Test Case Generation-Review

Shikha Rai
Department Of Information Technology, Northern India Engineering College, New Delhi
raishikha21@gmail.com

***Abstract* - Software testing is important component of software quality assurance as programmer can get view of specification, design and coding of the software.The main aim is to minimize the number of faults detected in software. Test case generation can be considered as an optimization problem .Test case data can be generated automatically and also with random testing. Aim of test case generation is to satisfy particular criteria. We can generate test case with help of genetic algorithm. Further comparison can be done between random testing and genetic algorithm.**

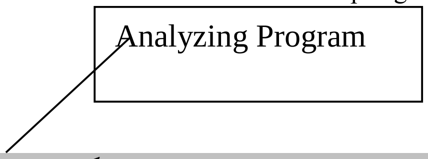
***Keywords*- Boundary value analysis, Genetic Algorithm, Software Testing, Automatic Testing, Random Testing**.

I. Introduction

Requirement analysis, Design, Coding and Testing are phases of software development. Testing requires maximum effort from programmer. In order to discover error software must be tested through test cases Testing of software can be done with help of two techniques i.e. white box testing and black box testing. Black box testing is based on comparing input and output without revealing how modules are working .White box testing is based on checking the control flow graph of modules .In this actual implementation of software is analyzed .But both the approaches are time consuming. In order to reduce the efforts there should be process which generate test case automatically .Random testing ,Anti random testing are the number of objective of all these techniques is to find minimal number of test cases to test the software fully. This can be considered as an optimization problem. To solve optimization problems there are number of techniques and one of them is Genetic Algorithms [1].
Performance of software product is most important factor to judge usage and quality of software .Software testing add on to major cost under software development. Testing increases the code complexity exponentially thus testing is tiresome job. Manual testing is expensive and very slow. Testing should be done in small sections so that it is easier to locate errors .Automation of testing make testing reliable, faster and cost efficient.
The objective of this paper is to present automation testing using optimization method called genetic algorithm. Genetic algorithm is an optimization technique that is based on evolutionary algorithm .Genetic algorithm is inspired from the Darwin's theory of natural selection. This method is based on gene recombination in same way as it takes place in nature. This algorithm involves selecting parents randomly from population to generate the offspring. The offspring will inherit the genes from parents. The fitness function is evaluated then mutation and crossover operators are applied on offspring .This process continues in iteration until optimal solution is not reached or number of generations is not achieved.

II. Problem Statement

The important concept of testing is that function of system can be verified with the help of input which is selected as test data. Software can be verified by doing complete testing i.e. testing need to be done by using all possible value as input and checking the behaviour of software under all situations. Of all the testing activities test case design, test execution, monitoring, test evaluation, test planning, test organization, and test documentation are essential steps that are contributed in test case design [4].
Figure 1 models a typical test data generator system, it consists of three process i.e. analysing program, selection of path and test case generation. The source code is run through a program analyzer, which produces the necessary data used by the path selector and the test data generator. The appropriate path is being inspected .Path will be called appropriate if it has high coverage .The paths are then given as argument to the test data generator which derives input values that exercise the given paths. The generator may provide the selector with feedback such as information concerning infeasible paths. [1]
Software testing incurs the maximum cost from software development. The cost of software development is reduced if automated testing is done. The generation of test case can be divided into three classes: Random, path-oriented and goal oriented test data generation [6].
Non linearity behaviour of software is due to if statements, loops conversion of problems into optimization task result in complex spaces. Search space method like hill climbing is not appropriate as it leads to reach local minima. Thus suitable evolutionary algorithm should be used to find the effective solution of varied dimension search space. Dimension of search space is directly related to the number of input parameters of the system under test. [1] We need to perform automation testing using genetic algorithm. Data with higher fitness value have higher probability of selection and generate recombination and new offspring are also generated with mutation.



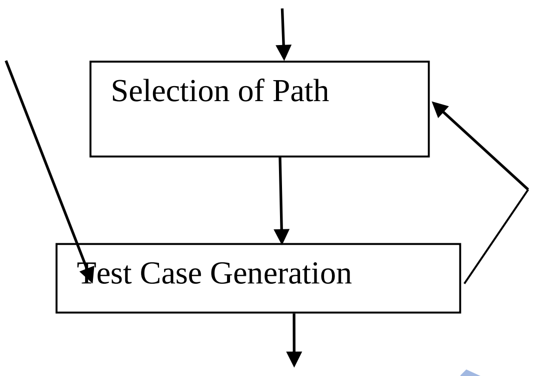


Figure 1: Test Case Generation Architecture

III. Literature Review

One of the simplest technique for test case data generation is random testing .Random testing can be used to generate input for any type of program such as integer ,string, heap as in end we are working on string of bits. Random testing is generally used for final testing stage [19-21]. This is cheapest method used for generation of test data [10].The main disadvantage of random testing is that it does not work well in terms of coverage. [13].Another disadvantage of random testing is that probability of finding semantically small faults is very low [8] and thus it leads to high coverage. If a fault is detected by a small percentage of input then it is called semantically small fault. For example, consider the code:

```
   void check(int x, int y)
{
   if (x ==y)
  cout<<"one"; // statement 1
  else
  cout<<"zero"; // statement 2
 }
```

