Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Environmental Sciences 11 (2011) 346 – 350

Procedia Environmental Sciences

A Genetic Algorithm Based on Combination Operators

Xunbo Shuai[a], Xiangguang Zhou[b]

Research Institute of Petroleum Exploration & Development-Langfang, PetroChina
Langfang, China
[a]shuaixb69@petrochina.com.cn, [b]sxblffy@126.com

Abstract

Dual operator and inverse operator are defined as two new genetic operators respectively in this paper. Then a genetic algorithm based on inverse and dual combination operator was designed to overcome the defect of genetic algorithm in local searching, which combined with uniform crossover. The genetic algorithm is proved to be convergent. Experiments show that it has better searching performance and it has good reference to string-coded genetic algorithm in solving nonlinear optimization problems.

*Keywords:*Genetic algorithm; local searching; global searching; dual operator; inverse operator

1. Introduction

Genetic algorithm has robustness and general optimization capability in solving some complex problems. However, for large-scale problems or problems of high precision, it often fall into local optimum [1,2]. In recent years, using genetic operators to improve the performance of genetic algorithm becomes a research focus [3-7]. Studies show that inverse genetic operator can improve genetic algorithm's local searching performance to some extent, but it destroys the population diversity. Threshold of inverse operator ε is often given based on the experience of solving problems [5,6].

In this paper, from the perspective of simulating biology gene order and allele, we propose inverse and dual operators which improve the genetic algorithm local searching performance. Combined with uniform crossover, a genetic algorithm based on dual and inverse combination operator is constructed. The genetic algorithm is proved to be convergent. Experiments show that it has better searching performance, and it can solve the conflict of local searching and global searching in genetic algorithm. The genetic algorithm has good reference to string-coded genetic algorithm in solving nonlinear optimization problems.

2. Inverse and dual combination operator

2.1. The Operator

In genetics, every gene of chromosomes contains a lot of information, and genes divide into dominant genes and recessive genes. Recessive gene can change the performance of biology in some specific cases. In addition, the arrangement structure of mater in real world can impact its performance. For example, different arrangement of carbon atoms can form distinct diamond and graphite.

In bit string encoding genetic algorithm, chromosome is denoted as an ordered bit string genome, which contains little information. Inverse operator simulates the order of gene arrangement in biology chromosomes. It can improve local searching performance of genetic algorithm, but it damages the diversity of population, which is not conducive in global searching [5,6]. In genetic algorithm, J.Carig Potts thought the reason for premature convergence is the lack of effective allele [8]. However, the allele in genetics is a very interesting concept, which fully reflects the diversity of genes. It inspired us to simulate allele of biology chromosome, and dual operator is proposed to realize searching of

1878-0296 © 2011 Published by Elsevier Ltd. Open access under CC BY-NC-ND license.
Selection and/or peer-review under responsibility of the Intelligent Information Technology Application Research Association.
doi:10.1016/j.proenv.2011.12.055

allele group. Combined dual operator with inverse operator to construct a new operator named dual and inverse combination operator. Further search the recessive gene group of chromosome and approach the best recessive gene group, which can improve the local searching performance and reduce the damage on population diversity from inverse operator.

Definition 1: (Generalized Inverse Operator) If the antitone sequence genome of chromosome is meaningful, generalized inverse operator is the conversion from order genome of chromosome $s = a_1 a_2 ... a_{n-1} a_n$ to its inverse genome $s' = a_n a_{n-1} ... a_2 a_1$.

Definition 2: (Dual Operator) Each gene bit of genome $s = a_1 a_2 ... a_{n-1} a_n$ is replaced by its allele, to form a new allele group $s' = \beta_1 \beta_2 ... \beta_{n-1} \beta_n$.

Definition 3: (Dual and Inverse Combination Operator) genome A completes the following four steps:

(1)Use generalized inverse operators to produce genome S_R, and use dual operators to produce genome S_A;

(2) We use dual operators to S_R then produce genome S_R_A (or use generalized inverse operators to S_A, then produce genome S_A_R);

(3) Chromosome S_N=Max (S_R,S_A,S_R_A) or S_N=Max(S_R,S_A,S_A_R), namely, select the chromosome whose fitness is best from new generated genome.

(4) If the fitness of S_N is better than S then the position of S will be replaced by S_N. On the contrary, S_N will be eliminated.

For a chromosome, the inverse genome's allele group and the allele group's inverse are the same new chromosome. Dual and inverse combination operator can realize three effective searching for the chromosome, and give priority to select new chromosome. It can also improve local searching performance. Compared with the inverse operator [5,6], the dual and inverse combination operator reduces the damage to population diversity. The experiments later will prove the combined operator's performance.

2.2. Operator performance experiment and analysis

We will do experiments about local searching performance of the combination operator, and select typical funcitions.

