J Glob Optim (2006) 36:517–535
DOI 10.1007/s10898-006-9024-1

ORIGINAL ARTICLE

The development of information guided evolution algorithm for global optimization

Chen-Wei Yeh · Shi-Shang Jang

Received: 3 March 2005 / Accepted: 13 March 2006 /
Published online: 27 June 2006

Abstract Evolutionary algorithm (EA) has become popular in global optimization with applications widely used in many industrial areas. However, there exists probable premature convergence problem when rugged contour situation is encountered. As to the original genetic algorithm (GA), no matter single population or multi-population cases, the ways to prevent the problem of probable premature convergence are to implement various selection methods, penalty functions and mutation approaches. This work proposes a novel approach to perform very efficient mutation to prevent from premature convergence by introducing the concept of information theory. Information-guided mutation is implemented to several variables, which are selected based on the information entropy derived in this work. The areas of search are also determined on the basis of the information amount obtained from previous searches. Several benchmark problems are solved to show the superiority of this information-guided EA. An industrial scale problem is also presented in this work.

Keywords Evolutionary algorithm · Premature convergence · Information entropy · Orthogonal design

Notation

$\mathbf{A}_j, \mathbf{B}_j$	parameters of Antoine equation of component j
C_m	index of the least discovered section of the mth selected variable
D	probable premature convergence detector
E	information entropy
E_j	information entropy of the jth variable
$\overline{E}$	threshold of information entropy
F	fitness value
$\mathbf{F}_n$	liquid flow rate of feed to nth tray

C.-W. Yeh · S.-S. Jang (✉)
Department of Chemical Engineering, National Tsing-Hua University, 101, Section 2 Kuang Fu Road, HsinChu, 30013 Taiwan, Republic of China
e-mail: ssjang@che.nthu.edu.tw

$\mathbf{Fv}_{n-1}$	vapor flow rate of feed to nth tray
f	objective function
G_j	set of the jth variable of I
$\mathbf{H}_{n-1}$	heat capacity of vapor entering nth tray
$\mathbf{Hf}_{n-1}$	heat capacity of vapor feed to nth tray
$\mathbf{h}_n$	heat capacity of liquid hold up of nth tray
$\mathbf{hf}_n$	heat capacity of liquid feed of nth tray
I	set of individuals in the population
L_j^0	lower bound of variable j
L_j^h	lower bound of section h of variable j
$L_{\omega_m}^{C_m}$	lower bound of the least discovered section of the mth selected variable
$L_M\left(Q^K\right)$	orthogonal array with K variables and Q levels
L'_m	modified lower bound of the mth selected variable for modifying premature individuals
$\mathbf{L}_{n+1}$	liquid flow rate entering nth tray
M_I	number of individuals have been sampled
M_{p}	total number of the individuals for each population
$\mathbf{ME}_{nj}$	Murphreee vapor efficiency for jth component on nth tray
N	number of variables
O_i	position in the ordered population of individual i
$P(z)$	the probability of the event z occurring
$P_{h,j}$	probability of the jth variable of individuals located on the hth in previous search
$P_{C_m,m}$	probability of the least discovered section C_m of the mth selected variable
$\mathbf{P}_{nj}^0$	vapor pressure of pure component j of nth tray
R	numbers of sections of a variable
S	solution space
S_j	solution space of the jth variable
S'_m	modified subspace of the mth selected variable for modifying premature individuals
S_n	liquid flow rate of stream leaving from nth tray
$\mathbf{Sv}_{n-1}$	vapor flow rate of stream leaving from nth tray
$s_{h,j}$	subspace of section h of variable j
s_{C_m,ω_m}	subspace of the least discovered section of the mth selected variable
$\mathbf{T}_n$	temperature of nth tray
U_j^0	upper bound of variable j
U_j^h	upper bound of section h of variable j
$U_{Y_m}^{C_m}$	upper bound of the least discovered section of the mth selected variable
U'_m	modified upper bound of the mth selected variable for modifying premature individuals
$\mathbf{V}_n$	vapor flow rate leaving nth tray
$\mathbf{W}_n$	liquid holdup on nth tray
X	variable set
X_i	set of the ith individual of I
$\mathbf{x}_{i,j}$	jth variable of the ith individual
$\mathbf{x}_{nj}$	actual composition of liquid leaving nth tray
Y	set of selected variable

$\mathbf{y}^*_{nj}$	composition of vapor in phase equilibrium with liquid on nth tray with composition w_{nj}
$\mathbf{y}_{nj}$	actual composition of vapor leaving nth tray
$\mathbf{y}^{\mathrm{T}}_{n-1,j}$	actual composition of vapor entering nth tray

Greek Symbols

$\alpha_{q,m}$	qth design of the mth selected variable of an orthogonal array
γ_m	random number for modifying the boundary of the least discovered section of the mth selected variable
δ	threshold of probable premature convergence detector
σ	scaling factor for adjusting the threshold
$\theta_{h,j}$	numbers of sampling of the jth variable that located on the hth section
ω_m	index of the mth selected variable

1 Introduction

Evolutionary algorithm (EA) has become one of the major trends in the so-called evolutionary computing (EC). Many applications can be found from molecular level [1, 10] to plant scale problems [9, 21]. Further, emerging areas that require global optimization are becoming more and more challenging for developing efficient ECs. For instance, many packages [3, 13] are designed to implement genetic algorithm (GA) to solve the placements of peptide docking. On the other hand, chemical engineers also widely use GA to determine the chemical reaction mechanisms [24]. Due to much increased application cases involving more and more computation cost of these complicated utilizations, it becomes important to improve the efficiency of evolution algorithms. Furthermore, accurate evaluation of global optimum is also essential particularly in the area of molecular design [10]. The objective of this research is to develop an information mutation rule for the EA to prevent the occurrence of the probable premature convergence problem from happening. A novel algorithm is further developed to improve the efficiency of mutation by finding more informative mutation range of each variable.

Global optimization can be widely applied in many areas. Evolutionary algorithms have a powerful and widely applicable stochastic search and optimization techniques. Many problems that are very difficult to solve by conventional techniques (optimization problems subject to complex constraints, NP-hard problems . . .etc.) can be solved by EAs [6, 8]. In real engineering application cases, problems with rugged contour involved in most practical cases are of the major concern. Regarding the problem of ruggedness, α-BB, a very advanced extension of Branch and Bound, has been the most attractive approach to be used among engineering researchers by implementing analytical local smoothing techniques. A textbook [7] on α-BB includes many applications and most of them are of industrial interest. The other stream of global optimization is the so-called stochastic method, for example, simulated annealing (SA) and EC. The difference between the above two approaches can be roughly listed in the following:

(i) The α-BB involves considerable algebraic and mathematical analysis to the problem itself, whereas the evolutionary approach does not have this problem.
(ii) The evolutionary approach basically cannot guarantee the global optimum of the problem, but the analytical approach can provide more theoretical insight into the problem.

It is clear that these two streams have their own values, and thus many extensive works have focused on both approaches. The aim of this work is to derive an information-guided mutation

algorithm such that the probable premature convergence problem in EA can be substantially improved.

