ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 8, August 2015

An Efficient Genetic Algorithm for Fault Based Regression Test Case Prioritization

Jyoti[1], Mrs. Lekha Bhambhu[2]

M.Tech, CSE, JCDM College of Engineering, Sirsa, India [1]

Asst Professor, CSE, JCDM College of Engineering, Sirsa, India [2]

Abstract: Software Testing is the process of validation and verification of product to provide the quality and efficiency and regression testing is the process of validating the modified software to detect errors that have been introduced into previously tested code. The test cases need to be generated as the software is modified and size of test suites will also increase. In this case, there is need of prioritization of test cases to reduce the cost of regression testing. There are number of methods for improving the efficiency in terms of time. The prioritization improves the effectiveness of regression testing by re ordering the test cases. The genetic algorithm (GA) has been proposed and implemented for prioritizing the test cases on the basis of fault coverage and execution time. The execution time and faults coverage has been given as input for test case number. The proposed algorithm has been automated and the results have been analyzed. The results representing the effectiveness of algorithm are presented with the help of GA. Total fault coverage with in time constrained environment on different examples is used for prioritization of test cases and their finite solution is obtained. Through Genetic Algorithm technique, an approach has been identified to find a suitable population, which was further formulated by GA operations to make it more flexible and efficient. The elaborations of results are shown with the help of .Net Technology. The algorithm has been automated and is analyzed for various examples.

Keywords: GA (Genetic Algorithm).

I. INTRODUCTION

Testing is the process of evaluating a system or its component with the intent to find that whether it satisfies the specified requirements or not, means, to detect the differences between existing and required conditions. Testing is executing a system in order to identify any gaps, errors or missing requirements in contrary to the actual desire or requirements. Software testing validates and verifies the software program. The errors are to be identified in order to fix those errors. The main objective of software testing is to maintain and deliver a quality product to the client. Each software is expected to meet certain needs. When software is developed it is required to check whether it fulfills those needs. Banking software is entirely different from software required in a shop. The needs and requirements of both those software are different. Hence it is important to check its potential. The main goal of software testing is to know the errors of the software before the user finds them. A good tester is one who makes the software fail.

NEED OF TESTING

The testing is the main part of software deployment. Software testing is needed to cover the risks of software implementation. Software testing helps to make sure that it meets the entire requirement it was supposed to meet.

i. It will bring out all the errors while using the software.
ii. Software testing helps to understand that the software that is being tested is a complete success.
iii. Software testing helps to give a quality certification that the software can be used by the client immediately.
iv. It ensures quality of the product.

WHITE BOX TESTING

In the white box testing, the testing of internal structures can be analyzed. There are different techniques under the white box testing. White box testing strategy deals with the internal logic and structure of the code. White box testing is also called as glass, structural, open box or clears box testing. The tests written based on the white box testing strategy incorporate coverage of the code written, branches, paths, statements and internal logic of the code etc. In order to implement white box testing, the tester has to deal with the code and hence is needed to possess knowledge of coding and logic i.e. internal working of the code. White box test also needs the tester to look into the code and find out which unit/statement/chunk of the code is malfunctioning.

REGRESSION TESTING

Regression means the act of going back to a previous place or state. Regression testing is a type of software testing that intends to ensure that changes means enhancements or defect fixes to the software have not adversely affected it. Whenever a change in a software application is made it is quite possible that other areas within the application have been affected by this change. To verify that a fixed bug hasn't resulted in another functionality or business rule violation is Regression testing. The intent of Regression testing is to ensure that a change, such as a bug fix did not result in another fault being uncovered in the application. It can also be defined as re-testing an application after its code has been modified to verify that it still functions correctly.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 8, August 2015

Regression testing consists of re-running existing test cases and checking that code changes did not break any previously working functions, inadvertently introduce errors or cause earlier fixed issues to reappear. These test cases should be run as often as possible with an automated regression testing tool, so that code modifications that damage how the application works can be quickly identified and fixed. There are necessary points for regression test suite

