

Constrained Operational Optimization of a Distillation Unit in Refineries with Varying Feedstock Properties

Qingda Chen, *Student Member, IEEE*, Jinliang Ding, *Senior Member, IEEE*, Shengxiang Yang, *Senior Member, IEEE*, and Tianyou Chai, *Fellow, IEEE*

Abstract—This paper studies the challenging operational optimization problem of a distillation unit under varying feedstock properties (e.g., density and carbon content). This problem, in which changes in the feedstock properties are incorporated, aims to quickly obtain the operating variables that control the operating condition of the distillation unit. To solve this problem, we first model this operational optimization problem considering the ever-changing feedstock properties and practical technological constraints. Then, we propose an efficient soft-sensing strategy to rapidly measure the feedstock properties. Finally, motivated by the challenges caused by the varying feedstock properties, product yield and tray temperature constraints, we propose an optimization algorithm with global search and self-repair capabilities to optimize the operating variables of the distillation unit. The proposed algorithm integrates the optimization time and survival information of each individual into the proposed mutation strategy to improve its global search capability in the irregular feasible region of the operating variables. Based on the ranking and survival information of each individual, the adaptive strategies of the mutation factor and crossover probability are designed to balance the exploration and exploitation capabilities of the optimization algorithm. Subsequently, we propose an effective correction strategy to correct the infeasible operating variables and improve the optimization efficiency of the algorithm. Computational experiments on practical production data show the accuracy of the soft-sensing model and the superiority of the optimization algorithm for operational optimization of the distillation unit.

¹ **Index Terms**—Operating variables, distillation unit, mutation strategy, adaptive strategies, correction strategy.

Indices

l	Product index.
w	Pump-around index.
s	Tray index.
j	Operating variable index.
b	Specific temperature index.
t	Time index.

Variables

SS	Flowrate of steam.
TF	Feedstock temperature in the bottom of column.
PL_w	Flowrate of distillate in the w th pump-around.
PQ_w	Calorific value of energy recovery in the w th pump-around.
y_l	Yield of product l .

FT_l	Final cut temperature of product l .
IT_l	Initial cut temperature of product l .
T	Temperature.
yp_T	Distillate yield of the feedstock at T .
yp_{FT_l}	Distillate yield from the feedstock at FT_l .
yp_{IT_l}	Distillate yield from the feedstock at IT_l .
FF	Flowrate of feedstock.
TT_s	Temperature of the s th tray.
ST_b	The b th specific temperature.
\mathbf{X}	Decision variable vector.
x_j	The j th operating variable in \mathbf{X} .
$c_m(t)$	The m th time-varying coefficient.

Parameters

Pr_l	Unit price of product l (yuan/m ³).
Pr_{res}	Unit price of residual oil (yuan/m ³).
CSS	Unit price of steam (yuan/m ³).
EP	Unit price of energy (yuan/KJ).
CFF	Unit price of feedstock (yuan/m ³).
LT_s	Lower bound of the s th tray temperature.
UT_s	Upper bound of the s th tray temperature.
LF_l	Lower bound of the flowrate of product l .
UF_l	Upper bound of the flowrate of product l .
CTF	Unit price of heating feedstock (yuan/(°C · m ³)).
lx_j	Lower bound of x_j .
ux_j	Upper bound of x_j .
ρ	Heat transfer efficiency in pump-arounds.
CS	Subsequent processing cost of products.
nv	Number of the operating variables.
TT_0	Normal atmospheric temperature.
d_r	Label number of the r th key tray.

I. INTRODUCTION

OPERATION OPTIMIZATION problems in industrial production have become increasingly important in the face of increasing global competition and environmental requirements, and they aim to determine the best values of the variables (e.g., temperature and material flow rate) that control the operating condition of real plant equipments based on production constraints [1]. Distillation unit (DU), a common separating unit used in refineries [2], separates feedstock into a series of intermediate products (hereafter called “products”). In the DU, the feedstock properties and the setting values of the operating variables (e.g., steam flow) control the tray temperatures that influence distillate yields and profitability of oil refineries. Optimizing the DU operating variables with consideration of the feedstock properties and production constraints is thus a challenging research problem of considerable importance for decreasing the production costs and improving the production benefit.

However, influenced by pipeline transportation, feedstock mixing, and stratification phenomena, the feedstock properties vary greatly and irregularly. The operating variables must be determined quickly (Note: within two minutes) once feedstock properties are detected to change. A feedstock consists of thousands of hydrocarbons, each with

This work was supported by the National Natural Science Foundation of China under Grand 61525302, 61590922, the National Key Research and Development Program of China under Grant 2018YFB1701104, the Project of Ministry of Industry and Information Technology of China under Grand 20171122-6, and the Fundamental Research Funds for the Central Universities under Grand N160801001, N161608001.

The authors are with the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang, 110819, China (e-mail: cq0309@126.com; jlding@mail.neu.edu.cn; syang@dmu.ac.uk; tychai@mail.neu.edu.cn).

S. Yang is also with the Centre for Computational Intelligence, School of Computer Science and Informatics, De Montfort University, Leicester, LE1 9BH, U. K.

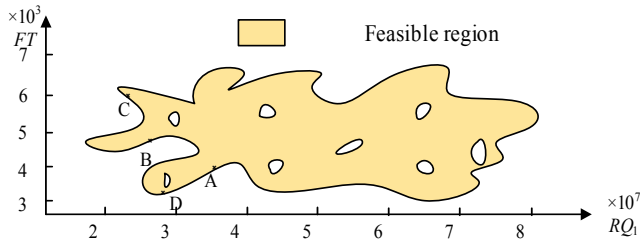


Fig. 1. Schematic diagram of the feasible range of two operating variables. A, B, C, and D denote different optimal values of operating variables under different feedstock properties

its own boiling point. Therefore, rapidly measuring the feedstock properties is difficult work. For the DU, the changing feedstock properties cause the optimal operating variables to vary irregularly, as shown in Fig. 1. Hence, designing promising optimization strategies to locate the best solution quickly is a considerable challenge. Additionally, the nonlinear relationships between the constraints (i.e., product yield and tray temperature constraints) and the DU operating variables make it impossible to determine the feasible regions. Thereafter, correcting the infeasible solutions is difficult.

Over the years, different approaches have been used to solve the operational optimization problem of DU. Liau *et al.* [4] adopted the optimization toolbox provided by MATLAB to optimize the DU operating variables., Ochoa-Estopoer *et al.* [5], [6] employed simulated annealing to solve the operational optimization problem of DU. However, using the simulated annealing method requires more time to achieve convergence. Mahalec *et al.* [8] and Fu *et al.* [9] adopted the generalized gradient method to search for the best operating variables. Osulate *et al.* [7], Li *et al.* [10], and Raghunathan *et al.* [11] used sequential quadratic programming (SQP) to optimize the operating variables. Lopez *et al.* [12] adopted the CONOPT solver to obtain the best operating variables. Both SQP and CONOPT are too time consuming for the DU operation problem discussed in this paper. Additionally, feedstock properties are not considered in [4]-[12]. In fact, the feedstock properties usually change after half an hour or more, resulting in changes in the optimal operating variables.