Shuber function:

$f_1(x,y) = \{\sum_{i=1}^{5} i \bullet \cos[(i+1.0) \bullet x + i]\} \{\sum_{i=1}^{5} i \bullet \cos[(i+1.0) \bullet y + i]\} + 0.5[(x+1.42513)^2 + (y+0.80032)^2]$, in which, $-10 < x, y < 10$, we will calculate the minimum value of the function;

Step function: $f_2(x) = \sum_{i=1}^{n} [x_i + 0.5]^2$, in which, $x = (x_1, x_2, ... x_n) \in [-100,100]^n$, we will calculate the minimum value of the function;

Spherical function: $f_3(x) = \sum_{i=1}^{n} x_i^2$, in which, $x = (x_1, x_2, ... x_n) \in [-100,100]^n$, we will calculate the minimum value of the function;

Function f_1 has 760 minimal values, and the minimum is -186.34, and the function easily falls into the local minimum -186.34. Function f_2 is a multi-peak function, which is the common problem in evolutionary algorithms. Function f_3 is the single-peak function, which is regarded as the typical problem to test algorithm local search performance. For function f_2 and function f_3, n=10.

The experiment will improve the standard genetic algorithm. In each generation operation, at first, perform local searching to recessive genome with dual and inverse combination operator, and search the optimal population. Then use single point hybridization and mutation genetic operator to search the optimal solution for new population. Compared the optimum result of the new genetic algorithm (RAGA) with the experiment results of standard genetic algorithm (GA) and reverse genetic algorithm (RGA) is put forward in reference [6]. They can construct three experiments. We use binary string encoding, tournament selection strategy, hybrid probability p_c=0.45, mutation rate p_m=0.01, population size m=80. Randomly select 20 results from three experiments, and calculate the average value, second sample moment and the optimal value which has searched. Apply second sample moment to show the searching results' discrete degree. The experiment results are shown in TABLE I.

The analysis of experiment results are as follows:

The second sample moment: RAGA< RGA < GA. It shows that the experiment results distribution of RAGA and RGA are denser around average value than GA. The searching result of GA is prone to divergent. The searching results of RAGA are relatively concentrated than RGA.

Searching performance denoted as average value: GA<RGA< RAGA. It shows that the searching performance of RAGA and RGA is better than GA , and the searching performance of RAGA is better than RGA.

Performance of searching the optimal value: RGA< RAGA < GA. The performance of searching the optimal value of GA is better than RAGA and RGA, and the performance of searching the optimal value of RAGA is better than RGA.

TABLE I. EXPERIMENT RESULTS

Algorithm		Function		
		$f_1(x,y)$	$f_2(x)$	$f_3(x)$
GA	Average value	-185.721	14.184	14.535
	Second moment	0.683	17.854	15.821
	Best value	-186.664	7.632	9.547
RAGA	Average value	-186.128	10.026	11.682
	Second moment	0.079	2.885	2.186
	Best value	-186.641	8.076	9.613
RGA	Average value	-186.016	11.364	13.485
	Second moment	0.248	5.872	9.554
	Best value	-186.535	9.862	11.308

We can conclude that the performance of searching the optimal value of RAGA is poorer than GA. and the searching performance and searching result distribution of RAGA is better than RGA. The reason is that RAGA further improves the local searching performance and reduces the damage on population diversity. However, the population mode is rapidly concentrated in a certain extent which goes against global searching. In order to solve the problem, we fully use the combination operator's local searching performance and select the uniform operator which has high probability to construct genetic algorithm in subsection 3.1.

3. Genetic algorithm based on dual and inverse operator

Theory analysis and experiment results show that the dual and inverse operator can effectively improve population searching performance, but it also make the population mode rapidly concentrated and go against global searching. Study found that some genetic operators have good global searching performance, therefore, combine local searching with global searching can perform searching well [9]. Uniform cross operator destroys mode with larger probability, and it can search modes which the point hybridization cannot search. The hybrid mode is a disadvantage for local searching, but it is very effective for global searching [5,6,9]. Uniform crossover operator with greater probability is used to "destroy" the population, then global search the population. To generate more new chromosomes and the next generation dual and inverse operator has more extensive searching space; the mutation probability is set a little larger. Combined with their advantages, which make the RAGA has good global searching performance and local searching performance, and improve the algorithm's overall searching performance.

3.1. The Algorithm

In each generation searching process, use dual and inverse operator to population, and improve the population's overall performance. Select strategy adopts the selection sort operator which can maintain the diversity of population and make the algorithm has good global convergence. Uniform crossover operator with greater probability is used to "destory" centralized mode population. Meanwhile, in order to generate more new chromosomes and the next generation dual and inverse operator has more extensive search space, the mutation probability is set a little larger. To take into account the global searching and local searching, experiments show that uniform crossover probability is between 0.4 and 0.85, and point mutation rate is between 0.03 and 0.1. In searching process, if there is no new optimal solution in a long time, we can dynamically increase the uniform crossover probability to "destory" the centralized mode population.