For statement 1 to get executed both x and y values need to be same. The probability of executing statement 1 is 1/n where n is maximum integer. Random testing selects the data from domain randomly and then tests these values in test cases. The automatic production of random test data is considered as default method and result of various test case generations can be compared with random testing [9].

Another technique for generation of test case is statistical testing .Statistical testing generates test based on distribution of data .Distribution of data is based on usage of data. The distribution of selected input data should have the same probability distribution of inputs which will occur in actual use in order to estimate the operational reliability, [10]. To handle the exceptional cases like using the mixed final testing i.e. first using random testing and after that using value testing method can be used for generating test data [15].

Testing is an important step in development and it is need to be done effectively. Testing cannot be done at each step repeatedly because of limited resources. It is observed that programmers make mistakes in choosing values as input in boundary values as there is more chances that software fails at boundary value. So here comes the role of automatic test generation as this focus on these kinds of test cases [21]. Boundary values described as values which are taken as input to check the program at upper and lower end rather than errors in centre of input. For example if we need to write boundary value test case for text box which has min char. limit is 4 and max char. Limit 10,than answer will be that we require 4 test cases, we have to test it for values , 4 (Lower limit), 10 (upper limit), 11 (upper limit +1) and 3 (lower limit +1).

In this paper the boundary values are identified automatically through genetic algorithm and then testing is done and then result can be compared with both techniques i.e. random testing and genetic algorithm. Both these techniques take initial input as random population but fitness of individual is calculated in genetic algorithm where as random testing works randomly .We can consider distance from boundaries as fitness of individual.

IV. Genetic Algorithm for test case generation:

Test case can be generated by considering following parameters

Representation: Representation of the individuals can be done using real values. The length of the chromosome depends on the number of variables.
Initial Population: Initial population is generated randomly .Pop size vectors of c_size length are generated randomly, where pop_size is the size of population, and c_size is the number of variables. For the search space values, initial population is generated with 5 less than lower bound of variable, and 5 more than upper bound values, i.e. if lower bound and upper bound are 5,15 respectively, then initial values are generated from 0 (5-5) and 20 (15+5). The appropriate value of pop size is experimentally determined. [2]

Fitness Function: Fitness of each chromosome is determined by its difference from the boundaries of the variable. Variable which is more near to boundary is considered fit. Code can be written as finding the upper and lower bound of the variable and them finding the difference with current population .Individual will be fit if it is closer to boundary value. [2]

```
Fitness(popsize, chromLength, curpop)

lBound = lower boundary of the variable;

uBound = upper boundary of the variable;

for I = 1 to popsize

      for j = 1 to chromLength

       diffLower = lBound – curpop(I,j);

        diffUpper = uBound – curpop(I,j);

         moreClose = min(diffLower, diffUpper);

         fitness(i) = moreClose;

           end

end

end;
```

Selection: As the fitness of individual is calculated then selection of test case is carried out from current population. Genetic algorithm uses roulette wheel method where as random testing use random selection method for selecting test cases.
(i) Roulette wheel: For the selection of a new population wheel is used hic is divided into slots according to fitness value of each individual. Higher the value of fitness, larger will be the slot size in wheel and greater is the probability of individual to get selected.

(ii) Random selection: In this method, the selection of parents is done randomly thus each individual has equal chance of getting selected.

Crossover: In crossover operator two individuals i.e. parents are selected to form recombinants. It operates at the individual level. In this there is exchange of string take place at random position in chromosome to generate two new strings. The objective of crossover is to generate the individual with higher fitness. Crossover is done according to crossover probability. The probability of crossover pc gives us the expected number pc · pop_size of chromosomes, which undergo the crossover operation which is as follows [22]:
For each chromosome in the (new) population:

• Generate a random (float) number r from the range [0...1];

• If $r < pc$ then select given chromosome for crossover.

