Chemical Engineering Journal 150 (2009) 527–535

Contents lists available at ScienceDirect

Chemical Engineering Journal

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

A DNA based genetic algorithm for parameter estimation in the hydrogenation reaction

Xiao Chen, Ning Wang*

National Laboratory of Industrial Control Technology, Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, PR China

ARTICLE INFO

Article history:
Received 1 December 2008
Received in revised form 11 March 2009
Accepted 11 March 2009

Keywords:
Non-linear optimization problems
DNA computing
Genetic algorithm
Hydrogenation
Kinetic model
Parameter estimation

ABSTRACT

Inspired by the mechanism of the biological DNA, a DNA based genetic algorithm (DNA-GA) is proposed to determine the kinetic parameters for the hydrogenation reaction. The considered chemical process contains five reactions and 25 unknown parameters. The DNA-GA uses the DNA encoding method to represent the potential parameters and genetic operators inspired from the biological DNA are designed to find the global optimum. The study on the performance for typical benchmark functions shows that the DNA-GA outperforms the other two methods in both convergence speed and accuracy. Based on the operating data gathered from an industrial hydrogenation unit, 25 parameters are obtained by the DNA-GA and the kinetic model for the hydrogenation reaction is established. To verify the validity of the established model, another four groups of data are used to test the established model and two previously reported models. The comparison results show that the sum of square relative errors of the model obtained by the DNA-GA is the least of the test models, and its prediction is in good agreement with the practical operating data.

1. Introduction

As one of the heavy feedstock refinery technology, hydrogenation reaction has become popular in the recent years. Its main function is to produce the feedstock meeting the requirement of the cracking reactions through removing the impurities from the residue oil. For the optimization purpose, engineers use kinetic models to describe the hydrogenation process. However, mathematical modeling of such complex chemical kinetics usually leads to non-linear parameter estimation problems, where many deterministic optimization methods suffer from getting trapped in local minima since such parameter estimation problems often contain more than one minimum among which one is the global minimum and the others are local optima [1,2].

In this context, genetic algorithm (GA) developed by Holland is employed in the parameter estimation problems for its global searching ability [3,4], and the application of GA in various chemical engineering disciplines has increased [5–7]. Although GA performs well in many problems, it has some limitations, such as weak local-search capability and premature convergence [8]. To improve the performance of GA, hybrid methods are studied in the recent years, but most of them are implemented by predicting the good initial points using GA and determining the final optimum with a local-search method [9–11]. Since their effectiveness relies on the global exploration capability of GA, they would be constrained if GA fails to predict the potential initial condition of the global optimum. Moreover, the traditional GA uses binary encoding or real encoding, and neither of them could represent the diverse genetic information and better imitate the regulation action of genes to the genetic processes. As such, some biological operations at the gene level cannot be effectively adopted in the existing GA [12].

To overcome the drawbacks of GA, a few of the improved GAs based on the mechanism of the biological DNA have been developed [12–14]. Tao et al. proposed a RNA genetic algorithm based on DNA computing to estimate the parameters of chemical engineering processes [15]. They encoded the chromosomes with nucleotide bases and modified GA operators with RNA molecular operations. This method can largely improve the diversity of the population and the ability to overcome the fraudulence compared with standard GA (SGA). However, this method sacrifices the rapidity of convergence to obtain the diversity of the population and it tends to fail in the high-dimensional optimization problems.

To overcome the above deficiencies, a DNA based GA (DNA-GA) is proposed. In this algorithm, we encode each individual with a sequence of nucleotide bases. Then, inspired by the operations of DNA molecular, we design genetic operators to enhance the global searching ability of the DNA-GA. Simulation studies on six benchmark functions, varying from two-dimensional to high dimensional, show the superiority of the DNA-GA in contrast to other two algorithms, RNA-GA and GA. Finally, the parameters in the hydrogenation kinetic models proposed by Xu et al. [16,17] are estimated by the DNA-GA.

* Corresponding author.
E-mail address: nwang@iipc.zju.edu.cn (N. Wang).

1385-8947/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.cej.2009.03.016

Nomenclature

C_{in}	impurity content of the feedback oil
C_{out}	impurity content of the exported product
E	activation energy
F_{ave}	average value of the optimization problem
F_{max}	maximum of the optimization problem
F_{min}	minimum of the optimization problem
G_{ave}	average evolution number of the optimization problem
G_i	the evaluation generation number in i_{th} run
G_{max}	maximum evolution number of the optimization problem
G_{min}	minimum evolution number of the optimization problem
k_0	pre-exponential parameter
l	each parameter encoding length
L	individual length
N	population size
ns	number of the sample data
P_c	crossover probability
P_{IA}	IA operator probability
P_m	NM operator probability
P_{mm}	MM operator probability
R	molar gas constant 8.314 J/(mol K)
t_2	retention time of the catalyst
t_1	running time of the unit
T_1	catalyst functional activated time
T_2	reaction temperature
y_p	real removal fraction of the P_{th} group
$\hat{y}_p$	model predicted removal fraction of the P_{th} group

Greek symbols

β	catalyst deactivation parameter
μ	reaction order

2. The DNA genetic algorithm

2.1. DNA encoding method

In the view of modern biology, DNA is the major genetic material for life and encodes plentiful genetic information. Similarly, as a string of binary data is encoded with 0 and 1, DNA is encoded with nucleotides. Due to their different chemical structure, nucleotides can be classified into four types of bases: adenine (A), guanine (G), cytosine (C), and thymine (T). According to Watson–Crick complementary principle, bonding occurs by the pairwise attraction of bases: A bonds with T and G bonds with C. For example, if sequence S is TTCGC, its complement S' is AAGCG. The pairs (A, T) and (G, C) are therefore known as complementary base pairs. Through this complementary property, a codon specified by a triplet of bases can be bonded with the specific anticodon consisting of the complementary bases on transfer RNA (tRNA) (where T is replaced by U in tRNA), and assists subsequent transmission of genetic information in the formation of a specific amino acid, as shown in Fig. 1. [18].

