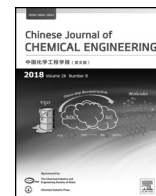




Contents lists available at ScienceDirect

Chinese Journal of Chemical Engineering

journal homepage: www.elsevier.com/locate/CJChE

Article

Logarithm-transform piecewise linearization method for the optimization of gasoline blending processes☆

Yu Li, Tong Qiu *

Department of Chemical Engineering, Tsinghua University, Beijing 100084, China

ARTICLE INFO

Article history:

Received 15 December 2017

Accepted 19 December 2017

Available online 3 February 2018

Keywords:

Piecewise linearization

Blending

Non-convex

Global optimization

ABSTRACT

Gasoline blending is a key process in a petroleum refinery, as it can yield 60%–70% of a typical refinery's total revenue. This process not only exhibits non-convex nonlinear blending behavior due to the complicated blend mechanism of various component feedstocks with different quality properties, but also involves global optimum searching among numerous blending recipes. Since blend products are required to meet a series of quality requirements and highly-sensitive to the proportion changes of blending feedstocks, global optimization methods for NLP problems are often difficult to be applied because of heavy computational burdens. Thus, piecewise linearization methods are naturally proposed to provide an approximate global optimum solution by adding binary variables into the models and converting the original NLP problems into MILP ones. In this paper, Logarithm-transform piecewise linearization (LTPL) method, an improved piecewise linearization, is proposed. In this method a logarithm transform is applied to convert multi-variable multi-degree constraints into a series of single-variable constraints. As a result, the number of 0–1 variables is greatly reduced. In the final part of this paper, an industrial case study is conducted to demonstrate the effectiveness of LTPL method. In principle, this method would be useful for blending problems with complicated empirical or theoretical models.

© 2018 The Chemical Industry and Engineering Society of China, and Chemical Industry Press. All rights reserved.

1. Introduction

The main objective of an oil refining process is to convert its input materials, e.g. a wide variety of crude oils, into consumer-demanded products such as jet fuels and diesel. Among these products, gasoline is considered to be one of the most important and valuable output products as it can yield 60%–70% of a typical refinery's total revenue [1–3]. The economic efficiency and profitability are associated with higher-quality products (higher prices) and less-use or less-waste of input materials, which could be achieved by critical and accurate gasoline blending. As it shows in Fig. 1 [4], which is a flowsheet of a simplified refinery, in a gasoline-blending process, components oil (butanes, alkylate, reformate, etc.) from supply-tanks or produced by other units are mixed in blending tanks or on-line blenders to make the gasoline products meet a variety of quality-requirements.

Some of the most important gasoline quality specifications include research octane number (RON), motor octane number (MON), Reid vapor pressure (RVP), density, sulfur, aromatics (A), and olefins (O) content. The octane number of gasoline is defined as the percentage of iso-octane (assigned an octane number of 100) in a blend with n-heptane (assigned an octane number of 0). It represents the performance of resistance to

knocking conditions. RON and MON are based on two different test methods [4]: (1) the RON represents antiknock property under low speed and frequent accelerations and (2) the MON represents engine performance under higher speed conditions. The Reid vapor pressure (RVP) matters as well since it affects the volatility of gasoline. However, RON, MON and RVP do not blend linearly; to the contrary, the blender qualities appear to be in strong nonlinear relationships with the qualities of feedstock components [5]. Thus, applying nonlinear blending rules for such quality properties is feasible to increase prediction accuracy and reduce quality giveaway, but most blending equations are non-convex [6].

The blending technology used by refiners to optimize the gasoline recipe typically refers to three levels [7]: off-line optimization, on-line optimization, and regulatory control. Given the component oil properties, the off-line optimizer generates the initial blend recipe. Then this recipe is subsequently downloaded to the on-line optimizer that determines the set points for the controllers. In practice, both off-line and on-line optimizers are typically built with either linear programming or successive LPs to solve nonlinear optimization models. Considering the specific characteristics of blending processes, some companies have developed commercial software packages [8] aiming to optimize and schedule on-line blending problems, such as Honeywell's BPC, HIS' H/BOSS, and ASPENTECH's ORION. However, because of high charge-fees and some restrictions of those software packages, a large amount of domestic refineries still adopt the method of off-line empirical tank blending, which mainly depends on plants' previous blending histories and operators' individual experiences, and often results in

☆ Supported by the National Basic Research Program of China (2012CB720500) and the National Natural Science Foundation of China (U1462206).

* Corresponding author.

E-mail address: qiutong@mails.tsinghua.edu.cn (T. Qiu).

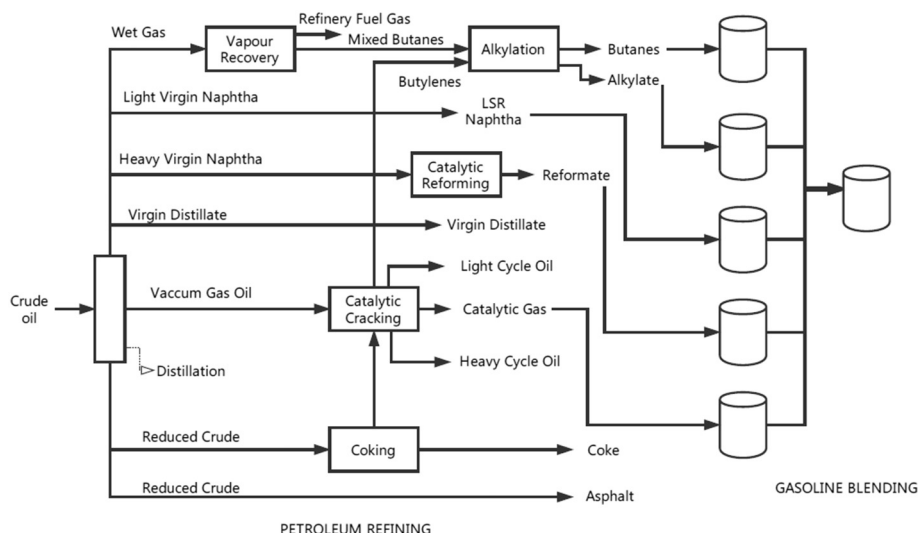


Fig. 1. A simplified petroleum refinery flowsheet.

highly-cost recipes due to cautious redundant operations for producing higher-quality gasoline products.

