} - \sum_{\substack{t'\in\mathbf{T}\\ t'\le t}} \mathrm{UP}_{p,j,t'}$$

$$\forall\, j\in\mathbf{J},\ p\in\mathbf{P}_j,\ t\in\mathbf{T} \tag{55}$$

$$\mathrm{PINV}_{p,j,t} \le \mathrm{pcap}_j \mathrm{XPJ}_{p,j,t} \qquad \forall\, j\in\mathbf{J},\ p\in\mathbf{P}_j,\ t\in\mathbf{T} \tag{56}$$

5.2.8.2. Conditions for the Allocation of Tank $j \in \mathbf{J}_p$ to Product p during Time Slot t. Equation 57 specifies that a storage tank $j \in \mathbf{J}_p$ allocated to product p during time slot t cannot contain another product at the end of slot $(t-1)$. In other words, either tank j was already assigned to product p or it is empty at the end of time slot $(t-1)$.

$$\mathrm{PINV}_{p',j,(t-1)} \le \mathrm{pcap}_j(1 - \mathrm{XPJ}_{p,j,t})$$

$$\forall\, (t-1),\ t\in\mathbf{T},\ p,\ p'\in\mathbf{P}_j(p'\ne p),\ j\in\mathbf{J}_p\cap\mathbf{J}_{p'} \tag{57}$$

5.2.9. Fulfillment of Customer Orders. 5.2.9.1. Allocating Deliveries of Final Products from Storage Tanks to Customer Orders. By eq 58, no allocation of product p to customer orders requiring that product during time slot t can be made if there is no discharge of p from tank j, i.e., $\mathrm{XDJ}_{p,j,t} = 0$. The parameter nr_p represents the total number of customer orders for the final product p.

$$\sum_{r\in\mathbf{R}_p} \mathrm{XRJ}_{r,j,t} \le \mathrm{nr}_p \mathrm{XDJ}_{p,j,t} \qquad \forall\, j\in\mathbf{J},\ p\in\mathbf{P}_j,\ t\in\mathbf{T} \tag{58}$$

Moreover, eq 59 requires that the entire amount of product discharged from storage tank j should be entirely allocated to customer orders demanding that product. Equation 60 shows that 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 q_r.

$$\mathrm{UP}_{p,j,t} = \sum_{r\in\mathbf{R}_p} \mathrm{UR}_{r,j,t} \qquad \forall\, j\in\mathbf{J},\ p\in\mathbf{P}_j,\ t\in\mathbf{T} \tag{59}$$

$$\sum_{t\in\mathbf{T}} \sum_{j\in\mathbf{J}_r} \mathrm{UR}_{r,j,t} = q_r \qquad \forall\, r\in\mathbf{R} \tag{60}$$

5.2.9.2. Limiting the Amount of Product Delivered for a Customer Order. Equations 61a and 61b restrict the value of $\mathrm{UR}_{r,j,t}$ to the interval $(\mathrm{sr}_{\min}, q_r)$:

$$\mathrm{UR}_{r,j,t} \ge \mathrm{sr}_{\min} \mathrm{XRJ}_{r,j,t} \qquad \forall\, j\in\mathbf{J},\ r\in\mathbf{R},\ t\in\mathbf{T} \tag{61a}$$

$$\mathrm{UR}_{r,j,t} \le q_r \mathrm{XRJ}_{r,j,t} \qquad \forall\, j\in\mathbf{J},\ r\in\mathbf{R},\ t\in\mathbf{T} \tag{61b}$$

where $\mathrm{sr}_{\min}$ is the minimum amount coming from a product tank that can be assigned to a customer order. If the parameter rdr_r represents the delivery rate of order r demanding product p and $\mathrm{pdr}_{p,j}$ denotes the maximum delivery rate of product p from tank j, then the values of $\mathrm{UR}_{r,j,t}$ and $\mathrm{XRJ}_{r,j,t}$ are also limited by eqs 62a and 62b.

$$\mathrm{UR}_{r,j,t} \le \mathrm{rdr}_r(\mathrm{CT}_t - \mathrm{ST}_t) \qquad \forall\, p\in\mathbf{P},\ j\in\mathbf{J}_p,\ t\in\mathbf{T} \tag{62a}$$

$$\sum_{r\in\mathbf{R}_p} \mathrm{rdr}_r \mathrm{XRJ}_{r,j,t} \le \mathrm{pdr}_{p,j} \qquad \forall\, j\in\mathbf{J}_p,\ \mathbf{p}\in\mathbf{P},\ t\in\mathbf{T} \tag{62b}$$

5.2.9.3. Starting and Completion Times of Product Deliveries from Storage Tanks. The positive variable $\mathrm{CR}_{r,j,t}$ represents the completion time for the delivery of product from tank j to order r during time slot t. Assuming that every unloading operation always begins at the start of a time slot, then ST_t should satisfy the constraint described by eq 63,

$$\mathrm{ST}_t \ge \mathrm{atw}_r \mathrm{XRJ}_{r,j,t} \qquad \forall\, r\in\mathbf{R},\ j\in\mathbf{J}_r,\ t\in\mathbf{T} \tag{63}$$

while the value of $\mathrm{CR}_{r,j,t}$ is bounded by eq 64.

$$\mathrm{CR}_{r,j,t} \ge \mathrm{ST}_t + \left(\frac{\mathrm{UR}_{r,j,t}}{\mathrm{rdr}_r}\right) - h(1 - \mathrm{XRJ}_{r,j,t})$$

$$\forall\, r\in\mathbf{R},\ j\in\mathbf{J}_r,\ t\in\mathbf{T} \tag{64}$$

Using eqs 62a and 64, $\mathrm{CR}_{r,j,t}$ is never greater than CT_t if $\mathrm{XRJ}_{r,j,t} = 1$, to meet the condition that the delivery of a customer order should occur within a single time slot.

5.2.9.4. Tardiness of Customer Order r. The continuous variable CRF_r is the time at which customer order r is completely satisfied. Its value is given by eq 65.

$$\mathrm{CRF}_r \ge \mathrm{CR}_{r,j,t} \qquad \forall\, r\in\mathbf{R},\ j\in\mathbf{J}_r,\ t\in\mathbf{T} \tag{65}$$

The tardiness of order r (TD_r) then can be determined by eq 66.

$$\mathrm{TD}_r \ge \mathrm{CRF}_r - \mathrm{btw}_r \qquad \forall\, r\in\mathbf{R} \tag{66}$$

The parameter btw_r is the upper limit of the time window for order r.

