

# Improved Real Coded Genetic Algorithm and Its Simulation

Jianmin Zhu

College of Mechanical Engineering, University of Shanghai for Science and Technology, 200093, Shanghai, China, Email:jmzhu6688@163.com

Donger Zhou, Fucui Li, Tingting Fu

College of Mechanical Engineering, University of Shanghai for Science and Technology, 200093, Shanghai, China, zder005@163.com

**ABSTRACT**—Aiming at improving search efficiency limitations of canonical real coded genetic algorithm, this paper improves three aspects for the canonical real coded genetic algorithm, that are initial population generating, overall process of algorithm and the mutation operator, then puts forward an improved real coded genetic algorithm. This improved algorithm combines the series operation and parallel operation of the three basic operators (selection, crossover and mutation), presents new methods of uniform generating initial population and progressive variation, the former contributes to generating good initial population, the latter processes more active mutation operation on the basic of controlling the average fitness in a certain range, which can obtain advantage individual and reduce evolution time. Using typical standard test functions for two algorithms' simulation research, the simulation results show that, the proposed real coded genetic algorithm is superior to the canonical real coded genetic algorithm on the aspects of search speed and search accuracy, improving search efficiency of canonical real coded genetic algorithm greatly.

**Index Term**—real coded genetic algorithm, improve, series operation and parallel operation, progressive variation, uniform generating of initial population

## I. INTRODUCTION

A fruitful progress in theoretical research and application has been achieved since the genetic algorithm (GA) was first introduced by John Holland in the late 1960s [1-2]. Genetic algorithm is an intelligent searching algorithm taking biological natural selection and natural genetic mechanisms for references and it is random and extremely invisible parallel. As a product of multi-disciplinary combination, GA has developed into a self-organizing, self-adaptive, self-learning optimization technique to get new solutions [3-5] by utilizing the current solutions and random information. Genetic algorithm has several common coding schemes, such as binary coding, real coding, matrix coding, tree encoding and quantum bits coding [6-8]. The coding scheme determines operation the mode of genetic operators and has significant effect on the search efficiency of the genetic algorithm [9-10]. Real coded genetic algorithm is

one of the most promising optimization algorithms in which each gene of individual is directly denoted by a real number and that makes it easy for numerical operations [11-12].

In 1989, the real coding of hairpin DNA structure was first put into practice by Lacasius and Kateman [13] to explore the fitting of NMR data of chemometrics, since then, the real coded genetic algorithm has been frequently used to solve real-world optimization problems. Related studies show that: compared with the binary-coded genetic algorithm, the real-coded genetic algorithm has faster converging speed and higher searching precision in solving continuous parameter optimization problems. Since the 1990s, the real coded genetic algorithm has got more and more attention and development [14]. With the in-depth study of the real-coded genetic algorithm, researchers pointed out that with larger search space the real-coded genetic algorithm has low efficiency in solving complex optimization problems [15]. In recent years, improvements on real-coded genetic algorithm have been made by scholars both domestic and foreign, Dong proposed the improved linear crossover operator which perfected the converging speed and accuracy of the algorithm in solving optimization problems [14]. Zheng proposed the netlike distribution of initial population creation and improved dynamic crossover probability and dynamic self-adaptive mutation probability which promoted evolution efficiency [16]. Mao proposed improved non-dominated sorting genetic algorithm which designed the constellation efficiently [17]. Deep proposed energy distribution-based mutation operator which increased the search speed [18]. Ling et.al proposed wavelet-based mutation which bettered solution quality and stability [19]. Although the searching effect of real coded genetic algorithm has been varyingly improved, there is still much room for boosting the search efficiency.

In allusion to the problem of limitations in search efficiency of the canonical real coded genetic algorithm, this paper presents an improved real coded genetic algorithm which can achieve higher search efficiency by improving three aspects for the canonical real coded genetic algorithm, that are initial population generating

method, overall process of algorithm and mutation operator.

## II. IMPROVEMENT OF REAL CODED GENETIC ALGORITHM (RCGA)

### A. Uniform Distributed Generation of Initial Population

The search efficiency and accuracy are significantly influenced by initial population. In order to achieve global optimal solution, initial population should be as disperse as possible in solution space. The canonical real coded genetic algorithm generates initial population by scheduled or randomized methods which not only make the distribution of initial solutions uneven in the solution space, but also affect the optimization performance and lower the search efficiency. In this paper, a new method of initial population generation (uniform generation of initial population) has been proposed. The main principles and steps are as follows:

- (1) Divide the solution space into  $M$  sub-spaces.
- (2) Quantify each sub-space, generate  $S$  chromosomes through uniform array.
- (3) Select  $P$  chromosomes with highest fitness as initial population from  $M \times S$  chromosomes.

$M$  and  $S$  above are determined by initial population scale  $P$ . Multiple simulation analysis show that when  $M \times S$  is two to three times of  $P$ , initial population distribution will be more ideal. If  $M \times S$  is too large, initial population distribution will be relatively concentrated, causing premature convergence, if  $M \times S$  is too small, initial population distribution will be too scattered that will lower convergence efficiency of the algorithm. If  $P=50$ , the recommended value of  $M$  and  $S$  are  $M=10$ ,  $S=10 \sim 15$ .

Initial population produced by uniform generation can be uniformly distributed in solution space, which increases the probability to get the globally optimal solution, also the optimal point would not be missed and be attained easily. Therefore generations of RCGA iteration can be effectively reduced.

