Proceedings of the 2007 International Conference on Intelligent Systems and Knowledge Engineering, Chengdu, 15-16 October 2007.

Three Real-Coded Genetic Algorithms with New Mutation Operators

Guoli Zhang[1] Guangquan Zhang[2] Jianwei Ma[3] Chunlei Zhou[4]

[1]Department of Mathematics and Physics, North China Electric Power University, Baoding 071003, P.R.China
[2]Faculty of Information Technology, University of technology Sydney, NSW 2007, Australia
[3]Information Technology Division, China Welfare Lottery Issuing and Management Centre, Beijing 100054 P.R.China
[4] China Electric Power Information Center, Beijing 100761, P.R.China

Abstract

Local search is mainly implemented by the reproduction and crossover operation, while global search is assured by the mutation operation in conventional genetic algorithm. In order to enhance the global search ability, three new mutation operators are proposed based on the idea that big change into small and small change into big for gene bit selected at random. The experimental verification shows that the proposed new genetic algorithms with new mutation operators are effective in seeking for the global optimal solutions.

Keywords: Function optimization, Genetic algorithm, Real-coded, Mutation operator.

1. Introduction

Conventional optimization methods suffer from the local optimality problem and request the function having good characteristic, such as differentiable, continuous etc., this limit application of many traditional optimization methods in real-world problem. Recent years, stochastic optimization techniques such as simulated annealing (SA), genetic algorithm (GA), and evolutionary algorithm (EA) have been given much attention by many researchers due to their abilities to seek for the near global optimal solution and no restrictions on the nature of the function. Therefore these methods have been used in widely fields [1]-[7].

Here, we focus on genetic algorithm (GA) which was introduced by John Holland in early seventies as a special technique for function optimization. GA has three operators of reproduction, crossover and mutation. Reproduction is devised to inherit good-working individuals from generation to generation. In the crossover phase, two individuals in a present population are randomly selected, and they exchange their bits for the crossing site determined by another random number. During the mutation process, every bit has an equal probability to be replaced by its complement number. In conventional genetic algorithm (CGA), local search is mainly implemented by the reproduction and crossover operation, global search is assured by the mutation operation.

The performance of CGA precedes mainly traditional optimization method in aspect of global search, but CGA suffers also premature convergence problem and expensive computing time, therefore many searchers proposed varieties method to solve these problems and have obtained an amount of achievements [8]-[12]. It is easy to see that the purpose of these methods is all centralized in three aspects, i.e., decrease computing burden, speed up convergence rate and enhance global search capability. CGA use binary code, which needs a lot of time to code and decode, some researchers proposed genetic algorithms using real numeral code to decrease computing burden. However the mutation operator is less for real-coded genetic algorithm. Common mutation operation is to create a random number, then add it to corresponding original value or use it to replaces of correspondence values in the individual.

Our idea is to seek for new mutation operators in order to increase global search capability and convergence rate. According to this idea, this paper proposes three new mutation operators, further get three new real-coded GAs, the main ideas and steps are described in section 3.

The rest of this paper is organized as follows. For clear, Section 2 describes briefly the global minimization problem described by Xin Yao and Yong Liu [13]. The new mutation operators and the main search steps of new GAs are discussed in Section 3. Section 4 presents the experimental verification of the proposed new algorithms, comparing the experimental results of the conventional genetic algorithm. Finally, the conclusions and future work are given in Section 5.

2. Function optimization

ISBN: 978-90-78677-04-8

Consider the global minimization problem described by Xin Yao and Yong Liu[13] for the purpose of development of new search algorithm. According to Yao and Liu, the problem can be formalized as a pair of real valued vectors (S, f), where $S \subseteq R^n$ is bounded set on R^n and $f: S \to R$ is an n-dimensional real-valued function. f needs not to be continuous but it must be bounded. The problem is to find a point $x_{\min} \in S$ such that $f(x_{\min})$ is a global minimum on S. In other words, it is required to find an $x_{\min} \in S$ such that

$$\forall x \in S: f(x_{\min}) \leq f(x) \qquad (1)$$

3. Real-coded genetic algorithm with new mutation operators

3.1. Three new mutation operators

3.1.1. The mutation operator I

Main effect of mutation operation is to enhance diversity of the population. In general, real-coded mutation operation is to add a random number to corresponding original value or reproduce a random number which replaces of correspondence values in the individual. These mutation operations lack of nature evolutionary foundation. In binary system genetic algorithm, the gene is represented by numeral 0 and 1. New gene changes according to 1 change into 0, and 0 change into 1. In decimal system genetic algorithm, the gene is represented by numeral 0,1,...,9, so that mutation operation should be change numeral 0,1,...,9, and the mutated gene should have biggish change with original gene. Based on the above idea, a new mutation operation that big numeral (gene) selected at random change into small one and small numeral change into big one is proposed as follows:

