Available online at www.sciencedirect.com
SCIENCE DIRECT®

Computers and Chemical Engineering 30 (2006) 614–634

Computers & Chemical Engineering

www.elsevier.com/locate/compchemeng

A simultaneous optimization approach for off-line blending and scheduling of oil-refinery operations

Carlos A. Méndez [a], Ignacio E. Grossmann [a,*], Iiro Harjunkoski [b], Pousga Kaboré [b]

[a] *Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, USA*
[b] *ABB Corporate Research Center, Ladenburg, Germany*

Received 22 December 2004; received in revised form 19 September 2005; accepted 7 November 2005
Available online 6 January 2006

Abstract

This paper presents a novel MILP-based method that addresses the simultaneous optimization of the off-line blending and the short-term scheduling problem in oil-refinery applications. Depending on the problem characteristics as well as the required flexibility in the solution, the model can be based on either a discrete or a continuous time domain representation. In order to preserve the model's linearity, an iterative procedure is proposed to effectively deal with non-linear gasoline properties and variable recipes for different product grades. Thus, the solution of a very complex MINLP formulation is replaced by a sequential MILP approximation. Instead of predefining fixed component concentrations for products, preferred blend recipes can be forced to apply whenever it is possible. Also, different alternatives for coping with infeasible problems are presented. Sufficient conditions for convergence for the proposed approach are presented as well as a comparison with NLP and MINLP solvers to demonstrate that the method provides an effective integrated solution method for the blending and scheduling of large-scale problems. The new method is illustrated with several real world problems requiring very low computational requirements.

Keywords: Scheduling; Planning; Blending; Refinery operations; Mixed-integer programming

1. Introduction

The main objective in oil refining is to convert a wide variety of crude oils into valuable final products such as gasoline, jet fuel and diesel. The short-term blending and scheduling are critical aspects in this large and complex process. The economic and operability benefits associated with obtaining better-quality and less expensive blends, and at the same time making a more effective use of the available resources over time, are numerous and significant. A wide variety of mathematical programming techniques have been extensively used for long-term planning as well as the short-term scheduling of refinery operations.

For planning problems, most of the computational tools have been based on successive linear programming models, such as RPMS from Honeywell, Process Solutions (formerly Booner & Moore, 1979) and PIMS from Aspen Technology (formerly Bechtel Corp., 1993). On the other hand, scheduling problems have been addressed through linear and non-linear mathematical approaches that make use of binary variables (MILP and MINLP codes) to explicitly model the discrete decisions to be made (Grossmann, Van den Heever, & Harjunkoski, 2002; Shah, 1998). Short-term scheduling problems have been mainly studied for batch plants. Extensive reviews can be found in Reklaitis (1992), Pinto and Grossmann (1998), Kallrath (2003) and, Floudas and Lin (2004). Much less work has been devoted to continuous plants. Lee, Pinto, Grossmann, and Park (1996) addressed the short-term scheduling problem for the crude-oil inventory management problem. Non-linearities of mixing tasks were reformulated into linear inequalities with which the original MINLP model was converted to a MILP formulation that can be solved to global optimality. According to the authors, this linearization was possible because only mixing operations were considered (see Quesada & Grossmann, 1995). However, it was later pointed out by Wenkai, Hui, Hua, and Tong (2002) that the proposed reformulation linearization technology (RLF) may lead to composition discrepancy (the amounts of individual crudes delivered from a tank to CDU are not proportional to the crude composition in the tank). The objective function was

* Corresponding author. Tel.: +1 412 268 2230; fax: +1 412 268 7139.
E-mail address: grossmann@cmu.edu (I.E. Grossmann).

0098-1354/$ – see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compchemeng.2005.11.004

the minimization of the total operating cost, which comprises waiting time cost of each vessel in the sea, unloading cost for crude vessels, inventory cost and changeover cost. Several examples were solved to highlight the computational performance of the proposed model. Moro, Zanin, and Pinto (1998) developed a mixed-integer non-linear programming planning model for refinery production. The model assumes that a general refinery is composed of a number of processing units producing a variety of input/output streams with different properties, which can be blended to satisfy different specifications of diesel oil demands. Each unit belonging to the refinery is defined as a continuous processing element that transforms the input streams into several products. The general model of a typical unit is represented by a set of variables such as feed flowrates, feed properties, operating variables, product flowrates and product properties. The main objective is to maximize the total profit of the refinery, taking into consideration sales revenue, feed costs and the total operating cost. Wenkai et al. (2002) proposed a solution algorithm that iteratively solves two mixed-integer linear programming (MIP) models and a non-linear programming (NLP) model, resulting in better quality, stability, and efficiency than solving the MINLP model directly. Kelly and Mann (2003a,b) highlight the importance of optimizing the scheduling of an oil-refinery's crude-oil feedstock from the receipt to the charging of the pipestills. The use of successive linear programming (SLP) was proposed for solving the quality issue in this problem. More recently, Kelly (2004) analyzed the underlying mathematical modeling of complex non-linear formulations for planning models of semi-continuous facilities where the optimal operation of petroleum refineries and petrochemical plants was mainly addressed.

In addition, the off-line blending problem, also known as blend planning has been addressed through several optimization tools. The main purpose here is to find the best way of mixing different intermediate products from the refinery and some additives in order to minimize the blending cost subject to meeting the quality and demand requirements of the final products. The term quality refers to meeting given product specifications. Rigby, Lasdon, and Waren (1995) discussed successful implementation of decision support systems for off-line multi-period blending problems at Texaco. Commercial applications such as Aspen Blend™ and Aspen PIMS-MBO™ from AspenTech are also available for dealing with online and offline blending optimization problems. Since these software packages are restricted to solving the blending problem, resource and temporal decisions must be made a priori either manually or by using a special method.

To solve both sub-problems simultaneously, Glismann and Gruhn (2001) proposed a two-level optimization approach where a non-linear model is used for the recipe optimization whereas a mixed-integer linear model (MILP) is utilized for the scheduling problem. The proposed decomposition technique for the entire optimization problem is based on solving first the non-linear model aiming at generating the optimal solution of the blending problem, which is then incorporated into the MILP scheduling model as fixed decisions for optimizing only resource and temporal aspects. In this way, the solution of a large MINLP model is replaced by sequential NLP and MILP models. Jia and Ierapetritou (2003) proposed a solution strategy based on decomposing the overall refinery problem in three subsystems: (a) the crude-oil unloading and blending, (b) the production unit operations, and (c) the product blending and lifting (see Fig. 1). The first sub-problem involves the crude oil unloading from vessels, its transfer to storage tanks and the charging schedule for each crude oil mixture to the distillation units. The second sub-problem consists of the production unit scheduling, which includes both fractionation and reaction processes. Reactions sections alter the molecular structure of hydrocarbons, in general to improve octane number, whereas fractionation sections separate the reactor effluent into streams of different properties and values. Lastly, the third sub-problem is related to the scheduling, blending, storage and lifting of final products. In order to solve each one of these sub-problems in the most efficient way, a set of mixed-integer linear models (MILPs) were developed, which take into account the main features and difficulties of each case. In particular, fixed product recipes were assumed in the third sub-problem, which means that blending decisions were not incorporated into this model. The MILP for-

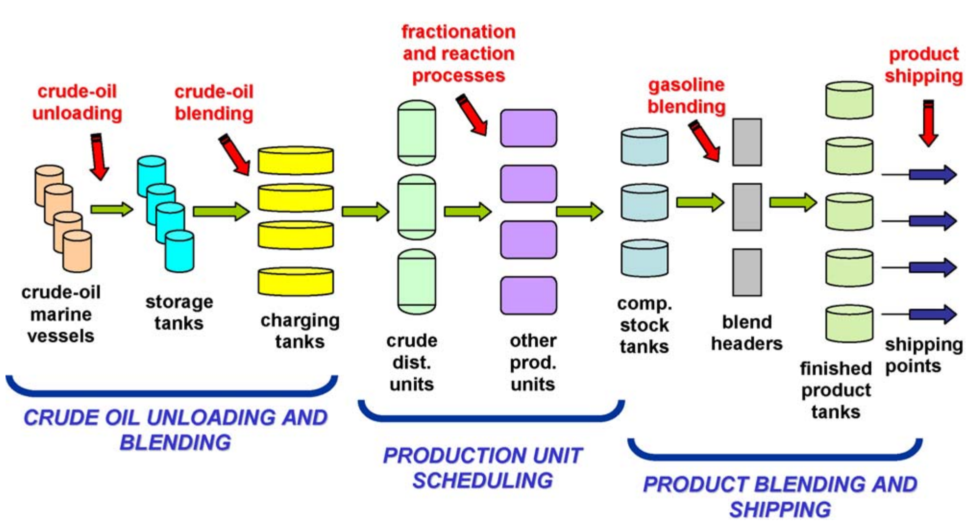

Fig. 1. Illustration of a standard refinery system.

mulation was based on a continuous time representation and on the notion of event points. The mathematical formulation proposed to solve each sub-problem involves material balance constraints, capacity constraints, sequence constraints, assignment constraints, demand constraints, and a specific objective function. Continuous variables are defined to represent flowrates as well as starting and ending times of processing tasks. Binary variables are principally related to assignment decisions of tasks to event points, or to some specific aspect of each sub-problem.

From the above review it can be seen that a variety of mathematical programming approaches are currently available to the short-term blending and scheduling problem. However, in order to reduce the inherent problem difficulty, most of them rely on special assumptions that generally make the solution inefficient or unrealistic for real world cases. Some of the common assumptions are: (a) fixed recipes for different product grades are predefined, (b) component and product flowrates are known and constant, and (c) all product properties are assumed to be linear. On the other hand, more general mixed-integer non-linear programming (MINLP) formulations are capable of considering the majority of the problem features. However, as pointed out by several authors solving logistics and quality aspects for large-scale problems is not possible in a reasonable time with current mixed-integer non-linear programming (MINLP) codes and global optimization techniques (Jia and Ierapetritou, 2003; Kelly and Mann, 2003a,b). The major issue here is related to non-linear and non-convex constraints with which the computational performance strongly depends on the initial values and bounds assigned to the model variables. Taking into account the major weaknesses of the available mathematical approaches, the major goal of this work is to develop a novel iterative mixed-integer linear programming (MILP) formulation for the simultaneous gasoline short-term blending and scheduling problem of oil-refinery operations, which is generally agreed as being the most important and complex subproblem. Its importance comes from the fact that gasoline can yield 60–70% of total refinery's profit. On the other hand, the complexity arises from the large number of product demands and quality specifications for each final product, as well as the limited number of available resources that can be used to reach the production goals. Non-linear property specifications based on variable and preferred product recipes are effectively handled through the proposed iterative linear procedure, which allows the model to generate optimal, or near-optimal solutions with modest computational effort.

2. Problem statement

The gasoline short-term blending and scheduling problem takes into account two major issues. The first one is related to aspects of production logistics, which mainly involves multiple production demands with different due dates, inventory pumping constraints for products and components, as well as different logistic and operating rules. Most of these features are part of typical scheduling problems and are usually modeled as discrete and continuous decisions in an optimization framework. On the other hand, the second issue is the production quality, which represents an additional difficulty for standard scheduling problems. This second issue is also known as the off-line blending problem and takes into account variable product recipes and property specifications such as minimum octane number, maximum sulfur and aromatic content, etc. The main objective is to produce on-spec blends at minimum cost, where product specifications are stringent and constantly changing in most of the markets. Product qualities are usually predicted through complex correlations that depend on the concentration and the properties of the components used in the blend. Depending on the product property, non-linear correlations may include linear, bilinear, trilinear and exponential terms. Some of these non-linear terms can sometimes be linearized (see Adams & Sherali, 1990; Oral & Kettani, 1992). The general process topology corresponds to a multi-stage system composed of component storage tanks, blend headers and product storage tanks. Specifically, we assume that we are given the following items:

(1) A predefined scheduling horizon, typically 7–10 days.
(2) A set of intermediate products from the refinery (components).
(3) A set of dedicated storage tanks with minimum and maximum capacity restrictions.
(4) Initial stocks for components.
(5) Component supplies with known flowrates.
(6) Properties or qualities for components.
(7) Minimum and maximum flowrates between component tanks and blend headers.
(8) A set of final products with predefined minimum and maximum quality specifications.
(9) A set of equivalent blend headers working in parallel that can be allocated to each final product.
(10) A set of correlations, mostly non-linear, for predicting the values of properties of each blend.
(11) Minimum and maximum component concentrations in final products.
(12) Preferred product recipes.