### B. Series and Parallel Operation of Selection, Crossover and Mutation

Series operation of selection, crossover and mutation operators is generally used in canonical real coded genetic algorithm, Although series operation has relatively mature theory and practice basis, the shortcomings still exist as follows: less high-quality individuals are produced in crossover procedure during early period of evolution, high-quality individuals may be damaged in mutation procedure which will significantly affect evolution extent of population, besides, search accuracy may also be influenced at the end of evolution, high-quality individuals generated in mutation procedure may also be damaged by crossover procedure in the next generation. In view of these weaknesses, parallel genetic structure in which operators such as selection, crossover and mutation act equally on parent population was proposed [20]. The advantage of parallel operation structure is that high-quality individuals generated by

selection, crossover mutation can reproduce themselves without any damage, however, the search speed is unsatisfactory when parallel operation structure was used separately. In order to further improve the search efficiency, this paper presents a theory to take full advantage of both series operation and parallel operation by integrating themselves. Steps are shown as follows:

Step 1: Get  $3P$  combined population with selection operator, crossover operator and mutation operator through parallel operation acted on initial population.

Step 2: Calculate fitness value of each individuals in the combined population, select  $P$  individuals with highest fitness as new population.

Step 3: Determine whether terminating condition is met. If so, then stop, if not, series operation of selection operator, crossover operator and mutation operator would be conducted.

The quality of initial population significantly affects the search efficiency of the algorithm. Parallel operation of selection operator, crossover operator and mutation operator is conducted on initial population separately, that will greatly reduce the survival rates of low-quality individuals and that will help providing high-quality individuals for series operation and improving the search accuracy and efficiency of the algorithm.

### C. Progressive Mutation

Representation of the population individuals is given by equation (1) in real code.

$$X_i^t = (x_{i1}^t, \dots, x_{ik}^t, \dots, x_{iL}^t) \quad (i=1, 2, \dots, P; t=1, 2, \dots, N; k=1, 2, \dots, L) \quad (1)$$

Where  $x_{ik}^t$  is a real variable,  $t$  is generations, initial generation is 0.  $i$  is individual number,  $L$  is variable number,  $P$  is population scale,  $N$  is maximum generation.

During the evolution, there will be a large number of identical or similar individuals in population. For canonical RCGA, individual  $x_{ik}^t$  is selected as mutation object in accordance with relevant conditions, as shown by equation (2).

$$x_{ik}^t = x_{ik}^t + \delta_i \quad (2)$$

In equation (2),  $\delta_i$  is mutation step determined by solution space.

There are certain randomness and blindness in canonical mutation which may lead to lack of diversity and low efficiency. In order to solve this problem, progressive mutation has been proposed in this paper. Principle steps are shown as follows:

Step 1: local Gaussian mutation

Set  $D$  as the threshold of variable area, calculate the fitness of the individuals as  $f(X_i^t)$  ( $i=1, 2, \dots, P; t=1, 2, \dots, N$ ). If  $f(X_i^t) \leq D$  then the mutating individual is  $X_i^t$ , determine the individual variable  $x_{ik}^t$  further. Individuals with the highest fitness can be selected as the optimal individual  $X_{id}^t$  ( $d=1, 2, \dots, P; t=1, 2, \dots, N$ ), select individual variable  $x_{dk}^t$  ( $d=1, 2, \dots, P; t=1, 2, \dots, N; k=1, 2, \dots, L$ ) as the center of Gaussian mutation, then the mutation process are given by equation (3) and equation (4)

$$x_{ik}^t = x_{dk}^t + \sigma_i \cdot N(\mu, \delta^2) \quad (i \neq d) \quad (3)$$

$$\sigma_i = \sqrt{\beta_i \cdot f(X_i^t) + \gamma_i} \quad (i=1, 2, \dots, P) \quad (4)$$

In equation (3),  $N(\mu, \delta^2)$  is an independent normal random variable with mean value  $\mu$  and variance  $\delta^2$ , it is determined by threshold  $D$  and mutation probability  $P_m$ . In equation (4),  $\beta_i$  and  $\gamma_i$  are specific parameters, generally  $\beta_i=1, \gamma_i=0$ .

Step 2: selective mutation

Generate a random number  $b(b \in (0,1))$  and determine to-be mutated individual variable  $x_{mk}^t (m=1, 2, \dots, P; k=1, 2, \dots, L; t=1, 2, \dots, N)$ ,

if  $b \geq 0.5$  then

$$x_{mk}^t = x_{mk}^t - (1 - rand) \times rand \times C \times (1 - G/N)^2 \quad (5)$$

if  $b < 0.5$  then

$$x_{mk}^t = x_{mk}^t + rand \times C \times (1 - G/N)^2 \quad (6)$$

In equation (5) and equation (6),  $G$  is the current generation,  $N$  is the maximum generation,  $rand$  is a random number between 0 and 1,  $C$  is a constant which could be adjusted by optimization goals.

According to mutation method above, individuals with low fitness are firstly conducted local Gaussian mutation then select individuals randomly for selective mutation. If two mutations overlap, then  $i=m$ , which means that the individuals with low fitness are mutated twice, that would be helpful to the diversity of generation. At the beginning of the evolution, a large number of individuals were conducted on Gaussian mutation that would contribute to the generation of individuals with high quality. Along with the evolution, overall fitness of population obviously increases and the number of individuals whose fitness value are lower than threshold  $D$  decreases, namely the number of individuals conducted on local Gaussian mutation is decreasing or approaching to zero which is beneficial to the convergence of the algorithm in the later evolution stage.

### III. ALGORITHM STEPS

The improved real coded genetic algorithm proposed in this paper combines the parallel operation and the series operation of three basic operators (selection operator, crossover operator and mutation operator), and adopts uniform generation of initial population and progressive mutation.

#### A. Flow Chart of the Algorithm

The flow chart of proposed real coded genetic algorithm is given in Fig. 1.

Fig. 2 shows the flow chart of canonical real coded genetic algorithm, it creates initial population randomly, conducts series operation of selection operator, crossover operator and mutation operator, and perform elitism strategy in the end. In Fig. 1 parameters needed to be set are: population scale  $P$ , maximum generation  $N$ , variable number  $L$ , crossover probability  $P_c$ , mutation probability  $P_m$ .