Based on the biological DNA structures, we can design a DNA encoding method for practical problems. Generally, an n-objective optimization problem can be written as follows:

$$\begin{cases} \min f(x_1, x_2, \cdots x_n) \\ x_{\min i} \le x_i \le x_{\max i}, i = 1, \cdots, n \end{cases} \quad (1)$$

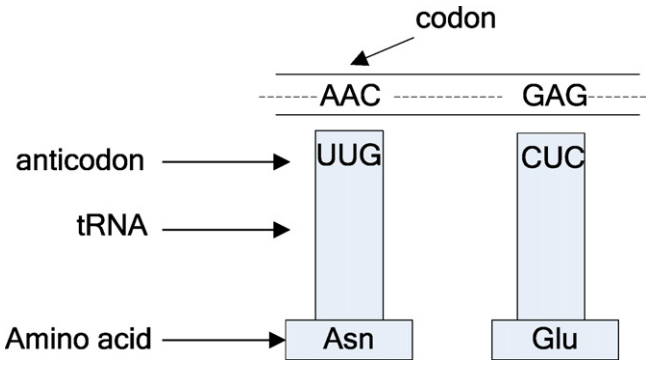

Fig. 1. Genetic information transmission.

where x is the vector of n decision or control variables, $f(x)$ is the objective function and $[x_{mini}, x_{maxi}]$ is the parameter bounds.

In this work, every variable x_i of the problem (1) is represented as a string consisting of a combination of four nucleotide bases, A, G, C, T. This means we have a four-letter alphabet $\sum\{A, G, C, T\}$ to encode potential solutions. Since such string cannot be possessed by digital computers, these bases are encoded with digital numbers. Here, integer 0, 1, 2, and 3 are adopted to encode the bases since they could represent the characteristics of bases, such as structure, function group and complementary relationship [19]. And the mapping from nucleotide bases to the digital integers is 0123/CGAT, which means that C accords with 0, G with 1, A with 2, and T with 3. Through inheriting the complementary properties of the nucleotide bases, the four integers pair as (2, 3) and (0, 1).

Then, every variable x_i is represented as an integer string of length l. The lower limit x_{mini} is represented by the decoded integer 0, and the upper limit x_{maxi} is represented by the decoded integer $4^l - 1$. And the precision of parameter x_i is $(x_{maxi} - x_{mini})/4^l$. The length of one individual is $L = n \times l$. Based on this DNA encoding method, we can introduce features of the biological DNA into the traditional GAs and develop a new DNA-GA.

2.2. Genetic operators

The correct choice of genetic operators is very important for the application of GAs. Hence, based on the above encoding method, we develop the genetic operators in the DNA-GA to enhance the searching ability of GA. They are crossover operator, selection operator and three mutation operator consisting of inverse-anticodon operator, maximum-minimum operator, and normal-mutation operator.

2.2.1. Crossover operator

Crossover is an operator that exchanges information between different chromosomes, which is important for the entire search process. In crossover operation, a pair of parents can be obtained by randomly choosing two individuals from the population. Then, every parent is separated into n parts, and two parts in the same location of two parents are mated as a pair of sub-parents. Such separation is convenient for the problems where each variable is encoded with different length. Afterward, two-point crossover is adopted over each pair of sub-parents and the crossover points are set at random. The crossover is adopted with probability P_c. An example is shown in Fig. 2.

2.2.2. Mutation operators

2.2.2.1. Inverse-anticodon operator (IA operator). IA operator is an operator which replaces the codon with its inverse anticodon. First, the individual is also separated into n parts and some consecutive bases are chosen as a codon in each part. Different from the codon in biological DNA, the number of the bases in a codon of

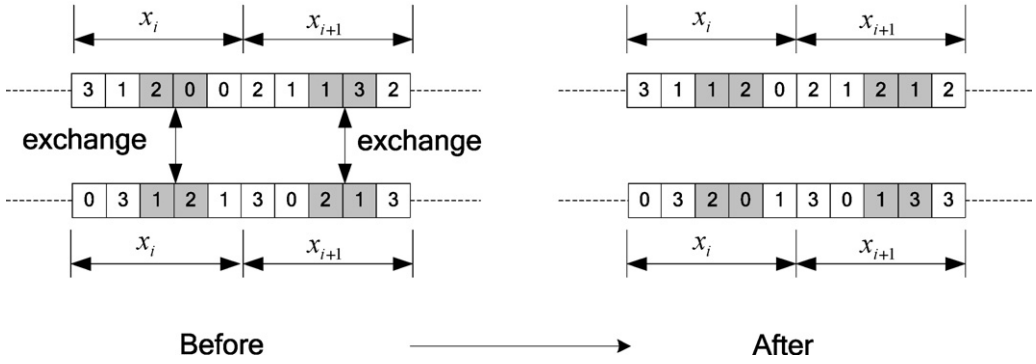

Fig. 2. An example of crossover operator.

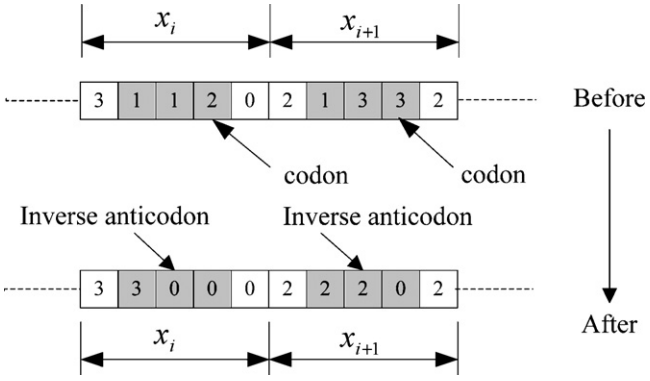

Fig. 3. An example of IA operator.

the DNA-GA is not fixed. The number of the bases and the location of codon are both assigned randomly. Then, the anticodon that consists of the complementary bases is obtained based on the Watson–Crick complementary principle. Afterwards, the order of the bases in the anticodon is inverted. Finally, the codon is replaced by its inverse anticodon. For example, in Fig. 3, if the codon is selected as 112(GGA), its anticodon will be 003(CCT) and its inverse anticodon is 300(TCC). Then, the bases 112 are replaced with 300. IA operator occurs with the probability P_{IA}.