5.2.10. Objective Function. The problem goal (eq 67) is the minimization of the total operating cost that includes the component consumption cost, the transition costs in blenders and product tanks, and the tardiness costs. In eq 67, the parameter ctd represents the tardiness cost per unit time, and scost_s is the unit cost of component s.

$$\min Z = \Big(\sum_{s\in\mathbf{S}} \sum_{i\in I} \sum_{p\in P} \mathrm{scost}_s \mathrm{QS}_{s,i,p} + \sum_{r\in\mathbf{R}} \mathrm{ctdTD}_r + \sum_{b\in\mathbf{B}} \mathrm{TTRB}_b + \sum_{p\in\mathbf{P}} \sum_{j\in\mathbf{J}_p} \sum_{\substack{t\in\mathbf{T}\\ t<|\mathbf{T}|}} \mathrm{TRJ}_{j,p,t}\Big) \tag{67}$$

Therefore, the MILP formulation of Cerdá et al.[6] has been generalized by including (a) the new constraints (eqs 32−34, 35a−35i, 36, and 38a−38c) to get a better estimate of key gasoline blend properties such as the RON, the MON, and the RVP; (b) eqs 24−28, 29a, and 30a to exactly monitor the inventory level in component tanks and (c) the nonlinear constraint (eq 37) only if the same recipe is to be adopted in all production campaigns of a final product.

6. THE MILP APPROXIMATE MODEL

In the proposed MINLP model, just the constraints described by eqs 24, 25, 27, 32−34, 36, and 37 are nonlinear. In order to replace eqs 24, 25, and 27 by approximate linear constraints, it is defined the binary variable $\mathrm{ZO}_{i',i}$ denoting the overlapping of the blending runs (i', i) with $i' < i$ whenever $\mathrm{ZO}_{i',i} = 1$. Equations 68a and 68b, which were first introduced by Cerdá et al.,[6] are used to determine the value of $\mathrm{ZO}_{i',i}$ with the parameter ε representing a very small number. When $\mathrm{ZO}_{i',i} = 0$, no overlapping of runs (i', i) occurs.

$$\mathrm{SB}_i \le \mathrm{CB}_{i'} - \varepsilon + H(1 - \mathrm{ZO}_{i',i}) \qquad \forall\, i\in\mathbf{I},\ i'\in\mathbf{IP}_i \tag{68a}$$

DOI: 10.1021/acs.iecr.6b01566
Ind. Eng. Chem. Res. 2016, 55, 7782−7800

$$CB_{i'} \le SB_i + H\,ZO_{i',i} \qquad \forall\, i \in \mathbf{I},\ i' \in \mathbf{IP}_i \tag{68b}$$

Because eq 29a in the proposed MINLP formulation intends to avoid the overloading of component tanks at the start time of a blending run i (i.e., at time SB_i), a conservative hypothesis consists of assuming that the total amount of component s allocated to an overlapping run $i' \in \mathbf{IP}_i$ featuring $ZO_{i',i} = 1$ is consumed after the start of run i, i.e., $QSP_{s,i',i} = \sum_{p \in \mathbf{P}} QS_{s,i',p}$ if $ZO_{i',i} = 1$. If runs (i, i') do not overlap, $ZO_{i',i} = 0$ and, consequently, $QSP_{s,i',i} = 0$. Therefore, the nonlinear eqs 24 and 25 are withdrawn from the model and the value of $QSP_{s,i',i}$ will now be given by eqs 69a and 69b instead of the constraint that is described by eq 25. No change in eq 28 is needed.

$$QSP_{s,i',i} \le MS_i\,ZO_{i',i} \tag{69a}$$

$$QSP_{s,i',i} \ge \sum_{p \in \mathbf{P}} QS_{s,i',p} - MS_{i'}(1 - ZO_{i',i})$$
$$\forall\, s \in \mathbf{S},\ (i', i) \in \mathbf{I}(i' \in IP_i) \tag{69b}$$

Moreover, the MINLP model includes eq 30a to avoid running-out conditions at the completion time of blending run i. If runs (i, i') with $i' \in IS_i$ overlap, the worst case occurs when they finish at the same time, i.e., $CB_i = CB_{i'}$. The nonlinear eq 27 then will be replaced by the linear constraints described by eqs 70a and 70b, by assuming that the total amount of component s assigned to the succeeding run $i' \in IS_i$ has been totally consumed at time CB_i when $ZO_{i,i'} = 1$. If runs (i, i') do not overlap, then $ZO_{i,i'} = 0$ and $QSS_{s,i,i'} = 0$.

$$QSS_{s,i,i'} \le MS_i\,ZO_{i,i'} \tag{70a}$$

$$QSS_{s,i,i'} \ge \sum_{p \in \mathbf{P}} QS_{s,i',p} - MS_{i'}(1 - ZO_{i,i'})$$
$$\forall\, s \in \mathbf{S},\ (i, i') \in \mathbf{I}(i' \in IS_i) \tag{70b}$$

In addition, the nonlinear equations described as eqs 32−36 will be replaced by the approximate linear constraints described by eqs 71−73 by assuming that the RON, MON, and RVP properties blend in an ideal fashion.

$$pron_p^{min} QB_{i,p} \le \sum_{s \in \mathbf{S}} sron_s QS_{s,i,p} \le pron_p^{max} QB_{i,p}$$
$$\forall\, i \in \mathbf{I},\ p \in \mathbf{P} \tag{71}$$

$$pmon_p^{min} QB_{i,p} \le \sum_{s \in \mathbf{S}} smon_s QS_{s,i,p} \le pmon_p^{max} QB_{i,p}$$
$$\forall\, i \in \mathbf{I},\ p \in \mathbf{P} \tag{72}$$

$$prvp_p^{min} QB_{i,p} \le \sum_{s \in \mathbf{S}} srvp_s QS_{s,i,p} \le srvp_p^{max} QB_{i,p}$$
$$\forall\, i \in \mathbf{I},\ p \in \mathbf{P} \tag{73}$$

The MILP formulation then comprises the objective function (eq 67) and a set of constraints that includes the linear equations described by eqs 7−23, 26, 28−31, 38−66, and 68−73. The approximate MILP formulation allows choosing different recipes if multiple production campaigns of the same product are performed. Equation 37, which forces us to use the same recipe for a product, even if multiple runs are executed, will just be considered when solving the MINLP model. The optimal solution to the MILP approximate model usually provides a good initial point for either the NLP or the MINLP solver.

7. THE MILP−NLP AND THE MILP−MINLP SOLUTION STRATEGIES

By fixing the integer variables to their MILP-optimal values, the MINLP formulation is reduced to a low-size NLP model. Since the MILP approximate model efficiently provides a very good initial point for the NLP solver, an effective MILP−NLP solution strategy is first proposed to determine a good feasible solution for the MINLP. In fact, the best MILP solution just slightly fails to meet some nonlinear quality specifications, especially the RON requirement. All other MINLP constraints are satisfied. As a result, the low-size NLP model is usually solved within less than a second and provides a near-optimal solution of the MINLP. An alternative solution approach consists of solving the MINLP formulation with the DICOPT solver using the MILP optimal solution as the starting point, i.e., a MILP−MINLP solution strategy. In this case, the values of the integer variables remain free when solving the MINLP formulation. With this strategy, the CPU solution time is still reasonable but longer than that required by the MILP−NLP scheme, because of the size of the MINLP. Just for larger problems, some improved results have been obtained using the MILP−MINLP approach. Then, the MILP−NLP method has very good performance, because it provides solutions quite close to the best ones at low computational cost. Both solution strategies were tested using the nonlinear correlations for RON and MON proposed by Healy et al.,[19] while those introduced by Stewart[20] was assessed by applying only the MILP−NLP strategy. Interestingly, vapor pressure constraints just become important for large gasoline blending problems with limited amounts of preferred gasoline components. Rules for selecting the number of blend runs and time slots are given in the Supporting Information.

