

# Study Of test Data Generation Method Based on Evolutionary Algorithm

NI Jian, Zhang Yunlong\*

Hebei University of Engineering, School of Information & Electronic Engineering

hebei handan, 056038

\*Corresponding author, email: [yunlonghandan@163.com](mailto:yunlonghandan@163.com)

## Abstract

*The study of automatically generate test data based on evolutionary algorithm method focuses on the path coverage direction. The key problem is how to construct a suitable and has a good orientation of the fitness function to evaluate the quality of a test data. This paper introduces several evolution test methods, shows its advantages and problems it can effectively solve, and proposes an improved evolutionary test data generation method, which can generate better test data.*

**Keywords:** automatic test data generation, evolution algorithm, fitness, function structural test

**Copyright © 2014 Institute of Advanced Engineering and Science. All rights reserved.**

## 1. Introduction

Software testing is to find defects in the software, it is one of the effective means to ensure the software quality, and is an indispensable component of the software development life cycle. A test program is good or bad depends on the selection of test cases. However, software testing is a complicated process. It throughout the stages of software development process, and requires a lot of manpower, material resources and time. Usually, test automation technology can provide a method to generate test cases with high quality, and improve the efficiency of testing. Research a kind of effective test automation technology, to reduce the cost of software testing has very important significance.

Evolutionary algorithm has been applied to main fields, such as contingent power network [1]. Evolutionary Testing make the test cases generation process into methods of using genetic algorithm for numerical optimization problems, using the fitness function as test objectives, mapped test space to the search space of the algorithm, generate test cases which meet the target of testing automatically and efficiently. The whole process without too much human intervention, thus using evolutionary testing technology to generate test cases greatly reduces the test cost.

Evolutionary testing mainly includes structural evolution testing, functional testing and performance testing, etc. At present the most widely studied of evolutionary testing technology is structural evolution test technology. Structural evolution test make test target into the cover of the program structure, such as statement/branch coverage, path coverage, etc. [2], the mainly used fitness constructor are distance oriented method, sorting cover oriented method and the control cover oriented method [3]. Path oriented test data generation is one of the main methods of test data generation based on the structural test. There are a lot of researches for path oriented method, especially on evolutionary algorithm to generate test data got great research progress.

Evolution test has the advantages of efficient, dynamic and strong guidance; the key is to design a reasonable fitness function. If the fitness function is not suitable, test efficiency may be lower than random testing. For different applications or test objects, fitness functions are different. And in order to get the best effect, fitness functions need to be constantly adjusted according to the processing of evolutionary testing. There are two main problems existing in the current evolutionary testing: the first one is invalid solution problem, which can't determine whether the solution is used in the effective search domain; the other is the problem of population degradation. There are only a few genotypes in the population, which directs the local optimal solution can always survive [4]. The mainly reason is the fitness function only according to whether it meet the test conditions or is close to the optimal judgment, rather than to judge

whether the solution is in the effective search domain, whether it leads to the factors such as degradation of population or not.

In addition, the evolutionary testing algorithms mostly use coverage [5] the indicators as the feedback information of testing procedures, test execution is not considered in cost, there is no guarantee that you get the test program set of execution efficiency. For a software, if completes all the required validation works need a lot of testing procedures, and however the test execution efficiency is low, thus can greatly lengthen the test time, and lessen efficiency. And in the software development cycle, need for regression testing for many times, repeated low execution efficiency of testing process, will seriously affect the project schedule. To solve these problems, this paper improved the adaptive fitness function of an integrated design approach, both to ensure coverage, improve the execution efficiency, and can effectively avoid population degradation and local optimal solution of the problem.

## 2. Improving the Fitness Function

For invalid solution and population degradation problems, researchers put forward an optimization method based on punishment function of evolutionary testing [6], this method can punish population individuals which are outside of the search domain, and reduce the fitness value of local optimal solution, make more solutions involve in the next iteration of evolution process. The fitness function is defined as shown in formula (1):

$$y(x) = f(x) - \theta_1 \times g(x) - \theta_2 \times h(x),$$

$$(\theta_1, \theta_2 \in [0, +\infty)) \quad (1)$$

In formula (1),  $f(x)$  is the original definition of fitness function,  $g(x)$  is the punishment function to deal with invalid solutions,  $\theta_1$  is invalid solution punish parameter,  $h(x)$  is the punishment function to deal with local optimal solutions,  $\theta_2$  is the local optimal solution punish parameter.  $\theta_1$  and  $\theta_2$  is constant which scope is  $[0, +\infty)$ , their values determines the corresponding penalties, and is needed to be determined before the test started. For invalid solutions and local optimal solutions, using static method and dynamic punishment method to do punishment, because for invalid solutions, effective input domain of solution has been determined before the start of the test, to judge whether they are valid only need to according to distance of the solution to the effective input fields. The greater of the distance, the larger of the punishment function's value, for efficient solutions, the punishment function does not work. The punishment for local optimal solutions is required to according to the individual's proportion of the population to dynamically adjustment. Usually use simulated annealing punishment function or adaptive punishment function two ways to punish local optimal solution. The longer of the time individual as local optimal solution, the larger of the punishment function's value. So, the fitness function can dynamic adjustment of the individual fitness value according to whether the individual is the efficient solution or local optimal solution, avoid invalid solution and degradation occurred.

