

A Hybrid Algorithm Based on Differential Evolution and Group Search Optimization and Its Application on Ethylene Cracking Furnace^{*}

NIAN Xiaoyu (年笑宇), WANG Zhenlei (王振雷)^{**} and QIAN Feng (钱锋)

Key Laboratory of Advanced Control and Optimization for Chemical Processes (Ministry of Education) East China University of Science and Technology, Shanghai 200237, China

Abstract To find the optimal operational condition when the properties of feedstock changes in the cracking furnace online, a hybrid algorithm named differential evolution group search optimization (DEGSO) is proposed, which is based on the differential evolution (DE) and the group search optimization (GSO). The DEGSO combines the advantages of the two algorithms: the high computing speed of DE and the good performance of the GSO for preventing the best particle from converging to local optimum. A cooperative method is also proposed for switching between these two algorithms. If the fitness value of one algorithm keeps invariant in several generations and less than the preset threshold, it is considered to fall into the local optimization and the other algorithm is chosen. Experiments on benchmark functions show that the hybrid algorithm outperforms GSO in accuracy, global searching ability and efficiency. The optimization of ethylene and propylene yields is illustrated as a case by DEGSO. After optimization, the yield of ethylene and propylene is increased remarkably, which provides the proper operational condition of the ethylene cracking furnace.

Keywords group search optimization, differential evolution, ethylene and propylene yields, cracking furnace

1 INTRODUCTION

Ethylene (C₂H₄) is one of the most important and the largest volume petrochemicals in the world today. It serves as the principal building block of the petrochemical industry. The yield of ethylene and propylene is a typical index to petrochemical industry of a country [1]. Ethylene cracking furnace is the key unit of the ethylene industry and it consumes as much as 50%–60% energy of the whole process. Hence, it is very important to optimize the operation conditions of the cracking furnace to increase the profit greatly [2]. The operational conditions should be changed with the properties of feedstock so as to maximize the yields of the products. However, the optimization of the operational condition such as coil output temperature (COT) and steam hydrocarbon ratio (SHR) is challenging. During the optimization [3, 4], multiple optimal solutions can be obtained which can be classified to be global optimal solution and local optimal solutions [5]. In order to avoid falling into a local optimum, a modified evolutionary algorithm is proposed in this paper.

During the past two decades, the evolutionary computation which is accepted as a promising tool for global optimization has been extensively studied in the fields of operations research and computer science [6]. Recently, the group search optimization (GSO) algorithm proposed by He and Wu [7] was proved to be robust and effective for solving the optimization problems [8–14]. In 1997, another algorithm for optimization, the differential evolution (DE), was proposed by Storm and Price [15, 16]. This algorithm performs

great in global convergence and robustness and is therefore suitable for many kinds of numerical optimization problems [17–21]. However, both GSO and DE have shortcomings: GSO converges very slowly and DE has poor ability of local search.

Combining the good performance of DE on global search and the good ability of GSO on local search, a hybrid optimization algorithm named DEGSO is proposed in this paper. Firstly, the DE is preformed to find the local solution space. When the change of the fitness value reaches the preset threshold, the GSO is launched and the DE is stopped. Secondly, as long as a local optimum is found through GSO, the DE is restarted. Finally, the best solution will be found iteratively. Furthermore, this novel algorithm is proved to be more effective through numeral simulation than GSO. On the basis of this model, the yield of ethylene and propylene which are calculated obtained from by the software COILSIM1D, which can calculate pyrolysis products of the furnace, can be maximized while the feedstock properties changes, which is of great importance in petrochemical industry.

2 DESCRIPTION OF THE CRACKING FURNACE

The structure of typical cracking furnace is shown in Fig. 1. The feedstocks (gas or liquid) and dilution stream are preheated in the convection section to form the mixture and then gasified to the targeted temperature for cracking. The cracked gas flows to the radiant section. The cracking reaction mainly occurs

Received 2012-07-30, accepted 2012-10-29.

