Available online at www.sciencedirect.com
SCIENCE DIRECT®

Computers and Chemical Engineering 27 (2003) 1393–1404

Computers & Chemical Engineering

www.elsevier.com/locate/compchemeng

Chaos-genetic algorithms for optimizing the operating conditions based on RBF-PLS model

Xuefeng F. Yan [a], Dezhao Z. Chen [b,*], Shangxu X. Hu [b]

[a] *Automation Institute, East China University of Science and Technology, Shanghai 200237, China*
[b] *Department of Chemical Engineering, Zhejiang University, Hangzhou 310027, China*

Received 22 January 2002; received in revised form 4 December 2002; accepted 24 February 2003

Abstract

A novel genetic algorithm (GA) including chaotic variable named chaos-genetic algorithm (CGA) was proposed. Due to the nature of chaotic variable, i.e. pseudo-randomness, ergodicity and irregularity, the evolutional process of CGA makes the individuals of subgenerations distributed ergodically in the defined space and circumvents the premature of the individuals of subgenerations. The performance of CGA was demonstrated through two examples and compared with that by the traditional genetic algorithms (TGA). The results showed the superior performances of CGA over TGA, and moreover, the probability of finding the global optimal value by using CGA is larger than that by using TGA. To illustrate the performance of CGA further, it was employed to optimize the operating conditions of aromatic hydrocarbon isomerization (AHI) process modeled by the radial basis functions (RBF) coupled with partial least squares (PLS) approach. Satisfactory results were obtained. Further, a generalized methodology, which employs RBF-PLS approach to model the complex chemical process based upon practical observation data and subsequently applies CGA to find the optimal operating conditions, was suggested too.

Keywords: Chaos-genetic algorithms; Chaotic variable; Optimization; Radial basis functions; Partial least squares; Model

1. Introduction

Genetic algorithm (GA) was introduced by Holland in 1970s last century as optimization approach and the main concept of this approach was derived from biological evolution in a competitive environment (Holland, 1975). Due to its favorable virtue on searching optimum (Upreti & Deb, 1997) and the recent progress in GA theory, many industrial applications have been developed with the aid of this tool (Hugget, Sébastian & Nadeau, 1999). The objects of these applications include unit operation equipment (Hugget et al., 1999), process control (Sender, 1994), gas pipeline design (Goldberg, 1989), pattern recognition analysis of multivariate chemical data (Lavine, Davidson & Moores, 2002), optimization of reaction rate parameters (Balland, Estel, Cosmao & Mouhab, 2000; Harris, Elliott, Ingham, Pourkashanian & Wilson, 2000), multipurpose chemical batch plant design (Leonardo, Catherine, Serge & Lue, 1998), and scheduling (Pongcharoen, Hicks, Braiden & Stewardson, 2002; Yun, 2002). In order to find a global or near-global optimal solution, the search by GA was extended to a group base instead of the former point-to-point search. The group, which contains several solution points, is named population and is represented by $P(t)$ with t denoting the number of generations. The current generation $P(t)$ evolves the next generation $P(t+1)$ using genetic operations such as proportionate reproduction following the well-known Darwinian theory of survival of the fittest. In this way, $P(t)$ continually evolves along with the proceeding of t till a global or near-global optimal solution is obtained. The solution points in the population, which are called individuals or chromosomes, are represented by $x_1(t)$, $x_2(t)$, ..., $x_{n_p}(t)$ with n_p denoting the number of the individuals in the population. The value of n_p usually keeps unchanged in whole evolution process. The individual $x_i(t)$ consists of a series of independent variables, which are called as

* Corresponding author.
E-mail addresses: xfyan@telekbird.com.cn (X.F. Yan), dzchen@mail.hz.zj.cn (D.Z. Chen), sxhu@mail.hz.zj.cn (S.X. Hu).

0098-1354/03/$ - see front matter © 2003 Elsevier Science Ltd. All rights reserved.
doi:10.1016/S0098-1354(03)00074-7

genes, namely: $x_{i1}(t)$, $x_{i2}(t)$, ..., $x_{in}(t)$, in which n denotes the number of independent variables. The main genetic operations such as the encoding of the solution of optimizing problem, the design and application of fitting function, and the crossover and mutation of individuals, play important roles in GA (Holland, 1975; Zhao, Chen & Hu, 2000).

Chaos often exists in nonlinear systems. It is the highly unstable motion of deterministic systems in finite phase space. Chaos theory is epitomized by the so-called 'butterfly effect' detailed by Lorenz (1963). Attempting to simulate numerically a global weather system, Lorenz discovered that minute changes in initial conditions steered subsequent simulations towards radically different final stales. This sensitive dependence on initial conditions is generally exhibited by systems containing multiple elements with nonlinear interactions, particularly when the system is forced and dissipative. Sensitive dependence on initial conditions is not only observed in complex systems, but even in the simplest logistic equation. In the well-known logistic equation (May, 1976):

$$x_{n+1} = f(\mu, x_n) = \mu x_n(1 - x_n), \tag{1}$$

in which μ is a control parameter, $n = 0, 1, 2, \ldots$, and x is a variable. Suppose $0 \le x_0 \le 1$, $0 \le \mu \le 4$. It is easy to find that Eq. (1) is a deterministic dynamic system without any stochastic disturbance. It seems that its long-time behavior can be predicted. But that is not true. The behavior of system Eq. (1) is greatly changed with the variation of μ. The value of the control parameter μ determines whether x stabilizes at a constant size, oscillates between a limited sequence of sizes, or whether x behaves chaotically in an unpredictable pattern. In the latter case, i.e. $\mu = 4$, the equation is changed to:

$$x_{n+1} = 4x_n(1 - x_n). \tag{2}$$

And, very small difference in the initial value of x causes large difference in its long-time behavior, which is the basic characteristic of chaos. The variable x is called as chaotic variable. Thus, a minute difference of the initial value of the chaotic variable would give rise to a considerable difference of the values of chaotic variable later. The track of chaotic variable can travel ergodically over the whole space of interest. The variation of the chaotic variable has a delicate inherent rule in spite of the fact that its variation looks like in disorder. In general, there are three main characteristics of the variation of the chaotic variable, i.e. pseudo-randomness, ergodicity and irregularity (Li & Jiang, 1998; Bountis, 1995; Ohya, 1998).

GA has aroused intense interest, due to the flexibility, versatility and robustness in solving optimization problems, which conventional optimization methods find difficult. However, there exist some flaws on GA. For example, it tends to converge prematurely and takes a large number of iterations to reach the global optimal solution. In order to overcome these flaws of GA and based on the research upon the efficiency of linear crossover operation in real-coded GA and upon the merits of chaotic variable, chaos-genetic algorithms (CGA), which integrates GA with chaotic variable, was proposed in our work. CGA possesses a joint advantage of GA and the chaotic variable. Firstly, GGA takes the advantage of the characteristics of the chaotic variable to make the individuals of subgenerations distributed ergodically in the defined space and thus to avoid from the premature of the individuals in the subgenerations. Secondly, CGA also takes the advantage of the convergence characteristic of GA to overcome the randomness of the chaotic process and hence to increase the probability of finding the global optimal solution.

The remainder of this paper is organized as follows. Section 2 gives an overview of the efficiency of linear crossover operator in real-coded GA, presents our modification to conventional GA and proposed CGA. In Section 3, the radial basis functions-partial least squares (RBF-PLS) method is applied to model a twin-reactor aromatic hydrocarbon isomerization (AHI) process and CGA is employed to search the optimization of the operational conditions based on the model. Section 4 provides some concluding remarks.

