

2012 AASRI Conference on Computational Intelligence and Bioinformatics

## The Research on Reliability Optimization of Software System Based on Niche Genetic Algorithm

Qian Yuexia, Gu Weijie \*

*Changzhou Institute of Mechatronic Technology, Changzhou 213164, China*

---

### Abstract

To deal with the problem of low efficiency and low convergence speed in searching the global optimum, Niche Genetic Algorithm was used in reliability optimization of software system. Linear programming works with multi-variable and multi-constraint were solved successfully. And also, the searching performance of the genetic algorithms was improved by introducing the stochastic tournament model, the uniform schema crossover operator and the non-uniform mutation in the genetic algorithm. The simulation results show that the Niche Genetic Algorithm can resolve the multi-module complex software system's reliability allocation effectively, which can improve the computing speed and the resolution quality effectively.

2012 Published by Elsevier B.V. Selection and/or peer review under responsibility of American Applied Science Research Institute Open access under [CC BY-NC-ND license](#).

*Keywords:* Genetic Algorithm; Niche Genetic Algorithm; Software Reliability

---

### 1. Introduction

The original of the Genetic Algorithm is the computer simulation research on biological systems. It was put forward by professor J.H.Holland in the university of Michigan in 1960s [1]. Genetic Algorithm is a random search technique based on imitating the natural selection, species evolution and population genetics. And it is particularly suitable for getting the optimal solution of combinatorial optimization problems [2]. In recent years, Genetic algorithm has been applied in many fields and has become an important method of researching artificial intelligence and information processing technology [3]. The reliability optimization of

---

\* Corresponding author. Tel.: 13775205406.  
E-mail address: [gwjysu@163.com](mailto:gwjysu@163.com).

software system, which is also called reliability allocation problem of software system, is to find an optimal reliability allocation scheme of software system under the conditions of a certain resource constraints. It makes the system obtain the highest reliability. Generally speaking, there are a large number of local extreme points. It is difficult to solve the global optimal solution accurately. Many scholars have achieved good results in solving the combinatorial optimization problem with intelligent optimization algorithms. But it is still rare to see the research on reliability allocation problem of software system with Niche Genetic Algorithm [4].

The reliability optimization of software system was researched based on Niche Genetic Algorithm in this paper. And it was compared with other heuristic optimization algorithms. The theoretical analysis and simulation results show that Niche Genetic Algorithm is suitable for solving reliability allocation problem of software system. And also the operating efficiency and the solution quality are improved.

## 2. The analysis of Niche Genetic Algorithm

In biology, a niche refers to the organizational function or role in a specific environment. Species refer to the common features of organization. The organisms tend to live with each other with similar features and shapes [5]. The individual process of evolution is put in the same population repeatedly in traditional genetic algorithm. With the smoothly genetic exchanging, it is difficult for respective developing of the individual. So the diversity of population is reduced. To solve the problem, the individuals in the initial population are sorted by fitness value firstly. And the similar individuals evolve independently in the niche. The scale of sub-population is changing with the parent population. Let  $N$  be the scale of parent population,  $K$  be the scale of sub-population, we must have  $K=N(D)$ ,  $K$  is a function of  $D$ . It can be set firstly according to the characteristics of the problem.  $D$  is the variance of the individuals in a population while  $\sigma$  is a constant. When the diversity decreases,  $D$  decreases. When  $D$  is less than a threshold  $\sigma$ , the sub-population size will reduce to the minimum value 2 in order to stimulate the improvement of the diversity of the population.

## 3. The reliability allocation model of software system based on the module subsystem

The reliability can be represented as formula 1 for a software system which has  $n$  modules and  $m$  operations.

$$R(\lambda_1, \lambda_2, \dots, \lambda_n; \tau) = \exp\left(-\sum_{j=1}^n \sum_{i=1}^m p_i q_{ij} \lambda_j \tau\right) \quad (1)$$

In the formula,  $n$  stands for the number of module. The  $m$  stands for the number of operations.  $\tau$  stands for the running time.  $\lambda_j$  stands for the failure rate of module  $j$ .  $p_i$  stands for the probability of operation.  $q_{ij}$  stands for the running time proportion of module  $j$  in operation  $i$ .

