Novel Approach for Test Case Prioritization in Regression Testing

[1]Anku Sharma, [2]Nancy Sehgal

[1]Research Scholar, Baddi University of Emerging Science and Technology,Punjab

[2]Assistant professor, Baddi University of Emerging Science and Technology, Punjab, India

Abstract-The process of verifying the modified software in the maintenance phase is known as Regression testing. Time and budget constraints are its major disadvantage due to complex process. This work is based on manual slicing and automated slicing for test case prioritization to detect maximum number of faults from the project in which some changes are done for the new version release. The mutation value is calculated from which the best fitness value is calculated which will be the importance of the particular function. To test the performance of proposed and existing algorithm MATLAB is being used by considering the dataset of ten projects. Each project has seven functions and four numbers of changes are defined for the regression testing. In the simulation it is being analyzed that fault detection rate is increased and execution time is reduced with the implementation of automated test case prioritization as compared to manual test case prioritization in regression testing.

Keywords**-**Regression Testing;Test Case Prioritization; Functional Importance.

I. INTRODUCTION

A software engineering is related to all the aspects that are used in the software production. Software is basically a generic term, which is used for organizing the data and instructions that are collected to develop it. The software is broken into the two categories: system software and the application software. The system software is used to manage the hardware components, so that other software or user sees it as a functional unit. Software is the program or set of programs [1]. It is different from the program in many ways. As in software many things are includes: as it consists of the programs, the complete documentation of that program, the procedure that is use to set up the software and the various operation of the software system. Any program is the subset of the software. As software requirement is increases day by day. So it is necessary to maintain the good quality software. To develop good quality software, software engineering is required. For this, the developer's needs to adopt the software engineering concepts, strategies, and practices to avoid the conflicts that are occur during the development process [2]. Software engineering is an approach to develop, maintain and operate the software. The software development plays a crucial role in software engineering. A test case is set of procedure use to test the software. Test case is a set of condition under which under which a software tester determine whether the application or software system is working correctly or not [3]. To design a test case for particular software the designer must design positive or negative test case for the software. Positive test cases are design to check software under normal condition and negative test case are design to check software at extreme condition. The order of test case execution affects the time at which goal of testing are fulfill. If the goal is fault detection then a improper execution order might reveal most of fault late which leads to delay in bug fixing activity and the delivery of software. A mechanism is needed for arranging a test case in appropriate order to increase their effectiveness at meeting some performance goal and rate of fault detection such mechanism is known as test case prioritization. Test case prioritization is a method to prioritize and schedule test cases in appropriate order. To run test cases of higher priority before than the lower priority test case in order to minimize time, cost and effort during software testing phase [4]. Software testing is a procedure of testing or comparing the actual outcome with the expected outcome. Testing of the software is done in order to check the correct functionality of the system or project. If the testing will not be performed then system may lead to catastrophic or improper results in the field. So it's better to check or test the system earlier, so that the excellent results can be produced. Regression testing is a testing that refers to that section of the test cycle in which programs are tested to make sure that changes do not affect features that are not believed to be affected. The process of verifying the customized software in the maintenance phase is known as Regression testing. Time and budget constraints are its major disadvantage due to complex process. Regression testing is the re-execution of a number of subset of test that has previously been conducted. In regression testing as integration testing takings, number of regression tests increases and it is not practical and ineffective to re execute every test for each program function if once change occurs. It is an expensive testing process used to detect regression faults. Research has shown that at least 50% of the total software cost is comprised of testing activities [5]. Companies are often faced with lack of time and resources, which limits their ability to effectively complete testing efforts. Prioritization of test cases in the order of execution in a test suite can be beneficial. Genetic Algorithm originated from the studies of cellular automata. A Genetic Algorithm is basically a searching techniques, it is used in the computer science. It helps to find approximate solutions for any optimization problems. The genetic algorithms are known as the evolutionary algorithms. In this many techniques are involved by evolutionary biology such as inheritance, mutation, natural selection, and recombination. In the representation of the genetic algorithms the fitness function is defined [6]. The genetic algorithm proceeds to initialize the solutions randomly. It used to improve it through repetitive application. In this case it involves many applications such as: mutation, crossover, and selection operators. Many Researchers have adopted genetic algorithms as a solution to optimization in various fields. The genetic algorithms acts as a solution to optimization problem started gaining popularity towards the end of the last century as used to solve optimization problems in construction.