Information theory was first derived by Shannon [16]. The basic spirit of information theory is to quantify the amount of information that can be obtained from sampling. It is hence more desirable to sample a point that owns more information. Based on this idea, the authors [4] derived a novel experimental design approach by implementing the so-called information free energy and an artificial neural network meta-model to minimize the number of experiments. Information theory was first implemented for GA by Tsujimura and Gen [20]. They implemented information theory to find the best selection chromosome for a traveling salesman problem (TSP). However, in this work, the problem of probable premature convergence of evolution algorithm is investigated. It is found that implementing information theory on the selections of the mutation variables is more essential. A novel approach to allocating the "rarely explored", or called "most informative", areas to perform efficient mutation is one step further derived. The simulation results show that the application of information theory is valid and useful.

Evolutionary algorithm has become quite substantial recently. The very recent development concerns a so-called multi-population approach [18, 19]. This is very important when the contour is extremely rugged and the search space is very large. By adding this concept, some probable premature convergence problems of EA can be solved. However, according to our experience, probable premature convergence can still happen in each population by itself, and thus the exchange of information among the populations does not help. The other development is to improve the stages of cross-over and initiation by implementing the concept of orthogonal array [11, 25]. In this way, the uniformity of search can be improved drastically. On the other hand, the so-called "hybrid method" of EA has also been widely implemented in many applications. The basic idea of these approaches is to add some local search approach to a "global" GA. These approaches can be either gradient-based search [21], or Nelder & Mead [24], or Soli & Wet's [15].

In this work, the body of the evolution algorithm implemented follows the standard textbook written by Gen and Cheng [8], but a novel information-based mutation approach is derived for evolution algorithm. In this novel approach, the selection of the variable to mutate is based on the total information entropy of each variable. The total performance is improved one step further by incorporating more accurate mutation adapted by local information entropy and orthogonal design. It should be noted that we only modify the mutation algorithm of the whole evolution algorithm, and this algorithm can be implemented to any type of EC. The rest of this paper is organized as follows. In Section 2, all elements of the evolution algorithm implemented by this work are summarized. In Section 3, the information-based mutation algorithm is derived. In Section 4, several benchmark problems and an industrial scale application problem are solved using both this novel approach and other existed approaches. In the last section, conclusive remarks are given.

2 Background

Consider the general optimization problem with the following form:

$$\hat{X} = \arg\min_{X} f(X)$$
$$\text{Subject to } X \in S \tag{1}$$

where f is the objective function, and X is a combination of N variables in the solution space S such that for each variable X_j of X, there exist an upper bound U_j^0 and a lower bound L_j^0. A combination of variables is called an individual in evolution algorithm. A set of individuals are called a population. Population size, M_p is the number of individuals sampled at each iteration (or generation).

Five basic operators have to be considered in classical GA: coding-decoding, fitness evaluation, selection, crossover, and mutation. In a real-parameter EA, variables are used directly to calculate the fitness values by skipping the step of coding-decoding. Each individual corresponds to one fitness value that is determined by a fitness function based on the objective values. The suitable individuals are chosen as survivals to perform crossover according to fitness value. The survive individuals are selected to be parents, and they are paired randomly to generate new individuals. With a given proper probability, variables of individuals are picked randomly to continue their mutation.

2.1 Fitness evaluation

Traditionally, the fitness function implemented in the GA is related to the result of the objective function $f(X)$ in Eq. (1) per se. Some modification [5] on the fitness function can drastically improve the performance of GA. In this work, a published code [2] from the website is adopted. It is found that the following fitness evaluation can perform much better than that to be carried out by $f(X)$ itself:

$$F(O_i) = 2 - P_\mathrm{S} + 2 \times \left(P_\mathrm{S} - 1\right) \times \frac{O_i - 1}{M_\mathrm{p} - 1}, \quad i = 1, \ldots, M_\mathrm{p} \tag{2}$$

where F is the fitness value, P_S is the selective pressure and O_i is the position in the ordered population of individual i. The objective values of each population will be sorted out first to get their position. The transformation is done to limit the reproductive range, so that no individuals generate an excessive number of offspring. In this work, the selective pressure was set as 2.

2.2 Selection

Diversity and convergence have to be considered as designing selection rules. Keeping proper diversity of populations will prevent genetic search from being terminated prematurely. But, applying too many forces to keep the diversity of populations may lead to a slow convergence. Respectively, selection focuses on the best individuals will make genetic search toward narrow regions faster, but the genetic search may be trapped on local optimal or to be terminated prematurely. Wild search is suggested in the beginning of a genetic search, and local search is suggested at the end of the genetic search.

2.3 Crossover

For a discrete recombination, single-point crossover, multi-point crossover, and uniform crossover are often used as reported in previous studies. There are many methods of crossover that have been implemented on real-parameter EA (linear crossover, a native crossover, blend crossover, simulated binary crossover, simplex crossover and fuzzy recombination). In case of linear crossover approach, intermediate recombination, line recombination and extended line recombination have been wildly used. Intermediate recombination is adapted in this study.

2.4 Mutation

New off-springs are generated after crossover, which may be modified with some probability. Non-uniform mutation is designed for fine-tuning capabilities. Directional mutation is taken to avoid the individuals jamming into a corner. Gaussian mutation is often used in a real-parameter EA. Mutation plays the main role in countering probable premature convergence. But the rate of mutation is often set low to prevent genetic search from random search. Therefore, a proper detector to detect probable premature convergence and an efficient population constructor will be very helpful to global optimization. In the next section, a novel mutation approach is presented.

3 Information entropy-guided mutation EA

In this section, the information-guided EA (IEA) proposed by this work will be discussed. Classical mutation may help EA to prevent probable premature convergence, while all individuals in a generation are the same. But the rate of mutation is often set low and may not take action immediately as the probable premature convergence happened. One of the major contributions of this work is to detect the probable premature convergence. Information-guided mutation is stimulated while all individuals are the same, or the sum distance between a randomly picked individual and others is lower than a setting threshold.

For the purpose of convenience, let's assume that M_I individuals have historically been sampled. Let's also define I to be a set of all historical individuals. Let $X_i = \left(x_{i,1}, x_{i,2}, \ldots, x_{i,N}\right)$ be a set of row elements of I, and $G_j = \left(x_{1,j}, x_{2,j}, \ldots, x_{M_I,j}\right)^{\mathrm{T}}$ be a set of column elements of I:

$$\begin{aligned} I &= \left\{x_{i,j} \,|i = 1, \ldots, M_I \text{ and } j = 1, \ldots, \ N\right\} \\ &= \{X_i \,|i = 1, \ldots, M_I\} = \left\{G_j \,|j = 1, \ldots, N\right\} \end{aligned} \tag{3}$$

where $x_{i,j}$ is the jth variable of the ith individual. Let's term X_i by an individual, G_j by the values of the jth variable of all generations.