8. RESULTS AND DISCUSSION

Eleven benchmark problems first introduced by Li et al.[10] and later studied by Li and Karimi[11] and Castillo-Castillo and Mahalec[12,13] have been solved, using the proposed MILP−NLP and MILP−MINLP approaches. They were called Examples 4−14 by Li et al.[10] and comprise 15−45 orders, 4−5 final products, 1−3 blend headers, 9 gasoline components, 11 product tanks, a time horizon of 192 h, and blend specifications for 9 critical properties, including RON, MOM, RVP, and flammability (FLA), as well as sulfur (SUL), olefins (OLE), benzene (BEN), aromatics (ARO), and oxygenates (OXY) contents. For each example, a pair of instances is considered: the single-period (SP) and the multiperiod (MP) cases. The component feed flow rates from upstream production units are assumed to be constant over the time horizon for the SP case, and piecewise constant functions of time for the MP instance. The two proposed solution approaches were applied to the MP instances of Examples 4−14, while the best solutions for the SP instances were found by using the MILP−NLP method. Data for the two instances of Examples 4−14 can be found in Li et al.[10] and are also provided as Supporting Information. Except for Example 5, the blend headers are initially idle. In Example 5, the unique blender is processing the final product P_1 and the production run underway has a length of 10 h and already yielded 150 kbbl at time zero. The models were implemented in GAMS 24.5.6[23] and solved using the solvers CPLEX 12.6 for

DOI: 10.1021/acs.iecr.6b01566
Ind. Eng. Chem. Res. 2016, 55, 7782−7800

the MILPs, CONOPT 3.15N for the NLPs, and DICOPT 24.2.2 for the MINLP formulations. All computations were performed on an Intel(R) Core i7 3632QM 2.20 GHz one-processor PC with 12 GB RAM and 4 cores. The relative optimality gap tolerance has been fixed to 0.001 for all the examples. DICOPT is stopped when a crossover occurs or the relative gap tolerance has been reached. Besides, a maximum CPU time of 3600 s was allowed.

8.1. Solving Examples 4−14 Using the MILP−NLP Strategy. Computational results for the SP and MP versions of Examples 4−14 using the MILP−NLP algorithm are shown in Tables 1 and 2, and these results are illustrated in a series of figures provided in the Supporting Information. Each figure includes a Gantt chart showing the sequence of operations performed in blenders and product tanks, and the inventory profiles in component and product tanks. The nonlinear correlation of Healy et al.[19] is used to predict the octane numbers of the gasoline blends. Model sizes and the number of time slots for each example are also given in the Supporting Information. From Table 1, it follows that the best solutions for both instances of Examples 4−10 were found within <15 s of CPU time, with an absolute gap equal to zero. The computational time still remains below 200 s to solve the MP instances of Examples 12−14. In all cases, most of the CPU time is allocated to solve the MILP model. Instead, the solution of the NLP subproblem is always found within <1 s using the MILP solution as the initial point. The SP version of the MILP model for Examples 11−14 is more difficult to solve than that for the MP instance, because it is less restricted in gasoline components and consequently presents a lower convergence rate to the optimal solution.

The best solutions found for all examples share some common features: no tardy orders and a minimum number of changeovers in blenders and product tanks (see Table 2). There are no product transitions in product tanks at Examples 4−8 just involving products P1−P4, and a single switching P3−P5 in the initially empty tank PT-2 originally assigned to product P3. Such a changeover arises because no product tank has been initially allocated to product P5. In addition, the least number of changeovers in blenders given by $[\mathrm{card}(\hat{P}) - \mathrm{card}(\mathbf{B})]$ is reached in all examples. $|\hat{P}|$ is the number of final products to be processed in the blenders to meet the customer demands. Table 2 provides the total component consumption cost, the changeover cost in blenders and product tanks, and the number of blending runs and order delivery operations in product tanks at the best solutions. In Tables 1 and 2, the MP^U instances of Examples 4−14 refer to MP instances that are constrained to have a unique recipe in all campaigns of the same final product. An interesting finding is obtained by replacing the rigorous nonlinear inventory equations by the approximate linear inventory modeling of Cerdá et al.[6] in the proposed MINLP formulation. Almost the same results shown in Table 1 are still found. Therefore, the linear inventory modeling of Cerdá et al.[6] already provides a precise tracking of the component tank inventory, thus avoiding the use of some nonlinearities in the MINLP model.

Compared with the values provided by the MILP approach of Cerdá et al.[6] for Examples 4−14, a growing increase in the minimum operating costs with the problem size is observed when using the proposed MINLP formulation. Such a comparison is given in the Supporting Information. The increase of the operating costs is almost exclusively due to the use of nonlinear correlations for estimating the RON, MON, and RVP properties.

Table 1. Computational Results for Examples 4−14 Using the MILP−NLP Solution Strategy

example		MILP Approximate Model cost (× 10^3 $)	MILP Approximate Model CPU time (s)	NLP Model cost (× 10^3 $)	NLP Model CPU time (s)
4−5	SP	4596.69	0.36	4612.68	0.03
	MP	4596.69	0.34	4612.68	0.06
6	SP	5252.87	0.33	5271.48	0.02
	MP	5252.87	0.47	5271.48	0.11
7	SP	8153.85	1.6	8183.07	0.11
	MP	8153.85	3.5	8183.07	0.13
8	SP	8133.85	1.5	8163.07	0.02
	MP	8133.85	7.5	8163.07	0.02
9	SP	10645.20	8.5	10663.30	0.14
	MP	10647.52	3.2	10664.79	0.11
10	SP	11367.47	5.0	11374.31	0.11
	MP	11391.38	4.5	11406.06	0.13
	MP^U	11391.38	13.9	11406.15	0.13
11	SP	13344.81	412.0	13358.01	0.14
	MP	13379.04	357.7	13403.23	0.27
	MP^U	13379.04	356.8	13403.29	0.39
12	SP	14884.67	1388.4	14921.24	0.30
	MP	15319.08	30.8	15341.67	0.28
	MP^U	15319.08	31.8	15342.60	0.53
13	SP	18121.43	470.3	18193.65	0.38
	SP^U	18124.65	512.6	18198.60	0.41
	MP	18780.48	198.1	18699.61	0.33
	MP^U	18780.48	125.5	18699.82	0.28
14	SP	20495.30	1229.2	20512.54	0.66
	SP^U	20495.30	1262.2	20519.63	0.25
	MP	21424.06	20.1	21181.10	0.70
	MP^U	21424.06	53.1	21195.95	0.27