^{*} Supported by the Major State Basic Research Development Program of China (2012CB720500), the National Natural Science Foundation of China (U1162202), the National Natural Science Foundation of China (61174118), the National High Technology Research and Development Program of China (2012AA040307), Shanghai Key Technologies R&D program (12dz1125100) and the Shanghai Leading Academic Discipline Project (B504).

^{**} To whom correspondence should be addressed. E-mail: wangzhen_l@ecust.edu.cn

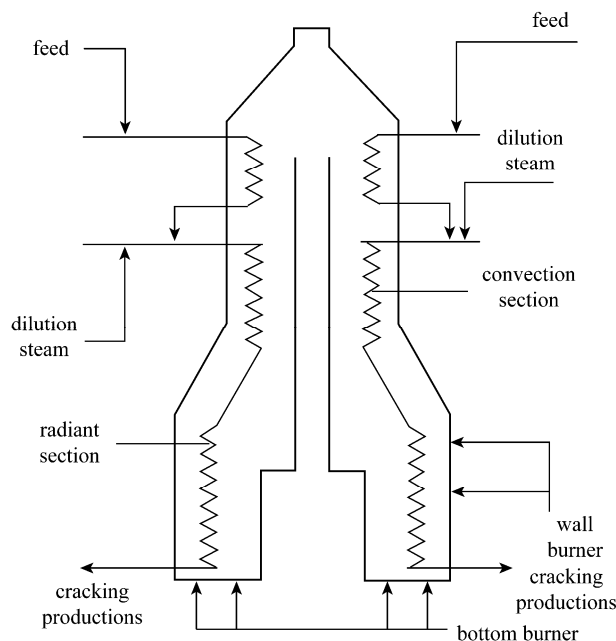


Figure 1 Structure of cracking furnace

in the radiant section at a high temperature. Meanwhile, the coil of radiant section and transfer line exchange (TLE) are coked. The mixture gas in the radiant section is cracked into ethylene, propylene, cycle ethane and other by-product such as cracked gasoline.

During the cracking process, the growth of coke on the inner wall of the coil and TLE leads to a decline of the efficiency of heat transfer, and therefore a rise of tube metal temperature (TMT) and the temperature of export of TLE. On the other hand, the growth of the thickness of coke reduces the inner diameter and increases the pressure drop, which decreases the selectivity towards cracking. Therefore, the cracking furnace should be shut down for every 50–70 d for decoking. The operational cycle of cracking furnace is defined as the period between two decoking operation. In this paper, the optimization is done in one operational cycle, and the model of the yields of ethylene and propylene is optimized by DEGSO.

3 DIFFERENTIAL EVOLUTION GROUP SEARCH OPTIMIZATION

3.1 Group search optimization

The GSO is based on the social behavior of animals hunting. There are two kinds of behavior in the hunting process, searching for food as a producer and gathering for sharing the food as scroungers. In the prey escape case, the rangers are also needed, which walk around randomly. Based on this principle, the GSO gives a producer-scrounger-ranger model. Each member in the group has the chance to be three roles and update itself in three different models.

In GSO, the swarm of members of preset number

is called a group, and each particle in the group is called a member. Each member has its own position, head angle and direction. Initially, each member's position is given randomly, and the head angle is given by an initial angle. The direction of each member can be calculated by the head angle. Each member's position is the candidate solution to the fitness function. The group is divided into three parts of members. The member which has the best value of fitness function is selected to be the producer, and the other members are randomly designated as scroungers or rangers.

The producer will scan at zero degree (laterally) by random sampling three points in the scanning field. Then the best point can be found with the best resource (fitness value). If the best point has a better resource than its current position, then it will fly to this point or it will stay in its current position and turn its head to a new angle. If the producer cannot find a better area after a iterations (a is a constant), it will turn its head back to zero degree.

During each searching bout, a number of group members are selected as scroungers. The scroungers will keep searching for opportunities to the resources found by the producer. Besides the producer and the scroungers, a small number of rangers have been also introduced into GSO. They perform search strategies, which include random walks and systematic search strategies to locate resources efficiently.

To maximize their chances for finding resources, animals use several strategies to restrict their search to a profitable patch. One important strategy is to turn back into a patch when its edge is detected. This strategy is employed by GSO to handle the bounded search space: when a member is outside the search space, it will turn back to its previous position.