Now selected parents are randomly mated. For each pair of selected parents arithmetic crossover is used with crossover probability 0.7. Arithmetic crossover operator defines a linear combination of two chromosomes

[19].Two chromosomes are selected randomly for crossover and produce two offspring's which is linear combination of their parents as per the following computation:

C igen +1 = a.C igen + (1-a).C jgen

C jgen +1 = a.C jgen + (1-a).C igen

Where C gen an individual from the parent generation gen +1 an individual from child generation, a (alpha) is the weight which governs dominant individual in reproduction d it is between 0 and 1.[2]

V. Conclusion

This discussion shows that in order to increase the efficiency and effectiveness and we need to decrease the cost of test case generation is required. Test cases can be generated with the help of random testing, genetic algorithm. Random testing is quick and simplest approach but it might not work well in complex programs because of its limited coverage. If we have to generate high quality software that technique must be used for test case generation which will lead to maximum coverage irrespective of cost. In this paper we have studied these techniques. In this paper we also discussed boundary value analysis which is used in calculating fitness in genetic algorithm. We have also discussed various parameters that can be used in genetic algorithm.

References

[1] Rakesh Kumar, Surjeet Singh, Girdhar Gopal,"Automatic Test Case Generation Using GeneticAlgorithm",International Journal of Scientific & Engineering Research, Volume 4, Issue 6, June-2013 ISSN 2229-5518
[2]Holland J., "Adaptation in natural and artificial systems", University of Michigan Press, Ann Arbor, 1975.
[3] Goldberg D. E., "Genetic algorithms in search, optimization, and machine learning", Addison Wesley Longman, Inc., ISBN 0-201- 15767-5, 1989.
[4] Dijkstra, E. W., Dahl, O. J., Hoare, C. A. R.: "Structured programming",Academic Press., 1972.
[5] Wegener, J. and Pitschinetz, R.: "TESSY – Yet Another Computer-Aided Software Testing Tool?" Proceedings of the Second International Conference on Software Testing, Analysis and Review, Bruxelles, Belgium, 1994.
[6] Beizer B. "Software Testing Techniques". Van Nostrand Reinhold, 2nd edition, 1990.
[7] Ferguson R. and Korel B. "The chaining approach for software test data generation". IEEE Transactions on Software Engineering, 5(1):63-86,January 1996.
[8] Wegener, J., Grimm, K., Grochtmann, M., Sthamer, H. and Jones, B.:"Systematic Testing of Real-Time Systems". Proceedings of the Fourth European International Conference on Software Testing, Analysis & Review, Amsterdam, Netherlands, 1996.
[9] Offutt J. and Hayes J., "A semantic model of program faults". In InternationaSymposium on Software Testing and Analysis (ISSTA 96),pages 195{200. ACM Press, 1996.
[10] Ince, D. C.: "The automatic generation of test data", The Computer Journal, Vol. 30, No. 1, pp. 63-69, 1987.1141
[11] Taylor R.: 'An example of large scale random testing', Proc. 7th annual Pacific North West Software Quality Conference, Portland, OR, pp.339-48, 1989.
[12] Hamlet D. & Taylor R., "Partition testing does not inspire confidence",IEEE Transactions on Software Engineering, Vol. 16, 1990, pp. 1402-1411.
[13] Deason, W. H., Brown, D. B., Chang, K. H., and II, J. H. C. (1991). "A Rule-Based Software Test Data Generator". IEEE Trans. on Knowl. And Data Eng., 3(1):108–117.
[14] Myers, G. J. (1979). Art of Software Testing. John Wiley & Sons.
[15] Duran J. W. and Ntafos S. C.: 'An Evaluation of Random Testing',IEEE Transactions on Software Engineering, Vol. SE-10, No. 4, pp.438-444, July 1984
[16] Duran, J. W. and Ntafos S., 'A report on random testing', Proceedings 5th Int. Conf. on Software Engineering held in San Diego C.A., pp.179-83, March 1981
[17] Bertolino, A.: 'An overview of automated software testing', Journal Systems Software, Vol. 15, pp. 133-138, 1991
[18] Tsoukalas M. Z., Duran J. W. and Ntafos S. C.: 'On some reliabilityestimation problems in random and partition testing', IEEE Transactions on Software Engineering, Vol. 19, No. 7, pp. 687-697, July 1993
[19] Girard, E. and Rault, F. C.: 'A programming technique for software reliability', IEEE Symp. Computer Software Reliability, pp. 44-50,1973
[20] Michalewicz Z., "Genetic Algorithms + Data Structures = Evolution Programs", Springer-Verlag, 2nd edition, 1994.
[21] Gaurav Kumar, Pradeep Kumar Bhatia, "Software Testing Optimization through Test Suite Reduction using Fuzzy Clustering", CSI Transactions on ICT, Publisher - Springer India, Vol. 1, Issue 3, pp. 253-260, Sep. 2013.
[22] Bhawna, Gaurav Kumar, Pradeep Kumar Bhatia, "Software Test Case Reduction using Genetic Algorithm: A Modified Approach", International Journal of Innovative Science, Engineering & Technology, Vol. 3 Issue 5, pp. 349-354, May 2016.