2. Literature Review

There are a number of researches studying on the global optimization of nonlinear non-convex problems. N. Boland [9] proposed a method which replaced the original formulation with an equivalent solution that keeping the feasible space with strong relaxation. His method eliminates the difficulty of calculating tight bounds for objective functions. Sherali and Alameddine [10] gave a reformed linearization technique, in which redundant constraints are added into the model to strengthen the relaxation while keeping the original space the same. However, it is still hard to decide where to add the constraints to realize a stronger relaxation. Spatial branch-and-bound [11,12] is a widely used method and an essential element of many commercial solvers such as BARON [13], ANTIGONE [14], and SCIP [15], because of its advantage in convergence speed for the global optimization problem. It works by iteratively reducing the domain of variables one by one. As a result, when the initial relaxation is weak, the calculation process could be very slow for a big-scale non-convex problem. This efficiency problem occurs with optimality-based bound tightening (OBBT) method [16] as well when dealing with a large number of variables. Hence, it is still in burden with computation by these methods for global optimization. A natural approach for solving this kind of problems is to approximate each general function by a piecewise linear expression and then reformulate the original problem as a discrete linear optimization problem. As for linear programming problems, there are many effective and widely-applied methods to find out the global optimization solutions in a short time.

Piecewise linearization methods are frequently used in various applications to approximate nonlinear problems with non-convex functions in the objective or constraints by adding extra binary variables, continuous variables, and constraints. Accordingly, various piecewise linearization methods have been proposed. In the early 1950s, 0–1 variables were firstly introduced to piecewise linearize a concave function [17,18]. Sherali and Tuncbilek [19] proposed a method of reformulation–linearization technique (RLT), which introduces polynomial constraints and then linearizes the resulting problem by giving new variables. The method of RLT is able to find out a lower bound in the context of a branch-and bound scheme. Then the author developed a hierarchy of relaxations for solving mixed 0–1 integer programming problems as an extension of RLT. However, the idea is

good but the implement procedure is difficult because there are numerous constraints that have to be transformed into linearized forms, which makes the computation even harder. Geoffrion [20] considered to minimize a concave function over a general round set and derive conditions, under which a piecewise linear approximation of the objective function achieves the smallest possible absolute error for a given number of pieces. After that, Morris [21] went further in approximating the objective function of a convex problem and deriving bounds on the number of partitions needed to guarantee a given absolute error. This made the piecewise linearization process controllable and the complexity predictable. Besides, how to find a better way to represent the piecewise linearization problem and reduce the scale of computing is an important topic in this field. Earlier piecewise relaxation techniques scale linearly, while recent formulations scale logarithmically. Compared to many previous methods, C. Huang [22] proposed a method for piecewise linear functions with fewer binary variables, which is achieved by introducing $\lceil \log_2 m \rceil$ instead of m extra binary variables with $m + 1$ break points.

Most blending rules are non-convex nonlinear formulas, and blend properties such as RON and MON, are highly-sensitive to the proportion changes of blending feedstocks. Considering these two aspects, it is rather difficult to find out the global optimum solutions. Applying piecewise linearization method is a good idea to provide an approximate optimum solution quickly. However conventional piecewise linearization methods mainly focus on scaling-down the model by relaxing constraints or reducing the number of 0–1 variables. In the next part we will propose a piecewise linearization method aiming to scale down models with multi-variable multi-degree expressions.

3. Piecewise Linearization

3.1. Conventional piecewise linearization

Given a general nonlinear function $f(x)$ of a single variable x , here $f(x)$ is a continuous function, and x is located within $[a_0, a_m]$.

In a conventional piecewise linear approximation process, firstly, denote $a_k (k \in K = \{0, 1, \dots, m-1\})$ as the break points of x in the whole interval, $a_0 < a_1 < \dots < a_m$, and Fig. 2 [23] shows the piecewise linearization of $f(x)$. According to the method proposed by Padberg [24] which is based on big-M method and introduces $[m]$ binary variables, consider

$$a_k - M(1 - \lambda_k) \leq x \leq a_{k+1} - M(1 - \lambda_k) \quad \forall k \in K \quad (1)$$

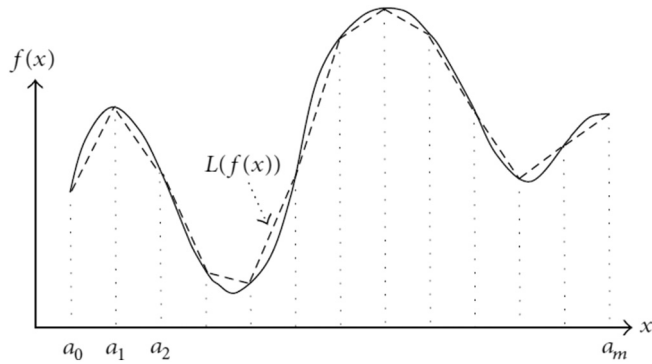


Fig. 2. Piecewise linearization of $f(x)$.

where $\sum_{k=0}^{m-1} \lambda_k = 1$, $\lambda_k \in \{0, 1\}$, M is a large positive constant value.

This indicates that the interval $[a_k, a_{k+1}]$ will be activated if $\lambda_k = 1$

$$\begin{cases} a_k \leq x \leq a_{k+1} \\ a_t - M \leq x \leq a_{t+1} + M \quad \forall t, t \neq k \end{cases} \quad (2)$$

$$(3)$$

It is obvious that we need to introduce $[m]$ binary variables to piecewise $f(x)$ with $[m+1]$ break points. Then $f(x)$ could be expressed as:

$$f(a_k) + s_k(x - a_k) - M'(1 - \lambda_k) \leq f(x) \leq f(a_k) + s_k(x - a_k) + M'(1 - \lambda_k) \quad \forall k \in K \quad (4)$$

where M' is a large positive constant value and s_k is the slope of the interval approximation function:

$$s_k = (f(a_{k+1}) - f(a_k)) / (a_{k+1} - a_k) \quad \forall k \in K \quad (5)$$

With reference to the above construction, the reformulation is finished for the nonlinear programming problem with single independent variable. However, when considering cases with multi independent variable in a non-convex problem, it can be difficult to solve. For example, a function with $[n]$ independent variables, $x_1, x_2, \dots, x_n$, if we separate each variable into m intervals, then the linearization form of $f(x)$ will be consisted of $[m^n]$ partitions and this will result in the problem of combinatorial explosion nullifying the efficiency of conversion from NLPs to MILPs.