The goal is to determine:

(a) the assignment of blenders to final products;
(b) the inventory levels of components and products in storage tanks;
(c) the volume fraction of components included in each product;
(d) the total volume of each product;
(e) the pumping rates for components and products;
(f) the optimal timing decisions for production and storage tasks.

The objective is to maximize the production profit while satisfying the process, operations and maintenance constraints, final product demands and quality specifications. The objective function includes the total product value, the raw material cost, inventory cost and penalties for deviation from preferred recipes. Additional terms involving slack variables for handling infeasible solutions can also be incorporated into the objective function to provide effective solutions for all circumstances. The next section describes a simultaneous optimization approach for the blending and scheduling operations involved in this problem.

3. Proposed optimization approach

The main features of the proposed approach can be summarized as follows:

- A multi-period optimization model is used that is able to deal with multiple product demands with different due dates and quality specifications.
- Discrete or continuous time domain representations can be used, depending on the problem characteristics. The term "time slot" is used in this paper to represent a time interval with known duration and position for discrete time, and unknown duration and position for continuous time.
- Linear approximations are used together with an iterative procedure to get better predictions of all product properties, even those naturally non-linear such as the octane number.
- The production logistics and quality specifications are solved simultaneously.
- Fixed or variable product recipes are specified, as well as minimum and maximum limits on component concentration.
- Binary variables are used to represent assignment decisions, as well as any other logistic or production rule found in the problem.

In order to describe the main model variables, Fig. 2 illustrates a simple example of a gasoline blending and scheduling problem, which has traditionally been treated as two separate problems. The solution of the scheduling problem defines the way in which the products are processed with respect to time and available equipment. On the other hand, the solution of the blending problem defines how the available components are blended or mixed together to produce on-spec products with minimum cost.

The key decision variables involved in a standard problem are the following. The continuous variable $F^I_{i,p,t}$ defines the volumetric flow of component i being transferred to product p during the time slot t whereas $F^P_{p,t}$ denotes the volumetric flow of product p being blended during each time slot t. The continuous variables $V^I_{i,t}$ and $V^P_{p,t}$ define the amount of component and product being stored at each time point t, respectively. Finally, the discrete variable $A_{p,t}$ defines which products are allocated to blenders in each time slot t. Additional continuous and discrete variables can be included into the mathematical model to tackle particular problem characteristics and operating constraints.

It is worth mentioning that both the discrete and the continuous time formulations are focused on a part of the details around real-world problem logic constraints. A comprehensive enumeration of most of them can be found in Kelly and Mann (2003a,b). When some features such as inter-temporal transfer of logic are addressed (i.e. run-lengths, mixing-delays, etc.), the MILP computational effort can be significantly increased, particularly when more than 20 time-periods or time-slots are defined in the model.

4. Off-line blending problem

Before describing the proposed MILP formulation, we present in this section the main features of the iterative scheme for predicting product properties for the blending problem.

A significant number of gasoline properties can be directly computed by using a volumetric average as shown in Eq. (1):

$$\mathrm{PR}_{p,k,t} = \sum_i \mathrm{pr}_{i,k} v^I_{i,p,t}, \quad \forall p, k, t \tag{1}$$

where $v^I_{i,p,t}$ is the volume fraction of component i in product p in time slot t, $\mathrm{PR}_{p,k,t}$ defines the exact value of the property k for product p in time slot t and $\mathrm{pr}_{i,k}$ is the value of the property k for component i. The volume fraction variable $v^I_{i,p,t}$ is linked to the volumetric flowrate variables $F^I_{i,p,t}$ *and* $F^P_{p,t}$ through the non-linear equality (2):

$$v^I_{i,p,t} F^P_{p,t} = F^I_{i,p,t}, \quad \forall p, k, t \tag{2}$$

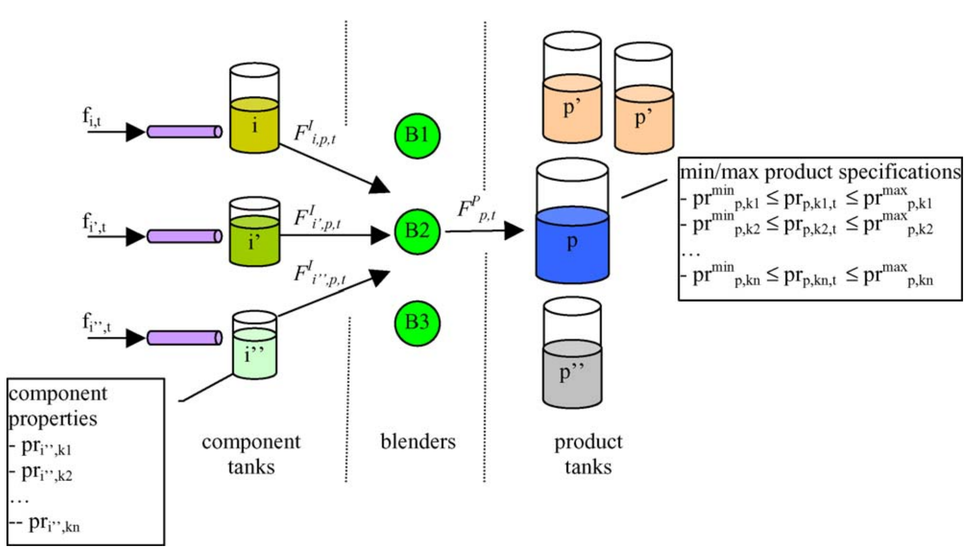

Fig. 2. Illustration of the meaning of the principal model variables.

Taking into account that volumetric flowrate variables are required to control inventory levels in tanks and volume fraction variables are needed to predict product properties, the general mathematical model for the integrated blending and scheduling problem comprises a set of constraints with bilinear terms, even if only linear product properties are considered. In order to preserve the linearity of the model, the original equality (1) can be expressed in an alternative way by multiplying it by $F^P_{p,t}$:

$$\mathrm{PR}_{p,k,t}F^P_{p,t} = \sum_i \mathrm{pr}_{i,k} v^I_{i,p,t} F^P_{p,t}, \quad \forall p, k, t \tag{3}$$

Then, equality (2) can be incorporated into Eq. (3), yielding Eq. (4):

$$\mathrm{PR}_{p,k,t}F^P_{p,t} = \sum_i \mathrm{pr}_{i,k} F^I_{i,p,t}, \quad \forall p, k, t \tag{4}$$

Subsequently, taking advantage of minimum and maximum product property specifications, constraint (4) can be replaced by constraint (5), in which the variable $\mathrm{PR}_{p,k,t}$ is substituted by their respective minimum and maximum property values ($\mathrm{pr}^{\min}_{p,k}$; $\mathrm{pr}^{\max}_{p,k}$), which are known problem data:

$$\mathrm{pr}^{\min}_{p,k} F^P_{p,t} \le \sum_i \mathrm{pr}_{i,k} F^I_{i,p,t} \le \mathrm{pr}^{\max}_{p,k} F^P_{p,t}, \quad \forall p, k, t \tag{5}$$

In this way, the variable $v^I_{i,p,t}$ is no longer required and the model remains linear. This linearization is valid only if properties computed volumetrically are considered in the blending problem. However, other gasoline properties can be approximated by adding minor changes to the previous equation. For instance, if the correlation for predicting a particular product property is based on a linear volumetric average plus additional non-linear terms, such as the case of the octane number, the non-linear part of the equation can be removed and replaced by a correction factor $\mathrm{bias}_{p,k,t}$, as shown in Eq. (6):

$$\mathrm{pr}^{\min}_{p,k} F^P_{p,t} + \mathrm{bias}_{p,k,t} F^P_{p,t} \le \sum_i \mathrm{pr}_{i,k} F^I_{i,p,t} \le \mathrm{pr}^{\max}_{p,k} F^P_{p,t} + \mathrm{bias}_{p,k,t} F^P_{p,t}, \quad \forall p, k, t \tag{6}$$

Thus, non-linear product properties can be approximated through the linear equation (6), which is composed of a volumetric average followed by a correction factor 'bias'. As can be seen, this correction factor depends on the product, property and time slot, and it is iteratively calculated by using the procedure described below.

On the other hand, it is worth mentioning that some product properties such as oxygen and sulfur content are blended gravimetrically, which means that component and product specific gravities are also taken into account for the prediction, as shown in Eq. (7). In this case, sg_i and $\mathrm{sg}_{p,t}$ define the specific gravity of component i and product p in time slot t, respectively. Given that $\mathrm{sg}_{p,t}$ is a model variable that is not directly computed through the proposed linear approach and with the intention of maintaining the model's linearity, the exact value of $\mathrm{sg}_{p,t}$ can be substituted by an approximated specific gravity $\mathrm{sgrav}_{p,t}$, which can be easily computed through the iterative procedure described below

$$\mathrm{PR}_{p,k,t} = \frac{\sum_i \mathrm{pr}_{i,k} \mathrm{sg}_i V^I_{i,p,t}}{\mathrm{sg}_{p,t}}, \quad \forall p, k, t \tag{7}$$

Therefore, the proposed linear approximation for gravimetric blending is as follows:

$$\mathrm{pr}^{\min}_{p,k} F^P_{p,t} \le \frac{\sum_i \mathrm{pr}_{i,k} \mathrm{sg}_i F^I_{i,p,t}}{\mathrm{sgrav}_{p,t}} \le \mathrm{pr}^{\max}_{p,k} F^P_{p,t}, \quad \forall p, k, t \tag{8}$$

We want to point out that there are some complex blending decisions (especially in crude-oil blending) that may require to explicitly define yield variables for streams within streams (see Kelly, 2004; Kelly & Mann, 2003a,b). For example, there may be the requirement to blend to a distillate diesel sulfur quality specification. Therefore, we need the diesel yield as well as the diesel specific gravity in each component blending stream to be calculated endogenously. This is a higher-degree multi-linear problem that is not addressed in this paper.

To illustrate the use of the iterative procedure and the proposed linear approximation, Fig. 3 shows a comparison between the values of the linear volumetric average, the non-linear original correlation and the proposed linear approximation for a real non-linear product property such as the motor octane number. In this example, the blend of two components, A and B, is only considered. The final product property is a non-linear function of component concentrations. As shown in Fig. 3, if 40% of component A is blended with 60% of component B, the values of the volumetric average and the real non-linear correlation are 88.5 and 88.74, respectively. This difference arises because all non-linear terms involved in the exact motor octane correlation are not included in the linear volumetric average. In order to correct this discrepancy, the correction factor bias is calculated and used to yield a better property prediction in the next iteration. For this specific mixture of components the correction factor bias is equal to 0.24. The linear approximation comprising the volumetric average together with the correction factor bias will always predict the exact value of the property if the same component concentration is utilized in the next iteration. Furthermore,

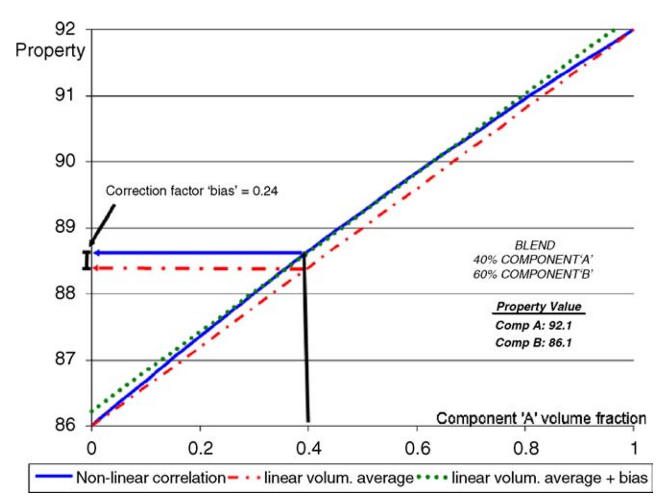

Fig. 3. A non-linear property and the proposed linear approximation.

it was observed that the proposed linear approximation tends to predict a very close value of the real property if component concentrations are not significantly changed in the next iteration as shown in Fig. 3.