3.2 Differential evolution

The DE generates new parameter vectors by adding the weighted difference vector between two population members and a third member, for each vector $\mathbf{x}_{i,G}$ ($i=1,2,\dots,N_p$), a trial vector $\mathbf{v}_{i,G+1}$ is generated according to

$$\mathbf{v}_{i,G+1} = \mathbf{x}_{r_1,G} + F \cdot (\mathbf{x}_{r_2,G} - \mathbf{x}_{r_3,G}) \quad (1)$$

where i represents the order of the population, G represents the number of iteration, N_p represents the scale of the population; r_1, r_2 and r_3 are three distinct indexes randomly selected. F is a positive real number that is usually less than 2.0.

The crossover is performed as follows:

$$\mathbf{u}_{i,G+1} = (\mathbf{u}_{1i,G+1}, \mathbf{u}_{2i,G+1}, \dots, \mathbf{u}_{Di,G+1}) \quad (2)$$

$$\mathbf{u}_{ji,G+1} =$$

$$\begin{cases} \mathbf{v}_{ji,G+1} & \text{if } (\text{randb}(j) \leq C_R) \text{ or } j = \text{rnbr}(i) \\ \mathbf{x}_{ji,G+1} & \text{if } (\text{randb}(j) > C_R) \text{ or } j \neq \text{rnbr}(i) \end{cases}$$

$$(i=1,2,\dots,N_p, j=1,3,\dots,D) \quad (3)$$

where $\mathbf{u}_{ji,G+1}$ is the vector of the crossover, D is the dimension of the problem, and C_R is a parameter ranging from 0 to 1 which controls the crossover rate of the trial individual and the old individual; $\text{randb}(j)$ is a random integer number which ensures that at least one component of the new individuals is inherited from the trial individuals and $\text{rnbr}(i)$ is a randomly chosen index, $i \in 1,2,\dots,D$.

After crossover, a greedy selection is performed as follows:

$$\mathbf{x}_{i,G+1} = \begin{cases} \mathbf{u}_{i,G+1} & \text{If } f(\mathbf{u}_{i,G+1}) < f(\mathbf{x}_{i,G}) \\ \mathbf{x}_{i,G} & \text{Otherwise} \end{cases} \quad (4)$$

3.3 DEGSO

A group of members (M) are searching in an n -dimensional space. The position of the i th member of the group at the k th iteration is $\mathbf{x}_i^k = (\mathbf{x}_{i1}^k, \dots, \mathbf{x}_{ij}^k, \dots, \mathbf{x}_{in}^k)$. In the k th iteration each member has a head angle $\varphi_i^k = (\varphi_{i1}^k, \dots, \varphi_{ij}^k, \dots, \varphi_{i(n-1)}^k)$, and its direction is given as $D_i^k = (d_{i1}^k, \dots, d_{ij}^k, \dots, d_{in}^k)$ ($1 \leq j \leq n, 1 \leq i \leq M$). When the DE is in use in the k th iteration, the member which has the best position is called a producer, and its position is characterized as $\mathbf{x}_p^k = (\mathbf{x}_{p1}^k, \dots, \mathbf{x}_{pj}^k, \dots, \mathbf{x}_{pn}^k)$.

From Fig. 2, in the $(k-s)$ th iteration, the members are assumed to execute DE. The change of F values used to characterize the change of the best values of fitness function, F , during the s iterations. If change of F converges, the DE reaches a local optimum, which indicates that a local solution is found. Then, the GSO is launched, and the best position in