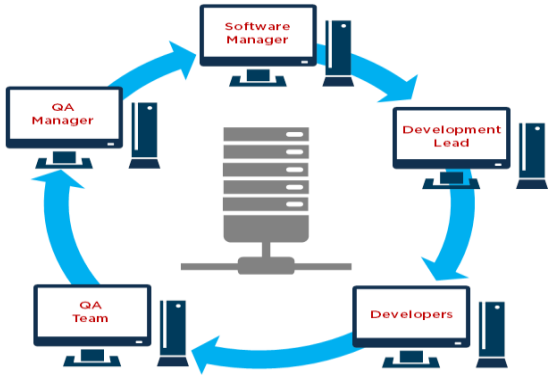

Figure 1: Regression Testing in Development

COST OF REGRESSION TESTING

Regression testing and retesting of modified software can be very costly as it tends to generate large number of test cases. Regression testing cost can be reduced significantly by identifying and testing the modified portion of the software. This avoids the costly construction of new test cases and the unproductive rerunning of existing test cases when it can be guaranteed that the unmodified code of software will produce the same results as it produced previously. There should be effective method of identifying the modified areas and use this information to reduce regression testing efforts. In this process, there is need to define and use a set of metrics to categorize the testing of software. There is also demonstration of the applicability of the approach using a genetic algorithm to reduce the cost of regression testing. There are different steps of GA (Genetic Algorithm) which reduce the cost and improve the success rate, which means, gives the best result according to the test cases simulation time.

GENETIC ALGORITHM

Genetic Algorithm (GA) is adaptive heuristic search algorithm premised on the evolutionary ideas of natural selection and genetic. The basic concept of GA is designed to simulate processes in natural system necessary for evolution. As such they represent an intelligent exploitation of a random search within a defined search space to solve a problem. To use a genetic algorithm, there is need to represent a solution to our problem as a genome (or chromosome). The genetic algorithm then creates a population of solutions and applies genetic operators such as mutation and crossover to evolve the solutions in order to find the best one. After an initial population is randomly generated, the algorithm evolves the through three operators.

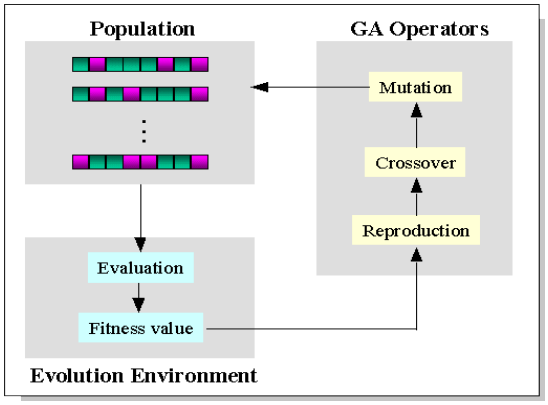

Figure 2: Genetic Algorithm Overview

GENETIC ALGORITHM ADVANTAGES

i. GAs deal directly with a population of solutions at any one time. These are spread throughout the solution space, so the chance of reaching the global optimum is increased significantly.
ii. Each solution consists of a set of discrete pipe sizes. One does not have to round diameters up or down to obtain the final solution.
iii. GAs identifies a set of solutions of pipe network configurations that are close to the minimum cost solution. These configurations may correspond to quite different designs that can be then compared in terms of other important but no quantifiable objectives.
iv. GAs use objective function or fitness information only, compared with the more traditional methods that rely on existence and continuity of derivatives or other auxiliary information.
v. It can solve every optimization problem which can be described with the chromosome encoding.
vi. It solves problems with multiple solutions.
vii. Structural genetic algorithm gives us the possibility to solve the solution structure and solution parameter problems at the same time by means of genetic algorithm.
viii. Genetic algorithms are easily transferred to existing simulations and models.

II. LITERATURE REVIEW

[1] Zheng Li et al [2] (2007), "Search Algorithms for Regression Test Case Prioritization", Transactions on Software Engineering, Vol. 33, No.4.
Regression testing is an expensive, but important, process. Unfortunately, there may be insufficient resources to allow for the re-execution of all test cases during regression testing. In this situation, test case prioritization techniques aim to improve the effectiveness of regression testing by ordering the test cases so that the most beneficial are executed first. Previous work on Regression test case prioritization has focused on Greedy Algorithms. However, it is known that these algorithms may produce suboptimal results because they may construct results that denote only local minima within the search space. By contrast, metaheuristic and evolutionary search algorithms aim to avoid such problems. Author presents results from

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 8, August 2015

an empirical study of the application of several greedy, metaheuristic, and evolutionary search algorithms to six programs, ranging from 374 to 11,148 lines of code for three choices of fitness metric. The paper addresses the problems of choice of fitness metric, characterization of landscape modality, and determination of the most suitable search technique to apply. The empirical results replicate previous results concerning Greedy Algorithms. They shed light on the nature of the regression testing search space, indicating that it is multimodal. The results also show that Genetic Algorithms perform well, although Greedy approaches are surprisingly effective, given the multimodal nature of the landscape. They described five algorithms for the sequencing problem in test case prioritization for regression testing. It presented the results of an empirical study that investigated their relative effectiveness.