The following part proposed an improved piecewise linearization method which presents an efficient scheme for linearizing $f(x)$ of more than one independent variable.

3.2. Logarithm-transform piecewise linearization method (LTPL method)

Considering that most blending laws describe the interaction between different components, there shall be interactive items in blending models. Based on the above conventional method, LTPL method applies a logarithm transform for the multi-variable multi-degree part of problems, which could avoid introducing a large amount of 0–1 variables. Considering the following problem:

Problem P

$$\min \sum_{i=1}^n f_i(x_i) \quad (6)$$

$$\text{s.t.} \quad \sum_{i=1}^n \sum_{j \geq i}^n \frac{b_{i,j} x_i x_j}{x_i + x_j} \leq p_{i,j} \quad (7)$$

$$lb_i \leq x_i \leq ub_i \quad \forall i \in I = \{1, 2, \dots, n\} \quad (8)$$

Given the sign of $\{lb_i\}$, two situations could be classified and discussed.

Situation 1: $\{lb_i\}, \{ub_i\}$ are positive constant values.

Problem P contains several multi-variable multi-degree constraints. The linear approximation of problem P is discussed. To simplify the discussion, only one continuous function of one single variable, $f_i(x_i)$, is considered here, but it should also be noted that LTPL method is also suitable for objective function forms in more complicated forms. There are two steps to solve this problem.

Step 1:

Replacing

$$y_{i,j} = \frac{x_i x_j}{x_i + x_j} \quad (9)$$

$$z_{i,j} = x_i + x_j \quad (10)$$

$$\forall i \in I, j \geq i$$

Since every x_i is positive, take the logarithm transformation, that is, replace $y_{i,j}$ by $\ln(y_{i,j})$, and replace $\frac{x_i x_j}{z_{i,j}}$ by $\ln(x_i) + \ln(x_j) - \ln(z_{i,j})$. Thus the programming problem changes to

$$\min \sum_{i=1}^n f_i(x_i) \quad (11)$$

$$\text{s.t.} \quad \sum_{i=1}^n \sum_{j \geq i}^n b_{i,j} y_{i,j} \leq p_{i,j} \quad (12)$$

$$z_{i,j} = x_i + x_j \quad \forall i \in I, j \geq i \quad (13)$$

$$\ln(y_{i,j}) = \ln(x_i) + \ln(x_j) - \ln(z_{i,j}) \quad \forall i \in I, j \geq i \quad (14)$$

$$lb_i \leq x_i \leq ub_i \quad \forall i \in I \quad (15)$$

$$\frac{lb_i \cdot lb_j}{ub_i + ub_j} \leq y_{i,j} \leq \frac{ub_i \cdot ub_j}{lb_i + lb_j} \quad \forall i \in I, j \geq i \quad (16)$$

$$lb_i + lb_j \leq z_{i,j} \leq ub_i + ub_j \quad \forall i \in I, j \geq i \quad (17)$$

Step 2:

Defining new variables, $Y_{i,j}, X_i, Z_{i,j}$

$$Y_{i,j} = \ln(y_{i,j}) \quad \forall i \in I, j \geq i \quad (18)$$

$$X_i = \ln(x_i) \quad \forall i \in I \quad (19)$$

$$Z_{i,j} = \ln(z_{i,j}) \quad \forall i \in I, j \geq i \quad (20)$$

Since $\ln(y_{i,j}), \ln(x_i), \ln(x_j), \ln(z_{i,j})$ are non-convex terms, it is necessary to convert them into convex expressions. Thus we use piecewise linear functions to approximate these non-convex terms, that is, separating $y_{i,j}, x_i$, and $z_{i,j}$ into a series of partitions and piecewise functions respectively

$$\left(\frac{lb_i \cdot lb_j}{ub_i + ub_j} = a_{i,j,0} \leq a_{i,j,1} \leq \dots \leq a_{i,j,k} \leq a_{i,j,k+1} \leq \dots \leq a_{i,j,m-1} = \frac{ub_i \cdot ub_j}{lb_i + lb_j} \right) \quad (21)$$

$$a_{i,j,k} - M(1 - \lambda_{i,j,k}) \leq y_{i,j} \leq a_{i,j,k+1} - M(1 - \lambda_{i,j,k}) \quad (22)$$

$$\ln(a_{i,j,k}) + \frac{\ln(a_{i,j,k+1}) - \ln(a_{i,j,k})}{a_{i,j,k+1} - a_{i,j,k}} (y_{i,j} - a_{i,j,k}) - M'(1 - \lambda_{i,j,k}) \leq Y_{i,j} \quad (23)$$

$$Y_{i,j} \leq \ln(a_{i,j,k}) + \frac{\ln(a_{i,j,k+1}) - \ln(a_{i,j,k})}{a_{i,j,k+1} - a_{i,j,k}} (y_{i,j} - a_{i,j,k}) + M'(1 - \lambda_{i,j,k}) \quad (24)$$

$$\left(\begin{aligned} lb_i = b_{i,0} \leq b_{i,1} \leq \dots \leq b_{i,k} \leq b_{i,k+1} \dots \leq b_{i,m-1} = ub_i & \quad (25) \\ b_{i,k} - M(1 - \mu_{i,k}) \leq x_i \leq b_{i,k+1} - M(1 - \mu_{i,k}) & \quad (26) \\ \ln(b_{i,k}) + \frac{\ln(b_{i,k+1}) - \ln(b_{i,k})}{b_{i,k+1} - b_{i,k}} (y_{i,j} - b_{i,k}) - M'(1 - \mu_{i,k}) \leq x_{i,j} & \quad (27) \\ x_{i,j} \leq \ln(b_{i,k}) + \frac{\ln(b_{i,k+1}) - \ln(b_{i,k})}{b_{i,k+1} - b_{i,k}} (y_{i,j} - b_{i,k}) + M'(1 - \mu_{i,k}) & \quad (28) \end{aligned} \right)$$