2.2.2.2. Maximum-minimum operator (MM operator). MM operator can change the chromosome by replacing the frequently used bases with the rarely used bases in the current chromosome. Notice that, MM operator is different from IA operator that it does not require the individual to be separated into several parts. One example of MM operator is shown in Fig. 4. In Fig. 4, it is clear that base G (1) is the most frequently used base in the chromosome, while base C (0) is the least frequently used base. Then, MM operator replaces every base G with base C. MM operator occurs with the probabilityP_{MM}.

2.2.2.3. Normal-mutation operator (NM operator). Normal-mutation operator is a background operator that produces spontaneous random changes in the chromosomes. In the DNA-GA, every base in the individual can be replaced by one of another three bases with the probability P_m. Fig. 5 gives an example of NM operator where base A is replaced with base G.

2.2.3. Selection operator

Selecting individuals for the next generation is also an important process. In this paper, tournament selection is used, in which two individuals are compared against each other and the one with higher fitness value advances into the next generation. Note that elitism is also used in conjunction with tournament selection in attempt to guarantee the best individual is replicated into the next generation.

2.3. Procedure of the DNA-GA

Based on the above encoding method and the genetic operators, the procedure of the DNA-GA can be summarized as follows, seen in Fig. 6.

Step 1: Initialize a population containing *N* individuals.
Step 2: Calculate the fitness value of each individual.
Step 3: Select two individuals from the population randomly as the parents and adopt crossover operator over the parents to generate new individuals. Repeat this step until *N*/2 new individuals are created.
Step 4: Insert all the new individuals generated in step 3 into the population without deleting old individuals.
Step 5: Adopt three mutation operators orderly over each individual, and generate 3/2*N* new individuals.
Step 6: Replace all the original individuals with the new ones produced in step 5.
Step7: Apply elitism in conjunction with tournament selection to choose *N* individuals from the population for advancing into the next generation.
Step 8: Repeating steps 2–7 until the stop criteria are met, and the final solution is found.

3. Computational experiments

Generally, the performance of the improved GA should be investigated through computational experiments. Hence, we select six

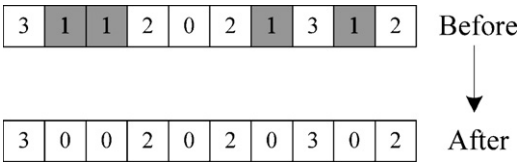

Fig. 4. An example of MM operator.

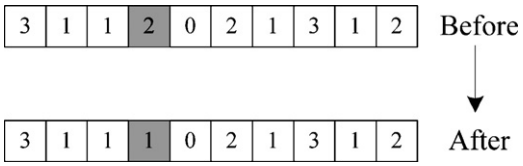

Fig. 5. An example of NM operator.

Table 1
Test functions.

Test functions	Optimal solution	Optimal value
$\min f(x) = 1 + \left((x_1 - 100)^2 + (x_2 - 100)^2\right)/4000 - \cos(x_1 - 100)\cos\left((x_2 - 100)/\sqrt{2}\right),\ x_i \in [-600, 600]$	(100, 100)	0
$\min f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2,\ x_i \in [-5.12, 5.12]$	(1, 1)	0
$\max f(x) = \left(\frac{a}{b+(x_1^2+x_2^2)}\right)^2 + (x_1^2 + x_2^2)^2, a = 3, b = 0.05,\ x_i \in [-5.12, 5.12]$	(0,0)	3600
$\min f(x) = 0.5 + \frac{\left(\sin\sqrt{x_1^2+x_2^2}\right)^2 - 0.5}{\left[1+a(x_1^2+x_2^2)^2\right]^2},\quad x_i \in [-10, 10]$	(0,0)	0
$\min f(x) = -c_1 \exp\left(-c_2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\cos\left(\sum_{i=1}^{D} c_3 x_i\right)\right) + c_1 + \exp(1),$ $x_i \in [-32.768, 32.768], i = 1 : D, D = 10, c_1 = 20, c_2 = 0.2, c_3 = 2\pi$	(0,0...0)	0
$\min f(x) = \sum_{i=1}^{n} x_i^2, n = 10, x_i \in [-5.12, 5.12]$	(0,0...0)	0

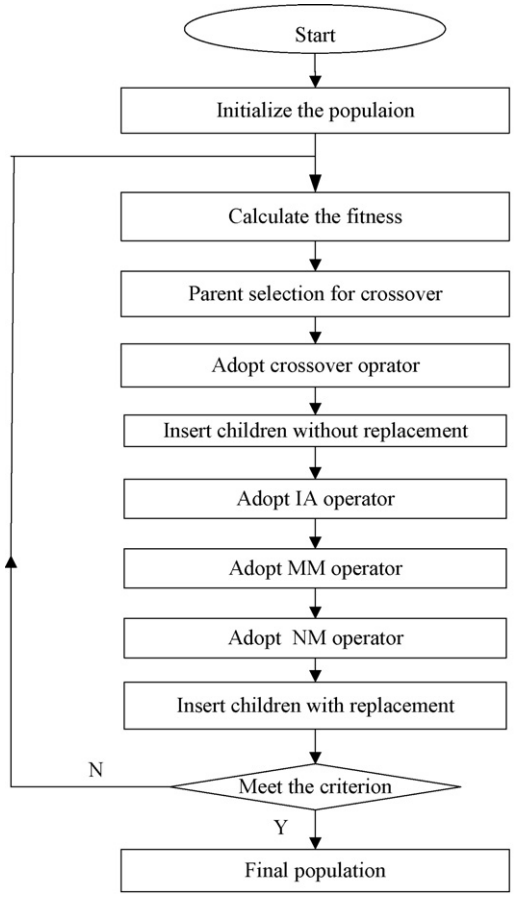

Fig. 6. Procedure of the DNA-GA.

benchmark functions reported in the literature to test the performance of the DNA-GA compared with RNA-GA [15] and GA. The details of the functions are shown in Table 1.