8.2. Comparison of the Results with Previous Approaches. Castillo-Castillo and Mahalec[12,13] applied two different methodologies to solve the MP versions of Examples {4, 8, 12, 14}, using the nonlinear correlations of Healy et al.[19] to predict the octane numbers of gasoline blends. A comparison with the results reported by Castillo-Castillo and Mahalec[12,13] is shown in Table 3. In our work, a product transition implies that two different products are produced in a pair of consecutive runs performed in the same blender. In contrast, Castillo-Castillo and Mahalec[12,13] assumed that a changeover occurs whenever a new production run is performed in a blender. If our changeover criterion were used, the optimal blend costs reported by Castillo-Castillo and Mahalec[12,13] for Examples {4, 8, 12, 14} would decrease to {4613.0; 8165.4; 15366.8; 21203.1} and {4612.7; 8163.1; 15362.6; 21203.1}, respectively. Therefore, the optimal values for Examples 4 and 8 at which every approach performs the same number of blend runs are mostly similar. However, this is not the case for Examples 12

DOI: 10.1021/acs.iecr.6b01566
Ind. Eng. Chem. Res. 2016, 55, 7782−7800

Table 2. Values of the Different Cost Items for Examples 4−14 Using the MILP−NLP Strategy

example		component cost (× 10^3 $)	Switching Cost (×10^3 $) in blenders	Switching Cost (×10^3 $) in product tanks	blend runs (BR)	delivery runs (DR)	total cost (× 10^3 $)
4−5	SP	4592.68	20		2	20	4612.68
	MP	4592.68	20		2	20	4612.68
6	SP	5231.48	40		3	22	5271.48
	MP	5231.48	40		3	22	5271.48
7	SP	8123.07	60		4	26	8183.07
	MP	8123.07	60		4	26	8183.07
8	SP	8123.07	40		4	26	8163.07
	MP	8123.07	40		5	25	8163.07
9	SP	10588.80	60	14.5	5	29	10663.30
	MP	10590.29	60	14.5	6	29	10664.79
10	SP	11299.81	60	14.5	6	32	11374.31
	MP	11331.56	60	14.5	6	32	11406.06
	MP^U	11331.65	60	14.5	6	32	11406.15
11	SP	13283.78	60	14.5	8	39	13358.28
	MP	13328.73	60	14.5	8	39	13403.23
	MP^U	13328.79	60	14.5	8	39	13403.29
12	SP	14853.33	60	14.5	8	45	14927.83
	MP	15267.17	60	14.5	8	45	15341.67
	MP^U	15268.10	60	14.5	8	45	15342.60
13	SP	18136.89	40	14.5	8	54	18191.39
	SP^U	18144.10	40	14.5	8	60	18198.60
	MP	18645.11	40	14.5	8	50	18699.61
	MP^U	18645.32	40	14.5	8	50	18699.82
14	SP	20458.04	40	14.5	9	61	20512.54
	SP^U	20465.13	40	14.5	9	62	20519.63
	MP	21126.60	40	14.5	8	54	21181.10
	MP^U	21141.45	40	14.5	8	57	21195.95

Table 3. Comparison with the Results Reported by Castillo-Castillo and Mahalec[12,13]

	Our MILP−NLP Method			Castillo-Castillo and Mahalec[12]			Castillo-Castillo and Mahalec[13]		
example	objective value	CPU time (s)	gap (%)	objective value	CPU time (s)	gap (%)	objective value	CPU time (s)	gap (%)
4	4612.68	0.4	0.0	4633.0	15	0.01	4632.7	3.0	0.0
8	8163.07	7.5	0.0	8205.4	10 800[a]	0.03	8203.1	6.0	0.0
12	15341.67	31.1	0.0	15406.8	43 200[a]	0.16	15402.6	17.0	0.0
14	21181.10	20.8	0.0	21283.1	43 200[a]	0.19	21263.1	24.0	0.0

[a]CPU time limit.

and 14, because two and three more runs are executed in our best solutions. As a result, the proposed MILP-NLP algorithm is able to get an extra cost-savings of 20.93 and 21.88 (× 10^3 $) for Examples 12 and 14, respectively.

8.3. Solving the MP Instances of Examples 4−14 Using the MILP−MINLP Approach. The multiperiod instances of Examples 4−14 were also solved using the MILP−MINLP solution strategy. Computational results are shown in Table 4. It can be observed that the total CPU time is rather evenly allocated to the solution of the MILP and MINLP models. Moreover, it presents a moderate increase, with regard to the MILP−NLP strategy for the larger examples (Examples 11−14) mostly due to the solution of the MINLP. Computational results provided by the MILP−NLP and the MILP−MINLP strategies for Examples 4−14 are compared in Table 5. Both approaches provide the same best solutions for Examples 4−9, while some slight improvements in operating costs are obtained for Examples 10−14 through the MILP−-MINLP method at the expense of a reasonable increase of the computational cost.

From Table 5, the following conclusions can be drawn: (i) the MILP−MINLP methodology just yields slightly better results for larger examples; (ii) the MILP−NLP usually solves the MP examples at low CPU time and always finds near-

DOI: 10.1021/acs.iecr.6b01566
Ind. Eng. Chem. Res. 2016, 55, 7782−7800

Table 4. Results for the MP Instance of Examples 4−14 Using the MILP−MINLP Approach

example		MILP Approximate Model cost (× 10^3 $)	MILP Approximate Model CPU time (s)	MINLP Model cost (× 10^3 $)	MINLP Model CPU time (s)
4	MP	4596.69	0.4	4612.68	0.75
6	MP	5252.87	0.5	5271.48	2.4
7	MP	8153.85	2.0	8183.07	6.9
8	MP	8133.85	1.9	8163.07	95.4
9	MP	10647.52	21.4	10664.79	46.0
10	MP	11391.38	3.6	11405.96	6.6
	MP^U	11391.38	3.6	11406.15	4.8
11	MP	13379.00	354.9	13402.96	408.8
	MP^U	13379.00	400.5	13403.19	411.5
12	MP	15319.08	73.4	15340.43	235.3
	MP^U	15319.08	71.9	15342.60	110.1
13	MP	18780.48	194.0	18697.39	395.6
	MP^U	18780.48	194.3	18699.82	43.7
14	MP	21424.06	141.0	21180.58	343.7
	MP^U	21424.06	140.3	21186.19	385.5

