PROCESS SYSTEMS ENGINEERING
Chinese Journal of Chemical Engineering, **20**(6) 1074—1080 (2012)

Improved Hybrid Differential Evolution-Estimation of Distribution Algorithm with Feasibility Rules for NLP/MINLP Engineering Optimization Problems*

BAI Liang (摆亮)[1,2], WANG Junyan (王钧炎)[3], JIANG Yongheng (江永亨)[1,2] and HUANG Dexian (黄德先)[1,2,]**

[1] Department of Automation, Tsinghua University, Beijing 100084, China
[2] National Laboratory for Information Science and Technology, Tsinghua University, Beijing 100084, China
[3] Marvell Technology (Shanghai) Ltd, Shanghai 201203, China

Abstract In this paper, an improved hybrid differential evolution-estimation of distribution algorithm (IHDE-EDA) is proposed for nonlinear programming (NLP) and mixed integer nonlinear programming (MINLP) models in engineering optimization fields. In order to improve the global searching ability and convergence speed, IHDE-EDA takes full advantage of differential information and global statistical information extracted respectively from differential evolution algorithm and annealing mechanism-embedded estimation of distribution algorithm. Moreover, the feasibility rules are used to handle constraints, which do not require additional parameters and can guide the population to the feasible region quickly. The effectiveness of hybridization mechanism of IHDE-EDA is first discussed, and then simulation and comparison based on three benchmark problems demonstrate the efficiency, accuracy and robustness of IHDE-EDA. Finally, optimization on an industrial-size scheduling of two-pipeline crude oil blending problem shows the practical applicability of IHDE-EDA.

Keywords differential evolution, estimation of distribution, hybrid evolution, mixed-coding, feasibility rules

1 INTRODUCTION

Engineering optimization problems are often formulated as NLPs and MINLPs. Deterministic methods and solvers for them have been available since years ago [1]. However, when non-convexities exist in the models, deterministic methods cannot guarantee the global optimum. During the past decades, stochastic algorithms, due to their global search capability independent of gradient information, have been developed rapidly and applied to complex engineering optimization problems successfully [2, 3]. Differential evolution algorithm (DE), as a promising evolutionary computation technique, has drawn increasing attention [4]. In DE system, differential information is extracted from current population of solutions to guide its further search and evolves to find optimum solutions through mutation, crossover and selection operations. Due to its simple concept and quick convergence, DE has been widely applied in various fields. Different from DE, estimation of distribution algorithm (EDA) [5] does not produce offspring through evolutionary operators but directly extracts the global statistical information from the search space by building probability models to generate new population in every step. This evolutionary strategy will make EDA overcome the shortcoming of neighborhood searching based algorithms and avoid premature [6]. Thus, EDA, as an important contribution to stochastic optimization, is also widely used.

It is well known that hybridization is an effective way to improve the performance of a single algorithm. Pena *et al*. [7] proposed a hybrid evolutionary algorithm called GA-EDA. It generates offspring by the mechanism of both genetic algorithm (GA) and EDA, and thus gets benefits from both approaches. Sun *et al*. [8] introduced a combination of DE and EDA to solve unconstrained continuous optimization problems. He *et al*. [9] combines GA with DE and sequential quadratic programming (SQP) technique to improve the solving performance for an economic dispatch problem. Chang *et al*. [6] presents a hybrid algorithm called immune-estimation of distribution algorithm, in which immune algorithm help consider the relation between individuals, while EDA improves the diversity of individuals. Liao [10] presents two hybrid DEs, the test results of which on several engineering design problems show both hybrid DEs outperform the original DE, and the cooperative hybrid outperforms the local-search hybrid.

In this paper, an improved hybrid differential evolution-estimation of distribution algorithm (IHDE-EDA) is proposed to solve NLP/MINLP engineering optimization problems. IHDE-EDA incorporates annealing mechanism-embedded EDA into DE to enhance searching ability. Moreover, discrete variable handling scheme is used to make IHDE-EDA applicable to MINLPs and the feasibility rules are introduced to deal with constraints without extra parameters. Simulation and practical application case studies show the accuracy, robustness and efficiency of IHDE-EDA.

Received 2012-05-28, accepted 2012-07-20.

* Supported by the National Basic Research Program of China (2012CB720500) and the National Natural Science Foundation of China (60974008).

** To whom correspondence should be addressed. E-mail: huangdx@tsinghua.edu.cn

2 BRIEF REVIEW ON DE AND EDA

2.1 Differential evolution

DE has several versions. The outstanding one with modified mutation scheme proposed at 2006 IEEE Congress on Evolutionary Computation [11] is considered here.

Step 1 Initialize the original population of individuals in the search space randomly, which can be represented as $\boldsymbol{x}_1^0, \cdots, \boldsymbol{x}_{NP}^0$, where *NP* is the population size, and the dimension of each individual is *D*, that is, $\boldsymbol{x}_i^0 = [x_{i,1}^0, \cdots, x_{i,D}^0]$.