The feedstock properties are characterized by their true boiling point (TBP, the percentage of weight of the distillate that is obtained from the feedstock at different temperatures) property [3], [8], and we propose a soft-sensing strategy to obtain this property quickly. Optimizing the DU operating variables with consideration of the feedstock properties needs to calculate the tray temperatures according to the operating variables. The rigorous models between the operating variables and the tray temperatures involve some coefficients (i.e., coefficients in heat of vaporization calculation) that vary irregularly with feedstock properties. Re-determining these coefficients spends much time (more than an hour) once the feedstock properties change. However, the operating variables must be determined quickly (note: less than two minutes) once feedstock properties change. So it is unrealistic to establish the rigorous model of operational optimization problem when the varying feedstock properties are considered. Using the artificial neural network (ANN) proposed in [13] to establish the model between the operating variables and the tray temperatures does not involve the coefficients related to the feedstock properties, and this model can be updated in a short time (note: less than 30 seconds). Therefore, this paper models the operational optimization problem based on the ANN. Evolutionary algorithms have been successfully applied to solve many optimization problems [14]-[22], and the differential

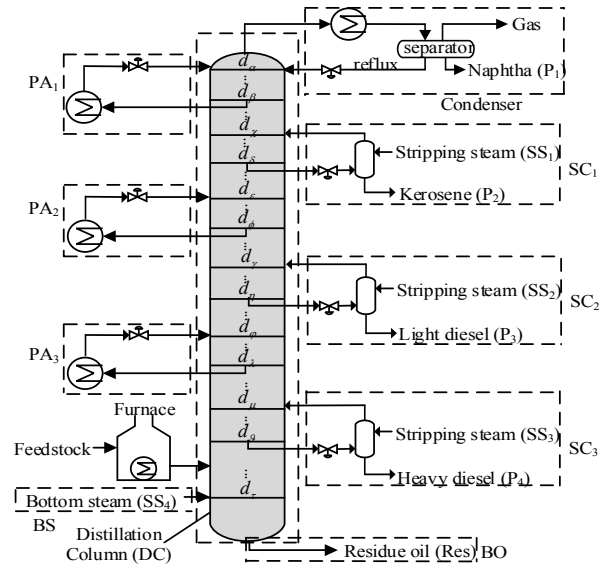


Fig. 2. Schematic diagram of distillation unit

evolution (DE) algorithm proposed by Price *et al.* [16] has shown its simplicity and robustness in solving uncertain problems [17] and constrained problems [18]. A good review can be found in [22]. Motivated by the challenges arising in the DU operation problem, we propose an operational optimization algorithm (OOA) to quickly locate the best operating variables that control the operation of the DU. The contributions of this paper are as follows:

- 1) This paper uses the ANN to establish a relationship between the operating variables and the tray temperatures. Based on this, we model the operational optimization problem with consideration of the tray temperature, device production capability, and product demand constraints.
- 2) We design a soft-sensing strategy to measure the feedstock TBP property. The designed soft-sensing strategy can obtain the feedstock TBP quickly (less than 30 seconds).
- 3) We propose a mutation strategy with consideration of the survival status of an individual and the consumed CPU time of the optimization process. The proposed mutation strategy can adaptively adjust the information proportion among current the individual, the random individual, and the best individual in the mutant individuals of each generation.
- 4) To track the position of the optimal solution when it moves irregularly with the feedstock properties, this paper proposes a parameter adaptation scheme based on the individual survival status and ranking information of each individual. The proposed strategy can adjust the mutation factor and the crossover probability adaptively, balancing exploitation and exploration capabilities of the algorithm.
- 5) To remove the infeasible operating variables, we propose an effective correction strategy based on the tray temperature, product demand, and device production capability constraints. The proposed correction strategy can repair the infeasible operating variables quickly and improve the optimization efficiency.

The rest of this paper is organized as follows. Section II presents the operational optimization problem of the DU. Section III discusses the details of the OOA. Section IV reports the experimental results with analysis, and Section VI gives concluding remarks.

II. PROBLEM FORMULATION

As shown in Fig. 2, a DU usually consists of a distillation column (DC), a condenser, a heating furnace, a residual oil

outlet (BO), a bottom steam inlet (BS), pump-arounds (PA₁, PA₂, and PA₃), and side-stripper columns (SC₁, SC₂ and SC₃), which is similar to that used in [4]–[9]. Note that Greek notation (i.e., $\alpha, \beta, \chi, \delta, \varepsilon, \phi, \gamma, \eta, \varphi, \lambda, \mu, \vartheta$, and τ), as shown in Fig. 2, denotes the indices of key tray labels, and that there are d_τ trays in the DC. In this section, the distillation process is described first, followed by its operational optimization problem and model.

A. Distillation Process

After being heated to a certain temperature (i.e., TF) in a furnace, the feedstock becomes a vapor-liquid mixture that consists of hydrocarbons with different boiling points. Then, the mixture vaporizes after entering the DC. Simultaneously, superheated bottom steam (SS₄) is added to the DU at the BS and goes upstream, merging with the mixture to enhance its separation and prevent excessive thermal cracking of the feedstock. Unvaporized residual oil falls to the bottom of the DC and is drawn out of the DC through the BO. Light hydrocarbons from the feedstock move up the column. Because the tray temperatures in the DC form a low-to-high temperature distribution from top-to-bottom (the temperature of the first tray is the lowest, and the bottom tray is the highest), hydrocarbons are liquefied into distillate on different trays, thereby forming different products, i.e., P₁ (Naphtha), P₂ (Kerosene), P₃ (Diesel), and P₄ (Gas oil), which are drawn out of the DC at SC₁, SC₂, and SC₃, respectively. Note that a product includes distillate from different trays. For example, P₃ includes distillate from the d_α th to d_η th trays. The lighter components (e.g., pentanes) in the mixture travel up the DC and leave from the top of the DC to the condenser, and a part of the condensed distillate is returned to the DU to adjust the tray temperatures. The remaining distillate becomes gas or naphtha. Additionally, pump-arounds are used to adjust the distillate flowrate and recycle excess heat in the DC. Stripping steam (i.e., SS₁, SS₂, and SS₃) is added to the side strippers to remove the volatile elements in products. In this paper, the trays connected with side strippers are called the key trays (i.e., the d_α th, d_δ th, d_η th, and d_θ th trays).

B. Operational optimization Problem and Model of DU

In the DU, the income is from products, residual oil, and the heat recovery from pump-arounds. The production cost includes the energy cost of heating the feedstock, feedstock cost, steam cost, and subsequent processing cost of products. The profitability equals sale minus production cost, which can be formulated as Equation (1):

$$\max f(\mathbf{X}) = \sum_{l=1}^4 (y_l \times FF \times Pr_l) + (1 - \sum_{l=1}^4 y_l) \times FF \times Pr_{res} + \sum_{w=1}^3 (PQ_w \times \rho \times EP) - SS \times CSS - CTF \times FF \times (TF - TF_0) - FF \times CFF - CS \quad (1)$$

where $\mathbf{X} = [PQ_1, PQ_2, PQ_3, PL_1, PL_2, PL_3, TF, SS]$, and $y_l \times FF \times Pr_l$ income of product l .

$\sum_{l=1}^4 (y_l \times FF) \times Pr_l$ total income of all products.

$(1 - \sum_{l=1}^4 y_l) \times FF$ flowrate of residual oil.

$(1 - \sum_{l=1}^4 y_l) \times FF \times Pr_{res}$ income of residual oil.

$PQ_w \times \rho$ heat recovery from the w th pump-around.

$PQ_w \times \rho \times EP$ economic value of the heat recovery from the w th pump-around.

$\sum_{w=1}^3 (PQ_w \times \rho \times EP)$ total economic value of the heat recovery

$SS \times CSS$ steam cost.

$CTF \times FF \times (TF - TF_0)$ energy cost of heating the feedstock.