$$\left(\begin{aligned} lb_i + lb_j = c_{i,j,0} \leq c_{i,j,1} \leq \dots \leq c_{i,j,k} \leq c_{i,j,k+1} \dots \leq c_{i,j,m-1} = ub_i + ub_j & \quad (29) \\ c_{i,j,k} - M(1 - \eta_{i,j,k}) \leq z_{i,j} \leq c_{i,j,k+1} - M(1 - \eta_{i,j,k}) & \quad (30) \\ \ln(c_{i,j,k}) + \frac{\ln(c_{i,j,k+1}) - \ln(c_{i,j,k})}{c_{i,j,k+1} - c_{i,j,k}} (z_{i,j} - c_{i,j,k}) - M'(1 - \eta_{i,j,k}) \leq z_{i,j} & \quad (31) \\ z_{i,j} \leq \ln(c_{i,j,k}) + \frac{\ln(c_{i,j,k+1}) - \ln(c_{i,j,k})}{c_{i,j,k+1} - c_{i,j,k}} (z_{i,j} - c_{i,j,k}) + M'(1 - \eta_{i,j,k}) & \quad (32) \end{aligned} \right)$$

Then we could obtain the following mixed-integer linear programming (MILP) problem

$$\min \sum_{i=1}^n f_i(x_i) \quad (33)$$

$$\text{s.t.} \quad \sum_{i=1}^n \sum_{j \geq i} b_{i,j} y_{i,j} \leq p_{i,j} \quad (34)$$

$$z_{i,j} = x_i + x_j \quad (35)$$

$$Y_{i,j} = X_i + X_j - Z_{i,j} \quad (36)$$

$$\frac{lb_i \cdot lb_j}{ub_i + ub_j} = a_{i,j,0} \leq a_{i,j,1} \leq \dots \leq a_{i,j,k} \leq a_{i,j,k+1} \dots \leq a_{i,j,m-1} = \frac{ub_i \cdot ub_j}{lb_i + lb_j} \quad (37)$$

$$a_{i,j,k} - M(1 - \lambda_{i,j,k}) \leq y_{i,j} \leq a_{i,j,k+1} - M(1 - \lambda_{i,j,k}) \quad (38)$$

$$\ln(a_{i,j,k}) + \frac{\ln(a_{i,j,k+1}) - \ln(a_{i,j,k})}{a_{i,j,k+1} - a_{i,j,k}} (y_{i,j} - a_{i,j,k}) - M'(1 - \lambda_{i,j,k}) \leq Y_{i,j} \quad (39)$$

$$Y_{i,j} \leq \ln(a_{i,j,k}) + \frac{\ln(a_{i,j,k+1}) - \ln(a_{i,j,k})}{a_{i,j,k+1} - a_{i,j,k}} \quad (40)$$

$$lb = b_{i,0} \leq b_{i,1} \leq \dots \leq b_{i,k} \leq b_{i,k+1} \dots \leq b_{i,m-1} = ub_i \quad (41)$$

$$b_{i,k} - M(1 - \mu_{i,k}) \leq x_i \leq b_{i,k+1} - M(1 - \mu_{i,k}) \quad (42)$$

$$\ln(b_{i,k}) + \frac{\ln(b_{i,k+1}) - \ln(b_{i,k})}{b_{i,k+1} - b_{i,k}} (y_{i,j} - b_{i,k}) - M'(1 - \mu_{i,k}) \leq x_{i,j} \quad (43)$$

$$x_{i,j} \leq \ln(b_{i,k}) + \frac{\ln(b_{i,k+1}) - \ln(b_{i,k})}{b_{i,k+1} - b_{i,k}} (y_{i,j} - b_{i,k}) + M'(1 - \mu_{i,k}) \quad (44)$$

$$lb_i + lb_j = c_{i,j,0} \leq c_{i,j,1} \leq \dots \leq c_{i,j,k} \leq c_{i,j,k+1} \dots \leq c_{i,j,m-1} = ub_i + ub_j \quad (45)$$

$$c_{i,j,k} - M(1 - \eta_{i,j,k}) \leq z_{i,j} \leq c_{i,j,k+1} - M(1 - \eta_{i,j,k}) \quad (46)$$

$$\ln(c_{i,j,k}) + \frac{\ln(c_{i,j,k+1}) - \ln(c_{i,j,k})}{c_{i,j,k+1} - c_{i,j,k}} (z_{i,j} - c_{i,j,k}) - M'(1 - \eta_{i,j,k}) \leq z_{i,j} \quad (47)$$

$$z_{i,j} \leq \ln(c_{i,j,k}) + \frac{\ln(c_{i,j,k+1}) - \ln(c_{i,j,k})}{c_{i,j,k+1} - c_{i,j,k}} (z_{i,j} - c_{i,j,k}) + M'(1 - \eta_{i,j,k}) \quad (48)$$

Situation 2: $\{lb_i\}$ are negative while $\{ub_i\}$ are positive or negative constant values.

If $\{lb_i\}$ are negative, firstly replace

$$x_i' = x_i - lb_i \quad (49)$$

thus

$$0 \leq x_i' \leq ub_i - lb_i \quad (50)$$

$$x_i' + lb_i = x_i \quad (51)$$

and the problem P is transformed into the following problem P'.

Problem P'

$$\min \sum_{i=1}^n f_i(x_i' + lb_i) \quad (52)$$

$$\text{s.t.} \quad \sum_{i=1}^n \sum_{j \geq i} \frac{b_{i,j} x_i' x_j' + b_{i,j} lb_j x_i' + b_{i,j} lb_i x_j' + b_{i,j} lb_i lb_j}{x_i' + x_j' + lb_i + lb_j} \leq p_{i,j} \quad (53)$$

$$0 \leq x_i' \leq ub_i - lb_i \quad \forall i \in I \quad (54)$$

Thus, we can go back to the former defined step 1 and step2 to continue the transforming process.

In general, LTPL method is able to convert any kinds of multi-variable multi-degree constraints into a series of single-variable constraints avoiding introducing a large number of 0–1 variables.