2. Chaos-genetic algorithm

2.1. The efficiency of linear crossover operator in real-coded genetic algorithm

The performance of GA mainly depends on the genetic operations applied to the individual, namely: the selection, crossover, mutation, and so forth. The crossover operation plays an important role in distributing the individuals over the space of interest through the GA. In the literature (Michalewicz, Janikow & Krawczyk, 1992; Wright, 1991), a real-coded GA was proposed and chromosome vector is coded as same as the solution vector. There are many merits of real-coded GA. The precision of real-coded GA is generally much better than that of the binary-coded GA. And, it is easy to represent quite large space (or cases of unknown spaces), on the other hand, the binary-coded GA must sacrifice the precision for an increase in space size, given fixed binary length. In real-coded GA, the arithmetic crossover operation and the uniform mutation operation are designed to preserve the constraint. The arithmetic crossover operation was defined as:

$$\begin{cases} y_{ip} = \alpha_{ijp}x_{ip} + (1 - \alpha_{ijp})x_{jp} & i, j = 1, 2, \ldots, n_p \\ y_{jp} = \alpha_{ijp}x_{jp} + (1 - \alpha_{ijp})x_{ip} & p = 1, 2, \ldots, n \end{cases}, \tag{3}$$

in which x_{ip} and x_{jp} are the two values of the pth independent variable of ith and jth father individuals, respectively, y_{ip} and y_{jp}, are the two values of the pth independent variable of ith and jth child individuals, respectively, and α_{ijp} is a random number assigned by a value in the interval $0 \leq \alpha_{ijp} \leq 1$. And, the uniform mutation operation was defined as follows. Each chromosome vector has exactly equal chance of undergoing process. The result of the application of the uniform mutation operation for the selected chromosome vector $\boldsymbol{x}$ is that a random vector $\boldsymbol{x}'$ from the constraint space is used instead of $\boldsymbol{x}$.

In order to illustrate the distribution of the individuals when α_{ijp} takes different random value [0, 1] for every generation, a test named κ-test was designed to show the distribution of individuals of subgenerations according to Eq. (3). Several restrictions were set for the κ-test to eliminate the influence of too many factors and to display the performance of linear crossover operation of real-coded GA easier, for instance, the survival probability for every individual takes the same value, the crossover probability takes the value 1, and the mutation probability takes the value 0. Besides, the number of independent variables used in κ-test is 2, i.e. x_1 and x_2, with the values [0, 1]. The distribution of the individuals of total ten generations was illustrated in Fig. 1. It can be seen from Fig. 1 that the distribution of subgeneration individuals gradually concentrates to the center of the defined space. The κ-test showed the inherent disadvantage of the linear crossover operation of real-coded GA that it goes gradually apart from the global optimal search.

To improve the performance of GA search, the crossover operation should keep individuals scattered in the whole search space and hold the good individuals as well. After studying the nature of the chaotic process, a novel GA search method integrating the chaotic variable was proposed.

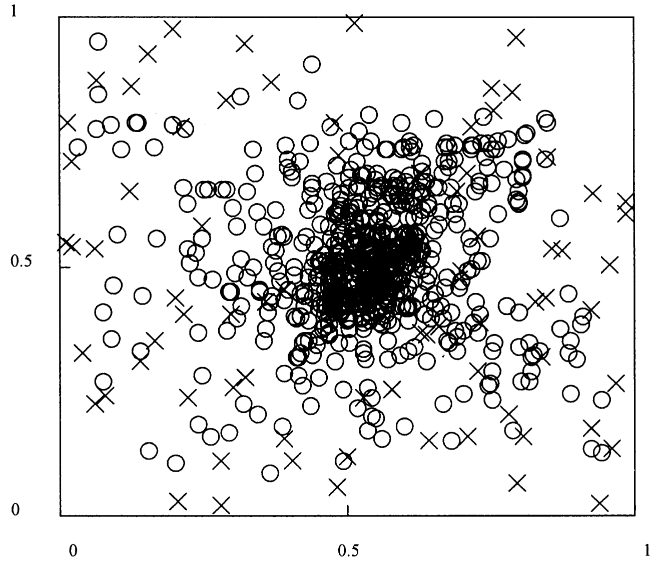


Fig. 1. The distribution of individuals by the arithmetic crossover operation of TGA. ×, Individuals of first generation; ○, individuals of subgenerations.

2.2. Chaotic mapping operator

Assume that the working vector of independent variables is denoted by $\boldsymbol{x}$ consisting of n elements. The elements of the working vector $\boldsymbol{x}$ are named as working parameters denoted by $x_1, x_2, \ldots, x_n$. Thus, a problem of searching minimum could be described as:

$$\begin{aligned} &\min_{x} \; f(x_1, x_2, \ldots x_n) \\ &x_i \in (a_i, b_i) \quad i = 1, 2, \ldots, n' \end{aligned} \tag{4}$$

where f denotes the function relating the value of dependent variables x and the dependent variable, which is subject to be optimized. An n-dimensional space S_0 is defined within the region $\{(a_i, b_i), i = 1, 2, \ldots, n\}$, of which the lower limit $[a_1, a_2, \ldots, a_n\ ]^T$ is represented by a vector $\boldsymbol{a}$, and the upper limit $[b_1, b_2, \ldots, b_n\ ]^T$ by a vector $\boldsymbol{b}$.

The evolution process of the chaotic variables could be defined through the following equation as same as Eq. (2) (Li & Jiang, 1998; May, 1976):

$$cx_i^{k+1} = 4cx_i^{(k)}(1 - cx_i^{(k)}), \quad i = 1, 2, \ldots, n, \tag{5}$$

in which cx_i is the ith chaotic variable, and (k) and ($k+1$) denote the number of iterations. The variable defined by Eq. (5) is chaotic and cx_i, is distributed in the range (0, 1.0) under the conditions that the initial $cx_i^{(t)} \in (0, 1.0)$ and that $cx_i^{(t)} \notin \{0.25, 0.5, 0.75\}$ to assure the evolution process going on properly.

In order to take advantage of chaotic evolution of variables, the follow procedures were proposed.

1) The working parameters $x_i^{(1)}$, $i = 1, 2, \ldots, n$ are linearly mapped from the defined space S_0 to a normalized chaotic space CS_1 defined as $\{(0, 1)_i, i = 1, 2, \ldots, n\}$. The linear mapping operator is:

$$cx_i = \frac{1}{b_i - a_i}(x_i - a_i), \quad i = 1, 2, \ldots, n. \tag{6}$$

where cx_i, $i = 1, 2, \ldots, n$ are chaotic variables. Denote the first iteration as $x_i^{(1)}$, and the result of applying Eq. (6) gives the first iteration of chaotic variables $cx_i^{(1)}$.
2) The next iteration chaotic variables $cx_i^{(2)}$ will be produced through applying the iteration operator defined by Eq. (5) to $cx_i^{(1)}$.
3) The chaotic variables $cx_i^{(2)}$ are linearly mapped from the space CS_1 to the space S_0 by the linear back-mapping operator to produce the working parameter $x_i^{(2)}$ as follows.

$$x_i^{(2)} = a_i + cx_i^{(2)}(b_i - a_i), \quad i = 1, 2, \ldots, n. \tag{7}$$

Thus, through the operators defined by Eqs. (6), (5) and (7) for all working parameters of the (k)th iteration working vector $x^{(k)}$ the working vector $x^{(k)}$ will be chaotically mapped forward and backward to produce the $(k+1)$th iteration of working vector $x^{(k+1)}$. The above procedure as a whole is named as chaotic mapping operator (CMO) represented by ψ, that is,

$$x^{(k+1)} = \psi(x^{(k)}). \tag{8}$$

And, the flow chart of chaotic mapping operator is described in the Appendix A.