These functions contain different degrees of fraudulence that is expected to test the ability of the algorithm to overcome the fraudulence. The former four functions are two-dimensional, and the latter two functions are ten dimensional that are used to test the algorithm's performance in high dimensional problems.

3.1. Parameter specification of the DNA-GA

In the DNA-GA, there are four operator parameters: P_c, P_m, P_{IA} and P_{MM}. From the description of the operators in Section 2.2, we can find the former two operators, crossover and NM operator, are similar to the N-point crossover and bitwise mutation in binary GA. Hence, the values of P_c, P_m can be decided from references to the recommended range reported in the previous literatures, which is [0.6, 1] and [0.001, 0.1] respectively [8,20,21]. Here, we set P_c to be 0.75 and P_m to be 0.001 in this paper. As for the latter two operators, they are different from any traditional operators in GA. In order to demonstrate the values of their probabilities and examine their effect, we adopt the DNA-GA to optimize f_4 with various values of P_{IA} and P_{MM}.

The first group of experiments use fixed probability $P_{IA} = 0.5$ but various P_{MM}; the second group use fixed $P_{MM} = 0.5$ but various P_{IA}. And the third group are adopted with fixed settings of P_{IA} and P_{MM}: $P_{MM} = P_{IA} = 0$ and $P_{MM} = P_{IA} = 1$. For each parameter setting, the DNA-GA runs 50 times independently. And the tendency curves of the average best-so-far objective function values of function f_4 are given in Figs. 7–9.

Figs. 7 and 8 exhibits that the convergence speed is the slowest and the algorithm will be trapped into the local optimum when none of the two new operators works, i.e. $P_{IA} = 0$ and $P_{MM} = 0$. If any of the two operators works, i.e. $P_{IA} = 0$ or $P_{MM} = 0$, the convergence speed of the DNA-GA will be increased. Furthermore, when P_{IA} is fixed, the increase of P_{MM} will increase the convergence rate of the algorithm seen in Fig. 7. The same effect is observed in Fig. 8 too. Fig. 9 shows the comparison of the convergence speed with different setting of P_{IA} and P_{MM}. It is obvious that the DNA-GA converges to the global optimum more quickly when $P_{IA} = 0.5$ and $P_{MM} = 1$, compared with $P_{IA} = 1$ and $P_{MM} = 0.5$. Furthermore, the highest convergence rate happens when both operators are employed with probability 1.

On the other hand, we should notice that the increase of the operator probabilities could result in an increase of the computing time. To demonstrate this effect, we list the average CPU time that the DNA-GA needs to reach the global optimum under different parameter settings in Table 2. Here, we assume the algorithm finds the global optimum when the distance between the best-so-far individual and the known global minimum is smaller than 0.1.

As shown in Figs. 7–9, the algorithm can reach the global optimum in the smaller generation with the increase of the operator probability. However, the data in Table 2 indicate that CPU time could become larger at the same time. For example, in f_4, the CPU

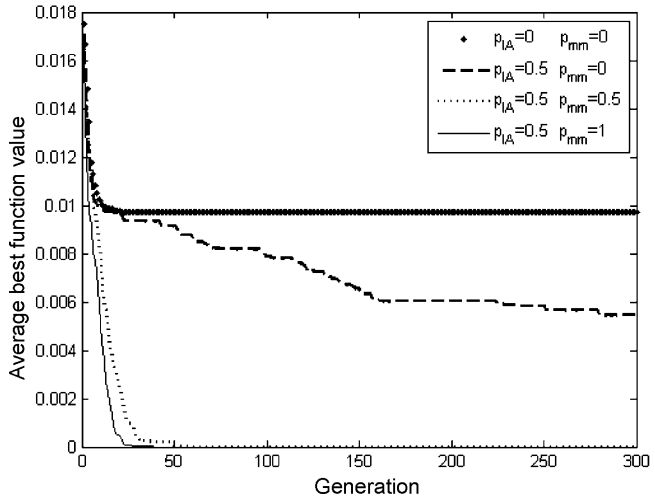

Fig. 7. Convergence curves of DNA-GA with different setting of P_{mm} for f_4.

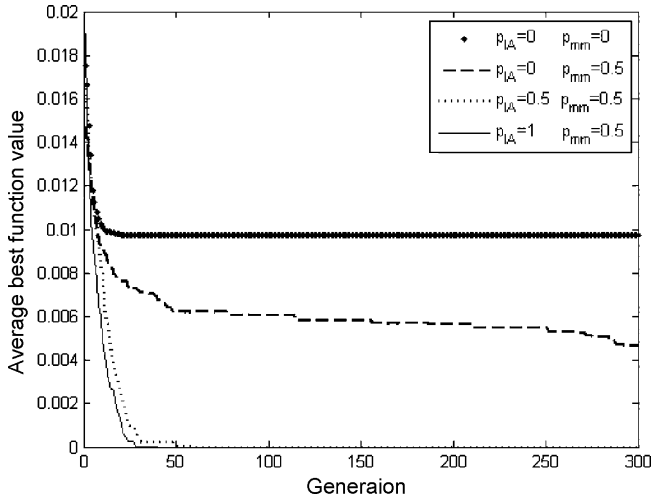

Fig. 8. Convergence curves of DNA-GA with different setting of P_{IA} for f_4.

time is 0.39 s when $P_{IA} = 1$ and $P_{MM} = 1$, which is a little longer than the time needed when $P_{IA} = 0.5$ and $P_{MM} = 1$:0.35 s. It is probably because the larger operator probability requires more time for the algorithm to finish the operation of the genetic operators in one

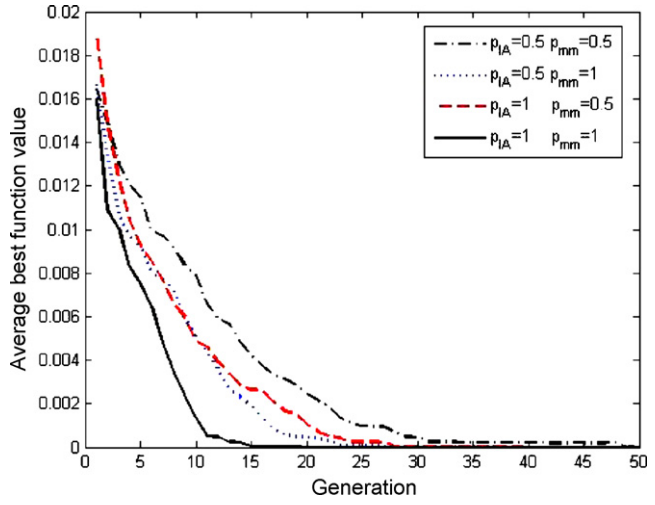

Fig. 9. Convergence curves of DNA-GA with different setting of P_{IA} and P_{mm} for f_4.