The running time proportion of module  $j$  in the total task can be defined as formula 2.

$$\phi_j = \sum_{i=1}^m p_i q_{ij} \quad (2)$$

$\phi_j$  stands for occupied time of module  $j$  in the total task. Then we can get the reliability of module  $j$  in the time of  $\phi_j$ . As shown in formula 3.

$$R(\lambda_j; \phi_j \tau) = \exp\left(-\sum_{i=1}^m p_i q_{ij} \lambda_j \tau\right) = \exp(-\phi_j \lambda_j \tau) \quad (3)$$

Then, the reliability of module  $j$  in the time  $\tau$  can be represented as formula 4.

$$R(\lambda_i; \tau) = \exp(-\lambda_i \tau) = R(\lambda_i; \phi_i \tau)^{1/\phi_i} \quad (4)$$

Then, the reliability of the software system which has  $n$  modules can be represented as formula 5.

$$R(\lambda_1, \lambda_2, \dots, \lambda_n; \tau) = \prod_{j=1}^n \exp(-\phi_j \lambda_j \tau) \quad (5)$$

So the cost of software development can be got as formula 6.

$$TC(\lambda_1, \lambda_2, \dots, \lambda_n) = \sum_{j=1}^n C(\lambda_j) \quad (6)$$

The reliability of the software system can be got to the expression as formula 7.

$$R(\lambda_1, \lambda_2, \dots, \lambda_n; \tau) \geq \rho \quad (7)$$

In the formula,  $\rho$  stands for the reliability target of the software system with  $n$  modules and  $m$  operations. That is a combinatorial optimization problem with a linear function and multiple constraints. So Niche Genetic Algorithm is imported to solve the engineering problem.

#### 4. Reliability optimization of software system based on Niche Genetic Algorithm

The genetic algorithm which simulates biological niche genetic is produced by using the concept of niche genetic. The purpose of the algorithm is to form and maintain a variety of sub-populations and do parallel search in the search space. It is often used to solve multimodal function, multi-objective optimization and simulation of complex systems [6]. Niche Genetic Algorithm simulates the fittest principle of biological evolution by maintaining a group of individuals. Repeated selection, crossover and mutation of the individuals are carried out until the global optimal solution is obtained.

##### 4.1. The coding and the generation of initial population

The primary task of an algorithm is the encoding method. It affects not only the design method of crossover operator, mutation operator, but also the convergence speed of the algorithm. The genetic algorithm which is based on real-coded is the actual description of the continuous parameter optimization problem. There is no binary encoding and decoding process which can greatly increase the efficiency of the algorithm.

Individual chromosomes are real-coded in genetic algorithm. Each chromosome is represented by feasible solution vector  $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_n)$ . At the same time, the upper and lower limits can be determined by the given constraints. That is the search space. The reliability allocation value can be obtained using formula 4 after getting the value of  $(\lambda_1, \lambda_2, \dots, \lambda_n)$ . And we can calculate the reliability index of the whole modules by using formula 4 and formula 5 repeated.

Cell generation is used in order to make the initial population distribute throughout the solution space. First, the whole solution space is divided into  $N$  small intervals.  $M$  individuals in each small interval are generated randomly. Then, an  $N \times M$  population is generated. The initial population is evenly distributed throughout the solution space by this method. Each individual has a greater probability to participate in the algorithm because of the significantly difference between each individual. The system's reliability goals can be tested by formula 7. New individuals can be generated until it meets the given constraints. That is to say, the cost of the software should be within the specific range and the reliability of the system should achieve the given goal.

#### 4.2. Selection operating

We use the  $(\mu + \lambda)$  selection strategy in the niche technology. It is considered to be the highest kind of selection pressure in evolutionary algorithm between several popular selection mechanisms. The  $(\mu + \lambda)$  selection mechanism can produce the fastest local convergence rate when the crossover operator paired in the population. The  $(\mu + \lambda)$  selection strategy is to select the  $\mu$  best individuals in the  $\mu$  parent individuals and  $\mu$  an individual cross  $\lambda$  tall individual.