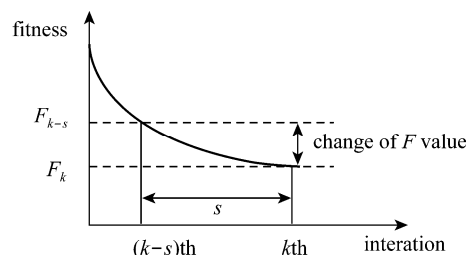


Figure 2 The schematic diagram of switch rate

DE is used as the initial producer in GSO. When change of F of GSO converges, the GSO is at a local optimum. Through this iterative hybrid method, DE and GSO can find the global optimum position. Additionally, the switch rate, R_{switch} , which is calculated by comparing F_k of the k th iteration with s iterations and that without s iterations, characterizes the switching between GSO and DE as shown in Eq. (5):

$$R_{\text{switch}} = \left| \frac{F_k - F_{k-s}}{F_{k-s}} \right| \quad (5)$$

where s is a positive integer less than k . R_{gate} is the threshold of the switch rate and assigned to be a constant in the range of 0–1. If $R_{\text{switch}} \leq R_{\text{gate}}$, the algorithm turns to the other algorithm.

The change flag (S_s) which is a boolean value determines whether the DEGSO is on the turn of DE or GSO in DEGSO, and it is defined as the following:

$$\begin{cases} S_s = 1 & \text{on the turn of GSO} \\ S_s = 0 & \text{on the turn of DE} \end{cases} \quad (6)$$

3.4 Algorithm steps

The executed steps of DEGSO are described as follows:

Step 1 Initialize the members randomly, including positions and head angles, with their fitness calculated. Then, choose the one which has the best fitness value as producer.

Step 2 Judge the value of S_s . If $S_s = 0$, go to Step 3, and if $S_s = 1$, go to Step 6.

Step 3 Use DE to update the members to search the optimum.

Step 4 Calculate the switch rate of DE (R_{switch}). If $R_{\text{switch}} \leq R_{\text{gate}}$, go to Step 5, otherwise go to Step 9.

Step 5 Set $S_s = 1$, and initialize the head angle for each member, and use the optimum in DE as the producer, then go to Step 9.

Step 6 Use GSO to update the members to search the optimum.

Step 7 Calculate R_{switch} of GSO. If $R_{\text{switch}} \leq R_{\text{gate}}$, go to Step 8, otherwise go to Step 9.

Step 8 Set $S_s = 0$, and go to Step 9.

Step 9 Calculate the fitness of each member, then choose the producer and update the member.

Step 10 If the terminal conditions are not met, go to Step 2, otherwise end the algorithm.

4 NUMERICAL EXPERIMENTS

In order to demonstrate the performance of the new hybrid algorithm, thirteen well-known test functions are used, listed in Table 1, where n is the dimension of the function, S is the allowed range of x_i , and $f_{\min}$ is the global minimum value of the function. The algorithm is compared with GSO. Table 1 shows the parameters of the thirteen test functions, i.e. the dimension n , the global minimum, and the allowed range.

For the basic GSO, the initial head angle of each member is 0.25π ; integer constant a is given as about $\sqrt{n+1}$; the maximum pursuit angle $\theta_{\max}$ is π/a^2 ; the maximum turning angle $\alpha_{\max}$ is $\pi/2a^2$; the $l_{\max}$ can be obtained according to

$$l_{\max} = \|U_i - L_i\| = \sqrt{\sum_{i=1}^n (U_i - L_i)^2} \quad (7)$$

where U_i and L_i are the upper bound and lower bound. The rate that a member is chosen to be a scrounger is 0.8. For DEGSO, the parameters of GSO are the same

as the basic GSO above. In the DE, the variation F is 0.5, the probability of crossover rate C_R is 0.1; $D_s = G_s \in [100, 200]$, where D_s and G_s represent the max number of iterations of DE and GSO, and DE and GSO will switch each other if the R_{switch} is greater than R_{gate} . $R_{\text{Dgate}} = G_{\text{Dgate}} \in [0.01, 0.1]$. The other parameters are given by the basic DE and GSO.

The number of GSO and DEGSO is 50, and the number of iteration is 1500, and the trail number is 1000 runs. Table 2 shows the performance of GSO and DEGSO algorithms. Mean represents the mean value of the minimum, Std means standard deviation, and the CPU represents the whole time (s).

From Table 2, it is shown that the average precision of these thirteen complex functions are improved by comparing GSO and DEGSO. The span of standard deviation is smaller and the speed for the optimum search is improved by a wide margin. It makes clear that the novel hybrid algorithm is more efficient to find the optimum and has an outstanding ability to reach the global optimum; the small value of Std shows that the robustness of DEGSO is strong.