Table 2
computing time under different setting parameters of DNA-GA.

Parameter setting	f_4	f_6
	CPU time(s)	CPU time(s)
$P_{IA} = 1$, $P_{mm} = 1$	0.39	1.05
$P_{IA} = 0.5$, $P_{mm} = 1$	0.35	0.97
$P_{IA} = 1$, $P_{mm} = 0.5$	0.62	1.61
$P_{IA} = 0.5$, $P_{mm} = 0.5$	0.56	1.51

generation. And this effect is more obvious with the increase of the number of decision variable or the complexity of the objective function, like in function f_6 that possesses 10 decision variables. Hence, considering the computing time, we choose the setting of $P_{IA} = 0.5$ and $P_{MM} = 1$ in this paper.

From the above discussion, we can conclude that the two novel operators can increase the convergence speed of the DNA-GA, and the convergence rate is sensitive to the value of P_{IA} and P_{MM}. However, like other GAs, the values of P_c, P_m, P_{IA} and P_{MM} are not fixed for all the problems and they can be adjusted according to different applications.

3.2. Comparison and discussion

In this subsection, we will compare the performance of three optimization methods: DNA-GA, RNA-GA and GA. To ensure the comparison is fair, the population size N and the length of the string representing each variable l are set uniformly as: $N = 60$ and $l = 20$. The parameters of the test algorithms are set as follows: In RNA-GA, the three crossover probabilities are set as 0.8,0.5 and 0.5, and the mutation probability is remained as in [15]; The parameters of DNA-GA are set as:$P_c = 0.75$, $P_{IA} = 0.5$, $P_{MM} = 1$, and $P_m = 0.001$;GA is realized by the GA toolbox of MATLAB.[1]

All the test algorithms are terminated when the distance between the objective function value of the best-so-far individual and that of the known global optimum is smaller than Δ or the generation number is up to 1000. Here, we set Δ to be 0.0001.

For every test function, each algorithm runs $R = 50$ times. Here, we use the average evaluation generation number G_{ave}, the maximum and minimum evaluation number G_{max} and G_{min} over R runs to measure the convergence speed of the algorithm, where G_{ave} computed by $G_{ave} = \frac{1}{R}\sum_{i=1}^{R} G_i$ and G_i is the actual evaluation generation number in i_{th} run. Obviously, the less G_{max}, G_{min} and G_{ave} are, the faster the algorithm is. The corresponding data are shown in Table 3. The global search ability is measured by, F_{max}, F_{min} and F_{ave}, which denotes the maximum, minimum and average optimal value of the best objective function over R runs, respectively. The corresponding data are shown in Table 4. Then, *Suc* in Table 5 is employed to show the reliability of the test algorithm, where $Suc = R_{suc}/R$ and R_{suc} is the number of the runs that the distance between the best solution and the known global optimum is smaller than 0.1 over R runs. The average CPU time of each algorithm is also presented in Table 5.

First, we analyze their performance on two-dimensional functions f_1–f_4. From the values of *Suc* shown in Table 5, we can find that GA cannot guarantee to reach the global optimum in these four test problems. Compared with GA, RNA-GA is more reliable.

[1] To get better performance of GA, the setting of GA is different for each test function: uniform selection, heuristic crossover for f_1 and f_5, roulette selection and two-point crossover for f_2 and f_6, uniform selection and arithmetic crossover for f_3 and f_4. The mutation is adaptive and the probability of crossover is 0.85.

Table 3
Comparison of efficiency of three algorithms.

Test function	DNA-GA			RNA-GA			GA		
	G_{max}	G_{min}	G_{ave}	G_{max}	G_{min}	G_{ave}	G_{max}	G_{min}	G_{ave}
f_1	32	4	14.38	1000	6	389	1000	31	606.4
f_2	135	9	47.12	987	8	154.2	1000	11	642.2
f_3	75	8	31.08	592	393	542	1000	165	540.52
f_4	27	4	13.7	1000	20	559.48	1000	11	623.34
f_5	122	19	45.74	1000	1000	1000	1000	582	837.46
f_6	32	16	19.4	770	661	695.52	241	136	185.84

Table 4
Comparison of accuracy ability of three algorithms.

Test function	DNA -GA			RNA-GA			GA		
	F_{max}	F_{min}	F_{ave}	F_{max}	F_{min}	F_{ave}	F_{max}	F_{min}	F_{ave}
f_1	9.07E-05	6.20E-13	1.46E-05	0.0074	2.51E-06	0.0021	0.0271	7.35E-07	0.0056
f_2	9.97E-05	1.72E-13	1.97E-05	9.99E-05	2.29E-08	5.13E-05	0.4583	3.38E-05	0.0407
f_3	3600	3600	3600	3600	3600	3600	3600	3460.6	1000
f_4	9.90E-05	1.84E-13	1.52E-05	0.0097	8.39E-02	0.0045	0.0097	1.42E-06	0.0049
f_5	9.83E-05	4.48E-08	3.49E-05	19.2303	4.5976	71.4085	19.8044	8.54E-05	2.2765
f_6	9.57E-05	5.13E-13	1.67E-05	9.98E-05	1.09E-05	7.23E-05	9.94E-05	5.16E-05	8.98E-05