Stochastic Tournament Model is selected in this paper. The winners become the next generation of individuals by the method of competing. In each generation of groups,  $K$  individuals are selected to constitute a small group randomly. And then, the fittest individuals are copied to next generation among the  $K$  individuals. The copy of the individual still returns to the parent groups and participate in the next random selection. This kind of selection will repeat  $M$  times and produce  $M$  next generation of individuals. In the method, we usually set  $K = 2$ .

The basic operation is as follows:

(1)  $K$  individuals are selected randomly between the individuals of generation  $T$ .

(2) The fittest individual will enter the generation  $T+L$  by comparing the fitness between  $K$  individuals.

And the copy individual will remains in the generation  $T$ .

(3)  $M$  individuals can be got by repeating the above two steps  $M$  times.

#### 4.3. Crossover operating

The purpose of crossover operating is to do global searching in genetic algorithms. The emergence of new excellent genetic model can promote the population and improve the accuracy of reconciliation.

Uniform Schema Crossover Operator is used in this paper. A random template is produced before the crossover operating. The crossover operating will carried out if the gene position of individual is 1. Compared to the single point crossover in traditional genetic algorithm, it has greater variation of the area and random mutation probability. And thus the search space is more effective and the local search ability is improved.

The uniform crossover operation is as follows:

(1) A long and individual encoded string  $w = \omega_1\omega_2 \cdots \omega_i \cdots \omega_L$  is generated randomly.  $L$  is the individual coding string length.

(2) Two new offspring  $A'$  and  $B'$  will be produced from  $A$  and  $B$  generations following the rules.

If  $\omega_i = 0$ , then the gene value in position  $I$  of  $A'$  inherits the value of  $A$  while the gene value in position  $I$  of  $B'$  inherits the value of  $B$ .

If  $\omega_i = 1$ , then the gene value in position  $I$  of  $A'$  inherits the value of  $B$  while the gene value in position  $I$  of  $B'$  inherits the value of  $A$ .

#### 4.4. The mutation operating

Mutation operation in genetic algorithm means the gene value of certain genes in the individual is replaced by other gene value. The local search ability of genetic algorithm is improved by mutation operation. The diversity is maintained and the premature phenomenon is prevented.

A Nun-Uniform Mutation is used in this paper. It refers to a random perturbation of the original genetic value. The new gene value is the result of the disturbance. A slight change in the solution space is made for each mutation operation.

Set that the mutation operation is from  $X = x_1x_2 \cdots x_k \cdots x_L$  to  $X' = x_1x_2 \cdots x_k' \cdots x_L$ . The range of

$x_k$  is  $x_k \in [x_{\min}^k, x_{\max}^k]$ . Then  $x_k'$  is determined by Formula 8.

$$x_k' = \begin{cases} x_k + \Delta(t, x_{\max}^k - x_k), & \text{if } \text{random}(0,1) = 0 \\ x_k - \Delta(t, x_k - x_{\min}^k), & \text{if } \text{random}(0,1) = 1 \end{cases} \quad (8)$$

In the formula,  $\Delta(t,y)$  is a random function in the range of  $[0,y]$ .  $y = x_{\max}^k - x_k$  or  $y = x_k - x_{\min}^k$ .

## 5. The analysis of simulation results

The Niche Genetic Algorithm is used to solve the problem of three-module software system reliability. The cost function of its software system takes the number of exponential cost model. We can get the function as follow.

$$\begin{aligned} C(\lambda_1) &= -126 \ln(1 - \exp(-\lambda_1)) \\ C(\lambda_2) &= -315 \ln(1 - \exp(-\lambda_2)) \\ C(\lambda_3) &= -238 \ln(1 - \exp(-\lambda_3)) \end{aligned} \quad (9)$$

The total cost of the development  $TC(\lambda_1, \lambda_2, \lambda_3)$  can be got by Formula 6.

The development and investment of the software system is 186,000 yuan and the profit is 48%. Other parameters are as follow:  $\phi_1 = 0.21$ ,  $\phi_2 = 0.48$ ,  $\phi_3 = 0.31$ ,  $\rho = 0.975$ . The lower limit value of each module is as follow by the analysis of fault trees and the division of the importance of the modules.  $(I_1 I_2 I_3) = (0.921 \ 0.978 \ 0.965)$ .  $(w_1 w_2 w_3) = (0.212 \ 0.487 \ 0.311)$ . The reliability allocation model can be got as follow.