Genetic operation may be trapped at a local optimum because of premature situation, i.e., all individuals in the population are too close. In conventional EA, mutation plays an important role to counter probable premature convergence. But it's hard to expect that small amount of mutated individual will still survive in the next generation algorithm as a low rate of mutation is set, indicating that probable premature convergence may replay very soon. Probable premature convergence can be detected by calculating the difference between individuals in population. Let's select an individual X_d in the population randomly, and then the difference of individuals between the selected one and others can be calculated. The detector D is written as follow:

$$D = \sum_{\substack{i=1 \\ i \neq d}}^{M_{\mathrm{p}}} |X_i - X_d| = \sum_{j=1}^{N} \sum_{\substack{i=1 \\ i \neq d}}^{M_{\mathrm{p}}} \left|x_{i,j} - x_{d,j}\right| \tag{4}$$

While $D = 0$, all the individuals are the same, then it means that the situation of probable premature convergence is happened or the true solution has been found. In practice, it is necessary to set a threshold δ such that $D \leq \delta$, and the approach presented in the rest of this section can be implemented.

$$D = \sum_{i=1}^{M_p} \sum_{\substack{j=1 \\ j \neq i}}^{M_p} \left\| X_i - X_j \right\| \leq \delta = \sigma \times \sum_{i=1}^{M_p} \sum_{\substack{j=1 \\ j \neq i}}^{M_p} \left\| X_i^0 - X_j^0 \right\| \tag{5}$$

where X_i^0 is an individual of the initial population I^0, and σ is a scaling factor for adjusting the threshold. In general, in case of rugged contour, there exists an average distance (D^*) among the nearest local minima and the global optimum. It is obvious that if $D > D^*$ (or $\sigma > \sigma^*$) the global minimum cannot be discovered through IEA since the algorithm cannot converged by the interruption of the above detector. On the other hand, if D is set low, then the information-guided mutation derived in the following section will be performed in a lower frequency. This will also affect the rate of convergence as discussed in the example section.

3.1 Information entropy

According to Shannon's definition of information entropy [16] for a set of variables Z, which can randomly take values z, the information entropy of the set Z is:

$$E(Z) = -\sum_{z \in Z} P(z) \log P(z) \tag{6}$$

where $P(z)$ is the probability of the event z that is occurring. If Z can only take a narrow range of values, $P(z)$, then these values are close to 1. For other values in Z, $P(z)$ is close to 0. Therefore, $E(Z)$ is close to zero. On the contrary, if Z can take a lot of different values in Z each time with a small $P(z)$, $E(Z)$ can be a large positive number. Therefore, information entropy is a measure of how random a variable is distributed. To illustrate the concept of information entropy, consider the following two random number generators that are designed to generate natural number from 1 to 4. The first generator generated $z1 = \{10,\ 10,\ 10,\ 10\}$ samples respect to nature number $Z1 = \{1,\ 2,\ 3,\ 4\}$, while the second generator generates $z2 = \{7,\ 12,\ 13,\ 8\}$ samples respect to nature number $Z2 = \{1,\ 2,\ 3,\ 4\}$. The information entropy of those two random number generators can be derived as the following:

$$E(Z1) = -\sum_{i=1}^{4} P(z1_i) \log P(z1_i) = -\left(\frac{10}{40}\log\frac{10}{40} + \frac{10}{40}\log\frac{10}{40} + \frac{10}{40}\log\frac{10}{40} + \frac{10}{40}\log\frac{10}{40}\right) \approx 1.386$$

$$E(Z2) = -\sum_{i=1}^{4} P(z2_i) \log P(z2_i) = -\left(\frac{7}{40}\log\frac{7}{40} + \frac{12}{40}\log\frac{12}{40} + \frac{13}{40}\log\frac{13}{40} + \frac{8}{40}\log\frac{8}{40}\right) \approx 1.353$$

The first random number generator gets a higher value of information entropy implying that more uniformly distributed, or more informative, numbers are generated by the first generator.

3.2 Information entropy measure to the diversity of sampling

For a real-parameter EA, information entropy can be used to measure diversity of dataset to be distributed in the solution space. Higher information entropy of a variable means a more diverse dataset of the variable to be distributed in the solution space. Consider the optimization problem (1), let's define $S = \{S_j \mid j = 1, \ldots, N\}$, with $x_{i,j} \in S_j$ and $L_i^0 < x_{i,j} \leq U_j^0$. Each solution space S_j is equally divided into R sections as shown in Fig. 1. Let

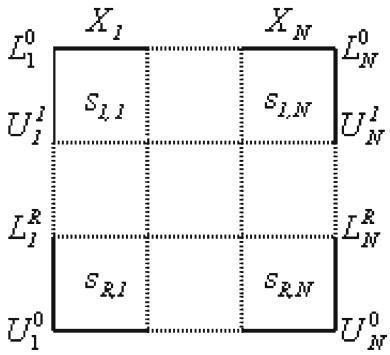

Fig. 1 Quantization of solution space

$$S_j = \left\{ s_{h,j} | h = 1, \ldots, R \right\}, \text{ and } s_{h,j} = \left[L_j^h, U_j^h \right] \tag{7}$$

$$\text{where} \begin{cases} L_j^h = L_j^0 + \frac{(h-1)}{R} \left(U_j^0 - L_j^0 \right) \\ U_j^h = U_j^0 - \frac{(R-j)}{R} \left(U_j^0 - L_j^0 \right) \end{cases}, h = 1, \ldots, R \text{ and } j = 1, \ldots, N$$

Let's define the probability that $x_{i,j} \in s_{h,j}$ in the previous search as the following: Assume X has been sampled Q_t times in the previous search, historically takes the values $\theta_{h,j}$ times in one of its subspace $s_{h,j} = \left[L_j^h, U_j^h \right]$; and denote the probability for the jth variable to be sampled in the hth subspace $s_{h,j}$ such that $L_j^h < x_{i,j} \leq U_j^h$, then

$$\theta_{h,j} = n \left(x_{i,j} \left| x_{i,j} \in s_{h,j} \right. \right) \tag{8}$$

$$Q_t = n(G_1) = n(G_2) = \cdots = n(G_N) \tag{9}$$

$$P_{h,j} = P \left(x_{i,j} \in s_{h,j} | i = 1, \ldots, Q_t, h = 1, \ldots, R, j = 1, \ldots, N \right) = \frac{\theta_{h,j}}{Q_t} \tag{10}$$

where n is an operator to number the variable set, $\theta_{h,j}$ are numbers of sampling of the jth variable that located on its hth section, and all the N variables are sampled Q_t times. P_{hj} is the probability of the jth variable G_j sampled at the hth section. Thus, the total information entropy of the jth variable is:

$$E_j = E(G_j) = -\sum_{h=1}^{R} P_{h,j} \times \ln \left(P_{h,j} \right) \tag{11}$$

As the information entropy of each variable $E = \{E_1, E_2, \ldots, E_N\}$ is calculated, the diversity of each variable G_j is measured. Variables with lower information entropy will be selected to generate new individuals as the following:

$$\Omega = \left\{ (\omega_1, \omega_2, \ldots, \omega_K) \left| E_{\omega_m} \leq \overline{E}, \quad \omega_m = 1, \ldots, N, m = 1, \ldots, K \right. \right\} \tag{12}$$

where $\overline{E}$ is a threshold to be adjusted at each generation. For instance, at the nth generation, it can be set:

$$\overline{E} = E_{\max} \mathrm{e}^{-n/N_{\max}} \tag{13}$$

where $N_{\max}$ is the assumed maximum generation of the whole search. Another more convenient approach is to choose fixed K variables from the variable set by setting $K = \lfloor 0.3 \times N \rfloor$, such that:

$$\Omega = \left\{(\omega_1, \omega_2, \ldots, \omega_K) \,|E_{\omega_1} \leq \cdots \leq E_{\omega_K} \leq \cdots \leq E_{\omega_m}, \omega_m = 1, \ldots, N, m = 1, \ldots, K\right\} \quad (14)$$

i.e., one may choose 30% from the total variables with the lowest information entropy to perform mutation as described below.