Its success rates both reach 100% in f_2 and f_3, and *Suc* is largely improved in another two functions. However, the improvement of success rate of RNA-GA may increase the computing time in some problems. Like in f_4, although the success rate of RNA-GA is rising, its CPU time is evidently longer than GA. As comparison, the DNA-GA is far more reliable than the other algorithms that it successfully converges to the global optimum in each run for all these four functions. Moreover, as shown in Table 5, the computing time of DNA-GA is much smaller than that of the other two methods. For example, in f_1, the average CPU time is only 0.61 s, which is remarkably shorter than that of RNA-GA and GA, 3.75 s and 5.04 s respectively. Then, the values of G_{ave}, G_{max} and G_{min} in Table 3 reflect the convergence speed of the testing algorithm. Obviously, G_{ave} and G_{max} of the DNA-GA are much lower than those of the two others, which mean the DNA-GA converges to the global optima in the smaller generations. The DNA-GA also shows its superiority in the quality of the optima. Table 4 clearly demonstrates that the DNA-GA finds more satisfactory optima than RNA-GA and GA.

Afterwards, the performance of three algorithms on high dimensional functions is discussed. Function f_5 and f_6 are both 10-dimensional functions. The former is multimodal, and the latter is unimodal. For f_6, the success rates of all three algorithms are all 100%. But the RNA-GA is much slower than the other methods that the computing time of RNA-GA is 19.97 s while that of DNA-GA and GA is 1.21 s and 2.02 s, respectively. This phenomenon becomes more obvious in the more complicated test function f_5. In f_5, RNA-GA cannot find the global optimum in any test run while GA is much better that its successful rate is 88%. As comparison, the successful rates of DNA-GA are 100% in both functions with less computing time. Such as in f_5, the average computing time of DNA-GA is 3.1 s, which is less than the half of GA.

To sum up, we can know that the DNA-GA is superior to RNA-GA and GA both in the convergence speed and the probability to converge to the global optimum. Moreover, it performs preferable in high-dimensional functions where RNA-GA failed. Hence, the DNA-GA is an efficient and reliable optimization method.

4. Parameter estimation for the kinetic hydrogenation models

4.1. Kinetic models of hydrogenation reactions

There are many published kinetic models for the hydrogenation reactions in literatures [22–25]. However, most of them have the following characteristics: they only aim at the removal of single impurity; the reaction period is calculated by hour; the deactivation of catalyst is not considered. As a result, these models are not suitable for hydrogenation reactions of an industrial hydrogenation unit, Maoming's Residue hydrodesulfurization treatment unit (S-RHT). In this context, Xu et al. proposed the kinetic models for hydrogenation reactions based on the operation data gathered from Maoming's S-RHT [16–17]. In Xu's models, the complicated process is simplified into five independent reactions: desulphurization, denitrification, decarbonization, denickel and devanadium. And each reaction can be described as follows:

Table 5
Comparison of efficiency and reliability ability of three algorithms.

Test function	DNA-GA		RNA-GA		GA	
	CPU time(s)	Suc (%)	CPU time(s)	Suc (%)	CPU time(s)	Suc (%)
f_1	0.61	100	3.75	72	5.04	44
f_2	1.8	100	3.01	100	5.61	76
f_3	1.14	100	4.23	100	3.37	74
f_4	0.54	100	6.58	62	3.3	46
f_5	3.1	100	26.24	0	7.46	88
f_6	1.21	100	19.97	100	2.02	100

If order $\mu = 1$.

$$\hat{C}_{\text{out}} = C_{\text{in}} \times \exp\left(\frac{1}{1+\left(\frac{t_1}{T_1}\right)^{\beta}} \times (-k_0) \times \exp\left(\frac{-E}{RT_2}\right) \times t_2\right) \quad (2)$$

If order $\mu \neq 1$

$$\hat{C}_{\text{out}} = \left(C_{\text{in}}^{1-\mu} + (\mu-1) \times \frac{1}{1+\left(\frac{t_1}{T_1}\right)^{\beta}} \times k_0 \times \exp\left(\frac{-E}{RT_2}\right) \times t_2\right)^{\frac{1}{1-\mu}} \quad (3)$$

In each reaction, there are five parameters to be estimated: μ, k_0, E, β, T_1. And there are 25 unknown parameters in five reactions.

Generally, such parameter estimation problems can be cast as non-linear optimization problems by minimizing the errors between the estimated outputs and the real outputs. Here, the objective function is defined as the sum of squared deviations of the removal fractions predicted by the models from the industrially measured values:

$$\begin{cases} f = \sum_{p=1}^{ns}\left(\frac{y_p - \hat{y}_p}{y_p}\right)^2 \\ y_p = 1 - \frac{C_{\text{outp}}}{C_{\text{inp}}}, \hat{y}_p = 1 - \frac{\hat{C}_{\text{outp}}}{C_{\text{inp}}} \end{cases} \quad (4)$$

where ns is the sample number, y_p is the real removal fraction of the P_{th} group, $\hat{y}_p$ is the model predicted removal fraction, C_{inp} is the concentration of the impurity in the feedstock of the P_{th} group, C_{outp} and $\hat{C}_{\text{outp}}$ is the real and predicted concentration of impurity of the product of the P_{th} group. By minimizing the objective function Eq. (4), all the 25 kinetic parameters can be determined.

4.2. Parameter estimation

Due to the superior performance of the DNA-GA, it is applied to the parameter estimation problem (4), where traditional methods failed because of the non-linearity and other complicated characteristics of chemical processes. 18 Groups of operation data were recorded after the unit running for 503 days. Each group include the average temperature in five reactors, the concentration of impurities at entry, the concentration of impurities at exit, the feedstock density and the feedstock mass flowrate. And t_2 in Eqs. (2) and (3) is computed as follows:

$$t_2 = 710\frac{\text{feedstock density}}{\text{feedstock mass flowrate}} \quad (5)$$

where 710 is the reactor volume. The maximum evolution number of the DNA-GA is set to be 300, the population size is 150, and the four operator probabilities remain the same as in Section 3. The kinetic parameters of the hydrogenation model estimated by the DNA-GA are listed in Table 6.