2.3. Chaos-genetic algorithm

The procedure of executing the CGA can be described in the following.

1) Determine the number of the population n_p the crossover probability P_c, the mutation probability P_m, and the maximum number of generation t_m. Set the current number of generation $t = 1$, and the first generation of individuals $x_1(t)$, $x_2(t)$, ..., $x_{n_p}(t)$, which are randomly produced with the values in the space S_0, and are denoted by $X_t^{(1)} = \{x_1(t), x_2(t), \ldots, x_{n_p}(t)\}$.
2) The chaotic individuals represented by $X_t^{(2)} = \{x_1(t)^{(2)}, x_2(t)^{(2)}, \ldots, x_{n_p}(t)^{(2)}\}$are produced by CMO ($\psi$) for the all individuals of $X_t^{(1)}$, i.e. $x_i(t)^{(2)} = \psi(x_i(t))$, $i = 1, 2, \ldots, n_p$.
3) Randomly select n_p individuals from the set consisting of $X_t^{(1)}$ and $X_t^{(2)}$ to form a reproduction set $X_t^{(r)}$ using the genetic operation of proportionate reproduction to meet the well-known Darwinian theory of survival of the fittest.
4) Produce the new population $X_t^{(3)}$ from $X_t^{(r)}$ through the crossover operation defined by Eq. (3).
5) Produce the new population $X_t^{(4)}$ from $X_t^{(3)}$ through the mutation operation. There are many kinds of mutation operations for choice, such as the uniform mutation operation of TGA, the inconsistent mutation operation introduced by Michalewicz et al. (1992), the adaptive mutation operation (Pan, Kang & Nie, 1996), and the others. Here, the uniform mutation operation is employed.
6) Set $t = t+1$, let $X_t^{(1)} = X_{t-1}^{(4)}$and repeat steps 2–5 as long as the number of generations is smaller than the allowable maximum number t_m.

The flow chart of CGA is described in the Appendix B. The distribution of the individuals of subgenerations generated by CGA was illustrated in Fig. 2. It can be seen from Fig. 2 that the individuals of subgeneartions are almost evenly scattered over the defined space and do not concentrates to the center of the defined space any more.

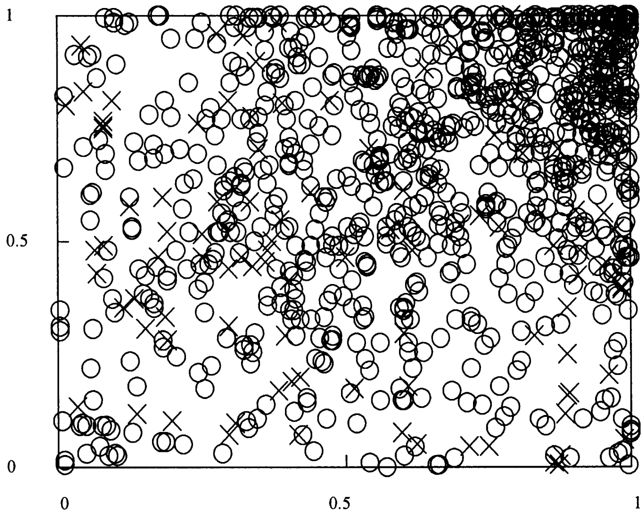


Fig. 2. The distribution of individuals by CGA. ×, Individuals of first generation; ○, individuals of subgenerations.

2.4. The Performance of CGA

The performance of CGA was firstly examined by two analytical functions, and compared with the performance of TGA. The optimum searching of analytical function does not depend on any knowledge of special domain and can be used to illustrate the performance of approach clearly. The two functions employed are:

$$1)\ f_1(\boldsymbol{x}) = \frac{1}{3}\sum_{i=1}^{3}(x_i^4 - 16x_i^2 + 5x_i) \text{ and the search problem is} \min_x f_1(\boldsymbol{x}), \quad -10 < x_i < 100, \quad i = 1, 2, 3. \tag{9}$$

$$2)\ f_2(\boldsymbol{x}) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2 \text{ and the search problem is} \min_x f_2(\boldsymbol{x}), \quad -100 < x_i < 100, \quad i = 1, 2. \tag{10}$$

The theoretical optimums of Eqs. (9) and (10) are −78.3323 and −1.0316, respectively.

CGA and TGA were used to search the global optimums of Eqs. (9) and (10), respectively. In order to compare the performance of CGA and that of TGA fairly, the best crossover probability and mutation probability is found for two approaches, respectively. In order to illustrate the performances of two optimization approaches with different algorithm parameters, the best value of object function obtained by them is denoted by v_{P_c,P_m}, in which P_c is crossover probability and P_m is mutation probability. The ranges of the crossover probability and the mutation probability are all [0.05, 0.95], and have been even 19 parts. Thus, P_c is one element of [0.05, 0.10, 0.15, ..., 0.95] as well as P_m. CGA or TGA with different combination between P_c and P_m is applied to search the optimal values of Eqs. (9) and (10). As for the other algorithm parameters, the two optimization approaches took the same values and the same random initial population as well. For Eq. (9), $n_p = 400$ and $t_m = 300$ were taken. For Eq. (10), $n_p = 100$ and $t_m = 100$ were taken. To reduce the influence of incidental variation, the total number of applying the optimization approach with the same combination to search the optimal value is 10, and each test has different random initial population. The mean value of v_{P_c,P_m} is calculated and denoted as $\bar{v}^*_{P_cP_m}$. Further, the mean performance of optimization approach with crossover probability P_c represented by $\bar{v}^*_{P_c}$ is calculated as:

$$\bar{v}^*_{P_c} = \frac{1}{19}\sum_{P_m=0.05}^{0.95} \bar{v}_{P_c,P_m}, \qquad (11)$$

and the mean performance of optimization approach with mutation probability P_m, represented by $\bar{v}^*_{P_m}$ is calculated as:

$$\bar{v}^*_{P_m} = \frac{1}{19}\sum_{P_c=0.05}^{0.95} \bar{v}_{P_c,P_m}. \qquad (12)$$

The mean performances of two optimization approaches with different crossover probability were calculated and shown in Figs. 3 and 5. And the mean performances of two optimization approaches with different mutation probability were calculated and shown in Figs. 4 and 6. It can be seen from Figs. 3–6 that the capability of CGA has superiority over that of TGA. And CGA is not so sensitive to the parameter of crossover probability as TGA, due to the fact that the mean performance of CGA with different crossover probability fluctuates within a limited size. As shown in Figs. 3 and 4, regarding Eq. (9), $P_c = 0.55$ and $P_m = 0.7$ are the best parameters for TGA, and $P_c = 0.35$ and $P_m = 0.1$ are the best parameters for CGA. And as shown in Figs. 5 and 6; regarding Eq. (10), $P_c = 0.95$ and $P_m = 0.8$ are the best parameters for TGA, and

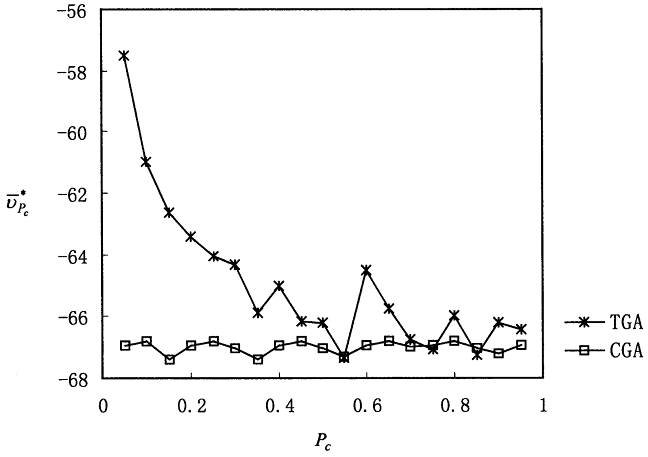


Fig. 3. The curves of mean performances of CGA and TGA with different crossover probability regarding Eq. (9).