$CFF \times FF$ feedstock cost

Increasing PQ_1 , PQ_2 , or PQ_3 will decrease the corresponding tray temperatures and thus the operating costs.

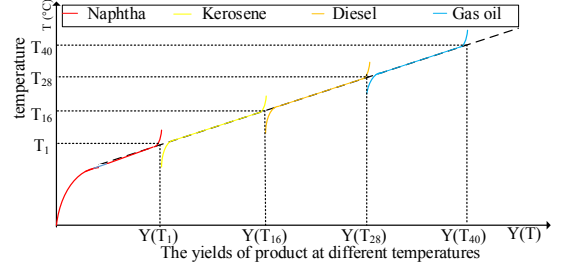


Fig. 3. Feedstock TBP curve

Superfluous heat energy will exit the DC with the increment of PL_1 , PL_2 , and PL_3 , thus decreasing tray temperatures. Increasing CSS increases the heat energy in the DC, elevating the tray temperatures. Increasing feedstock temperature increases the heat energy in the DC and tray temperatures. The nonlinear relationship between the operating variables and the s th tray temperature is provided in Equation (2).

$$TT_s = g_s(x_1, \dots, x_j, \dots, x_m) \quad (2)$$

where $g_s(\cdot)$ is the ANN model that denotes the relationship between the s th tray temperature and the operating variables.

The operational optimization problem of the DU needs to satisfy the following constraints:

Device production capability constraints: Each operating variable is constrained by lower and upper bounds.

$$lx_j \leq x_j \leq ux_j \quad (3)$$

Key Tray Temperature Constraints: Key tray temperature must remain within specific ranges to ensure product quality.

$$LT_s \leq TT_s \leq UT_s \text{ where } s = d_\alpha, d_\delta, d_\eta \text{ and } d_\theta \quad (4)$$

To determine the economic profit of the DU, the product yields must be determined according to the feedstock TBP property. The feedstock TBP property is usually expressed as a third-order polynomial, as shown in Equation (5) [3].

$$yp_T = Y(T) = c_0(t) + c_1(t)T + c_2(t)T^2 + c_3(t)T^3 \quad (5)$$

The feedstock TBP property curve is shown in Fig. 3. Each product has two important temperatures: the initial cut temperature (ICT, i.e., the temperature at which a specific product starts to boil) and the final cut temperature (FCT, i.e., the temperature at which a specific product has boiled completely). Generally, the ICT and FCT of a product equal the temperature of its corresponding key trays, respectively.

For ICTs, $IT_1 = TT_0$, $IT_2 = TT_{d_\alpha}$, $IT_3 = TT_{d_\delta}$, $IT_4 = TT_{d_\eta}$.

For FCTs, $FT_1 = TT_{d_\alpha}$, $FT_2 = TT_{d_\delta}$, $FT_3 = TT_{d_\eta}$, $FT_4 = TT_{d_\theta}$.

The yield of a product is equal to the distillate yield from the feedstock at its FCT minus that of the feedstock at its ICT. For example, the yield of light diesel (i.e., P₃) is thus equal to $yp_{TT_{d_\eta}} - yp_{TT_{d_\delta}}$.

Label Constraints between Key Trays and Products: Each product drawn out of the DC through a side stream corresponds to a key tray. The relationships between the labels of key trays and products are as follows:

$$s = \begin{cases} d_\alpha, & \text{if } l = 1 \\ d_\delta, & \text{if } l = 2 \\ d_\eta, & \text{if } l = 3 \\ d_\theta, & \text{if } l = 4 \end{cases} \quad (6)$$

Constraints of the Cut Temperatures of Products: The ICT of the l th product is equal to the FCT of the $(l-1)$ th product.

$$\begin{cases} IT_1 = TT_0 \\ IT_l = FT_{l-1}, & l = 2, 3, 4 \end{cases} \quad (7)$$

Relationships between the Cut Temperatures of Products and Key Tray Temperatures: The FCT of a product is equal to its

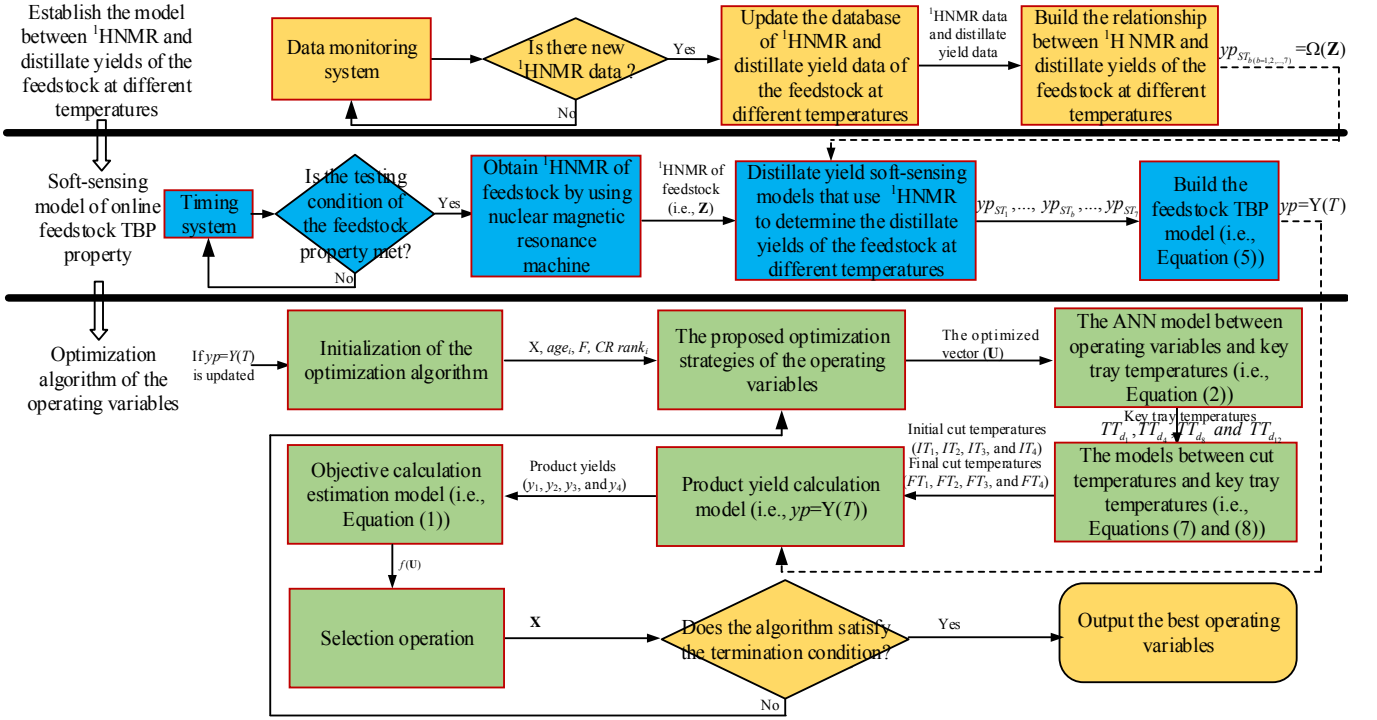


Fig. 4. The operational optimization strategy with combination of feedstock TBP property corresponding key tray temperature shown in Equation (6).

$$FT_l = TT_s \quad (8)$$

Product Yield Constraints: The flowrate of each product must be within the feasible range of fossil fuel demands.

$$LF_l \leq FF \times y_l \leq UF_l \quad (9)$$

$$y_l = yp_{FT_l} - yp_{IT_l}, \quad l = 1, 2, 3, 4 \quad (10)$$

$$yp_{FT_l} = Y(FT_l) = c_0(t) + c_1(t)FT_l + c_2(t)(FT_l)^2 + c_3(t)(FT_l)^3 \quad (11)$$

$$yp_{IT_l} = Y(IT_l) = c_0(t) + c_1(t)IT_l + c_2(t)(IT_l)^2 + c_3(t)(IT_l)^3 \quad (12)$$

III. OPERATIONAL OPTIMIZATION STRATEGY OF THE DU WITH CONSIDERATION OF FEEDSTOCK TBP PROPERTY

To locate the best operating variables of the DU when the feedstock properties change, we propose an operational optimization strategy combined with the feedstock TBP property, which is shown in Fig. 4. This paper uses nuclear magnetic resonance spectrum data (hereafter called “¹H NMR”) of the feedstock to obtain feedstock properties.