The algorithm shown in Fig. 1 consists two parts. Part 1: selection operator, crossover operator and mutation operator are conducted on initial population  $G^0 = \{X_1^0, X_2^0, \dots, X_i^0\} (i=1, 2, \dots, P)$ , at the same time, get transitional combination of populations  $G = \{X_1, X_2, \dots, X_j\} (j=1, 2, \dots, 3P)$  through parallel operation. Compare the fitness of

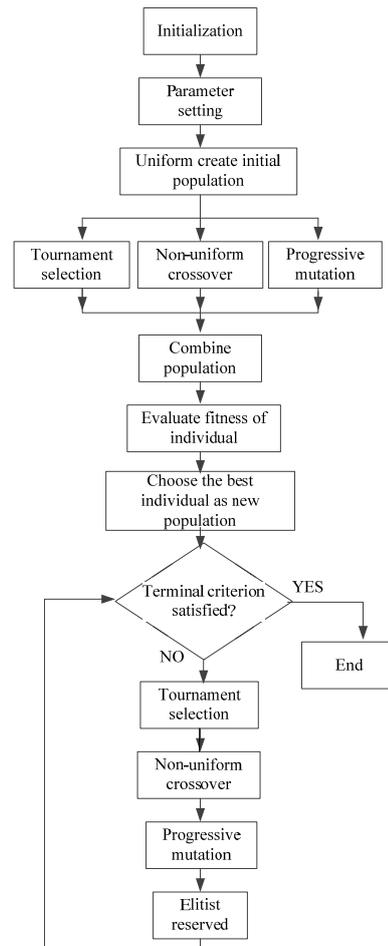


Figure 1. Flow chart of the improved real coded genetic algorithm

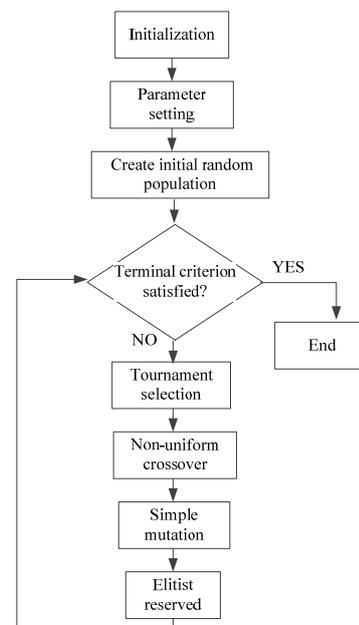


Figure 2. Flow chart of canonical real coded genetic algorithm

individuals  $f(X_j) (j=1, 2, \dots, 3P)$ , then select  $P$  individuals with the highest fitness as progeny population  $G^1 = \{X_1^1, X_2^1, \dots, X_i^1\} (i=1, 2, \dots, P)$ . Part 2: conduct the series operation with selection operator, crossover operator and mutation operator on the  $t$  generation population

$G^t(t=1, 2, \dots, N)$ , acquire progeny population  $G^{t+1}(t=1, 2, \dots, N)$  with elitism strategy, estimate whether the termination conditions are met, if so, algorithm terminates, if not, generation  $t=t+1$ , repeat part 2.

### B. Coding Scheme

Compared with binary coding, the real coding adopted in this paper encodes in the form of solution space directly which makes it specific in code meaning and easy in introducing information of specific areas and short in the length of the coding string. In addition, frequent encoding and decoding for real coded genetic algorithm are no longer needed, therefore, smaller memory space would be required, relevant numerical operation would be simpler, the computing complexity of GA is also decreased. Above all, the searching speed is increased significantly.

### C. The Generation of Initial Population

Initial population is generated uniformly as mentioned before. The number of subspace  $M$  and the number of chromosomes  $S$  are determined by population size  $P$  in the beginning, initial population is then generated by procedure (1), (2) and (3) of part A in section II.

### D. Tournament Selection

The basic idea of tournament selection is that: select a certain number of individuals randomly from the current population, save the individuals with highest fitness to the next generation, repeat the process above until the scale of the next generation reaches predetermined size. The tournament size is usually set to 2 based on a large number of optimization examples. Set current population as  $G^t=\{X_1^t, X_2^t, \dots, X_i^t\}(i=1, 2, \dots, P; t=1, 2, \dots, N)$ , select two individuals  $X_1^t, X_2^t$  randomly from  $G^t$  each time, evaluate and compare the fitness value of  $f(X_1^t)$  and  $f(X_2^t)$ . If  $f(X_1^t) > f(X_2^t)$ , then select  $X_1^t$  as a member of progeny generation. After  $P$  times selection, a new progeny generation is created as  $G^{t+1}$ .

### E. Non-uniformed Crossover

When the random number generated is less than crossover rate  $P_c$ , two individuals  $X_i^t, X_j^t (i, j=1, 2, \dots, P; i \neq j)$  are selected to be crossed over in the reference with relevant selection criteria, and variables  $k(k=1, 2, \dots, L)$  would be further determined to be crossed over. Conduct non-uniform crossover operation as equation (7) and equation (8).

$$x_{ik}^t = (x_{jk}^t \times a + x_{ik}^t \times (1-a)) \quad (7)$$

$$x_{jk}^t = (x_{ik}^t \times a + x_{jk}^t \times (1-a)) \quad (8)$$

In equation (7) and equation (8),  $a$  is random number and  $a \in [0, 1]$ . When  $a$  is a constant number, the operation is uniform crossover. Because of randomness of  $a$ , two new individuals would be got even if crossover is conducted on two same individuals. Therefore, the diversity of population is improved and non-uniform crossover operation has better effect than uniform crossover.

### F. Progressive Mutation

Local Gaussian mutation and selective mutation were progressively conducted in this paper. Execute mutation operation step 1 and step 2 of part C in section II in sequence. When generated random number is less than mutation rate  $P_m$ , implement local Gaussian mutation with equation (3) and equation (4) after the determination of to-be mutated individual  $x_{ik}^t$  and the center of Gaussian mutation  $x_{dk}^t$ , then implement selective mutation with equation (5) and equation (6) after the determination of to-be mutated individual variable  $x_{mk}^t$ .