3.3 The information-based orthogonal design

Leung and Wang [11] applied a special class of orthogonal arrays $L_M(Q^K)$ to EA for generating both initial data and crossover, that $M = Q^J$ is the size of population, J is a positive integer satisfying

$$K = \frac{Q^J - 1}{Q - 1} \quad (15)$$

Applying orthogonal design to modify the premature individuals in the K subspaces can be treaded as a K factors experimental design problem. The design level Q is determined by the size of population and the number of selected variables.

While the K variables are determined to be renewed for the premature individuals, the change will be focused on the least discovered section C_m of each selected variable Y_m of the individuals and an orthogonal design will be applied on the subspace s_{C_m, Y_m}, where

$$C = \left\{\mathrm{C}_m | \theta_{\mathrm{C}_m, m} = \min_h \left(\theta_{h,m}\right),\ h = 1, \ldots, R,\ m = 1, \ldots, K\right\} \quad (16)$$

While applying the orthogonal array $\beta = \left\{\alpha_{q,m} \,|q = 1, \ldots, Q,\ m = 1, \ldots, K\right\}$ to modify the premature individuals, the boundary of the design space will be randomly bounded inside the subspace s_{C_m, ω_m} as S'_m,

$$S'_m = \left[L'_m, U'_m\right] = \left[L_{\omega_m}^{C_m} + \gamma_{m,1} \times \frac{\left(U_{\omega_m}^{C_m} - L_{\omega_m}^{C_m}\right)}{R}, L_{\omega_m}^{C_m} + \gamma_{m,2} \times \frac{\left(U_{\omega_m}^{C_m} - L_{\omega_m}^{C_m}\right)}{R}\right] \quad (17)$$

where $\gamma_{m,1}$, $\gamma_{m,2}$ are random numbers, and $0 \leq \gamma_{m,1} < \gamma_{m,2} \leq 1, m = 1, \ldots, K$. The new subspace S'_m of each selected variable will be quantized to Q levels,

$$\beta'_m = \left(\alpha_{1,m}, \alpha_{2,m}, \ldots, \alpha_{Q,m}\right) \quad (18)$$

where $\alpha_{q,m} = L'_m + (m-1) \times \frac{(U'_m - L'_m)}{Q}$, $q = 1, \ldots, Q$ and $m = 1, \ldots, K$. Finally, the replaced elements can be expressed as follows:

$$x'_{i,m} = \alpha_{\beta_{i,m}, m} \quad i = 1, \ldots, M \quad \text{and } m = 1, \ldots, K \quad (19)$$

3.4 The algorithm

The flowchart of information-entropy guided EA is shown in Fig. 2. Besides the mutation section, we modified the well-published GA code [2] and the following steps are summarized for applying our proposed approach:

Step 1: Initializing the population randomly
The set of initial individuals I^0 will be generated in the solution space randomly. Let the size of population be M_p, and the numbers of variable be N, the initial population will be an $M_\mathrm{p} \times N$ array $I^0_{M_\mathrm{p} \times N}$.

Step 2: Calculating objective values of individuals and check convergence. The individuals of each population will be calculated to get their own fitness values.

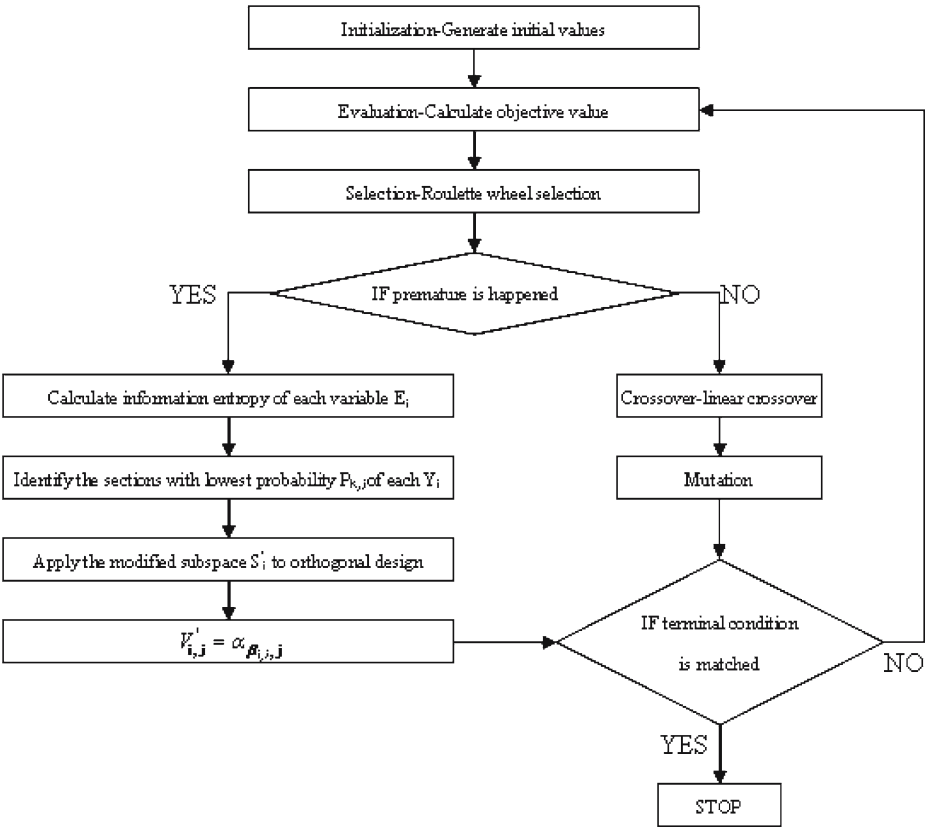

Fig. 2 The flowchart of Information based genetic algorithm

Step 3: Roulette-Wheel selection
Roulette-Wheel selection will be performed to select the survivals.

Step 4: Detecting probable premature convergence
The difference between individuals of each population will be calculated to check the situation of probable premature convergence. If the sum of absolute difference is less than a threshold, then there is probable premature convergence. If the situation of probable premature convergence is not detected, go to Step 5, else go to Step 6.