In the proposed strategy, we first establish the models (i.e., $yp_{ST_b(b-1,2,\dots,7)} = \Omega_{ST_b(b-1,2,\dots,7)}(\mathbf{Z})$, where \mathbf{Z} denotes an array of 700-dimensional ¹H NMR data) that use ¹H NMR to determine the distillate yields of the feedstock at several specific temperatures (hereafter called “distillate yield model”) according to the updated data (i.e., ¹H NMR and the distillate yield data of the feedstock at different temperatures). Note that the distillate yield model and the database used to establish this model are updated daily. Subsequently, these models are sent to the on-line soft-sensing strategy of the feedstock TBP property. When the testing condition of the feedstock TBP property is met, a sample of feedstock is fed into the nuclear magnetic resonance machine to yield the ¹H NMR spectrum of the feedstock (i.e., \mathbf{Z}). Afterwards, $yp_{ST_1} \dots yp_{ST_b} \dots yp_{ST_7}$ are determined according to $yp_{ST_b(b-1,2,\dots,7)} = \Omega_{ST_b(b-1,2,\dots,7)}(\mathbf{Z})$. Let T be equal to $ST_1, \dots, ST_b, \dots, ST_7$ and yp_T equal to $yp_{ST_1} \dots yp_{ST_b} \dots yp_{ST_7}$, respectively. They are inputted into the TBP model (i.e., Equation (5)) to fit $c_0(t)$, $c_1(t)$, $c_2(t)$, and $c_3(t)$.

Finally, the OOA is executed to optimize the operating

variables once a change in the feedstock is detected. In the optimization algorithm, an optimized solution (hereafter called trial vector \mathbf{u}) is generated, and it is sent into the ANN model of tray temperature (i.e., Equation (2)) to obtain key tray temperatures. Simultaneously, the ICT and FCT values of each product are calculated according to tray temperatures (i.e., Equations (7) and (8)). The product yields are then obtained using Equations (10), (11), and (12); subsequently, they are input into the objective function (i.e., Equation (1)) to calculate the profit of \mathbf{u} (i.e., $f(\mathbf{u})$). After this process, the population is updated. The optimization of the operating variables will not stop until a termination condition is met.

A. Soft-sensing Strategy of the Feedstock TBP Property

The proposed soft-sensing strategy mainly includes two models: the distillate yield model and feedstock TBP property model (i.e., $yp=Y(T)$). For the distillate yield model, to use the ¹H NMR spectrum to measure the feedstock yields at different temperatures, the ¹H NMR spectrum is first pretreated for the purpose of modeling analyses, where the pretreatment techniques include phase adjustment, Fourier transformation etc. The pretreated ¹H NMR spectrum is usually composed of high-dimensional data, which causes a high computational complexity and a reduction in modeling efficiency and accuracy. Therefore, a dimensional reduction technique is necessary to reduce its dimensionality. Note that the dimensional reduction technique used in this paper is the principal component analysis provided in the MATLAB. In this paper, we use the least square support vector machine provided in the MATLAB to build the distillate yield model. Finally, $c_0(t)$, $c_1(t)$, $c_2(t)$, and $c_3(t)$ in $yp=Y(T)$ are obtained by fitting the distillate yields from the feedstock at different temperatures using the least squares method.

B. Optimization Algorithm of the Operating Variables

The OOA starts with n different individuals (called target vectors) and follows similar procedures (mutation (e.g. Equation (13)), crossover (e.g. Equation (14)), and selection (e.g. Equation (15)) shown in the DE algorithm [16].

$$\mathbf{V}_i^{g+1} = \mathbf{X}_{r_1}^g + F \cdot (\mathbf{X}_{r_2}^g - \mathbf{X}_{r_3}^g) \quad (13)$$

where $r1$, $r2$, and $r3$ are different integers randomly chosen from the range $[1, n]$, \mathbf{V}_i^g denotes the i th mutant vector of the population at generation g , F is a scaling factor, and \mathbf{X}_{r1}^g is the ($r1$)th target vectors in the population at generation g .

$$u_{i,j}^g = \begin{cases} v_{i,j}^g, & \text{if } \text{rand}(0,1) \leq CR_i \text{ or } j = j_{\text{rand}} \\ x_{i,j}^g, & \text{otherwise} \end{cases} \quad (14)$$

where CR_i is the crossover probability of the i th target vector, $v_{i,j}^g$ and $u_{i,j}^g$ are the j th operating variable in the i th mutant vector \mathbf{V}_i^g and trial vector \mathbf{u}_i^g at generation g , respectively, and j_{rand} is a random integer selected from $[1, nv]$.

$$\mathbf{X}_i^{g+1} = \begin{cases} \mathbf{u}_i^g, & \text{if } f(\mathbf{u}_i^g) \geq f(\mathbf{X}_i^g) \\ \mathbf{X}_i^g, & \text{otherwise} \end{cases} \quad (15)$$

Where $f(\mathbf{X}_i^g)$ and $f(\mathbf{u}_i^g)$ are respectively the objective values of \mathbf{X}_i^g and \mathbf{u}_i^g .

Regarding the operational optimization problem of the DU, the value of the energy recovery usually occurs at a million-fold level, causing a large feasible region of the operating variables. Therefore, the designed algorithm should have a strong exploration capability to fully explore this large feasible region. With respect to the subspace around the best individual, a small improvement in the operating variables can improve the profitability of the DU [7]. Thus, the designed algorithm should be able to exploit this subspace of the best solution. Additionally, the running status of the DU affects the economic benefits of downstream units. Adjusting the operating variables in an untimely manner may endanger the production safety of the refinery. Hence, the optimization efficiency must be considered. Motivated by the above discussion, we designed an OOA algorithm with a warm-starting strategy, adaptive mutation and crossover strategies, and an effective correction strategy to locate the best operating variables of the DU when the feedstock properties change.

1) Initialization Strategy

The change in the feedstock properties affects the product yields, which may make some solutions infeasible (i.e., violate product yield constraints described in Equation (9)). Additionally, some previous feasible solutions may perform well. Therefore, this paper re-evaluates solutions obtained from the previous feedstock properties, and all feasible solutions are reserved. Infeasible solutions are replaced by the ones that are regenerated by Equation (16).

$$x_j = lx_j + \text{rand}(0,1) \cdot (ux_j - lx_j) \quad (16)$$

2) Mutation Strategy

Studies have verified that mutant vectors generated by randomly chosen target vectors are conducive to improving the exploration capability of the DE algorithm [16]. This paper introduces the DE/rand/1 (i.e., Equation (13)) into the mutation strategy based on its strong exploration capability and simplicity. DE/pbest/1 (i.e., Equation (17)) has been proven to have good exploration and exploitation capabilities, and it is introduced into the mutation strategy.

$$\mathbf{V}_i^{g+1} = \mathbf{X}_i^g + F \cdot (\mathbf{X}_{p\text{best}}^g - \mathbf{X}_i^g) + F \cdot (\mathbf{X}_{r1}^g - \mathbf{X}_{r2}^g) \quad (17)$$

where $\mathbf{X}_{p\text{best}}^g$ is randomly selected as one of the top 100p% vectors in the current population.

