AIChE

Inventory Pinch Algorithm for Gasoline Blend Planning

Pedro A. Castillo Castillo, and Vladimir Mahalec
Dept. of Chemical Engineering, McMaster University, Canada L8S 4L8

Jeffrey D. Kelly
Industrial Algorithms, Toronto, Canada M1P 4C3

DOI 10.1002/aic.14113
Published online June 4, 2013 in Wiley Online Library (wileyonlinelibrary.com)

Current gasoline blend scheduling practice is to optimize blend plans via fixed duration (e.g., days) multiperiod NLP or MINLP models and schedule blends via interactive simulation. Solutions of multiperiod models typically have different blend recipes for each time period. We introduce inventory pinch points and use them to construct an algorithm based on single-period nonlinear model to minimize the number of different blend recipes. The algorithm optimizes multigrade blend recipes for each period delimited by the inventory pinch points and then uses a fine-grid multiperiod fixed-recipe MILP to compute blend volumes profile. If MILP is infeasible, a corresponding period between the pinch points is subdivided and recipes are reoptimized. In our case studies, solutions are computed in significant less time and are most often within 0.01% of the solutions by multiperiod MINLP. Reduced number of blend recipes makes it easier for the blend scheduler to create a schedule by interactive simulation. © 2013 American Institute of Chemical Engineers *AIChE J*, 59: 3748–3766, 2013

Keywords: gasoline blend planning, inventory pinch, recipe optimization, minimal number of recipes, periods with constant blend recipes

Introduction

Crude oil refineries produce various liquid fuels by blending intermediate product streams in a manner which minimizes use of more valuable components, while meeting product specifications. Product specification are either "greater than or equal" or "less than or equal" constraints for various product properties (also referred as qualities, e.g., octane number, olefins content, sulfur content, specific gravity, etc.). Usually, there is one blender for each type of liquid products, that is, one blender for gasoline, one blender for diesel, etc. Refineries operate to meet the contracted product demand. Hence, an optimal refinery operation is the one that meets contracted products liftings, while minimizing operating costs and inventory carrying costs. Total refinery production is planned on a long term basis (e.g., month-to-month) while the refinery operation over short time horizons (e.g., 2 or 4 weeks) is usually separated in two major tasks (1) scheduling of process units, and (2) planning and scheduling of liquid products blending. Refinery production planning linear programming (LP) models are being replaced by nonlinear programming (NLP) models due to new regulations, more expensive raw materials, higher utility costs, and other factors.[1] Most of the commercial software use successive linear programming (SLP) techniques to solve NLP models,[2] while AspenTech's nonlinear PIMS uses sequential quadratic programming (SQP) methods. The refinery planning models are formulated as multiperiod models[3] where the planning horizon is divided into several time periods of specified duration (i.e., discrete time representation). Constraints are satisfied at the time period boundaries and serve as the basis for scheduling operations of the process units. Gasoline comprises the largest volume of liquid products and minimization of the gasoline blending costs has a major impact on refinery profitability.

Since the refinery operations are based on the results of the longer term production plan, process units continuously produce gasoline blend components as determined by that plan. These components are stored in an intermediate storage and then used to blend gasoline. The refinery carries the inventories either as blend components or as finished gasoline, as determined by the refinery production plan. Hence, the cost of carrying the product inventory over the gasoline blend planning and scheduling horizon (typically 1 to 2 weeks) does not need to be included in the gasoline planning objective function, since the refinery will carry either the blend components or the finished gasoline.

In a typical gasoline blend system (Figure 1), blend components arrive from upstream process units and are stored in their respective tanks. In the blend headers ("blenders") the blend components are mixed in a predetermined ratio ("blend recipe") to meet the specifications for the gasoline grade (e.g., regular, medium, premium) that is being blended. There can be one or more blenders operating in parallel. A gasoline blender switches from blending one grade to another grade. During the switching the analytical instruments (e.g., octane engine) need to be recalibrated to ensure accurate online measurement of the blend properties.

Correspondence concerning this article should be addressed to V. Mahalec at mahalec@mcmaster.ca.

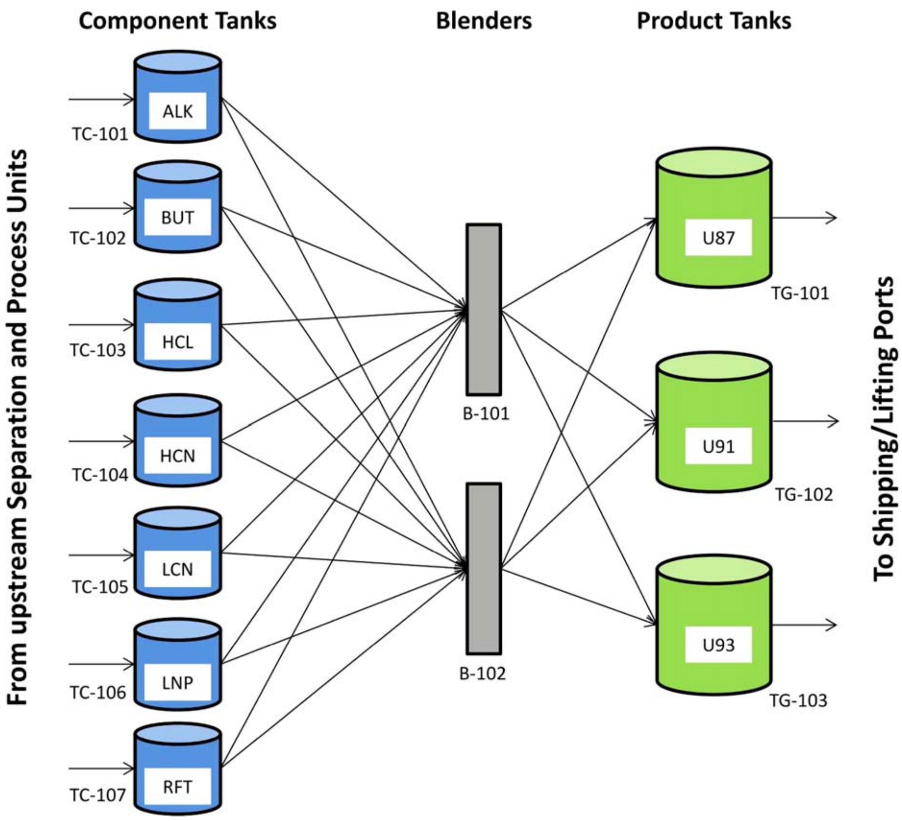

Figure 1. Typical gasoline blending system.

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

Preparation time results in a lost blend capacity. Hence, an optimal gasoline blending operation will minimize the total cost of all blends across the blend planning horizon and also minimize the number of switches.

Nonlinearities in the blend planning model arise from (1) nonlinear blending properties which introduce nonconvex terms,[4] (2) including unknown future quality and future volume (tank heel) for each grade of gasoline in the multiperiod model, and (3) other attributes of the pooling problem.[5]

Properties of the blended gasoline are nonlinear functions of the properties of the components that are blended to make a specific grade. During the 60s and 70s the practitioners developed transformations of individual properties into so called blending indices (BIs), which blend linearly component BIs into gasoline BIs. If blending index transformations are applied also to product constraints, then the equations governing the blend properties become linear, if the properties of every blend component are known. However, recent changes in gasoline blend specification (Environmental Protection Agency (EPA), *Title 40 Code of Federal Regulations Part 80.45: Complex Emissions Model* [40CFR80.45, 2007][6]) introduce highly nonlinear constraints with respect to product properties. Misener et al.[7] presented a single-period MINLP model that includes binary variables to represent the quality breakpoints defined by the EPA model. To solve this highly nonlinear mixed-integer model to global optimality, they introduced a specialized algorithm. One of their case studies consisted of 1104 continuous variables, 150 binary variables, and 640 nonlinear terms; it was solved in 5274 CPU seconds with an optimality gap of 0.5%. Operation scheduling was not included in their model.

The earliest approach to gasoline blending decomposes the problem in two steps: blend planning (also known as offline blending problem or recipe optimization problem) and blend scheduling. This is still the prevailing practice, as witnessed by the commercial success of blend planning tools (e.g., Honeywell BLEND, Aspen PIMS-MBO) and scheduling tools (e.g., Production Scheduler from Honeywell, ORION from AspenTech). In the planning step, a discrete-time multiperiod MINLP model is used to optimize the total cost by computing the volume of each blend and its corresponding blend recipe (e.g., Honeywell BLEND). The term “blend recipe” refers to the volume fractions of the components used to blend one unit of a specific grade of gasoline during a time period. Blender operation is then scheduled via interactive simulation (e.g., ORION) within the constraints imposed by the solution of the multiperiod model. The planning model includes constraints w.r.t. minimum size of the blend, maximum blend rate, and component and product inventories. Cost of switching from blending one grade to another is included in the total cost, while the duration of the switch can be included as a reduction of the available blender capacity.

An advantage of decomposing the problem into planning and scheduling is a relative simplicity of solving the planning model. Disadvantages of the approach are:

1. Blend plan is guaranteed to be feasible only at the period boundaries; however, the operation may be infeasible at some intermediate point within a period.
2. Blend recipes vary from one period to another. In other words, if the blend planning horizon is 14 periods, typically there will be 14 different blend recipes for each grade of gasoline.

Variations in blend recipes require more decisions by the scheduler, making it more difficult to schedule operation with minimum number of switches. Hence, the current practice is to keep the blend recipes relatively constant, since the schedulers know that the optimal blending of the same gasoline grade by using the same recipe should be possible for extended lengths of time. In order to avoid meandering of the blend recipe, one can minimize deviations of blend recipes in every period from some preferred or average blend recipe. This strategy has been implemented by adding penalty terms to the objective function (e.g., Mendez et al.[2]). However, such formulation means that the optimum value of the objective function is a combination of the costs and the penalties for deviations from the average blend recipes, that is, the solution might not end up being the same real economic optimum of the unconstrained recipe problem. Another option to reduce recipe variation is to use preferred blend recipes (e.g., Jia and Ierapetritou[8]); in this case, the possibility to obtain the best solution depends on the selection of this set of preferred fixed recipes.

Since blend plan computed from a multiperiod model is feasible at the period boundaries and possibly infeasible within some time periods, Thakral and Mahalec[9] introduced a composite algorithm which computes blend recipes via multi-period MINLP and then uses a genetic algorithm to minimize the number of switches followed by agent-based simulation to detect infeasibilities, if they exist. If an infeasibility is encountered, the corresponding period is subdivided, the blend recipes are recomputed via MINLP, and the process is repeated until there are no infeasibilities.

Glismann and Gruhn[10] used a discrete-time multiperiod NLP to compute blend recipes and a discrete-time multiperiod MILP to solve the short term scheduling problem. The planning periods are defined by product demands and other specific planning priorities and the scheduling periods were defined to be 2 h long. If a feasible solution cannot be found for the scheduling problem or if the deviations from the goals cannot be accepted, the planning step is solved again but this time including the information from the MILP solution through the addition of constraints regarding the blend components consumption. The new blend recipes computed are added to the scheduling problem and can be chosen as alternatives to the old recipes. In order to avoid many recipe changeovers, constraints are added to enforce a minimum running time for particular recipes on a blender.

Joly and Pinto[11] presented a discrete-time MILP model to optimize the scheduling of fuel oil and asphalt production. They assumed linear blending properties and solved three real-world examples using a schedule horizon from 1 to 6 days, divided in uniform time periods of 2 h each. In one of their examples, feasible solutions with relative optimality gaps varying from 1.3 to 8.2% were found in less than 4 CPU hours. They reported that smaller relative gaps required more time and sometimes algorithm failure was observed.

An alternative to the discrete time representation is a continuous-time model. Jia and Ierapetritou[8] solved simultaneously gasoline blend scheduling and distribution. They decomposed the refinery system in three subsystems (1) the crude oil unloading and blending, (2) the production units, and (3) the product blending and shipping. They presented a continuous-time event-based MILP model for the scheduling problem of the third subsystem. The model includes multipurpose product tanks (tank switching), delivery of the same order from multiple product tanks, and one product tank delivering multiple orders. They solved realistic large-scale problems to global optimality in less than 5 CPU hours. According to Li et al.[12], Jia and Ierapetritou model may lead to infeasible solutions which have one tank holding more than one product at a time.

Mendez et al.[2] introduced both discrete-time and continuous-time models to optimize blend recipes and schedule operations using an iterative algorithm. The model includes nonlinear quality constraints and employs their linearization to formulate a MILP. After the blend recipes are computed, correction factors for the product properties are calculated. Algorithm stops when the correction factors converge and the products properties meet the specifications. Minimum blend run constraints and multipurpose tanks are not included in the models. Mendez et al.[2] and Kelly and Mann[13,14] have pointed out that solving logistics and quality aspects for large-scale problems required significant amount of time with then-current MINLP codes and global optimization techniques. Even though recent advances in MINLP and MILP codes have made it possible to solve much larger problems, there is a continuous research effort for improved formulations that lead to shorter solution times and/or more accurate results.

