ISSN 2349-6193
Impact Factor: 2.805

IJESMR

International Journal OF Engineering Sciences & Management Research

COMPARISON OF TEST CASES USING GENETIC ALGORITHM IN SOFTWARE TESTING

D. A. Madhivadhani *[1] & Mr. K. Radhakrishnan[2]

[*1]Research Scholar PG and Research Department of Computer Science, Dr. Ambedkar Government Arts College, Chennai - 600 039, India.

[2]Assistant Professor PG and Research Department of Computer Science, Dr. Ambedkar Government Arts College, Chennai - 600 039, India

Keywords: Genetic Algorithm, Fitness Function, Test data, R Programming

ABSTRACT

The limit of particularly offering encoded data to different customers through open cloud storage may This paper presents a set of methods that uses a software testing technique in genetic algorithm for a automatic test-data generation. These test case generations are used in evaluate the fitness function in genetic algorithm for selecting the best test case methods. In use a genetic algorithm to generate test cases for evaluating the fitness function in the case generation methods. In use of test cases in genetic algorithm to calculate the fitness function to improve the performance. To aim of the research paper is to implement in genetic algorithm to reduce the cost, time and effort and to minimization of test cases to deliver the good quality software.

INTRODUCTION

Software testing is a term to Quality and Reliability. And the technique used to validating and verifying to developing the software Quality. In software testing technique to remains the integral part of (SDLC) Software Development Life Cycle. Then it has done important phase is to remove the errors and to deliver a good Quality of the software. The first publication on 'search-based software testing' appeared in 1976, and was the work of two American researchers Webb Miller and David spooner[1].

INTRODUCTION TO GENETIC ALGORITHM

Genetic Algorithm is an optimization technique. In mostly used in software testing is to reduce the effort, time and cost and to deliver a good quality of the software. Genetic Algorithm is to find the solution of modified optimization problem. In a genetic algorithm to generate an automatic test cases in the random testing.
The genetic algorithm using and finding the evaluating a fitness functions, in the two test cases are generated

INTRODUCTION TO GENETIC ALGORITHM

```
Initialization population;
Randomly generated n number of individuals;
Evaluate fitness;
while(fitness value!=termination of criteria)do
{
        Selection;
        Crossover;
        Mutation;
        Calculate fitness function;
}
```

GENETIC ALGORITHMS USE IN THREE OPERATIONS

- SELECTION
- CROSSOVER
- MUTATION

Selection

Selection is the process to analyzing the chromosome or the population and to generate a fitness value. In one of the various selection method in RANDOM SELECTION

ISSN 2349-6193
Impact Factor: 2.805

IJESMR
International Journal OF Engineering Sciences & Management Research

Crossover
Crossover is a technique to used to generate a new chromosome in the existing values and to generate a new population.

Mutation
In mutation process to evaluate a fitness values in one by one and to execute the average fitness value in the existing function

First case:
Population size = 50
Number of generations = 1000
Crossover rate=0.7
Mutation rate = 0.001

Table 1: Average fitness value with mutation rate = 0.001

Generation	Average Fitness	Max Fitness
1	31.88	33
2	32.88	34
3	32.67	34
4	32.75	35
5	32.96	34
10	34.17	38
25	37.29	42
50	41.21	44
100	39.67	47
150	44.33	49
200	46.33	47
250	47.63	49
500	50.23	50
750	51.79	51
1000	52	51
	607.79	638
	Average Fitness Value	95%

Second Case
Population size = 50
Number of generations = 500
Crossover rate=0.7
Mutation rate = 0.01

ISSN 2349-6193
Impact Factor: 2.805

IJESMR
International Journal OF Engineering Sciences & Management Research

Table 2: Average fitness value with mutation rate = 0.01

Generation	Average Fitness	Max Fitness
1	32.25	32
2	32.13	32
3	33.92	33
4	32.42	33
5	33.79	33
10	34.71	34
25	38.42	36
50	39.08	37
100	37.42	36
150	35.54	40
200	38.79	40
250	41.83	39
500	42.18	43
	472.48	468
	Average Fitness Value	0.990518117

Population size = 50
Number of generations = 250
Crossover rate=0.7
Mutation rate = 0.001

Table 2: Best fitness value generation

Best Fitness	Average Fitness	Standard Deviation
16.77	11.59	5.18
20.39	15.09	5.3
20.39	15.71	4.68
22.99	15.84	7.15
22.99	16.03	6.96
23.24	17.65	5.59
25.12	20.87	4.25
25.89	21.42	4.47
26.27	20.88	5.39
26.77	23.28	3.49
26.77	20.38	6.39
26.78	22.41	4.37
26.98	22.68	4.3

