An Improved GA with Random Gene Crossover and Polyploidy Strategy

Xin-xin CAO¹ and Hai-yan QUAN¹

¹Faculty of Information and Automation, Kunming University of Science and Technology, Kunming, 650500, China

Keywords: Genetic algorithm, Real-coded, Polyploidy, Crossover on random gene.

Abstract. A kind of improved genetic algorithm with real-coded crossover on random gene and polyploidy strategy (RGPGA) was proposed. Based on the term of polyploidy, The paper adopted polyploidy which composed of optimal chromatid, reserved chromatid and mutant chromatid, and real-coding to improve the diversity of population; the roulette strategy was employed to the selection operation; and a kind of crossover on random gene was proposed. At part 4, RGPGA was applied to solve benchmark problems, and compared with the standard genetic algorithm. The results show that RGPGA not only keeps the diversity of population, represses the premature convergence effectively, but also reduces the algorithm complexity, improves the precision, which can find global optima of the complex high-dimension functions with higher precision.

Introduction
Genetic algorithm is an algorithm model simulated by Darwinian genetic selection and the natural elimination process of biological evolution, first proposed in 1975 by J. Holland professor at the University of Michigan [1]. Its main feature is the search does not rely on gradient information, and does not require the optimized object has continuous, derivable, singlet and other characteristics.[2-3] It is especially suitable for processing the complex and nonlinear problems which traditional search methods difficult to solve, it can be widely used in combinatorial optimization, machine learning, adaptive control, and artificial life [4-7].

Seeking the minimum (maximum) of a function, can be described by the following mathematical model:

\[
\begin{align*}
\min f(X) & \text{OR} \\
\max f(X) & \\
\text{s.t. } & \ X \in \mathbb{R}^n
\end{align*}
\]  

where, \(X = \{x_1, x_2, \cdots, x_i, \cdots x_n\}\) is decision variables, \(f(X)\) is object function, and,

\[
\bar{x} \leq x_i \leq \bar{x}, \quad i = 1, 2, \cdots, n.
\]

where, \(x, \bar{x}\) are constants, upper and lower limits of the decision variables respectively.

However, basic genetic algorithm still have premature phenomenon [8-11]. To prevent trapped in local optima, the paper propose the concept of polyploidy, increasing the diversity of the individuals. Secondly, the search process is done through random genetic cross and polyploidy strategy.

Crossover on Random Gene
First, search process of this algorithm is done through random genetic cross.

Genetic algorithm usually select two individuals with a certain probability to do crossover, such as dingle-point crossover, multiple point crossover or uniform crossover. Linear cross [12]is also a common operation, that all the parent individuals cross with one factor. So the offspring are located the lines where the parent on, it’s possible to produce unfeasible solution, and easy to trap in local optima.
In the paper, a population is composed of \(m\) individuals that each of them is composed of \(n\) genes. The probability of randomly selected genes is \(p\). So the parent individuals can be expressed as:

\[
\{x_1^r, x_2^r, \ldots, x^n_r\}, \quad X_r = \{x_1^r, x_2^r, \ldots, x^n_r\}, (1 \leq r, s \leq m).
\]

The steps of crossover on random gene as follows:

1) Define the number \((k)\) of genes of each chromosome as,

\[
k = p \times n \tag{3}
\]

Where \(p\) is the probability of selecting genes; \(n\) is the number of genes in each chromosome.

2) Select the genes from each chromosome as \(k C_n\).

3) If the selected genes are \(x^i_r(t)\) and \(x^i_s(t)\); define the genes after cross \(x^i_r(t+1)\) and \(x^i_s(t+1)(1 \leq i \leq n)\) as,

\[
\begin{align*}
    x^i_r(t+1) &= \lambda x^i_r(t) + (1-\lambda)x^i_s(t), \\
    x^i_s(t+1) &= \lambda x^i_s(t) + (1-\lambda)x^i_r(t). 
\end{align*} \tag{4}
\]

Where \(\lambda\) is the random number between \([0,1]\), and \(\lambda\) of every cross operation is different.

**Polyplody Strategy**

Secondly, refer to the concept of polyploidy in biology, which contains more than three chromosomes in one individual. The polyploidy constituted by genes which carried different function. Optimal chromatid, reserved chromatid and mutant chromatid have different effection:

1) Optimal chromatid control the direction of search to;

2) Reserved chromatid retain the information of parent individual, and keep effective information not to be destroyed;

3) Mutant chromatid increase the diversity of individual, and makes the algorithm can escape from the local optima.