II. LITERATURE REVIEW

Thillaikarasi Muthusamy et.al (2014) studied that Regression testing concentrates on finding defects after a major code change has occurred. Specifically, it exposes software regressions or old bugs that have reappeared. It is an expensive testing process that has been estimated to account for almost half of the cost of software maintenance. To improve the regression testing process, test case prioritization techniques organizes the execution level of test cases. Further, it gives an improved rate of fault identification, when test suites cannot run to completion. The algorithm is based on analysis of the percentage of test cases performed to find the faults and on APFD metric's results. Abiding by the percentage of executing test cases in earlier fault detection is important as sometimes regression testing ends without executing all test instances [7].

Bharti Suri et.al (2012) have proposed Hybrid technique based on BCO for analyzing text case selection and by applying this technique new tool generate. Their results show that a huge amount of reduction in test suite takes place. Reduction in test suite reduces time as well as cost. They have proposed hybrid approach combining BCO and genetic algorithm which proves much faster than ACO technique. The tool which they developed runs much faster to provide the minimum subset of test cases. The tool can provide different results in each run. This implementation is done to improve correctness and efficiency of the tool [8].

Suman et.al (2012) have discussed Regression testing is the process of validating modified software to assure that changed parts of software behave as intended and unchanged parts of software have not been adversely affected by the modification. 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 approach, a new Genetic Algorithm to prioritize the regression test suite is introduced that will prioritize test cases dynamically on the basis of complete code coverage. Meanwhile, an approach to generating new test cases is presented using PMX and cyclic crossover and analysis is done on the basis of process cost and test cost. The overall aim of this research is to reduce the number of test cases that need to be run after changes have been made [9].

Thillaikarasi Muthusamy (2013) revealed that scheduling test cases by using test case prioritization technique enhances their efficiency of attaining some performance criteria. The rate at which the errors are detected within the testing process is one such criterion. To achieve performance requirements test cases with higher priority are executed than those with lower priority by test case prioritization techniques. This proposed test case prioritization algorithm prioritizes the test cases based on four groups of practical weight factors such as Time factor, Defect factors, Requirement factor and complexity factors. The proposed technique is validated with three different validation metrics and is experimented using two projects. The algorithm illustrated detects serious errors at earlier phases of testing process and effectiveness between prioritized and non-prioritized test cases is compared using ASFD [10].