Based on the above parameters and the operation data from S-RHT, the kinetic hydrogenation model for S-RHT is described as follows:

Desulphurization:

$$C_{R105.S} = \left(C_{R101.S}^{-1.3} + 1.3 \times \frac{1}{1+\left(\frac{t_1}{466}\right)^{7.6}} \times 100 \times \exp\left(\frac{-27983}{8.3145T_2}\right) t_2\right)^{-\frac{1}{1.3}} \quad (6)$$

Denitrification:

$$C_{R105.N} = \left(C_{R101.N}^{-0.63} + 0.63 \times \frac{1}{1+\left(\frac{t_1}{413}\right)^{0.1}} \times 1.8 \times \exp\left(\frac{-34663}{8.3145T_2}\right) t_2\right)^{-\frac{1}{0.63}} \quad (7)$$

Decarbonization:

$$C_{R105.C} = \left(C_{R101.C}^{-0.88} + 0.88 \times \frac{1}{1+\left(\frac{t_1}{489}\right)^{9.6}} \times 48.5 \times \exp\left(\frac{-40289}{8.3145T_2}\right) t_2\right)^{-\frac{1}{0.88}} \quad (8)$$

Denickel:

$$C_{R105.Ni} = \left(C_{R101.Ni}^{-0.4} + 0.4 \times \frac{1}{1+\left(\frac{t_1}{525}\right)^{5}} \times 8.9 \times \exp\left(\frac{-25685}{8.3145T_2}\right) t_2\right)^{-\frac{1}{0.4}} \quad (9)$$

Devanadium:

$$C_{R105.V} = \left(C_{R101.V}^{-0.32} + 0.32 \times \frac{1}{1+\left(\frac{t_1}{491}\right)^{9.7}} \times 11.4 \times \exp\left(\frac{-24741}{8.3145T_2}\right) t_2\right)^{-\frac{1}{0.32}} \quad (10)$$

In order to verify the validity of the obtained models, another four group data gathered from S-RHT are selected as test data. The predicted concentration at exit and the predicted removal fraction of five reactions according to each test group are shown in Table 7. The relative errors according to each test group are also given in Table 7. Here, relative error $= \frac{y_{\text{reali}} - y_{\text{predicted}i}}{y_{\text{reali}}}$, where, y_{reali} is the real removal fraction in i_{th} group, and $y_{\text{predicted}i}$ is the predicted removal fraction in i_{th} group.

From Table 7, we can find that:

(1) For the kinetic desulphurization model, the maximum relative error is 3.9%, the minimum relative error is 0.15%, and the average relative error is 1.73%.
(2) For the kinetic denitrification model, the maximum relative error is 2.75%, the minimum relative error is 1.21%, and the average relative error is 2.13%.
(3) For the kinetic decarbonization model, the maximum relative error is 4.4%, the minimum relative error is 0.05%, and the average relative error is 2.81%.
(4) For the kinetic denickel model, the maximum relative error is 6.74%, the minimum relative error is 0.93%, and the average relative error is 3.72%.

Table 6
Estimated kinetic model parameters.

Reaction	μ (dimensionless)	k_0 (dimensionless)	E (J mol^{-1})	β (dimensionless)	T_1/day
Desulphurization	2.3	100	27983	7.6	466
Denitrification	1.63	1.8	34663	0.1	413
Decarbonization	1.88	48.5	40289	9.6	489
Denickel	1.4	8.9	25685	5	525
Devanadium	1.32	11.4	24741	9.7	491

Table 7
Predicted results of the obtained kinetic models.

Test group	Predicted concentration at exit, μg g^{-1}			Predicted removal fraction (%)			Relative error of fraction		
	Sulphur	Nitrogen	Carbon	Sulphur	Nitrogen	Carbon	Sulphur	Nitrogen	Carbon
19	0.3575	1084	5.0489	0.8345	0.5117	0.4946	−0.0070	−0.0243	−0.0381
20	0.3431	999.6	5.1968	0.9014	0.5377	0.5882	−0.0218	−0.0215	−0.0440
21	0.3677	1221	5.2138	0.8782	0.6432	0.6109	0.0390	0.0121	−0.0005
22	0.3630	853.6	4.6367	0.8487	0.6936	0.5893	0.0015	0.0275	0.0301
	Nickel	Vanadium		Nickel	Vanadium		Nickel	Vanadium	
19	6.7639	8.074		0.6090	0.7387		−0.0279	−0.0042	
20	6.3955	10.1470		0.6370	0.8114		−0.0093	0.0267	
21	7.2876	11.7793		0.6224	0.8303		0.0674	0.0201	
22	5.6681	6.8804		0.6479	0.7952		−0.0442	−0.0210	

Table 8
Comparison of estimated kinetic models obtained by different methods.

Reaction	This paper	[16–17]	[26]
Desulphurization	0.0020	0.0028	0.0034
Denitrification	0.0020	0.0153	0.02
Decarbonization	0.0043	0.0232	0.0513
Denickel	0.0074	0.0202	
Denickel	0.0016	0.0021	

(5) For the kinetic devanadium model, the maximum relative error is 2.67%, the minimum relative error is 0.42%, and the average relative error is 1.8%.

The average relative error of the five models is 2.44%. Considering the data we used are gathered from S-RHT, the industrial unit working conditions are very complicated and some occasional factors are unpredictable. In view of this, the precision of the estimated models is acceptable.

Moreover, we use the same test data to examine the previously reported models obtained by other methods [16,17,26]. For every reaction, the sum of square relative errors of each model is listed in Table 8. From this table, we can see that the prediction of the model established by the DNA-GA is in better agreement with the practical operating data than the other models.