Table 5. Comparing Results Provided by the MILP−NLP and the MILP−MINLP Approaches

example		MILP−NLP Approach cost (× 10^3 $)	MILP−NLP Approach CPU time (s)	MILP−MINLP Approach cost (× 10^3 $)	MILP−MINLP Approach CPU time (s)
4−5	MP	4612.68	0.4	4612.68	1.2
6	MP	5271.48	0.6	5271.48	2.9
7	MP	8183.07	3.6	8183.07	8.9
8	MP	8163.07	7.5	8163.07	97.3
9	MP	10664.79	3.3	10664.79	67.4
10	MP	11406.06	4.6	11405.96	10.2
	MP^U	11406.15	14.0	11406.15	8.4
11	MP	13403.23	358.0	13402.96	763.7
	MP^U	13403.29	357.2	13403.19	812.0
12	MP	15341.67	31.1	15340.43	308.7
	MP^U	15342.60	32.3	15342.60	182.0
13	MP	18699.61	198.4	18697.39	589.6
	MP^U	18699.82	125.8	18699.82	238.0
14	MP	21181.10	20.8	21180.58	484.7
	MP^U	21195.95	53.4	21186.19	525.8

optimal solutions. Figure 2 shows the best solution found for Example 9 (MP), using either approach, while Figures 3 and 4

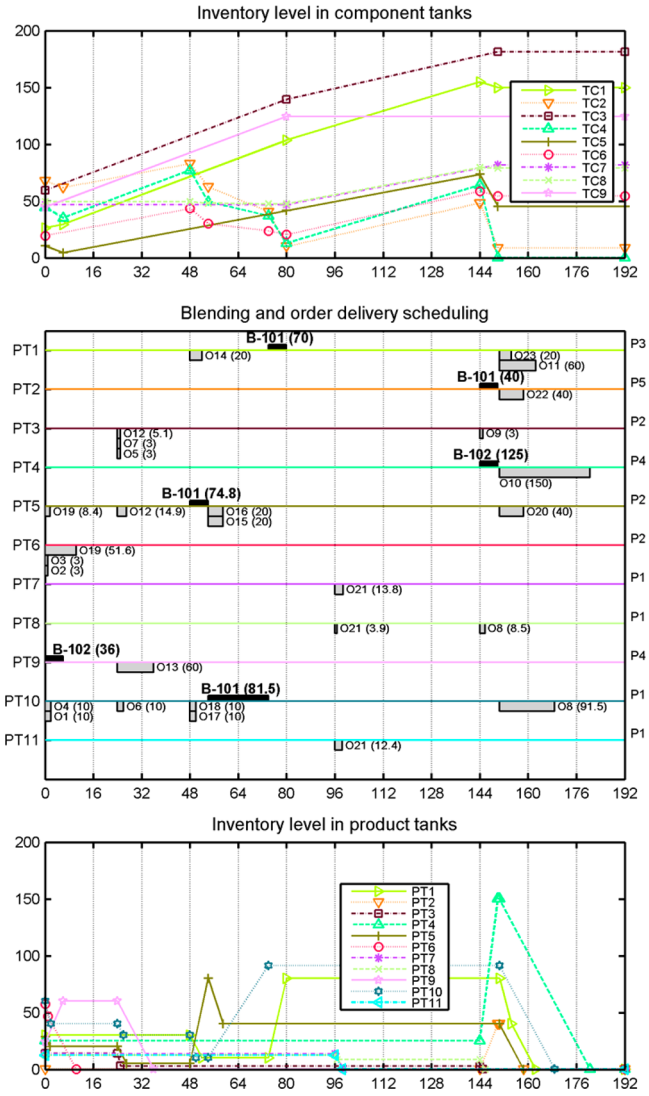


Figure 2. Best solution found for the MP instance of Example 9.

present the results for the MP instances of Examples 12 and 14 discovered by the MILP−MINLP method.

8.4. Optimal Product Recipes. Table 6 shows the product recipes selected at the best solutions of Examples 4−12 (MP). Similar results for the SP instances are provided in the Supporting Information. It is observed that the optimal product recipes for both versions of Examples 4−8 are exactly the same, thus indicating no limitation in the available amounts of the preferred gasoline components. This recipe pattern still remains without changes for the SP instances of Examples 9 and 10, featuring a constant component flow rate from upstream units over the entire time horizon.

In contrast, Example 9 (MP) presents some shortages in the amount of the preferred component C4, because of higher product demands and a decreasing feed flow rate of C4 with time. Therefore, the optimal recipe of product P4 with the lowest RON specification experiences some small changes. As a result, the total component cost slightly increases and the optimal values for the SP and MP instances of Example 9 are no longer equal (see Table 6).

Looking at Figure 2, it is observed that the inventory level in the component tank devoted to C4 drops to zero at time = 150 h. Moreover, there are no inventories in product tanks at the

DOI: 10.1021/acs.iecr.6b01566
Ind. Eng. Chem. Res. 2016, 55, 7782−7800

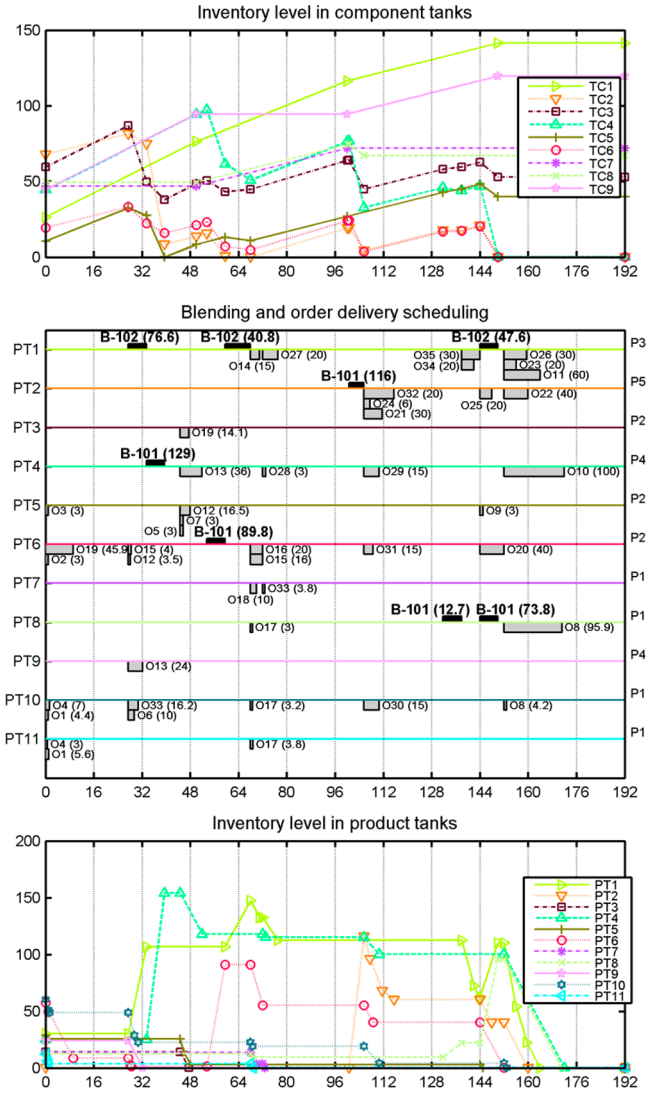


Figure 3. Best solution found for the MP instance of Example 12.