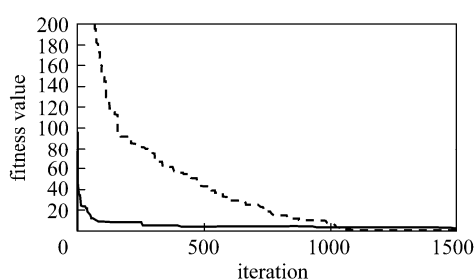
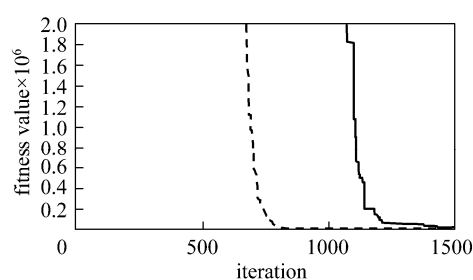
Figures 3 and 4 represent the curve for the convergence of f_5 and f_{10} , respectively. From Fig. 3, GSO

Table 1 Thirteen benchmark functions

Function	n	S	$f_{\min}$
$f_1 = \sum_{i=1}^n x_i^2$	30	$[-100, 100]^n$	0
$f_2 = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	$[-10, 10]^n$	0
$f_3 = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	30	$[-100, 100]^n$	0
$f_4 = \max_i \{ x_i , 1 \leq i \leq n \}$	30	$[-100, 100]^n$	0
$f_5 = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	$[-30, 30]^n$	0
$f_6 = \sum_{i=1}^n (\lfloor x_i + 0.5 \rfloor)^2$	30	$[-100, 100]^n$	0
$f_7 = \sum_{i=1}^n ix_i^4 + \text{random}[0,1)$	30	$[-1.28, 1.28]^n$	0
$f_8 = -\sum_{i=1}^n [x_i \sin(\sqrt{ x_i })]$	30	$[-500, 500]^n$	-12569.5
$f_9 = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]^2$	30	$[-5.12, 5.12]^n$	0
$f_{10} = -20 \exp\left(-0.2 \times \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	30	$[-32, 32]^n$	0
$f_{11} = \frac{1}{4000} \sum_{i=1}^{30} (x_i - 100)^2 - \prod_{i=1}^n \cos\left(\frac{x_i - 100}{\sqrt{i}}\right) + 1$	30	$[-600, 600]^n$	0
$f_{12}(x) = \frac{\pi}{n} \left\{ 10 \sin^2(\pi y_1) + \sum_{i=1}^{29} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2 \right\} + \sum_{i=1}^{30} u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{1}{4}(x_i + 1)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	30	$[-50, 50]^n$	0
$f_{13} = 0.1 \left\{ \sin^2(\pi 3x_1) + \sum_{i=1}^{29} (x_i - 1)^2 [1 + \sin^2(\pi 3x_{i+1})] + (x_n - 1)^2 [1 + \sin^2(\pi 3x_{30})] \right\} + \sum_{i=1}^{30} u(x_i, 5, 100, 4)$	30	$[-50, 50]^n$	0