### G. Reservation and Replacement of Elite Individuals

Calculate the fitness of individuals  $f(X_i^t)(i=1, 2, \dots, P; j=1, 2, \dots, N)$  in the  $j$ th generation. If  $f(X_b^j) > f(X_i^j)$  ( $i=1, 2, \dots, P; j=1, 2, \dots, N; i \neq b$ ), namely  $X_b^j$  is the best individual of the current population. If  $f(X_w^j) < f(X_i^j)$  ( $i=1, 2, \dots, P; j=1, 2, \dots, N; i \neq w$ ), namely  $X_w^j$  is the worst individual of the current generation. Comparing the fitness value of the best individuals between the current population and the last population, if  $f(X_b^{j-1}) > f(X_b^j)$ , then replace the current worst individual with the current best individual, namely  $X_w^j = X_b^j$ , if  $f(X_b^{j-1}) < f(X_b^j)$ , then replace the best individual in the last population with the current best individual, namely  $X_b^{j-1} = X_b^j$ .

Elite individual reservation and replacement strategy encourages the survival and reproduction of individuals greatly, also provides the next evolution with excellent population. It made it faster for the algorithm to find the optimal solution.

## IV. SIMULATIONS

In order to verify the effectiveness of the improved algorithm, four standard functions, Schaffer function, Schaffer1 function, Himmelblau function and Griewank function are chosen to validate the search accuracy and speed of the algorithm through searching their global minimum values. All software codes are written in Matlab (R2012a). The improved algorithm was executed on a core i3, 2.3GHz, 2GB RAM machine with a Windows 7 OS. The expression of Schaffer Function is given by equation (9)

$$F = 0.5 + \frac{[\sin(\sqrt{x^2 + y^2})]^2 - 0.5}{[1.0 + 0.001(x^2 + y^2)]^2}, \quad |x| \leq 100, |y| \leq 100 \quad (9)$$

Fig. 3 is a three-dimensional diagram drawn according to equation (9), from which the binary multimodality of Schaffer functions would be observed. When  $(x^*, y^*) = (0, 0)$ , the global minimum value, surrounded by multiple local minimum values is  $F(x^*, y^*) = 0$ . The function of multivariate multimodality is shown as equation (9), the anti-premature performance of the proposed algorithm can be testified by this function. The expression of Schaffer1 Function is given by equation (10).

$$F = (x^2 + y^2)^{0.25} \{ \sin^2[50(x^2 + y^2)^{0.1}] + 1 \}, \quad |x| \leq 10, |y| \leq 10 \quad (10)$$

When  $(x^*, y^*) = (0, 0)$ , the global minimum value is  $F(x^*, y^*) = 0$ . Equation (10) shows the optimization problem of multivariate unimodality functions, the search efficiency of the proposed algorithm can be testified by this function. The expression of Himmelblau Function is given by equation (11)

$$F = (x^2 + y - 11)^2 + (x + y^2 - 7)^2, \quad |x| \leq 6, |y| \leq 6 \quad (11)$$

Fig. 4 is a three-dimensional diagram drawn according to equation (11), from which four global minimum values would be observed, namely when  $(x^*, y^*) = \{(3.0, 2.0), (3.584428, -1.848126), (-2.805118, 3.013131), (-3.779310, -3.283186)\}$ , the global minimum value of this function is  $F(x^*, y^*) = 0$ . Equation (11) shows the optimization problem of multivariate multimodality functions the global quick searching ability can be verified by this function. The expression of Griewank function is given by equation (12)

$$F = (x^2 + y^2) / 4000 - \cos x \times \cos(y / \sqrt{2}) + 1, \quad |x| \leq 600, |y| \leq 600 \quad (12)$$

When  $(x^*, y^*) = (0, 0)$ , the global minimum value of this function is  $F(x^*, y^*) = 0$ . Equation (12) shows the optimization problem of multivariate unimodality functions, the search efficiency of the proposed algorithm can be testified by this function with multivariate unimodality.

*A. Comparisons in Searching Accuracy*

$P_c, P_m$  and  $P$  are specified as 0.8, 0.1 and 50 each in the simulation. The maximum generations of algorithm is set to  $N=100$  as the termination condition. Search the global minimum value of four different standard testing functions with the canonical real coded genetic algorithm and the improved real coded genetic algorithm.

Fig. 5 shows that the optimum value of Schaffer function using the canonical real coded genetic algorithm is  $1.4705e-004$  in 100 generations. Fig. 6 shows that the optimum value of the improved real coded genetic algorithm is  $7.2960e-010$  in 100 generations.