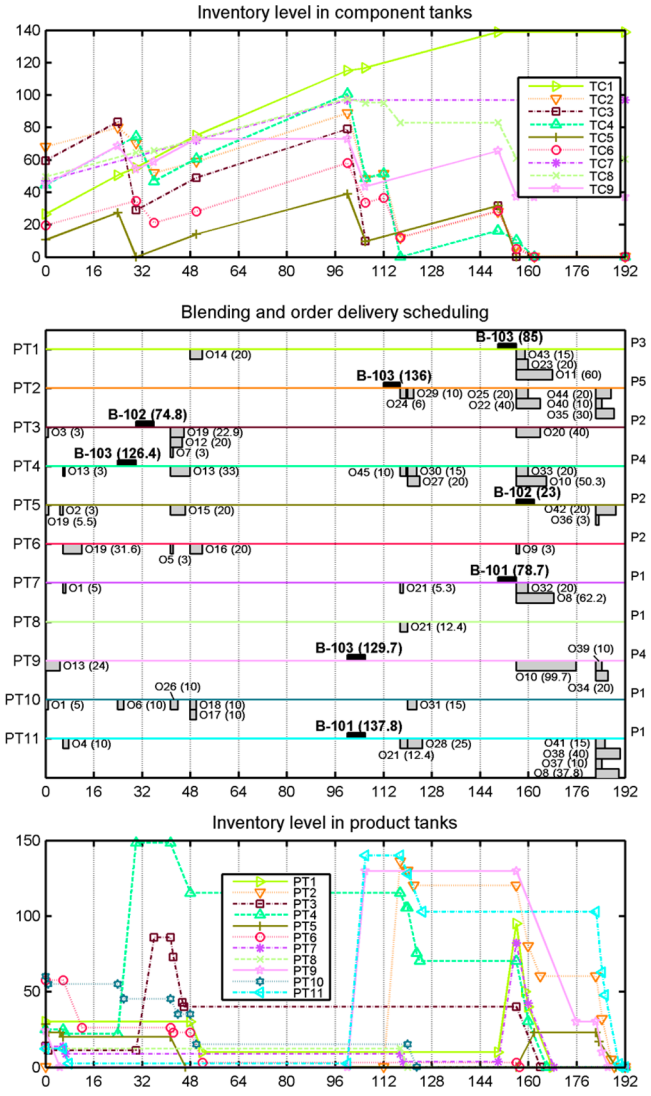


Figure 4. Best solution found for the MP instance of Example 14.

horizon end. Interestingly, the product recipes chosen by the MILP model are slightly modified when solving the NLP. Almost all the selected gasoline components remain the same but their relative proportions undergo some small changes. This pattern is observed in almost all examples. Note that the recipe of the product assigned to any nonfictitious run is still a problem variable when solving the NLP subproblem. Recipe changes are expanded to other products in Examples 10−12, because of increasing shortages of the preferred components C2 and C4 (see Figure 3). In addition, there appears to be a need to run multiple campaigns of some final products to reduce the feedstock cost. At Example 14, shortages of component C2 continue to grow and the available amount of component C3 is not sufficient to cover such a limitation of component C2. As a result, the low-cost component C9 is incorporated in the recipes of P1−P4 until reaching the RVP upper limit (0.905 bar) for product P3. Component C9 has a low RON of 81 and a high RVP (2.068 bar). The low RON of C9 is compensated by incorporating or increasing the volume fraction of component C8 in the blend recipes.

The nonlinear correlation of Healy et al.[19] provides RON values for the blends lower than those obtained with the linear correlation used by Cerdá et al.[6] As a result, the stocks of the preferred components [C2, C4, and C6] are depleted earlier and the optimal blends given by the MINLP approach incorporate or contain higher proportions of the less-preferred gasoline components [C3, C5, and C9].

8.5. Using the Same Recipe in All Production Runs of a Final Product. For large examples, multiple runs of the same product are performed to minimize the total component cost, but not necessarily share a common recipe. Because such production campaigns can be allocated to the same product tank, the scheduler may desire to adopt the same recipe in all runs. Such constrained problems are identified by the superscript "U". Multiple runs are executed at the best solutions of the SP instances of Examples 13 and 14, and the MP versions of Examples 10−14. By incorporating eq 37 in the formulation of either the NLP subproblem or the MINLP model, new solutions are found using a unique recipe in all campaigns of a final product. Computational results for the SP^U instances of Examples 13 and 14 and the MP^U versions of Examples 10−14 are shown in Tables 1 and 2 and Tables 4 and 5, respectively. As expected, the common recipe constraint usually produces a slight increase in the optimal total cost. Figures illustrating the best unique-recipe solutions for Examples 13 and 14 (SP) and Examples 10−14 (MP), and

DOI: 10.1021/acs.iecr.6b01566
Ind. Eng. Chem. Res. 2016, 55, 7782−7800

Table 6. Optimal Product Recipes at the Best Solutions of the MP Versions of Examples 4−12

		Component Fraction (%)								
example	product	C1	C2	C3	C4	C5	C6	C7	C8	C9
4−5	P1		39.34		**40.00**		**20.00**		0.66	
	P4	8.58	22.42		**44.00**	**25.00**				
6	P1		39.34		**40.00**		**20.00**		0.66	
	P3		47.95		**43.00**		9.05			
	P4	8.58	22.42		**44.00**	**25.00**				
7−8	P1		39.34		**40.00**		**20.00**		0.66	
	P2		31.14		**45.00**		**22.00**		1.86	
	P3		47.95		**43.00**		9.05			
	P4	8.58	22.42		**44.00**	**25.00**				
9	P1		39.34		**40.00**		**20.00**		0.66	
	P2		31.14		**45.00**		**22.00**		1.86	
	P3		47.95		**43.00**		9.05			
	P4	7.91	24.82		42.27	**25.00**				
	P5		31.05		**40.00**		**20.00**		8.95	
10	P1		39.34		**40.00**		**20.00**		0.66	
	P2		31.14		**45.00**		**22.00**		1.86	
	P3		29.17		**43.00**	9.83	**18.00**			
	P4	6.85	29.57		30.84	**25.00**	7.74			
	P5		31.05		**40.00**		**20.00**		8.95	
11	P1		39.34		**40.00**		**20.00**		0.66	
	P2		31.14		**45.00**		**22.00**		1.86	
	P3		29.17		**43.00**	9.83	**18.00**			
	P4	8.98	23.59		22.43	**25.00**	**20.00**			
		7.84	26.54		28.95	**25.00**	11.67			
		5.05	35.07		34.88	**25.00**				
	P5		31.05		**40.00**		**20.00**		8.95	
12	P1		20.05	13.47	**40.00**	6.48	**20.00**			
			10.98	18.91	**40.00**	10.11	**20.00**			
	P2		19.35	13.12	**45.00**	0.53	**22.00**			
	P3		11.57	10.36	**43.00**	17.07	**18.00**			
			19.38	41.29	10.30	11.03	**18.00**			
	P4		53.81	13.78		**25.00**	7.41			
	P5		18.80	14.31	**40.00**		**20.00**		6.89	

the blend recipes selected for every final product are provided in the Supporting Information.