Step 5: Acquiring linear crossover. Go to Step 7.

Step 6: Performing information-based mutation as described in the previous section.

(i) Calculating information entropy of each variable E_j
(ii) Identifying variables with lower information entropy ω_m
(iii) Identifying the lowest $P_{h,m}$ of each ω_m
(iv) Modifying the boundary of subspace s_{C_m,ω_m}
(v) Applying the modified subspace to orthogonal design

Step 7: Checking terminal condition, if terminal conditions are matched, stop iteration, else go to Step 2.

The original code acquired from the web site [2] is in a more advanced structure than that described on many reports in regard to the EA. Multi-population is also available in their program. In this work, only mutation part is furnished to make local search available.

4 Case studies

Numerical experiments are carried out to show the validity of our theory developed in the previous section. Three popular benchmark problems and one industrial scale numerical example are solved. The size of population for our testing is 32, Roulette-wheel selection is adopt, the rate of selection is 80%, the mutate rate is 5%, and the terminal condition is the number of maximum function evaluations of each benchmark problem.

4.1 The benchmark problems

The well-known three benchmark problems are listed below:

(1) Griewank's Function:

$$f_1 = \frac{1}{4000}\sum_{i=1}^{N} x_i^2 - \prod_{i=1}^{N}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \tag{20}$$

Solution space: $S = [-600, 600]^{30}$

(2) Ackley's Function:

$$f_2 = -20 \times \exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + \mathrm{e} \tag{21}$$

Solution space: $S = [-32, 32]^{30}$

(3) Generalized Schwefel's Function:

$$f_3 = \sum_{i=1}^{n}\left(-x_i \sin\sqrt{|x_i|}\right) \tag{22}$$

Solution space: $S = [-500, 500]^{30}$.

The comparisons of the performances of the algorithms for the above three cases based on the mean and standard deviation with a maximum number of function calls, for instance, Griewank's function: 128,000, Ackley's & Generalized Schwefel's function : 512,000, are shown in Table 1. The function calls appeared in Table 1 for each algorithm is for the first time that a final objective value is hit. In Table 1, all three cases are repeated by 30 times in order to obtain statistically meaningful results. Let's term the approach proposed in this work as IEA and MIEA (multi-population information-guided EA) in comparison with CEA (conventional EA [22]) and MCEA (multi-population conventional GA [14]). The information-guided approach not only improves a modified single population EA, it also finds the same contribution for all cases of multi-population as shown in Table 1. Note that in Table 1, we also list the benchmark result found from the literature [17] using a single population GA, which is even worse than the CEA listed in this paper. This achievement is due to the adoption of modified fitness function (see Eq. 2) by our code [5]. Figure 3 gives the histories of the best individual, that decreasing of the objective values as a function of number of function

Table 1 Comparison between CGA, M-CEA, M-IEA, CEA and IEA

Function		CGA [14]	M-CEA	M-IEA	CEA	IEA
	Mean	1.258	0.01353	0.006789	0.02048	0.0177
f_1	STD	1.657×10^{-2}	1.0948×10^{-2}	6.2364×10^{-3}	1.9975×10^{-2}	1.0573×10^{-2}
	Function call	346971	128000	93632	128000	113408
	Mean	2.697	6.805×10^{-5}	3.047×10^{-5}	6.815×10^{-5}	5.389×10^{-5}
f_2	STD	5.668×10^{-2}	7.0702×10^{-6}	5.3118×10^{-6}	5.4525×10^{-6}	8.1342×10^{-6}
	Function call	336,481	175,104	512,000	104,448	512,000
	Mean	−8444.7583	−12569.48766	−12569.4866	−12563.5647	−12569.4866
f_3	STD	65.7326	4.1632×10^{-8}	3.4155×10^{-8}	26.4836	4.3602×10^{-8}
	Function call	458,653	248,320	217,088	512,000	499,200

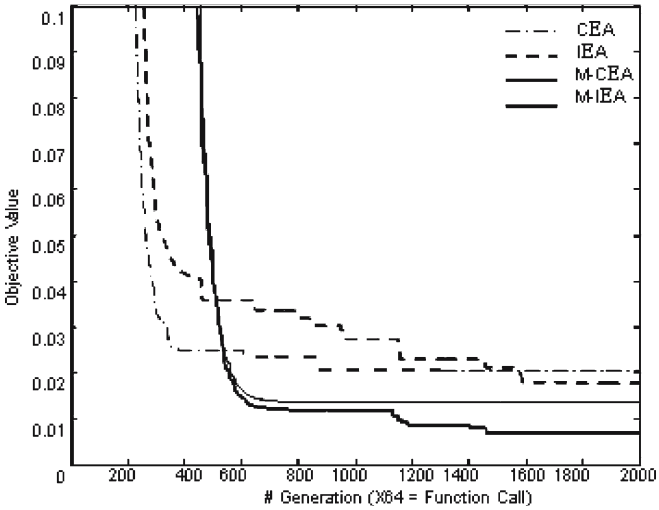

Fig. 3 Comparisons of performance of CEA/IEA for Griewank's function

calls for Griewank's function. For the same plots for Ackley's and Generalized Schwefel's functions, the readers are referred to [23]. The trends of all three functions are the same. Note that all plots are based on the averages of 30 simulation runs. As one might expect, the multi-population approach can substantially improve the discoveries of the global minimum of all three cases. However, Fig. 3 also shows that by incorporating the information-based approach, regardless of the cases of single population or multi-population, the EA has been proven to be faster convergent when compared with the cases where no information algorithm is included. It also should be noted that Fig. 3 plots the objective values versus number of function call, and this is equivalent to the objective to number of generation in the sense that the population size of each generation is the same as we noted in the figure. Table 2 also gives the results of 30 runs for Griewank's function based on the same CPU time using a dual AMD Opteron 240 system with 2 Gb RAM on a Windows XP SP2 platform. This shows that by including more computation efforts to EA, the information-guided approach still substantially improve the original approach.