The proposed iterative procedure to solve simultaneously the blending and scheduling problem using only linear equations is shown in Fig. 4. The first step is to find an initial recipe for all products. If preferred product recipes are known they can be proposed as initial product recipes. Preferred recipes are the best alternative for blending because they satisfy all product specifications with minimum cost. However, the use of them strongly depends on the scheduling decisions, component inventories and product demands and for this reason, they should not be treated as fixed mixtures in a blending tool. On the other hand, if preferred recipes are not defined, one possibility for generating initial recipes is to solve the LP model including only linear product properties. Once initial recipes are generated, they provide the component volume fractions used in each blend, which can then be employed as fixed parameters in more realistic non-linear correlations. The value predicted by the non-linear correlation and the linear volumetric average are both used to calculate the correction factor 'bias' (see Fig. 3). Given that we are dealing with a multi-period optimization problem, the correction factor will be calculated for all non-linear properties, products and time intervals as the difference between the value predicted by the original non-linear equation and the linear volumetric average. The specific gravity of each product and time slot is also computed. After that, the LP model is solved that includes linear approximation with the parameter $\text{bias}_{p,k,t}$, for volumetric properties and the parameter $\text{sgrav}_{p,k,t}$, for gravimetric properties. The parameter bias will be equal to zero for all linear properties that can be computed volumetrically. For non-linear properties this parameter will converge to a non-zero value that reflects the difference with the linear approximation. Subsequently, the solution of this problem is updated and the product recipes for those products meeting all specifications in a specific time interval are fixed. If different recipes are used for the same product in different time intervals, only those that are feasible will be fixed. This process is repeated until all product recipes meet the product specifications, i.e. all product recipes are fixed. The main objective of this iterative procedure is to progressively find feasible recipes for all products while optimizing all temporal and resource constraints in the scheduling problem. The proposed method can be conceptually interpreted as a successive LP method for the blending problem or a successive MILP model for the simultaneous blending and scheduling problem. In Appendix A, we provide sufficient conditions under which the successive LP method is guaranteed to converge to a local solution of the blending problem. These conditions are rigorous for the case that a specification of a non-linear property does not become active at the solution. When that is not the case, the proof requires the assumption that

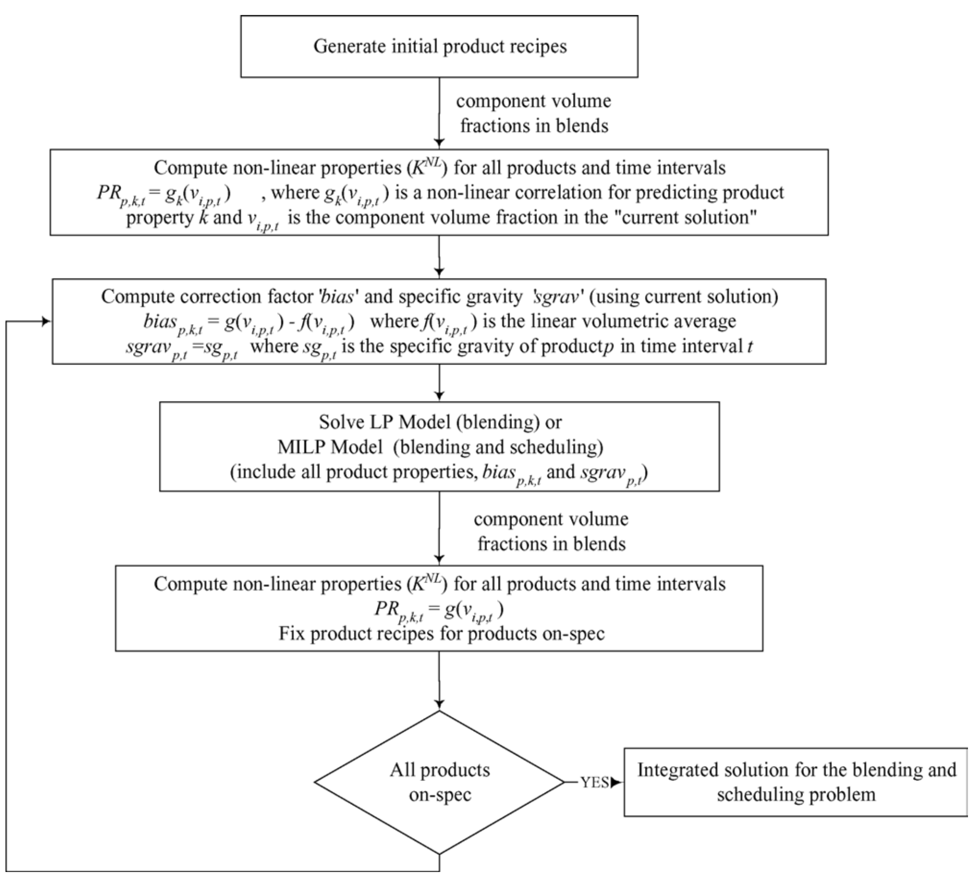

Fig. 4. Proposed iterative approach for simultaneous blending and scheduling.

the non-linear properties are a weak function of the compositions, which of course is a strong assumption. However, as it will be shown later in the paper, only few iterations are needed to obtain a very good solution for the blending and scheduling problem. Furthermore, in the section of computational results we present a comparison with NLP and MINLP methods in which they confirm the solutions obtained with the proposed method, but at either considerable higher computational cost, or in some cases failing to converge depending on the starting point used. Thus, the proposed method is fast and reliable, providing very good solutions, which is particularly relevant for industrial applications.

5. Integrated blending and scheduling model

A central aspect of any scheduling model is related to timing decisions. Mathematical formulations can be based on either a discrete or continuous time domain representation. The discrete time representation only allows processing tasks to take place at certain time points, which correspond to the boundaries of a set of predefined time slots. The main advantage of using a discrete time grid is that mass balance and inventory constraints are easier to handle but at the same time the solution loses flexibility, unless smaller time intervals are used, which may significantly decrease the computational performance of the method. In contrast, continuous time representations are capable of generating more flexible solutions in terms of timing decisions, although with higher CPU time requirements. Also, inventory and mass balance constraints are generally more difficult to model since they have to be checked at any time during the scheduling horizon in order to ensure that a feasible solution will be generated. Since the best choice of the time representation strongly depends on the problem characteristics and the desired solution quality, we developed a mathematical formulation for each type of representation assuming a common time grid for all resources working in parallel. Before presenting the proposed mathematical models the nomenclature is as follows:

Indices

d	due dates of product demands
i	intermediates or components
k	properties or qualities
p	final products or gasoline grades
t	time slots

Sets

D	set of product due dates
I	set of intermediates to be blended
K	set of properties for intermediates and products
K_{NL}	set of properties that are predicted with non-linear correlations
P	set of final products
T	set of time slots
T_d	set of time slots postulated for the sub-interval ending at due date d (continuous time)

Parameters

$\mathrm{bias}_{p,k,t}$	correction factor of the value of property k of product p in time slot t
c_i	cost of component i
d	demand due date
$\mathrm{dd}_{p,d}$	demand of product p to be satisfied at due date d
e_t	predefined ending time of time slot t (discrete time representation)
f_i	constant flowrate of component i
h	time horizon
ini_i	initial inventory of component i
ini_p	initial inventory of product p
$l_p^{\min}$	minimum time slot duration when it is allocated to product p
n_t^{B}	maximum number of blenders that can be working in parallel in time slot t
$\mathrm{plty}_{i,p}^{\mathrm{R}+}$	penalty cost for excess of component i in product p
$\mathrm{plty}_{i,p}^{\mathrm{R}-}$	penalty cost for shortage of component i in product p
$\mathrm{plty}_{k,p}^{\mathrm{S}+}$	penalty cost for a deviation from the minimum specification for property k in product p
$\mathrm{plty}_{k,p}^{\mathrm{S}-}$	penalty cost for a deviation from the maximum specification for property k in product p
$\mathrm{plty}_{i}^{\mathrm{SH}}$	penalty cost for purchasing component i from third-party
p_p	price of product p
$\mathrm{pr}_{i,k}$	value of property k for component i
$\mathrm{pr}_{p,k}^{\max}$	maximum value of property k for product p
$\mathrm{pr}_{p,k}^{\min}$	minimum value of property k for product p
$\mathrm{rate}_p^{\max}$	maximum flowrate of product p
$\mathrm{rate}_p^{\min}$	minimum flowrate of product p
$\mathrm{rcp}_{i,p}$	preferred concentration of component i in product p according to product recipe
$\mathrm{rcp}_{i,p}^{\max}$	maximum concentration of component i in product p
$\mathrm{rcp}_{i,p}^{\min}$	minimum concentration of component i in product p
$\mathrm{sgrav}_{p,t}$	specific gravity of product p in time slot t
s_t	predefined starting time of time slot t (discrete time representation)
sp_i	penalty cost for inventory of component i
sp_p	penalty cost for inventory of product p
$V_i^{\max}$	maximum storage capacity of component i
$V_i^{\min}$	minimum storage capacity of component i
$V_p^{\max}$	maximum storage capacity of product p
$V_p^{\min}$	minimum storage capacity of product p

Variables

$A_{p,t}$	binary variable denoting that product p is blended in time slot t
$D_{i,p,t}^{\mathrm{R}-}$	shortage of component i that is used for product p in time slot t according to the preferred product recipe
$D_{i,p,t}^{\mathrm{R}+}$	excess of component i that is used for product p in time slot t according to the preferred product recipe
$D_{k,p,t}^{\mathrm{S}-}$	deviation from the minimum specification of property k for product p in time slot t
$D_{k,p,t}^{\mathrm{S}+}$	deviation from the maximum specification of property k for product p in time slot t

E_t ending time of time slot t (continuous time representation)
$F^I_{i,p,t}$ amount of component i being transferred to product p during time slot t
$F^P_{p,t}$ amount of product p being blended during time slot t
$\mathrm{PR}_{p,k,t}$ exact value of the property k for product p in time t
$S_{i,t}$ amount of component i to be purchased in time slot t
S_t starting time of time slot t (continuous time representation)
$v^I_{i,p,t}$ volume fraction of component i in product p at time t
$V^I_{i,t}$ amount of component i stored at the end of time slot t
$V'^I_{i,t}$ amount of component i stored at the beginning of time slot t
$V'^P_{p,d}$ amount of product p stored at due date d
$V^P_{p,t}$ amount of product p stored at the end of time slot t

6. Discrete time representation

In this section, we present an MILP model that assumes that the entire scheduling horizon is divided into a finite number of consecutive time slots that are common for all units and can be allocated to different products, i.e. blending tasks. The proposed model has the following features:

1. A discrete time domain representation is used where the scheduling horizon is divided into a set of consecutive time slots.
2. Equivalent blenders working in parallel are available for different product grades.
3. A particular product demand can be satisfied by one or more time slots whenever they are allocated to this product and finished before the product due date.
4. Variable product recipes are considered and product properties are predicted by linear correlations.
5. Constant flowrate of components is assumed during the entire scheduling horizon.
6. Constant flowrate of products is assumed during the allocated time slot.

Model constraints and variables are introduced below.

6.1. Assignment constraint

Constraint (9) defines through the binary variables $A_{p,t}$ the final products p to be processed in time slot t. Given that a set of equivalent blenders are available to produce different gasoline grades simultaneously, n^{B}_t specifies the maximum number of units that can be working in parallel during time slot t:

$$\sum_p A_{p,t} \le n^{\mathrm{B}}_t, \quad \forall t \tag{9}$$

6.2. Product composition constraint

Every final product or gasoline grade p is a blend of different components i, as expressed by constraint (10):

$$\sum_i F^I_{i,p,t} = F^P_{p,t}, \quad \forall p, t \tag{10}$$

Note that a significant reduction in the number of continuous variables can be obtained if Eq. (10) is deleted from the model and $F^P_{p,t}$ is replaced by $\sum_i F^I_{i,p,t}$. However, in order to make the model easier to understand, $F^P_{p,t}$ has been included in all model equations.

6.3. Minimum/maximum component concentration

In order to satisfy product qualities and/or market conditions, upper and lower bounds can be forced on the component concentration for specific gasoline grades. Then, constraint (11) ensures that product composition will always satisfy the predefined component specifications. Parameters $\mathrm{rcp}^{\min}_{i,p}$ and $\mathrm{rcp}^{\max}_{i,p}$ define the minimum/maximum concentration of component i for product p, respectively:

$$\mathrm{rcp}^{\min}_{i,p} F^P_{p,t} \le F^I_{i,p,t} \le \mathrm{rcp}^{\max}_{i,p} F^P_{p,t}, \quad \forall i, p, t \tag{11}$$

It should be noted that a fixed recipe for a particular product p can also be taken into consideration by fixing the values of $\mathrm{rcp}^{\min}_{i,p}$ and $\mathrm{rcp}^{\max}_{i,p}$ to the predefined concentration of component i for product p. However, the use of fixed recipes should be avoided unless they are the only possibility to produce a particular product. As a better option, preferred recipes can be proposed as an initial solution of the proposed iterative procedure. In this way, the generation of infeasible solutions will be avoided.