Compared with the multiple-recipe solutions, there is no change in the gasoline components included in the blends; only their volume fractions suffer some minor modifications. Another common feature of the optimal recipes, even in the presence of component shortages, is that the RON lower limit specified for each final product is strictly satisfied in all examples. In this way, quality giveaways are avoided. Limiting specifications for the other blend properties are never active, except for the RVP upper limit in Example 14. Values of the blend properties at the best solutions of Examples 10−14 (MP^U) are given in the Supporting Information.

8.6. Using the Nonlinear Correlations of Stewart[20] To Predict the Octane Numbers. To corroborate the computational efficiency of the proposed MILP−NLP algorithm to solve nonlinear blend scheduling problems, the multiperiod versions of Examples 8−14 were again solved, but this time using the nonlinear correlations of Stewart[20] to predict octane numbers. Computational results shown in Table 7 confirm the good performance of the proposed MILP−NLP solution algorithm.

In all cases, the minimum RON specification is strictly satisfied. Therefore, it can be concluded that the Stewart[20] correlation predicts higher RON values and lower operating costs than Healy et al.[19] for Examples 4−14. When the Stewart[20] correlation is used, DICOPT fails to improve the initial solution found by solving the first NLP, because of error problems that are due to the denominator of the RON expression tending to zero.

Table 7. Results Found Using the Nonlinear Correlations of Stewart[20] and Healy et al.[19]

	MILP−NLP Methods			
	Stewart[20]		Healy et al.[19]	
example (MP)	objective function	CPU time (s)	objective function	CPU time (s)
8	8119.01	1.1	8163.07	7.5
9	10620.02	3.2	10664.79	3.3
10	11359.11	3.7	11406.06	4.6
11	13358.65	100.4	13403.23	358.0
12	15306.10	73.2	15341.67	31.1
13	18599.16	46.9	18699.61	198.4
14	21102.31	131.5	21181.10	20.8

By analyzing the solutions yielded by the MILP and NLP models when using the Healy et al.[19] correlation, it is observed negative deviations from ideal mixing for products P1−P4 and a positive deviation for P5 at Examples 4−12 (MP). As a result, the total operating cost increases when solving the NLP

DOI: 10.1021/acs.iecr.6b01566
Ind. Eng. Chem. Res. 2016, 55, 7782−7800

formulation. In contrast, positive deviations from linear blending are predicted by the Healy et al.[19] correlation for products P1−P2 and P4−P5 at Examples 13 and 14 (MP). Consequently, the optimal value of the NLP model is lower than the value of the approximate MILP.

9. CONCLUSIONS

A mixed-integer nonlinear programming (MINLP) continuous-time formulation for the simultaneous optimization of gasoline recipes and the scheduling of blending and order delivery operations has been developed. The proposed model is an extension of the MILP approach introduced by Cerdá et al.,[6] which now includes a set of nonlinear equations to better estimate key blend properties, such as octane numbers and Reid vapor pressure (RVP), and exactly tracking inventory levels in product tanks when multiple blend headers are operated. Additional nonlinear constraints can also be incorporated to force using the same recipe if multiple blending runs yielding the same finished product are to be performed to minimize the operating cost. The proposed MINLP formulation is based on floating slots that are so named because they are not preassigned to time periods but dynamically allocated to them while solving the problem. Simple mathematical expressions for choosing the number of blending runs and floating time slots were also developed. A pair of two-stage MINLP solution strategies has been defined: the mixed-integer linear programming−nonlinear programming (MILP−NLP) and the mixed-integer linear programming−mixed-integer nonlinear programming (MILP−MINLP) methods. In both cases, an approximate MILP model derived from assuming ideal mixing provides a good initial integer point. After fixing the integer variables, the resulting NLP is solved to determine an MINLP near-optimal solution. When applying the MILP−MINLP strategy using the MILP−integer solution as the starting configuration slightly improved solutions are often discovered in a few iterations.

Eleven benchmark examples first introduced by Li et al.[10] and later studied by Li and Karimi[11] and Castillo-Castillo and Mahalec[12,13] have been successfully solved using the two solution approaches. The component flow-rate from upstream units was assumed to be either constant (single period (SP) scenario) or a piecewise constant function of time (multiperiod (MP) scenario). For the latter case, the MILP−NLP algorithm systematically finds near-optimal solutions at low computational cost. In turn, the MILP−MINLP approach discovers slightly better solutions after three or four additional major DICOPT iterations at the expense of an extra, usually reasonable, CPU time. Single-period scenarios are more difficult to tackle just for larger examples featuring limited amounts of preferred gasoline components. A comparison with the results reported in previous contributions shows that our approach achieves a small cost-savings for large MP examples, mostly because two or three more blending runs are executed. If the scheduler desires to keep a unique recipe in all production runs, the proposed approach can include an additional constraint forcing that condition and producing a slight increase in the feedstock cost. A common feature of the optimal recipes in both versions of all examples, even in the presence of component shortages, is that the minimum research octane number (RON) specified for each final product is always strictly satisfied. In this way, quality giveaways are avoided. Limiting specifications for the other blend properties are never active, with the exception of the RVP property under severe shortages of preferred gasoline components.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge on the ACS Publications website at DOI: 10.1021/acs.iecr.6b01566.

It presents rules for selecting the number of production runs and time slots, and a set of tables with all the problem data, the model sizes, and detailed computational results for Examples 4−14; also included is a set of figures showing the best solutions for the single-period and multiperiod instances of all examples (PDF)

■ AUTHOR INFORMATION

Corresponding Author

*Tel.: +54 342 4559175. Fax: +54 342 4550944. E-mail: jcerda@intec.unl.edu.ar.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

The authors acknowledge financial support from FONCYT-ANPCyT under Grant PICT 2014-2392, from CONICET under Grant PIP-940, and from Universidad Nacional del Litoral under Grant CAI+D 2011-256.