ISSN 2349-6193
Impact Factor: 2.805

IJESMR

International Journal OF Engineering Sciences & Management Research

IMPLEMENTATION OF GENETIC ALGORITHM IN R PROGRAMMING

In implementing the genetic algorithm in R tool, to generate a chart for the fitness values

Case 1

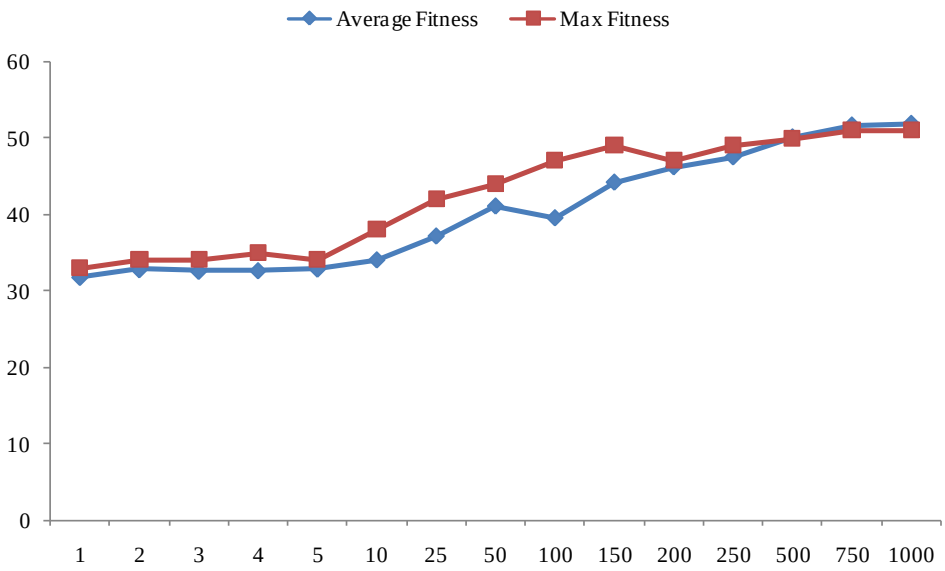

Fig 1 : Average fitness value with mutation rate = 0.001

Case 2

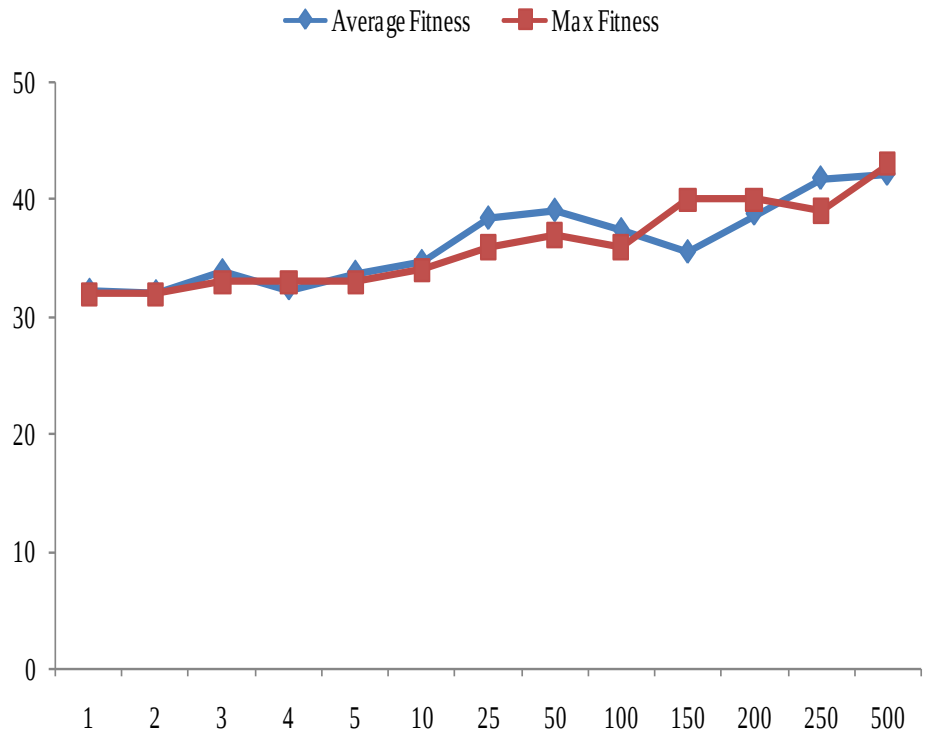

Fig 2 : Average fitness value with mutation rate = 0.01

ISSN 2349-6193
Impact Factor: 2.805

IJESMR

International Journal OF Engineering Sciences & Management Research

Best Fitness

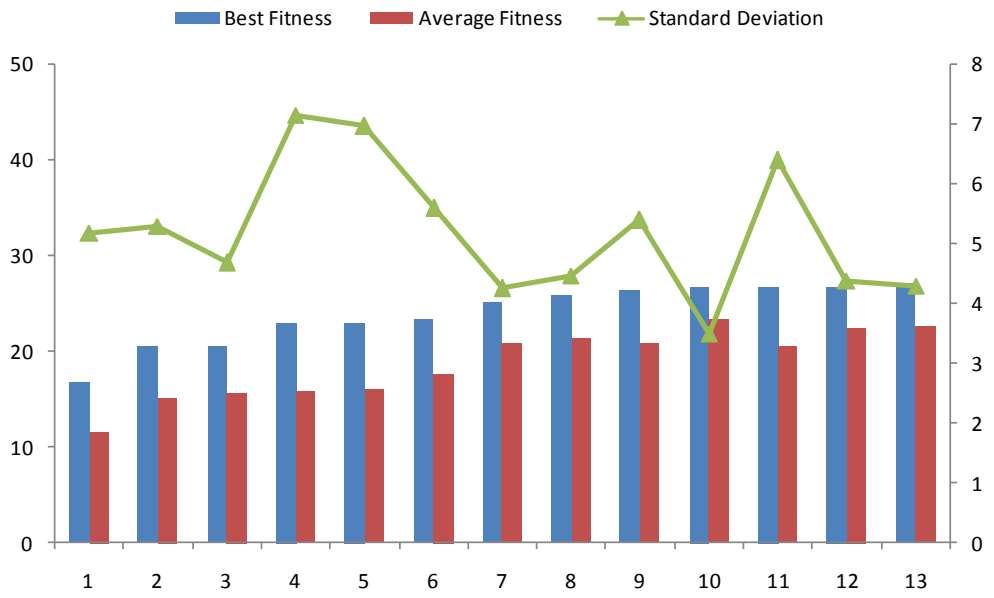

Fig 3 : Best fitness value generation

CONCLUSION

In this paper to generated a test cases, in implementing the genetic algorithm and to evaluate the fitness values in the Minimum values in the Minimum values. To finds the optimum solution for the fitness function then to using the test cases in Genetic Algorithm.To most commonly use of a Genetic Algorithm is to initializing the population or chromosome ,and to done the selection process and after is also done a crossover and mutation. Its a genetic operators then evaluating mechanism and the optimum solution is finded. To implementing the test cases in the use of genetic algorithm as to given the optimal solution .When the help of R tool is used to generate a graph and to resulted as the optimal solutionvalues.In future work is to implementing the tase case values are higher,and to use this test cases are in uuse of another algorithm and to use of PSO(Particle Swarm Optimization). Then it has to be concluded has the method of test case generation and to implementing the fitness values is optimized solution.