Li et al.[12] presented a continuous-time slot-based model that uses process slots to simultaneously consider computation of recipes, blender operation, inventory constraints and order scheduling. Quality constraints are handled using blend indices (i.e., they are linear constraints). Their MILP model includes parallel nonidentical blenders, multipurpose tanks and other attributes that describe systems encountered in practice. Their objective function does not include penalties to minimize deviations from a preferred or average blend recipe since their model computes a blend recipe for each specific blend run. In order to ensure a constant blend rate, they included a schedule adjustment step in their algorithm. While their model poses many of the characteristics of the industrial systems, computational times for large examples are still not reasonable; nevertheless, their solutions were better than those provided by DICOPT and BARON trying to solve the corresponding MINLP model. Li and Karimi[15] replaced process slots with unit slots and expanded the model by Li et al.[12] to include blender setup times, limited inventory of components and simultaneous receipt/delivery by the product tanks. Compared to work of Li et al.[12], they accomplish a significant reduction of computational times for smaller size problems. Their results show that some realistic-scale problems (2 to 3 blenders, 4 to 6 products, 9 components, 9 properties, 11 product tanks, 10 to 45 orders, and a planning horizon of 8 days) can be solved, but others were unsolvable to proven optimality within the allocated CPU times (10,800 to 108,000 s, depending of the problem).

This work introduces an algorithm designed to improve discrete-time approach to blend planning. Our goals are:

1. Maintain the blend recipe for each grade of gasoline as constant as possible across the planning horizon, that is, minimize the number of different blend recipes
2. Determine a blend plan that is feasible at the boundaries of time periods, thereby making it easier to produce a feasible schedule
3. Use nonlinear single-period blending models to arrive at solutions that have objective function value equal to (or very close to) the solutions obtained by the multi-period approach, thereby reducing the computational requirements of the problem.

Basis for the algorithm is a notion of an inventory pinch point. An inventory pinch point is defined as the point where the cumulative average total production curve, CATP curve, (which is based on optimal blend recipes that will meet the demands for various grades of gasoline) is tangent to the cumulative total demand curve, CTD curve, and if CATP curve is extrapolated from this point onward it will not cross the CTD curve (see Figure 5). The top level of the algorithm optimizes blend recipes, while the lower level of the algorithm computes blend profile (when and how much of each grade to blend) based on the recipes computed at the top level. The algorithm uses different period lengths at each level (see Figure 4). At the top level, periods are delimited by the inventory pinch points, while the lower level has fine-grid fixed-length periods. In order to distinguish between the periods at top and the lower level, we shall call them *t*-period and *l*-period, respectively If inventory infeasibilities (i.e., insufficient gasoline inventory) are detected in some *l*-period in the solution of the MILP, the corresponding *t*-period at the top level is subdivided; then the blend recipes are reoptimized at the top level and the lower level MILP is resolved. The objective at the top level is to minimize the total cost of blending. As explained earlier, the carrying costs of the finished products cancel out with the carrying costs of the blend components and need not be included in the objective function. The lower level MILP computes how much of each grade of gasoline to blend in each *l*-period. If the recipes computed at the top level lead to a feasible fine-grid production plan, then the objective function at the lower level is equal to zero.

The remainder of this article presents description of the problem addressed by this work, followed by the introduction of the inventory pinch concept and description of the single-period inventory pinch algorithm for blend planning. We present 12 case studies, comparing the results of our algorithm with those computed via multiperiod MINLP solved by DICOPT.

Problem Statement

As mentioned in the Introduction, the current practice in gasoline blending relies on discrete-time decomposition of the blending problem in two levels. At the top level, a multiperiod MILP or MINLP computes blend recipes, while the lower level computes (heuristically or algorithmically) the corresponding schedule. This work pursues improvements in the discrete-time formulation, with the intent to enhance the current practice.

We will use a sample system (Figure 1) to describe attributes of a gasoline blend planning model. Blend planning problem is described by:

Component supply and product demand data:

1. Planning horizon [0, H] divided into fixed-duration time periods 1,2,…N.
2. Set of blend components, their properties, initial inventories, costs, and flow rates along the planning horizon (i.e., supply profile).
3. Set of products (i.e., gasoline grades) with prescribed minimum and maximum quality specifications, their initial inventories and corresponding initial quality.
4. Set of delivery orders for each product along the planning horizon (i.e., demand profile). If there are multiple orders for the same grade of gasoline in a given time period, these orders are lumped into a total demand for that grade of gasoline in that period.

Blending system structure and operating characteristics considered:

1. Minimum and maximum inventories (for every time period) for each blend component and for each grade of gasoline. If there are multiple tanks available for storage of the same material, they are treated as one aggregate inventory capacity for that material.
2. There can be one or more blenders operating in parallel. Maximum blending capacity of each blender is given.
3. Blender capacity loss due to switching the blender from one grade to another.
4. If a specific grade of gasoline is to be blended, then the amount blended must be greater than or equal to some threshold amount.

We need to compute:

1. Which gasoline grades to produce in each time period and their respective volumes.
2. How much of each blend component to use for each gasoline blend (blend recipes) for each grade and for each period.
3. Inventory profiles of the components and of finished gasoline.

Assumptions are:

1. Refinery production plan has determined the crude feed rate as required to meet the demand for various products. Hence, gasoline blend components are available in required quantities to meet the total gasoline demand during any time period of the production plan.
2. Component qualities are constant for the whole planning horizon. Since blend components are produced by upstream units which are operated to meet target qualities of the blend components, this assumption is valid so long as the process units produce blend components to their target qualities.
3. Quality constraints are either linear or nonlinear. For simplicity, we include only RVP nonlinear constraint in some case studies.
4. Perfect mixing occurs in the blender.
5. Changeover times between product runs on the blender are product and sequence independent.
6. Each order involves only one product.
7. Each order is completed during the planning horizon.
8. Product liftings (demands) need to be met.
9. If the initial inventory (heel) of some product tank is off-spec, it will be used as blend component to produce on-spec gasoline.
10. Blend planning horizon is typically 1 to 2 weeks long with periods being 1 day or half a day in duration.

Note that the above multiperiod model implies:

1. Within each time period, the products required during that period are first produced in the required quantities (if they are not available in the storage tanks), and then the products are shipped (“lifted”) in the required quantities.
2. Product inventory tank may receive new blend while gasoline is being withdrawn from the tank.
3. Component inventory tank may receive additional amounts of the component while the material is being withdrawn from the tank and sent to the blender.

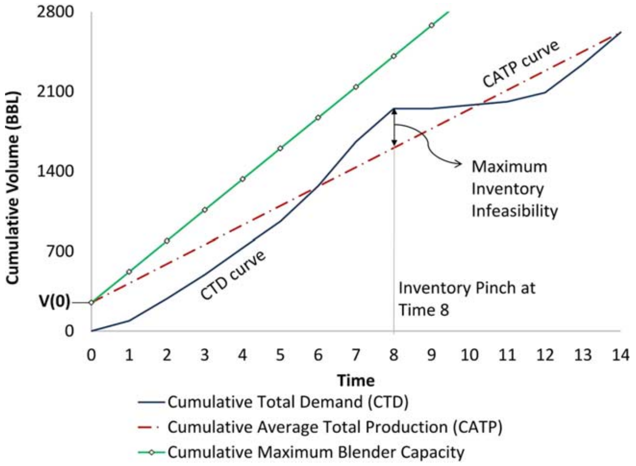

Figure 2. Problem with one pinch point.

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

The Inventory Pinch Concept

For simplicity's sake, let us consider a single product gasoline blender, i.e., imagine the system shown in Figure 1 but having only one blender and one product. In order to simplify exposition and the graphs, we will use data corresponding to fixed time periods, where a "fixed time period" is specified by its start and end time. This does not impact the method or the conclusions. We can plot cumulative demand curve (CTD) vs. time. Note that this curve is constructed from actual demand data (e.g., a shipment can start 3 days and 7 h or at some other time from the start of the planning horizon), i.e., CTD does not have to use aggregate data per day. Cumulative total production (CTP) consists of the available initial product inventory V(0), and the amount produced up to a specific point in time If we assume that the blending will be carried out at the same average rate throughout the horizon, then the cumulative average total production (CATP) is a straight line connecting the initial available inventory to the final total demand. Clearly, in the example shown in Figure 2, blending at average rate for the entire horizon is not feasible (cumulative demand curve CTD becomes greater than the cumulative average total production CATP starting in period 6). In order to have a feasible operation, cumulative average total production (CATP) must always be greater than or equal to the cumulative demand (CTD). Hence, a feasible blending operation requires that the cumulative average total production curve CATP traverse from the initial available inventory V(0) to the point where it touches the cumulative total demand curve CTD and then to the final point on CTD curve (see Figure 3). Inventory pinch points are the points where cumulative average production curve CATP becomes tangent to the cumulative total demand curve CTD. Shown in Figure 2 is also the maximum possible product volume, which corresponds to the maximum available blender capacity.

One lower bound on the optimum solution is obtained by solving the total aggregate blending problem (Eqs. 1–9). This solution is a single blend recipe (per product) to be applied along the entire horizon; hence the entire horizon is considered as a "fixed recipe interval". If the blending occurs at the same blend rate throughout the horizon, then the blender operates at constant rate equal to the average rate across the horizon. Corresponding to the average blend rate is the average available product volume, represented by CATP line. Since this is infeasible, we need to increase the blend rate until the cumulative product volume line becomes a tangent to the cumulative total demand curve (segment CATP1 in Figure 3), i.e., we need to produce sufficient amount of gasoline to meet the cumulative demand at the pinch point. In the example shown in Figure 3, product blending in the interval from the pinch point to the end of the horizon proceeds along CATP2 segment. If blend components are available in sufficient quantities at the appropriate points along the horizon, and if we do not encounter any inventory constraints along the horizon, then a single blend recipe can be used to blend the volumes corresponding to the cumulative product curve (CATP1 and CATP2 segments in Figure 3). Whether or not such operation is feasible can be determined by solving a multiperiod MILP which uses the aggregate blend recipe (which is constant for all periods) to determine how much to blend in each period.

If blending with fixed recipe across the entire horizon is not feasible, the infeasibility is caused by insufficient amount of blend components required for blending based on optimal recipe while meeting the demand at the pinch points. Therefore, we need to compute the aggregate blend recipe from the start of the horizon to the first inventory pinch point (period 1). This recipe will make the best possible use of the components available in period 1. Similarly, another aggregate blend recipe needs to be computed from the pinch point to the end of the horizon (or to the next pinch point, if there are more than one), as shown in Figure 3.

If the amount of available blend components and their qualities in period 1 are such that the cumulative demand at the pinch point cannot be met, then the problem is infeasible. If the product unit cost corresponding to the blend recipe computed for interval is higher than the aggregate blend recipe for the entire horizon, this is due to the fact that the mix of components available during this period does not allow us to blend at the lowest possible cost.

From here on, the term *t*-period (i.e., top-level period) will be used as the length of time where one aggregate recipe will be computed (i.e., aggregate interval). A *t*-period contains at least one *l*-period from the lower level (see Figures 3 and 4).

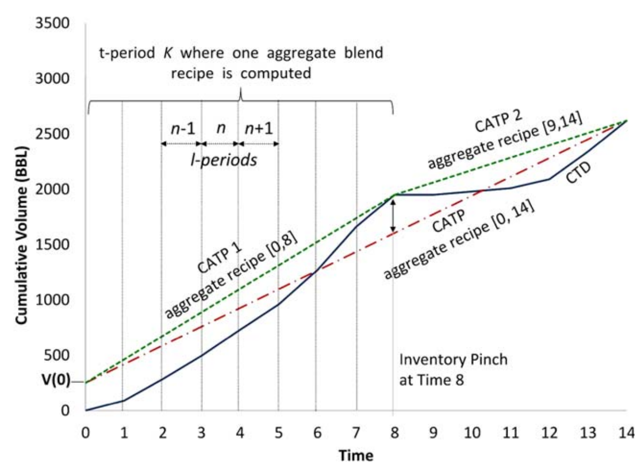

Figure 3. Graphic representation of the inventory pinch algorithm.

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

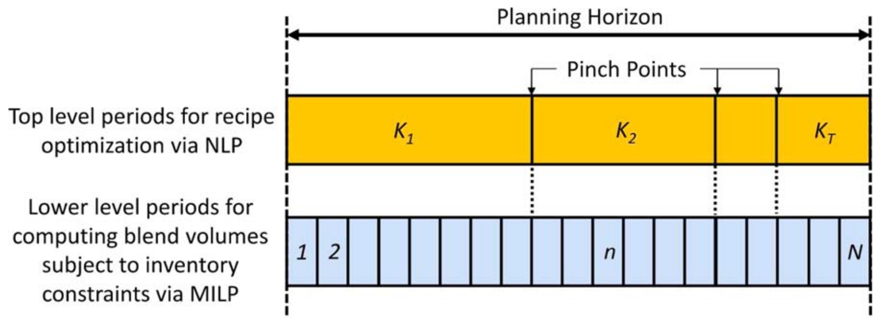

Figure 4. Time grids for the two levels of the algorithm.

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

It should be pointed out that cumulative average total production curve CATP should be constructed as shown by segment CATP2 in Figure 5. End point of a segment of the CATP curve must be a tangent to the cumulative demand curve CTD and if CATP segment is extended beyond the tangent point, the extension must not intersect the cumulative demand curve. If we solve aggregate blending problem corresponding to CATP2, the solution includes the "local" inventory pinch in 3. Hence, solving for an aggregate blend recipe for CATP1 is not necessary. Figure 6 shows an example with two true inventory pinch points.