■ NOTATION

Sets

$\mathbf{B}$ = blender headers
$\mathbf{K}$ = ordered set of time periods
$\mathbf{I}$ = ordered set of production runs in blenders
$\mathbf{IP}_i$ = preceding runs of campaign i that can be processed in parallel with i
$\mathbf{IS}_i$ = succeeding runs of campaign i that can be processed in parallel with i
$\mathbf{P}$ = final products
$\mathbf{P}_j$ = final products that can be stored in tank j
$\mathbf{T}$ = process time slots

Parameters

$\mathrm{ctrb}_{b,p,p'}$ = 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
$\mathrm{iij}_{p,j}$ = initial inventory of product p in the tank $j \in \mathbf{J}_p$
iis_s = initial inventory of gasoline component s
llim_k = lower time limit for period k
M_{B} = a relatively large number
pcap_j = capacity of the product tank j
$\mathrm{pdr}_{p,j}$ = delivery rate of product p from product tank j
$\mathrm{ppr}_{g,p}^{\max}$ = maximum limiting value of property g per unit amount of product p
$\mathrm{ppr}_{g,p}^{\min}$ = minimum limiting value of property g per unit amount of product p
q_r = size of order r
$\mathrm{rb}_{b,p}^{\max}$ = maximum processing rate limit for final product p in blender b
$\mathrm{rb}_{b,p}^{\min}$ = minimum processing rate limit 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

DOI: 10.1021/acs.iecr.6b01566
Ind. Eng. Chem. Res. 2016, 55, 7782−7800

$smon_s$ = motor octane number of gasoline 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
$sron_s$ = research octane number of component s
$srvp_s$ = Reid vapor pressure of component s
$svr_{s,k}$ = feed flow rate of component s during period k
$ulim_k$ = upper time limit for period k
$vc_{s,k}$ = feed rate of blending component s during period k
$vc_{s,p}^{max}$ = maximum proportion of component s in the final product p
$vc_{s,p}^{min}$ = minimum proportion of component s in the final product p
$\tau_{b,p,p'}$ = 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 from 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)

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 customer order r is 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 $j \in \mathbf{J}_p$ at the end of slot t
$PMON_{i,p}$ = motor octane number of blend p produced by run i
$PRON_{i,p}$ = research octane number of blend p produced by run i
$PRVP_{i,p}$ = Reid vapor pressure of product p yielded by run i
$QB_{i,p}$ = 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 $s \in \mathbf{S}$ assigned to run i producing product p
$QSP_{s,i',i}$ = amount of component s consumed at the preceding run $i' \in \mathbf{IP}_i$ after time SB_i
$QSS_{s,i,i'}$ = amount of component s consumed at the succeding run $i' \in \mathbf{IS}_i$ before time CB_i
$RS_{s,i}$ = consumption rate of component s during 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 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 time slot t
$UR_{r,j,t}$ = amount delivered for order r from tank j during time slot t
$USP_{s,i}$ = amount of component s consumed up to time SB_i
$USS_{s,i}$ = amount of component s consumed up to time CB_i
$VFR_{s,i,p}$ = volume fraction of component s in blend p yielded by run i
$WBP_{i,b,p}$ = identifies the final product yielded by production run i in blender b

REFERENCES

(1) DeWitt, C. W.; Lasdon, L. S.; Waren, A. D.; Brenner, D. A.; Melhem, S. A. OMEGA: An improved gasoline blending system for Texaco. *Interfaces* **1989**, *19*, 85−101.
(2) Palmer, F. H.; Smith, A. M. The performance and specification of gasolines. In *Technology of Gasoline*; Hancock, E.G., Ed.; Blackwell Scientific Publications: London, 1985.
(3) Gary, J. H.; Handwerk, G. E. *Petroleum Refining Technology and Economics*, 3rd Ed.; Marcel Dekker: New York, 1994.
(4) Agrawal, S. S. Integrate blending control, optimization and planning. *Hydrocarbon Process.* **1995**, *74* (8), 129−139.
(5) Givens, L. Modern petroleum refining: An overview. *Automot. Eng.* **1985**, *93*, 69−77.
(6) Cerdá, J.; Pautasso, P. C.; Cafaro, D. C. A cost-effective model for the gasoline blend optimization problem. *AIChE J.* **2016**, DOI: 10.1002/aic.15208.
(7) Glismann, K.; Gruhn, G. Short-term scheduling and recipe optimization of blending processes. *Comput. Chem. Eng.* **2001**, *25*, 627−634.
(8) Jia, Z.; Ierapetritou, M. Efficient short-term scheduling of refinery operations based on a continuous time formulation. *Comput. Chem. Eng.* **2004**, *28*, 1001−1019.
(9) Méndez, C. A.; Grossmann, I. E.; Harjunkoski, I.; Kabore, P. A simultaneous optimization approach for off-line blending and scheduling of oil-refinery operations. *Comput. Chem. Eng.* **2006**, *30*, 614−634.
(10) Li, J.; Karimi, I. A.; Srinivasan, R. Recipe determination and scheduling of gasoline blending operations. *AIChE J.* **2009**.
(11) Li, J.; Karimi, I.A. Scheduling gasoline blending operations from recipe determination to shipping using unit slots. *Ind. Eng. Chem. Res.* **2011**, *50*, 9156−9174.
(12) Castillo-Castillo, P. A.; Mahalec, V. Improved continuous-time model for gasoline blend scheduling. *Comput. Chem. Eng.* **2016**, *84*, 627−646.
(13) Castillo-Castillo, P. A.; Mahalec, V. Inventory pinch gasoline blend scheduling algorithm combining discrete- and continuous-time models. *Comput. Chem. Eng.* **2016**, *84*, 611−626.
(14) Kolodziej, S.; Grossmann, I. E.; Furman, K. C.; Sawaya, N. W. A discretization-based approach for the optimization of the multiperiod blend scheduling problem. *Comput. Chem. Eng.* **2013**, *53*, 122−142.
(15) Teles, J. P.; Castro, P. M.; Matos, H. A. Multi-parametric disaggregation technique for global optimization of polynomial programming problems. *J. Global Optim.* **2013**, *55*, 227−251.
(16) Castro, P. M. New MINLP formulation for the multiperiod pooling problem. *AIChE J.* **2015**, *61*, 3728−3738.
(17) Lotero, I.; Trespalacios, F.; Grossmann, I. E.; Papageorgiou, D. J.; Cheon, M.-S. An MILP-MINLP decomposition method for the

DOI: 10.1021/acs.iecr.6b01566
Ind. Eng. Chem. Res. 2016, 55, 7782−7800

global optimization of a source based model of the multiperiod blending problem. *Comput. Chem. Eng.* **2016**, *87*, 13−35.
(18) Rusin, M. H.; Chung, H. S.; Marshall, J. F. A "transformation" method for calculating the research and motor octane numbers of gasoline blends. *Ind. Eng. Chem. Fundam.* **1981**, *20*, 195−204.
(19) Healy, W. C., Jr.; Maasen, C. W.; Peterson, R. T. A new approach to blending octanes. In *Proceedings of the 245h Midyear Meeting of the American Petroleum Institute's Division of Refining*, Vol. *39*, New York, 1959.
(20) Stewart, W. E. Predict octanes for gasoline blends. *Pet. Refin.* **1959**, *38*, 135−139.
(21) Stewart, W. E. Predict RVP of blends accurately. *Pet. Refin.* **1959**, *38*, 231−234.
(22) Morris, W.E. The interaction approach to gasoline blending. Presented at the *National Petroleum Refiners Association Annual Meeting*; NPRA Paper No. AM-7530, March 1975.
(23) Brooke, A.; Kendrick, D.; Meeraus, A. *GAMS—A User's Guide*, GAMS release 24.5.6; GAMS Development Corporation, Washington, DC, USA, 2015.

DOI: 10.1021/acs.iecr.6b01566
Ind. Eng. Chem. Res. 2016, 55, 7782−7800