Genetic algorithm based on dual and inverse combination operator is described as follows:

Step 1: Based on constraint condition, initial population is generated using string encoding, and the global optimal solution is initialized to p;

Step 2: Using dual and inverse combination operator to strengthen local searching, and calculate the fitness of each chromosome in new populations.

Step 3: Using selection sort algorithm to sort the group, if the optimal solution of this generation is better than p, then p is replaced by the optimal solution.

Step 4: If the crossover probability need to increase, then dynamically adjust the crossover probability.

Step 5: Use uniform crossover operator and mutation operator for global searching and local searching.

Step 6: Judge whether satisfies algorithm end condition, if satisfy, then output the final solution. Otherwise, go to step 2.

3.2. Experiment analysis

In order to verify the genetic algorithm based on dual and inverse operator has good local searching performance and more efficiency in searching, we solve functions f_1,f_2,f_3.

Randomly select 10 times best values, average values and second sample moment, and compared with standard genetic algorithm, the results are shown in TABLE II.

f_3 can search the minimum 0.0. f_1 and f_2 can search the best values in the following: f_1 (-1.42512843，-0.80032110) = -186.73090883 and f_2 (-0.390625，-0.48828125，-0.48828125，-0.83007813，-0.537109375，-0.537109375，-0.48828125， -0.390625，-0.390625，-0.537109375)＝0.149383545.

TABLE II. EXPERIMENT RESULTS

Algorithm		Function		
		$f_1(x,y)$	$f_2(x)$	$f_3(x)$
GA	Best value	-186.7294	8.3544	10.6451
	Average value	-185.8172	13.6609	14.5675
	Second moment	0.6743	17.9138	16.1244
RAGA	Best value	-186. 7309	0.1494	0.0
	Average value	-186.6526	0.6101	0.0603
	Second moment	0.0243	0.1246	0.0098

We can see from the experiment results that the genetic algorithm based on dual and inverse operator has good local searching performance and more efficiency in searching than standard genetic algorithm. Compared with experiment 1, the algorithm improves the local searching performance. Use uniform crossover operator with greater probability to overcome the influence of combination operator on global searching performance, and the algorithm has better searching performance.

4. Conclusion

In this paper, the dual and inverse combination operator can effectively improve local searching performance of genetic algorithm. Combined with uniform crossover operator with greater probability can solve the conflict between global searching and local searching. The RAGA has strong local searching performance and better searching performance. However, compared with standard genetic algorithm, the algorithm's running time is slightly longer, which is the spirit of "no free lunch" theorem [10]. Simulation results show that the algorithm can be applied in the nonlinear complex optimization problems whose genome can reverse. The probability of uniform crossover operator can dynamically adjust based on specific questions and the intensity of local searching.

5. References

[1] Kemin Zhou, Yunchang Hu. Improvement of computational efficiency for genetic algorithms[J]. Control Theory & Applications, 2002, 19(5): 812-814.

[2] Hong Ge, Zongyuan Mao. The Analysis of the Local Search Efficiency of Genetic Neural Networks and the Improvement of Algorithm [C]. In processing of the 4th World Congress on Intelligent and Automation. Hefei, China: Press of East China University of Science and Technology, 2002.

[3] Weiqing Xiong, Ping Wei and Jieyu Zhao. An Adjust Operator of Genetic Algorithms[J]. Mini-micro Systems, 2003, 24(3): 531-533.

[4] Meiying Liao, Yongjun Zhang. Study on the Effect of Cataclysm Operator on Genetic Algorithm[J]. Computer Engineering and Applications, 2005,41(13): 54-56.

[5] Shunan Ma, Xunbo Shuai and Fengxue Cao. An Optimization Combination Genetic Algorithm based on Inverse Operator[J]. Application of Electronic Technique,2006, 32(6): 19-21.

[6] An Gong, Xunbo Shuai and Shunan Ma. An Optimization Combination Genetic Algorithm Based on Reverse by Threshold Operator[J]. Computer Simulation, 2006, 23(9): 175-178.

[7] Shaoyan Wu, Zhuoqun Xu. A Heuristic Policy for Constructing Crossover in Genetic Algorithms[J]. Chinese Journal of

Computers, 1998, 21(11): 1003-1008.
[8] Potts J C, Terri D G, Surya B Y. The development and evolution of an improved genetic algorithm based on migration and artificial selection [J]. IEEE Transactions on SMC (S0018-9472), 1994, 24(1): 73-86.
[9] Wen Zhang, Xiang Li. On Optimum Combinations For Genetic Operations[J]. Journal of Numerical Methods and Computer Applications, 2005, 26(3): 208-214.
[10] Wolpert D H, Macready W G. No free lunch theorems for optimization [J]. IEEE Transactions on Evolutionary Computation (S1089-778X), 1997, 1(1): 67-82.