When introducing inventory pinch, we assumed that there is only one product being blended. That assumption does not alter the methodology, if we (1) define cumulative demand curve to be the cumulative total demand (CTD) of all products, and (2) define cumulative average total production (CATP) to be the sum of all product volumes.

Having computed aggregate blend recipe for each t-period delimited by the pinch points, we still need to determine how much of each gasoline grade to produce at what point in time. For this purpose, we formulate a multiperiod (l-periods) fixed-recipe volume-only MILP model. Purpose of the optimization is to find the blend volumes for each l-period in such a manner that possible infeasibilities in the product inventories are pushed as far into the future as possible. Constraints are fixed blend recipe equations, inventory constraints on the product and component tanks, product demands, and blend component availability. Volumetric constraints on the inventory tanks include nonnegative positive and negative slack variables. Cost coefficients for the product inventory tanks are decreasing as we progress along the time horizon, thereby "pushing" any possible product inventory infeasibility as far forward as possible. In addition, one may include minimum threshold blend amount for each grade and capacity lost due to the set-up time when switching between the grades.

If blend components are not available as required by the blend recipe for a given t-period, then the solution of the multi-period volume-only MILP model will have infeasible product inventories in some l-periods which are contained within that t-period. This can be seen directly on the product inventory profiles and also in the cumulative curves (Figure 7). The algorithm presented in this work eliminates these infeasibilities via an iterative procedure.

Conceptual Use of Inventory Pinch for Gasoline Blend Planning

Refinery production planning optimizes a multiperiod model comprised of periods that are weeks or/and months long. Resulting production plan specifies how much of various blend components will be produced in each period to ensure that the required product amounts will be possible to blend. If ratio of components in the blend pool supply does not vary a lot from one day to another, which is true most of the time (unit shutdowns being an exception), then one can proceed to compute blend recipes by solving independently

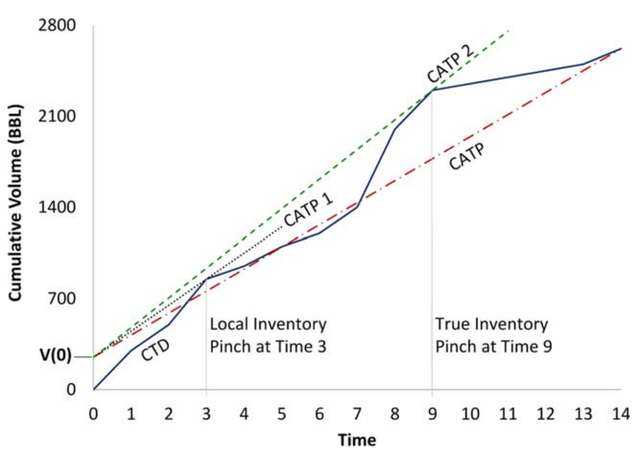

Figure 5. One "local" inventory pinch and one true inventory pinch.

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

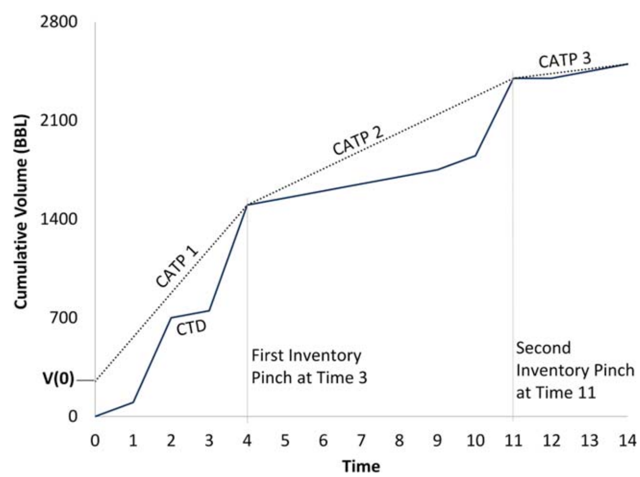

Figure 6. Example with two true inventory pinch points.

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

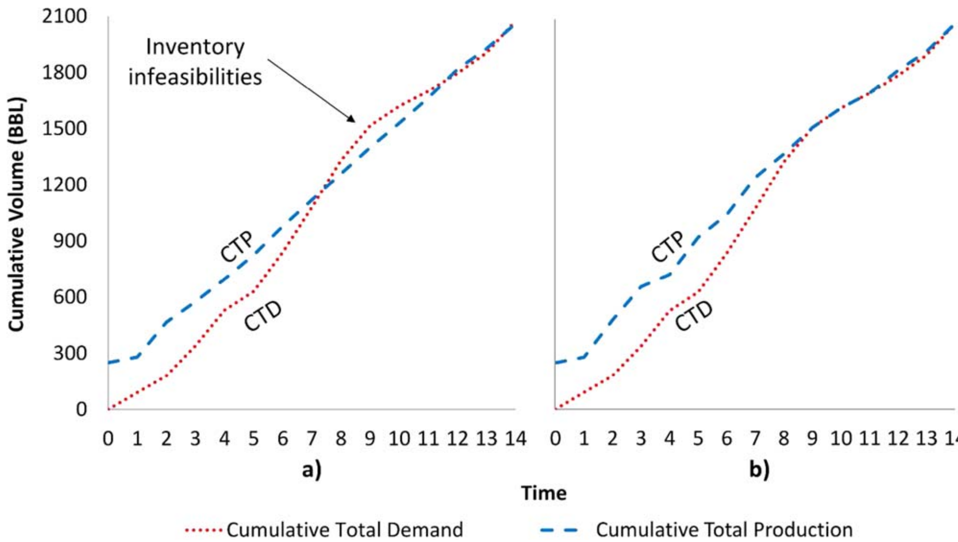

Figure 7. Inventory infeasibilities elimination by inventory pinch algorithm

(a) Infeasible aggregate recipes, and (b) feasible aggregate recipes. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

aggregate blend models for periods between the pinch points. The recipes are computed by moving forward along the time horizon, from one period to another.

An alternative approach is to formulate a multiperiod NLP or MINLP where period boundaries correspond to inventory pinch points. Since our goal is to solve as simple as possible nonlinear problems, we present a single period inventory pinch algorithm which is described in detail in the following section.

Single Period Inventory Pinch Algorithm for Gasoline Blend Planning

The inventory pinch algorithm decomposes the gasoline blend planning in two levels (Figure 8), each one based on a different view of the blending process:

Top level: Computation of aggregate blend recipes (quality constraints fulfillment) that leads to the lowest possible cost per unit of the volume blended, subject to availability of the components in a given t-period.

Lower level: Computation of volumes to be blended in each l-period while meeting the inventory, demand, and blend threshold constraints. The blend recipe computed for a t-period is fixed for all the l-periods that it contains.

At the top level, we divide the planning horizon in K_T t-periods delineated by the inventory pinch points. Blend recipes between inventory pinch points are often constant. The model at the top level does not consider variations of component inventories over the horizon only the overall component availability. As mentioned previously, since gasoline blend planning is carried out within the constraints computed by the longer range production plan, the inventory carried by the refinery is determined by that plan. Hence, the refinery carries either the inventory of blending components or the inventory of blended gasoline. Over the gasoline blend planning horizon, the carrying costs of these inventories are fixed and need not to be included in the objective function. Once blend recipes are known, we compute how much of each grade to blend across the entire blend planning horizon via fixed-recipe volume-only MILP model. Length of l-periods (total of N) in the MILP model is selected in a manner that makes it simple to make decisions at the scheduling level (e.g., one day, half a day, or shorter). If the blend recipes from the top level lead to a state where at some point along the horizon there are insufficient amounts of the blend components, then infeasible product inventories (denoted as inventory infeasibilities) will appear in the lower level. Formulation of the MILP is such that any potential infeasibility is "pushed" forward, as far as possible, in the planning horizon. If product infeasibilities are found in the lower level for a specific l-period, it means that either the blender capacity is too small or that the amount of components available in the l-period is insufficient. In either case, it means that an additional amount of the product needs to be blended prior to the infeasibility; that additional amount is equal to the infeasible volume for that product. If there are any infeasibilities, then the t-period at the top level is subdivided and new recipes are computed (i.e., K_T increments in each iteration). In our case studies, under nondecreasing component supply, the algorithm leads to the optimum

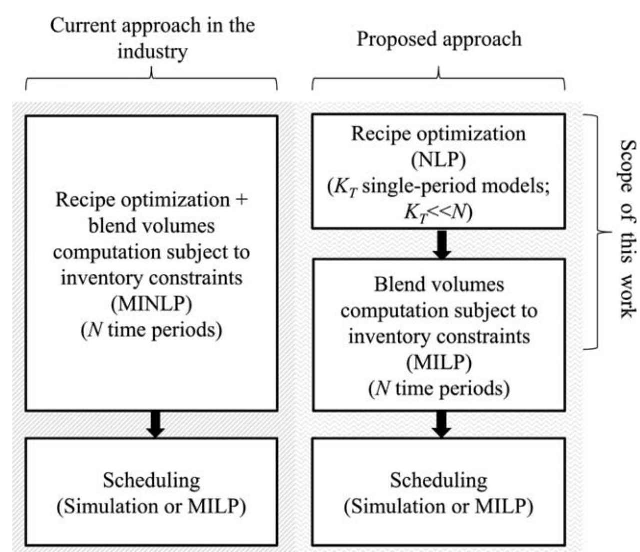

Figure 8. Proposed decomposition of gasoline blending.

solution computed via multiperiod MINLP. If there are significant variations in component supply, then the algorithms produces solutions that are very close to the optimum.

Top level: Computing aggregate blend recipes

The blend recipes are computed solving an aggregate NLP model (Eqs. 1–9) for each *t*-period. The aggregate models are solved in sequence (solution from one *t*-period define the starting point of the next *t*-period). If the aggregate model is solved for the whole planning horizon, this is referred as the *total aggregate model* and its solution is a lower bound of the problem. If the total aggregate model is feasible, we then proceed to compute the blend recipes for periods delimited by the pinch points. We include non-negative inventory slack variables on the inventory constraints which will be nonzero if there are infeasibilities. Table A1 explain the notation used in this article.

Equation 1 is the objective function to minimize, it comprises of the components cost, and the penalties for the inventory slack variables ($\text{Penalty}_{\text{P}}(g) >> \text{Penalty}_{\text{C}}(i)$)

$$\min\left\{\begin{array}{l}\sum_g\sum_i V_{C,K}(i,g)\times\text{Cost}(i)+\sum_i\left(S^+_{C,K}(i)+S^-_{C,K}(i)\right)\times\text{Penalty}_C(i)\\ +\sum_g\left(S^+_{P,K}(g)+S^-_{P,K}(g)\right)\times\text{Penalty}_P(g)\end{array}\right\} \tag{1}$$

Equation 2 represents the volume balance around a component tank. Equation 3 ensures that the closing inventory of each component tank is within its limits

$$\text{V}^{\text{close}}_{C,K}(i)=\text{V}^{\text{open}}_{C,K}(i)+\text{V}^{\text{in}}_{\text{C,K}}(i)-\sum_g \text{V}_{\text{C,K}}(i,g) +S^+_{C,K}(i)-S^-_{C,K}(i)\quad \forall i \tag{2}$$

$$\text{V}^{\min}_{\text{C}}(i)\le \text{V}^{\text{close}}_{\text{C,K}}(i)\le \text{V}^{\max}_{\text{C}}(i)\quad\forall i \tag{3}$$

Equation 4 is the volume balance around a product tank, and Eq. 5 represents the inventory constraints

$$\text{V}^{\text{close}}_{\text{P,K}}(g)=\text{V}^{\text{open}}_{\text{P,K}}(g)+\text{V}_{\text{B,K}}(g)-D_{\text{P,K}}(g)+\text{S}^+_{\text{P,K}}(g)-\text{S}^-_{\text{P,K}}(g)\quad\forall g \tag{4}$$

$$\text{V}^{\min}_{\text{P}}(g)\le \text{V}^{\text{close}}_{\text{P,K}}(g)\le \text{V}^{\max}_{\text{P}}(g)\quad\forall g \tag{5}$$

Please note that the slack variables in Eqs. 2 and 4 are introduced to ensure that the solver always returns a solution. If any of the slack variables are nonzero, the solution is physically unrealizable, since the inventory constraints are violated. We use separate variables for opening and closing inventories to have a representation that is repeatable from one period to another. The volume balance around the blender is given by 6 and 7. We include the fraction of the components in the product (i.e., we compute the blend recipe directly) since the volume of product to be blended is known and because Eq. 9 requires these fractions

$$\text{V}_{\text{C,K}}(i,g)-x_K(i,g)\cdot \text{V}_{\text{B,K}}(g)=0\quad\forall g,i \tag{6}$$

$$\sum_i \text{X}_{\text{K}}(i,g)=1\quad\forall g \tag{7}$$

When the initial product inventory is on-spec and the quality constraints are assumed to be linear, the quality balance is given by Eq. 8

$$\text{Q}^{\min}_{\text{P}}(g,s)\cdot\text{V}_{\text{B,K}}(g,k)\le\sum_{\text{i}}\text{Q}_{\text{C}}(i,s)\cdot\text{V}_{\text{C,K}}(i,g,k)\le\text{Q}^{\max}_{\text{P}}(g,s)\cdot\text{V}_{\text{B,K}}(g,k)\quad\forall g,s \tag{8}$$