Step 2 In generation *k*, for the *i*th individual $\boldsymbol{x}_i^k$, execute the DE offspring generation scheme to generate an offspring $\boldsymbol{x}_i^{k+1}$ as follows.

(1) *Mutation* Choose one individual $\boldsymbol{x}_d^k$ randomly such that $f(\boldsymbol{x}_d^k) \leqslant f(\boldsymbol{x}_i^k)$, and another two individuals $\boldsymbol{x}_b^k$ and $\boldsymbol{x}_c^k$ randomly from the current population. Generate a mutated individual $\boldsymbol{v}_i^k = [v_{i,1}^k, \cdots, v_{i,D}^k]$ as follows

$$\boldsymbol{v}_i^k = (0.5+F)\cdot\boldsymbol{x}_d^k + (0.5-F)\cdot\boldsymbol{x}_i^k + F\cdot\left(\boldsymbol{x}_b^k - \boldsymbol{x}_c^k\right) \quad (1)$$

where *F* is a scaling factor, indices *b*, *c*, and *i* are different from each other.

(2) *Crossover* Generate a trial individual $\boldsymbol{u}_i^k = [u_{i,1}^k, \cdots, u_{i,D}^k]$ with the formulation:

$$u_{i,j}^k = \begin{cases} v_{i,j}^k, & \text{if rand}(j) \leqslant \text{CR} \\ x_{i,j}^k, & \text{otherwise} \end{cases} \quad (2)$$

where CR is called crossover probability, rand(*j*) is the *j*th independent random number uniformly distributed in the range of [0, 1].

(3) *Selection* $\boldsymbol{u}_i^k$ is compared to $\boldsymbol{x}_i^k$ by one to one greedy selection criterion:

$$\boldsymbol{x}_i^{k+1} = \begin{cases} \boldsymbol{u}_i^k, & \text{if } f\left(\boldsymbol{u}_i^k\right) \leqslant f\left(\boldsymbol{x}_i^k\right) \\ \boldsymbol{x}_i^k, & \text{otherwise} \end{cases} \quad (3)$$

Step 3 If $k > K^{\max}$, output the results, otherwise $k = k+1$ and go back to Step 2.

2.2 Estimation of distribution algorithm

The major issue in EDA is how to build a probability distribution model [5]. Here we use Gaussian mixture distribution and assume that *D* dimensions of individual vectors are independent. The joint density function of the *k*th generation is

$$p(\boldsymbol{x}^k) = \prod_{j=1}^{D} p\left(x_j^k\right) \quad (4)$$

where $p(x_j^k)$ is the margin density function of the *j*th dimension and formulated as

$$p\left(x_j^k\right) = \sum_{i=1}^{N} \alpha_i^k \cdot N\left(\mu_{i,j}^k, \sigma_{i,j}^k\right) \quad (5)$$

where α_i^k denotes the weight coefficient of Gaussian mixture distribution, $\boldsymbol{N}\left(\mu_{i,j}^k, \sigma_{i,j}^k\right)$ denotes that the *j*th dimension of the *i*th individual at the *k*th generation is normally distributed. $\mu_{i,j}^k$ and $\sigma_{i,j}^k$ can be estimated as follows

$$\hat{\mu}_{i,j}^k = x_{i,j}^k \quad (6)$$

$$\hat{\sigma}_{i,j}^k = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(x_{i,j}^k - \hat{\mu}_j^k\right)^2} \quad (7)$$

where $\hat{\mu}_j^k = \sum_{i=1}^{N} x_{i,j}^k \big/ N$ denotes the mean and $\hat{\sigma}_{i,j}^k$ denotes the standard deviation of the *j*th dimension of *k*th generation individuals.

3 THE APPROACH OF IHDE-EDA

3.1 Annealing mechanism for EDA

In order to avoid premature in earlier generations, the annealing mechanism is employed to determine the weight coefficient of Gaussian mixture distribution:

$$\alpha_i^k = \mathrm{e}^{-\frac{f(x_i^k)-f_{\min}^k}{t(k)}} \Big/ \sum_{i=1}^{N} \mathrm{e}^{-\frac{f(x_i^k)-f_{\min}^k}{t(k)}} \quad (8)$$

where *t*(*k*) denotes the temperature and $f_{\min}^k$ denotes the minimum objective value at the *k*th generation. The initial temperature is determined by the empirical formula:

$$t_0 = -\left(f_{\max}^0 - f_{\min}^0\right)\big/\ln(0.1) \quad (9)$$

where $f_{\max}^0$ and $f_{\min}^0$ are the maximum and minimum objective values of the individuals in the initial population, respectively. Besides, the temperature is gradually decreased during the optimization process, *i.e.* $t(k+1) = \lambda \cdot t(k)$, where the annealing rate is set as $\lambda = 0.95$.

