Application of Genetic Algorithms in Software Testing

Faheem Ahmad*, Dr. Shafiqul Abidin**
**(Research Scholar, Shri Venkateshwara University Gajraula, Uttar Pradesh India)*
***(Head, Department of IT, HMR Institute of Technology & Management, New Delhi India)*
Corresponding Author: Faheem Ahmad

ABSTRACT: *Aim of software testing is to deliver a quality and reliable software to the client. To guarantee the quality of software, we need an effective software testing. This task need to confront certain issues like an effective test case, prioritization of experiments, etc. To defeat these issues, different systems and technique have been proposed. Genetic Algorithm (GA) is one of the evolutionary algorithm which produce an ideal solution for any issue. In this paper, we are going to quickly examine utilization of Genetic Algorithm in different types of software testing.*

KEYWORDS: *Genetic Algorithm, Software Development Life Cycle, Software Testing*

Date of Submission: 30-01-2019 Date of acceptance: 20-03-2019

I. INTRODUCTION

Software Testing is an activity in Software Development Life Cycle (SDLC) where the errors remaining from all the previous activities must be detected. Hence, Software testing performs a vital role in SDLC for ensuring software quality and reliability. During testing, system's behavior is monitored, so that we determine whether or not there is a failure. Testing can only reveal the presence of faults, not their absence [1].

Verification and validation processes can also be used to checking the software that whether or not it meets its requirement specification and the functionality expected by the user. Verification is made to ensure that the software meets specification and is related to structural testing whereas validation is related to the functional testing and is made by executing software under test [2]. The ultimate goal of verification and validation processes is to establish confidence that the software system is „fit for purpose" [3].

There are different kinds of software testing techniques. Broadly, there are two basic types of testing techniques: Black box testing and White box testing. Black Box Testing is also called functional testing because this testing is only concerned with the functionality of the software being developed.

White box testing is also called structural testing in which only concern is internal structure of the software. In white box testing, path testing considers 1. Control flow testing - it tests all possible paths of the control flow graph. 2. Data flow testing - during testing, it tests the definitions of variables and their subsequent use.

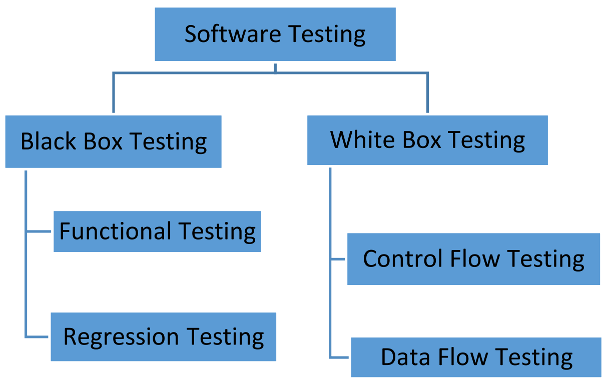


Fig 1. Basic types of Software Testing

Testing can be done either manually or automatically by using tools. As per as quality, performance, and cost of software development are a concern, it is found that automatic testing is better than manual. However, very few automatic test case generating tools are available today.

Various types of techniques have been proposed for generating test cases automatically. Recently, a lot of work is being done for automatic test cases generation using soft computing techniques like fuzzy logic, neural networks, Genetic Algorithm, and evolutionary computation providing keys to the problem areas of software testing. Genetic Algorithm often gives an optimal solution to all type of problems. Genetic Algorithm is an emerging methodology for automatic test case generation for various types of testing techniques. In this paper, various software testing techniques which perform using Genetic Algorithms are presented.

Testing should be possible either physically or naturally by utilizing tools. As per as quality, execution, and cost of software development are a worry, it is discovered that automatic testing is superior to manual. Notwithstanding, not very many automatic test tools or devices are accessible today.