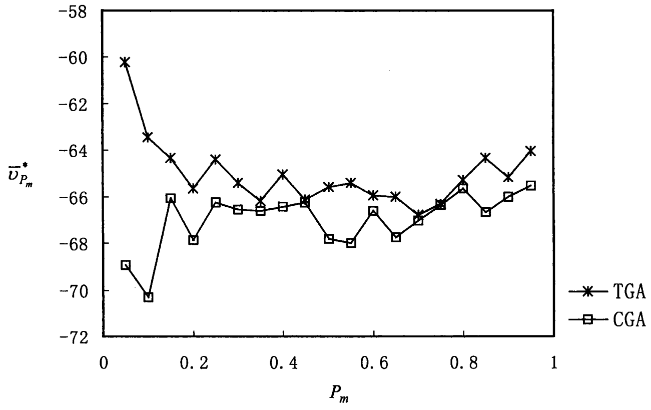


Fig. 4. The curves of mean performances of CGA and TGA with different mutation probability regarding Eq. (9).

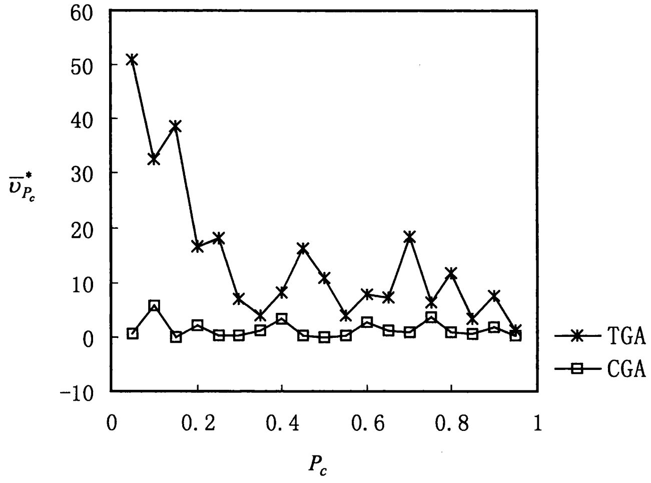


Fig. 5. The curves of mean performances of CGA and TGA with different crossover probability regarding Eq. (10).

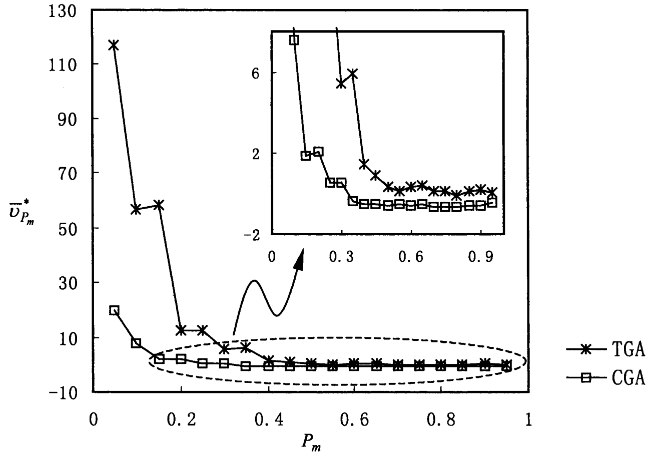


Fig. 6. The curves of mean performances of CGA and TGA with different mutation probability regarding Eq. (10).

$P_c = 0.15$ and $P_m = 0.75$ are the best parameters for CGA.

Thus, when compare the performance of CGA with that of TGA, for Eq. (9), CGA takes $P_c = 0.35$, $P_m = 0.1$, $n_p = 400$ and $t_m = 300$, and TGA takes $P_c = 0.55$, $P_m = 0.7$, $n_p = 400$ and $t_m = 300$; for Eq. (10), CGA takes $P_c = 0.15$, $P_m = 0.75$, $n_p = 100$ and $t_m = 100$, and TGA takes $P_c = 0.95$, $P_m = 0.8$, $n_p = 100$ and $t_m = 100$. And, in every test, CGA and TGA have the same random initial population. To illustrate the performance of optimization approach with generation clearly, two well-known criteria were used as indicated in the following. The off-line performance was defined as:

$$v^{off}(T) = \frac{1}{T}\sum_{r=1}^{T} v^*(r), \quad (13)$$

in which $v^{off}(T)$ denotes the off-line performance of optimization approach, T is the number of current generation, and $v^*(t)$ is the best value of object function obtained up to tth generation. In the above cases, $v^*(t)$ is the best value of $f_1(x)$ or $f_2(x)$ obtained up to tth generation. The on-line performance was defined as:

$$v^{on}(T) = \frac{1}{n_p}\sum_{i=1}^{n_p} v_i(T), \quad (14)$$

in which $v^{on}(T)$ denotes the on-line performance of optimization approach, n_p is the number of the individuals of population, and $v_i(T)$ is the value of object function of ith individual in current generation. In the above cases, $v_i(T)$ is the value of $f_1(x)$ or $f_2(x)$ of ith individual in generation under consideration.

It can be seen from the above definitions that the off-line performance demonstrates the track of the change of optimum value based upon all generations and the on-line performance demonstrates the track of the change of mean value corresponding to every generation.

To reduce the influence of incidental variation, the total number of tests executed for searching the optimal values of Eqs. (9) and (10) are 30, respectively, and different random initial population is employed for every test. Then, the mean off-line performances and the mean on-line performances were calculated and shown in Figs. 7–10. It can be seen from Figs. 7–10 that the performance of CGA is better than that of TGA. Moreover, in the 30 tests on Eq. (9) there are four tests reached the global optimal value when CGA was applied, while only two tests reached the optimal value when TGA was used; and in the 30 tests on Eq. (10) there are 14 tests reached the global optimal value when CGA was applied, while only six tests reached the optimal value when TGA was used.

To illustrate the superior performance of CGA further, the well-known conventional optimization approach, named as steepest descent method, is applied to search the optimal values of Eqs. (9) and (10).

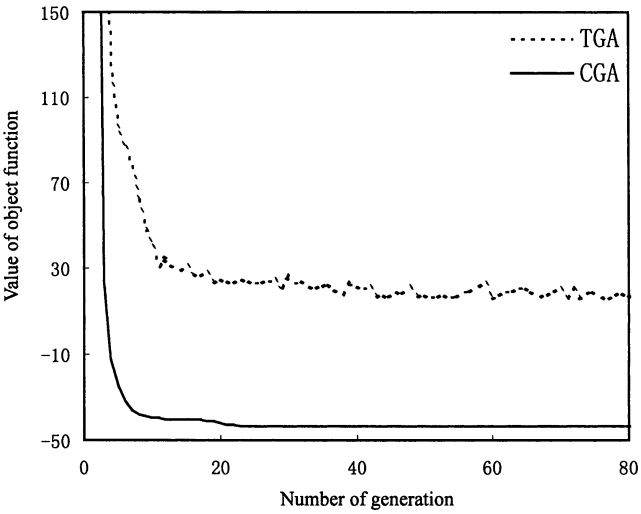


Fig. 7. Mean on-line performance regarding Eq. (9) by CGA vs. TGA.