3.2 Mechanism of hybridization

To further explore the global information of search space, motivated by [8], the mechanisms of EDA and DE are incorporated to generate new individuals.

Step 1 Instead of Eqs. (1) and (2), the trial solutions are generated as follows:

$$u_{i,j}^k = \begin{cases} (0.5+F)\cdot x_{d,j}^k + (0.5-F)\cdot x_{i,j}^k \\ \quad + F\cdot(x_{b,j}^k - x_{c,j}^k), & \text{if } \mathrm{rand}(j) \leqslant \rho \\ \text{build probabilistic model } p(\boldsymbol{x}_j^k) \text{ by} \\ \text{Eqs. (4) – (9) and sample from it,} & \text{otherwise} \end{cases} \tag{10}$$

where ρ ($0 \leqslant \rho \leqslant 1$) is a new added parameter.

Step 2 Generate new individual $\boldsymbol{x}_i^{k+1}$ following selection criterion as Eq. (3).

In the hybrid offspring generation scheme of IHDE-EDA, the trial individuals contain both differential information and global statistical information, and ρ is used to balance contributions of these two types of information.

3.3 Constraint handling

As shown in Eq. (3), there is no constraint handling mechanism in original DE system. In order to make IHDE-EDA able to handle constraints, the feasibility rules [12], which can guide the population to the feasible region quickly without adding extra parameters, are incorporated into the operation of selecting and updating individuals, that is, $\boldsymbol{x}_i^k$ will be replaced by $\boldsymbol{u}_i^k$ as follows:

(1) $\boldsymbol{x}_i^k$ is infeasible, but $\boldsymbol{u}_i^k$ is feasible;

(2) $\boldsymbol{x}_i^k$ and $\boldsymbol{u}_i^k$ are feasible, but $f(\boldsymbol{u}_i^k) \leqslant f(\boldsymbol{x}_i^k)$;

(3) $\boldsymbol{x}_i^k$ and $\boldsymbol{u}_i^k$ are infeasible, but $\mathrm{voil}(\boldsymbol{u}_i^k) \leqslant \mathrm{voil}(\boldsymbol{x}_i^k)$.

Where $f(\cdot)$ and voil($\cdot$) represent the objective value and penalty value, respectively.

3.4 Discrete variable handling

To handle discrete variables in MINLPs, IHDE-EDA uses rounding scheme [13, 14] for binary variables and random keys [15] for permutation variables. Then, discrete variables are just temporarily mapped from the real vector for evaluating the objective function, while HDE-EDA itself works with floating-point values internally.

(1) *Binary variables* When $f(\boldsymbol{x}_i^k)$ are evaluated in Eq. (3), the floating-point values $x_{i,j}^k$ $0 \leqslant x_{i,j}^k \leqslant 1$ can be mapped to binary variables $y_{i,j}^k$ by rounding scheme:

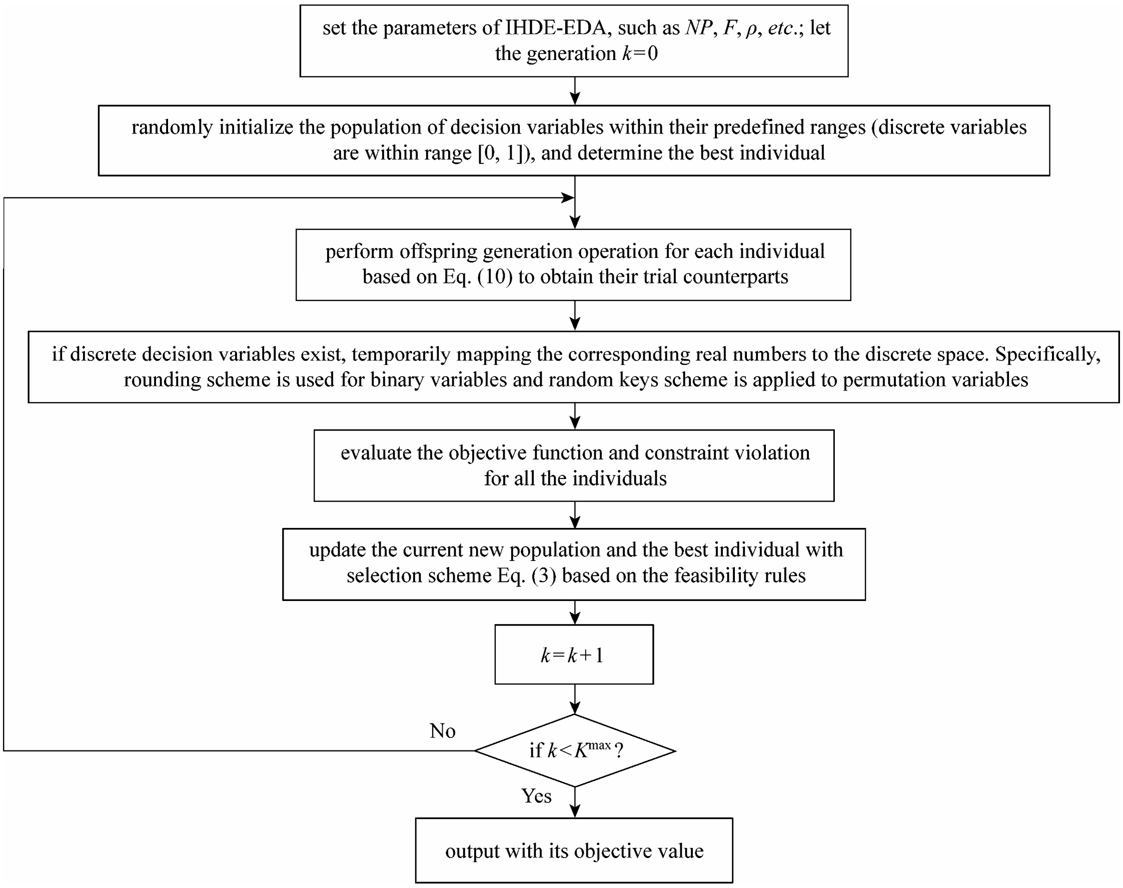

set the parameters of IHDE-EDA, such as NP, F, ρ, etc.; let the generation k=0
randomly initialize the population of decision variables within their predefined ranges (discrete variables are within range [0, 1]), and determine the best individual
perform offspring generation operation for each individual based on Eq. (10) to obtain their trial counterparts
if discrete decision variables exist, temporarily mapping the corresponding real numbers to the discrete space. Specifically, rounding scheme is used for binary variables and random keys scheme is applied to permutation variables
evaluate the objective function and constraint violation for all the individuals
update the current new population and the best individual with selection scheme Eq. (3) based on the feasibility rules
k=k+1
No
if k<K^max ?
Yes
output with its objective value

Figure 1 The Flow chart of IHDE-EDA

$$y_{i,j}^{k}=\begin{cases}0 & \text{if } x_{i,j}^{k}<0.5,\ j=1,2,\cdots,D_{\mathrm{d}}\\ 1 & \text{otherwise}\end{cases} \qquad (11)$$

where D_{d} is the dimension of discrete variables in $\boldsymbol{x}_i^k$.

(2) *Permutation variables* According to random keys [15], the floating-point vector $[x_{i,1}^k,\cdots,x_{i,D_{\mathrm{d}}}^k]$ ($0\leqslant x_{i,j}^k\leqslant 1$) in individual $\boldsymbol{x}_i^k$ can be mapped to the permutation vector $[\pi_{i,1}^k,\cdots,\pi_{i,D_{\mathrm{d}}}^k]$ by sorting the values and sequencing them in ascending order. For instance,

$$\left[x_{i,1}^k,\cdots,x_{i,4}^k\right]=[0.46, 0.91, 0.33, 0.75]\rightarrow\left[\pi_{i,1}^k,\cdots,\pi_{i,4}^k\right]$$
$$=[3\ -1\ -4\ -2]$$

3.5 Algorithm steps

With the main elements explained, the steps of IHDE-EDA are illustrated in Fig. 1.

4 EXPERIMENTAL STUDY

In this section, the effectiveness of hybridization mechanism of IHDE-EDA is first discussed. Then, three benchmark problems are used to investigate the performance of IHDE-EDA. Problems 1 and 2 are engineering design problems with NLP models and Problems 3 are chemical process design and synthesis problem with MINLP model. All the simulation is executed under the Matlab2007a environment and with Intel (R) Core(TM) 2 Quad CPU Q6600 @2.40 GHz and 1.96 GB RAM.

4.1 Discussion of hybridization mechanism

In this part, we take the welded beam design problem [16] (Problem 1) as an example to investigate the performance of algorithm with and without hybridization. As introduced in Section 3.1, a new parameter ρ ($0\leqslant\rho\leqslant 1$) is added to balance the hybridization of DE and EDA. It is obvious that IHDE-EDA reduces to EDA when $\rho=0$ and to DE when $\rho=1$. For convenience, we only compare the objective value evolving process of EDA ($\rho=0$), DE ($\rho=1$) and IHDE-EDA ($\rho=0.8$).

Figure 2 illustrates the evolution process for solving the problem by EDA, DE and IHDE-EDA. The objective function values found in IHDE-EDA decrease faster and perform better than those found in DE and EDA. Therefore, we can claim that the combination of differential information and global statistical information does indeed improve the searching performance.