6.4. Minimum/maximum volumetric flowrates for products

Constraint (12) specifies that minimum and maximum volumetric flowrates must be satisfied when product p is blended during time slot t. Due to the fact that a constant product flowrate is assumed in this work, the volumetric flowrate can be computed by multiplying the upper and lower flowrates by the time slot duration whenever product p is allocated to a particular time slot t ($A_{p,t} = 1$). Moreover, since a discrete time representation is used, the time slot duration is a known parameter computed through the predefined starting s_t and ending times e_t of each time slot t. It should be noted that if product p is not processed during time slot t ($A_{p,t} = 0$), the volumetric flowrate will be also equal to zero:

$$\mathrm{rate}^{\min}_p (e_t - s_t) A_{p,t} \le F^P_{p,t} \le \mathrm{rate}^{\max}_p (e_t - s_t) A_{p,t}, \quad \forall p, t \tag{12}$$

6.5. Material balance equation for components

Given that a discrete time representation allows the blending tasks to start and finish at the boundaries of the time slot allocated, inventory limits have only to be checked at the end of each time slot. Then, as expressed by constraint (13), the amount of component i being stored in tank at the end of time slot t is equal to the initial inventory of component i plus the component produced up to the end of time slot t minus the component transferred to blenders up to the end of time slot t:

$$V^I_{i,t} = \text{ini}_i + f_i e_t - \sum_{p,t' \le t} F^I_{i,p,t'}, \quad \forall i, t \tag{13}$$

where ini_i is the initial inventory of component i at time $t = 0$, the parameter f_i specifies the constant production rate of component i and e_t defines the ending time of time slot t. Given that a discrete time representation is used, both parameters are known in advance.

6.6. Component storage capacity

Constraint (14) imposes lower/upper bounds $V^{\min}_i$ and $V^{\max}_i$ on the total amount of component i being stored in a storage tank during the scheduling horizon. Given that constant component flowrates are assumed, a perfect coordination between the production of components and final products is required to satisfy the storage constraints through the entire scheduling horizon:

$$V^{\min}_i \le V^I_{i,t} \le V^{\max}_i, \quad \forall i, t \tag{14}$$

6.7. Material balance equation for products

Constraint (15) computes the amount of product p being stored in tank at the end of time slot t taking into account the initial inventory, production and demands of product p:

$$V^P_{p,t} = \text{ini}_p + \sum_{t' \le t} F^P_{p,t'} - \sum_{d \le t} \text{dd}_{p,d}, \quad \forall p, t \tag{15}$$

6.8. Product storage capacity

A minimum safety stock and a finite storage capacity is assumed for final products:

$$V^{\min}_p \le V^P_{p,t} \le V^{\max}_p, \quad \forall p, t \tag{16}$$

6.9. Minimum/maximum product qualities

Assuming that properties are volumetrically computed, constraint (17) guarantees that the value of property k for product p in time slot t will always satisfy minimum and maximum product specifications. To maintain the model's linearity, property k is not directly computed and bounds are only imposed on each property. Otherwise, non-convex bilinear equations would be generated in the model, which would then become non-linear. Although this linearization is only valid for properties volumetrically computed, the original equation (17) can be slightly modified as Eq. (17′) to account for real-world product properties, as described in Section 4 with the use of the parameter $\text{bias}_{p,k,t}$. The best value of this parameter can be obtained through the proposed iterative procedure. In this way, the MILP mathematical model is able to effectively deal with the quality issue, including variable recipes and non-linear properties:

$$\text{pr}^{\min}_{p,k} F^P_{p,t} \le \sum_i \text{pr}_{i,k} F^I_{i,p,t} \le \text{pr}^{\max}_{p,k} F^P_{p,t}, \quad \forall p, k, t \tag{17}$$

$$\text{pr}^{\min}_{p,k} F^P_{p,t} \le \sum_i \text{pr}_{i,k} F^I_{i,p,t} + \text{bias}_{k,p,t} F^I_{i,p,t} \le \text{pr}^{\max}_{p,k} F^P_{p,t}, \quad \forall p, k, t \tag{17′}$$

In turn, Eq. (17″) defines the proposed linear approximation for those product properties that are gravimetrically predicted:

$$\text{pr}^{\min}_{p,k} F^P_{p,t} \le \frac{\sum_i \text{pr}_{i,k} \rho i F^I_{i,p,t}}{\text{sgrav}_{p,t}} \le \text{pr}^{\max}_{p,k} F^P_{p,t}, \quad \forall p, k, t \tag{17″}$$

Note that constraints (17), (17′) and (17″) are only required for those gasoline grades that can be produced using variable recipes. If a fixed recipe is enforced, product properties must be satisfied in advance through the predefined component concentrations.

6.10. Multiple product demands

Refinery operations typically require that multiple demands for the same gasoline grade be satisfied during the entire scheduling horizon. Constraint (18) guarantees that a sufficient amount of product p will always be available to satisfy each product demand $\text{dd}_{p,d}$:

$$\text{ini}_p + \sum_{t \le d} F^P_{p,t} \ge \sum_{d' \le d} \text{dd}_{p,d'}, \quad \forall p, d \tag{18}$$

6.11. Objective function (maximize net profit)

While satisfying all quality and logistic issues, the main objective of the scheduling problem is to maximize the net profit defined as the total product value minus the total component cost:

$$\max \sum_t \sum_p \left(p_p F^P_{p,t} - \sum_i c_i F^I_{i,p,t} \right) \tag{19}$$

The formulation can also accommodate alternative objective functions. An example is Eq. (20), where penalties related to component and product inventories has been included in order to also reduce storage costs:

$$\max \sum_t \sum_p \left(p_p F^P_{p,t} - \sum_i c_i F^I_{i,p,t} \right) - \sum_p \sum_t \text{sp}_p V^P_{p,t} - \sum_i \sum_t \text{sp}_i V^I_{i,t} \tag{20}$$

7. Continuous time representation

The model introduced in the previous section relies on a discrete time domain representation. As an alternative option, the continuous time formulation that is presented in this section aims at generating more flexible schedules capable of maximizing the plant performance without significantly increasing the model size. However, special attention must be paid to the limited storage capacity since continuous time representation tends to make the modeling of inventory constraints more difficult. The main idea here is first to partition the entire time horizon into a predefined number of sub-intervals. The length of each sub-interval will depend on the product due dates. For instance, the first sub-interval will start at the beginning of the scheduling horizon and finish at the first product due date. The second one will be extended from the first up to the second product due date. A similar idea is applied to the next sub-intervals. Therefore, the number of sub-intervals will be equal to the number of product due dates and the starting and ending time of each one will be known in advance.

Once the sub-intervals are defined, a set of time slots with unknown duration are postulated for each one. The number of time slots for each sub-interval will depend on the sub-interval length as well as the grade of flexibility desired for the solution. Variable starting and ending times of time slots are introduced as new model continuous variables that allow the production events to happen at any time during the scheduling horizon. Fig. 5 illustrates the main features of the proposed continuous time domain representation. In this case, four product demands with different due dates are to be satisfied, which means that four sub-intervals are predefined. Then, nine time slots can be postulated for the entire scheduling horizon, where two time slots are defined for each one of the first three sub-intervals whereas three are postulated to the last one.

The proposed model has the following features:

1. A continuous time domain representation is used where the scheduling horizon is divided into sub-intervals and a set of time slots with unknown duration and position are postulated for each one.
2. Equivalent blenders working in parallel are available for different product grades
3. A particular product demand can be satisfied by one or more time slots whenever they are allocated to this product and finished before product due date.
4. Final product properties are based on a volumetric average and a correction factor computed through the proposed iterative process.
5. A constant flowrate of components is assumed during the entire scheduling horizon.

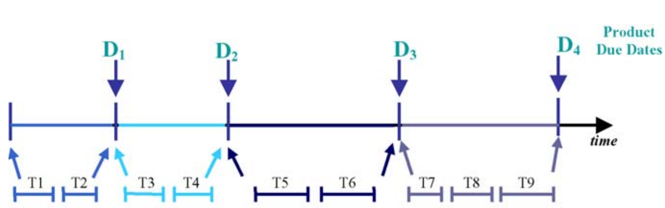

Fig. 5. Proposed continuous time representation.

6. A constant flowrate of product is assumed during the allocated time slot.

When the mathematical model is based on a continuous time domain representation, starting and ending times for the time slots are new continuous decision variables. For that reason, part of the original constraints used for the discrete time representation must be updated in order to maintain the linearity of the model as well as to account new problem features. In this section we describe the set of constraints that must be modified as well as the new ones to be added. Constraints that are not required to change must be included into the model in the same way they were presented in the previous section, such as Eqs. (9)–(11), (14)–(17), (17′), (17″) and (19).

7.1. Minimum/maximum volumetric flowrates for products

Constraints (12′) and (12″) replace original constraint (12) when a continuous time representation is used. When product p is not allocated to time slot t, the binary variable $A_{p,t}$ is equal to zero and constraint (12″) enforces the variable $F^P_{p,t}$ to be equal to zero as well. On the other hand, $A_{p,t}$ will be equal to one whenever product p is processed during time slot t. In this case, constraint (12″) becomes redundant and constraint (12′) imposes minimum and maximum volumetric flowrates depending on the time slot duration:

$$\text{rate}_p^{\min}(E_t - S_t) - \text{rate}_p^{\min} h(1 - A_{p,t}) \le F^P_{p,t} \le \text{rate}_p^{\max}(E_t - S_t), \quad \forall p, t \qquad (12')$$

$$F^P_{p,t} \le \text{rate}_p^{\max} h A_{p,t}, \quad \forall p, t \qquad (12'')$$

7.2. Material balance equation for components

To ensure that only feasible solutions are generated, the amount of component stored in tank has to be checked not only at the end but also at the beginning of each time slot. To make this possible, a new variable $V'^I_{i,t}$ is included into the model and the original equation (13) is replaced by constraints (13′) and (13″). The same idea for computing the inventory of components is applied to these new constraint:

$$V^I_{i,t} = \text{ini}_i + f_i E_t - \sum_{p,t' \le t} F^I_{i,p,t'}, \quad \forall i, t \qquad (13')$$

$$V'^I_{i,t} = \text{ini}_i + f_i S_t - \sum_{p,t' < t} F^I_{i,p,t'}, \quad \forall i, t \qquad (13'')$$

Note that despite the fact that E_t and S_t are model variables, both constraints remain linear because a constant production rate f_i is assumed for the components.

7.3. Component storage capacity

An additional constraint (14′) is required to impose lower/upper bounds $V_i^{\min}$ and $V_i^{\max}$ on the total amount of component i being stored in tank at the beginning of time slot t:

$$V_i^{\min} \leq V'^{I}_{i,t} \leq V_i^{\max}, \quad \forall i, t \tag{14'}$$

7.4. Material balance equation for products

Constraint (15′) computes the inventory of product p available right after satisfying the demand of product p taking place at due date d. In this way, a minimum product safety stock can be guaranteed at any time during the scheduling horizon, even after a product lifting is carried out:

$$V'^{P}_{p,d} = \text{ini}_p + \sum_{t<d} F^P_{p,t} - \sum_{d' \leq d} \text{dd}_{p,d'}, \quad \forall p, d \tag{15'}$$

Constraint (16′) explicitly defines the lower bound on the new inventory variable:

$$V_p^{\min} \leq V'^{P}_{p,d}, \quad \forall p, d \tag{16'}$$

7.5. Set of time slot timing constraints

Instead of defining starting and ending times of time slots as fixed parameters, in continuous time representation models these decisions are treated as additional continuous variables to be optimized. In order to allow more flexible solutions and avoid overlapping time slots, a correct order and sequence between postulated time slots must be established through the next set of constraints.

7.6. Time slot duration

Constraint (21) defines a minimum time slot duration when product p is allocated to time slot t. It is generally used to model an existing operating condition, but at the same time permits eliminating schedules using very short time slots, which are usually inefficient in practice:

$$E_t - S_t \geq l_p^{\min} A_{p,t}, \quad \forall p, t \tag{21}$$

To ensure that duration of a slot is zero if it is not used, Eq. (22) is included into the model:

$$E_t - S_t \leq h \sum_p A_{p,t}, \quad \forall t \tag{22}$$

7.7. Time slot sequencing

Constraint (23) establishes a sequence between consecutive time slots t and $t+1$:

$$E_t \leq S_{(t+1)}, \quad \forall t \tag{23}$$

7.8. Sub-interval bounds

The set T_d comprises all time slots that are postulated for a sub-interval related to a particular due date d. This sub-interval begins at the previous due date $d-1$ and finishes at due date d. Constraint (24) defines that time slots pre-allocated to this sub-interval must start after due date $d-1$ whereas constraints (25) imposes that they must end before due date d. The main goal of this assumption is that neither additional variables nor new constraints are required to establish which time slots can satisfy a specific product demand. As a result, more flexible schedules can be obtained without increasing the complexity of inventory constraints:

$$S_t \geq d-1, \quad \forall t \in T_d \tag{24}$$

$$E_t \leq d, \quad \forall t \in T_d \tag{25}$$

7.9. Time slot assigment

Constraint (26) imposes an order for using the set of predefined time slots. In other words, a time slot $t+1$ can be only allocated to a product p whenever the previous time slot within the same sub-interval has been used:

$$\sum_p A_{p,(t+1)} \leq n_t^{\text{B}} \sum_p A_{p,t}, \quad \forall d, (t, t+1) \in T_d \tag{26}$$

8. Treatment of infeasible solutions

The short-term blending and scheduling of oil-refinery operations is a very complex and highly constrained problem, where even feasible solutions may be difficult to generate in some circumstances. For that reason, in this section we present an additional set of variables and equations that defines penalties that can be added to the objective function of the proposed model. These penalties can partially relax some hard problem specifications that can generate infeasible solutions when real world problems are addressed.