The Reid vapor pressure (RVP) is a property that blends nonlinearly. Singh et al.[16] used the blending index approach to compute the RVP choosing Eq. 9 based on its predictive accuracy, parsimony, and ease of implementation. We will use the same equation in case studies 5 to 10 (we will consider the RVP to blend linearly in the other cases)

$$Q_{P,K}(g,s{=}\text{RVP})=\left[\sum_{i=1}^{I}x_K(i,g)\cdot\left(Q_C(i,s{=}\text{RVP})\right)^{1.25}\right]^{0.8}\quad\forall g \tag{9}$$

Equations 1–9 define the aggregate model for the case when the initial inventory is on-spec. Equations 10–12 are required to provide parameters to the aggregate model and they need to be solved before. Equations 10 and 11 are required to link the inventories of adjacent *t*-periods. Eq. 12 will set the volume to be blended in the *t*-period equal to the volume plus the inventory infeasibilities given by the multiperiod solution. Basically, Eq. 12 links the aggregate model with the multiperiod model

$$\text{V}^{\text{open}}_{\text{C,K}}(i)-\text{V}^{\text{close}}_{\text{C,K-1}}(i)=0\quad\forall i,K=K_2,\ldots,K_T \tag{10}$$

$$\text{V}^{\text{open}}_{\text{P,K}}(g)-\text{V}^{\text{close}}_{\text{P,K-1}}(g)=0\quad\forall g,K=K_2,\ldots,K_T \tag{11}$$

$$\text{V}_{B,K}(g)=\sum_{n\in K}\left(\text{V}_B(g,n)+\text{S}^+_{\text{P}}(g,n)-\text{S}^-_{\text{P}}(g,n)\right)\quad\forall g \tag{12}$$

If the initial product inventory is off-spec, the inventory needs to be brought back to specification before it can be shipped/lifted. In order to do this, the off-spec heel needs to be considered as a blend component by introducing Eqs. 13 and 14. The subscript "0" specifies the variables during this quality correction *t*-period K_0. The length of this *t*-period K_0 must be enough to bring back the product inventories into specification

$$\sum_{\text{i}}\text{Q}_{\text{C}}(i,s)\cdot\text{V}_{\text{C,0}}(\text{i},g)+\text{Q}^{\text{open}}_{\text{P,0}}(g,s)\cdot\text{V}^{\text{open}}_{\text{P,0}}(g)\ge\text{Q}^{\min}_{\text{P}}(g,s)\cdot\left(\text{V}^{\text{open}}_{\text{B,0}}(g)+\text{V}^{\text{open}}_{\text{P,0}}(g)\right)\quad\forall g,s \tag{13}$$

$$\sum_{\text{i}}\text{Q}_{\text{C}}(i,s)\cdot\text{V}_{\text{C,0}}(\text{i},g)+\text{Q}^{\text{open}}_{\text{P,0}}(g,s)\cdot\text{V}^{\text{open}}_{\text{P,0}}(g)\ge\text{Q}^{\max}_{\text{P}}(g,s)\cdot\left(\text{V}_{\text{B,0}}(g)+\text{V}^{\text{open}}_{\text{P,0}}(g)\right)\quad\forall g,s \tag{14}$$

For the rest of the planning horizon (subscript "1") the heel is not required to be considered a blend component since it is already on-spec. Therefore, the quality balance is written as

$$Q_P^{min}(g,s)\cdot V_{B,1}(g) \le \sum_i Q_C(i,s)\cdot V_{C,1}(i,g) \le Q_P^{max}(g,s)\cdot V_{B,1}(g) \quad \forall g,s \tag{15}$$

The volume balance around the blender is given by Eqs. 16 and 17 for the first *t*-period when the heel is corrected and for the rest of the planning horizon, respectively. Equations 18 and 19 set the closing product inventories to be within the limits

$$V_{P,0}^{open}(i) - V_{P,0}^{close}(g) + \sum_i V_{C,0}(i,g) - D_{P,0}(g) + S_{P,0}^+(g) - S_{P,0}^-(g) \quad \forall g \tag{16}$$

$$V_{P,1}^{open}(g) - V_{P,0}^{close}(g) + \sum_i V_{C,1}(i,g) - D_{P,1}(g) + S_{P,1}^+(g) - S_{P,1}^-(g) \quad \forall g \tag{17}$$

$$V_P^{min}(g) \le V_{P,0}^{close}(g) \le V_P^{max}(g) \quad \forall g \tag{18}$$

$$V_P^{min}(g) \le V_{P,1}^{close}(g) \le V_P^{max}(g) \quad \forall g \tag{19}$$

Equations 20 and 21 identify the volume used during the heel correction ($V_{C,0}$) and the volume used afterwards ($V_{C,1}$)

$$V_{C,0}^{close}(i) = V_{C,0}^{open}(i) + V_{C,0}^{in}(i) - \sum_g V_{C,0}(i,g) + S_{C,0}^+(i) - S_{C,0}^-(i) \quad \forall i \tag{20}$$

$$V_{C,1}^{close}(i) = V_{C,0}^{close}(i) + V_{C,1}^{in}(i) - \sum_g V_{C,1}(i,g) + S_{C,1}^+(i) - S_{C,1}^-(i) \quad \forall i \tag{21}$$

Equation 1 is replaced by Eq. 22. *XC* is a coefficient set to a value greater than 1 and much less than the penalties for the slack variables (in this work, $XC = 10$). It forces the off-spec heel to be corrected with the smallest possible blend volume and, therefore, to be ready to be lifted/shipped as soon as possible

$$\min \left\{ \begin{array}{l} XC \cdot \left(\sum_g \sum_i V_{C,0}(i,g) \cdot \mathrm{Cost}(i) \right) + \sum_g \sum_i V_{C,1}(i,g) \cdot \mathrm{Cost}(i) \\ + \sum_i \left(S_{C,0}^+(i) + S_{C,0}^-(i) + S_{C,1}^+(i) + S_{C,1}^-(i) \right) \cdot \mathrm{Penalty}_C(i) \\ + \sum_g \left(S_{P,0}^+(g) + S_{P,0}^-(g) + S_{P,1}^+(g) + S_{P,1}^-(q) \right) \times \mathrm{Penalty}_P(g) \end{array} \right\} \tag{22}$$

Then, the aggregate model for the case when the initial product inventory is off-spec is defined by Eqs. 13–22.

Lower level: Computing volumes to be blended in each time period

After the aggregate blend recipes are defined, it is required to calculate the volumes to be blended in each *l*-period. This step of the algorithm is modeled as a MILP (Eqs. 23–37) to set constraints on the minimum and maximum amount of product to be blended in each *l*-period (threshold constraints). Objective function, Eq. 23, is constructed in such a manner that the solution will delay occurrence of potential inventory infeasibilities as far into the future as possible. The objective function contains only the penalties for the inventory slack variables. If the blend recipes computed at the top level correspond to a feasible operation, the inventory slack variables will be zero at the solution of MILP. If the blend recipes are infeasible, then the MILP solution will show which specific products and in which *l*-periods cannot be produced in the amounts required to meet the demands. The penalty for the components' inventory slack variables is much greater than the penalty for the products' inventory slacks ($\mathrm{Penalty}_C(i) >> \mathrm{Penalty}_P(g,n) \quad \forall i,g,n$), which forces the inventory infeasibilities to be on the products' side. Such formulation allows us to know how much additional volume for the product is required to meet the inventory constraints (if there is infeasibility) and the aggregate model can compute the proper blend recipe. In addition, the penalty for a product inventory slack decrease with time ($\mathrm{Penalty}_P(g,n) > \mathrm{Penalty}_P(g,n+1) \quad \forall g,n$) to move the infeasibilities as late as possible in the planning horizon. The blend plan cost is computed but not included in the objective function since fixed blend recipes and product target inventories are defined by the solution of the aggregate model. Since we have not included the cost of switching in the MILP model, the optimum solution of this model is 0.0 (i.e., slack variables are zero for a feasible solution; therefore, their use in this model does not affect the solution obtained as in the case where they are used to minimize deviations between blend recipes). If the solution of the MILP has inventory infeasibilities, it signifies that either component supply or product demand are such that the recipes computed at the top level are not feasible within a *t*-period. The algorithm will subdivide such *t*-periods and reoptimize the blend recipes

$$\min \left\{ \sum_n \left(\begin{array}{l} \sum_i \left(S_C^+(i,n) + S_C^-(i,n) \right) \times \mathrm{Penalty}_C(i) \\ + \sum_g \left(S_P^+(g,n) + S_P^-(g,n) \right) \times \mathrm{Penalty}_P(g,n) \end{array} \right) \right\} \tag{23}$$

The volume balance equations and inventory constraints for the component and product tanks for every *l*-period *n* are given by 24 to 27

$$V_C^{close}(i,n) = V_C^{open}(i,n) + V_C^{in}(i,n) - \sum_{bl} \sum_g V_C(i,g,n,bl) + S_C^+(i,n) - S_C^-(i,n) \quad \forall i,n \tag{24}$$

$$V_C^{min}(i) = V_C^{close}(i,n) \le V_C^{max}(i) \quad \forall i,n \tag{25}$$

$$V_P^{close}(g,n) = V_P^{open}(g,n) + \sum_{bl} V_B(g,n,bl) - D_P(g,n) + S_P^+(g,n) - S_P^-(g,n) \quad \forall g,n \tag{26}$$

$$V_P^{min}(g) \le V_P^{close}(g,n) \le V_P^{max}(g) \quad \forall g,n \tag{27}$$

Equations 28 and 29 set the closing inventories as the opening inventories of the next *l*-period *n*

$$V_C^{open}(i,n+1) - V_C^{close}(i,n) = 0 \quad \forall i,\ n=1,\ldots,N-1 \tag{28}$$

$$V_P^{open}(g,n+1) - V_P^{close}(g,n) = 0 \quad \forall g,\ n=1,\ldots,N-1 \tag{29}$$

The balance around the blender is given by Eqs. 30–35. The blend recipe is fixed within the *l*-period boundaries $\pm 1 \times 10^{-7}$ to avoid finding very small infeasibilities due to numerical discrepancy between the two levels of the algorithm. Equation 35 indicates that the number of grades that can be blended in one *l*-period in blender *bl* must be less than or equal to the parameter *np*(*bl*). If the binary variable A(*g*,*n*,*bl*) is 1, the volume to blend of grade *g* in blender *bl* will be between the minimum and maximum specified limits, and less than or equal to the maximum blending rate. Since we are using a discrete-time formulation, t(*n*) is known in advance. If A(*g*,*n*,*bl*) is 0, then product *g* will not be blended in *l*-period *n* in blender *bl*. The lost volume term due to setup times for switching from blending one grade to another is included in Eq. 34, and it is equal to the maximum blender capacity times the setup time

$$V_C(g,i,n,bl) - \left(x(g,i,n) + 1\times10^{-7}\right)\cdot V_B(g,n,bl) \le 0 \quad \forall g,i,n,bl \tag{30}$$

$$V_C(g,i,n,bl) - \left(x(g,i,n) + 1\times10^{-7}\right)\cdot V_B(g,n,bl) \ge 0 \quad \forall g,i,n,bl \tag{31}$$

$$V_B^{min}(g)\cdot A(g,n,bl) - V_B(g,n,bl) \le 0 \quad \forall g,n,bl \tag{32}$$

$$V_B^{min}(g)\cdot A(g,n,bl) - V_B(g,n,bl) \ge 0 \quad \forall g,n,bl \tag{33}$$

$$\sum_g V_B(g,k,bl) + V_B^{lost}(bl)\cdot\sum_g A(g,n,bl) - F_B^{max}(bl)\cdot t(n) \le 0 \quad \forall n,bl \tag{34}$$

$$0 \le \sum_g A(g,n,bl) \le np(bl) \quad \forall n,bl \tag{35}$$

Equations 36 and 37 link the multiperiod model with the aggregate model by fixing the blend recipes in the *l*-periods *n* within the *t*-period *K*, and by setting the target closing inventories of the *l*-periods that are at the boundary end of the *t*-periods

$$x(i,g,n) = x_K(i,g) \quad \forall g,i,n\in K \tag{36}$$

$$V_P^{close}(g,n) = V_{P,K}^{close}(g) \quad \forall g,n \text{ at the end of } K \tag{37}$$

The economic cost is minimized at the top level by computing the appropriate blend recipes. Note that the multi-period MILP model does not deal with the blend recipe optimization and does not include the nonlinear terms. Thus, all the nonlinearities intrinsic to the blending problem are solved in the aggregate model. The corresponding MINLP model can be obtained by appending Eqs. 6–9 (for each *l*-period) to this MILP model, and if the initial product inventory is off-spec, Eq. 8 is replaced only in the first *l*-period by Eqs. 13 and 14.

Algorithm Statement

The algorithm presented in this section uses gasoline blend planning as the underlying example. Since none of the algorithm attributes are specific to gasoline blending, the algorithm can be used for any multipurpose plant that has structural characteristics similar to gasoline blending. Figure 9 shows the inventory pinch algorithm flowchart.