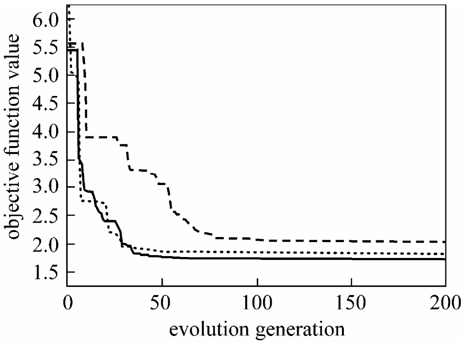


Figure 2 The evolution process of EDA, DE and IHDE-EDA
– – –EDA; ······DE; ——IHDE-EDA

4.2 Simulation results for benchmark problems

In the following, we use three NLP/MINLP benchmark problems to make comparison between IHDE-EDA and other approaches reported in literature. The main parameters of IHDE-EDA are set as follows: $NP=(7\text{–}10)\times D$, $F=0.5$ and $\rho=0.8$. Moreover, for each testing problem, we independently run IHDE-EDA 30 times.

(1) *Problem 1* (*Welded beam design problem*) This problem is taken from Rao (1996) [16]. Other approaches applied to this problem include GA-based co-evolution model (CGA) [17], modified niched-pareto genetic algorithm (MNPGA) [18], co-evolutionary particle swarm optimization (CPSO) [19], co-evolutionary differential evolution (CDE) [20], and differential evolution with level comparison (DELC) [21]. The statistical results are listed in Table 1, in which the average searching quality of IHDE-EDA is better than those of other methods. Moreover, the standard deviation by IHDE-EDA is the smallest and the number of function evaluations (NFE) equaling to a product of population size and number of iteration is the least among all the

Table 1 Statistical results of different methods for Problem 1

Methods	Best	Mean	Worst	Std. dev.	NFE
CGA	1.74831	1.77197	1.78584	1.12200×10^{-2}	9.000×10^{5}
MNPGA	1.72823	1.79265	1.99341	7.47130×10^{-2}	8.000×10^{4}
CPSO	1.72802	1.74883	1.78214	1.29260×10^{-2}	2.000×10^{5}
CDE	1.73346	1.76816	1.82411	2.21940×10^{-2}	2.048×10^{5}
DELC	1.72485	1.72485	1.72485	4.10000×10^{-13}	2.000×10^{4}
IHDE-EDA	1.72485	1.72485	1.72485	6.77522×10^{-14}	1.800×10^{4}

Table 2 Statistical results of different methods for Problem 2

Methods	Best	Mean	Worst	Std. dev.	NFE
CGA	1.27048×10^{-2}	1.27690×10^{-2}	1.28220×10^{-2}	3.93900×10^{-5}	9.000×10^{5}
MNPGA	1.26810×10^{-2}	1.27420×10^{-2}	1.29730×10^{-2}	5.90000×10^{-5}	8.000×10^{4}
CPSO	1.26747×10^{-2}	1.27300×10^{-2}	1.29240×10^{-2}	5.19850×10^{-5}	2.000×10^{5}
CDE	1.26702×10^{-2}	1.27030×10^{-2}	1.27900×10^{-2}	2.70000×10^{-5}	2.048×10^{5}
DELC	1.26652×10^{-2}	1.26653×10^{-2}	1.26656×10^{-2}	1.30000×10^{-7}	2.000×10^{4}
IHDE-EDA	1.26652×10^{-2}	1.26652×10^{-2}	1.26652×10^{-2}	1.59867×10^{-9}	1.925×10^{4}

approaches.

(2) *Problem 2* (*Tension/compression string design problem*) This problem was stated by Belegundu [22]. It has also been solved by CGA [17], MNPGA [18], CPSO [19], CDE [20] and DELC [21]. The statistical results obtained by previous approaches and IHDE-EDA are listed in Table 2, which show that IHDE-EDA finds the optimal results consistently with the best quality among all the other approaches. Moreover, the standard deviation of IHDE-EDA improves a lot and NFE is the least among all the methods considered.

(3) *Problem 3* (*Multi-product batch plant problem*) This problem was proposed by Grossmann and Sargent [23]. The global optimum given in the literature is (N_1, N_2, N_3, V_1, V_2, V_3, B_1, B_2, T_{L1}, T_{L2}; f) = (1, 1, 1, 480, 720, 960, 240, 120, 20, 16; 38499.8). Using the method proposed in this paper, we get the same optimal solution. Other approaches applied to this problem include GA [24], M-SIMPSA [24], MI-HDE [25], MDE [14], and R-PSO [26]. The statistical optimization results are listed in Table 3, in which the performance of IHDE-EDA is the best. Although there is slight improvement compared to R-PSO in terms of NFE, IHDE-EDA converges 100% for all the 30 executions.

Based on the above results, it can be concluded that IHDE-EDA is of superior searching quality and robustness for NLP/MINLP engineering optimization problems.

Table 3 Statistical results of different methods for Problem 3

Methods	NFE/NRC
GA	225176/0
M-SIMPSA	257536/97
MI-HDE	50976/100
MDE	40550/100
R-PSO	15000/90
IHDE-EDA	14500/100

Note: NRC is the abbreviation of "number of runs converged to the global optimum in the 100 executions, namely, convergence rate".