3.3. Flowchart of LTPL method

Fig. 3 shows the process of applying LTPL method. With reference to LTPL method, the reformulation is finished and the conversion from NLP to MILP introduces only $[n \cdot m]$ binary variables instead of $[m^n]$ given that $[n]$ variables are separated into $[m]$ interval partitions.

4. Blending Problem Description

This part aims to prove the applicability of LTPL method and test its performance to blending problems. LTPL method is applied to an automotive gasoline blending problem, which is based on a case conducted by Forbes and Marlin [25]. The blending problem studied here involves simultaneously blending two grades of automotive gasoline with five feedstocks. Its flowsheet is shown as Fig. 4 [25].

For the illustrative purposes of this blending example, the product quality specifications for each grade of gasoline (regular and premium) will be limited to: minimum limits on RON and MON, and a maximum limit on the RVP. Maximum and minimum limits for the products' quantity are also presented in the case. All relevant data [4] are provided in Tables 1–3.

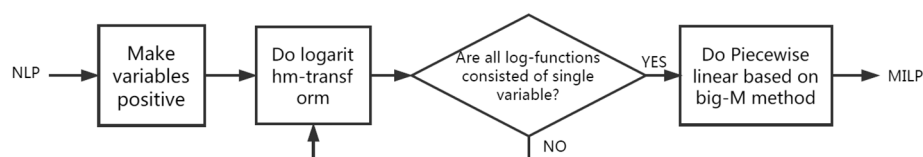


Fig. 3. Flowchart of LTPL method.

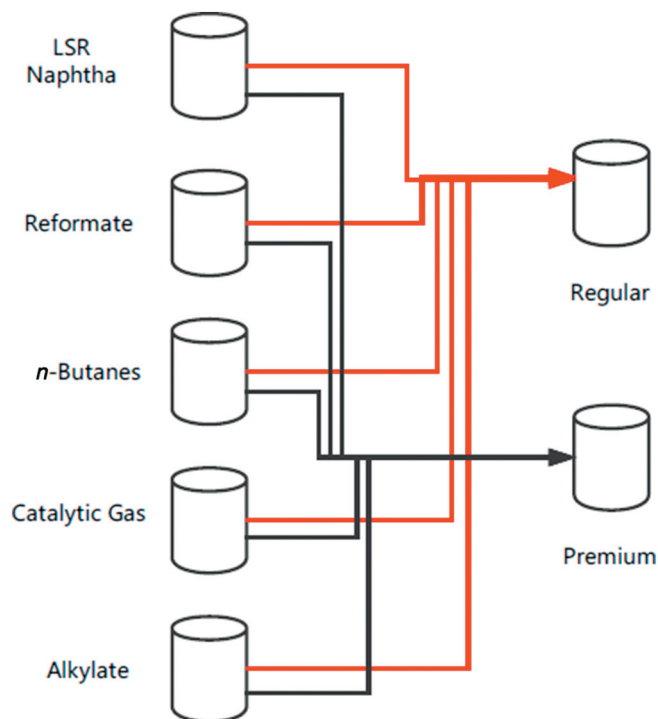


Fig. 4. Flowsheet for gasoline blending case study.

Here, bbl is referred to the unit of measurement of crude oil, barrel: 1 bbl = 42 gal = 0.159 m³. Psi is referred to the unit of measurement of pressure, pounds per square inch: 1 psi = 6894.757 Pa = 0.068046 atm. These explanations are applicable in the following passage unless given special remarks.

Table 1
Production requirements

	Regular	Premium
Value USD · bbl ⁻¹	33.00	37.00
Max. demand limit/bbl · d ⁻¹	80000	10000
Min. demand limit/bbl · d ⁻¹	70000	10000
Min. RON	88.5	91.5
Min. MON	77.0	80.0
Max. RVP/psi	10.8	10.8

Table 2
Feedstock economic data

	Available/bbl · d ⁻¹	Cost/USD · bbl ⁻¹
Reformate	12000	34.00
LSR naphtha	6500	26.00
n-Butane	3000	10.30
Catalytic gasoline	4500	31.30
Alkylate	7000	37.00

Table 3
Feedstock qualities

Feedstock	Reformate (C14a)	LSR naphtha (C10a)	n-Butane (C25a)	Catalytic gasoline (C12a)	Alkylate (C7a)
RON	94.1	70.7	93.8	92.9	95.0
MON	80.5	68.7	90.0	80.8	91.7
Olefin/%	1.0	1.8	0	48.8	0
Aromatics/%	58	2.7	0	22.8	0
RVP/psi	3.8	12.0	138	5.3	6.6

4.1. Process model

For prediction and simulation purposes, a set of equations is established to represent the blending process for RON, MON and RVP. Among a lot of empirical models, the ethyl RT-70 [26] models have been proved to be one of the most accurate models in expressing the mixing rules for octane numbers. They represent blending nonlinearities as functions of compositions of compounds (e.g. olefins, aromatics and so forth) and are widely used in practice. The ethyl RT-70 models are

$$\text{RON}_{\text{blend}} = \overline{\text{RON}} + a_1 (\overline{\text{RON} \cdot s} - \overline{\text{RON}} \overline{s}) + a_2 (\overline{o^2} - \overline{o}^2) + a_3 (\overline{A^2} - \overline{A}^2) \quad (55)$$

$$\text{MON}_{\text{blend}} = \overline{\text{MON}} + b_1 (\overline{\text{MON} \cdot s} - \overline{\text{MON}} \overline{s}) + b_2 (\overline{o^2} - \overline{o}^2) + b_3 (\overline{A^2} - \overline{A}^2) / 100^2 \quad (56)$$

where:

RON each blend component's research octane number.
 MON each blend component's motor octane number.
 s each blend component's sensitivity (RON-MON).
 o each blend component's olefin content (% by volume).
 A each blend component's aromatic content (% by volume).
 $\overline{X}$ the volume fraction-weighted X of the mixer, $X = \text{RON}, s, \text{MON}, \text{RON} \cdot s, \text{MON} \cdot s, o, A, o^2, A^2$

$a_1, a_2, a_3, b_1, b_2, b_3$ are model parameters. Referring to 135 blending experiments, Healy *et al.* [26] used data from those experiments for regression and determined the model parameter values as follows

$$a_1 = 0.03224, a_2 = 0.00101, a_3 = 0, b_1 = 0.04450, b_2 = 0.00081, b_3 = -0.0645$$

As for predicting the Reid vapor pressure quality of the mixer, the blending index approach [27] developed by the Chevron Research Company is regarded to be one of the easiest method and is often applied for predicting blended RVP. We adopted this model in the case

$$\text{RVP}_{\text{blend}} = \left[\sum_{i=1}^n x_i (\text{RVP}_i)^{1.25} \right]^{0.8} \quad (57)$$

where:

x_i the volume fraction of component i.
 n the number of components in the blend.
 RVP_i each blend component's Reid vapor pressure.