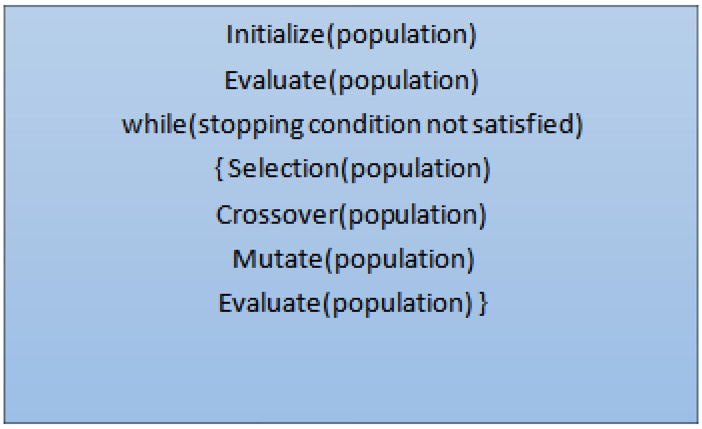
Different sorts of methods have been proposed for producing experiments naturally. As of late, a great deal of work is being accomplished for automatic test cases generation by utilizing fuzzy logic, neural systems, Genetic Algorithm, andevolutionary computation giving keys to the issue zones of software testing. Genetic Algorithm frequently gives an ideal answer for all kind of issues. Genetic Algorithm is a rising methodology for automatic test cases different sorts of testing methods. In this paper, different automatic testing strategies which perform utilizing Genetic Algorithms are exhibited

II. GENETIC ALGORITHM

Genetic algorithm (GA) is a kind of evolutionary algorithms. It is a general purpose and robust optimization technique based on the way nature evolves species using the natural selection of the fittest individuals. The possible solutions to the problem are represented by a population of chromosomes.

Genetic algorithm, (GA) is a sort of evolutionary algorithms. It is a broadly useful and vigorous optimization method dependent on the manner in which nature advances species utilizing the regular determination of the fittest people. The potential answers for the issue are spoken to by a populace of chromosomes.

A chromosome is a string of binary digits and each one digit that creates a chromosome is called a gene. This initial population can be totally random or can be created manually using the greedy algorithm. The pseudo code of a basic algorithm for GA is as follows [4].



```
Initialize(population)
Evaluate(population)
while(stopping condition not satisfied)
{Selection(population)
Crossover(population)
Mutate(population)
Evaluate(population)}
```

Fig. 2 The Pseudo code for basic GA Algorithm

GA uses three operators on its population which is described below:

A. Selection

To decide how people are favored for mating dependent on their wellness esteems is finished by choice plan. To start with, the wellness esteems can be characterized as the capacity of a person to endure and reproduce in an environment

Fitness values determined utilizing fitness capacity proposed in the calculation. Loads are utilized to locate the overall commitment of a way to the fitness computation. In result, more weight is doled out to a way which is increasingly "critical".

Determination plan produces the new populace from the bygone one, along these lines beginning another age. Each chromosome is assessed in present age to decide its fitness values. This fitness worth is utilized to choose the better chromosomes from the populace for the people to come. The fitness capacity is utilizing here is

$F= \sum_{i=1}^{n} wi$

Where, wi = weight assigned to ith edge on the path under consideration.

The algorithm works by assigning weights to the edges of Control Flow Graph on the basis of the importance of path in which the edge lies. Higher weights are assigned to the edges of the path corresponding to the critical section of the code for example branch statements, loops, control statements etc. for which testing is necessary. After all the fitness function values are intended, the possibility of selection pj for each path j, so that

$$pj = Fj/\sum Fj$$

n= initial population size

$$ck = \sum_{j=1}^{k} pj$$

Then cumulative possibility ck is measured for each path k with an equation [5][6].

B. Crossover or Reproduction (Recombination)

After selection, the crossover activity is connected to the chosen chromosomes. It includes a trade of genes or sequence of bits in the string between two people. Crossover occurs as per a crossover probability pc, which is a customizable parameter. For each parent chose, produce a random genuine number r in the range [0, 1]; if r< pc, at that point select the parent for crossover. From that point forward, the chose information is designed arbitrarily. Each pair of parents creates two new ways, called offspring. The crossover procedure utilized is one-point crossover done at the midpoint of the input bit string. After crossover, the mutation operator is used to a haphazardly chose subset of the populace [5].

