

Cloud computing environment based on Web log mining algorithm implementation of test

Wang Bin

College of Information Science &
Electronic Technology
Jiamusi university
Jiamusi, china
e-mail:jmsuwang@163.com

Yong-cheng Jiang*(Author for
correspondence)

College of Mechanical Engineering,
Jiamusi University
Jiamusi,154007,China;
E-mail: jiangyongcheng@126.com

Li jing

College of Information Science &
Electronic Technology
Jiamusi university
Jiamusi, china
e-mail:lijing2483@163.com

Abstract—Software test is the important means that guarantee software quality and reliability, and in this respect, it plays the role that other method cannot replace. However software test is a complex process, it needs to consume huge manpower, material resources and time, which takes the 40%-50% of entire software development cost approximately. Paper presents the inherent in software test case designing based on genetic algorithm is using genetic algorithm to solve a set of optimization test cases, and the framework includes three parts which are test environment construction, genetic algorithm and the environment for test.

Keywords- software test, test case, genetic algorithm

I. INTRODUCTION

Algorithm (genetic algorithm) is the simulation of Darwin's selection and the natural elimination process of biological evolution model. Web log mining algorithm was first proposed by Professor J.Holland in 1975 by the United States. Its main characteristic is the direct operation to structural object, there is no derivation and the continuity of function limit; inherent implicit parallelism and better ability of global searching; using the probability optimization method, can automatically access and guide optimal search space, adaptively adjust the search direction, does not need to determine the rules. The algorithm processes include coding, generation of initial population, fitness value was calculated, selection, crossover, mutation operation. The general flow diagram as shown in figure 1:

II. BASED ON THE WEB LOG MINING ALGORITHM FOR GENERATING TEST CASE

Based on the algorithm of test case generation basic processes as follows:

- (1) analysis of the source code, access to program control flow diagram;
- (2) by a program control flow diagram to get program branching path set, choose the target path;
- (3) according to each predicate terms to the program instrumentation and the development of fitness function;
- (4) set the algorithm parameters, including population size, variation, randomly generated initial test data set;
- (5) using the test data to perform after plug-in source code, to obtain the fitness value, according to the fitness

value judgment, if meet the program termination condition output results and exit the program, if the conditions are not satisfied to step (6);

(6) according to the fitness value, using the algorithm selection, crossover, mutation operation, generating new test data, and go back to step (5), repeat;

Program frame as shown in Figure 2, including the test environment structure, algorithm and test running environment part three.

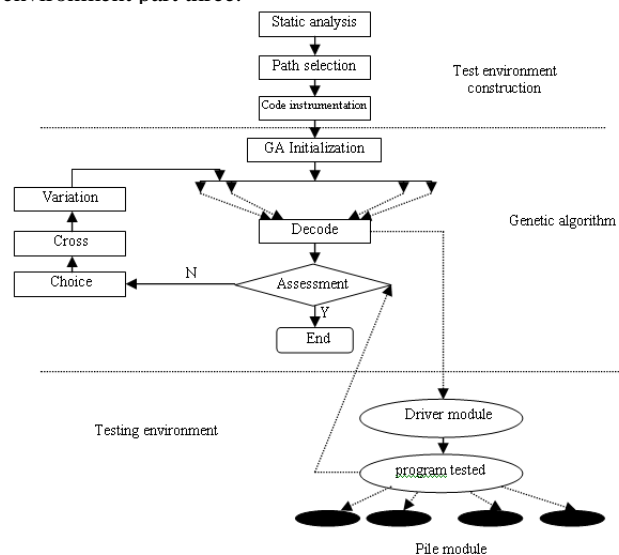


Figure 2. the program frame diagram

III. TESTED PROGRAM ANALYSIS

A. The program to be tested into

The triangle classification procedure as an example to verify the algorithm. The triangle classification procedure in many software testing research is as the benchmark program to use, because it contains clear and complex logic, and even a larger range of integers as input, and only a small amount of input combinations to meet specific branch code. For example, when the input is 1 to 10 integer, 1000 input 10 groups only can satisfy the judgement as "equilateral triangle" branch. Triangular source for:

```
String tri_type(int a,int b,int c)
{ string type;
```

```

if(a>b) change(a,b);
if(a>c) change(a,c);
if(b>c) change(b,c);
if(a+b>c)
{
  if(a==b||b==c)
  {
    if(a==c) type="Equilateral triangle";
    else type="An isosceles triangle";
  }else type="Ordinary triangle";
} else type="Is not a triangle";
return type; }
    
```

B. program flow analysis

The program flow diagram as shown in Figure 3, the program is divided into two parts, the first input value from the sort, then classify triangles.