[2] R.Krishnamoorthi et al [3] (2009), "Regression Test Suite Prioritization using Genetic Algorithms", International Journal of Hybrid Information Technology Vol.2, No.3.
Regression testing is an expensive, but important process in software testing. Unfortunately, there may be insufficient resources to allow for the re-execution of all test cases during regression testing. In this situation, test case prioritization techniques aim to improve the effectiveness of regression testing by ordering the test cases so that the most beneficial are executed first. In this paper they proposed a new test case prioritization technique using Genetic Algorithm (GA). The proposed technique prioritizes subsequences of the original test suite so that the new suite, which is run within a time-constrained execution environment, will have a superior rate of fault detection when compared to rates of randomly prioritized test suites. This experiment analyzes the genetic algorithm with regard to effectiveness and time overhead by utilizing structurally-based criterion to prioritize test cases. An Average [3] Percentage of Faults Detected (APFD) metric is used to determine the effectiveness of the new test case orderings.
Arvinder Kaur et al [4] (2011), "A Genetic Algorithm for Regression Test Case Prioritization Using Code Coverage", Vol. 3 No. 5, IJCSE.
The author has been assimilated the knowledge about Regression testing and the techniques for implementation. Regression testing is a testing technique which is used to validate the modified software. The regression test suite is typically large and needs an intelligent method to choose those test cases which will detect maximum or all faults at the earliest. Many existing prioritization techniques arrange the test cases on the basis of code coverage with respect to older version of the modified software. In their approach, a new Genetic Algorithm to prioritize the Regression test suite has been introduced that will prioritize test cases on the basis of complete code coverage. The genetic algorithm has also automated the process of test case prioritization. The results representing the effectiveness of algorithms are presented with the help of an Average Percentage of Code Covered (APCC) metric. The algorithm has been proposed to prioritize test cases using Genetic Algorithm. Here, different prioritization approaches have been analyzed, namely: total fault coverage with in time constrained environment and amount of code coverage on different examples and their finite solution obtained, respectively. Through Genetic
Algorithm technique, an approach has been identified to pull out suitable population, which was further formulated by GA operations to make it more flexible and efficient. The elaborations of results are shown with the help of APCC values. The APCC has been calculated for example for code coverage testing to evaluate the usefulness of the proposed algorithm.

[4] Wang Jun et al [5] (2011), "Test Case Prioritization Technique based on Genetic Algorithm", IEEE With the rapid development of information technology, software testing, as a software quality assurance, is becoming more and more important. In the software life cycle, each time the code has changed there needs to be regression testing. The huge test case library makes running a full test case library being challenged. To this end, they designed a genetic algorithm-based test case prioritization algorithm and improved the genetic algorithm proposed software test case prioritization algorithm.

[5] Liang You et al [6] (2012), "A Genetic Algorithm for the Time-Aware Regression Testing Reduction Problem", IEEE.
After the programmer fixes the bugs and enhances the functionality of the software project, regression testing reruns the regression testing suite to ensure that the new version software projects can run smoothly and correctly. Because the regression testing is the most expensive phase of the software testing, regression testing reduction eliminates the redundant test cases inthe regression testing suite and saves the cost of the regression testing. Author formally defines the time-aware regression testing reduction problem. It also proposes a novel genetic algorithm for the time-aware regression testing reduction problem. It defines the representation and fitness function of the genetic algorithm; it also describes the parent selection, crossover and mutation operator of the genetic algorithm. The novel algorithm not only removes redundant test cases in the regression testing suite but also minimizes the total running time of the remaining test cases. Finally, the paper evaluates the genetic algorithm using eight example programs. The experimental result illustrates the efficiency of the proposed genetic algorithm for the time-aware regression testing reduction problem.

[6] Amr Abdel et al [7] (2012), "Software Testing Suite Prioritization Using Multicriteria Fitness Function", ICCTA.
Regression testing is the process of validating modifications introduced in a system during software maintenance. It is an expensive, yet an important process. As the test suite size is very large, system retesting consumes large amount of time and computing resources. Unfortunately, there may be insufficient resources to allow

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 8, August 2015

for the re execution of all test cases during regression testing. Test case prioritization techniques aim to improve the effectiveness of regression testing, by ordering the test cases so that the most beneficial are executed first with higher priority. The objective of test case prioritization is to detect faults as early as possible. An approach for automating the test case prioritization process using genetic algorithm with Multi-Criteria Fitness function is presented. It uses multiple control flow coverage metrics. These metrics measure the degree of coverage of conditions, multiple conditions and statements that the test case covers. Theses metrics are weighted by the number of faults revealed and their severity. The proposed Multi-criteria technique showed superior results compared to similar work. The core of this technique was the Genetic algorithm which considers the test cases as genes and thegroups of test suite as chromosomes. It searches in a large search space of chromosomes for a solution so it can simply find the optimal solution fast in the early iterations. Genetic algorithm was employed to help in improving prioritization of test suites by proposing a new fitness function that considered the weights of the test cases, fault severity, fault rates and number of structural coverage items covered by each test case. The proposed multi-criteria technique was compared to other related work using a standard metric which is average percentage of fault detected (APFD).