5. Conclusions

Kinetic modeling is an important issue for the optimization of the chemical process and mathematical modeling of these complex process leads to non-linear parameter estimation problems which often contain more than one minimum. In this work, a DNA-GA is proposed to determine 25 kinetic parameters of hydrogenation reactions for its strong global search ability. The DNA encoding method is adopted to represent the potential solutions and new genetic operators are designed to enhance the global searching ability of the DNA-GA. The implementation of two operators, IA operator and MM operator, was shown to be powerful in seeking the global optimum of the complex function. To testify the effectiveness of the DNA-GA, its performance has been statistically analyzed using a number of typical benchmark functions. Compared with RNA-GA and GA, the DNA-GA possesses strong global searching ability to overcome different degree of fraudulence with fast convergence speed, and its improvement of performance in high-dimensional problems is obvious. The kinetic models obtained by the DNA-GA are compared with the previous models based on the operating data gathered from an industrial unit, and the predictions of the estimated models are in good agreement with the practical operating data. Nevertheless, the proposed operators only can be adopted in the algorithms where individuals are represented with four bases by far. Hence, a further study to employ the proposed operators over other encoding type chromosomes will be valuable.

Acknowledgement

This work was supported by the National Natural Science Foundation of China under grants nos. 60874072 and 60721062.

References

[1] W.H. Su, H.Z. Huang, Development and calibration of a reduced chemical kinetic model of *n*-heptane for HCCI engine combustion, Fuel 84 (2005) 1029–1040.
[2] Z.Y. Zhang, K. Hidajat, Ajay K. Ray, Determination of adsorption and kinetic parameters for methyl tert-butyl ether synthesis from tert-butyl alcohol and methanol, J. Catal. 200 (2001) 209–221.
[3] J.H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan press, Ann Arbor, 1975.
[4] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, New York, 1989.
[5] José María Nougués, M. Dolors Grau, Luis Puigjaner, Parameter estimation with genetic algorithm in control of fed-batch reactors, Chem. Eng. Process. 41 (2002) 303–309.
[6] M. Izadifar, O.D. Baik, Application of genetic algorithm for determination of the effective diffusivity of andrographolide in plant materials, Biochem. Eng. J. 33 (2007) 178–187.
[7] G. Milani, F. Milani, Genetic algorithm for the optimization of rubber insulated high voltage power cables production lines, Comput. Chem. Eng. 32 (2008) 3198–3212.
[8] K.A. De Jong. An analysis of the behavior of a class of genetic adaptive systems. Ph.D. Dissertation, University of Michigan.1975.
[9] R. Moros, H. Kalies, H.G. Rex, S. Schaffarczyk, A genetic algorithm for generating initial parameter estimations for kinetic models of catalytic process, Comput. Chem. Eng. 20 (1996) 1257–1270.
[10] L. Balland, N. Mouhab, J.M. Cosmao, L. Estel, Kinetic parameter estimation of solvent-free reactions: application to esterification of acetic anhydride by methanol, Chem. Eng. Proc. 41 (2002) 395–402.
[11] Behrang Mansoornejad, Navid Mostoufi, Farhang Jalali-Farahani, A hybrid GA–SQP optimization technique for determination of kinetic parameters of hydrogenation reactions, Comput. Chem. Eng. 32 (2008) 1447–1455.

[12] L.H. Ren, Y.S. Ding, (2004). Parameter optimization for a class of general TS fuzzy controllers via a new DNA-based genetic algorithm. Intelligent Control and Automation, 2004. WCICA 2004. Fifth World Congress on. pp. 2149-2153.
[13] T. Yoshikawa, T. Furuhashi, Y. Uchikawa, DNA coding method and a mechanism of development for acquisition of fuzzy control rules., in: Proceedings of the 5th IEEE International Conference on Fuzzy Systems, New Orleans, LA, USA, September 8–11, 1996, pp. 2194–2200.
[14] J.A. Rose, M. Hagiya, R.J. Deaton, A. Suyama, DNA-based in vitro genetic program, J. Biol. Phys. 28 (2002) 493–498.
[15] J.L. Tao, N. Wang, DNA computing based RNA genetic algorithm with applications in parameter estimation of chemical engineering processes, Comput. Chem. Eng. 31 (2007) 1602–1618.
[16] X.K. Xu, L.C. Zhang, H.X. Weng, Study on kinetics model for residue hydrofining, Petrol. Refinery Eng. 35 (2005) 44–48.
[17] L.C. Zhang, X.K. Xu, H.X. Weng, Hydrodemetallization kinetic model for Maoming's residue Hydrodesulfurization Treatment Unit, J. Chem. Eng. Chin. Univ. 20 (2006) 227–232.
[18] David Clerk, Molecular Biology: Understand the genetic revolution, Elsevier, 2005.
[19] S.C. Li, J. Xu, Digital Coding for RNA Based on DNA Computing, Comput. Eng. Appl. 39 (2003) 46–47.
[20] J. Grefenstette, Optimization of control parameters for genetic algorithms, IEEE Trans. Syst. Man Cybern. SMC-16 (1986) 122–128.
[21] J.D. Schaffer, R.A. Caruana, L.J. Eshelman, R. Das, A study of control parameters affecting online performance of genetic algorithms for function optimization, in: J.D. Schaffer (Ed.), Proc 3rd Int. Conf. on Genetic algorithms, Fairfax, VA, June, Morgan Kaufmann, San Mateo, CA, 1989, pp. 51–60.
[22] F.X. Long, Börje S. Gevert, Kinetics of vanadyl etioporphyrin hydrodemetallization, J. Catal. 200 (2001) 91–98.
[23] F.X. Long, Kinetic parameter estimation and statistical analysis of vanadyl etioporphyrin hydrodemetallization, Comput. Chem. Eng. 27 (2003) 697–700.
[24] W.X. Zhao, D.Z. Chen, S.X. Hu, Differential fraction-based kinetic model for simulating hydrodesulfurization process of petroleum fraction, Comp. Chem. 26 (2002) 141–148.
[25] X.F. Yan, D.Z. Chen, S.X. Hu, Chaos-genetic algorithms for optimizing the operating conditions based on RBF-PLS model, Comput. Chem. Eng. 27 (2003) 1393–1404.
[26] Q.F. Zheng, H.Z. Liu, Modified holographic research strategy and its application to optimization in chemical engineering, J. Chem. Ind. Eng. Chin. 57 (2006) 2349–2354.