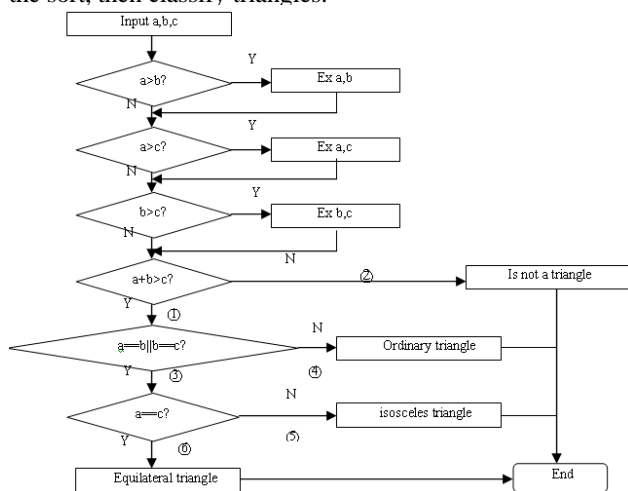


Figure 3. the triangle classification process

C. path analysis

Through the process of analysis, the procedure to classify triangles statement as in Figure 3 is marked with a " 1 " to " 6 " on the branch, the branch path analysis as shown in table I:

TABLE I. PATH ANALYSIS

Path	Branch	Result
w1	②	Is not a triangle
W2	①-④	Ordinary triangle
W3	①-③-⑤	An isosceles triangle
W4	①-③-⑥	Equilateral triangle

IV. CODE INSTRUMENTATION

```
String tri_type(int a,int b,int c)
```

```

{
  string type;
  if(a>b) change(a,b);
  if(a>c) change(a,c);
  if(b>c) change(b,c);
  f1=c-(a+b); // Pile plug 1
  if(a+b>c)
  {
    f2=min(abs(a-b),abs(b-c)); // Pile plug 2
    if(a==b||b==c)
    {
      f3=abs(a-c); // Pile plug 3
      if(a==c) type="Equilateral triangle";
      else type="An isosceles triangle";
    } else type="Ordinary triangle";
  } else type="Is not a triangle";
  return type;
  F=1/(1+f1)+1/(1+f2)+1/(1+f3); // Fitness
}
return F; }
    
```

V. RESULTS ANALYSIS

The triangle classification procedures are condemned as " an equilateral triangle " path W4 for example, run the program, found in the 51 generation to find the optimal solution, table 2, 3 respectively for the zeroth generation, thirteenth generation, the fifty-first generation of fitness for the top 10 individual case:

From the two table is visible, along with the evolution process continued, the overall fitness in the description of the algorithm increases, are moving towards the optimal solution in the direction of convergence, to find the optimal solution. This algorithm in the program of automatic test data generation is has a role, it will gradually improve individual.

ACKNOWLEDGMENT

This research was supported by the National Nature Foundation of China (Grant No. 61002004), the National Science Foundation of Heilongjiang Province (Grant No. E200908) , the Natural Science Foundation of Heilongjiang Province Education Office(Grant No. 12511551), the National Science Foundation of Jiamusi University(Grant No. L2012-074) , the National Science Foundation of Jiamusi City(Grant No. 12004) all support is gratefully acknowledged.

REFERENCES

- [1] D.Bird and C.Munoz. Automatic generation of random self-checking test cases[J]. IBM System J. vol.22, NO.3. 2010:229-245
- [2] P.D.Coward. Symbolic execution and testing[J]. Information and Software Teehnology. 2011, 2: 53-64
- [3] C.V.Ramamoorthy, S. Ho. And W.Chen. On the automated generation of Program test data[J]. IEEE Trans. Software Eng. Vol.SE – 2, NO.1. 2011 : 117-127
- [4] Ramamoorthy C.V. On the automated generation of Program test data[J]. IEEE Trans on Software Eng, 2010, 4 : 215-222

TABLE II. THE ZEROETH GENERATION TO THE TOP 10 INDIVIDUALS

Individual number	Chromosome encoding	Parameter value			Fitness
		A	B	C	
1	110101011100011100	53	28	28	67
2	100110101001100111	38	41	39	58
3	011011011010010110	27	26	22	55
4	000111001100001000	7	12	8	55
5	100110101101101100	38	45	44	54
6	011011100011100010	27	35	34	53
7	111011111010101011	59	58	43	51
8	110110111000111011	54	56	59	49
9	100001011110011100	33	30	28	49
10	110011111000111010	51	56	58	48