Figure 4 shows the effects of threshold δ to the premature detector D in Eq. (4). In Fig. 4, the normalized objective function $(f(x) - f_{\text{global}}/f_{\max} - f_{\text{global}})$ is plot as function of scaling factor. Note that threshold is equal to the scaling factor multiplied by a constant for each

Table 2 Comparison the performance of CGA, M-CEA, M-IEA, CEA and IEA for Griewank's function based on fixed CPU time = 125 s

Methodology	M-CEA	M-IEA	CEA	IEA
Mean	7.47149×10^{-3}	2.62874×10^{-3}	1.85320×10^{-2}	1.29335×10^{-2}

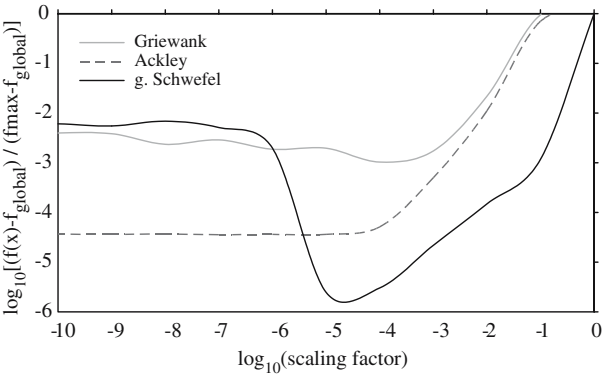

Fig. 4 The effects of threshold δ to the premature detector D

Table 3 The number of target hit in 120 simulation runs under the different thresholds

Scaling factor	1	10^{-2}	10^{-4}	10^{-6}	10^{-8}	10^{-10}	10^{-12}	10^{-14}	10^{-16}	10^{-18}	10^{-20}
f_1	0	0	104	95	89	79	89	82	76	86	79
f_2	0	0	71	120	120	120	120	120	120	120	120
f_3	0	28	120	105	85	88	88	82	86	82	87

specific problem as given in Eq. (5). All three functions are solved using the MIEA. The plot is based on 120 simulation runs of all three functions. As shown in Fig. 4, there exist critical thresholds for these extremely rugged functions. In case of too high threshold, the proposed algorithm does not perform appropriately. Table 3 gives the number of target hit in 120 simulation runs under the different scaling factors. As shown in Table 3 and Fig. 4, there exist optimal thresholds for Griewank's and generalized Schwefel's functions. Note that the critical threshold of Griemank's function (f_1) is $10^{-2.763}$, the critical threshold of Ackley's function (f_2) is $10^{-0.631}$, and the critical threshold of the generalized Schwefel's function (f_3) is $10^{-0.364}$. As described in Section 3, Fig. 4 and Table 3 show that the global minimum cannot be found if the threshold of Eq. (5) is greater than the critical threshold.

Figure 5 shows the importance of information-based orthogonal design for solving the Griewank's function. There are three cases in each figure, (i) original GA; (ii) information selection of mutation variables, but information-based orthogonal design is not implemented, that is, mutation range is random; and (iii) the full approach derived by this work. For the same plots for Ackley's and Generalized Schwefel's functions, the readers are also referred to [23]. It is shown that without the information-based orthogonal design, the improvement becomes very limited.

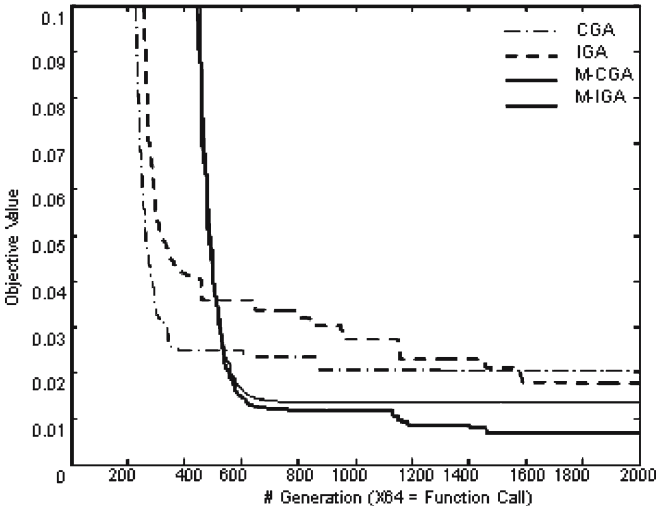

Fig. 5 Comparison of information-guided and random-guided variable-selection of Griewank's function

4.2 Multi-component distillation column

The dynamic behavior of a triple component distillation column using 15 trays is simulated. The steady state, tray parameter and components could be found in reference [12]. The following assumptions are applied in our simulation:

- Murphree efficiency is used

$$\mathbf{ME}_{nj} = \frac{\mathbf{y}_{nj} - \mathbf{y}^{\mathrm{T}}_{n-1,j}}{\mathbf{y}^{*}_{nj} - \mathbf{y}^{\mathrm{T}}_{n-1,j}} \tag{23}$$

  where y^{*}_{nj} is composition of vapor in phase equilibrium with liquid on the nth tray with composition $\mathbf{x}_{nj}$, $\mathbf{y}_{nj}$ is actual composition of vapor leaving the nth tray, $\mathbf{y}^{\mathrm{T}}_{n-1,j}$ is actual composition of vapor entering the nth tray, and $\mathbf{ME}_{nj}$ is Murphree vapor efficiency for the jth tray component on the nth tray.
- Dynamics of the condenser and the re-boiler will be negligible
- Energy equation on each tray is algebraic
- Volumetric liquid holdups in the reflux drum and column base are held perfectly constant
- Liquid hydraulics are calculated from the Francis weir formula
- Pressure is constant and known on each tray

There is one continuity equation for the nth tray:

$$\frac{d\mathbf{W}_n}{dt} = \mathbf{L}_{n+1} + \mathbf{F}_n + \mathbf{F}\mathbf{v}_{n-1} + \mathbf{V}_{n-1} - \mathbf{V}_n - \mathbf{L}_n - \mathbf{S}_n - \mathbf{S}\mathbf{v}_n \tag{24}$$

There are two component continuity equations for each tray:

$$\frac{d\mathbf{W}_n\mathbf{x}_{nj}}{dt} = \mathbf{L}_{n+1}\mathbf{x}_{n+1} + \mathbf{F}_n\mathbf{x}\mathbf{f}_{nj} + \mathbf{F}\mathbf{v}_{n-1}\mathbf{y}\mathbf{f}_{n-1,j} + \mathbf{V}_{n-1}\mathbf{y}_{n-1,j} - \mathbf{V}_n\mathbf{y}_{nj} - \mathbf{L}_n\mathbf{x}_{nj} - \mathbf{S}_n\mathbf{x}_{nj} - \mathbf{S}\mathbf{v}_n\mathbf{y}_{nj} \tag{25}$$

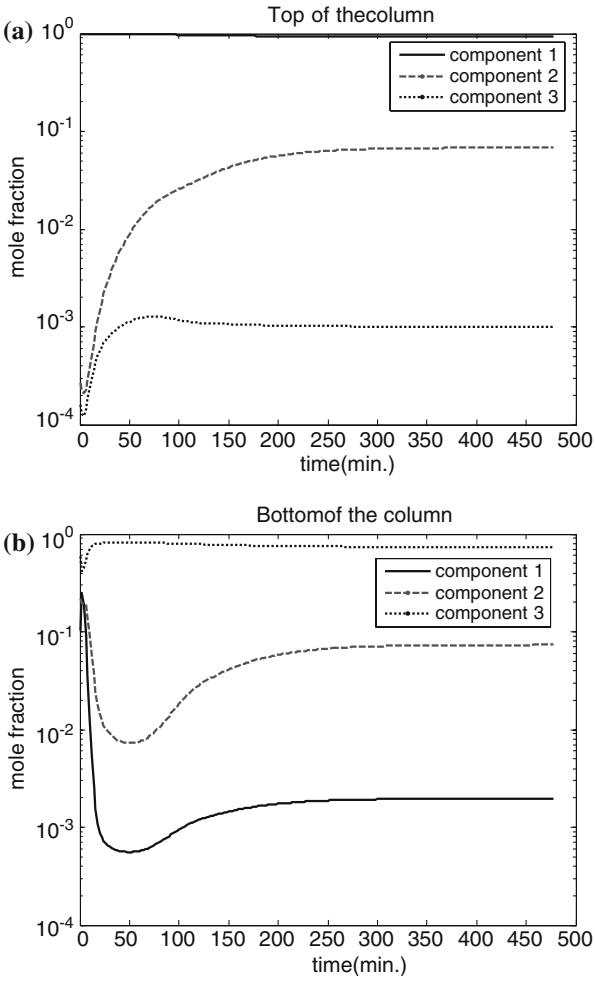

Fig. 6 Dynamic behavior of composition in the top and the bottom column: (**a**) top of the column and (**b**) bottom of the column