C. Mutation

Mutation is done on a bit-by-bit basis. Mutation changes chromosomes in small ways to produce new good traits. It is used to bring variations in the population. Every bit of every chromosome in the offspring has an equal chance to mutate (change from “0” to “1” or from “1” to “0”), and the mutation takes place according to a mutation probability pm, which is also an adjustable parameter. To perform mutation, for each chromosome in the offspring and for each bit within the chromosome, create a random real number r in the range [0, 1]; if r < pm then mutate the bit.

III. GENETIC ALGORITHM IN SOFTWARE TESTING TECHNIQUES

In this segment, we will examine in insight regarding the uses of Genetic Algorithm in different testing systems [7]. Software testing is an advancement issue to limit the quantity of experiments and limit the time, cost and exertion. And furthermore enhance the quality of the product.

A. Uses of GA in White Box Testing

White Box Testing is utilized to test interior structure of a program. It incorporates articulations, restrictive explanations, circle proclamations. Structural testing intends to accomplish the experiments that will compel the ideal inclusion of various structures. In a portion of the exploration work examine the code inclusion, information stream testing, control stream testing and change testing utilizing Genetic Algorithm.

1)Control Flow Testing or Path Testing:

Praveen Ranjan Srivastava and Tai-Hoon Kim [8], has been proposed a method for distinguishing the most basic way groups in a program utilizing the genetic algorithm to produce experiments. This methodology utilizes a weighted CFG (Control Flow Graph). Way testing pursuits an appropriate experiment that covers each conceivable way in the program to discover the mistakes. In the event that the program has circles, at that point there will be an unending number of the way.

To cover every way, we need an enormous number of experiments, it turns out to be computationally unrealistic. Since it is difficult to cover all ways in the program, the way testing chooses a subset of ways to execute and discover experiments to cover it. CFG chooses an autonomous way for another arrangement of proclamations or condition. While testing, each freeway should navigate in any event once.

S. Keshavarz and Reza Javidan [9], proposed another method utilizing Genetic Algorithm to produce test information. In inclusion way testing, a test information is great information if causes to a free traversal of a way. The principle stress over the product testing is automatic and ordered information is required and adequate for testing. Information is a required and adequate just on the off chance that it causes a traversal on a freeway. For that, we offer a deliberate and computerized methodology to create information essential and adequate trial of a program dependent on program control stream chart and the secured point of basic edges.

Yeresime Suresh and Santanu Ku Rath [10], Worked on computerized test information age utilizing GA. Here the test information characterized as the populace in GA. In beginning populace, every individual piece string (chromosome) is a test information. This arrangement of chromosomes is utilized to create test information for practical premise ways. The system for creating test information for feasible basis path by utilizing GA is coded utilizing MATLAB. It arbitrarily creates the underlying populace, assesses the individual chromosome dependent on the fitness values and applies the GA activities, for example, selection, crossover, and mutation to deliver results. This iterative procedure stops when the genetic algorithm discovers ideal test information.

2). Information Flow Testing:

Moheb R. Girgis, Ahmed S. Ghiduk, and Eman H. Abd-Elkawy[11]. Dealt with automatic test way generation dependent on two proposed GA-based and PSO-based strategies that spread the all-utilizes measure for the program under test. These two procedures do their inquiry by developing new ways from recently created ways that are assessed as powerful test ways. At that point, we exhibited a GSO-based strategy that successfully consolidates the proposed GA-based and the PSO-based strategies to improve the person's score for common determination of the fitness and for good information sharing in the meantime.

In every emphasis of the proposed GSO calculation, the populace is isolated into two sections and they are developed with the two systems individually. They are then recombined in the refreshed populace, that is again partitioned haphazardly into two sections in the following cycle for another keep running of hereditary or molecule swarm administrators.