The above equations will be used to for the blending process in the latter part, which constitute the nonlinear constraints of the programming problem.

4.2. Programming expressions

The performance of this method is measured by the comparison between it and the theoretical maximum profit. The blending optimization problem is often consisted of the following parts:

objective function = max profit

constraints $\left\{ \begin{array}{l} \text{material balance} \\ \text{quantities limits} \\ \text{RON constraints} \\ \text{MON constraints} \\ \text{RVP constraints} \end{array} \right\}$

Table 4
Quantities consuming of feedstocks

	Quantities/bbl · d ⁻¹
Reformate	6431
LSR naphtha	3992
<i>n</i> -Butane	348
Catalytic gasoline	4500
Alkylate	2730

Table 5
Information of products

	RON	MON	RVP	Quantities/bbl · d ⁻¹
Regular gasoline	88.5	77	10.8	8000
Premium gasoline	91.5	80	10.8	10000

This is an NLP problem and in this case it can be formulated as:

$$\max \text{ profit} = \sum_{i=1}^{n=2} \text{price}_i P_i - \sum_{j=1}^{n=5} \text{price}_j V_j \quad (58)$$

$$\text{s.t.} \quad LB_i \leq P_i \leq UB_i \quad \forall i \quad (59)$$

$$LB_j \leq V_j \leq UB_j \quad \forall j \quad (60)$$

$$\sum_{j=1}^{n=5} x_{j,i} = 1 \quad \forall i, j \quad (61)$$

$$V_j = \sum_{i=1}^{n=2} x_{j,i} P_i \quad \forall i, j \quad (62)$$

$$\begin{aligned} \text{RON}_i = & \sum_{j=1}^{n=5} x_{j,i} \text{RON}_j + a_1 \left(\sum_{j=1}^{n=5} x_{j,i} \text{RON}_j s_j - \left(\sum_{j=1}^{n=5} x_{j,i} \text{RON} \right) \left(\sum_{j=1}^{n=5} x_{j,i} s_j \right) \right) \\ & + a_2 \left(\sum_{j=1}^{n=5} x_{j,i} O_j^2 - \left(\sum_{j=1}^{n=5} x_{j,i} O_j \right)^2 \right) + a_3 \left(\sum_{j=1}^{n=5} x_{j,i} A_j^2 - \left(\sum_{j=1}^{n=5} x_{j,i} A_j \right)^2 \right) \quad \forall i \end{aligned} \quad (63)$$

$$\begin{aligned} \text{MON}_i = & \sum_{j=1}^{n=5} x_{j,i} \text{MON}_j + b_1 \left(\sum_{j=1}^{n=5} x_{j,i} \text{MON}_j s_j - \left(\sum_{j=1}^{n=5} x_{j,i} \text{MON} \right) \left(\sum_{j=1}^{n=5} x_{j,i} s_j \right) \right) \\ & + b_2 \left(\sum_{j=1}^{n=5} x_{j,i} O_j^2 - \left(\sum_{j=1}^{n=5} x_{j,i} O_j \right)^2 \right) + b_3 \left(\left(\sum_{j=1}^{n=5} x_{j,i} A_j^2 - \left(\sum_{j=1}^{n=5} x_{j,i} A_j \right)^2 \right) / 100 \right)^2 \quad \forall i \end{aligned} \quad (64)$$

$$\text{RVP}_i = \left[\sum_{j=1}^{n=5} x_{j,i} (\text{RVP}_j)^{1.25} \right]^{0.8} \quad \forall i, j \quad (65)$$

$$\text{RONstandard}_i \leq \text{RON}_i \leq \text{RONstandard}_i + 0.5 \quad \forall i \quad (66)$$

$$\text{MONstandard}_i \leq \text{MON}_i \leq \text{MONstandard}_i + 1.0 \quad \forall i \quad (67)$$

$$\text{RVP}_i \leq \text{RVPstandard}_i \quad \forall i \quad (68)$$

where:

- i* denote 2 blend products, regular gasoline and premium gasoline
- j* denote 5 feedstocks, Reformate, LSR naphtha, *n*-Butane, Catalytic gasoline, Alkylate
- P_i* volume of products
- V_j* volume of feedstocks
- x_{j, i}* volume fraction of component *j* in product *i*
- price* sales price of products or costs of feedstocks
- LB* lower bound of products or feedstocks
- UB* upper bound of products or feedstocks