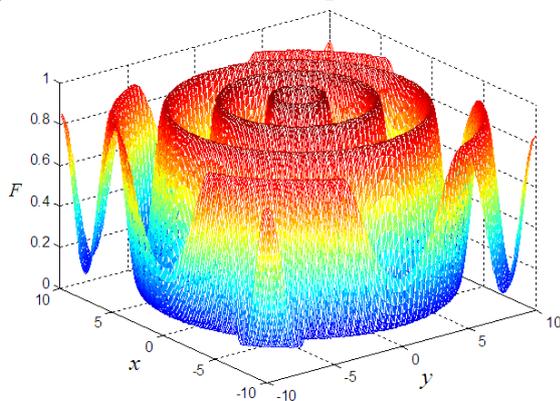


Figure 3. The three-dimensional diagram of Schaffer Function

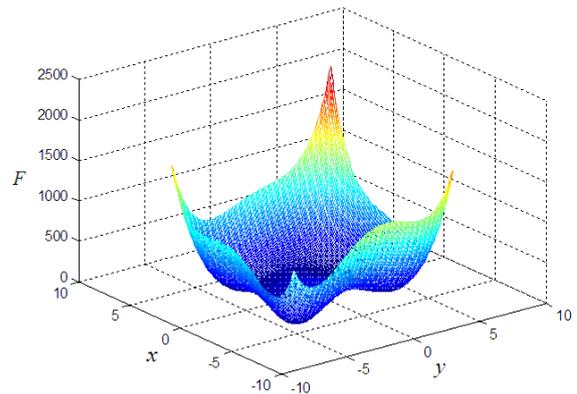


Figure 4. The three-dimensional diagram of Himmelblau Function

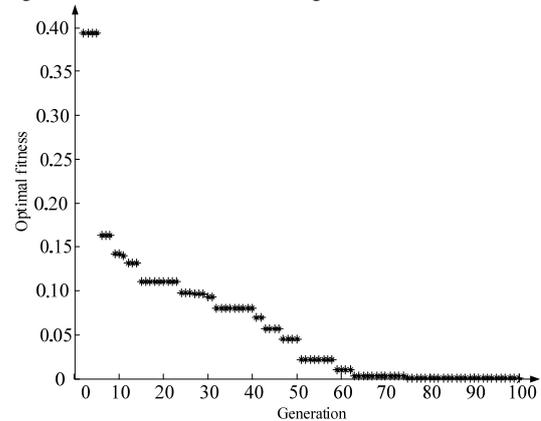


Figure 5. Optimized results of the canonical real coded genetic algorithm

Fig. 7 shows that the optimum value of Schaffer1 function using the canonical real coded genetic algorithm is 0.1166 in 100 generations. Fig. 8 shows that the optimum value of the improved real coded genetic algorithm is 0.0047 in 100 generations.

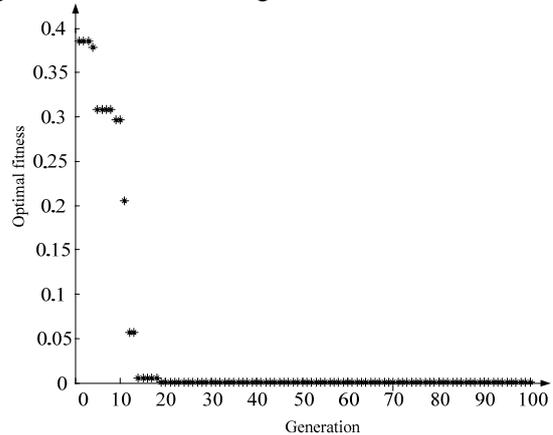


Figure 6. Optimized results of the improved real coded genetic algorithm

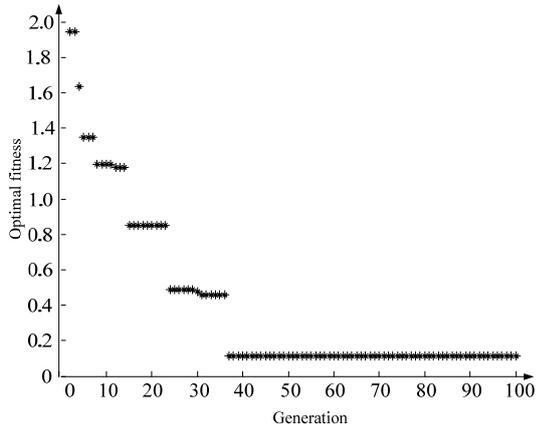


Figure 7. Optimized results of the canonical real coded genetic algorithm

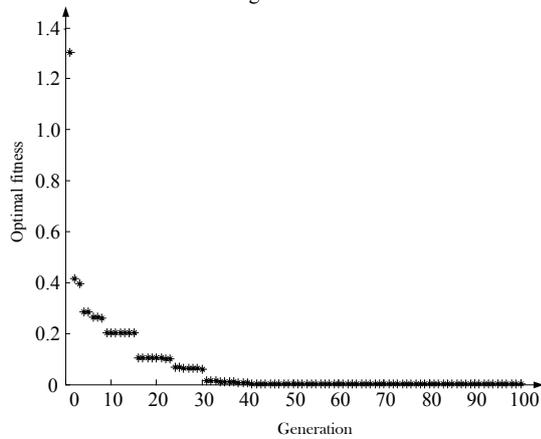


Figure 8. Optimized results of the improved real coded genetic algorithm

Fig. 9 shows that the optimum value of Himmeblau function using the canonical real coded genetic algorithm is 0.0048 in 100 generations. Fig. 10 shows that the optimum value of the improved real coded genetic algorithm is 2.9531e-007 in 100 generations.

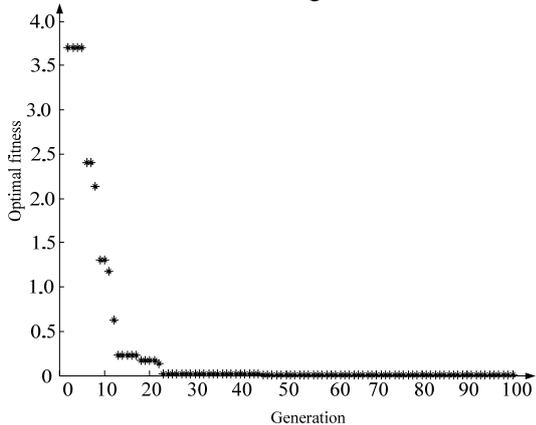


Figure 9. Optimized results of the canonical real coded genetic algorithm

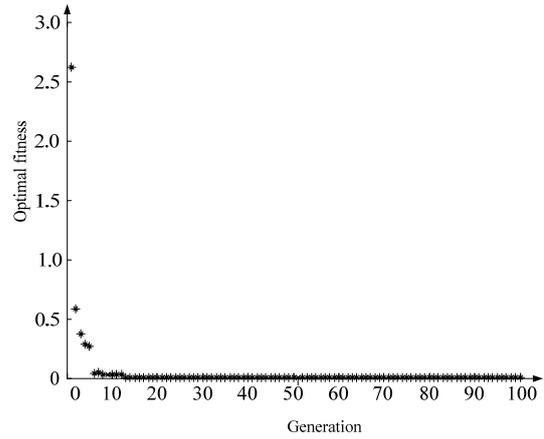


Figure 10. Optimized results of the improved real coded genetic algorithm

Fig. 11 shows that the optimum value of Griewank function using the canonical real coded genetic algorithm is 0.2367 in 100 generations. Fig. 12 shows that the optimum value of the improved real coded genetic algorithm is 2.5750e-010 in 100 generations.