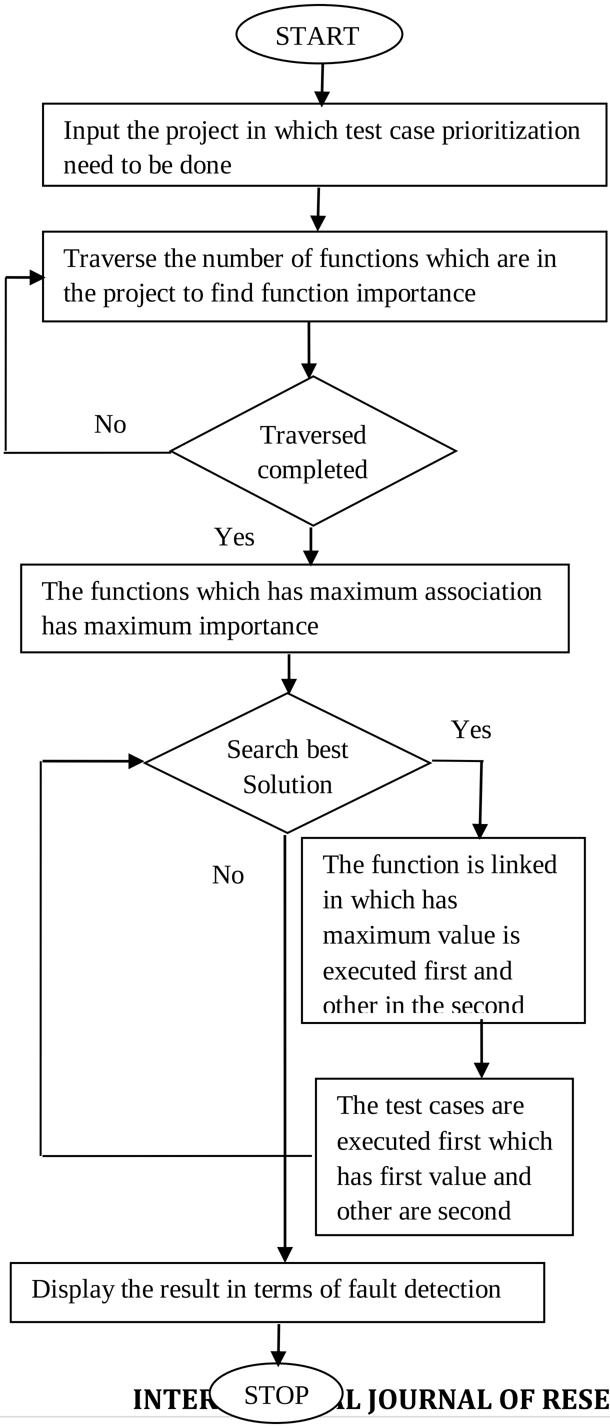
Shifa-e-Zehra Haidry et.al (2012), presented that several existing test case prioritization techniques consider that test can be run in any order. But due to functional dependency that exist between the test case means the one test case must be executed before another. This paper presents test case prioritization techniques that use dependency information from test suit to prioritize the test suit. These techniques observe the dependency between the test cases and use this information to prioritize the test case. In this paper two techniques are used to find the dependency between the test cases. These are open dependency and closed dependency. The test cases that have more dependents the coverage value is higher of those test cases. This paper offers a solution to test case prioritization problem when dependency exists between the test cases [11].

Siripong Roongruangsuwan et.al (2010) proposed two new efficient methods for test case prioritization. The first method is developed to solve the problem of many test cases having the same weight values. The second method is designed to prioritize multiple suites efficiently. These two methods minimize a prioritization time. This paper mainly gives attention to test case prioritization techniques only. They compare test program behaviour against the source code. In contrasts with functional black-box testing, this technique compares test program behaviour against a requirements specification. Third, the cost effective-based techniques which prioritize test cases based on cost factors, like cost of analysis and cost of prioritization. Last, the chronographic history-based techniques which prioritize test cases base on test execution history factors. From this paper we also learn that there are many difficulties and gaps in the test case prioritization [12].

III. RESEARCH METHODOLOGY

To increase the fault detection rate of the test case prioritization, automated test case prioritization is being implemented in this work.

Fig.1: Flowchart of Proposed Work


START
Input the project in which test case prioritization need to be done
Traverse the number of functions which are in the project to find function importance
No
Traversed completed
Yes
The functions which has maximum association has maximum importance
Search best Solution
Yes
No
The function is linked in which has maximum value is executed first and other in the second
The test cases are executed first which has first value and other are second
Display the result in terms of fault detection
STOP

In the first step of the algorithm, the population values are taken as input which is the number of times function encountered and number of functions associated with a particular function. In the second step, the algorithm will start traversing the population values and error is calculated after every iteration. The iteration at which the error is the highest at that point the mutation value is calculated as the best mutation value of the function. The function mutation value will be the function importance from where the test cases are prioritized according to the defined changes. In the last step of the algorithm the function importance values are accessed according to the defined changes and best fitness value is calculated which will be the final percentage of faults detected from the project after the particular change.