TABLE III. THE THIRTEENTH GENERATION TO THE TOP 10 INDIVIDUALS

Individual number	Chromosome encoding	Parameter value			Fitness
		A	B	C	
1	010000001011010000	16	11	16	72
2	101000111101101000	40	61	40	68
3	101000101000010010	40	40	18	68
4	101011001011101011	43	11	43	67
5	010100010111010110	20	23	22	58
6	111101101100111110	61	44	62	51
7	100111100110001010	39	38	10	51
8	011010111101111100	26	61	60	50
9	000011101001101000	3	41	40	50
10	011111100101100111	31	37	39	48

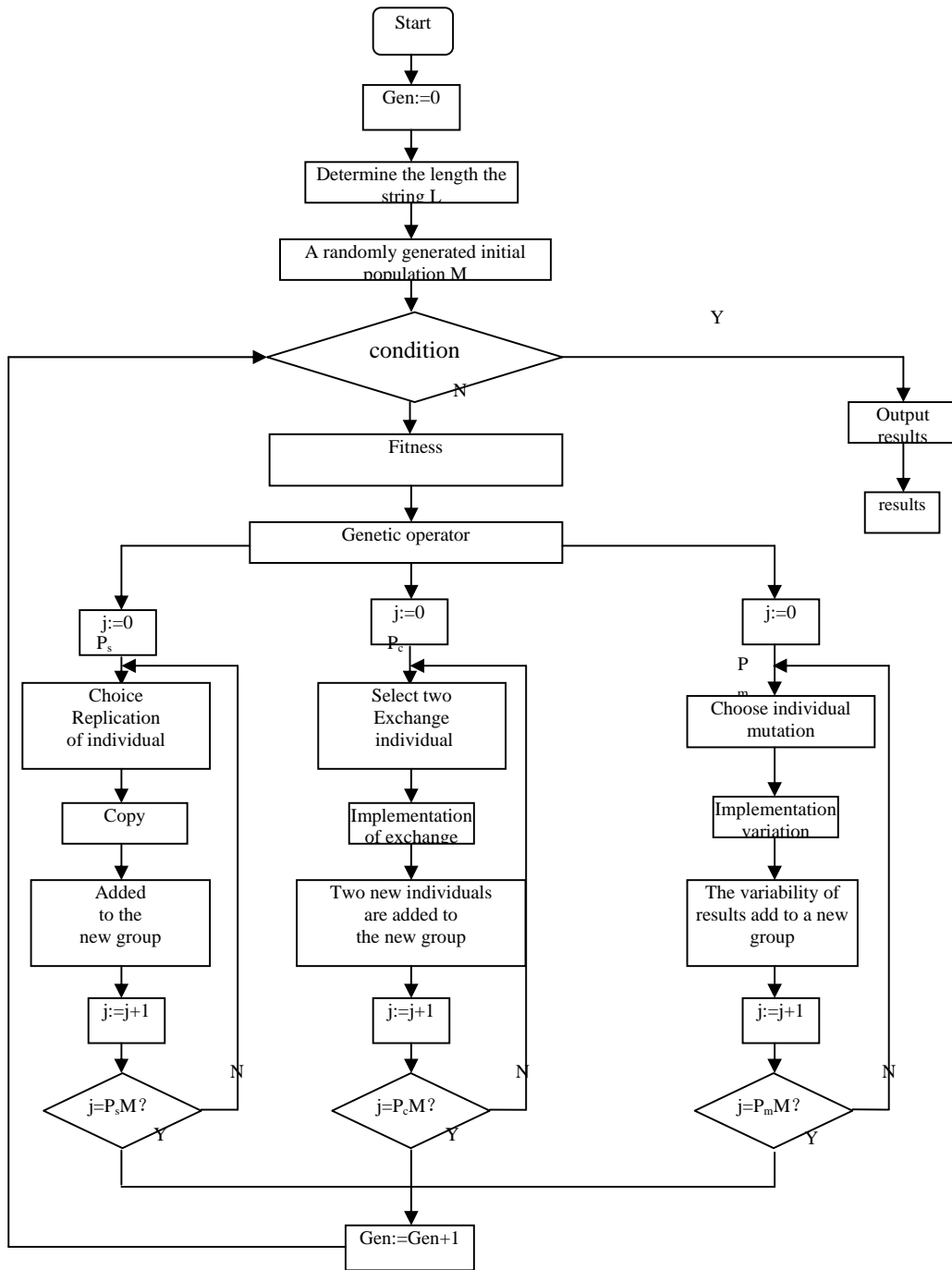


Figure 1. algorithm