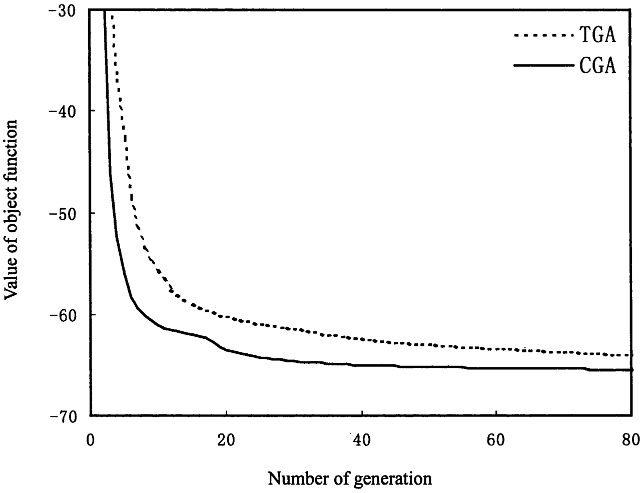


Fig. 8. Mean off-line performance regarding Eq. (9) by CGA vs. TGA.

The total number of the application of steepest descent method for Eqs. (9) and (10) are 300, respectively. Each application takes different random initial solution. In the 300 applications for Eq. (9), the global optimal value is not obtained by steepest descent method. And, in the 300 applications for Eq. (10), there is only one application reached the global optimal value. It is well known that the function to be minimized and the constraints must both be continuous when steepest descent method is applied. What's more, it is shown from above tests that steepest descent method may only give local solutions and the obtained solution depends on the initial solution completely. Steepest descent method is one representative conventional optimization approach. Many modified conventional optimization approaches are proposed, but the problem that the

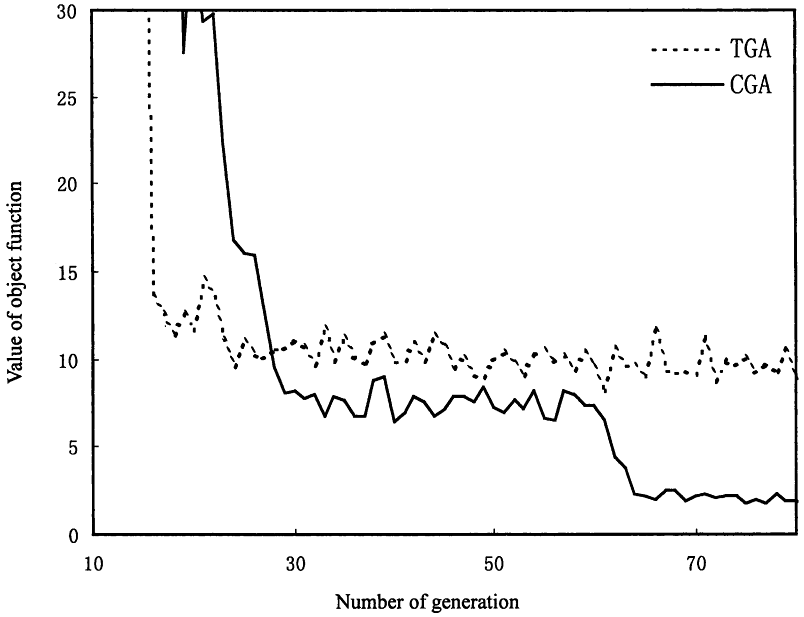


Fig. 9. Mean on-line performance regarding Eq. (10) by CGA vs. TGA.

obtained solution depends on the initial solution is not solved.

3. A practical optimization problem

The optimization of the operational conditions of a twin-reactor AHI process as shown in Fig. 11 was taken as a practical sample problem for testing the performance of the above proposed method. Considering that the problem was interpreted through a series of data, RBF-PLS method, introduced by Walczak and Massart (1996), was taken as the method for modeling this kind of problem which does not have a clearly defined relationship between the independent and dependent variables.

3.1. Aromatic hydrocarbon isomerization process

It was known that there are many factors influencing the conversion of ethyl benzene in AHI process. The reaction temperature (T, °C), the reaction pressure (P,

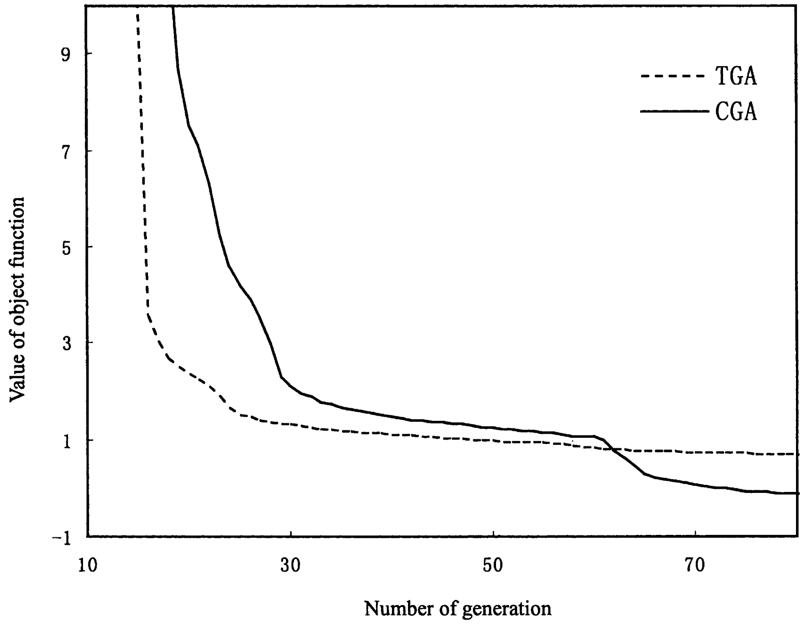


Fig. 10. Mean off-line performance regarding Eq. (10) by CGA vs. TGA.

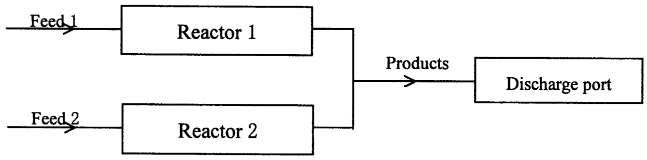


Fig. 11. Flow diagram of AHI process.

MPa), the hydrogen/oil ratio ($C_{\mathrm{H_2/Oil}}$, mol/mol), the liquid phase space velocity (v_{SV}, kg/h), the age of catalyst (τ, kg feed processed/kg catalyst), the weight percentage of m-xylene in feed (C_{MX}, %), the weight percentage of o-xylene in feed (C_{OX}, %), the weight percentage of p-xylene in feed ($\mathrm{C_{PX}}$, %), and the weight percentage of ethyl benzene in feed (C_{EB}, %) are considered as the major factors influencing the conversion of ethyl benzene. The conversion ratio of ethyl benzene in AHI is denoted by C in [%].

The object under our consideration consists of two AHI reactors parallelly connected and the material flow is shown in Fig. 11. The two reactors treat feeds of different compositions under different operating conditions, respectively. Thus, the independent variables include all operation parameters of the two reactors, and the dependent variable is the overall conversion ratio of the two reactors. Finally, the number of the independent variables equals 18, and the number of the dependent variable is 1. The number of the observation vectors in the data set used for modeling equals 100.