There is one energy equation for each tray:

$$\frac{d\mathbf{W}_n\mathbf{h}_n}{dt} = \mathbf{L}_{n+1}\mathbf{h}_{n+1} + \mathbf{F}_n\mathbf{hf}_n + \mathbf{Fv}_{n-1}\mathbf{Hf}_{n-1} + \mathbf{V}_{n-1}\mathbf{H}_{n-1} - \mathbf{V}_n\mathbf{H}_n - \mathbf{L}_n\mathbf{h}_n - \mathbf{S}_n\mathbf{h}_n - \mathbf{Sv}_n\mathbf{H}_n \quad (26)$$

The Antoine equation can be described as follows:

$$\ln\left(P_{nj}^0\right) = \frac{A_j}{\mathbf{T}_n} + \mathbf{B}_j \quad (27)$$

where $\mathbf{P}_{nj}^0$ is the vapor pressure of pure component j, and $\mathbf{T}_n$ is the temperature of the nth tray.

Liquids that obey Raoult's law are called ideal:

$$\mathbf{P} = \sum_{j=1}^{3} \mathbf{w}_{nj}\mathbf{P}_{nj}^0 \quad (28)$$

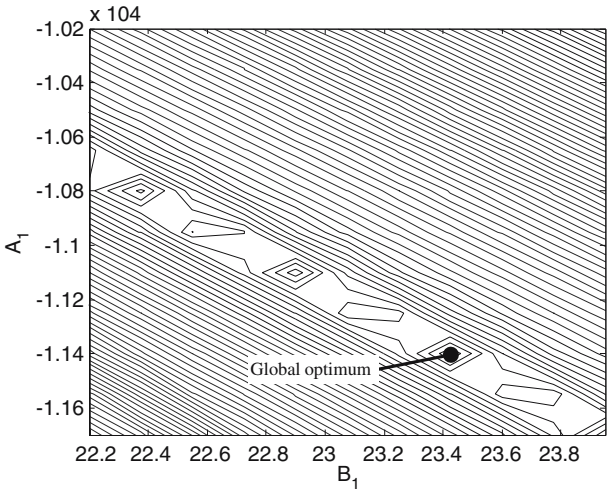

Fig. 7 Contour plot of roughness for distillation problem

Table 4 Comparison of CEA/IEA for multi-distillation problem

	A1	A2	A3	B1	B2	B3
Optimal	−11, 638	−18, 703	−13, 142	23.8484	33.3486	22.3025
M-CEA	−11, 700.00	−18, 601.17	−13, 193.14	23.9641	33.2035	22.3733
M-IEA	−11, 671.10	−18, 659.99	−13, 173.91	23.90934	33.2891	22.3472

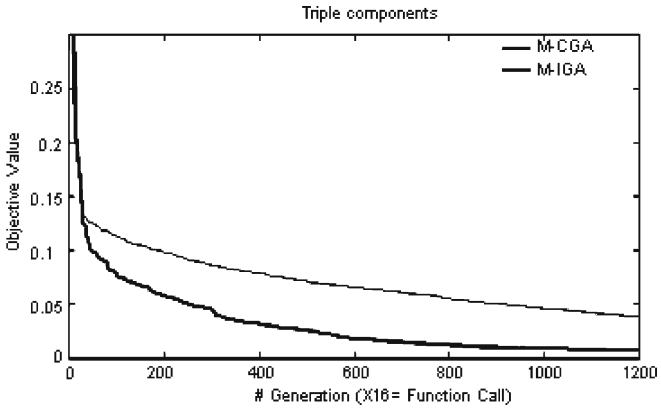

Fig. 8 Comparisons of performance of CEA/IEA for ternary distillation problem

$$\mathbf{y}_{nj} = \frac{\mathbf{x}_{nj}\mathbf{P}^0_{nj}}{\mathbf{P}} \tag{29}$$

There are 66 original differential equations and 252 algebraic equations used in our simulation. This problem is solved using a cluster computer with 16 AMD Athlon MP 2400+ CPUs and 8 Gb RAM in National Tsing-Hua University, Taiwan.

In this industrial application cases, the phase equilibrium equations are very difficult to derive, and hence there is a need to find the numerical values by fitting the dynamic

data. That is, $\mathbf{A}_j$ and $\mathbf{B}_j$'s in Eq. (27) are assumed unknown for all three components. For this testing, $\mathbf{A} = \{\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3\} = \{-11638,\ -18703,\ -13142\}$ and $\mathbf{B} = \{\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3\} = \{23.8484,\ 33.3486,\ 22.3025\}$ are set. In this work, it is assumed that the equally spaced product compositions can be measured from the plant, i.e., the compositions of top tray and bottom tray can be obtained. Figures 6a and 6b show the typical dynamic profiles in the top and bottom trays for all three components. Figure 7 illustrates the contour plot of the fitness function for $\mathbf{A}_1$ and $\mathbf{B}_1$ by fixing the values of $\mathbf{A}_2$, $\mathbf{B}_2$, $\mathbf{A}_3$ and $\mathbf{B}_3$. In Fig. 7, the fitness plot of two parameters identification problem of the distillation column is showed. It is very clear that the fitness plot of this kind of two parameters identification problem exits multi-local optimums. While this problem is extended to six parameters identification, the fitness plot will be extremely rugged. Table 4 and Fig. 8 compare the proposed approach with the original multi-population GA. Figure 8 shows the average convergence diagram of 20 runs that IEA got better converge curve than non-guided ones. Once again, its shows that implementing the information-guided mutation to EA could promote the search more efficiently.

5 Conclusions

An information-based EA is derived. The basic improvements from the original EA are:

(i) The detection of the situation of probable premature convergence.
(ii) The mutation variables are selected based on the information entropy of each variable.
(iii) The way of mutation is based on the information entropies of each range of the selected variables.
(iv) The design of the mutation variables are based on an orthogonal array experimental design approach.

The above mutation approach is general to all types of EA if the algorithm includes mutation as part of that algorithm. Three benchmark problems and an industrial scale problem are simulated to show the validity of this algorithm. The results indicate that it is promising to include information mutation to EA.