8.1. Penalty for preferred recipe deviation

If a preferred combination of components is defined for a particular product through the parameter $\text{rcp}_{i,p}$, the following constraints can be included in the model to try to use the desired recipe whenever it is possible:

$$\text{rcp}_{i,p} F^P_{p,t} + D^{\text{R}^-}_{i,p,t} \geq F_{i,p,t}, \quad \forall i, p, t \tag{27}$$

$$\text{rcp}_{i,p} F^P_{p,t} - D^{\text{R}^+}_{i,p,t} \leq F_{i,p,t}, \quad \forall i, p, t \tag{28}$$

where $D^{\text{R}^+}_{i,p,t}$ and where $D^{\text{R}^-}_{i,p,t}$ define the excess and the shortage of component i that is used in product p in time slot t, according to the preferred product recipe. Constraint (29) penalizes the slack variables $D^{\text{R}^+}_{i,p,t}$ and $D^{\text{R}^-}_{i,p,t}$ in the objective to ensure that deviations from the preferred recipe are minimized:

$$\text{penalty} = \sum_t \sum_p \sum_i (\text{plty}^{\text{R}^+}_{i,p} D^{\text{R}^+}_{i,p,t} + \text{plty}^{\text{R}^-}_{i,p} D^{\text{R}^-}_{i,p,t}) \tag{29}$$

8.2. Penalty for minimum/maximum specification deviation

If desired product qualities cannot be fully achieved, and at the same time, they can partially be violated for certain products, the following constraints can be used in order to minimize the deviation:

$$\text{prop}^{\min}_{p,k} F^{P}_{p,t} - D^{S+}_{k,p,t} \leq \sum_{i} \text{pr}_{i,k} F^{I}_{i,p,t}, \quad \forall p, k, t \tag{30}$$

$$\text{prop}^{\max}_{p,k} F^{P}_{p,t} + D^{S-}_{k,p,t} \geq \sum_{i} \text{pr}_{i,k} F^{I}_{i,p,t}, \quad \forall p, k, t \tag{31}$$

where the continuous variables $D^{S+}_{k,p,t}$ and $D^{S-}_{k,p,t}$ define a value that, in some way, represents the deviation from the minimum and maximum specification for property k, respectively. If property k for product p is between minimum and maximum specification values, both variables will be equal to zero. The corresponding objective penalty terms are shown in Eq. (32):

$$\text{penalty} = \sum_{t}\sum_{p}\sum_{k} (\text{plty}^{S+}_{k,p} D^{S+}_{k,p,t} + \text{plty}^{S-}_{k,p} D^{S-}_{k,p,t}) \tag{32}$$

8.3. Penalty for intermediate shortage

A common source of infeasible solutions is the lack of a minimum amount of intermediate required to satisfy either predefined component concentrations or certain market specifications. In this case, intermediate products can be purchased at higher cost from a third-party. The continuous variable $S_{i,t}$ defines the amount of intermediate i needed in time slot t, which allows to relax minimum inventory constraints:

$$V^{I}_{i,t} = \text{ini}_i + \text{prod}_i e_t - \sum_{p,t' \leq t} F^{I}_{i,p,t'} + S_{i,t}, \quad \forall i, t \tag{33}$$

The penalty term (34) is directly proportional to the component purchase cost:

$$\text{penalty} = \sum_{t}\sum_{i} (\text{plty}^{\text{SH}}_{i} S_{i,t}) \tag{34}$$

It should be noted that for the case of infeasible specifications the stopping criterion of the iterative procedure should be that the relative change in the bias parameter be less or equal than a specified tolerance.

9. Numerical results

The proposed discrete and continuous time MILP models can be solved using the iterative procedure outlined in Fig. 4 for the simultaneous blending and scheduling operations. The performance of the proposed MILP-based approach was tested with several real-world examples. The data are shown in Tables 1 and 2. Note that from the 12 product specifications shown in Table 2, properties P1, P2, P8 and P12 are non-linear, while the rest are linear. In the proposed iterative approach, properties P1 and P2 are estimated through Eq. (17′) whereas P8 and P12 are computed in Eq. (17″) (gravimetrically predicted). The basis of the example comprises nine intermediate products or components from the refinery, which can be blended in different ways to satisfy multiple demands of three gasoline grades with different specifications over an 8-day scheduling horizon. Twelve key component and product properties are taken into consideration for solving the blending problem, where the first eight can be predicted by a linear volumetric average whereas the remainder is based on non-linear correlations. All the information about components such as cost, constant production rate, initial, minimum and maximum stocks and properties is shown in Table 1. Product data including price, requirements, inventory constraints, rate, recipe limits and specifications are given in Table 2. Dedicated storage tanks with limited capacities for components and products and three equivalent blend headers working in parallel are available in the refinery. The main goal is to maximize the total profit (see Eq. (19)), considering component cost, product values and different penalties for component shortages and out-spec products. Note that no inventory costs were considered.

Table 1
Component data

	Component								
	C1	C2	C3	C4	C5	C6	C7	C8	C9
Cost ($/bbl)	24.00	20.00	26.00	23.00	24.00	50.00	50.00	50.00	50.00
Prod. rate (Mbbl/day)	15.00	33.00	20.00	14.00	18.00	10.00	0.00	0.00	0.00
Initial stock (Mbbl)	48.00	20.00	75.00	22.00	30.00	54.00	12.00	20.00	15.00
Min stock (Mbbl)	5.0	5.0	5.0	5.0	5.0	5.0	0.0	0.0	0.0
Max stock (Mbbl)	100.00	250.00	250.00	100.00	100.00	100.00	100.00	100.00	100.00
Property									
P1	93.00	104.00	104.90	94.80	87.40	118.00	87.30	95.20	93.30
P2	92.10	91.90	91.90	81.50	86.10	100.00	79.50	85.80	81.90
P3	0.7069	0.8692	0.6167	0.6731	0.6540	0.7460	0.7460	0.8187	0.7339
P4	3.60	1.00	100.00	94.90	91.50	15.00	0.00	1.30	34.30
P5	16.30	4.50	100.00	97.10	95.50	100.00	0.00	6.00	57.10
P6	94.30	93.50	100.00	100.00	100.00	100.00	0.00	93.90	95.90
P7	35.00	22.70	351.10	117.10	93.00	31.30	63.30	16.00	52.40
P8	0.007	0.00	0.00	0.009	0.0002	0.05	0.0063	0.1805	0.057
P9	0.00	88.60	0.00	2.30	0.20	0.00	43.98	65.30	21.30
P10	0.00	0.1	61.30	48.90	36.00	0.00	1.04	0.60	33.30
P11	0.00	3.30	0.00	1.10	0.10	0.00	3.33	0.90	0.80
P12	0.00	0.00	0.00	0.00	0.00	15.40	0.00	0.00	0.00

Table 2
Product data

	Product								
	G1 (price ($/bbl) = 31.00)			G2 (price ($/bbl) = 31.00)			G3 (price ($/bbl) = 31.00)		
	MIN	MAX	LIFT	MIN	MAX	LIFT	MIN	MAX	LIFT
Requirement (Mbbl)									
Day 1	5.00	45.00	10.00	5.00	50.00	12.00	5.00	50.00	10.00
Day 3				5.00	50.00	25.00			
Day 4	5.00	45.00	25.00	5.00	50.00	23.00			
Day 5									
Day 7	5.00	45.00	30.00						
Day 8	5.00	45.00	10.00				5.00	50.00	22.00
Inventory (Mbbl)	5.00	150.00		5.00	150.00		5.00	150.00	
Rate (Mbbl/day)	5.00	45.00		5.00	50.00		5.00	50.00	

	Product					
	G1 (price ($/bbl) = 31.00)		G2 (price ($/bbl) = 31.00)		G3 (price ($/bbl) = 31.00)	
	MIN	MAX	MIN	MAX	MIN	MAX
Recipe (%)						
C1	0.00	22.00	0.00	25.00	0.00	25.00
C2	0.00	20.00	0.00	24.00	0.00	24.00
C3	2.00	10.00	0.00	10.00	0.00	10.00
C4	0.00	6.00	0.00	23.00	0.00	23.00
C5	0.00	25.00	0.00	25.00	0.00	25.00
C6	0.00	10.00	0.00	10.00	0.00	10.00
C7	0.00	100.00	0.00	0.00	0.00	0.00
C8	0.00	100.00	0.00	0.00	0.00	0.00
C9	0.00	100.00	0.00	0.00	0.00	0.00
Specifications						
P1	95.00		98.00		98.00	
P2	85.00		88.00		88.00	
P3	0.72	0.775	0.72	0.775	0.72	0.775
P4	20.00	50.00	20.00	48.00	22.00	50.00
P5	46.00	71.00	46.00	71.00	46.00	71.00
P6	85.00		85.00		85.00	
P7	45.00	60.00	45.00	60.00	60.00	90.00
P8		0.015		0.015		0.008
P9		42.00		42.00		42.00
P10		18.00		18.00		18.00
P11		1.00		1.00		1.00
P12		2.70		2.70		2.70

Four different examples were solved with the purpose of analyzing the strong interaction between blending and scheduling decisions. In order to guarantee that feasible solutions are found, slack variables for property deviations and intermediate shortages were included in all cases, which were null for all solutions generated. Example 1 is only focused on the blending problem and its solution is used as initial product recipes for the other. Examples 2–4 are solved using the proposed model with the discrete and the continuous time domain representation. When the discrete time representation is used, the scheduling horizon is divided into six consecutive time intervals, where intervals 1, 3, 4 and 6 have 1-day duration whereas intervals 2 and 5 have 2-day duration. In order to make a direct comparison with the continuous time formulation, the time discretization is determined based on the product due dates. For the continuous time representation, one time slot with unknown duration is postulated for each one of the six subintervals defined by the product due dates.

9.1. Example 1 (blending problem)

Example 1 deals with a single-period blending problem of three products (G1–G3). The main goal is to find the best or ‘preferred’ recipe for each product that minimizes blend cost and simultaneously satisfies all quality specifications. Preferred recipes are proposed as the initial blends for the integrated blending and scheduling problems addressed in Examples 2–4. For this particular problem, temporal, inventory and resource constraints coming from the scheduling problem are disregarded by assuming that enough resources, component stocks and time are available as needed to produce 1 Mbbl of each product once. In this way only a pure blending problem is taken into consid-

Table 3
Iterative blending problem for product G1

Quality	Min. Spec.	Initial recipe (blend cost ($/bbl) = 29.30)	Iteration 1 (blend cost ($/bbl) = 29.97)		Iteration 2 (blend cost ($/bbl) = 29.99)		Max Spec.
		Value	Value	Approx	Value	Approx	
P1	95.00	97.891	97.898	97.7737	97.893	97.8928	
P2	85.00	88.417	88.470	88.0493	88.438	88.4335	
P3	0.72	0.7418	0.7325		0.7324		0.775
P4	20.00	34.455	35.418		35.409		50.00
P5	46.00	46.00	50.80		50.833		71.00
P6	85.00	96.460	91.797		91.780		
P7	45.00	60.00	60.00		60.00		60.00
P8		0.0378	0.0152	0.0150	0.0150	0.0150	0.015
P9		28.458	22.974		22.923		42.00
P10		14.256	15.974		16.005		18.00
P11		0.8964	1.00		1.00		1.00
P12		1.1223	1.5684	1.5488	1.5687	1.5684	2.70

eration. Component costs and properties, variable recipe limits and stringent product specifications are the central features to be considered for solving Example 1, where it is assumed that all scheduling decisions are made a priori. The proposed LP-based iterative procedure was used to find preferred recipes for all required products. As reported in Table 13 in Section 11 on computational results, the problem involves 81 constraints and 127 continuous variables and its solution was found in 0.13 s. In this case, initial product recipes were generated taking into account only linear product properties. Then the iterative procedure was performed to update the initial recipes with the purpose of satisfying all product specifications. Preferred recipes for products G2 and G3 were found by executing just one iteration of the proposed procedure, whereas an additional iteration was needed to satisfy all specifications for product G1, since the maximum specification for property P8 was violated both in the initial recipe as in the first iteration (see Table 3). In order to generate feasible recipes, component concentrations for each product were updated by the LP model in each iteration, which gradually increased the blend cost. The recipe evolution for product G1 in terms of component concentration is presented in detail in Fig. 6. Blend cost and product properties associated to each recipe are shown in Table 3. In addition to the exact values for each property predicted by non-linear correlations, the approximations predicted by the proposed linear functions are also presented in Table 3. It should be noted that predictions of non-linear properties tend to improve when the number of iterations is increased. Finally, best product recipes and 'bias' factors for all products are reported in Table 4.