3.2. Modeling AHI process by radial basis functions-partial least squares (RBF-PLS)

Assume that the practical observation vector p consists of n independent variables and l dependent variables, the independent variable vector x consists of n independent variables, the dependent variable vector y consists of l dependent variables, and the data set available for modeling consists of m observation vectors. Thus, the dimension of the independent variable matrix $\boldsymbol{X}$ is $m \times n$, and that of the dependent variable matrix $\boldsymbol{Y}$ is $m \times l$. In the RBF-PLS method, RBFs are used to carry out the nonlinear transformation of $\boldsymbol{X}$ to form an activation matrix $\boldsymbol{X}_A$. The elements of $\boldsymbol{X}_A$ are defined as:

$$a_{ij} = r_j(x_i), \quad i, j = 1, 2, \ldots, m, \tag{15}$$

where x_i is a vector consisting the values of independent variables taken from the ith observation, a_{ij} is the element of $\boldsymbol{X}_A$ at the ith row and the jth column, r_j is the jth RBF.

The most commonly used RBF is Gaussian function, which takes the following form:

$$a_{ij} = \exp(-\|c_j - x_i\|^2/\sigma_j^2) \quad i, j = 1, 2, \ldots, m, \tag{16}$$

in which $\|\ \|$ is a norm and denotes the Euclidean distance when the argument is a difference of two vectors, c_j and σ_j are two parameters of jth Gaussian function, the center and the width. The parameter c_j of jth Gaussian function is calculated by:

$$c_j = x_j, \quad j = 1, 2, \ldots, m,$$

and the elements of the parameter σ_j of jth Gaussian function is calculated by:

$$\sigma_{j1} = \sigma_{j2} = \cdots = \sigma_{jn} = \frac{e}{m}\sum_{i=1}^{m} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|,$$
$$j = 1, 2, \ldots, m,$$

in which e is a constant assigned a value $e > 0$. Thus, the diagonal elements of activation matrix $\boldsymbol{X}_A$ have the value 1. It is clear that the number of RBFs is m, and the RBFs themselves are vector functions, the dimensions of which are all n.

Now the PLS procedure will be applied. The goal is to minimize the variance of the prediction, while maximizing the covariance of $\boldsymbol{X}_A$ and $\boldsymbol{Y}$. As for our case of applying PLS to the matrices $\boldsymbol{X}_A$ and $\boldsymbol{Y}$, the linear PLS model can be set up as:

$$\boldsymbol{Y} = \boldsymbol{TC} + \boldsymbol{E} = \boldsymbol{X}_A\boldsymbol{UC} + \boldsymbol{E}, \tag{17}$$

in which $\boldsymbol{T}$ is the low-dimensional score matrix of $\boldsymbol{X}_A$ with the dimension of $m \times n_T$, $\boldsymbol{C}$ represents the regression coefficient matrix with the dimension of $n_T \times l$, $\boldsymbol{U}$ is the transformation matrix of $\boldsymbol{X}_A$ with the dimension of $n \times n_T$, and $\boldsymbol{E}$ is the residuals matrix with the dimension of $m \times l$. Due to the fact that $\boldsymbol{T}$ are the linear combination of the Gaussian functions (i.e. the row vectors of $\boldsymbol{X}_A$) that will maximize the covariance between $\boldsymbol{X}_A$ and $\boldsymbol{Y}$, n_T plays an important role in the predicting ability of RBF-PLS model. It can be determined by the well-known cross-validated method. In this way the matrices $\boldsymbol{U}$ and $\boldsymbol{C}$ are obtained, and thus the RBF-PLS model can be obtained and used for prediction or other purposes.

In general, before calculating the activation matrix, the variables of the practical observation vectors are always scaled to the range [0, 1]. Assume that a set of independent variable vectors beyond the data used in modeling is chosen to form a matrix $\boldsymbol{X}_p$ for prediction, and the corresponding dependent matrix $\boldsymbol{Y}_p$ has not been obtained or unknown. X_p will be pre-processed in an identical manner as $\boldsymbol{X}$ of the data set used for modeling, i.e. its variables are scaled (using the minimum and the maximum values of $\boldsymbol{X}$) and the activation matrix $\boldsymbol{X}_{Ap}$ of $\boldsymbol{X}_p$ is calculated keeping the values of centers and widths of Gaussian functions. Finally, the predicted $\boldsymbol{Y}_p$ is calculated as:

$$\boldsymbol{Y}_p = \boldsymbol{X}_{Ap}\boldsymbol{UC}. \tag{18}$$

RBF-PLS method is employed to model AHI process through the data set. The number of observation vectors in the data set used for modeling is m. The optimal value for n_T used in the score matrix $\boldsymbol{T}$ can be determined by the leave-one-out cross-validated method. The procedure is illustrated as follows.

1) Assign the initial value $n_T = 1$.
2) Carry out the leave-one-out cross-validated method, namely, every time leave one observation vector out for validating model, the other vectors in the data set are used for modeling, and thus each observation vector is used as vector for validation one time and used for modeling $m-1$ times. Execute the PLS analysis according to Eqs. (17) and (18) and calculate the mean absolute relative deviation (MARD_{n_T}) under the current n_T by:

$$\mathrm{MARD}_{n_T} = \frac{1}{m}\sum_{i=1}^{m} |(y_{pi} - y_i)/y_i|, \tag{19}$$

   in which y_i is the value of the ith observed dependent variable, and y_{pi} is the predicted value of y_i by RBF-PLS model.
3) As long as n_T is smaller than m, set $n_T = n_T + 1$ and repeat step 2.
4) Stop the procedure until $n_T \geq m$.

Fig. 12 shows the relationship between MARD_{n_T} and n_T. It can be seen from Fig. 12 that MARD_{n_T} has the minimum value 0.070472 when $n_T = 18$. It means that the RBF-PLS model has the best predicting ability at this point. Thus, the optimal n_T is 18. As soon as the optimal n_T is determined, the other parameters of RBF-PLS model, namely, c_j, σ_j, $\boldsymbol{U}$ and $\boldsymbol{C}$, can be determined through the data set, and the appropriate RBF-PLS model of AHI process is obtained.

3.3. Optimize the operating conditions of AHI process

In this section, CGA was used to optimize the operating conditions of AHI process based on the RBF-PLS model. The weight percentages of components in feed and liquid phase space velocity are determined by the upstream process, and the age of catalyst is evaluated by the total amount of feed processed starting from the day loading catalyst. Therefore, the above mentioned variables are uncontrollable independent variables. The rest variables, i.e. the reaction temperature, the reaction pressure and hydrogen/oil ratio, can be considered as the controllable operating variables. The age of catalyst can be evaluated by:

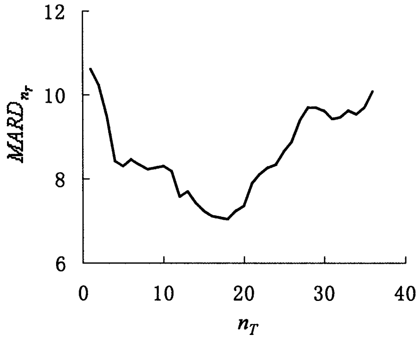


Fig. 12. The relationship between MARD_{n_T} and n_T.