Let parent $x^i = (x_1, x_2, \cdots, x_n)$, the offspring $x^{i'} = (x_1', x_2', \cdots, x_n')$ is created by the mutation operation according to following step:

Assume the r-th components x_r of parent x^i is represented by

$$x_r = y_1 y_2 \cdots y_j \cdots y_l \qquad (2)$$

Then after mutation, r-th components x_r' of the offspring can be calculated by the formula (3)

$$x_r' = y_1' y_2' \cdots y_j' \cdots y_l' \qquad (3)$$

where

$$y_j' = \begin{cases} y_j, & y_j \text{ is not mutation bit} \\ 9 - y_j, & y_j \text{ is mutation bit} \end{cases} \qquad (4)$$

The above mutation operator reflects the ideas of big numeral change into small and small change into big. However the new gene after mutation should be uncertain in nature, based on this idea the other two mutation operators are proposed in next section.

3.1.2. The mutation operator II and III

The other two new mutation operators can be defined respectively by formula (5) and (6) as follows:

$$y_j' = \begin{cases} y_j, & y_j \text{ is not mutation bit} \\ \alpha_j(9 - y_j), & y_j \text{ is mutation bit and } y_j \geq 5 \\ \beta_j(9 - y_j), & y_j \text{ is mutation bit and } y_j < 5 \end{cases} \qquad (5)$$

where $\alpha_j \in [0,1]$, $\beta_j \in [\frac{1}{2}, 1]$ are random numbers.

$$y_j' = \begin{cases} y_j, & y_j \text{ is not mutation bit} \\ \alpha_j(9 - y_j), & y_j \text{ is mutation bit} \end{cases} \qquad (6)$$

where α_j denotes a normal distribution random number with mean 0 and standard deviation σ, if $y_j' > 9$, let $y_j' = 9$. According to the mutation probability, the every numeral (gene) bit of the each component of the individuals is selected randomly to execute the mutation operation, i.e. the mutated numeral is calculated by the mutation formula (6).

3.2. Real-coded genetic algorithms with new mutation operators

Three new genetic algorithms (NGA) can be obtained by using one of three novel mutation operators above, the main search steps of NGA are as follows:

Step1 create randomly the initial population including N individuals.

Step2 calculate fitness f_i, $i = 1, 2, \ldots, N$ for every individual x^i.

Step3 implement elitist strategy, i.e., reserve best individuals in proper proportion; only one best individual is reserved for every generation in latter experiment.

Step4 reproduction is operated by roulette wheel, the reproduction probability of the individual x^i is calculated by following formula (7)

$$p_i = \frac{f_i}{\sum_{i=1}^{N} f_i}, i = 1, 2, ..., N \qquad (7)$$

Where $p_i \in [0,1]$ specifies the reproduction probability of individual x^i.

Step5 let P_c be the crossover probability, crossover operation is processed according to following step: select $[\frac{NP_c}{2}]$ pair of parents randomly; for every pair of selected parent, use two-point crossover to produce two new individuals by linear combination; replace parents by new offspring.

Step6 mutation operation is realized by one of three new mutation operators proposed above.

Step7 if the stopping criterion is satisfied, go to Step8; else go to Step2.

Step8 output the best individual.

4. Experimental verification

4.1. Test functions

In order to test the effectiveness of the proposed new algorithms in aspect of the global search capability and the convergence rate, the CGA and new algorithms with three new mutation operators are implemented. Four functions were chosen from the set of 23 benchmark functions[13] and re-numbered as f_1 to f_4, in which function f_1 and f_2 are typical unimodal functions; f_3 and f_4 are multimodal functions. These functions are challenging to every search algorithm. The definitions of these functions are depicted in Table 1.

	Test function	Dimension	Domain	Minimum value f_{min}
1	$f_1 = \sum_{i=1}^{n} x_i^2$	10	$[-100, 100]^{10}$	0.0
2	$f_2 = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	10	$[-100, 100]^{10}$	0.0
3	$f_3 = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos 2\pi x_i) + 20 + e$	10	$[-32, 32]^{10}$	0.0
4	$f_4 = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1$	10	$[-600, 600]^{10}$	0.0

Table 1: Definitions of test benchmark functions.