In the later stage of optimization, the target vectors usually concentrate on a small search region. To improve the optimization efficiency, we propose that the neighborhood around the best target vector should be exploited in the later stage of evolution. Therefore, we introduce DE/best/1 (i.e., Equation (18)) as the operator of the mutation strategy due to

its simplicity.

$$\mathbf{V}_i^{g+1} = \mathbf{X}_{\text{best}}^g + F \cdot (\mathbf{X}_{r1}^g - \mathbf{X}_{r2}^g) \quad (18)$$

where $\mathbf{X}_{\text{best}}^g$ is the best target vectors at generation g .

To balance the search capabilities of three mutation operators, we designed the following mutation strategy.

$$\begin{aligned} \mathbf{V}_i^{g+1} &= a \times (\mathbf{X}_i^g + F_i \cdot (\mathbf{X}_{p\text{best}}^g - \mathbf{X}_i^g) + F_i \cdot (\mathbf{X}_{r1}^g - \mathbf{X}_{r2}^g)) + b \times (\mathbf{X}_{\text{best}}^g \\ &\quad + F_i \cdot (\mathbf{X}_{r1}^g - \mathbf{X}_{r2}^g)) + (1-a-b) \times (\mathbf{X}_{r1}^g + F_i \cdot (\mathbf{X}_{r1}^g - \mathbf{X}_{r3}^g)) \\ &= a \times (1-F_i) \times \mathbf{X}_i^g + a \times F_i \times \mathbf{X}_{p\text{best}}^g + b \times \mathbf{X}_{\text{best}}^g + \\ &\quad (1-a-b) \times \mathbf{X}_{r3}^g + F_i \cdot (\mathbf{X}_{r1}^g - \mathbf{X}_{r2}^g) \end{aligned} \quad (19)$$

where a and b are two different parameters. Note that $(1-a-b) \times (\mathbf{X}_{r1}^g + F_i \times (\mathbf{X}_{r2}^g - \mathbf{X}_{r3}^g))$ can be revised as $(1-a-b) \times (\mathbf{X}_{r3}^g + F_i \times (\mathbf{X}_{r1}^g - \mathbf{X}_{r2}^g))$ as $r1$, $r2$, and $r3$ are random.

The parameters a and b are important because they control the exploitation and exploration capabilities of the algorithm. We propose that the values of a and b are determined according to the survival state of each individual and the CPU time consumed for the optimization algorithm. Aging mechanism (i.e., Equation (20)) has been used in optimization algorithms to characterize the survival states of individuals [33], and we introduce this mechanism to obtain parameters a and b .

$$\text{age}_i = \begin{cases} \text{age}_i + 1, & \text{if } f(\mathbf{X}_i^g) \geq f(\mathbf{u}_i^g) \\ 0, & \text{otherwise} \end{cases} \quad (20)$$

where age_i denotes the age of the i th target vector

In the optimization process, an individual with a high age has a poor performance. At this point, parameter a should have a small value so that the mutation vector contains more information about random target vectors. Conversely, this parameter must be increased when a target vector is updated because it may find a good search direction. This paper proposes that a is updated as Equation (21).

$$a = \max(0.8 \times (1 - \frac{\text{age}_i}{10}), 0.2) \quad (21)$$

The consumed CPU time is considered in designing b to ensure that the OOA can explore more feasible region (i.e., b has a small value) in the early stage of optimization and can converge in the later stage of evolution (i.e., b has a large value). Under this condition, we propose to the use of Equation (22) to update parameter b .

$$b = 0.8 \times (1-a) \times (\frac{t}{TL}) \quad (22)$$

where TL denotes the stopping time of an algorithm in a run.

3) Parameter Adaptation Strategy

To improve the global search capability of the algorithm, we propose that F_i and CR_i are updated in each generation according to the survival state and ranking of each individual. For the top $n/2$ target vectors, considering that they may carry good search information, we propose that F_i must be relatively small to exploit the neighborhood around them. For a target vector with a poor ranking, it has a poor performance because it has survived many generations. The probability that a target vector is a near-optimal solution decreases with its age and ranking. Thus, F_i should be increased to enlarge its search region. However, too large F_i may lead to feasible target vectors moving into the infeasible region. According to the discussion above, the updating method of F_i is shown in Equation (23).

$$F_i = \begin{cases} \min(0.7 + \text{age}_i / 30, 0.9), & \text{if } \text{rank}_i < n/2 \\ \min(0.7 + \text{age}_i / 10 \times \text{rank}_i / n, 1), & \text{otherwise} \end{cases} \quad (23)$$

where rank_i is the ranking of the i th target vector.

A small crossover probability value will cause the trial vector to excessively contain information about the i th target vector. Conversely, a large value will cause the transfer of more information about a mutant vector to the corresponding trial vector. When an individual is updated and has a good ranking, especially in the later stage of evolution, this individual may search for a promising feasible region. A small crossover probability value is thus essential to ensure that more information about the current target vector is included in its trial vector. With the increase in the age and ranking of a target vector, the crossover probability value should also be increased to ensure the trial vector contains more information from the mutant vector. Based on our preliminary experiments, the crossover probability is updated as Equation (24).

$$CR_i = \min(0.8 + age_i / 10 \times rank_i / n, 1) \quad (24)$$

4) Correction Strategy

The feasible ranges of the operating variables are determined by bound constraints, the key tray temperature, and product demand constraints. In this paper, we propose an effective correction strategy in which the infeasible operating variables that violate the bound constraints are first modified, followed by the infeasible solutions that violate the key tray temperature and product demand constraints.

The best value may be in its upper bound direction when an infeasible variable is above the upper bound (i.e., ux_j). The adjustment strategy should search for its best value within $[x_{i,j}, ux_j]$. To maintain the population diversity, we propose the following correction strategy (i.e., Equation (25)) for infeasible operating variables that are above the corresponding upper bounds:

$$v_{i,j} = \begin{cases} ux_j - rand(0,1) \times (v_{i,j} - ux_j), & \text{if } v_{i,j} > ux_j \text{ and } ux_j - x_{i,j} > v_{i,j} - ux_j \\ ux_j - rand(0,1) \times (ux_j - x_{i,j}), & \text{if } v_{i,j} > ux_j \text{ and } ux_j - x_{i,j} \leq v_{i,j} - ux_j \end{cases} \quad (25)$$

Similarly, we propose the correction strategy shown in Equation (26) to repair the infeasible operating variables that are below the corresponding lower bounds:

$$v_{i,j} = \begin{cases} lx_j + rand(0,1) \times (lx_j - v_{i,j}), & \text{if } v_{i,j} < lx_j \text{ and } lx_j - v_{i,j} \leq x_{i,j} - lx_j \\ lx_j + rand(0,1) \times (x_{i,j} - lx_j), & \text{if } v_{i,j} < lx_j \text{ and } lx_j - v_{i,j} > x_{i,j} - lx_j \end{cases} \quad (26)$$

Different key tray temperature constraints and product demands cause the discontinuity of the feasible region. Additionally, the breadth of the feasible space and the rapidity of the feedstock flow rate pose challenges for rapid convergence and the diversity of solutions. Therefore, the correction strategy should have low computational complexity, and the repaired operating variables should have more randomness. In this paper, the operating variables will be regenerated by Equations (14) and (19) if they violate only the key tray temperature and product demand constraints.