$$\tau_i^{d+1} = \tau_i^d + (\tau_i^d - \tau_i^{d-1}), \quad i = 1,\ 2,$$

in which the subscript i denotes the code of reactor; the superscript d denotes the current day; $d-1$, 1 day before the current day; and d+1 the next day after the current day. Assuming that $C_{MX,1}^{d+1}$, $C_{MX,2}^{d+1}$, $C_{OX,1}^{d+1}$, $C_{OX,2}^{d+1}$, $C_{PX,1}^{d+1}$, $C_{PX,2}^{d+1}$, $C_{EB,1}^{d+1}$, $C_{EB,2}^{d+1}$, $\upsilon_{SV,1}^{d+1}$ and $\upsilon_{SV,2}^{d+1}$ are little different from $C_{MX,1}^{d}$, $C_{MX,2}^{d}$, $C_{OX,1}^{d}$, $C_{OX,2}^{d}$, $C_{PX,1}^{d}$, $C_{PX,2}^{d}$, $C_{EB,1}^{d}$, $C_{EB,2}^{d}$, $\upsilon_{SV,1}^{d}$ and $\upsilon_{SV,2}^{d}$ in continuous processing, so the values of $C_{MX,1}^{d+1}$, $C_{MX,2}^{d+1}$, $C_{OX,1}^{d+1}$, $C_{OX,2}^{d+1}$, $C_{PX,1}^{d+1}$, $C_{PX,2}^{d+1}$, $C_{EB,1}^{d+1}$, $C_{EB,2}^{d+1}$, $\upsilon_{SV,1}^{d+1}$ and $\upsilon_{SV,2}^{d+1}$ can be approximated by those of $C_{MX,1}^{d}$, $C_{MX,2}^{d}$, $C_{OX,1}^{d}$, $C_{OX,2}^{d}$, $C_{PX,1}^{d}$, $C_{PX,2}^{d}$, $C_{EB,1}^{d}$, $C_{EB,2}^{d}$, $\upsilon_{SV,1}^{d}$ and $\upsilon_{SV,2}^{d}$, respectively.

After $C_{MX,1}^{d+1}$, $C_{MX,2}^{d+1}$, $C_{OX,1}^{d+1}$, $C_{OX,2}^{d+1}$, $C_{PX,1}^{d+1}$, $C_{PX,2}^{d+1}$, $C_{EB,1}^{d+1}$, $C_{EB,2}^{d+1}$, $\upsilon_{SV,1}^{d+1}$, $\upsilon_{SV,2}^{d+1}$, τ_1^{d+1} and τ_2^{d+1} were assigned, CGA were employed to predict the optimized operating conditions, namely: T_1^{d+1}, T_2^{d+1}, P_1^{d+1}, P_2^{d+1}, $C_{\mathrm{H_2/Oil,1}}^{d+1}$ and $C_{\mathrm{H_2/Oil,2}}^{d+1}$ based on the RBF-PLS model for the next day. The operation parameters of CGA were taken as $n_P = 200$, $P_c = 1$, $P_m = 0.2$ and $t_m = 200$. Finally, the optimized conversion ratio $C = 29.63\%$ was calculated while the corresponding operation parameters T_1^{d+1}, T_2^{d+1}, P_1^{d+1}, P_2^{d+1}, $C_{\mathrm{H_2/Oil,1}}^{d+1}$ and $C_{\mathrm{H_2/Oil,2}}^{d+1}$ were evaluated as 392.1, 400.9 °C, 1.30, 1.28 MPa, 9.38 and 5.08 mol/mol, respectively. The optimized conversion ratio is 30.87% higher than the mean conversion ratio of the practically recorded data 22.64% and is also 28.27% higher than the current conversion ratio 23.1%. The calculated optimal values might be somewhat different from real optimal conditions, but it can be used as a predictive direction for finding the better operation parameters to improve the conversion ratio of the next day. In the next day, the practice data will be collected and added to the data set for modeling. The new model from the data set will be set up by RBF-PLS method again and CGA will be applied to predict the optimized operating conditions based on the new model for the next day of the next day subsequently. According to the methodology, the global optimal operating conditions of the two reactors will be obtained finally.

4. Conclusions

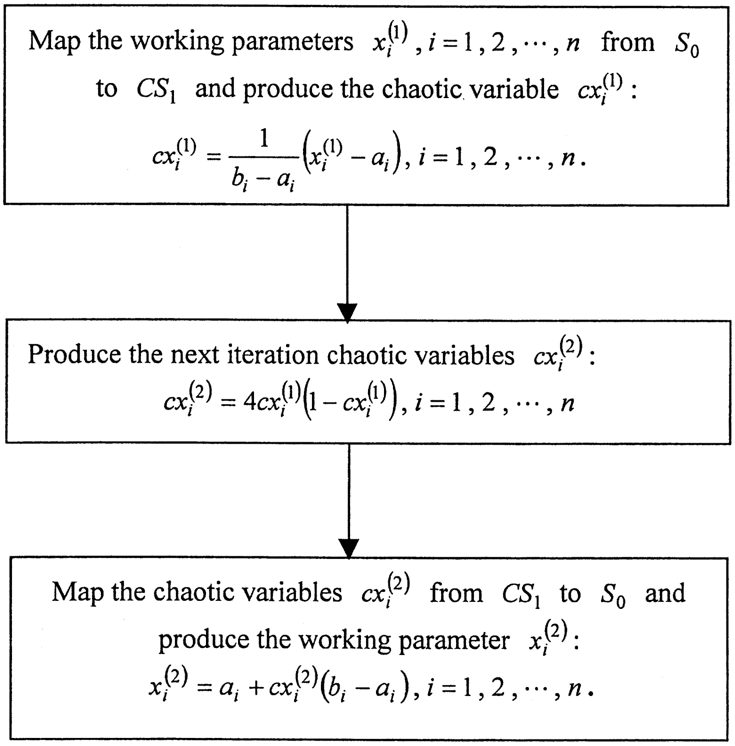
A CGA taking the advantage of chaotic variables was proposed to make the search of optimal values more effective and faster. The two experiments of analytical functions demonstrated the proposed CGA was more effective on searching optimum than TGA. Moreover, a novel and global approach including RBF-PLS and CGA was presented and validated for AHI process. Firstly, RBF-PLS was used to represent AHI model through the practical observation data. Secondly, CGA was applied to optimizing the operating conditions, and the satisfactory results were obtained.

Acknowledgements

The project was supported by the National Natural Science Foundation (20276063) of China.