9.2. *Example 2 (blending and scheduling with limited production)*

In Example 2 preferred product recipes found in Example 1 were used as the initial solution for the proposed iterative MILP-based procedure. Despite using linear approximations, the proposed MILP model was capable of finding in just one iteration the same solution generated by non-linear optimization tools. However, although the discrete and continuous time representations obtained the same profit in terms of component cost and product value ($ 1,611,210), the continuous time representation is able to find a schedule that operates the blenders at

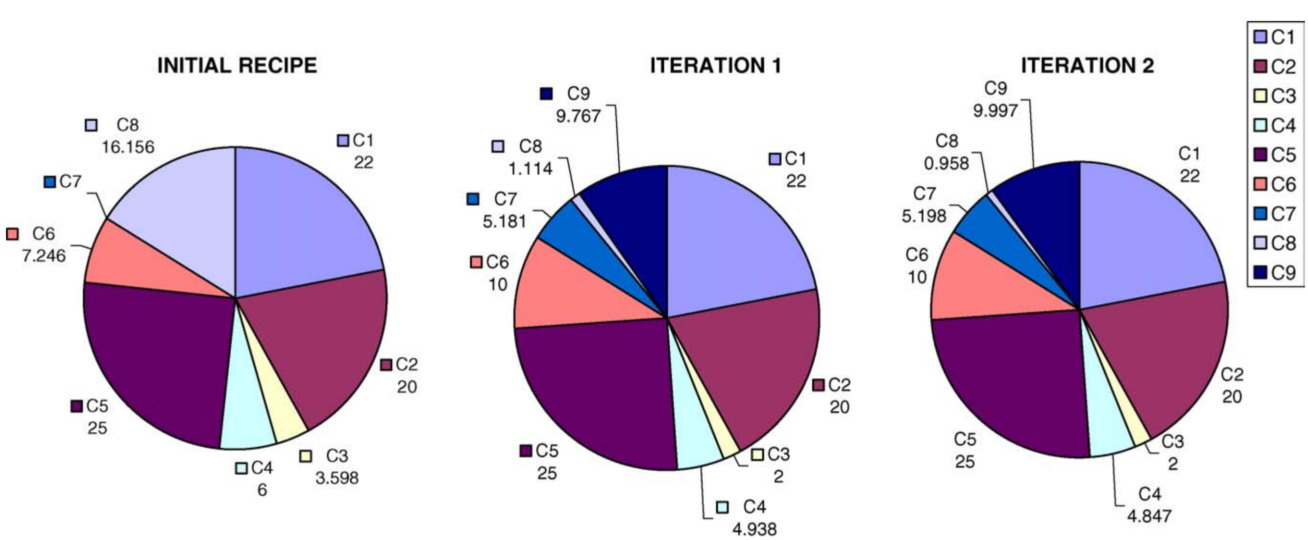

Fig. 6. Convergence to preferred recipe for product G1 (iterative procedure).

Table 4
Preferred product recipes

	Product		
	G1 (blend cost ($/bbl) = 29.99)	G2 (blend cost ($/bbl) = 25.28)	G3 (blend cost ($/bbl) = 24.98)
Recipe (%)			
C1	22.00	25.00	25.00
C2	20.00	23.947	24.00
C3	2.00	16.794	1.372
C4	4.847	25.00	16.636
C5	25.00	9.259	25.00
C6	10.00		7.992
C7	5.198		
C8	0.958		
C9	9.997		
Quality			
P1	97.893 (bias = 1.527)	98.4122 (bias = 1.5611)	98.2214 (bias = 1.5208)
P2	88.438 (bias = −0.659)	88.4594 (bias = −1.0439)	88.3310 (bias = −1.0861)
P3	0.7324	0.7305	0.7289
P4	35.409	41.3410	42.3734
P5	50.833	54.5932	54.5475
P6	91.780	97.0184	97.0150
P7	60.00	60.00	64.2465
P8	0.0150	0.0079	0.0072
P9	22.923	21.6536	21.6966
P10	16.005	17.2363	18.00
P11	1.00	1.00	1.00
P12	1.5687	1.4561	1.2597

full capacity for 2.67 days less than the discrete time representation, which can significantly reduce the total operating cost. Product schedules based on a discrete and continuous time representation are reported in Tables 5 and 6, respectively. Gantt charts and inventory evolution of components for both discrete and continuous time representations are shown in Fig. 7. As shown in Table 13, the discrete time formulation involves 679 constraints, 9 binary variables, and 757 continuous variables. The continuous time formulation comprises 832 constraints, 9 binary variables, and 841 continuous variables. Both models were solved in 0.26 s.

9.3. Example 3 (blending and scheduling with flexible production)

This example evaluates in Example 2 the effect of predefining minimum and maximum requirements for each time interval. In this way the amount to be produced in each time interval

Table 5
Product schedule (Example 2—discrete time representation)

Product	Period	Start	End	Prod	Lift	Inventory
G1	T1	0.00	1.00	15.02	10.00	5.02
	T2	1.00	3.00	0.00	0.00	5.02
	T3	3.00	4.00	45.00	25.00	25.02
	T4	4.00	5.00	0.00	0.00	25.02
	T5	5.00	7.00	45.00	30.00	40.02
	T6	7.00	8.00	45.00	10.00	75.02
G2	T1	0.00	1.00	50.00	12.00	38.00
	T2	1.00	3.00	50.00	25.00	63.00
	T3	3.00	4.00	50.00	23.00	90.00
	T4	4.00	5.00	0.00	0.00	90.00
	T5	5.00	7.00	0.00	0.00	90.00
	T6	7.00	8.00	0.00	0.00	90.00
G3	T1	0.00	1.00	50.00	10.00	40.00
	T2	1.00	3.00	0.00	0.00	40.00
	T3	3.00	4.00	0.00	0.00	40.00
	T4	4.00	5.00	0.00	0.00	40.00
	T5	5.00	7.00	0.00	0.00	40.00
	T6	7.00	8.00	50.00	22.00	68.00

Table 6
Product schedule (Example 2—continuous time representation)

Product	Period	Start	End	Prod	Lift	Inventory
G1	T1	0.00	1.00	45.00	10.00	35.00
	T2	1.00	2.00	0.00	0.00	35.00
	T3	3.00	4.00	45.00	25.00	55.00
	T4	4.00	5.00	0.00	25.00	55.00
	T5	5.00	5.33	15.02	0.00	40.02
	T6	7.00	8.00	45.00	10.00	75.02
G2	T1	0.00	1.00	50.00	12.00	38.00
	T2	1.00	2.00	50.00	0.00	63.00
	T3	3.00	4.00	50.00	23.00	90.00
	T4	4.00	5.00	0.00	23.00	90.00
	T5	5.00	5.33	0.00	0.00	90.00
	T6	7.00	8.00	0.00	0.00	90.00
G3	T1	0.00	1.00	50.00	10.00	40.00
	T2	1.00	2.00	0.00	0.00	40.00
	T3	3.00	4.00	0.00	0.00	40.00
	T4	4.00	5.00	0.00	0.00	40.00
	T5	5.00	5.33	0.00	0.00	40.00
	T6	7.00	8.00	50.00	22.00	68.00

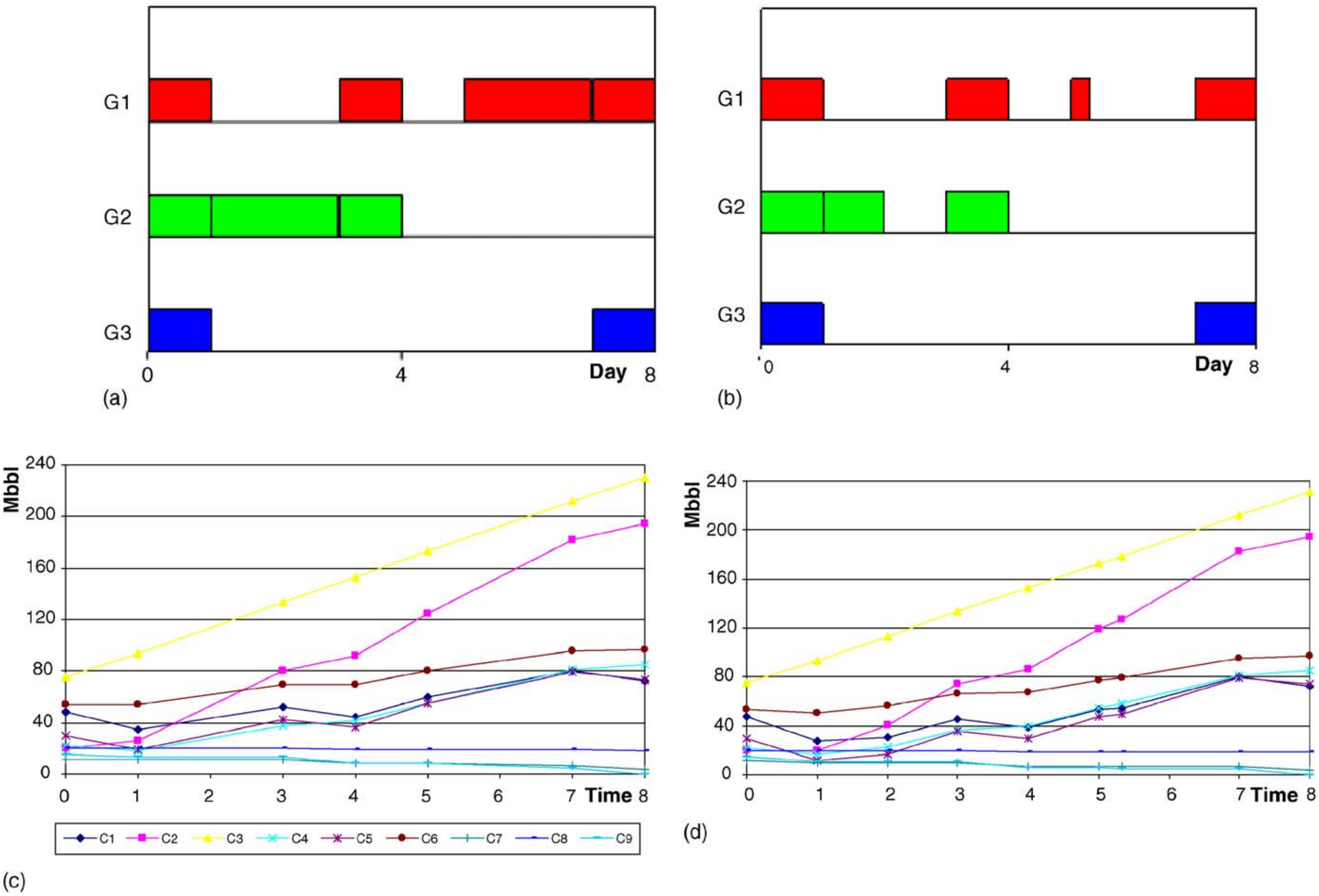

Fig. 7. Gantt charts and evolution of component stocks (Example 2): (a) discrete time; (b) continuous time; (c) discrete time; (d) continuous time.

becomes a model variable only restricted by minimum and maximum production rates. The amount of product to be lifted at specific due dates is still a hard constraint to be satisfied. This modification allows the model to increase the total production by almost 36%, i.e. from 400.02 to 542.02 Mbbl, which represents increasing the total profit to $ 2,448,050, which is almost a 52% increase (see Table 12). Preferred product recipes are used for all products and one iteration is only executed. Product schedules based on a discrete and continuous time representation are shown in Tables 7 and 8, respectively. In this example we note that the continuous time representation needs 2.60 days less of total operating time to reach the same production level as the discrete time model. Fig. 8 shows Gantt-charts and evolution of component stock for Example 3. The discrete time formulation comprises 679 constraints, 18 binary variables, and 757 continuous variables and its solution was found in 0.23 s. The continuous time formulation comprises 832 constraints, 18 binary variables, and 841 continuous variables and its solution was generated in 0.26 s (see Table 13).