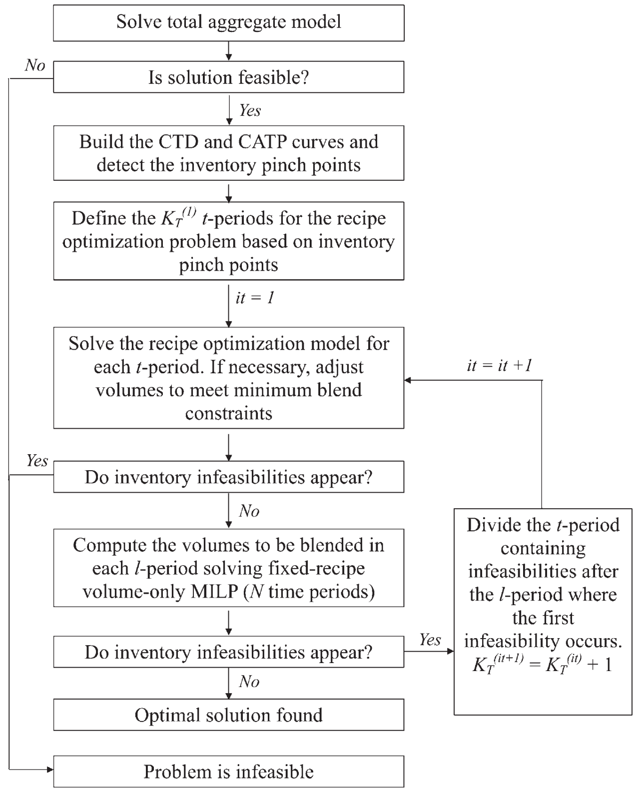

Figure 9. Inventory pinch single period algorithm.

Single Period Inventory Pinch Production Planning Algorithm (SPIP algorithm)

- Step 1: If the initial product inventory is on-spec, solve the aggregate model given by Eqs. 1 to 9 for the whole planning horizon. If the initial product inventory is off-spec, solve the aggregate model given by Eqs. 13 to 22. If the solution is feasible, continue with Step 2 (if the initial product inventory is off-spec, the blend recipe and volumes to be blended in K_0 are stored); otherwise, the blend components supply rates must be modified in order to fulfill the demand, that is, refinery production plan has created infeasible constraints.
- Step 2: Construct the Cumulative Total Demand (CTD) and the Cumulative Average Total Production (CATP) curves. Determine the pinch point(s) location.
- Step 3: Divide the planning horizon (at the top level) in the number of *t*-periods indicated by the pinch points (i.e., $K_T^{(1)}$). Set iteration counter $it = 1$.
- Step 4: Solve each one of these *t*-periods separately as single period problems using Eqs. 1 to 9, and Eqs. 10 to 12 to link them.

○ In the first iteration ($it = 1$), the volumes to produce of each product in each *t*-period *K* are the amounts that will make the final product inventories to be at the minimum limits.

○ In following iterations ($it > 1$), the volumes to produce are defined according to the solution of the volume allocation problem (Step 5). If necessary, the volumes to blend are adjusted as follows:

Table 1. Components Data (Properties, Cost, Supply Rates and Inventory Limits)

Components	ALK	BUT	HCL	HCN	LCN	LNP	RFT
ARO (%vol aromatics)	0	0	0	25	18	2.974	74.9
BEN (%vol benzene)	0	0	0	0.5	1	0.595	7.5
MON	93.7	90	79.8	75.8	81.6	66	90.8
OLF (%vol olefin)	0	0	0	14	27	0	0
RON	95	93.8	82.3	86.7	93.2	67.8	103
RVP (psi)	5.15	138	22.335	2.378	13.876	19.904	3.622
SPG	0.703	0.584	0.695	0.791	0.744	0.677	0.818
SUL (%vol sulfur)	0	0	0	0.485	0.078	0.013	0
Cost ($/BBL)	29.2	11.5	20	22	25	19.7	24.5
Minimum Inventory ($\times 10^3$ BBL)	5	5	5	5	5	5	5
Maximum Inventory ($\times 10^3$ BBL)	150	75	50	50	150	100	150
Initial Inventory ($\times 10^3$ BBL)	20	20	20	10	50	30	30
Supply Rate ($\times 10^3$ BBL/day) Cases 1 – 8, 11 – 12	25	5	3	5	25	20	50

■ If the volume to be blended in *t*-period K is greater than the maximum blender capacity, the blender at *t*-period K needs to work at full capacity and the remaining volume must be blended in the previous *t*-period.

■ If the volume to be blended in *t*-period K is less than the minimum allowed, the difference to reach the minimum will be taken from the next *t*-period. The exception is when K is the last *t*-period; in this case, the volume must be blended in the previous *t*-period.

- Step 5: Solve the volume allocation problem (Eqs. 23 to 37) for the entire horizon as one MILP, which uses fixed recipes computed in Step 4 in the corresponding *l*-periods.
- Step 6: If the inventory slack variables from Step 5 are zero, a feasible set of blend volumes, based on optimal recipes computed at the top level, has been found. Stop, since the optimal production plan has been found. Otherwise, continue to Step 7.
- Step 7: The planning horizon (at the top level) is divided as follows:
- If a *t*-period K does not contain inventory infeasibilities in the solution from Step 5, it is unchanged.
- If a *t*-period K contains infeasibilities in the solution from Step 5, it is divided into two new time periods; the first new *t*-period ends after the *l*-period n where the first infeasibility was detected.
- Step 8: $K_T^{(it+1)} = K_T^{(it)} + 1$. $it = it + 1$. Go back to Step 4.

Case Studies

All case studies have been computed on a HP Pavilion dv6 Notebook PC, AMD A8–3510MX APU processor, 1.80GHz, Windows 7 OS and 8 GB RAM. GAMS IDE 23.7.3 was used to solve each one of the case studies. The aggregate LP/NLP models were solved using IPOPT, and the MILP model was solved using CPLEX 12.3. To compare the results obtained with the inventory pinch algorithm, the corresponding MINLP model was solved using DICOPT.

The gasoline blending system shown in Figure 1 with either one or two blenders has been studied. Each blender can produce all products (i.e., $np(bl) = 3$). In the cases of one blender, the maximum blend capacity is 200×10^3 BBL/day; for the two-blender cases, blender A and blender B have a maximum blend capacity of 120×10^3BBL/day and 80×10^3 BBL/day, respectively. The minimum blend allowed for one product in each blender (i.e., V_B^{min}) is 30×10^3 BBL/day. For both blenders, the volume lost per switch (i.e., V_B^{lost}) is 8×10^3 BBL. In all cases, the system uses seven blend components (ALK, BUT, HCL, HCN, LCN, LNP, and RFT) and produces three products (grades of gasoline U87, U91, and U93). Each component and each product have their particular storage tank. A planning horizon of 14 one-day *l*-periods was considered for all case studies ($H = 14$ days, $t(n) = 1$ day for all n). The demand orders involve a single product and their time windows are assumed to be 1 day. Eight blend properties are considered: aromatic content (% by volume, ARO), benzene content (% by

Table 2. Supply Rate of Components along Planning Horizon—Case 9 and 10

Case Study	9							10						
l-period	$\times 10^3$ BBL/day													
Component	ALK	BUT	HCL	HCN	LCN	LNP	RFT	ALK	BUT	HCL	HCN	LCN	LNP	RFT
1	25	5	0	5	20	30	50	25	5	3	5	25	20	50
2	25	5	0	5	25	30	50	25	5	3	5	25	20	50
3	30	5	0	5	25	30	25	25	5	3	5	25	20	50
4	30	4	3	6	25	20	25	20	4	0	3	30	25	60
5	30	4	3	6	30	20	50	20	4	0	3	30	25	60
6	25	4	3	6	30	20	50	20	4	0	3	30	25	60
7	25	3	3	7	30	20	60	25	5	3	5	25	20	50
8	20	5	3	6	30	20	60	25	5	3	5	25	20	50
9	20	5	3	5	30	10	60	30	6	6	7	20	15	40
10	20	6	4	4	30	10	60	30	6	6	7	20	15	40
11	25	6	5	4	25	10	50	30	6	6	7	20	15	40
12	25	6	5	4	25	20	50	25	5	3	5	25	20	50
13	25	6	5	4	25	20	50	25	5	3	5	25	20	50
14	25	6	5	3	0	20	60	25	5	3	5	25	20	50

Table 3. Minimum and Maximum Quality Specifications of the Products

Specification Product	Minimum U87	Minimum U91	Minimum U93	Maximum U87	Maximum U91	Maximum U93
ARO (%vol aromatics)	0	0	0	60	60	60
BEN (%vol benzene)	0	0	0	5.9	5.9	5.9
MON	81.5	85.7	87.5	–	–	–
OLF (%vol olefin)	0	0	0	24.2	24.2	24.2
RON	91.4	94.5	97.5	–	–	–
RVP (psi)	0	0	0	15.6	15.6	15.6
SPG	0.73	0.73	0.73	0.81	0.81	0.81
SUL (%vol sulfur)	0	0	0	0.1	0.1	0.1

volume, BEN), olefin content (% by volume, OLF), motor octane number (MON), research octane number (RON), Reid vapor pressure (psi, RVP), specific gravity (SPG), and sulfur content (% by volume, SUL). Table 1 contains the blend components data, Table 2 shows the supply profile of blend components for cases 9 and 10 (uneven supply), and Table 3 presents the minimum and maximum specifications for the product properties. Table 4 shows the initial quality, initial inventory, and inventory limits of the products. Table 5 contains the demand profiles of each case and the cumulative curves for some case studies are shown in Figure 10. For the MILP volume allocation problem, the cost coefficients for component inventory slack variables are set to 1×10^{12}, and the cost coefficients for product inventory slack variables are shown in Table 6. Case Study 6 and 12 are explained for a better understanding of the algorithm and Table 11 and 16 presents the results of all case studies. All data is presented

Table 4. Products' Initial Quality, Initial Inventory, and Inventory Bounds

Initial Product Quality						
Product	U87	U91	U93	U87	U91	U93
Cases	1 – 10			11 – 12		
ARO (%vol aromatics)	5	10	20	20	40	63
BEN (%vol benzene)	4	4	5	4	5	7
MON	83.2	87	88.2	80	85	86
OLF (%vol olefin)	15	12	18	25	24	26
RON	91.4	95	98.2	90	93	96
RVP (psi)	15	8	12	16	15	17
SPG	0.75	0.76	0.75	0.73	0.82	0.83
SUL (%vol sulfur)	0.05	0.03	0.04	0.11	0.08	0.04
Product	**U87**		**U91**		**U93**	
Initial Product Inventory ($\times 10^3$ BBL)						
Cases 1 – 10	80		180		20	
Case 11	100		100		20	
Case 12	50		50		40	
Minimum Product Inventory ($\times 10^3$ BBL)						
All cases	10		10		10	
Maximum Product Inventory ($\times 10^3$ BBL)						
Cases 1 – 6, 8 – 12	300		300		100	
Case 7	800		500		200	

Table 5. Demand profiles ($\times 10^3$BBL)

		l-periods													
Case Study	Product	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	**U87**	30	30	50	50	50	75	100	120	200	220	150	75	50	30
	U91	60	120	80	70	50	0	0	30	0	0	50	0	0	60
	U93	0	0	30	45	0	40	40	0	35	30	30	45	0	20
2	**U87**	50	80	120	150	200	60	80	70	70	50	50	0	30	50
	U91	40	40	40	40	70	50	40	50	40	60	60	0	0	30
	U93	20	20	20	20	40	30	20	20	30	30	30	30	30	30
3	**U87**	50	80	120	150	200	120	50	30	20	20	50	140	150	50
	U91	40	40	40	40	40	40	40	40	40	40	40	40	40	40
	U93	20	20	20	20	20	25	20	20	20	20	20	20	20	20
4	**U87**	50	120	130	200	30	30	50	50	50	60	60	60	80	100
	U91	40	40	50	70	80	0	40	40	40	30	30	30	50	30
	U93	0	50	50	50	40	0	20	20	30	30	0	30	30	30
5	**U87**	50	50	50	150	200	120	50	30	20	50	50	160	200	50
	U91	40	40	40	40	40	40	40	40	40	40	40	40	40	40
	U93	20	20	20	20	20	25	20	20	20	20	20	20	20	20
6	**U87**	30	30	50	75	100	120	200	220	150	75	50	50	50	30
	U91	60	60	80	70	0	50	0	30	0	0	30	0	60	120
	U93	0	0	30	45	0	40	40	0	35	30	0	45	0	20
7	**U87**	0	0	0	0	0	720	0	0	0	0	0	0	0	510
	U91	0	0	0	0	0	280	0	0	0	0	0	0	0	280
	U93	0	0	0	0	0	145	0	0	0	0	0	0	0	140
8	**U87**	50	80	120	150	200	120	50	30	20	20	140	170	50	30
	U91	40	40	40	40	40	40	40	40	40	40	40	68	32	20
	U93	20	20	20	20	20	25	20	20	20	20	20	0	40	20
9	**U87**	50	50	50	150	200	120	50	30	20	50	50	160	200	50
	U91	40	40	40	40	40	40	40	40	40	40	40	40	40	40
	U93	20	20	20	20	20	25	20	20	20	20	20	20	20	20
10	**U87**	30	30	50	75	100	120	200	220	150	75	50	50	50	30
	U91	60	60	80	70	0	50	0	30	0	0	30	0	60	120
	U93	0	0	30	45	0	40	40	0	35	30	0	45	0	20
11	**U87**	30	30	50	50	50	75	100	120	200	220	150	75	50	30
	U91	60	120	80	70	50	0	0	30	0	0	50	0	0	60
	U93	0	0	30	45	0	40	40	0	35	30	30	45	0	20
12	**U87**	60	100	120	120	50	80	30	50	60	80	50	60	50	30
	U91	50	40	60	40	40	40	0	0	60	50	50	50	30	20
	U93	30	30	30	30	0	50	50	0	20	30	30	40	40	0