Na Zhang, Biao Wu and XiaoanBao [12], Proposed a strategy for creating test information consequently utilizing Multi-Population Genetic Algorithm. The calculation characterized the idea of outer weight which as the level of rivalry between people. Completely thinking about the impact of inclusion, branch condition and level of rivalry between individual types of three viewpoints, and give various loads, we structure a wellness capacity to assess the benefits of the individual species.

JanviBandlaney, RohitGhatol, RomitJadhwani [13], Presented a paper on a prologue to information stream testing. In his paper, they produced the possibility of a control stream testing. As indicated by them, control stream charts are a cornerstone in testing the structure of programming programs. By looking at the progression of control between the different segments, they planned and chose experiments. Information stream testing is a control-stream testing procedure which likewise looks at the existence cycle of information factors. The primary objective of their paper is to talk about the idea of information stream testing and applying it to a running model.

B. Use of GA in Black Box Testing

Discovery testing is which trying the functionality of a software and softwarefullfills their particular and client prerequisite. In some exploration performed, useful testing and relapse testing utilizing Genetic Algorithm.

1) Functional Testing

Francisca Eanuelle [14] has exhibited a GA-based procedure to produce great test plans for functionality testing in a fair way to evade the master's obstruction. The inspiration driving this work is to demonstrate that the GA can create great test plan despite the fact that the best grouping of the test plan is obscure. The test plan or test arrangement thoroughly depends on the specialists or the general population who comprehend the application well.

The accentuation is given on the way that "a blunder in a program isn't really because of the last activity executed by the client yet may have been because of an arrangement of recently executed tasks that leads an application in a conflicting state".

As it were, as an arrangement of tasks is executed, the condition of irregularity is non-diminishing or an issue in a product application is straightforwardly relative to the degree of irregularity of the state in which application is. In this work, the activity of huge granularity has been picked so the arrangement of activity that leads application to the conflicting state can be recognized.

Ruilian Zhao, Shanshan lv [15], utilized the neural system and GA for the practical testing. The neural system is utilized to make a model that can be taken as a capacity substitute for the SUT. The accentuation is given on the yields which display the significant highlights of SUT than information sources. All things considered, experiments ought to be produced from the yield area as opposed to enter space.

The feed forward neural system and backpropagation training algorithm are utilized for making a model. The neural system is prepared by recreating the SUT. The yields produced from the made model are sustained to the GA which is utilized to locate the comparing inputs with the goal that mechanization of experiments age from yield area is finished.

In this paper, inputs to the GA are the functional model produced from the neural system, various info factors n, scope of information factors that is upper [n] and lower [n], populace measure, most extreme cycle number, objective yield g, maximum fitness function f, crossover possibility and mutation possibility.

The fitness function is characterized as max where c is the genuine yield and the g is the objective yield of the SUT. The populace is assessed by applying GA.

The contrast between objective yield and the real yield of SUT utilizing the neural system is utilized for ascertaining fitness values of the people in the populace. In the event that fitness values surpass or achieves the greatest fitness value, at that point search stops and the current individual is taken as the test contributions for the relating yields.

2)Relapse Testing

N. Kaushik, M. Salehie, L. Tahvildari, and S. Li, M. Moore [16], proposed a worldview called Dynamic Prioritization which includes changing the request of experiments during the testing procedure. Since the experiment pool changes through the improvement cycle, the rundown of organized experiments would change too.

A. Kaur, S. Goyal [17] proposed another Genetic Algorithm to organize the relapse test suite is presented that will organize experiments based on complete code inclusion. The GA would likewise computerize the procedure of experiment prioritization.

A. Jiang, Y. Mu, Z. Zhang [18], Selects the experiments that can test some portion of changes and afterward do the decrease for these chose experiments. Notwithstanding the techniques referenced in this segment, an enormous number of strategies were proposed previously. A decent study of experiment prioritization strategies, just as calculations for ideal test arrangement investigation, can be found in [19] and [20].

IV. CONCLUSION:

Different methods have been proposed for test case generation; nobody could accomplish the best execution for each bit of code. Test case generation turns into an enhancement issue today. Along these lines, there are extension stays open for applying some more method to accomplish a superior outcome.