9.4. *Example 4 (full re-blending and re-scheduling with limited production)*

Finally, this example deals with a modified version of the original Example 2 where the following changes are introduced: (1) properties P1 and P2 are decreased by one for components C1–C3 and C6, (2) the price of G3 is increased to 31.05 $/bbl, (3) component cost is increased to 27 and 23 $/bbl for C1 and C2 and (4) production rates for C1 and C2 are reduced to 13

Table 7
Product schedule (Example 3—discrete time representation)

Product	Period	Start	End	Prod	Lift	Inventory
G1	T1	0.00	1.00	45.00	10.00	35.00
	T2	1.00	3.00	60.02	0.00	95.02
	T3	3.00	4.00	0.00	25.00	70.02
	T4	4.00	5.00	0.00	0.00	70.02
	T5	5.00	7.00	0.00	30.00	40.02
	T6	7.00	8.00	45.00	10.00	75.02
G2	T1	0.00	1.00	50.00	12.00	38.00
	T2	1.00	3.00	0.00	25.00	13.00
	T3	3.00	4.00	50.00	23.00	40.00
	T4	4.00	5.00	0.00	0.00	40.00
	T5	5.00	7.00	60.00	0.00	100.00
	T6	7.00	8.00	50.00	0.00	150.00
G3	T1	0.00	1.00	50.00	10.00	40.00
	T2	1.00	3.00	72.00	0.00	112.00
	T3	3.00	4.00	0.00	0.00	112.00
	T4	4.00	5.00	0.00	0.00	112.00
	T5	5.00	7.00	10.00	0.00	122.00
	T6	7.00	8.00	50.00	22.00	150.00

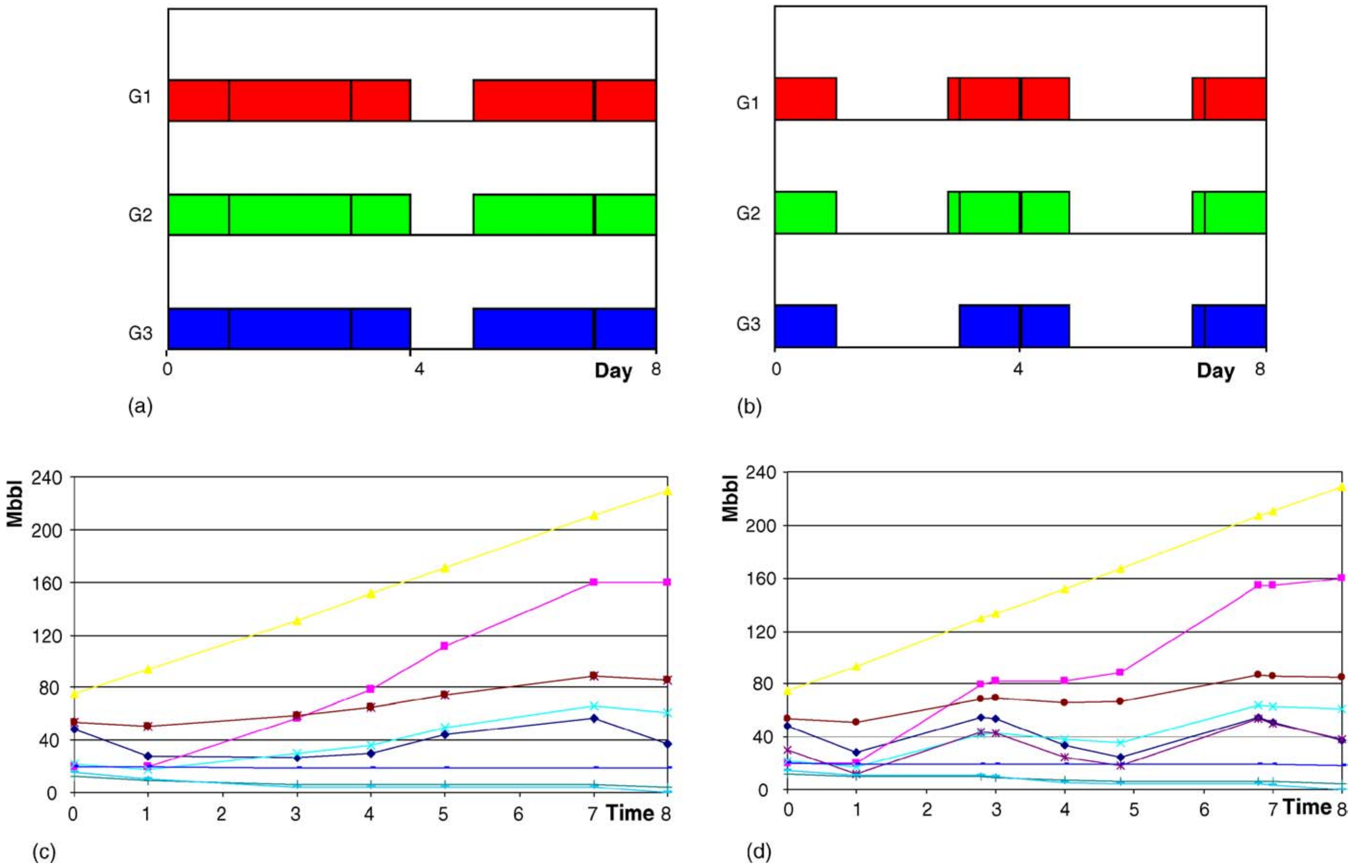

Fig. 8. Gantt charts and evolution of component stocks (Example 3): (a) discrete time; (b) continuous time; (c) discrete time; (d) continuous time.

and 31 Mbbl/day respectively. All other data remain as in the original example. The main goal here is to analyze the effect of these changes in the blending and scheduling decisions. Detailed product schedules for discrete and continuous time representations for Example 4 are shown in Tables 9 and 10, respectively. Regarding the blending decisions, preferred recipes found in Example 1 are proposed as the initial solution. However, they have to be updated because some preferred recipes become infeasible because of the modifications introduced. Only one iteration is required to modify the infeasible recipes related to

Table 8
Product schedule (Example 3—continuous time representation)

Product	Period	Start	End	Prod	Lift	Inventory
G1	T1	0.00	1.00	45.00	10.00	35.00
	T2	2.80	3.00	9.00	10.00	44.00
	T3	3.00	4.00	45.00	35.00	64.00
	T4	4.00	4.80	4.00	35.00	68.00
	T5	6.80	7.00	9.00	65.00	47.00
	T6	7.00	8.00	38.02	75.00	75.02
G2	T1	0.00	1.00	50.00	12.00	38.00
	T2	2.80	3.00	10.00	37.00	23.00
	T3	3.00	4.00	50.00	60.00	50.00
	T4	4.00	4.80	40.00	60.00	90.00
	T5	6.80	7.00	10.00	60.00	100.00
	T6	7.00	8.00	50.00	60.00	150.00
G3	T1	0.00	1.00	50.00	10.00	40.00
	T2	2.80	3.00	0.00	10.00	40.00
	T3	3.00	4.00	50.00	10.00	90.00
	T4	4.00	4.80	40.00	10.00	130.00
	T5	6.80	7.00	10.00	10.00	140.00
	T6	7.00	8.00	32.00	32.00	150.00

Table 9
Product schedule (Example 4—discrete time representation)

Product	Period	Start	End	Prod	Lift	Inventory
G1	T1	0.00	1.00	45.00	10.00	35.00
	T2	1.00	3.00	0.00	0.00	35.00
	T3	3.00	4.00	5.00	25.00	15.00
	T4	4.00	5.00	0.00	0.00	15.00
	T5	5.00	7.00	20.00	30.00	5.00
	T6	7.00	8.00	10.00	10.00	5.00
G2	T1	0.00	1.00	50.00	12.00	38.00
	T2	1.00	3.00	100.00	25.00	113.00
	T3	3.00	4.00	50.00	23.00	140.00
	T4	4.00	5.00	0.00	0.00	140.00
	T5	5.00	7.00	0.00	0.00	140.00
	T6	7.00	8.00	0.00	0.00	140.00
G3	T1	0.00	1.00	50.00	10.00	40.00
	T2	1.00	3.00	0.00	0.00	40.00
	T3	3.00	4.00	0.00	0.00	40.00
	T4	4.00	5.00	0.00	0.00	40.00
	T5	5.00	7.00	0.00	0.00	40.00
	T6	7.00	8.00	50.00	22.00	68.00

Table 10
Product schedule (Example 4—continuous time representation)

Product	Period	Start	End	Prod	Lift	Inventory
G1	T1	0.00	1.00	16.00	10.00	6.00
	T2	1.00	3.00	0.00	10.00	6.00
	T3	3.00	4.00	45.00	35.00	26.00
	T4	4.00	5.00	0.00	35.00	26.00
	T5	5.00	5.20	9.00	35.00	5.00
	T6	7.00	8.00	10.00	75.00	5.00
G2	T1	0.00	1.00	50.00	12.00	38.00
	T2	1.00	3.00	100.00	37.00	113.00
	T3	3.00	4.00	50.00	60.00	140.00
	T4	4.00	5.00	0.00	60.00	140.00
	T5	5.00	5.20	0.00	60.00	140.00
	T6	7.00	8.00	0.00	60.00	140.00
G3	T1	0.00	1.00	50.00	10.00	40.00
	T2	1.00	3.00	0.00	10.00	40.00
	T3	3.00	4.00	0.00	10.00	40.00
	T4	4.00	5.00	0.00	10.00	40.00
	T5	5.00	5.20	0.00	10.00	40.00
	T6	7.00	8.00	50.00	32.00	68.00

products G2 and G3. Original preferred and updated recipes for these products are compared in Table 11. As shown, the new recipes satisfy all product specifications but at the same time, updated component concentrations increase the blending cost with which the profit is reduced from $ 2,448,050 to 1,234,490. This difference mainly arises because component costs were increased and octane numbers were reduced. It should be noted that key properties such as P1 and P2 are satisfied with a very small margin, which means that quality giveaway is also minimized through the proposed method. Computational requirements for this example are summarized in Table 13.

10. Computational results

Different blending and scheduling problems were solved in the previous section in order to evaluate the efficiency of the proposed method. Example 1 dealt with a pure blending problem whereas Examples 2–4 also accounted for optimal scheduling decisions. Examples 3 and 4 correspond to modified versions of the original Example 2 where minimum and maximum requirements were relaxed (Example 3) and certain changes in component properties and cost and product prices were incorporated (Example 4). Table 12 summarizes the results for Examples 2–4, while Table 13 provides the computational statistics on the four examples. As can be seen, the size of the MILP problems is not very large and involves a modest number of 0–1 variables. For this reason every single problem needs no more than 1 s of CPU time with CPLEX 8.1, which highlights the computational efficiency of the proposed models and the iterative MILP procedure. In addition, a very small number of iterations were required to satisfy all product specifications in all the examples. As a general characteristic, it was observed that discrete time formulations usually have a better computational performance when compared to continuous models. On the other hand, continuous formulations are able to generate more flexible schedules that significantly reduce the operating time of the available equipment.

Table 11
Updated product recipes (Example 4)

	Product			
	G2		G3	
	Preferred (blend cost ($/bbl) = 25.28)	Updated (blend cost ($/bbl) = 26.92)	Preferred (blend cost ($/bbl) = 24.98)	Updated (blend cost ($/bbl) = 26.67)
Recipe (%)				
C1	25.00	25.00	25.00	25.00
C2	23.947	24.00	24.00	24.00
C3	16.794	0.223	1.372	3.195
C4	25.00	16.09	16.636	14.869
C5	9.259	24.831	25.00	24.269
C6		9.856	7.992	8.640
Quality				
P1	97.8204	98.0235	97.6283	98.052
P2	87.8588	88.0133	87.7294	88.0455
P3	0.7305	0.7309	0.7289	0.7285
P4	41.3408	40.831	42.3734	41.9724
P5	54.5936	54.571	54.5476	54.6305
P6	97.0184	97.015	97.015	97.015
P7	60.00	60.00	64.2473	68.1274
P8	0.0079	0.0081	0.0072	0.0074
P9	21.6533	21.6837	21.6966	21.6546
P10	17.2362	16.968	18.00	18.00
P11	1.00	0.9938	1.00	0.9799
P12	1.4562	1.5491	1.2597	1.3626

Table 12
Summary of results

Example	Blend value (M$)	Comp. stock production (M$)	Comp. inventory build (M$)	Total profit (M$)	Profit per barrel ($/bbl)
2	12400.61	22352	11562.6	1611.21	4.03
3	16802.61	22352	7997.44	2448.05	4.52
4	11785	23504	12953.49	1234.49	3.25

Table 13
Model size and computational requirements

Example	Binary vars, cont. vars, constraints	CPU time	Iterations
1	–, 127, 81	0.13[a,b]	2
2 (discrete)	9, 757, 679	0.26[a]	1
2 (continuous)	9, 841, 832	0.26[a]	1
3 (discrete)	18, 757, 679	0.23[a]	1
3 (continuous)	18, 841, 832	0.26[a]	1
4 (discrete)	9, 757, 679	0.23[a]	1
4 (continuous)	9, 841, 832	0.26[a]	1

[a] Seconds on Pentium IV PC with CPLEX 8.1 in GAMS 21.2.
[b] All scheduling decisions are predefined.