III.PROPOSED METHODOLOGY

This section will cover the topic problem formulation, objective, methodology and results of Simulation. The problem definition section explains the existing problem in the test cases prioritization and explains the factor needs to be considered. After analyzing the problems, the objective and methodology has been explained for implementing the Genetic Algorithm and experimental results has been shown. This algorithm has been implemented in .Net Technology with Framework 4.0 and VC# Programming Language.

1. Problem Statement

In software testing, tests cases need to be generated for detecting the errors in software. But in regression testing, the modified software needs to be tested for identifying the errors and to validate them. Once the software has been modified, the test cases need to be executed again but now the number of test cases has been increased and there is no need to check every test case as in this situation, the cost of regression testing will be increased in terms of time.

So as per our literature survey we have finalized the following Problems and objectives.

i. Study of Test Cases Prioritization based on total code coverage.

ii. Analysis for Improvement in Regression Testing.

iii. Improve the effectiveness of regression testing by ordering the test cases for early execution of beneficial tests.

iv. Analysis on different kind of data with faults and Time.

2. Objective

i. To select and prioritize regression test suite within a time constrained environment

ii. To cover the total fault coverage.

iii. To implement efficient Genetic Algorithm (GA) for test cases that will cover major risks / faults in minimum time.

iv. To improve the success rate of Genetic Algorithm (GA).

v. Compare the efficiency of improved GA with existing Results.

IV.RESULTS

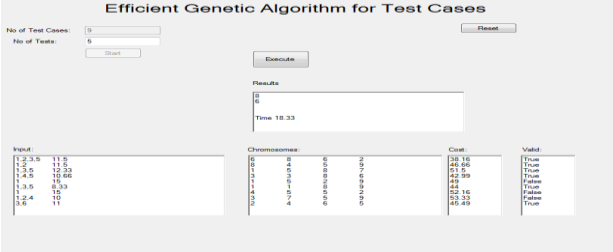

Figure 3: Output of Test Suite1

Result: 60% Efficiency

V. CONCLUSION

In the proposed GA (Genetic Algorithm), we prioritize test cases using Genetic Algorithm. The GA works on approximation and takes the input, total Test Case No, Faults coverage with execution time and has been implemented on different input values. Algorithm has prioritized the test cases with best time and their finite solution is obtained. By the genetic algorithm approach, the best suitable population has been analyzed and furthermore, the efficient results are generated in less time. The algorithm gives the best success rate. The algorithm has been automated and is analyzed for various examples..

REFERENCES

[1] G. Rothermel, "Prioritizing Test Cases For Regression Testing", IEEE Transactions On Software Engineering, Vol. 27, No. 10, 2011.

[2] Zheng Li, Mark Harman, and Robert M. Hierons, "Search Algorithms for Regression Test Case Prioritization", Transactions on Software Engineering, Vol. 33, No.4, 2007.

[3] R.Krishnamoorthi, S.A.Sahaaya, Arul Mary, "Regression Test Suite Prioritization using Genetic Algorithms", International Journal of Hybrid Information Technology Vol.2, No.3, 2009.

[4] Arvinder Kaur, ShubhraGoyal, "A Genetic Algorithm for Regression Test Case Prioritization Using Code Coverage", International Journal of Computational Science and Engineering, Vol. 3 No. 5, 2011.

[5] Wang Jun, Zhuang Yan, "Test Case Prioritization Technique based on Genetic Algorithm", IEEE,2011.

[6] Liang You Yansheng Lu, "A Genetic Algorithm for the Time-Aware Regression Testing Reduction Problem", IEEE,2012.

[7] Amr Abdel Fatah Ahmed, Dr. Mohamed Shaheen, "Software Testing Suite Prioritization Using Multicriteria Fitness Function", International Conference on Computer Theory and Applications,2012.

[8]Suman, Seema, "A Genetic Algorithm for Regression Test Sequence Optimization",International Journal of Advanced Research in Computer and Communication Engineering, Vol. 1, Issue 7, September, 2012.

[9] BhartiSuri, IshaMangal ,"Analyzing Test Case Selection using Proposed Hybrid Technique based on BCO and Genetic Algorithm and a Comparison with ACO", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 4, April 2012.

[10] Ruchisehrawat, Varunsrivastava, "A Neuro-Genetic Algorithm for Selection of Test Suites in Regression testing", International Journal of Computer Application, Issue2, Volume 1, 2012.