For evolutionary algorithm using coverage indicators as judgment of test program's optimization goal, and lead to the problem of low efficiency, researcher proposed test program generation method based on multiple target optimizations [7]. The method consists of two objective functions: the first one is to achieve test procedure's largest coverage; the second is to reduce test execution cost to a minimum. In test program generation problems, for coverage is the main indicator for assessing how the test program is. However, for a test program with very low coverage, while its execution cost is very small, execution efficiency is high, but it is not the solution we need. If not remove those solutions, they will always at the top for they have high execution efficiency. So we need to set punishment function to eliminate the solution of small coverage. We can use formula (2) to determine whether a solution should be eliminated:

$$y(x) = \frac{f_{\max} - f(x)}{f_{\max} - f_{\min}} \times 100\% \quad (2)$$

In formula (2),  $f_{\max}$  and  $f_{\min}$  is the maximum and minimum value of coverage individuals allow in a population. The formula reflects the population individuals' deviation degree of minimum coverage rate relative to the maximum value. If the deviation degree exceeds the

threshold which is set at the beginning of test, the individuals will be regarded as invalid solution, and can be eliminated.

In addition, the traditional evolutionary test data generation method exists problems of algorithm complex, and coding difficulties and parameters are not easy to set up and so on, some scholars put forward a kind of software test data automated generation method based on differential evolution algorithm [8]. Operator to the operation of the algorithm was improved, solves the problem of test data to generate discrete differential evolution algorithm. The basic idea of differential evolution algorithm is first generate an initial population in the search space, then produce new individual by weighting any two individuals' vector differential and then superimpose a third individual. Then compare the fitness value of the new individual and the individual, if the new individual's fitness value is better than that of the parent individual's fitness value, then replace parent individual with the new individual, or parent individual is still preserved. By iterative calculation unceasingly, keep excellent individuals surviving, and eliminate inferior individuals, thus the search process was guided to approach the optimal solution.

According to the mentioned evolutionary testing technology improvement idea, we put forward the following method to construct of fitness function: First of all, in order to measure the quality of the test program, in addition to using coverage indicators, at the same time also use performance indicators and distance indicators. Execution efficiency indicators can guarantee test speed of test program in large software, and guarantee the test cases to cover the current node or statements to the test set goals at the beginning of distance information, and judge the current node's or a statement's distance information with test goal while covering the test case. Three indicators evaluate a test program at the same time, which can assure the high coverage rate of evolutionary testing. In addition, in order to solve the problem of population degradation and local optimal solution, the introduction of punishment function and the fitness function to influence population evolution test cases are eliminated. The fitness function as shown in formula (3):

$$y(x) = f(x) - \theta_1 \times e(x) - \theta_2 \times d(x) - \theta_3 \times g(x) - \theta_4 \times h(x) \quad (3)$$

$(\theta_1, \theta_2, \theta_3, \theta_4 \in [0, +\infty))$

The  $f(x)$  is the original fitness function,  $e(x)$  is the execution efficiency of the decision function,  $d(x)$  is the test cases to cover the distance with the target node functions,  $g(x)$  is the punishment for invalid solution function,  $h(x)$  is the local optimal solution of the penalty function,  $\theta_1, \theta_2, \theta_3, \theta_4$  is the four relevant parameters of the functions, they are setup before the start of the test. The value of the four parameters determines the functions' influence degree of test cases.

### 3. Experimental Verification

In order to verify the validity of the design of fitness function, the triangle program test experiment was carried out. Take the three sides of the triangle  $a, b, c$  as for the input items, the test goal is to achieve three sides equal nodes, that is  $a=b=c$ . Representation for the evolution of population individuals is  $(a, b, c)$ . The experiment of the total number of evolution  $T=500$ , the size of the population size  $P=200$ , the largest number of iterations  $I=20$ . The input fields for the input item  $a, b, c$  is  $[30, 50]$ . For the efficiency of decision function  $e(x)$  using computer clock cycles to determine, the computer's main frequency for executing the test program is 1.7GHz. Therefore  $e(x) = 1.7 - T_x \times I_x / 1.7$ , distance function  $d(x) = \min(|30 - x|, |50 - x|)$ , punishment function of ineffective solution  $g(x) = (1 / (0.001 + 1/d(x)))$ , punishment function of local optimal solution  $h(x) = \exp(-2/T_x) \times P_x$ ,  $\theta_1, \theta_2, \theta_3, \theta_4$  are set to 0.5, 1, 0.5, 30. So the fitness function  $y(x)$  as shown in the formula (5):

$$y(x) = f(x) - 0.5 \times \left(1.7 - T_x \times \frac{I_x}{1.7}\right) - \min(|30 - x|, |50 - x|) - 0.5 \times \left(1 / (0.001 + 1/d(x))\right) - 30 \times (\exp(-2/T_x) \times P_x) \quad (5)$$

In experiment, each different size of the population was separately run for 10 times, record each time to find the optimal solution of the iteration number and running time, and

calculate the average iterative count and iteration time. Figure 1 and figure 2 show the experimental results comparison to  $f(x)$  and  $y(x)$ .