Calculate the value of every chromatid after crossover, polyploidy strategy save the chromatid as the following cases

Case 1: if the value of chromaitd is superior to the optima, then replaces the optimal. The reserved chromatid and mutant chromatid keep to retaining the information of parent individuals, and making the optimal information not destroyed;

Case 2: if the value of chromatid is inferior to the optima, the optimal chromatid keeps unchangeable to insure the direction of search unchanged. Randomly choose an chromatid to replaces the reserved chromatid, and another chromatid replaces the mutant chromatid after mutation.

If the probability of mutation is \(pm\), \(x^i_j(t)\) is gene of parent individual, define the genes \(x^i_j(t+1)\) after mutation as:

\[
x^i_j(t+1) = \begin{cases} 
    x^i_j(t) + (x_{max} - x_{min}) \times \text{rand} & \text{if } \text{rand’} < pm, \\
    x^i_j(t) & \text{else}.
\end{cases} \tag{5}
\]

where \(x_{max}\) is the upper bound, \(x_{min}\) is the lower bound. \(\text{rand}, \text{rand’}\) are the random numbers between \([0,1]\).
An Improved GA with Random Gene Crossover and Polyploidy Strategy

According to the basic idea of above, the following is a detailed description of improved genetic algorithm.

In this paper adopt real coding [13,14] to reduce the complexity of the algorithm and roulette selection. But the optimal chromatid not involve the selection, to prevent individuals with high fitness values to be eliminated. Crossover probability determined by experiment, the results of different probabilities were discussed below, then the offspring will be choose according the polyploidy strategy. The steps are as follows:

step1: Set the control parameters, initialize the population, set the number of iteration \( t=0 \), the largest iteration is \( T \).
step2: Calculate the fitness of all individuals, save the best as optimal chromatid.
Step3:If \( t<T \), continue, otherwise export the optima, finish.
Step4:Random gene crossover, generate new chromosomes.
Step5:calculate the value of new chromosomes.
Step6: Analyzing the fitness of new chromosomes and parent chromosomes, save the new population according polyploidy strategy.
Step7:If the number of new individuals is less than the number of individuals, go to step 4; otherwise go to step 2, update the new population.

Experiments

In this experiments,6 benchmark functions are used. All the experiments are executed in Matlab. The dimension of the test function \( f_1, f_2 \) is 10, \( f_3 \) is 20, \( f_4, f_5, f_6 \) is 2. The number of individuals is 70. Each function was repeated 50 times with the randomly-initialized seeds.

\[
\begin{align*}
    f_1 &= \sum_{i=0}^{n} x_i^2 - 10 \cos(2\pi x_i) + 10, \ x_i \in (-5.12, 5.12); \\
    f_2 &= -20 \exp(-0.2 \sqrt{\sum_{i=1}^{n} x_i^2}) - \exp \left(-\frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_i)\right) + 20 + e, \ x_i \in (-32, 32); \\
    f_3 &= \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \frac{\cos(x_i)}{\sqrt{i}} + 1, \ x_i \in (-5.12, 5.12); \\
    f_4 &= -\cos(x_1) \cdot \cos(x_2) \cdot \exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2) + 1, \ x_i \in (-100, 100); \\
    f_5 &= \sum_{i=1}^{n} [x_i^2 - \cos(18x_i)] + 2, \ x_i \in (-1, 1); \\
    f_6 &= 1 - \frac{\cos(12\sqrt{x_i^2 + x_j^2}) + 1}{2} + \frac{x_i^2 + x_j^2}{2}, \ x \in (-5.12, 5.12).
\end{align*}
\]

We chose the different parameter \( p \) to test. The result is displayed respectively in Fig.1~Fig. 6.
Figure 1. Evolution of mean best value for $f_1$.

Figure 2. Evolution of mean best value for $f_2$.

Figure 3. Evolution of mean best value for $f_3$. 
Figure 4. Evolution of mean best value for $f_4$.

Figure 5. Evolution of mean best value for $f_5$.

Figure 5. Evolution of mean best value for $f_6$. 
Then we compare this a GA with random gene crossover and polyploidy strategy (RGPGA) with the scripture genetic algorithm (SGA). The optima, average value and variance are listed in Table 1.