A Genetic algorithm is one such optimization technique. In this paper, the uses of Genetic algorithm in different programming testing methods have been examined. This will clear the way for further work toward this path

REFERENCES:

[1]. Pankaj Jalote, An integrated approach to software engineering, 3rd edition, Springer

[2]. Paul C. Jogersen, Software testing: A craftsman approach. 3rd edition, CRC presses, 2008.

[3]. Ian Somerville, Software engineering, 9th edition, Pearson, 2011.

[4]. Goldberg, D.E, Genetic algorithms: in search, optimization and machine learning, Addison Wesley, M.A, 1989.

[5]. Klir, G.J., Yuan, B.: Fuzzy sets and fuzzy logic: Theory and applications. Prentice-Hall Inc., Englewood Cliffs (1995)

[6]. Last, M., Eyal, S.: A Fuzzy-based lifetime extension of genetic algorithms. Fuzzy sets and systems 149(1), 131–147 (2005).

[7]. Chayanika Sharma, Sangeeta Sabharwal, RituSibal, Software testing techniques using genetic algorithm, IJCSI International Journal of Computer Science Issues, Volume 10, Issue 1,2013, 1694-0784

[8]. Praveen Ranjan Srivastava and Tai-hoon Kim, Application of genetic algorithm in software testing, International Journal of Software Engineering and Its Applications Volume 3,Issue 4,2009

[9]. S. Keshavarz and Reza Javidan, Member, IACSIT, Software quality control based on genetic algorithm, International Journal of Computer Theory and Engineering, Vol. 3, No. 4, August 2011

[10]. Yeresime Suresh and Santanu Ku Rath, A genetic algorithm based approach for test data generation in basis path testing, The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue: e-ISSN: 2251-7545, March 2013.

[11]. Moheb R. Girgis, Ahmed S. Ghiduk, and Eman H. Abd-Elkawy, Automatic data flow test paths generation using the genetical swarm optimization technique, International Journal of Computer Applications (0975 – 8887) Volume 116 – No. 22, April 2015.

[12]. Na Zhang, Biao Wu and XiaoanBao, Automatic generation of test cases based on multi-population genetic algorithm, International Journal of Multimedia and Ubiquitous Engineering Vol.10, No.6 (2015).

[13]. JanviBandlaney, RohitGhatol, RomitJadhwani, An introduction to data flow testing, NCSU CSC TR-2006.

[14]. Francisca Emanuelle et. al., "Using genetic algorithms for test plans for functional testing", 44th ACM SE proceeding, 2006, pp. 140 - 145.

[15]. Ruilianzhao, shanshan lv, "Neural network-based test cases generation using genetic algorithm" 13 IEEE international symposiums on Pacific Rim dependable computing. IEEE, 2007, pp.97 - 100.

[16]. N. Kaushik, M. Salehie, L. Tahvildari, S. Li, M. Moore (2011) "Dynamic prioritization in regression testing" IEEE fourth international conference on software testing, verification and validation workshops, pp: 135-138

[17]. A. Kaur, S. Goyal (2011) "A genetic algorithm for regression test case prioritization using code coverage", International journal on computer science and engineering, vol. 3, pp:1839-1847
[18]. A. Jiang, Y. Mu, Z. Zhang (2010) "Research of optimization algorithm for path-based regression testing suite", 2nd IEEE International workshop on education technology and computer science, pp:303-306
[19]. R. Kavitha, N. S. Kumar (2010) "Test case prioritization for regression testing based on severity of fault" (ijcse) international journal on computer science and engineering Vol. 02, No. 05, pp: 1462-1466
[20]. S. Raju, G. V. Uma (2012) "Factors oriented test case prioritization technique in regression testing using genetic algorithm" European Journal of Scientific Research, Vol.74, No.3, pp. 389-402.

Faheem Ahmad" Application of Genetic Algorithms in Software Testing" International Refereed Journal of Engineering and Science (IRJES), vol. 08, no. 03, 2019, pp 24-29