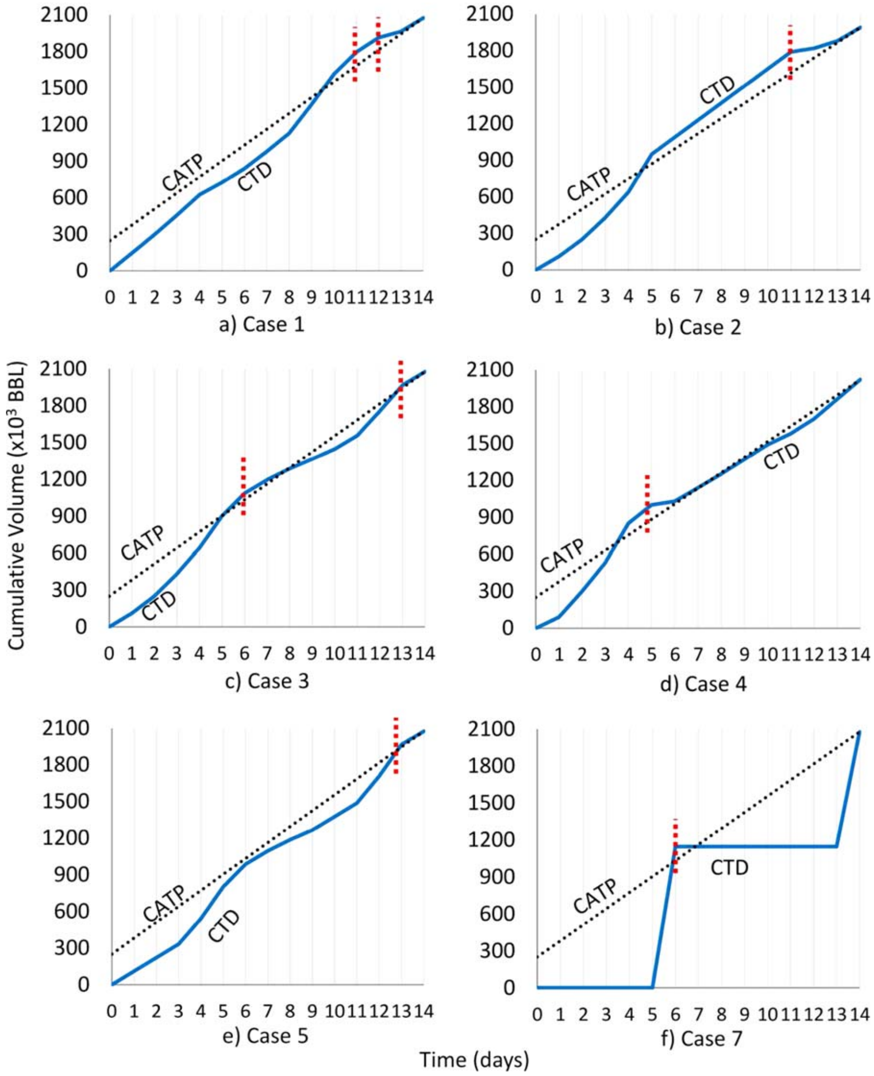

Figure 10. Cumulative demand profiles of some case studies.

The vertical dashed lines indicate the pinch points. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

in English system of units since it prevails in the refining industry.

In order to compute the number of different blend recipes in the MINLP solution, recipes of each blend component were grouped in composition intervals of 1 and 5%. Recipes in the same group were considered to be the same. A blend recipe is considered to be different if at least two components fall in different composition intervals. Since we are blending three grades of gasoline, the total number of different blend recipes has been computed by adding the number of different blend recipes for each grade and then dividing it by 3.

Detailed example: Case study 6 (one blender, RVP blends nonlinearly)

Following the inventory pinch algorithm, the first step is to solve the total aggregate model. Since the initial product inventories are on-spec, Eqs. 1–9 are used. The feasible solution has an objective function equal to \$43,611.95 $\times$ 10^3. Figure 11 shows the demand profile for this example. Following the second step of the algorithm, the only inventory pinch point identified appears in the 9th day as shown in Figure 12.

Then, in the third step of the algorithm, the planning horizon is divided according to the pinch points: t-period $K_1^{(1)}$ goes from l-period $n = 1$ to $n = 9$, and t-period $K_2^{(1)}$ goes

Table 6. Cost Coefficients Profile for the Product Inventory Slack Variables

	Cost coefficients		
l-period	Case 1 to 8, 11, and 12	Case 9	Case 10
1	9×10^8	9×10^8	9×10^9
2	8×10^8	8×10^8	8.5×10^9
3	7×10^8	7×10^8	8×10^9
4	6×10^8	6×10^8	7.5×10^8
5	5×10^8	5×10^8	7×10^8
6	1×10^8	4×10^8	6.5×10^8
7	8×10^7	9×10^5	6×10^8
8	7×10^7	8×10^3	5.5×10^8
9	5×10^6	7×10^3	5×10^5
10	1×10^6	6×10^3	4.5×10^5
11	5×10^4	5×10^3	4×10^5
12	1×10^3	4×10^3	3.5×10^5
13	50	3×10^3	3×10^2
14	1	1	1

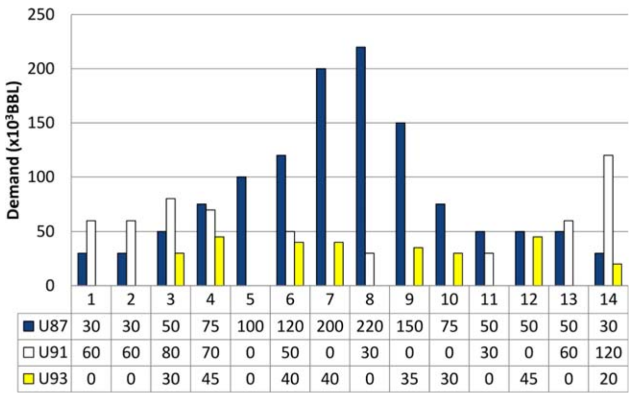

Figure 11. Demand Profile - Case 6.

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

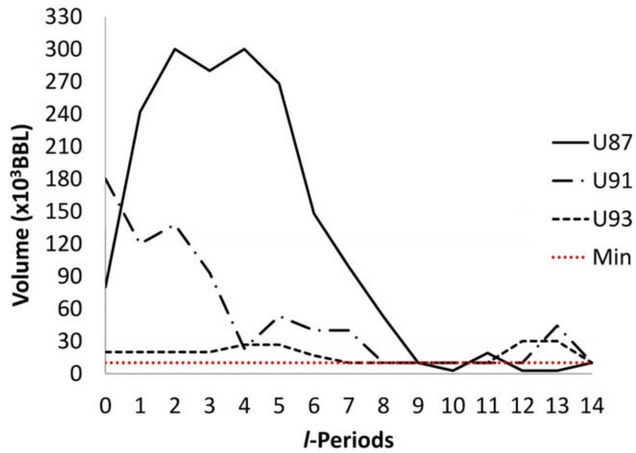

Figure 13. Product inventory profiles, 1st iteration - Case 6.

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

from l-period $n = 10$ to $n = 14$ (the superscript "(1)" indicates that these t-periods correspond to the first iteration of the algorithm). In the fourth step, the blend recipe model given by Eqs. 1–9 is solved for each t-period, using Eqs. 10–12 to link them at the boundaries through the inventories. Because this is the first iteration, the product closing inventories of the t-periods are chosen to be at the minimum allowed (i.e., 10×10^3BBL).

In the fifth step, the volume allocation problem (i.e., Eqs. 23–37) is solved. In this model, the blend recipes computed previously in Step 4 are fixed in the corresponding l-periods. The solution contains infeasibilities (see Figure 13), therefore it is necessary to divide a t-period. The first infeasibility (equal to 7.38×10^3BBL) appears at l-period $n = 10$ on the U87 inventory; thus, t-period $K_2^{(1)}$ is divided into two new periods: l-period $n = 10$ is contained by the new t-period $K_2^{(2)}$, and new t-period $K_3^{(2)}$ goes from l-period $n = 11$ to $n = 14$. t-period K_1 remains the same and it is not necessary to re-optimize the blend recipes for this t-period. No volume adjustments are required since there is enough capacity at $K_2^{(2)}$ (i.e., $n = 10$) to blend the volume required to overcome the infeasibility.

The blend recipes computed for the t-periods K_2 and K_3 of the 2nd iteration, as well as the blend recipe for K_1 from the first iteration, are shown in Table 7. The volume allocation

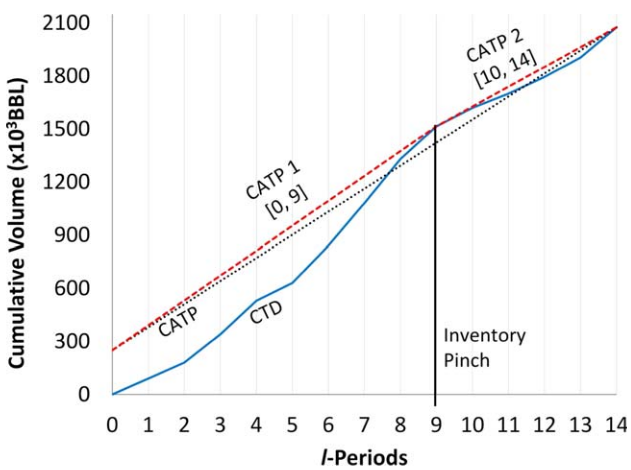

Figure 12. Inventory pinch points - Case 6.

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

problem is solved with the new blend recipes and this time no inventory infeasibilities are present; therefore, this is an optimal solution and the algorithm stops. Figures 14 and 15 show the inventory profiles of the products and blend components, respectively. The blend cost is equal to \$43,611.97 $\times 10^3$, which is 0.0006% higher than the solution from the corresponding MINLP model. The blend plan is shown in Table 8.

Detailed example: Case study 12 (two blenders, RVP blends linearly)

In this case, the initial product inventories are off-spec and properties are assumed to blend linearly; thus, the model comprised by Eqs. 13–22 is solved. The feasible solution of the total aggregate model has an objective function equal to \$41,211.25 $\times 10^3$. The aggregate recipe for the heel correction is shown in Table 9. l-period $n = 1$ is enough to bring the

Table 7. Aggregate Recipes—Case 6

t-periods l-periods	$[K_1^{(1)}]$ $[n=1$ to $9]$	$[K_2^{(2)}]$ $[n=10]$	$[K_3^{(2)}]$ $[n=11$ to $14]$
Product		**U87**	
ALK	0.1375	0.0038	0.0399
BUT	0.0257	0.0242	0.0239
HCL	0.0281	0.0104	0.0246
HCN	0.0287	0.0510	0.0479
LCN	0.2389	0.3046	0.3011
LNP	0.1952	0.2078	0.1938
RFT	0.3458	0.3982	0.3688
Product		**U91**	
ALK	0.2342	0.0000	0.1468
BUT	0.0364	0.0000	0.0350
HCL	0.0584	0.0000	0.0260
HCN	0.0784	0.0000	0.0364
LCN	0.1699	0.0000	0.1736
LNP	0.0607	0.0000	0.1185
RFT	0.3619	0.0000	0.4638
Product		**U93**	
ALK	0.1770	0.0503	0.1379
BUT	0.0438	0.0411	0.0431
HCL	0.0335	0.0740	0.0326
HCN	0.0549	0.0390	0.0574
LCN	0.1291	0.0719	0.1439
LNP	0.0236	0.0525	0.0279
RFT	0.5382	0.6712	0.5573

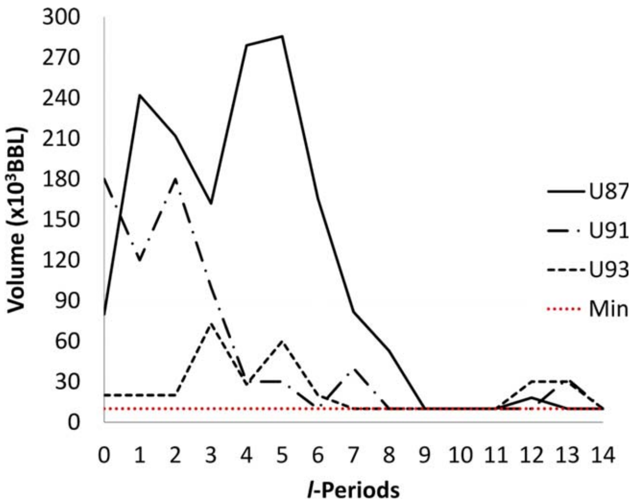

Figure 14. Product inventory profiles, 2nd iteration - Case 6.