Table 2 Results of two algorithms on all benchmarks

Function	Algorithm	Mean	Std	CPU/s
f_1	GSO	1.9481×10^{-8}	1.1629×10^{-8}	27.6
	DEGSO	5.4492×10^{-19}	2.3814×10^{-19}	21.5
f_2	GSO	3.7039×10^{-5}	8.6185×10^{-5}	29
	DEGSO	8.8766×10^{-12}	7.5485×10^{-12}	25.1
f_3	GSO	5.7829	3.6813	24.5
	DEGSO	5.206×10^{-2}	1.196×10^{-2}	19
f_4	GSO	0.1078	3.9981×10^{-2}	27.4
	DEGSO	2.8574×10^{-2}	9.1240×10^{-3}	21.2
f_5	GSO	49.8359	30.1436	27.8
	DEGSO	0.5473	1.0573	20.2
f_6	GSO	1.6000×10^{-2}	0.1333	27.1
	DEGSO	4.4543×10^{-5}	8.2814×10^{-4}	21.7
f_7	GSO	7.3773×10^{-2}	9.2557×10^{-2}	31.4
	DEGSO	2.2730×10^{-2}	0.1783	26.7
f_8	GSO	-12569.4882	2.2140×10^{-2}	32.7
	DEGSO	-12569.4713	4.6543×10^{-2}	27.5
f_9	GSO	1.0179	0.9509	50.1
	DEGSO	6.6673×10^{-7}	5.7191×10^{-7}	44.4
f_{10}	GSO	2.6548×10^{-5}	3.0820×10^{-5}	34.4
	DEGSO	1.8083×10^{-5}	4.4536×10^{-4}	30.1
f_{11}	GSO	3.0792×10^{-2}	3.0867×10^{-2}	35.9
	DEGSO	1.6653×10^{-15}	1.3486×10^{-15}	30.4
f_{12}	GSO	2.7648×10^{-11}	9.1674×10^{-11}	34.4
	DEGSO	3.9293×10^{-12}	7.6406×10^{-11}	29.8
f_{13}	GSO	4.6948×10^{-5}	7.001×10^{-4}	35.6
	DEGSO	5.9790×10^{-7}	1.9728×10^{-6}	29.4

Figure 3 The performance of two algorithms in f_5
— GSO; - - - DEGSOFigure 4 The performance of two algorithms in f_{10}
— GSO; - - - DEGSO

converged to better solutions and faster than DEGSO at the beginning. However, DEGSO got a higher accuracy and closer to the global optimum, while GSO got trapped in a local optimum solution. From Fig. 4, DEGSO converged to better solutions quickly and it had higher accuracy than GSO. In a word, DEGSO performs higher accuracy in finding the global optimum than GSO, which reaches our target of improvement.

5 APPLICATION

The yield of ethylene and propylene is an important criterion to judge the efficiency of an ethylene cracking furnace. Maintaining the operating cycle of the cracker, operators always want to obtain the maximum yield of ethylene and propylene. Several factors

that affect the yield are detected: properties of feedstock, coil output temperature (COT), coil output pressure (COP), steam hydrocarbon ratio (SHR), feedstock flow rate and crossover temperature. In our case, COP is in the position of high temperature and high pressure which makes it difficult to get the accurate value. Therefore, COP is not used in the model of yield. In addition, the crossover temperature has very little effect on the yield. The feedstock flow rate is assigned a constant, and generally unchanged. The properties of feedstock are also not tunable. Therefore, only COT and SHR can be controlled freely.

The volume yields of some gaseous phase in cracking gas can be measured by online chromatographic analyzer according to the sample condition. Hence, it is impossible to represent mass yield of the whole components for cracking gas. Considering the processing situation, we build the model of yield based on COILSIM1D [22, 23] in this paper. COILSIM1D is a kind of cracking reaction software based on the free radical reaction. The error between COILSIM1D and cracking furnace output can be corrected by the mass ratio of propylene over ethylene, which should be same for the simulated values and measured ones. The mass yield of the whole components can be obtained from the software after correction. Therefore, the input variables of model are the mole percentage of *n*-alkanes, *iso*-alkane, cycloalkane and aromatics in the feedstock, feed flow rate, COT and SHR, and the output variable is the yield of ethylene and propylene. The target is to maximize the yield of ethylene and propylene by tuning COT and SHR as soon as the properties of feedstock changes.

The operating conditions are optimized separately by DEGSO for three kinds of feedstock. The properties of feedstock are shown in Table 3. The results for the optimization of each feedstock are shown in Table 4, as well as the actual operating condition of COT and SHR. The quantity of feedstock flow is $4 \times 10^4 \text{ kg} \cdot \text{h}^{-1}$ which is fixed. The optimization curves of feature of each feedstock are shown in Figs. 5–7.