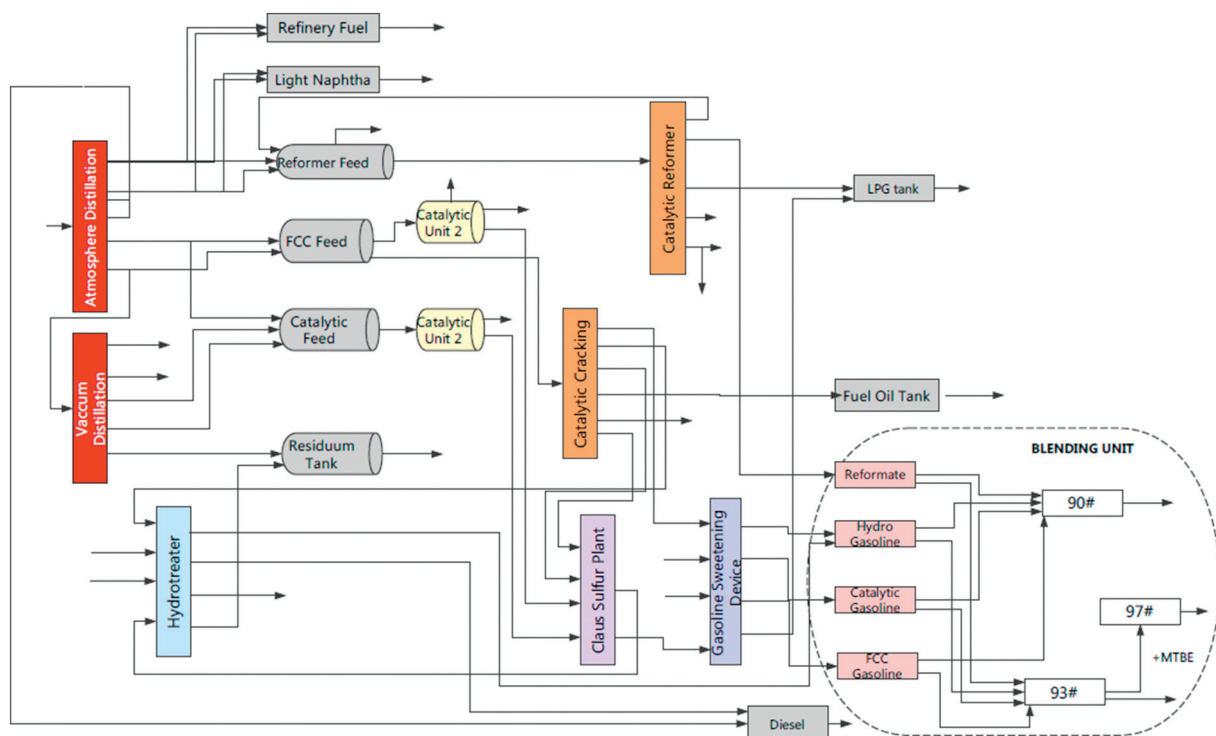


Fig. 5. A petroleum refinery flowsheet chart.

Table 6
Units parameters

	Lower bound/t	Upper bound/t	Loss rate/%	Siding yields
Atmosphere distillation	4200	7500	0.5	0.07, 0.22, 0.06, 0.645
Vacuum distillation	500	900	0	0.001, 0.25, 0.25, 0.499
Catalytic reformer	240	300	1.2	0.013, 0.87, 0.05, 0.005, 0.05
Catalytic cracking	2800	3600	1.2	0.42, 0.22, 0.105, 0.068, 0.105, 0.07
Hydrotreater	600	900	0.5	0.008 (+0.005), 0.9866 (+0.005), 0.0004 (+0.005)
Claus sulfur	500	900	0.5	0.971–0.995
Gasoline sweetening device	1500	2700	0	0.9998–1

Table 7
Components oils properties

	RON	MON	Olefin/%	Aromatics/%
Reformate	94.1	80.5	1.0	58.0
Hydro gasoline	88.5	86.5	14	65.9
Catalytic gasoline	92.9	80.8	48.8	22.8
FCC gasoline	93.0	83.0	28	27.3
MTBE	117	101	0	0

RONstandard the RON quality demand of products
 MONstandard the MON quality demand of products
 RVPstandard the RVP quality demand of products

The above model is an NLP problem, and LTPL method has been applied to convert it from an NLP to MILP. The calculation is performed on LINGO®, and the results are shown in the following part.

4.3. Performance

To meet the quality requirements and quantity limits, the approximation global optimum solutions are listed in Tables 4 and 5. The profit of the whole blending system calculated by LTPL method is 65565USD, and achieves 66136.20USD given smaller partitions. It is better than the results, 64622USD and 65491USD, which were given by methods proposed by A. Singh [4]. As A. Singh [4] has pointed out the ideal profit for the problem is 66276.55USD, the deviation is only 0.21%.

5. Industrial Case Study

LTPL method has been applied to a petroleum refinery to determine the maximum profit of the whole blending process [28]. The refinery includes seven processing units and a gasoline blending workshop as well as some auxiliary units. The seven main processing units are atmosphere distillation, vacuum distillation, hydrotreater, catalytic reformer, catalytic cracking, claus sulfur plant, and gasoline sweetening device. Input materials are crude oil and MTBE. The refinery produces more than 15 products and the main products are gasoline 90#, 93#, 97#, straight-run gasoline, residuum, LPG, fuel oil, refinery fuel, and light naphtha. The flowchart is shown in Fig. 5, and some related data and parameters [28] are given in Tables 6–8.

LTPL method was applied to this case, converting the non-convex nonlinear programming model into a MILP model by piecewise linearization. Computational results are shown in Table 9, compared with results calculated by B-and-B global solver. By using LTPL method, approximate maximum profit obtained by the gasoline blending workshop is 4.1×10^6 CNY, and the variation is only 4.1% compare with the

Table 8
Prices data

Material	Crude oil	MTBE	90#	93#	97#	LPG	Residuum	Fuel oil	Refinery fuel	Light naphtha	Straight-run gasoline
Price/CNY·t ⁻¹	2572	5500	7913	8441	8997	2710	2200	2550	6900	4670	4670

Table 9
Computational results

Profit/quantities	Value for MILP	Value for NLP	Variation
Profit earned/CNY	4130048	4309688	4.1%
Running time	1 h 27 min	4 h 6 min	
Crude oil/t	6974.5912	7028.462	0.7%
MTBE/t	50.0000	50.0000	0.0%
90# gasoline/t	1869.2302	1889.838	1.1%
93# gasoline/t	110.2712	114.1103	3.4%
97# gasoline/t	226.0144	235.4174	4.0%
LPG/t	505.6252	506.5827	0.2%
Residuum/t	438.3451	444.1159	1.3%
Fuel oil/t	254.5236	260.2989	2.2%
Refinery fuel/t	229.1523	230.1050	0.4%
Light naphtha/t	0.0000	0.0000	0.0%
Straight-run gasoline/t	5.0187	5.346535	6.1%

profit 4.3×10^6 CNY calculated by the NLP global solver, while time cost of LTPL method is much shorter than that of NLP model calculated by B-and-B global solver.

The variation of straight-run gasoline is 6.1% which is the largest one among all the computational results. MTBE and light naphtha have no deviation, and the crude oil, LPG, refinery fuel achieve very good accuracy level whose variations are less than 1%.