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

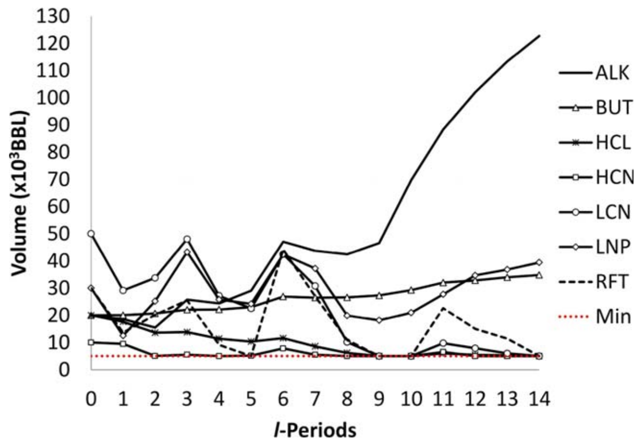

Figure 15. Component inventory profiles, 2nd iteration - Case 6.

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

inventories back to spec. Figure 16 shows the demand profile for this example. The inventory pinch points are identified to be in the 4th, 6th, and 13th day as shown in Figure 17.

The planning horizon is divided according to the pinch points: *t*-period $K_1^{(1)}$ goes from *l*-period $n = 2$ to $n = 4$, *t*-period $K_2^{(1)}$ goes from *l*-period $n = 5$ to $n = 6$, *t*-period $K_3^{(1)}$ goes from *l*-period $n = 7$ to $n = 13$, and *t*-period $K_4^{(1)}$ consists of *l*-period $n = 14$. In the fourth step, the blend recipe model given by Eqs. 1–8 is solved for each *t*-period, using Eqs. 10 to 12 to link them at the boundaries through the inventories. Because this is the first iteration, the product closing inventories are chosen to be at the minimum allowed (i.e., 10×10^3BBL), except for *t*-period $K_3^{(1)}$ where the closing inventory of U91 at $n = 13$ is set to 30×10^3BBL in order to avoid blending 30×10^3BBL (i.e., the minimum allowed) at *l*-period $n = 14$, where the demand is only 20×10^3BBL, thus, avoiding blending more than required. The aggregate blend recipes computed are shown in Table 9.

In the fifth step, the volume allocation problem (i.e., Eqs. 23 to 37) is solved. The solution does not contain infeasibilities; therefore, this is an optimal solution and the algorithm stops. Figure 18 and 19 show the inventory profiles of the products and blend components, respectively. The blend cost is equal to $\$41,470.74 \times 10^3$, which is 0.0007% higher than the solution from the corresponding MINLP model. The blend plan is shown in Table 10.

Discussion

The costs of the blend plans computed by the single period inventory pinch algorithm are very close to the optimal solutions computed from the corresponding fine-grid multiperiod MINLP model (see Table 11). The difference between the inventory pinch algorithm and MINLP results is less than 0.01% when the supply flow rates of blend components are constant along the planning horizon (case studies 1–8, 11, and 12). When the supply rates vary along the horizon (e.g., case studies 9 and 10) usually more inventory infeasibilities appear at the lower level, more iterations of the algorithm are required, and the costs computed via inventory pinch algorithm can be higher than those computed via multiperiod MINLP by a fractional percentage. The maximum difference observed in our case studies is 0.1284%. The higher the number of periods at the top level, the more likely is that solution will be suboptimal since the algorithm does not consider simultaneously all periods at the top level.

Decomposition of blend planning in two levels, as presented in this work, significantly simplifies the model solved at each level in comparison with the fine grid MINLP model. Table 13 shows that the number of equations, continuous variables, and nonzero elements in the NLP model (i.e., blend recipe computation) and MILP model (i.e., volume allocation problem) is smaller than those in the MINLP model.

The solution times of the inventory pinch algorithm (using IPOPT for the NLP model and CPLEX for the MILP model) are much smaller than those required by DICOPT to solve the corresponding MINLP model. If (quality*volume) constraints are nonlinear, then the computational effort is reduced by at least an order of magnitude. This is expected since the constraints at each level of the inventory pinch algorithm are simpler to solve and also the algorithm is a heuristic algorithm that does not prove that the optimality conditions are met.

The inventory pinch algorithm solution leads to a smaller number of different blend recipes per product compared to the solution of the corresponding multiperiod MINLP model

Table 8. Computed Blend Plan—Case 6

		l-periods													
Blender	Product	1	2	3	4	5	6	7	8	9	10	11	12	13	14
A	**U87**	192.0			192.0	106.6		116.0	191.2	107.2	75.0	50.0	58.1	41.9	30.0
	U91		120.0				30.0	30.0				30.0		82.2	97.8
	U93			83.0		32.0		30.0		35.0	30.0		65.0		

Table 9. Aggregate Recipes—Case 12

t-periods *l*-periods	$[K_0]$[a] $[n=1]$	$[K_1^{(1)}]$ $[n=2$ to $4]$	$[K_2^{(1)}]$ $[n=5$ o $6]$	$[K_3^{(1)}]$ $[n=7$ to $13]$	$[K_4^{(1)}]$ $[n=14]$
Product			**U87**		
ALK	0.0000	0.1061	0.1509	0.1011	0.0000
BUT	0.0445	0.0377	0.0445	0.0342	0.0550
HCL	0.2959	0.0224	0.0191	0.0220	0.1000
HCN	0.1676	0.0193	0.0311	0.0364	0.1667
LCN	0.0000	0.3323	0.2172	0.2437	0.0000
LNP	0.0000	0.1810	0.1975	0.2000	0.1792
RFT	0.4920	0.3013	0.3396	0.3626	0.4991
Product			**U91**		
ALK	0.0000	0.2147	0.2281	0.2086	0.0000
BUT	0.0809	0.0562	0.0585	0.0465	0.0000
HCL	0.1234	0.0427	0.0278	0.0288	0.0000
HCN	0.1657	0.0472	0.0456	0.0469	0.0000
LCN	0.0000	0.2142	0.1870	0.1967	0.0000
LNP	0.0000	0.0713	0.0848	0.0885	0.0000
RFT	0.6300	0.3535	0.3682	0.3841	0.0000
Product			**U93**		
ALK	0.2964	0.1577	0.1684	0.1644	0.0000
BUT	0.0680	0.0703	0.0706	0.0558	0.0000
HCL	0.0000	0.0319	0.0259	0.0244	0.0000
HCN	0.0000	0.0494	0.0462	0.0427	0.0000
LCN	0.0514	0.1349	0.1359	0.1488	0.0000
LNP	0.0000	0.0228	0.0250	0.0287	0.0000
RFT	0.5843	0.5329	0.5279	0.5350	0.0000

[a]Blend recipe to correct the heel

when considering a composition range of 1% to count the number of unique recipes. When using a 5% composition range sometimes the number given by the inventory pinch algorithm is greater (see Table 12).Significant reduction is seen in the cases with only one pinch point and constant supply rates of the blend components. In addition, repeated recipes from the MINLP solution may or may not be used in adjacent periods; thus, the number of recipe switching is equal or greater than the number of different recipes.

In the inventory pinch algorithm the solver selection for the recipe optimization level is very important. In general, blend recipes that disregard the use of several blending components are not likely to generate feasible blend plans at the lower level. Blend recipes computed by IPOPT and COUENNE lead to feasible solutions (in this work, only the results using IPOPT are included) while recipes computed by BARON, MINOS, CONOPT and GUROBI often lead to MILP solutions containing inventory infeasibilities (i.e., nonzero slack variables) at the lower level.

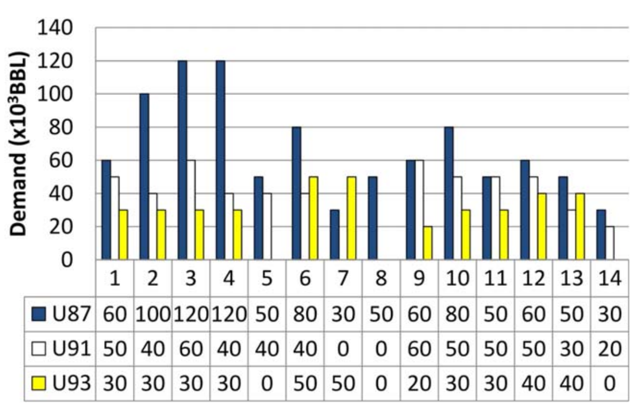

Figure 16. Demand Profile - Case 12.

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

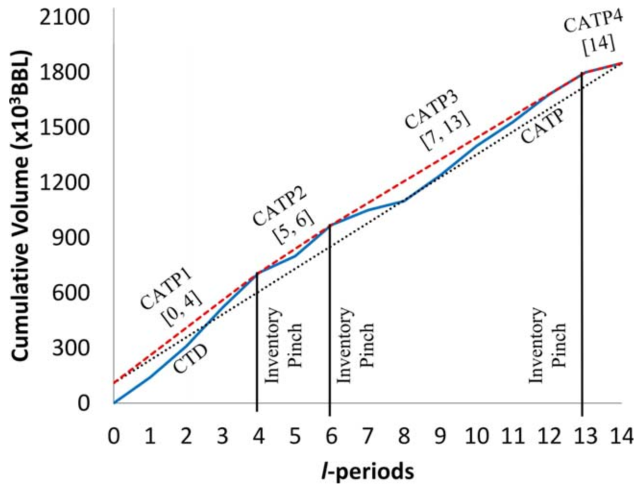

Figure 17. Inventory pinch points - Case 12.

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

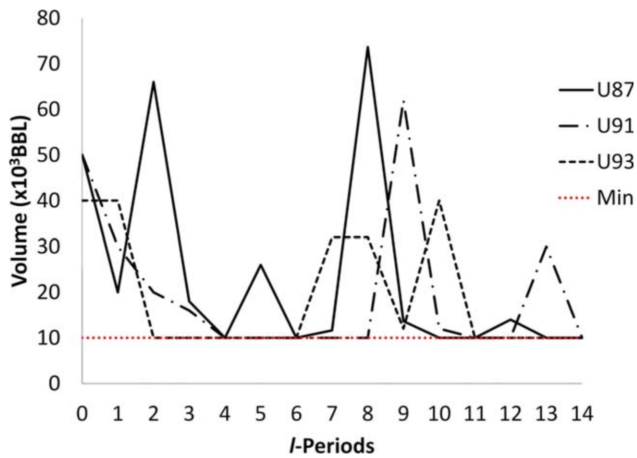

Figure 18. Product inventory profiles, 1st iteration - Case 12.

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

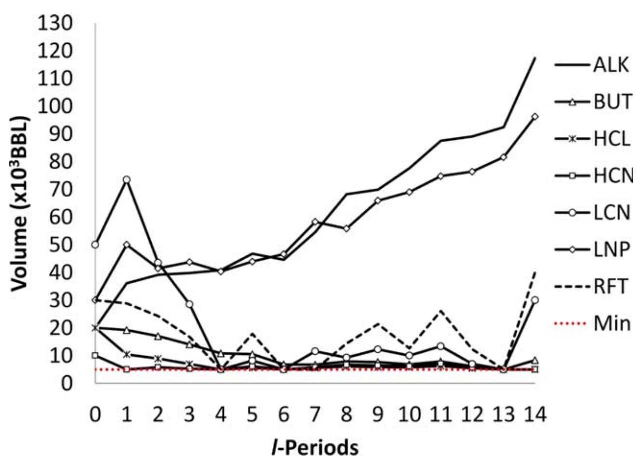

Figure 19. Component inventory profiles, 1st iteration - Case 12.

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

Table 10. Computed Blend Plan— Case 12

		l-periods													
Blender	Product	1	2	3	4	5	6	7	8	9	10	11	12	13	14
A	**U87**	30.0	74.0		112.0	66.0	64.0	31.7	112.0		76.3		64.0	46.0	30.0
	U91	30.0	30.0	56.0			40.0			112.0		48.0		50.0	
	U93	30.0		30.0									40.0		
B	**U87**		72.0	72.0								50.0			
	U91				34.0	40.0							50.0		
	U93				30.0		50.0	72.0			58.0			40.0	

Table 11. Objective function Values and Solution Times

				DICOPT Solution (MINLP model)		Inventory Pinch Algorithm Solution (IPOPT, CPLEX)		
Case Study	Number of blenders	Initial Product Inventory	RVP Blend Property	Objective Function ($\times10^3$\$)	Total CPU time (s)	Objective Function ($\times10^3$\$)	Total CPU time (s)	Objective function difference (%)
1	1	On-Spec	Linear	43420.20	12.26	43421.74	4.20	0.0035
2	1	On-Spec	Linear	41349.62	29.08	41350.01	1.23	0.0009
3	1	On-Spec	Linear	43161.03	37.69	43161.34	1.40	0.0007
3	2	On-Spec	Linear	43161.05	63.52	43161.33	1.50	0.0007
4	1	On-Spec	Linear	41874.17	20.53	41874.44	1.36	0.0006
4	2	On-Spec	Linear	41874.45	115.15	41874.45	2.75	0.0000
5	1	On-Spec	Nonlinear	43657.71	13.84	43658.00	0.98	0.0007
6	1	On-Spec	Nonlinear	43611.69	12.81	43611.97	2.19	0.0006
7	1	On-Spec	Nonlinear	43611.69	33.20	43611.98	1.08	0.0007
7	2	On-Spec	Nonlinear	43611.69	34.64	43611.97	1.19	0.0006
8	1	On-Spec	Nonlinear	43934.67	18.51	43934.95	1.71	0.0006
8	2	On-Spec	Nonlinear	43934.95	82.32	43934.95	2.04	0.0000
9	1	On-Spec	Nonlinear	43627.27	372.59	43627.45	8.77	0.0004
9	1	On-Spec	Linear	43142.09	424.95	43142.25	7.80	0.0004
10	1	On-Spec	Nonlinear	43611.69	641.76	43667.68	5.02	0.1284
10	1	On-Spec	Linear	43101.12	235.61	43126.28	4.33	0.0584
11	1	Off-Spec	Linear	43424.87	15.17	43425.15	2.87	0.0006
12	1	Off-Spec	Linear	41470.45	7.73	41470.74	3.05	0.0007
12	2	Off-Spec	Linear	41470.45	56.87	41470.74	3.57	0.0007