In summary, the pseudo-code of the proposed OOA for the DU is provided in Algorithm 1. Note that k is a logical variable in this pseudocode.

IV. EXPERIMENTAL STUDY

In this section, we first provide the results obtained by the soft-sensing strategy of the feedstock TBP property, and then provide the results obtained by the OOA. Note that we use the crude DU depicted in Fig. 1 as a special example to examine the effectiveness of the proposed algorithm. In this unit, $d_\alpha=1$, $d_\beta=4$, $d_\gamma=13$, $d_\delta=16$, $d_\varepsilon=17$, $d_\phi=20$, $d_\zeta=25$, $d_\eta=28$, $d_\theta=29$, $d_\iota=32$, $d_\mu=37$, $d_\nu=40$, and $d_\tau=45$.

A. Experiments on Soft-sensing of the Crude TBP Property

Algorithm 1. The pseudo-code of the proposed OOA for the DU

```

1: generate  $n$  initial target vectors by the warm-starting strategy.
2: while termination criterion is not satisfied do
3:   sort target vectors according to their objective values.
4:   for  $i=1:n$  do
5:     let  $k=0$ 
6:     while  $k=0$  do
7:       let  $p=rand() \% 5$ , and select  $r1, r2$ , and  $r3$  ( $r1 \neq r2 \neq r3 \neq i$ ) randomly.
8:       calculate the parameter  $a$  and  $b$  using Equations (21) and (22), respectively.
9:       calculate  $F_i$  and  $CR_i$  according to Equations (23) and (24), respectively.
10:      generate a mutant vector using Equation (19).
11:      for  $j=1:mv$  do.
12:        if  $v_{i,j}$  is more than  $ux_j$ , then
13:          adopt Equation (25) to repair  $v_{i,j}$ 
14:        else if  $v_{i,j}$  is less than  $lx_j$ , then
15:          use Equation (26) to repair  $v_{i,j}$ 
16:        end if
17:      end for
18:      generate a trial vector ( $\mathbf{u}_i$ ) by Equation (14).
19:      calculate the key tray temperatures using Equation (2).
20:      calculate the product yields using Equations (6)-(12).
21:      if the key tray temperatures do not satisfy their constraints or the product yields do not satisfy market demands then let  $k=0$ .
22:      else  $k=1$ .
23:      end if
24:    end while
25:    evaluate the objective value of  $\mathbf{u}_i^g$ 
26:    if  $f(\mathbf{u}_i^g) > f(\mathbf{X}_i^g)$  then
27:       $\mathbf{X}_i^{g+1} = \mathbf{u}_i^g$  and  $age_i=0$ .
28:    else
29:       $\mathbf{X}_i^{g+1} = \mathbf{X}_i^g$  and  $age_i = age_i + 1$ .
30:    end if
31:  end for
32: end while

```

The accuracy of the yield of the soft-sensing model for measuring the corresponding distillate yields from the feedstock at different temperatures was first verified, followed by the accuracy verification of the least squares method for fitting the crude TBP. Generally, the feedstock temperature should not exceed 400°C because higher temperatures cause the feedstock to begin to crack, resulting in carbon deposits in the equipment and pipes. Thus, we choose 80°C, 120°C, 160°C, 240°C, 300°C, 350°C and 400°C as the seven tested temperatures.

A total of 479 groups of ¹HNMR data with different crude properties and distillate yields from crude at the seven different temperatures were acquired from the refinery. Of these, 400 groups of data were used as training data and 79 groups as test data. Two crude samples were selected to examine the performance of the feedstock TBP fitting model.

The root mean square error (RMSE) between the actual and soft measurement values was used to evaluate the performance of the distillate yield model. The RMSE is calculated as follows:

$$RMSE = \sqrt{\frac{\sum_{o=1}^{m} (y_{p_{CT_o}} - \widehat{y}_{p_{CT_o}})^2}{ny}} \quad (27)$$

where o is the tested temperature index, $\widehat{y}_{p_{CT_o}}$ is the true value of distillate yield from crude at CT_o , and ny is the number of the test samples.

For the 79 test samples, the RMSE results of distillate yields from crude at the seven different temperatures are provided in Table I. The fitting curves of the two crude TBP samples are shown in Fig. 5.

As seen in Table I, the RMSE of the distillate yield from the feedstock at each temperature is less than 0.1 (with 0.1

TABLE I
RMSE RESULTS OF DISTILLATE YIELDS FROM CRUDE AT THE SEVEN
DIFFERENT TEMPERATURES

YT	80°C	120°C	180°C	240°C	300°C	350°C	400°C
RMSE	0.072	0.030	0.022	0.018	0.014	0.013	0.014

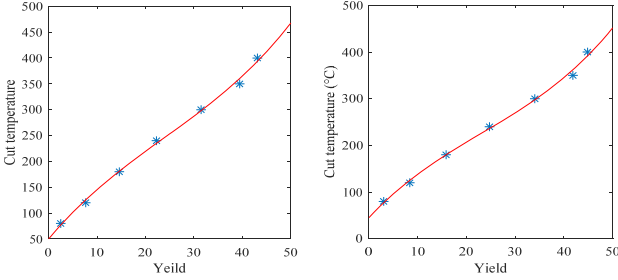


Fig. 5. The TBP fitting curves of two crude samples, x-axis is the distillate yield, and y-axis is the feedstock temperature.

being the maximum error that is tolerable for this firm). The RMSE of the distillate yield from the feedstock is maximized when the feedstock temperature is 80°C. This is because some hydrocarbons evaporated at room temperature.

B. Experiments in Optimizing Operating Variables

In this experiment, ten instances with different crude properties are selected to verify the effectiveness of the OOA. We used an elapsed CPU time limit of $TL=80s$ as the termination criterion for each algorithm in each run because, in practice, the best operating variables must be determined in two minutes. Considering that coding environment takes an effect on the computational time of an algorithm, all optimization algorithms, including those implemented in solvers (e.g., KNITRO), were coded using VC++6.0, and the simulations were carried out on a Windows 7 operating system on a computer with 16 GB of memory and a 3.40 GHz Intel® Core™ i7-2600 CPU.

In the following sections, the objective value in Equation (1) is taken as the performance metric for each algorithm, and each algorithm is executed 30 times for each instance. The means and standard deviations (denoted as the Mean±Std) of the results obtained by each algorithm are summarized in each table, and t -tests were used to judge whether the differences in the results of the compared algorithms and the OOA are statistically significant. Note that the significance level is 5%, and the results obtained by the OOA are placed in the last column of each table. “Ins” denotes the instance number. For clarity, the results of the best algorithm are marked in **boldface**. In the proposed algorithm, the main parameter to be determined is the number (n) of target vectors. According to our preliminary experiments, the OOA obtains the best performance when $n=60$.

1) Effectiveness Analysis of Different Strategies

To effectively analyze the designed mutation strategy, we consider three variants that differ from the OOA only in the mutation strategy: DEI (i.e., the mutation strategy in [23] is used to replace that in the OOA), DE1 (i.e., the DE/rank/1 replaces the mutation strategy of the OOA), and DE2 (i.e., the mutation strategy of the OOA is replaced by DE/best/1).