The optimized results indicate that: searching the global minimum value of four different standard testing functions, on the aspect of search accuracy, the improved real coded genetic algorithm is superior to the canonical real coded genetic algorithm within the same evolution times.

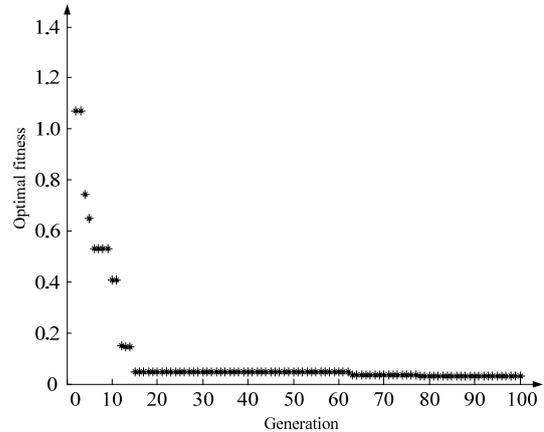


Figure 11. Optimized results of the canonical real coded genetic algorithm

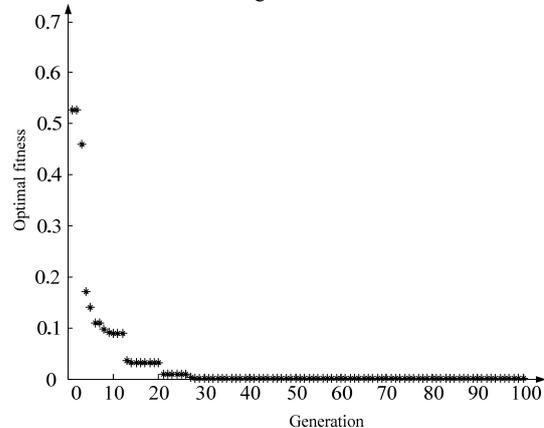


Figure 12. Optimized results of the improved real coded genetic algorithm

*B. Comparisons in Searching Speed*

$P_c$ ,  $P_m$  and  $P$  were specified as 0.8, 0.1 and 50 each in the simulation. Search accuracy was specified as  $\delta=10^{-6}$  as the termination rule. The maximum generations of algorithm was set to 100 which means that the algorithm would be stopped when the generation reaches 100 even the result doesn't meet the accuracy requirement. Search the global minimum value of four different standard testing functions with the canonical real coded genetic algorithm and the improved real coded genetic algorithm.

Fig. 13 shows that the termination result of Schaffer function using the canonical real coded genetic algorithm meets the requirement of accuracy in 73 generations. Fig. 14 shows that the searching result of the improved real coded genetic algorithm meets the requirement of accuracy in 9 generations.

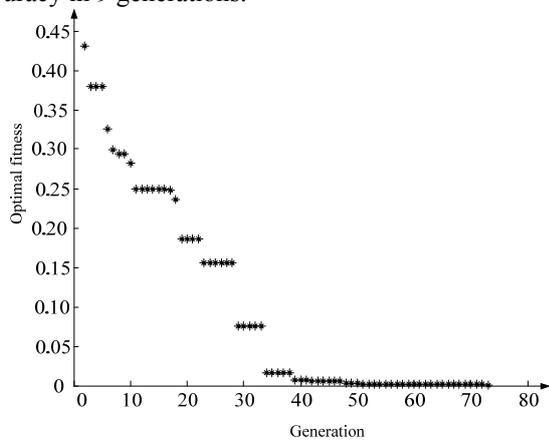


Figure 13. Optimized results of the canonical real coded genetic algorithm

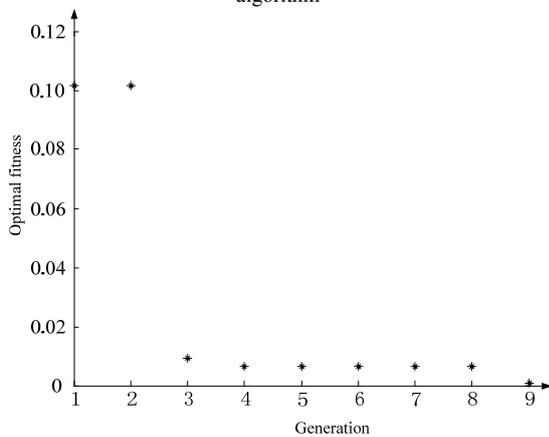


Figure 14. Optimized results of the improved real coded genetic algorithm

Fig. 15 shows that the termination result of Schaffer1 function using the canonical real coded genetic algorithm doesn't yet meet the requirement of accuracy in 100 generations. Fig. 16 shows that the termination result of the improved real coded genetic algorithm meets the requirement of accuracy in 34 generations.