The objective function:

$$\text{Min}\{f = -126 \ln(1 - \exp(-\lambda_1)) - 315 \ln(1 - \exp(-\lambda_2)) - 238 \ln(1 - \exp(-\lambda_3))\}$$

The constraints are as follow:

$$\begin{cases} e^{-(0.21\lambda_1 + 0.48\lambda_2 + 0.31\lambda_3)} \geq 0.975 \\ \lambda_i \geq 0 & (i=1,2,3) \\ e^{-0.21\lambda_1} \geq 0.921 \\ e^{-0.48\lambda_2} \geq 0.978 \\ e^{-0.31\lambda_3} \geq 0.9756 \\ 24 + 15e^{-0.21\lambda_1} \leq 0.52 \times 74.5 \\ 20 + 71e^{-0.48\lambda_2} \leq 0.52 \times 175 \\ 26 + 32e^{-0.31\lambda_3} \leq 0.52 \times 111.5 \\ 70 + 15e^{-0.21\lambda_1} + 71e^{-0.48\lambda_2} + 32e^{-0.31\lambda_3} \leq 186 \end{cases} \quad (10)$$

(unit : 1000yuan)

It is a combinational optimization problem with one linear objective function and  $3n+2$  linear constraints.  $N$  means the number of modules.  $\lambda_i (i=1,2,3)$  is the failure rate parameter in the modules. That is to say, the cost of the software should be minimized in a certain task time. And the reliability of the software should be greater than 97.5%.

The Niche Genetic Algorithm is used in the paper to solve the reliability of the three modules. And it is

compared with the simple genetic algorithm as shown in Table 1. In the simulation, the scale of the population is 150. The maximum number of the generations is 500. The crossover probability is 0.98 and the mutation rate is 0.01. The approximate optimal solution is:  $(R1\ R2\ R3) = (.96879\ .98176\ 0.97965)$ ,  $RS = 0.97642$ . That is the maximum reliability of the software.

Table 1 Comparison of niche genetic algorithm and simple genetic algorithm

Software system	algorithm	The number of precocity	The number of finding optimal solution	Success rate
Three-module	SGA	156	324	63.5%
	NGA	18	487	87.9%
Five-module	SGA	267	308	52.3%
	NGA	89	415	83.4%

From the Table 1, we can see that the precocity phenomenon is inhibited effectively. And the global optimal solution is found quickly.

## 6. Conclusions

Linear programming works with multi-variable and multi-constraint were solved successfully by using Niche genetic algorithm in multi-module software system. At the same time, the search ability of genetic algorithm was improved by using Stochastic Tournament Model, Uniform Schema Crossover Operator and Non-Uniform Mutation. The simulation results show that Niche Genetic Algorithm in solving the multiple modules of complex software systems reliability allocation problem was efficient and effective. The next step, we will do some research on improving encoding, genetic operators and selection strategies. In short, the reliability of software systems based on genetic algorithm optimization problem worthy of further research.

## References

- [1] XG Mao, YJ Deng. A general model for component-based software reliability[J]. Journal of Software, 2009,20(1):27-32
- [2] Kapur PK, Goswami DN, Gupta A. A software reliability growth model with testing effort dependent learning function for distributed systems[J]. Journal of Reliability, Quality and Safety Engineering, 2010,17(4):365-377
- [3] LU Qing, LIANG Chang-Yong. An Adaptive Niche Genetic Algorithm for Multimodal Function Optimization[J]. Pattern Recognition and Artificial Intelligence, 2009,22(1):91-100
- [4] HUANG Cong-ming, CHEN Xiang-xiu. Improvements on Niche Genetic Algorithm[J]. Journal of Beijing Institute of Technology, 2011,31(8):675-678.
- [5] FENG Yi, LI Li, GAO Yan-ming. A Niching Hybrid Genetic Simulated Annealing Algorithm[J]. Mechanical Science and Technology, 2004,23(12):1494-1498.
- [6] XI Hong-lei, XING Xiao-shuai, ZHANG Qing-quan. Adaptive Niche Genetic Algorithm Based on Gradi-optimization[J]. Computer Engineering, 2008,34(11):186-188.