Table 12. Number of Blend Recipes

				DICOPT Solution (MINLP model)		Inventory Pinch Algorithm Solution (IPOPT, CPLEX)
Case Study	Initial Product Inventory	RVP Blend Property	Number of Pinch Points	Number of different recipes (1%)[a]	Number of different recipes (5%)[b]	Number of different recipes (0%)[c]
1	On-Spec	Linear	2	8	6	4
2	On-Spec	Linear	1	7	4	2
3	On-Spec	Linear	2	6	3	3
4	On-Spec	Linear	1	7	3	2
5	On-Spec	Nonlinear	1	7	3	2
6	On-Spec	Nonlinear	1	9	5	3
7	On-Spec	Nonlinear	1	7	4	2
8	On-Spec	Nonlinear	3	9	5	4
9	On-Spec	Nonlinear	1	7	4	6
9	On-Spec	Linear	1	7	5	6
10	On-Spec	Nonlinear	1	6	3	4
10	On-Spec	Linear	1	8	4	4
11	Off-Spec	Linear	2	7	3	4
12	Off-Spec	Linear	3	7	5	5

[a]Blend recipes were counted using composition intervals of 1%
[b]Blend recipes were counted using composition intervals of 5%
[c]Repeated blend recipes are exactly the same

Table 13. Model Size Comparison

	# Equations	# Continuous Variables	# Discrete Variables	# Non-zeros
MINLP model (1 blender)	2,017	1,569	42	6,221
MINLP model (2 blenders)	3,473	2,577	84	11,051
NLP model (SPIP algorithm)	127	97	0	371
MILP model (1 blender, SPIP algorithm)	1,261	939	42	3,197
MILP model (2 blenders, SPIP algorithm)	1,961	1,317	84	5,003

Conclusions

We have introduced a novel heuristic algorithm based on the concept of inventory pinch points and the two-level decomposition of the gasoline blend planning problem. An inventory pinch point is the point in time where the cumulative average total production curve is tangent to the cumulative total demand curve, and if extrapolated from this point onward, it does not cross the CTD curve. At the top level of the algorithm, blend recipes are optimized based on total demand between pinch points using a NLP model and, at the lower level, blend volumes are computed from a fine-grid MILP model. The algorithm addresses two aspects of gasoline blend planning (1) minimization of the number of different blend recipes, and (2) use of computationally intensive nonlinear models for blend planning via a single-period model. Experiments with a number of examples show that the solutions computed by the single period inventory pinch algorithm are most of the time less than 0.01% away from the optimum solutions computed by the corresponding discrete-time multiperiod MINLP formulation when the supply of components is constant. In all cases, solution times were smaller than those required by a MINLP solver; as a rule, in cases with nonlinear blend constraints, execution times are an order of magnitude lower. The number of blend recipe switches along the blend planning horizon is much smaller than computed by the MINLP model; this feature makes it simpler for schedulers that use interactive simulation software to create blend schedules. Solver selection for the recipe optimization level is important and has a great effect in the algorithm performance; IPOPT and COUENNE provided the best results among the solvers tested.

Acknowledgments

This work has been supported by Ontario Research Foundation. V. Mahalec would like to dedicate this work to Prof. E. Bauman in recognition of his leadership which transformed chemical engineering in Croatia during 1970 to 2000.

Notation

g = gasoline grades, U87, U91, U93
i = blend components, ALK, BUT, HCL, HCN, LCN, LNP, RFT
n = l-periods in the production planning level, 1,…, N
bl = blender, A, B
s = quality properties, ARO, BEN, OLF, RON, MON, RVP, SPG, SUL

Subscripts

B = blender
C = blend component
K = t-period
P = product

Parameters

$\mathrm{Cost}(i)$ = cost of blend component i
$\mathrm{D_P}(g)$ = demand of product g in the total aggregate formulation
$\mathrm{D_P}(g,n)$ = demand of product g in l-period n
$\mathrm{D_{P,K}}(g)$ = demand of product g in t-period K
$\mathrm{F_B}^{\max}$ = maximum blending capacity
H = length of the planning horizon
$\mathrm{np}(bl)$ = number of products that can be produced in a l-period in blender bl
$\mathrm{Penalty_C}(i)$ = penalty for the inventory slack variables of blend components i
$\mathrm{Penalty_{P,K}}(g)$ = penalty for the inventory slack variables of product g
$\mathrm{Penalty_P}(g,n)$ = penalty for the inventory slack variables of product g in l-period n
$\mathrm{Q_C}(i,s)$ = quality s of blend component i
$\mathrm{Q_P}^{\max}(g,s)$ = maximum requirement of quality s in grade g
$\mathrm{Q_P}^{\min}(g,s)$ = minimum requirement of quality s in grade g
$t(n)$ = duration of l-period n
t_K = duration of t-period K
$\mathrm{V_B}^{\mathrm{lost}}(bl)$ = volume lost due to switching from one grade to another in blender bl
$\mathrm{V_B}^{\max}(g)$ = maximum volume allowed to blend of grade g during any blender run
$\mathrm{V_B}^{\min}(g)$ = minimum volume allowed to blend of grade g during any blender run
$\mathrm{V_{C,K}}^{\mathrm{in}}(i)$ = volume supply of blend component i in t-period K
$\mathrm{V_C}^{\mathrm{in}}(i,n)$ = volume supply of blend component i in l-period n
$\mathrm{V_C}^{\max}(i)$ = maximum volume of tank with blend component i
$\mathrm{V_C}^{\min}(i)$ = minimum volume of tank with blend component i
$\mathrm{V_P}^{\max}(g)$ = maximum volume of tank with product g
$\mathrm{V_P}^{\min}(g)$ = minimum volume of tank with product g

Variables

$\mathrm{A}(g,n)$ = binary variable to determine if product g is blended in l-period n
$\mathrm{Q_P}(g,s,n)$ = quality s of product g at l-period n
$\mathrm{Q_P}^{\mathrm{close}}(g,s,n)$ = quality s of product g at the end of l-period n
$\mathrm{Q_P}^{\mathrm{open}}(g,s,n)$ = quality s of product g at the beginning of l-period n
$\mathrm{Q_{P,0}}^{\mathrm{close}}(g,s)$ = quality s of product g at the end of t-period K_0
$\mathrm{Q_{P,0}}^{\mathrm{open}}(g,s)$ = quality s of product g at the beginning of t-period K_0
$\mathrm{RVP}(i)$ = reid vapor pressure of blend component i
$\mathrm{RVP_K}(g)$ = reid vapor pressure of product g at t-period K
$\mathrm{S_C}^{+}(i,n)$ = positive inventory slack variable of blend component i at l-period n
$\mathrm{S_C}^{-}(i,n)$ = negative inventory slack variable of blend component i at l-period n
$\mathrm{S_{C,K}}^{+}(i)$ = positive inventory slack variable of blend component i at t-period K
$\mathrm{S_{C,K}}^{-}(i)$ = negative inventory slack variable of blend component i at t-period K
$\mathrm{S_P}^{+}(g,n)$ = positive inventory slack variable of product g at l-period n
$\mathrm{S_P}^{-}(g,n)$ = negative inventory slack variable of product g at l-period n
$\mathrm{S_{P,0}}^{+}(g)$ = positive inventory slack variable of product g at t-period K_0
$\mathrm{S_{P,0}}^{-}(g)$ = negative inventory slack variable of product g at t-period K_0
$\mathrm{S_{P,K}}^{+}(g)$ = positive inventory slack variable of product g at t-period K
$\mathrm{S_{P,K}}^{-}(g)$ = negative inventory slack variable of product g at t-period K
$\mathrm{V}(0)$ = sum of the initial inventories of all products
$\mathrm{V_B}(g)$ = volume of product g blended in total aggregate formulation
$\mathrm{V_B}(g,n)$ = volume of product g blended in l-period n
$\mathrm{V_{B,0}}(g)$ = volume of product g blended in t-period K_0
$\mathrm{V_{B,K}}(g)$ = volume of product g blended in t-period K
$\mathrm{V_C}(i,g)$ = volume of blend component i into product g in total aggregate formulation
$\mathrm{V_C}(i,g,n)$ = volume of blend component i into product g in l-period n
$\mathrm{V_C}^{\mathrm{close}}(i)$ = closing volume of tank with blend component i in total aggregate formulation
$\mathrm{V_C}^{\mathrm{close}}(i,n)$ = closing volume of tank with blend component i in l-period n
$\mathrm{V_C}^{\mathrm{open}}(i)$ = opening volume of tank with blend component i in total aggregate formulation
$\mathrm{V_C}^{\mathrm{open}}(i,n)$ = opening volume of tank with blend component i in l-period n
$\mathrm{V_{C,0}}(i,g)$ = volume of blend component i into product g in t-period K_0
$\mathrm{V_{C,K}}(i,g)$ = volume of blend component i into product g in t-period K
$\mathrm{V_{C,K}}^{\mathrm{close}}(i)$ = closing volume of tank with blend component i in t-period K
$\mathrm{V_{C,K}}^{\mathrm{open}}(i)$ = opening volume of tank with blend component i in t-period K
$\mathrm{V_P}^{\mathrm{close}}(g)$ = closing volume of tank with product g in total aggregate formulation

$V_P^{close}(g,n)$ = closing volume of tank with product g in l-period n
$V_P^{open}(g)$ = opening volume of tank with product g in total aggregate formulation
$V_P^{open}(g,n)$ = opening volume of tank with product g in l-period n
$V_{P,0}^{close}(g)$ = closing volume of tank with product g in t-period K_0
$V_{P,0}^{open}(g)$ = opening volume of tank with product g in t-period K_0
$V_{P,K}^{close}(g)$ = closing volume of tank with product g in t-period K
$V_{P,K}^{open}(g)$ = opening volume of tank with product g in t-period K
$x(i,g,n)$ = volume fraction of blend component i in product g at l-period n
$x_K(i,g)$ = volume fraction of blend component i in product g at t-period K
XC = extra cost for volume of components used to correct the heel off-spec

Literature Cited

1. Kelly JD. Formulating production planning models. *Chem Eng Prog*. 2004;100:43–50.
2. Mendez CA, Grossman IE, 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.
3. Neiro SMS, Pinto J. Multiperiod optimization for production planning of petroleum refineries. *Chem Eng Comm*. 2005;192:62–88.
4. Misener R, Floudas C. Advances for the pooling problem: modeling, global optimization, and computational studies - survey. *Appl Comput Math*. 2009;8:3–22.
5. Main RA. Large recursion models: practical aspects of recursion techniques. In: Ciriani TA, Leachman RC, eds. Optimization in Industry. New York, NY: John Wiley & Sons, Ltd.; 1993.
6. US Government Printing Office. Code of Federal Regulations: Complex emissions model. Year: 2007. Title: 40. Section: CFR80.45. http://www.gpoaccess.gov/cfr/retrieve.html. Accessed August 13, 2012.
7. Misener R, Gounaris CE, Floudas C. Mathematical modeling and global optimization of large-scale extended pooling problems with epa complex emissions constraints. *Comput Chem Eng*. 2010;34:1432–1456.
8. Jia Z, Ierapetritou M. Mixed-integer linear programming model for gasoline blending and distribution scheduling. *Ind Eng Chem Res*. 2003;42:825–835.
9. Thakral A, Mahalec V. Composite planning and scheduling algorithm addressing intra-period infeasibilities of gasoline blend planning models. *Can J Chem Eng*. 2012;9999:1–12.
10. Glismann K, Gruhn G. Short-term scheduling and recipe optimization of blending processes. *Comput Chem Eng*. 2001;25:627–634.
11. Joly M, Pinto JM. Mixed-Integer Programming Techniques for the Scheduling of Fuel Oil and Asphalt Production. *Trans IChemE A*. 2003;81:427 – 447.
12. Li J, Karimi IA, Srinivasan R. Recipe Determination and Scheduling of Gasoline Blending Operations. *AIChE J*. 2010;56:441–465.
13. Kelly JD, Mann JL. Crude-oil blend scheduling optimization: An application with multi-million dollar benefits: Part I. *Hydrocarbon Process*. 2003:47-53.
14. Kelly JD, Mann JL. Crude-oil blend scheduling optimization: An application with multi-million dollar benefits: Part II. *Hydrocarbon Process*. 2003:72–79.
15. Li J, Karimi IA. Scheduling gasoline blending operations from recipe determination to shipping using unit slots. *Ind Eng Chem Res*. 2011;50:9156–9174.
16. Sing A, Forbes JF, Vermeer PJ, Woo SS. Model-based real-time optimization of automotive gasoline blending operations. *J Process Control*. 2000; 10:43–58.

Manuscript received Aug. 29, 2012, and revision received Feb. 21, 2013.