In order to examine the solution of the scheduling and blending problem addressed in this paper using directly the non-linear correlations, Table 14 presents the computational results for Examples 1 and 2. Consideration of the original non-linear correlations gives rise to NLP and MINLP models, which were solved by several non-linear general-purpose optimizers. Local solutions are obtained with MINOS, CONOPT, and DICOPT, whereas global solutions can be obtained with BARON. Since the component concentrations in each product may have a significant influence on the non-linear model performance, examples were solved considering different initial values for these key problem variables.

The analysis of the results reported in Table 14 reveals some important features. First, it is worth mentioning that both the linear and the non-linear models were able to find the same optimal solution for these examples. However, in several cases the non-linear models failed to converge due to the non-convexities in the non-linear model. An additional problem was the execution errors that arose from evaluation errors in the non-linear functions. These errors are generated because the original non-linear correlations may be not defined for the entire domain of the variables or some possible combinations of feasible values. Usually, non-linear solvers have difficulty recovering after attempting an undefined operation such as dividing by zero or raising a negative number to a real power. Although these problems can partially be solved in some cases, the required changes may compromise the optimality of the solution. Also, it can be seen in Table 14 that the proposed method is significantly faster than the NLP solvers in Example 1, and particularly, compared to the MINLP solvers in Example 2. Thus, from Table 14 it is clear that the

Table 14
Comparison with non-linear codes

Example	Solver	Initial component concentration (%)	Objective function	CPU time[a]
1	Proposed approach	–	80.251	0.13
1	MINOS	0	Execution error[c]	–
1	CONOPT	0	Infeasible solution[d]	–
1	MINOS	10	80.251	0.4
1	CONOPT	10	80.251	0.3
1	MINOS	Minimum allowed[b]	Execution error[c]	–
1	CONOPT	Minimum allowed	80.251	0.6
1	MINOS	Maximum allowed	Execution error[c]	–
1	CONOPT	Maximum allowed	Execution error[c]	–
2	Proposed approach	–	1611.21	0.26
2	DICOPT/MINOS	0	Execution error[c]	–
2	DICOPT/CONOPT	0	1611.21	1.5
2	BARON	0	1611.21	∼1000
2	DICOPT/MINOS	10	1611.21	1.4
2	DICOPT/CONOPT	10	1611.21	1.3
2	BARON	10	1611.21	∼1000
2	DICOPT/MINOS	Minimum allowed	Execution error[c]	–
2	DICOPT/CONOPT	Minimum allowed	1611.21	1.9
2	BARON	Minimum allowed	1611.21	∼1000
2	DICOPT/MINOS	Maximum allowed	Execution error[c]	–
2	DICOPT/CONOPT	Maximum allowed	Execution error[c]	–
2	BARON	Maximum allowed	Execution error[c]	–

[a] Seconds on Pentium IV PC with GAMS 21.2.
[b] For specified recipe.
[c] Execution errors arise from evaluation errors in the non-linear functions.
[d] Infeasible solutions suggest that convergence problems may arise from non-convexities of non-linear functions.

proposed successive linear approach is fast and robust, and very useful for addressing real-world cases.

11. Conclusions

An integrated MILP-based approach has been proposed to simultaneously optimize the gasoline off-line blending and the short-term scheduling problem in oil-refinery. The method is able to deal with non-linear product properties and variable recipes through a successive LP or MILP iterative procedure that can be used either on discrete or continuous time formulations. Several examples representative of real world problems were presented to illustrate the flexibility and efficiency of the proposed models and solution technique. Also, sufficient conditions for the convergence of the successive LP procedure to a local solution of the blending problem have been presented, as well as numerical comparisons with NLP and MINLP solvers showing that the proposed method converged to the same solutions, but faster and more reliably.

Acknowledgements

The authors would like to thank ABB Corporate Research and the Center for Advanced Process Decision-making (CAPD) at Carnegie Mellon for financial support of this work.

Appendix A. On the convergence of the successive LP method for blending operations

For simplicity we consider for the blending problem the case of a single period problem and hence we drop the subscript t for time. The equations in (1)–(6) with a cost objective function and assuming only upper bound for the specifications can be written as follows:

(P):

$$\max \sum_p \alpha_p F_p^P - \sum_i \sum_p \beta_i F_{i,p}^I \tag{A.1}$$

$$\text{s.t.} \quad F_p^P = \sum_i F_{i,p}^I, \quad \forall p \tag{A.2}$$

$$\sum_i \mathrm{pr}_{i,k} F_{i,p}^I \le \mathrm{pr}_{p,k}^{\max} F_p^P, \quad \forall p, \forall k \in K_{\mathrm{LIN}} \tag{A.3}$$

$$\sum_i \mathrm{pr}_{i,k} F_{i,p}^I \le \mathrm{pr}_{p,k}^{\max} F_p^P + \delta_{p,k} F_p^P, \quad \forall p, \forall k \in K_{\mathrm{NL}} \tag{A.4}$$

$$\delta_{p,k} = \max\left\{0, \mathrm{PR}_{p,k}\left(\frac{F_{i,p}^I}{F_p^P}\right)\right\}, \quad \forall p, \forall k \in K_{\mathrm{NL}} \tag{A.5}$$

$$F_{i,p}^I, F_p^P \ge 0 \tag{A.6}$$

where in the above K_{LIN} and K_{NL} represent the linear and non-linear properties, respectively, and $\mathrm{PR}_{p,k}(F_{i,p}^I/F_p^P)$ represents the evaluation of the non-linear property at the volume fraction $v_{i,k} = F_{i,p}^I/F_p^P$. Also, note that the successive LP method assumes fixed values of $\delta_{p,k}$ at each successive iteration.

The two following cases provide a sufficient condition to the convergence of the successive LP procedure to a local minimum of problem (P).

Case 1. The non-linear property $\mathrm{PR}_{p,k} < \mathrm{pr}_{p,k}^{\max}$ at the local minimum of problem (P).

Proof. Since in this case $\delta_{p,k} = 0$ and (A.4) is redundant, problem (P) reduces to the problem given by (A.1)–(A.3) and (A.6), which is the LP solved at the first iteration of the procedure. Hence, it trivially follows that a solution to this problem is equivalent to the solution of problem (P). □

Case 2. The non-linear property $\mathrm{PR}_{p,k} < \mathrm{pr}_{p,k}^{\max}$ at the local minimum of problem (P) and $(\delta \mathrm{PR}_{p,k})/(\delta v_{i,k}) = 0$.

Proof. For convenience we represent problem (P) in compact form as:

(PC):

$$\min \sum_j c_j x_j \tag{A.7}$$

$$\text{s.t.} \sum_j a_{i,j} x_j \le b_i, \quad i \in I_1 \tag{A.8}$$

$$\sum_j a_{i,j} x_j + g_i(x) x_i \le d_i, \quad i \in I_2 \tag{A.9}$$

$$x_j \ge 0 \tag{A.10}$$

where we assume the function $g_i(x) > 0$ represents (A.5).

The Karush–Kuhn–Tucker conditions of (PC) yield the following stationary condition:

$$c_j + \sum_{i \in I_1} \lambda_i a_{i,j} + \sum_{i \in I_2} \mu_i a_{i,j} + \mu_j \left[g_j(x^*) + \frac{\partial g_j}{\partial x_j} x_j \right] - \rho_j = 0 \tag{A.11}$$

where λ_i, μ_i, ρ_i are the multipliers of (A.8)–(A.10), respectively. When the successive LP procedure is used (A.9) is replaced by

$$\sum_j a_{i,j} x_j + \delta_i x_i \le d_i, \quad i \in I_2 \tag{A.12}$$

where δ_i is treated as a constant. For this case the stationary condition of the Karush–Kuhn–Tucker condition yields:

$$c_j + \sum_{i \in I_1} \lambda_i a_{i,j} + \sum_{i \in I_2} \mu_i a_{i,j} + \mu_j \delta_j - \rho_j = 0 \tag{A.13}$$

Since δ_j can be set equal to $g_j(x^*)$ at the optimum solution x^*, this implies that (A.11) and (A.12) are identical if $(\delta g_j)/(\delta x_j) = 0$.

Note that from (A.5) case 2 means that convergence to a local solution of the non-linear programming problem (P) can be guaranteed if the non-linear properties are not a strong function of the compositions. □

References

Adams, W., & Sherali, H. (1990). Linearization strategies for a class of zero-one mixed integer programming problems. *Operations Research*, *38*, 217.

Bechtel Corp. (1993). PIMS (Process Industry Modeling System). *User's manual, version 6.0*. Houston, TX.

Booner & Moore. (1979). Management Science. RPMS (Refinery and Petrochemical Modeling System). *A system description*. Houston, TX.

Floudas, C. A., & Lin, X. C. (2004). Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. *Computers and Chemical Engineering*, *28,* 2109.

Glismann, K., & Gruhn, G. (2001). Short-term scheduling and recipe optimization of blending processes. *Computers and Chemical Engineering*, *25*, 627.

Grossmann, I. E., Van den Heever, S. A., & Harjunkoski, I. (2002). Discrete optimization methods and their role in the integration of planning and scheduling. *AIChE Symposium Series No. 326* (pp. 98, 150).

Jia, Z., & Ierapetritou, M. (2003). Mixed-integer linear programming for gasoline blending and distribution scheduling. *Industrial Engineering and Chemistry Research*, *42*, 825.

Kallrath, J. (2003). Planning and scheduling in the process industry. In *Advanced planning and scheduling solutions in process industry*. Springer-Verlag, p. 11, ISBN 3-540-00222-7.

Kelly, J. D. (2004). Formulating production planning models. *Chemical Engineering Progress*, 43–50.

Kelly, J. D., & Mann, J. L. (2003a). Crude-oil blend scheduling optimization: An application with multi-million dollar benefits: Part I. *Hydrocarbon Processing*, 47–53.

Kelly, J. D., & Mann, J. L. (2003b). Crude-oil blend scheduling optimization: An application with multi-million dollar benefits: Part II. *Hydrocarbon Processing*, 72–79.

Lee, H., Pinto, J. M., Grossmann, I. E., & Park, S. (1996). Mixed-integer linear programming model for refinery short-term scheduling of crude oil unloading with inventory management. *Industrial Engineering and Chemistry Research*, *35*, 1630.

Moro, L. F. L., Zanin, A. C., & Pinto, J. M. (1998). A planning model for refinery diesel production. *Computers and Chemical Engineering*, *22*, S1039.

Oral, M., & Kettani, O. (1992). A linearization procedure for quadratic and cubic mixed-integer problems. *Operations Research*, *40,* S109.

Pinto, J. M., & Grossmann, I. E. (1998). Assignment and sequencing models for the scheduling of chemical processes. *Annals of Operation Research*, *81*, 433.

Quesada, I., & Grossmann, I. E. (1995). Global optimization of bilinear process networks with multicomponents flows. *Computers and Chemical Engineering*, *19*, 1219.

Reklaitis, G. V. (1992). Overview of scheduling and planning of batch process operations. *Technical Report*. Antalya, Turkey: NATO Advanced Study Institute.

Rigby, B., Lasdon, L. S., & Waren, A. D. (1995). The evolution of Texaco blending systems—From Omega to StarBlend. *Interfaces*, *25*, 64.

Shah, N. (1998). Single- and multisite planning and scheduling: Current status and future challenges. *AIChE Symposium Series No 320* (pp. 94, 75).

Wenkai, L., Hui, C.-W., Hua, B., & Tong, Z. (2002). Scheduling crude-oil unloading, storage and processing. *Industrial Engineering and Chemistry Research*, *41*, 6723.