Table 3 Properties of three kind of feedstocks

Feed type	<i>n</i> -Alkanes/%	<i>iso</i> -Alkane/%	Cycloalkane/%	Aromatic/%
A	40.49	35.33	14.60	9.57
B	38.79	36.48	16.91	7.81
C	32.13	30.79	22.61	8.98

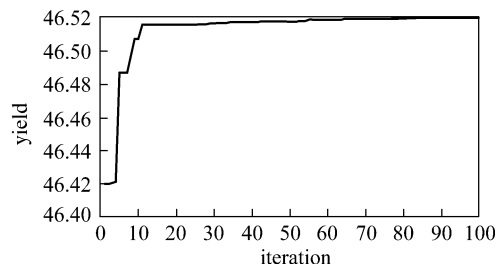


Figure 5 Optimization curve of ethylene and propylene yield for feedstock A

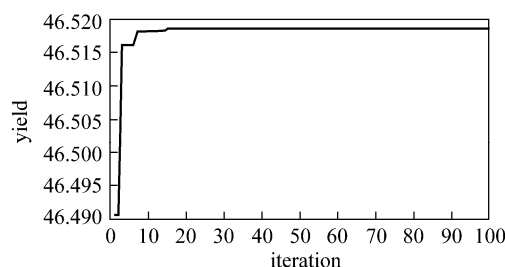


Figure 6 Optimization curve of ethylene and propylene yield for feedstock B

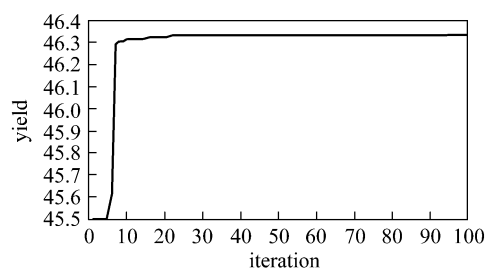


Figure 7 Optimization curve of ethylene and propylene yield for feedstock C

From Table 4 and Figs. 5–7, it is found that the optimal COT and SHR can be found by DEGSO. The yield of ethylene and propylene is greatly improved and the yield curve converges quickly. Therefore, it is necessary to optimize the operational condition when the properties of feedstock change.

6 CONCLUSIONS

Ethylene cracking unit has to be suited for the change of properties of feedstocks. In this paper, a hybrid algorithm DEGSO is proposed for the optimization

Table 4 Comparison of optimization result and actual result

Feed type	COT/°C		SHR		Yield/% (by mass)	
	Actual	Optimized	Actual	Optimized	Actual	Optimized
A	837.2023	839.9147	0.6	0.6166	45.8953	46.5199
B	837.8158	837.4545	0.5852	0.6315	45.8396	46.5185
C	838.1217	835.6909	0.6057	0.6558	45.5746	46.3312

of the operational conditions for maximization the yield of ethylene and propylene. DEGSO combines the advantages of DE and GSO by using them in turn according to the switch rate so as to prevent the solution from being converged prematurely. DE is used for global search while GSO is used for local search. Benchmark functions are employed to prove that DEGSO has a superior searching performance for high accuracy, global searching ability and high efficiency by comparing with GSO. When applied to an ethylene cracking furnace, DEGSO can find the optimal operating conditions when the properties of feedstock change, which increases the yield of ethylene and propylene remarkably.