From the experimental results of figure 1 and figure 2, it can be seen that after using the adaptive value function  $y(x)$ , the triangle program have obvious improve both iteration count and iteration time, which indicates after added the punishment function and efficiency function,  $y(x)$  is very efficient to improve the coverage and punishment ineffective solutions and local optimal solution.

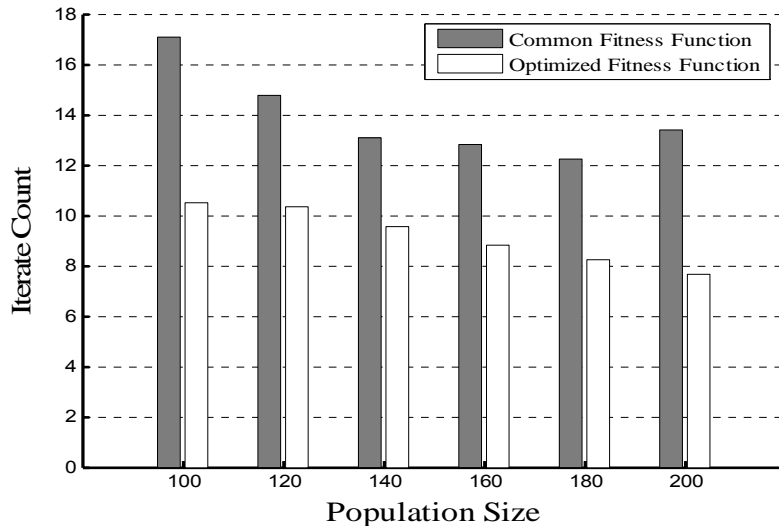


Figure 1. Generate an equilateral triangle of the iteration count

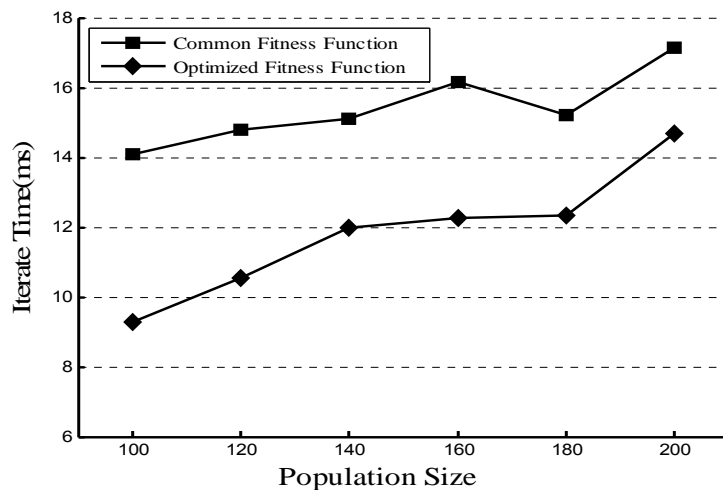


Figure 2. Generate an equilateral triangle of the iteration time

#### 4. Summary

In this paper, the method of adopting adds a series of control means on the basis of the original fitness function to solve the problems of evolutionary testing. And supervise the process of population evolution, both retained the evolutionary test coverage, and can guarantee test program's execution efficiency, and can also avoid degradation and invalid solution in the process of population evolution, optimize the quality and efficiency of test data generation.

**References**

- [1] Sawan Sen, Priyanka Roy, Abhijit Chakravarti, Samarjit Sengupta. Genetator Contribution Based Congestion Management using Multiobjective Genetic Algorithm. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2011; 9(1): 1-8.
- [2] Sandip Chanda and Abhinandan De. Congestion Relief of Contingent Power Network with Evolutionary Optimization Algorithm. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(1): 1-8.
- [3] XIE Xiaoyuan. Survey of Evolutionary Testing. *Journal of Frontiers of Computer Science and Technology*. 2008, 2(5):449-466.
- [4] Shi Liang. Research on Test Data Automatic Generation. Nanjing: Southeast University, 2007.
- [5] Zhang Liang. Coverage Matrix based Evolutionary Test Program Generation for Microprocessor Verification. *Journal of Computer- Aided Design & Computer Graphics*. 2011; 23(3): 456-462.
- [6] Zhang Nan. Research on Evolutionary Testing OptimizationBased on Penalty Function. *Computer & Digital Engineering*. 2009; 37(4): 1-3.
- [7] Zhang Liang. Test Program Generation Based on Multi-objective Evolutionary Algorithm. *Journal of Frontiers of Computer Science and Technology*. 2010; 22(8): 1382-1389.
- [8] Huang Xiaocheng. Application of modified differential evolution in test data generation. *Journal of Computer Applications*. 2009; 29(6): 1722-1754.