5 APPLICATION CASE

For practical application, an industrial-size problem is presented here for testing the performance of IHDE-EDA. Scheduling of two-pipeline crude oil blending (STCOB) problem is an optimization problem in chemical engineering we encountered in practical production [27], which is an extension to [28]. As shown in Fig. 3, a two-pipeline crude oil blending system is composed of several charging tanks, two transfer pipelines, one mixing pipeline, one flow control system and a crude distillation unit. Several types of crude oil, each of which has a certain amount of reserves, are stored in oil tanks. The tanks charge oils through two pipelines in a certain sequence, and then oils are

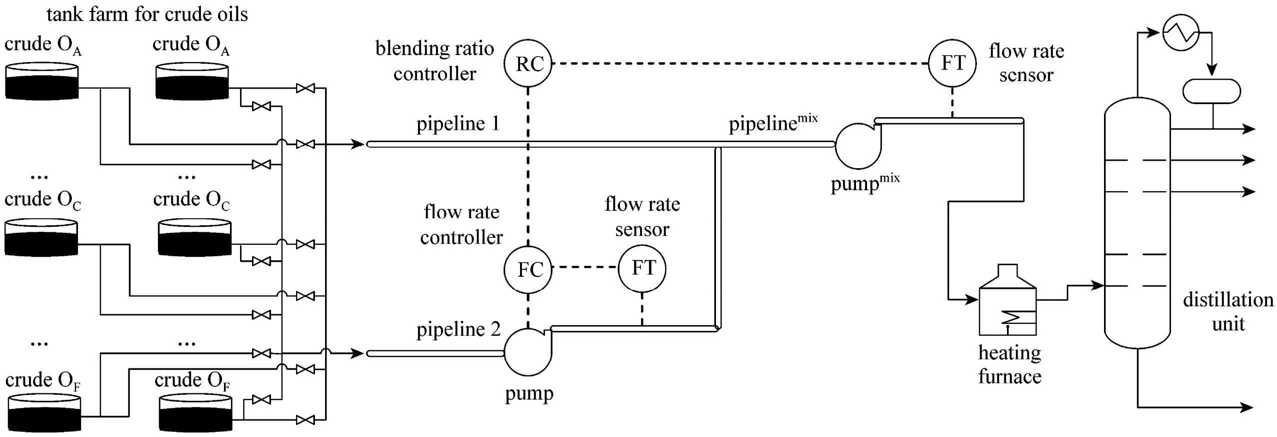


Figure 3 The flow-sheet of a practical two-pipeline crude oil blending process

Table 4 Statistical results of different methods for STCOB problem

Methods	600 s		1200 s		1800 s	
	Mean	Variance	Mean	Variance	Mean	Variance
GA	14.130	0.943	13.694	0.392	13.515	0.214
OPSO	13.507	0.480	13.230	0.244	13.021	0.073
MI-HDE	13.107	0.149	12.901	0.074	12.857	0.027
IHDE-EDA	13.069	0.167	12.855	0.037	12.828	0.019

Table 5 An optimized schedule for STOMB problem

Scheduling time slots	Start time/day	End time/day	Added tanks	Tanks for p1	Tanks for p2	Emptied tanks	Crude oil in p1	Crude oil in p2	Flow in p1 /t·h^{-1}	Flow in p2 /t·h^{-1}	Sulfur conc. ×10^{6}
1	0	1.81	**15**	1	15	1	O_A	O_F	**344**	**156**	1384
2	1.81	7.86	**2**	2	15	2	O_A	O_F	**344**	**156**	1384
3	7.86	9.60	**5**	5	15	5	O_B	O_F	**240**	**260**	1467
4	9.60	11.18	**4**	4	15	15	O_B	O_F	**240**	**260**	1467
5	11.18	12.78	**14**	4	14	14	O_B	O_F	**240**	**260**	1467
6	12.78	13.12	**12**	4	12	4	O_B	O_E	**201**	**299**	816
7	13.12	15.56	**3**	3	12	12	O_B	O_E	**201**	**299**	816
8	15.56	16.96	**13**	3	13	13	O_B	O_E	**201**	**299**	816
9	16.96	17.32	**10**	3	10	3	O_B	O_D	**177**	**323**	657
10	17.32	20.03	**9**	9	10	10	O_C	O_D	**236**	**264**	1628
11	20.03	20.85	**11**	9	11	9	O_C	O_D	**236**	**264**	1628
12	20.85	22.62	**8**	8	11	8	O_C	O_D	**236**	**264**	1628
13	22.62	24.38	**6**	6	11	6	O_C	O_D	**236**	**264**	1628
14	24.38	27.92	**7**	7	11	7	O_C	O_D	**236**	**264**	1628
15	27.92	28.12	**16**	16	11	11	O_A	O_D	**147**	**353**	710

mixed in pipeline^{mix} to charge the distillation unit. The flow control system will keep the bending flow rates (ratios) stable. Our work is to study how to arrange oil tanks to deliver oil through two pipelines in proper sequence and flow rate, so as to provide the distillation unit with qualified and stable feedstock. There are totally 20 groups of constraints and two groups of decision variables in the STCOB model, which are the charging sequence of oil tanks and the corresponding charging flow rates.