4.2. Test results

The experimental environment is Visual C++, and the computational results of f_1 to f_4 are demonstrated in Table 2. The number of iteration for every algorithm is 5000, the size of population is 100, the reproduction probability is 0.2, the mutation probability is 0.2, and the crossover probability is 0.6. Where NGA1, NGA2, NGA3 are Genetic Algorithms by using respectively one of three new mutation operators proposed above in this paper.

From the results in Table 2, it is easy to see that the NGA can obtain better function values in most computations. With the different standard deviations σ, can obtain different optimal function values. Therefore, better results can be achieved by adjusting the value of σ.

5. Conclusions and further study

Three real-coded GA with new mutation operators are proposed in this paper. No matter what both unimodal and multimodal functions, the proposed NGA can obtain better result. With the different test functions, how to choose the appropriate mutation operator and the value of σ will be discussed in future work.

Acknowledgment

This research is supported by the Homecoming Fund of North China Electric Power University (20514001).

References

[1] C.H. Im, H.K. Jung, and Y.J. Kim, Hybrid genetic algorithm for electromagnetic topology optimization, *IEEE Transactions on Magnetics*, 39: 2163-2169, 2003.

	CGA	GA1	GA2	GA3 (σ=0.5)	GA3 (σ=2)	GA3 (σ=3)
f_1	1.21785	0.487483	1.15483	1.75537	0.0216	0.0259977
f_2	5.04354	0.98733	0.25808	0.519881	0.16655	0.6239
f_3	2.3717	3.49073	1.33937	1.33863	0.625613	0.198802
f_4	0.135901	0.0495491	0.0845253	0.0745425	9.19208e-006	0.115418

Table 2: The experimental results.

[2] E. Gil, J. Bustos, and H. Rudnick, Short-term hydrothermal generation scheduling model using a genetic algorithm. *IEEE Transactions on Power System*, 18: 1256-1264, 2003.

[3] J.M. Arroyo, A.J Conejo, A parallel repair genetic algorithm to solve the unit commitment problem, *IEEE Transactions on Power System*, 17: 1216-1224, 2002.

[4] T.Yalcinoz and H. Altun, Power economic dispatch using a hybrid genetic algorithm, *IEEE Power Engineering Review*, 21: 59-60, 2001.

[5] R.M. Ramos, R.R Saldanha, R.H.C. Takahashi, and F.J. Moreira, The real-biased multiobjective genetic algorithm and its application to the design of wire antenna, *IEEE Transactions on Magnetic*, 39: 1329-1332, 2003.

[6] W. Ziomek, M. Reformat and E. Kuffel, Application of genetic algorithms to pattern recognition of defects in GIS, *IEEE Transactions on Dielectrics and Electrical Insulation*, 7:161-168, 2000.

[7] S. Chakravarty, R. Mittra and Application of the micro-genetic algorithm to the design of spatial filters with frequency-selective surfaces embedded in dielectric media, *IEEE Transactions on electromagnetic Compatibility*, 44: 338-346, 2002.

[8] F. Herrera, M. Lozano, Gradual distributed real-coded genetic algorithms, *IEEE Transactions on Evolutionary Computation*, 4: 43-63, 2000.

[9] E. Alba, F. Luna, and A.J. Nebro, Parallel heterogeneous genetic algorithms for continuous optimization, *International Parallel and Distributed Processing Symposium Proceedings*, pp. 22-26, 2003.

[10] S. Choi and C. Wu, Partitioning and allocation of objects in heterogeneous distributed environments using the niched Pareto genetic-algorithm, *Software Engineering Conference, Asia Pacific* , pp. 322-329, 1998.

[11] E. Alba, and M. Tomassini, Parallelism and evolutionary algorithms, *IEEE Transactions on Evolutionary Computation*, 6: 443-462, 2002.

[12] T. Kuo and S.Y. Hwang, A genetic algorithm with disruptive selection, *IEEE Transactions on Systems, Man and Cybernetics*, Part B, 26: 299-307, 1996.

[13] X. Yao and Y. Liu, Evolutionary Programming Made Fast, *IEEE Transactions on Evolutionary Computation*, 3: 82-102, 1999.