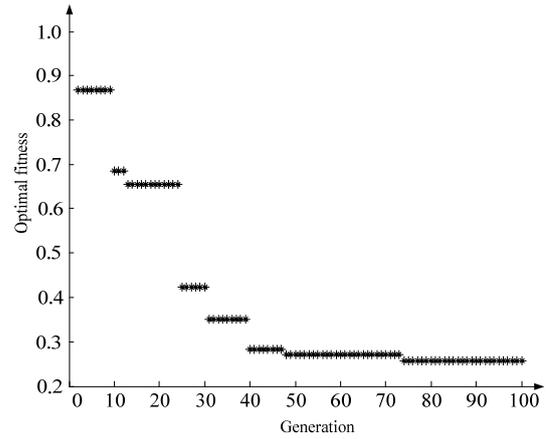


Figure 15. Optimized results of the canonical real coded genetic algorithm

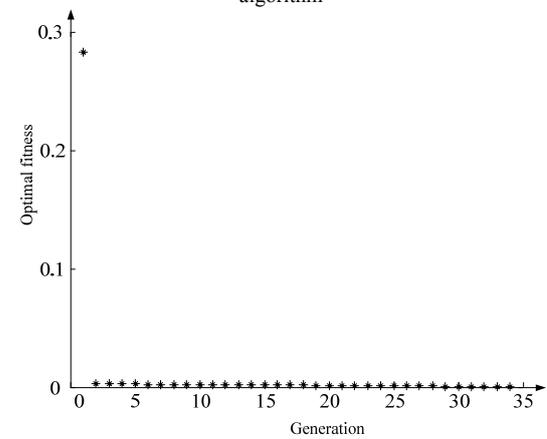


Figure 16. Optimized results of the improved real coded genetic algorithm

Fig. 17 shows that the termination result of Himmeblau function using the canonical real coded genetic algorithm meets the requirement of accuracy in 87 generations. But Fig. 18 shows that the termination result of the improved real coded genetic algorithm meets the requirement of accuracy in 9 generations.

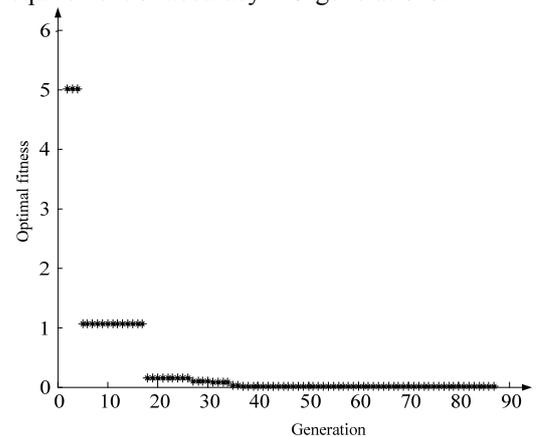


Figure 17. Optimized results of the canonical real coded genetic algorithm

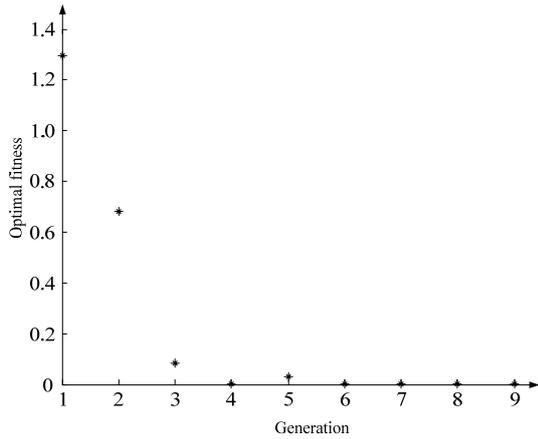


Figure 18. Optimized results of the improved real coded genetic algorithm

Fig. 19 shows that the termination result of Griewank function using the canonical real coded genetic algorithm doesn't yet meet the accuracy requirement in 100 generations which. Fig. 20 shows that the termination result of the improved real coded genetic algorithm meets the requirement of accuracy in 28 generations.

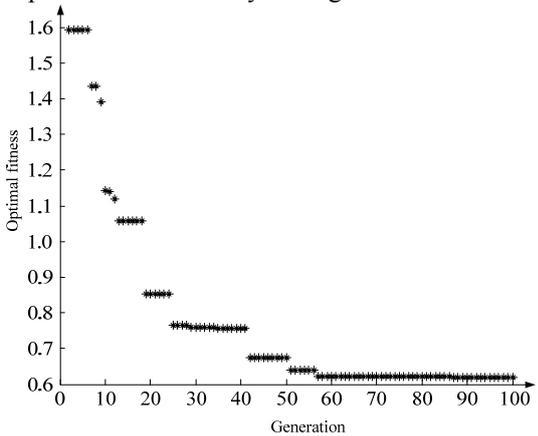


Figure 19. Optimized results of the canonical real coded genetic algorithm

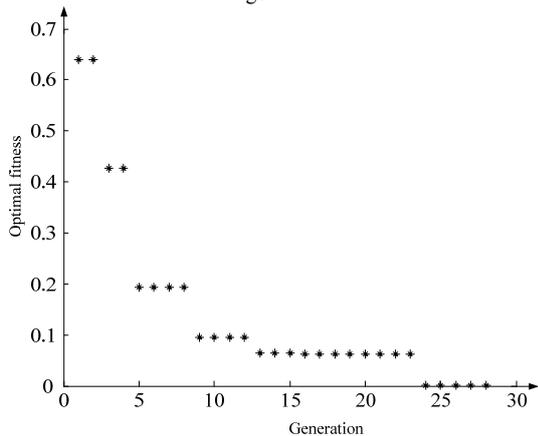


Figure 20. Optimized results of the improved real coded genetic algorithm

The optimized results indicate that: searching the global minimum value of the four different standard testing functions, on the aspect of search speed, the proposed real coded genetic algorithm is superior to the canonical real coded genetic algorithm based on the same searching accuracy.

V. CONCLUSION

This paper presents an improved real coded genetic algorithm by improving three aspects for the canonical real coded genetic algorithm, that are initial population generating method, overall process of algorithm and the mutation operator. The simulation results show that:

- (1) On the aspect of search accuracy, the proposed real coded genetic algorithm is superior to the canonical real coded genetic algorithm within the same evolution times.
- (2) On the aspect of search speed, the proposed real coded genetic algorithm is superior to the canonical real coded genetic algorithm based on the same search accuracy.
- (3) The proposed real coded genetic algorithm improves the search efficiency of canonical real coded genetic algorithm significantly.