Acknowledgement

The authors thank the financial support for this work from National Science Council, Taiwan, through the grant NSC-92-2214-e007-008.

Appendix

The detail to construction an orthogonal array is as follows:

```
Step 1: Construct the basic columns:
    FOR r = 1 : J
        m = (Q^(r-1) - 1)/(Q - 1) + 1
        FOR i = 1 : Q^J
            α_{i,m} = mod( ⌊ (i-1)/Q^(J-r) ⌋ , Q )
        END
    END
```

Step 2: Construct the non-basic columns

FOR k = 2 : J

$m = \frac{Q^{r-1}-1}{Q-1} + 1$

FOR s = 1 : (m - 1)

FOR t = 1 : (Q - 1)

FOR i = 1 : Q^J

$\alpha_{i,m+(s-1)\times(Q-1)+t} = \mathrm{mod}\left(\left(\alpha_{i,s} \times t + \alpha_{i,m}\right), Q\right)$

END

END

END

END

Step 3: Increment by one for all $\alpha_{i,m}$

$1 \le i \le M$ and $1 \le m \le N$

Suppose, there are 4 variables (V_1, V_2, V_3, V_4) will be selected to renew the population, and the lower bound of less discovered section of each variable is $L_{1,k1}$, $L_{2,k2}$, $L_{3,k3}$, $L_{4,k4}$. If the less discovered sections of each selected variable will be quantized to 4 levels ($Q = 4$), and then J will be found to be 2. We can get 16 combinations ($M = 16$) of the 4 variables.

$$\beta = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 \\ 1 & 3 & 3 & 3 \\ 1 & 4 & 4 & 4 \\ 2 & 1 & 2 & 3 \\ 2 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \\ 2 & 4 & 1 & 2 \\ 3 & 1 & 3 & 1 \\ 3 & 2 & 4 & 2 \\ 3 & 3 & 1 & 3 \\ 3 & 4 & 2 & 4 \\ 4 & 1 & 4 & 3 \\ 4 & 2 & 1 & 4 \\ 4 & 3 & 2 & 1 \\ 4 & 4 & 3 & 2 \end{bmatrix}_{16\times 4}$$

References

1. Alavi, S., Thompson, D.L.: A molecular-dynamics study of structural and physical properties of nitromethane nanoparticles. J. Chem. Phys. **120**, 10231–10239 (2004)
2. Baker, J.E.: Adaptive selection methods for genetic algorithms. Proc. Int. Conf. Genetic Algorithms **1**, 101–111 (1985)
3. Bursulaya, B.D., Totrov, M., Abagyan, R., Brooks, C.L.: Comparative study of several algorithms for flexible ligand docking. J. Comp.-Aided Mol. Des. **17**, 755–763 (2003)
4. Chen, J.H., Wong, S.H., Jang, S.S.: Product and process development using artificial neural-network model and information analysis. AIChE J. **44**, 876–887 (1998)
5. Chipperfield, A.J., Fleming, P.J., Pohlheim, H., Fonseca, C.M.: Genetic Algorithm Toolbox User's Guide. pp. 6–25 ACSE Research Report No. 512, University of Sheffield, Sheffield (1994)
6. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, Chichester (2001)
7. Floudas, C.A., Pardalos, P.M.: Optimization in Computational Chemistry and Molecular Biology: Local and Global Approaches. Kluwer Academic Publishers, Boston (2000)

8. Gen, M., Cheng, R.: Genetic Algorithms & Engineering Optimization. John Wiley & Sons, New York (2000)
9. Katare, S., Caruthers, J.M., Delgass, W.N., Venkatasubramanian, V.: An intelligent system for reaction kinetic modeling and catalyst design. Industr. Eng. Chem. Res. **43**, 3484–3512 (2004)
10. Klepeis, J.L., Floudas, C.A.: Ab initio tertiary structure prediction of proteins. J. Global Optim. **25**, 113–114 (2003)
11. Leung , Y.W., Wang, Y.: An orthogonal genetic algorithm with quantization for global numerical optimization. IEEE Trans. Evol. Comput. **5**, 41–53 (2001)
12. Luyben, W.L.: Process Modeling, Simulation and Control for Chemical Engineers. McGraw -Hill, New York (1990)
13. Morris, G.M., Goodsell, D.S., Halliday, R.S., Huey, R., Hart, W.E., Belew, R.K., Olson, A.J.: Automated docking using a lamarckian genetic algorithm and and empirical binding free energy function. J. Comput. Chem. **19**, 1639–1662 (1998)
14. Mühlenbein, H., Schomisch, M., Born, J.: The parallel genetic algorithm as a function optimizer. Parallel Comput. **17**, 619–632 (1991)
15. Ong, Y.S., Keane, A.J.: Meta-Lamarckian learning in memetic algorithms. IEEE Trans. Evol. Comput. **8**, 99–110 (2004)
16. Shannon, C.E.: A mathematical theory of communication. Bell Syst. Tech. J. **27**, 379–423, 623–657 (1948)
17. Siarry, P., Berthiau, G., Durbin, F., Haussy, J.: Enhanced simulated annealing for globally minimizing functions of many-continuous variables. ACM Trans. Math. Software **23**, 209–228 (1997)
18. Tomassini, M., Vanneschi, L., Fernandez, F., Galeano, G.: Diversity in multipopulation genetic programming. In: Genetic and Evolutionary Computation - GECCO 2003, PT II, Proceedings Lecture Notes In Computer Science 2724, pp. 1812–1813 (2003)
19. Tomassini, M., Vanneschi, L., Fernandez, F., Galeano, G.: A study of diversity in multipopulation genetic programming. Artif. Evol. Lect. Notes Comp. Sci. **2936**, 243–255 (2004)
20. Tsujimura, Y., Gen, M.: Entropy-based genetic algorithm for solving TSP. In: Second International Conference on Knowledge-Based Intelligent Electronic Systems, pp. 21–23 (1998)
21. Wang, K.F., Qian, Y., Yuan, Y., Yao, P.J.: Synthesis and optimization of heat integrated distillation systems using an improved genetic algorithm. Comp. Chem. Eng. **23**, 125–136 (1998)
22. Wright, A.H.: Genetic algorithms for real parameter optimization. In: Rawlins, G.J.E. (ed) Foundations of Genetic Algorithms, Morgan Kaufmann Publishers, pp. 205–218 (1991)
23. Yeh, C.W.: Information guided evolutionary algorithm. Ph.D Thesis, National Tsing-Hua University, Hsin Chu, Taiwan (2005)
24. Yen, J., Liao, J.C., Lee, B.J., Randolph, D.: A hybrid approach to modeling metabolic systems using a genetic algorithm and simplex method. IEEE Trans. Syst. Man Cybernet. Part B–Cybern. **28**, 173–191 (1998)
25. Zhong, W.C., Liu, J., Xue, M., Jiao, L.C.: A multiagent genetic algorithm for global numerical optimization. IEEE Trans. Syst. Man Cybernet.–Part B **34**, 1128–1141 (2004)