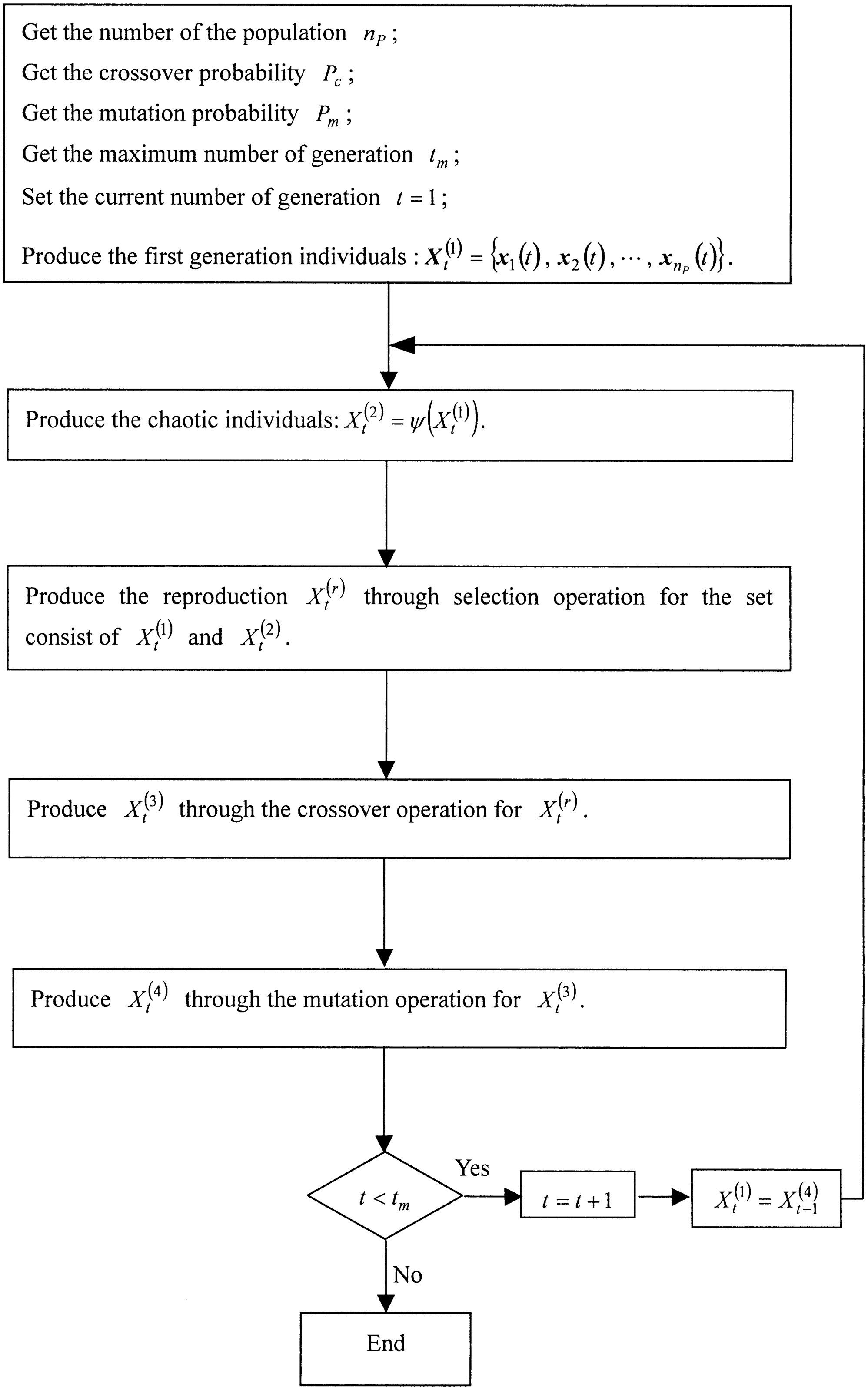
Appendix A: Chaotic mapping operator

Chaotic mapping operator


Map the working parameters $x_i^{(1)}, i=1,2,\cdots,n$ from S_0 to CS_1 and produce the chaotic variable $cx_i^{(1)}$:
$cx_i^{(1)} = \frac{1}{b_i - a_i}\left(x_i^{(1)} - a_i\right), i=1,2,\cdots,n.$
Produce the next iteration chaotic variables $cx_i^{(2)}$:
$cx_i^{(2)} = 4cx_i^{(1)}\left(1 - cx_i^{(1)}\right), i=1,2,\cdots,n$
Map the chaotic variables $cx_i^{(2)}$ from CS_1 to S_0 and produce the working parameter $x_i^{(2)}$:
$x_i^{(2)} = a_i + cx_i^{(2)}\left(b_i - a_i\right), i=1,2,\cdots,n.$

Appendix B: Chaos-genetic algorithm

Chaos-genetic algorithm


Get the number of the population n_P;
Get the crossover probability P_c;
Get the mutation probability P_m;
Get the maximum number of generation t_m;
Set the current number of generation $t = 1$;
Produce the first generation individuals : $\boldsymbol{X}_t^{(1)} = \{\boldsymbol{x}_1(t), \boldsymbol{x}_2(t), \cdots, \boldsymbol{x}_{n_P}(t)\}$.
Produce the chaotic individuals: $X_t^{(2)} = \psi\left(X_t^{(1)}\right)$.
Produce the reproduction $X_t^{(r)}$ through selection operation for the set consist of $X_t^{(1)}$ and $X_t^{(2)}$.
Produce $X_t^{(3)}$ through the crossover operation for $X_t^{(r)}$.
Produce $X_t^{(4)}$ through the mutation operation for $X_t^{(3)}$.
$t < t_m$
Yes
$t = t + 1$
$X_t^{(1)} = X_{t-1}^{(4)}$
No
End

References

Balland, L., Estel, L., Cosmao, J. M., & Mouhab, N. (2000). A genetic algorithm with decimal coding for the estimation of kinetic and energetic parameters. *Chemometrics and Intelligent Laboratory Systems 50*, 121–135.

Bountis, T. (1995). Fundamental concepts of classical chaos. Part I. *Information Dynamics and Open Systems*, *3* (23) 23–96.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning*. New York: Addison-Wesley.

Harris, S. D., Elliott, L., Ingham, D. B., Pourkashanian, M., & Wilson, C. W. (2000). The optimization of reaction rate parameters for chemical kinetic modeling of combustion using genetic algorithms. *Computing Methods and Applied Mechanical Engineering 190*, 1065–1090.

Holland, J. H. (1975). *Adaptation in natural and artificial systems*. The University of Michigan Press.

Hugget, A., Sébastian, P., & Nadeau, J. P. (1999). Global optimization of a dryer by using neural networks and genetic algorithms. *American Institute of Chemical Engineering Journal*, *45* (6) 1227–1238.

Lavine, B. K., Davidson, C. E., & Moores, A. J. (2002). Innovative genetic algorithms for chemoinformatics. *Chemometrics and Intelligent Laboratory Systems 60*, 161–171.

Leonardo, B. H., Catherine, A. P., Serge, D., & Lue, P. (1998). Design of multipurpose batch chemical plants using a genetic algorithm. *Computers and Chemical Engineering 22*, s777–s780.

Li, B., & Jiang, W. S. (1998). Optimizing complex functions by chaos search. *Cybernetics and Systems: An International Journal 29*, 409–419.

Lorenz, E. N. (1963). Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences 20*, 130–141.

May, R. (1976). Simple mathematical models with very complicated dynamics. *Nature 261*, 459–467.

Michalewicz, Z., Janikow, C. Z., & Krawczyk, J. B. (1992). A modified genetic algorithm for optimal control problems. *Computer and Mathematical Applications 23* (12), 83–94.

Ohya, M. (1998). Complexities and their applications to characterization of chaos. *International Journal of Theoretical Physics*, *37* (1) 495–505.

Pan, Z. J., Kang, L. S., & Nie, S. X. (1996). Evolving both the topology and weights of neural networks. *Parallel Algorithm and Application 9*, 299–307.

Pongcharoen, P., Hicks, C., Braiden, P. M., & Stewardson, D. J. (2002). Determining optimum genetic algorithm parameters for scheduling the manufacturing and assembly of complex products. *International Journal of Production Economics 78*, 311–322.

Sender, J. M. (1994). *Algorithmes Génétiques et Réseaux de Neurones*. Paris: Hermès.

Upreti, S. R., & Deb, K. (1997). Optimal design of an ammonia synthesis reactor using genetic algorithms. *Computers and Chemical Engineering 21* (1), 87–92.

Walczak, B., & Massart, D. L. (1996). The radial basis functions-partial least squares approach as a hexible non-linear regression technique. *Analytica Chimica Acta 331*, 177–185.

Wright, A. H. (1991). In G. J. E. Rawlins (Ed.), *Genetic algorithms for real parameter optimization. Foundations of genetic algorithms* (pp. 205–218). California: Morgan Kaufmann.

Yun, Y. S. (2002). Genetic algorithm with fuzzy logic controller for preemptive and non-preemptive job–shop scheduling problems. *Computers and Industrial Engineering 43*, 623–644.

Zhao, W. X., Chen, D. Z., & Hu, S. X. (2000). Optimizing operating conditions based on ANN and modified gas. *Computers and Chemical Engineering 24*, 61–65.