Despite of the existence of small deviation, the approximate linearization method could quickly give a pretty good solution and direct operators to search for global optimum solutions close to the present solution given by approximation piecewise linearization method. In this way complicated large-scale non-convex nonlinear programming problems can be solved accurately and quickly.

6. Summary and Conclusion

When applying piecewise linear transformation approach to non-convex programmings, it often concerns the scale of problem size and computational time, especially to the product blending problems in complex refinery plants. Conventional piecewise linearization methods focus on univariate expressions or linear combinations of them. LTPL method aims to convert complex multi-variable multi-degree expressions into single variable expressions by doing a series of logarithm transforms, and then reform new MILP programming problems by piecewise linearization based on big-M method. According to the above discussion, the number of 0–1 variables could be reduced greatly by LTPL method and the scale of models can be controlled. This method has been applied successfully to a gasoline blending process and an industrial petroleum refinery plant with a gasoline blending workshop. It is able to find out approximate global optimum solutions in a short time.

In conclusion, LTPL method is able to scale down the linearization size of problems containing mixed multi-degree constraints, and convert complicated expressions of NLP into MILP. LTPL method could be more widely applied in different practical situations and solve out nearly-accurate solution within a short time.

References

- [1] C.W. DeWitt, S.L. Leon, D.W. Allan, A.B. Donald, A.M. Simon, OMEGA: An improved gasoline blending system for Texaco, *Interfaces* 19 (1) (1989) 85–101.
- [2] C.A. Méndez, I.E. Grossmann, I. Harjunkoski, P. Kaboré, A simultaneous optimization approach for off-line blending and scheduling of oil-refinery operations, *Comput. Chem. Eng.* 30 (4) (2005) 614–634.
- [3] J. Li, I. Karimi, Scheduling gasoline blending operations from recipe determination to shipping using unit slots, *Ind. Eng. Chem. Res.* 50 (15) (2011) 9156–9174.
- [4] A. Singh, J. Forbes, P. Vermeer, S. Woo, Model-based real-time optimization of automotive gasoline blending operations, *J. Process Control* 10 (1) (2000) 43–58.
- [5] P. Castillo, P. Castro, V. Mahalec, Global optimization of nonlinear blend-scheduling problems, *Engineering* 3 (2) (2017) 188–201.
- [6] J. Pinto, M. Joly, L. Moro, Planning and scheduling models for refinery operations, *Comput. Chem. Eng.* 24 (9–10) (2000) 2259–2276.
- [7] J. Cerdá, P. Pautasso, D. Cafaro, A cost-effective model for the gasoline blend optimization problem, *AIChE J.* 62 (9) (2016) 3002–3019.
- [8] T.Y. Chi, Solving of gasoline blending nonlinear model and oil online blending, Ph. D. Thesis, Dalian University of Technology, China, 2007.
- [9] N. Boland, T. Kalinowski, F. Rigterink, New multi-commodity flow formulations for the pooling problem, *J. Glob. Optim.* 66 (4) (2016) 669–710.
- [10] H. Serali, A. Alameddine, A new reformulation-linearization technique for bilinear programming problems, *J. Glob. Optim.* 2 (4) (1992) 379–410.
- [11] H. Ryoo, N. Sahinidis, A branch-and-reduce approach to global optimization, *J. Glob. Optim.* 8 (2) (1996) 107–138.
- [12] E.M.B. Smith, C.C. Pantelides, A symbolic reformulation/spatial branch-and-bound algorithm for the global optimisation of nonconvex MINLPs, *Comput. Chem. Eng.* 23 (4–5) (1999) 457–478.
- [13] M. Tawarmalani, N. Sahinidis, A polyhedral branch-and-cut approach to global optimization, *Math. Program.* 103 (2) (2005) 225–249.
- [14] R. Misener, C. Floudas, ANTIGONE: Algorithms for continuous/integer global optimization of nonlinear equations, *J. Glob. Optim.* 59 (2–3) (2014) 503–526.
- [15] T. Achterberg, SCIP: Solving constraint integer programs, *Math. Program. Comput.* 1 (1) (2009) 1–41.
- [16] P. Belotti, J. Lee, L. Liberti, F. Margot, A. Wächter, Branching and bounds tightening techniques for non-convex MINLP, *Optim. Methods Softw.* 24 (4–5) (2009) 597–634.
- [17] W.I. Zangwill, The Piecewise Concave Function [J], *Manag. Sci.* 13 (11) (1967) 900–912.
- [18] E. Smith, Global optimisation of nonconvex MINLPs, *Comput. Chem. Eng.* 21 (1–2) (1997) 791–796.
- [19] H. Serali, C. Tuncbilek, A global optimization algorithm for polynomial programming problems using a reformulation-linearization technique, *J. Glob. Optim.* 2 (1) (1992) 101–112.
- [20] A. Geoffrion, Objective function approximations in mathematical programming, *Math. Program.* 13 (1) (1977) 23–37.
- [21] F. Güder, J. Morris, Optimal objective function approximation for separable convex quadratic programming, *Math. Program.* 67 (1–3) (1994) 133–142.
- [22] C. Huang, An effective linear approximation method for separable programming problems, *Appl. Math. Comput.* 215 (4) (2009) 1496–1506.
- [23] M. Lin, J. Carlsson, A review of piecewise linearization methods, *Math. Probl. Eng.* 2013 (2013) 1–8.
- [24] M. Padberg, Approximating separable nonlinear functions via mixed zero-one programs, *Oper. Res. Lett.* 27 (1) (2000) 1–5.
- [25] J. Forbes, T. Marlin, Model accuracy for economic optimizing controllers: The bias update case, *Ind. Eng. Chem. Res.* 33 (8) (1994) 1919–1929.
- [26] W.C. Healy, C.W. Maassen, R.T. Peterson, A new approach to blending octanes, *API Div. Refin.* 39 (1959) 132–192.
- [27] J.H. Gary, G.E. Handwerk, *Petroleum Refining Technology and Economics*, 2nd edition Marcel Dekker, New York, 1994.
- [28] X.R. He, *Graphic I/O Petrochemical Industry Modeling System in Integration of Refining and Chemical Industry*, China Petrochemical Press, Beijing, 2006 12–20 (in Chinese).