To examine the effectiveness of the proposed parameter adaptation strategy, the three variants of the OOA with different parameter generation strategies are used: DEQ (i.e., the scaling factor and crossover probability are generated by the strategy proposed by Qiu *et al.* [24]), DEZ (i.e., the parameter generation strategy proposed by Zhong *et al.* [25] replaces that in the OOA), DEH (i.e., the parameter adaptation strategy designed by Hui *et al.* [26] is used to

replace that in the OOA), and DE3 (i.e., the scaling factor is set to 0.8, and crossover probability is set to 0.8).

The following four variants are used to analyze the effectiveness of the correction strategy: DEP (i.e., the correction strategy in [16] is used to replace that in the OOA), DE4 (i.e., the nearest operating variable to the infeasible one replaces the latter), DED (i.e., the adjustment method proposed by Das *et al.* [33] is employed to replace that in the OOA), and DE5 (i.e., the correction strategy for the infeasible operating variables is removed).

The comparison results of OOA and its variants on the ten instances are shown in Table II. As observed, the OOA statistically outperforms all variants on both the average results and standard deviations for the ten test problems. Therefore, the proposed strategies can contribute to the OOA performance and produce promising solutions.

2) Comparison to State-of-the-Art Algorithms

To further validate the effectiveness of the OOA, we compare it with the following algorithms: JADE [27], TSDE [28], DEHL [29], and DEMS [30], NLP [31], and NLP with a warm-start strategy proposed in [32] (NLPWS), whose parameters were set to the same values as stated in their original papers. Note that the NLP algorithm designed in [31] has been implemented in a NLP solver called KNITRO. The experimental results of algorithms and the statistical analysis results obtained by t -test are summarized in Table III. It is worth noting that the experimental results obtained by NLP and NLPWS have no standard deviations and symbol “†” because they are deterministic algorithms.

From Table III, we can see that the proposed algorithm significantly outperforms JADE, TSDE, DEHL, DEMS, NLP, and NLPWS on all instances. Here, an instance is selected from the 10 examples to study the convergence performance of the OOA. The obtained convergence curve on the first instance in Fig. 6 shows that JADE has converged when the optimization time is 20 seconds, which implies a poor performance. TSDE, DEHL, and DEMS have slow convergence speeds and a poor performance. Conversely, the OOA has favorable convergence characteristics.

V. CONCLUSION

This work studied the operational optimization problem for the DU in combination with the feedstock TBP property. We proposed a soft-sensing strategy for obtaining the feedstock TBP property and an OOA to search for the best operating variables that control the operating status of the DU. The proposed soft-sensing strategy can obtain the feedstock TBP property quickly. In the OOA, we proposed a mutation strategy, a parameter adaptation generation scheme, and an effective correction strategy. We have examined the effectiveness of each strategy. The proposed algorithm can also be used to optimize the operational variables arising in other real-world systems (e.g., fluid catalytic cracking unit, hydrogenation cracking plant).

In the future, research will be conducted on the following aspects: 1) the dynamic process model of the DU will be considered based on the present research work; 2) a low dimensional model with combination of the varying feedstock properties will be established and a fast nonlinear programming will be designed to solve the DU operational optimization problem, and 3) the collaborative optimization problem among the DU, catalytic cracking unit and hydrocracking unit will be considered.

TABLE II
EXPERIMENTAL RESULTS OF OOA AND ITS VARIANTS WITH DIFFERENT STRATEGIES

Ins	DE1	DE1	DE2	DEQ	DEZ	DEH	DE3	DEP	DE4	DED	DE5	OOA
1	46190±48†	46450±6†	46067±88†	45750±54†	46327±44†	46362±28†	46470±13†	43230±101†	43153±94†	43415±121†	46334±20†	46525±7
2	11072±11†	11145±14†	11192±21†	11016±8†	11159±11†	11196±13†	11204±12†	8637±100†	8850±118†	9292±133†	11238±6†	11264±8
3	39552±18†	39789±4†	39824±3†	39754±11†	39807±2†	39811±2†	39805±3†	39175±85†	39438±59†	39211±83†	39827±2†	39832±2
4	28326±116†	28414±14†	28484±2†	28330±12†	28470±17†	28503±16†	28467±21†	25511±14†	25643±74†	25883±93†	28584±9†	28612±9
5	18327±1†	18424±2†	18432±8†	18261±32†	18448±4†	18453±4†	18452±3†	18268±30†	18286±22†	18159±30†	18470±1†	18472±0
6	27534±17†	27647±10†	27670±27†	27531±10†	27663±20†	27749±15†	27676±14†	23905±138†	24101±175†	24865±228†	27784±4†	27800±5
7	39952±1†	39891±4†	39963±2†	39814±8†	39935±5†	39953±4†	39917±4†	39427±118†	39111±215†	39339±138†	39955±2†	39963±2
8	11744±17†	11751±14†	11866±14†	11628±6†	11798±11†	11862±10†	11812±15†	10643±46†	10627±48†	10773±47†	11898±7†	11918±7
9	10187±0†	10837±2†	10763±21†	10457±35†	10848±3†	10854±2†	10851±1†	10086±109†	10271±86†	9481±144†	10836±2†	10863±0
10	46689±19†	46567±17†	46698±25†	46427±12†	46590±18†	46664±15†	46634±18†	44417±112†	44692±4†	44688±0†	46785±12†	46808±6

The symbol † in table indicates OOA performs statistically better than the compared algorithm.

TABLE III

EXPERIMENTAL RESULTS OF OOA AND STATE-OF-THE-ART ALGORITHMS							
Ins	JADE	TSDE	DEHL	DEMS	NLP	NLPWS	OOA
1	46137±30†	45211±59†	45058±44†	45883±34†	43705	43708	46525±7
2	11010±12†	10662±25†	10712±17†	10731±16†	9539	9848	11264±8
3	39798±2†	39204±28†	39708±18†	39748±4†	38286	38889	39832±2
4	28225±11†	27869±36†	27818±32†	27954±24†	26824	27031	28612±9
5	18427±3†	17870±18†	17433±60†	18361±5†	15399	17160	18472±0
6	27497±13†	27191±21†	27140±25†	27347±17†	26047	25258	27800±5
7	39892±5†	39625±12†	39765±7†	39828±11†	38438	38743	39963±2
8	11597±10†	11266±28†	11430±14†	11397±17†	10380	10544	11918±7
9	10817±12†	10213±21†	9948±56†	10808±3†	8758	9112	10863±0
10	46354±13†	46230±13†	45811±48†	46112±26†	45028	45079	46808±6