### Table 1. Comparisons between RGPGA and SGA.

| No. | Optima | AVG   | σ   | Dim. |
|-----|--------|-------|-----|------|
| SRGA | RGPGA  | SRGA  | RGPGA | SRGA  | RGPGA |
| 1   | 0.5214 | 2.1·10^{-6} | 2.7435 | 1·10^{-4} | 1.2036 | 7.4·10^{-3} | 10 |
| 2   | 0.0473 | 2.3·10^{-6} | 0.0881 | 1·10^{-4} | 0.0010 | 5.7·10^{-5} | 10 |
| 3   | 0.0010 | 4.3·10^{-9} | 0.00739 | 1·10^{-7} | 0.0083 | 7.2·10^{-7} | 20 |
| 4   | 0.9999 | 0 | 0.9997 | 1·10^{-9} | 4·10^{-12} | 5.4·10^{-5} | 2 |
| 5   | 0.0094 | 0 | 0.1019 | 1·10^{-11} | 0.0107 | 1.8·10^{-9} | 2 |
| 6   | 0.0513 | 0 | 0.0688 | 1·10^{-9} | 0.0117 | 5.8·10^{-8} | 2 |

According to the result, RGPGA can find optima or close-to-optimal solutions, its solution quality is stable, and its search is efficient.

### Conclusions

In the paper, a GA with random gene crossover and polyploidy strategy (RGPGA) is proposed. In the experiments, RGPGA is applied to solve the 6 benchmark problems, and is compared with SGA. The results demonstrate that the search of RGPGA is efficient, the solution quality of RGPGA is stable, and can find optima or close-to-optimal solutions of the complex functions with high-dimension.

### Acknowledgement

This research was financially supported by National Natural Science Foundation of China (No. 41364002), State Science Foundation of Yunnan (No. 2009ZC048M). The authors are graceful for the support of the grants.

### References

[1] Holland J.H. Adaptation in Nature and Artificial Systems [M]. Cambridge MIT Press, 1975.

[2] Zhou M., Sun S.D. The Theory and Application of Genetic Algorithm. [M]. Beijing: National Defend Industy Press, 1999. 6

[3] Chen G.L., Wang X.F., Zhuang Z.Q., Wang D.S. Genetic Algorithm and Application [M]. Beijing: Posts & Telecom Press,1996.

[4] Nebro A.J., Durillo J.J., Luna F., et.al. A Cellular Genetic Algorithm for Multi-object Optimization [J]. International Journal of Intelligent Systems, 2009, 24: 726-746.

[5] Al-Naqi A., Erdogan A.T., Arslan T. Adaptive three-dimensional cellular genetic algorithm for balancing exploration and exploitation processes [J]. Soft Computing, 2013, 17(7): 1145-1157.

[6] Ma Y.J., Yun W.X. Research progress of genetic algorithm [J]. Application research of computers, 2009, 29(4): 1201-1210.

[7] Zhang L.P., Chai Y.T. Actuality and developmental trend for genetic algorithms [J]. Information and Control, 2001, 30(6): 531-536.
[8] Wang M.L., Gao X.G., Wang D.W. Optimization of genetic algorithm control parameters [J]. Computer Engineering, 2003, 29(5): 43-41.

[9] Pholdee N., Bureerat, S. Hybrid real-code population-based incremental learning and approximate gradients for multi-objective truss design [J]. Engineering Optimization, 2014, 46(8): 1032-1051.

[10] Shen X.N., Guo Y., Chen Q.W., etc. Multi-objective optimization genetic algorithm keeping diversity of population [J]. Control and Decision, 2008, 23(12): 1435-1440.

[11] Qi C., Wang D.X., Han Y. Performance comparison analysis of multiple genetic algorithms for function optimization [J] Journal of Liaoning University of Technology(Nature Science Edition).

[12] Cao K., Chen G.H., Jiang H. Guided self-adaptive evolutionary genetic algorithm [J], Journal of Electronics & Information Technology, 2014, 36(8): 1884-1890.

[13] Bessaou M., Slarry P. A genetic algorithm with real-value coding to optimize multimodal continuous functions [J]. Struct Multidisc Optim, 2001, 23(1): 63-74.

[14] Baskar S., Subbaraj P., Rao M.V.C. Hybrid real coded genetic algorithm solution to economic dispatch problem [J]. Computers and Electrical Engineering, 2003, 29: 407-419.