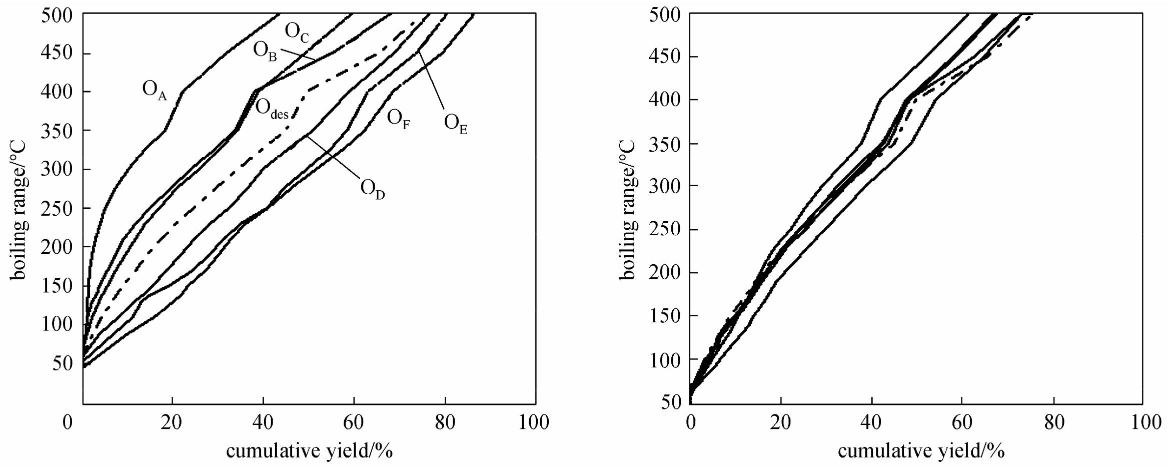
As introduced in Ref. [27], the STCOB model is a complex MINLP model with characteristics of high combination and strong nonlinearity, which is unsolvable with a gradient-based MINLP-solver directly. Here we use four kinds of stochastic optimization approaches, namely GA [24], MI-HDE [25], original PSO (OPSO) [26] and IHDE-EDA, to solve a real-case STCOB problem, where 6 kinds of crude oils ($O_A, \cdots,$ O_F) stored in $N = 15$ scheduling oil tanks are delivered through pipelines p1 and p2.

The performance with computation time of 600, 1200, and 1800 seconds is compared, and 30 independent runs are executed in each case. The statistical results are listed in Table 4, which use mean objective value and variance value to determine the optimization performance. From Table 4, we can see that whether the computation time is limited or sufficient, IHDE-EDA is overall superior to other methods.

One detailed optimized schedule by IHDE-EDA with 1800 s are shown in Table 5 (solutions are in bold type), which represents heavy and light, sour and sweet crude oils are reasonably arranged to blend, and all the specified production requirements are satisfied. Moreover, Fig. 4 illustrates the mixed crude oil in different scheduling periods all have a similar property and close to the desired feedstock after scheduling.

6 CONCLUSIONS

This paper introduces a novel improved hybrid differential evolution-estimation of distribution algorithm (IHDE-EDA). In IHDE-EDA, the annealing mechanism-embedded EDA is incorporated into DE procedure in order to improve the searching efficiency. Besides, the feasibility rules provide IHDE-EDA with an effective constraints handling way to overcome the disadvantage of penalty function methods, and discrete



(a) TBP curves of crude oils before blending (b) TBP curves of mixed feed oils after blending

Figure 4 The improvement of TBP (temperature of boiling point) curves of crude oils under the optimized schedule

variable handling scheme makes IHDE-EDA enable to deal with MINLPs. After the effectiveness of hybridization mechanism is experimentally analyzed, three benchmark NLP/MINLP problems are tested to show the superior searching performance of IHDE-EDA over other reported methods. Finally, the optimization on an industrial-size scheduling of two-pipeline crude oil blending problem demonstrates the practical applicability of IHDE-EDA.

REFERENCES

1 Grossmann, I.E., "Review of nonlinear mixed-integer and disjunctive programming techniques", *Optim. Eng.*, **3** (3), 227–252 (2002).

2 Bäck, T., Evolutionary Algorithms in Theory and Practice, Oxford University Press, New York (1996).

3 Nolle, L., Köppen, M., Schaefer, G., Abraham, A., Intelligent computational optimization in engineering: Techniques and Applications, Springer, Germany (2011).