REFERENCES

- [1] L. Chen, X. Wang, and L. Tang, "Operation optimization in the hot-rolling production process," *Ind. Eng. Chem. Res.*, vol. 53, no. 28, pp. 11393-11410, July, 2014.
- [2] R. E. Young, "Petroleum refining process control and real-time optimization," *IEEE Control Syst. Mag.*, vol. 26, no. 6, pp. 73-83, Dec. 2006.
- [3] C. Pasquini and A. F. Bueno, "Characterization of petroleum using near-infrared spectroscopy: Quantitative modeling for the true boiling point curve and specific gravity," *Fuel*, vol. 86, no. 12, pp. 1927-1934, Aug. 2007.
- [4] L. C. K. Liao, T. C. K. Yang, and M. T. Tsai, "Expert system of a crude oil distillation unit for process optimization using neural networks," *Expert Syst. Appl.*, vol. 26, no. 2, pp. 247-255, Feb. 2004.
- [5] L. M. Ochoa-Estropier, M. Jobson, and R. Smith, "Operational optimization of crude oil distillation systems using artificial neural networks," *Comput. Chem. Eng.*, vol. 59, no. 5, pp. 178-185, Dec. 2013.
- [6] L. M. Ochoa-Estropier and M. Jobson, "Optimization of heat-integrated crude oil distillation systems. Part I: the distillation model," *Ind. Eng. Chem. Res.*, vol. 54, no. 18, pp. 4988-5000, Mar. 2015.
- [7] F. N. Osuolale and J. Zhang, "Thermodynamic optimization of atmospheric distillation unit," *Comput. Chem. Eng.*, vol. 103, no. 4, pp. 201-209, Aug. 2017.
- [8] V. Mahalec and Y. Sanchez, "Inferential monitoring and optimization of crude separation units via hybrid models," *Comput. Chem. Eng.*, vol. 45, no. 12, pp. 15-26, Oct. 2012.
- [9] G. Fu, Y. Sanchez, and V. Mahalec, "Hybrid model for optimization of crude oil distillation units," *Aiche J.*, vol. 64, no. 4, pp. 1065-1078, 2016.
- [10] P. Li, H. Arellano-Garcia, and G. Wozny, "Optimization of a semibatch distillation process with model validation on the industrial site," *Ind. Eng. Chem. Res.*, vol. 37, no. 4, pp. 1341-1350, Mar. 1998.
- [11] A. Raghunatha, M. Diaz, and L. Biegler, "An MPEC formulation for dynamic optimization of distillation operations," *Comput. Chem. Eng.*, vol. 28, no. 10, pp. 2037-2052, Sep. 2004.
- [12] D. C. Lopez C., L. J. Hoyos, C. A. Mahecha, H. Arellano-Garcia, and G. Wozny, "Optimization model of crude oil distillation units for optimal crude oil blending and operating conditions," *Ind. Eng. Chem. Res.*, vol. 52, no. 36, pp. 12993-13005, Aug. 2013.
- [13] M. Alhamdoosh and D. H. Wang, "Fast decorrelated neural network ensembles with random weights," *Inf. Sci.*, vol. 264, no. 6, pp. 104-117, Apr. 2014.
- [14] K. Y. Chen, M. S. Huang, and R. F. Fung, "Adaptive minimum-energy tracking control for the mechatronic elevator system," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 5, pp. 1790-1799, Sep., 2017.
- [15] P. M. Namara, R. R. Negenborn, B. D. Schutter, and G. Lightbody, "Optimal coordination of a multiple HVDC link system using centralized and distributed control," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 2, pp. 302-314, Mar., 2013.
- [16] K. V. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A practical approach to global optimization*. Secaucus, NJ, USA: Springer-Verlag, 2005.
- [17] S. Tofghi, S. A. Torabi, and S. A. Mansouri, "Humanitarian logistics network design under mixed uncertainty," *Eur. J. Oper. Res.*, vol. 250, no. 1, pp. 239-250, Apr. 2016.

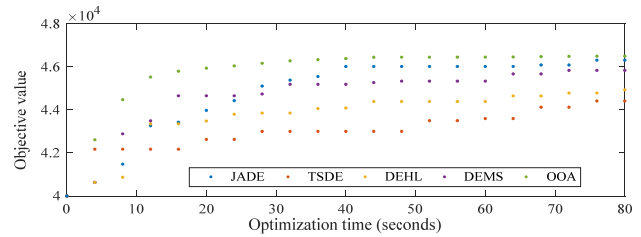


Fig. 6. Convergence curves of JADE, TSDE, DEHL, DEMS, and OOA for the first instance.

- [18] N. M. Hamza, D. L. Essam, and R. A. Sarker, "Constraint consensus mutation-based differential evolution for constrained optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 3, pp. 447-449, Jun. 2016.
- [19] W. Gong, A. Zhou, and Z. Cai, "A multioperator search strategy based cheap surrogate models for evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 746-758, Oct. 2015.
- [20] W. Gong, Y. Wang, Z. Cai, and L. Wang, "Finding multiple roots of nonlinear equation systems via a repulsion-based adaptive differential evolution," *IEEE Trans. Syst. Man Cyber.*, DOI: 10.1109/TSMC.2018.2828018.
- [21] Z. Liao, W. Gong, X. Yan, L. Wang, and C. Hu, "Solving nonlinear equations system with dynamic repulsion-based evolutionary algorithms," *IEEE Trans. Syst. Man Cyber.*, DOI: 10.1109/TSMC.2018.2852798.
- [22] S. Das, S. S. Mullick, and P. N. Suganthan, "Recent advances in differential evolution-an update survey," *Swarm Evol. Comput.*, vol. 27, pp. 1-30, Apr. 2016.
- [23] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, "Differential evolution using a neighborhood-based mutation operator," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 526-553, Jun. 2009.
- [24] X. Qiu, J. X. Xu, K. C. Tan, and H. A. Abbass, "Adaptive cross-generation differential evolution operators for multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 2, pp. 232-244, Apr. 2016.
- [25] J. H. Zhong, M. Shen, J. Zhang, H. S. H. Chung, Y. H. Shi, and Y. Li, "A differential evolution algorithm with dual populations for solving periodic railway timetable scheduling problem," *IEEE Trans. Evol. Comput.*, vol. 17, no. 4, pp. 512-527, Aug. 2013.
- [26] S. Hui and P. N. Suganthan, "Ensemble and arithmetic recombination-based speciation differential evolution for multimodal optimization," *IEEE Trans. Cyber.*, vol. 46, no. 1, pp. 64-74, Jan. 2016.
- [27] J. Zhang and A. C. Sanderson, "JADE: adaptive differential evolution with optional external archive," *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 945-958, Oct. 2009.
- [28] L. M. Zhang, S. X. Zhang, S. Y. Zheng, and Y. M. Pan, "Differential evolution algorithm with two-step subpopulation strategy and its application in microwave circuit designs," *IEEE Trans. Ind. Inf.*, vol. 12, no. 3, pp. 911-923, Jun. 2016.
- [29] Y. C. Hung, F. J. Lin, J. C. Hwang, J. K. Chang, and K. C. Ruan, "Wavelet fuzzy neural network with asymmetric membership function controller for electric power steering system via improved differential evolution," *IEEE Trans. Power Electron.*, vol. 30, no. 4, pp. 2350-2362, Apr. 2015.
- [30] X. G. Zhou, G. J. Zhang, X. H. Hao, L. Yu, and D. W. Xu, "Differential evolution with multi-stage strategies for global optimization," in *Proc. IEEE Congr. Evol. Comput.*, Vancouver, BC, Canada, Jul. 2016, pp. 24-29.
- [31] R. Byrd, M. E. Hribar, and J. Nocedal, "An interior point algorithm for the large-scale nonlinear programming," *SIAM J. Optim.*, vol. 9, no. 4, pp. 877-900, 1999.
- [32] M. Diehl, H. G. Bock, J. P. Schlöder, R. Findeisen, Z. Nagy, and F. Allgower, "Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations," *J. Pro. Control*, vol. 12, no. 4, pp. 577-585, Jun. 2002.
- [33] S. Das, A. Mandal, and R. Mukherjee, "An adaptive differential evolution algorithm for global optimization in dynamic environments," *IEEE Trans. Cyber.*, vol. 44, no. 6, pp. 966-978, Jun. 2014.