ACKNOWLEDGEMENT

The work presented in this paper is supported by Research and Innovation Project of Shanghai Education Commission (Grant No.: 11ZZ136), National Natural Science Foundation of China (Grant No.: 50975179), Research project of Shanghai Science and Technology Commission (Grant No.: 12DZ2252300).

REFERENCES

- [1] Yan, L., Shuhong, L., Xiurong, L., "Test Paper Generating Method Based on Genetic Algorithm," Proc. of AASRI Conference on Computational Intelligence and Bioinformatics, pp. 549-553, 2012.
- [2] Bunnag D., Sun M., "Genetic algorithm for constrained global optimization in continuous variables," Applied Mathematics and Computation, vol. 171(1), pp. 604-636, 2005.
- [3] Grigioni, M. M., Morbiducci, U., Tura A., "Genetic algorithms for parameter estimation in mathematical modeling of glucose metabolism," Computers in Biology and Medicine, vol. 35(10), pp. 862-874, 2005.
- [4] Zebin, T., Alam, M. S., "Modeling and control of a two-link flexible manipulator using fuzzy logic and genetic optimization techniques," Journal of Computers, vol. 7(3), pp. 578-585, 2012.
- [5] Mac F., Gioll Bhride, Meginnity T. M., et al., "Landscape classification and problem specific reasoning for genetic algorithms," Emerald Group Publishing Limited," vol. 34(9-10), pp. 469-1497, 2005.
- [6] Panteleev A.V., Metlistsckaya D. V., "An Application of Genetic Algorithms with Binary and Real Coded for Approximate Synthesis of Suboptimal Control in Deterministic Systems," Automation and Remote Control., vol. 72(11), pp. 2328-2338, 2011.
- [7] Shifei D., Xinzhen X., Hong Z., Jian W., Fengxiang J., "Studies on Optimization Algorithms for Some Artificial Neural Networks Based on Genetic Algorithm (GA)," Journal of Computers, vol. 6(5), pp. 939-946, 2011.
- [8] W. M. Jenkins, "A decimal-coded evolutionary algorithm for constrained optimization," Computer & Structures, vol. 80(23), pp. 471-480, 2002.
- [9] Shi, L., Deng, Y.K., Sun, H.F., et al, "An Improved Real-Coded Genetic Algorithm for the Beam Forming of

- Spaceborne SAR,” IEEE Trans on Antennas and Propagation, vol. 60(6), pp. 3034-3040, 2012.
- [10] Ha J. L., Fung R. F., Han C. F., “Optimization of an impact drive mechanism based on real-coded genetic algorithm,” Sensors and Actuators, vol. 121(2), pp. 488-493, 2005.
- [11] Kazi Shah Nawaz Ripon, Sam Kwong, Man K. F., “A real-coded jumping gene genetical gorithm (RJGGA) for multiobjective optimization,” Information Sciences, vol. 177(2), pp.632-654, 2007.
- [12] Deb K., Anand A., Joshi D., “A computationally efficient evolutionary algorithm for real-parameter evolution,” Evolutionary Computation, vol. 10(4), pp. 371-395, 2002.
- [13] Lucasius, C. B., Kateman G., “Applications of genetic algorithms in chemometrics,” Proc. of the third International Conference on Genetic Algorithms. CA: Morga Kaufman Publishers, Inc, pp. 170-176, 1989.
- [14] Ying, D., Huanjie, L., Baodong, X., et al., “A improved genetic algorithm based on real coded,” Journal of Northeastern University (natural science), vol. 26(4), pp. 219-221, 2005.
- [15] Qing, Z., “Comparative study on three coded strategies for genetic algorithm,” Journal of Micro-computer Applications, vol. 24(5), pp. 47-49, 2008.
- [16] Shengrong, Z., Jiamei, L., Guoliang L., et al., “An improved real coded hybrid genetic algorithm,” Computer Application, vol. 26(8), pp. 1959-1962, 2006.
- [17] Tengyue, M., Zhengquan, X., Rui H., “Efficient Constellation Design Based on Improved Non-dominated Sorting Genetic Algorithm-II,” Journal of Computers, vol. 7(6), pp. 1337-44, 2012.
- [18] Deep K., Thakur, “A new mutation operator for real coded genetic algorithms,” Applied Mathematics and Computation, vol. 193(1), pp. 211-230, 2007.
- [19] Ling S.H., Leung F. H. F., “An improved genetic algorithm with average-bound crossover and wavelet mutation operations,” Soft Computing, vol. 11(1), pp. 7-31, 2007.
- [20] Xiaoping, Z., Yanfei, C., Yongming, L., “A decimal adaptive parallel genetic algorithm,” Journal of Southwest National University (natural science), vol. 33(2), pp. 213-221, 2007.



**Jianmin Zhu** received bachelor degree in 1991 and master degree in 1994 both from Luoyang Institute of Technology, and received doctor degree in 2000 from Huazhong University of Science and Technology. Now he is a professor in University of Shanghai for Science and Technology. His main research directions are precision measurement technology, intelligent control of mechatronic systems.

**Donger Zhou** was born 1989. He is a Master candidate in the College of Mechanical Engineering at University of Shanghai for Science and Technology. His research areas are intelligent control of mechatronic systems.

**Fucai Li** was born 1986. He is a Master candidate in the College of Mechanical Engineering at University of Shanghai for Science and Technology. His research areas are intelligent control of mechatronic systems.

**Tingting Fu** was born 1988. She is a Master candidate in the College of Mechanical Engineering at University of Shanghai for Science and Technology. His research areas are intelligent control of mechatronic systems.