REFERENCES

- Liu, S.T., Wang, H.G., Qian, F., Hu, G.H., "Coupled simulation of combustion with heat transfer and cracking reaction in SL-II industrial ethylene pyrolyzer", *Journal of Chemical Industry and Engineering (China)*, **62** (5), 1309–1317 (2011). (in Chinese)
- Li, P., Li, Q.A., Lei, R.X., Chen, A.J., Ren, L.L., Cao, W., "Development and application of advanced process control system for ethylene cracking heaters", *Journal of Chemical Industry and Engineering (China)*, **62** (8), 2217–2220 (2011). (in Chinese)
- Wu, Y.L., Lu, J.G., Sun, Y.X., "An improved differential evolution for optimization of chemical process", *Chin. J. Chem. Eng.*, **16** (2), 228–234 (2008).
- Qian, Y., Pan, M., Huang, Y.C., "Modeling and optimization for scheduling of chemical batch processes", *Chin. J. Chem. Eng.*, **17** (1), 1–7 (2009).
- Wu, Y.L., Lu, J.G., Sun, Y.X., "An improved differential evolution for optimization of chemical process", *Chin. J. Chem. Eng.*, **16** (2), 228–234 (2008).
- Huang, F.Z., Wang, L., "A hybrid differential evolution with double population for constrained optimization", In: IEEE Congress on Evolutionary Computation, Hong Kong, 18–25 (2008).
- He, S., Wu, Q.H., Saunders, J.R., "A novel group search optimizer inspired by animal behavioural ecology", In: IEEE Congress on Evolutionary Computation, Vancouver 1272–1278 (2006).
- He, S., Wu, Q.H., Saunders, J.R., "Group search optimizer: An optimization algorithm inspired by animal searching behavior", In: IEEE Transactions on Evolutionary Computation, **13** (5), 973–990 (2009).
- He, S., Wu, Q.H., Saunders, J.R., "A group search optimizer for neural network training", *Computer Science*, **3982**, 934–943 (2006).
- He, S., "Training artificial neural networks using Lévy group search optimizer", *J. Multiple-Valued Logic Soft Comput.*, **16**, 527–545 (2010).
- He, S., Li, X., "Application of a group search optimization based artificial neural network to machine condition monitoring", In: ETFA 2008 IEEE International Conference, 1260–1266 (2008).
- Shen, H., Zhu, Y., Niu, B., Wu, Q.H., "An improved group search optimizer for mechanical design optimization problems", *Prog. Natur.*, **19**, 91–97 (2009).
- Wu, Q.H., Lu, Z., Li, M.S., Ji, T.Y., "Optimal placement of FACTS devices by a group search optimizer with multiple producer", In: CEC 2008 IEEE World Congress on Computational Intelligence, Hong Kong, 1033–1039 (2008).
- Qin, G., Liu, F., Li, L., "A quick group search optimizer with passive congregation and its convergence analysis", In: Computational Intelligence and Security, Beijing, 249–253 (2009).
- Storm, R., Price, K., "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces", *J. Glob. Optim.*, **11** (4), 341–359 (1997).
- Storm, R., Price, K., "Differential evolution for multi-objective optimization", *Evol. Comput.*, **4**, 8–12 (2003).
- Tasgetiren, M.F., Suganthan, P.N., Chua, T.J., Al-Hajri, A., "Differential evolution algorithms for the generalized assignment problem", In: IEEE Congress on Evolutionary Computation, Trondheim, 2606–2613 (2009).
- Aslantas, V., Tunckanat, M., "Differential evolution algorithm for segmentation of wound images", In: IEEE International Symposium on Intelligent Signal Processing, Alcalá de Henares, 1–5 (2007).
- Sum-Ia, T., Taylor, G.A., Irving, M.R., Song, Y.H., "A differential evolution algorithm for multistage transmission expansion planning", In: Universities Power Engineering Conference 2007, Brighton, 357–364 (2007).
- Zhang, W.G., Chen, H.P., Lu, D., Hao, S., "A novel differential evolution algorithm for a single batch-processing machine with non-identical job size", In: The Fourth International Conference Natural Computation, Jinan, 447–451 (2008).
- Zhu, X.B., Yu, Q., Wang, X.J., "A hybrid differential evolution algorithm for solving nonlinear bilevel programming with linear constraints", In: Proc. 5th IEEE Int. Conf. on Cognitive Informatics (ICCI'06), Beijing, 126–131 (2006).
- Van Geem, K.M., Hudebine, D., Reyniers, M.F., Wahl, F., Verstraete, J.J., Marin, G.B., "Molecular reconstruction of naphtha steam cracking feedstocks based on commercial indices", *Computers & Chemical Engineering*, **31** (9), 1020–1034 (2007).
- Pyl, S.P., Van Geem, K.M., Reyniers, M.F., Guy, B.M., "Molecular reconstruction of complex hydrocarbon mixtures: An application of principal component analysis", *AIChE Journal*, **56** (12), 3174–3188 (2010).