4 Storn, R., Price, K., "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces", *J. Global Optim.*, **11** (4), 341–359 (1997).

5 Larranaga, P., Lozano, J.A., Estimation of distribution algorithms: A new tool for evolutionary computation, Kluwer Academic Publishers, Boston (2002).

6 Chang, W., Yeh, W., Huang P., "A hybrid immune-estimation distribution of algorithm for mining thyroid gland data", *Expert Syst. Appl.*, **37** (3), 2066–2071 (2010).

7 Pena, J.M., Robles, V., Larranaga, P., Herves, V., Rosales, F., Perez, M.S., "GA-EDA: Hybrid evolutionary algorithm using genetic and estimation of distribution algorithms", *Lect. Notes Comput. Sci.*, **3029**, 361–371 (2004).

8 Sun, J., Zhang, Q., Tsang, E.P.K., "DE/EDA: A new evolutionary algorithm for global optimization", *Inform. Sciences*, **169** (3-4), 249–262 (2005).

9 He, D.K., Wang, F.L., Mao, Z.Z., "A hybrid genetic algorithm approach based on differential evolution for economic dispatch with valve-point effect", *Int. J. Elec. Power*, **30** (1), 31–38 (2008).

10 Liao, T.W., "Two hybrid differential evolution algorithms for engineering design optimization", *Appl. Soft Comput.*, **10** (4), 1188–1199 (2010).

11 Price, K.V., "Differential evolution *vs.* the functions of the 2nd ICEO", In: Proc. 1997 IEEE Inter. Conf. on Evolut. Comput. (ICEC 97), USA, 153–157 (1997).

12 Deb, K., "An efficient constraint handling method for genetic algorithms", *Comput. Method Appl. M.*, **186** (2-4), 311–338 (2000)

13 Lin, Y., Hwang, K., Wang, F., "A mixed-coding scheme of evolutionary algorithms to solve mixed-integer nonlinear programming problems", *Comput. Math. Appl.*, **47** (8-9), 1295–1307 (2004).

14 Angira, R., Babu, B.V., "Optimization of process synthesis and design problems: A modified differential evolution approach", *Chem. Eng. Sci.*, **61** (14), 4707–4721 (2006).

15 Bean, J.C., "Genetic algorithms and random keys for sequencing and optimization", *Informs J. Comput.*, **6** (2), 154–160 (1994).

16 Rao, S.S., Engineering Optimization, Wiley, New York (1996).

17 Coello, C., "Use of a self-adaptive penalty approach for engineering optimization problems", *Comput. Ind.*, **41** (2), 113–127 (2000).

18 Coello, C., Montes, E.M., "Constraint-handling in genetic algorithms through the use of dominance-based tournament selection", *Adv. Eng. Inform.*, **16** (3), 193–203 (2002).

19 He, Q., Wang, L., "An effective co-evolutionary particle swarm optimization for constrained engineering design problems", *Eng. Appl. Artif. Intel.*, **20** (1), 89–99 (2007).

20 Huang, F.Z., Wang, L., He, Q., "An effective co-evolutionary differential evolution for constrained optimization", *Appl. Math. Comput.*, **186** (1), 340–356 (2007).

21 Wang, L., Li, L.P., "An effective differential evolution with level comparison for constrained engineering design", *Struct. Multidiscip. O.*, **41** (6), 947–963 (2010).

22 Belegundu, A.D., "A study of mathematical programming methods for structural optimization", Ph.D. Thesis, University of Iowa, USA (1982).

23 Grossmann, I. E., Sargent, R., "Optimum design of multipurpose chemical-plants", *Ind. Eng. Chem. Process Des. Dev.*, **18** (2), 343–348 (1979).

24 Costa, L., Oliveira, P., "Evolutionary algorithms approach to the solution of mixed integer non-linear programming problems", *Comput. Chem. Eng.*, **25** (2-3), 257–266 (2001).

25 Lin, Y., Hwang, K., Wang, F., "A mixed-coding scheme of evolutionary algorithms to solve mixed-integer nonlinear programming problems", *Comput. Math. Appl.*, **47** (8-9), 1295–1307 (2004).

26 Luo, Y.Q., Yuan, X.G., Liu, Y.J., "An improved PSO algorithm for solving non-convex NLP/MINLP problems with equality constraints", *Comput. Chem. Eng.*, **31** (3), 153–162 (2007).

27 Bai, L., Jiang, Y.H., Huang, D.X., "A novel two-level optimization framework based on constrained ordinal optimization and evolutionary algorithms for scheduling of multipipeline crude oil blending", *Ind. Eng. Chem. Res.*, **51** (26), 9078–9093 (2012).

28 Bai, L.A., Jiang, Y.H., Huang, D.X., "A novel scheduling strategy for crude oil blending", *Chin. J. Chem. Eng.*, **18** (5), 777–786 (2010).