REFERENCES

1. Goldberg, D.E, “Genetic Algorithms: in Search, Optimization & Machine Learning,” Addison Wesley, MA. 1989.
2. Horgan, J., London, S., and Lyu, M., “Achieving Software Quality with Testing Coverage Measures”,IEEE Computer, Vol. 27 No.9 pp. 60-69, 1994.
3. Berndt, D.J., Fisher, J., Johnson, L., Pinglikar, J., and Watkins, A., “Breeding Mark Last, Shay Eyal1, and Abraham Kandel”, “Effective Black-Box Testing with Genetic Algorithms,” IBM conference.
4. Lin, J.C. and Yeh, P.L, “Using Genetic Algorithms for Test Case Generation in Path Testing,” In Proceedings of the 9th Asian Test Symposium (ATS’00). Taipei, Taiwan, December 4-6, 2000.
5. André Baresel, Harmen Sthamer and Michael Schmidt, “fitness function design to improve evolutionary structural testing,” proceedings of the genetic and evolutionary computation conference, 2002.
6. Algorithms for Dynamic Test Data Generation,” Proceedings of the 1997 International Conference on Automated Software Engineering (ASE'97) (formerly: KBSE) 0-8186-7961-1/97 © 1997 IEEE. Somerville, I., “Soft ware engineering,” 7th Ed. Addison-Wesley,
7. Aditya P mathur,”Foundation of Software Testing”, 1st edition Pearson Education 2008. Alander, J.T., Mantere, T., and Turunen, P, “Genetic Algorithm Based Software Testing,”
8. http://citeseer.ist.psu.edu/40769.html, 1997.
9. Nashat Mansour, Miran Salame,” Data Generation for Path Testing”, Software Quality Journal, 12, 121–136, 2004,Kluwer Academic Publishers