Following are the various steps of proposed multi-objective algorithm.

A. In the improved multi-objective algorithm, the function importance is also calculated on the basis of number of functions associated. The function which has maximum association is considered as the most important function

B. To calculate the number of functions associated, the technique of automated slicing is been applied which traverse the DFD and generate final result.

C. The automated slicing will work in the iterative manner and search the best value of the test case as which maximum number of errors get detected from the project.

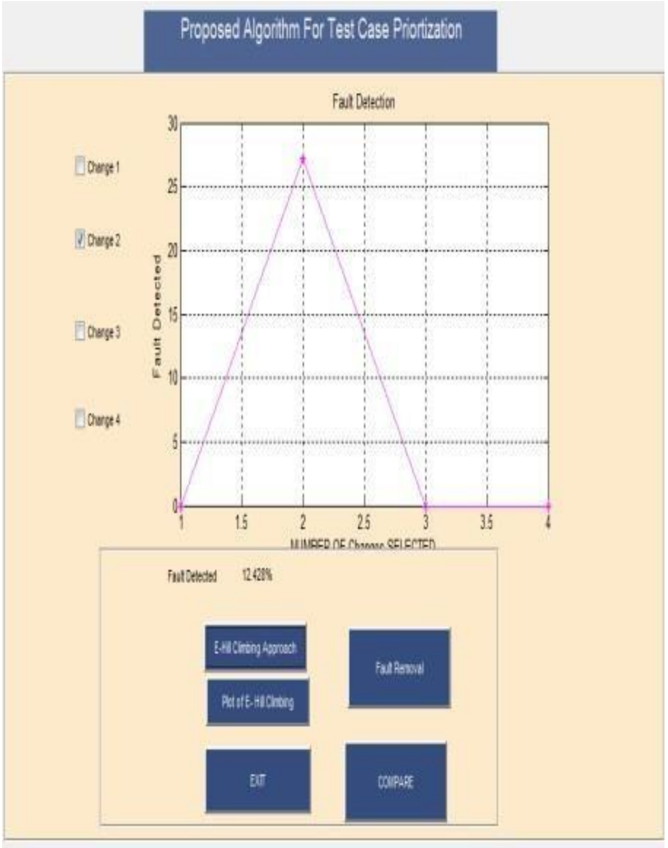
IV. EXPERIMENTAL RESULTS

The proposed work has been implemented in MATLAB and the results are evaluated in terms of several parameters.

The proposed technique is implemented in MATLAB and the simulation results achieved are compared in terms of fault detection rate and execution time. The simulation parameters are described in table 1

Table 1: Simulation Parameters

Simulation	Values
Number of test cases	10
Coverage value	YES
Prioritization	Ascending order
Number of changes	4



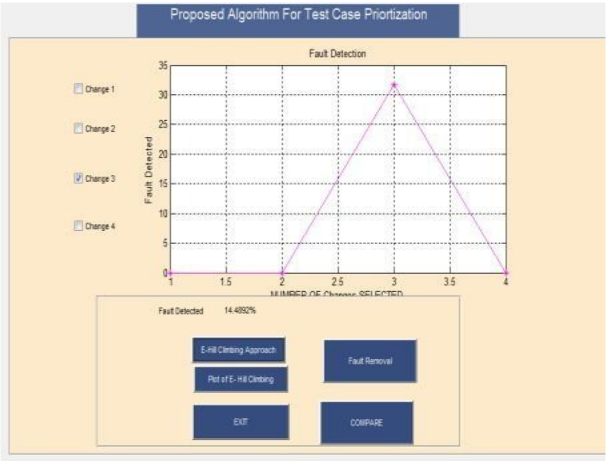
Fault Detected 12.428%

Fig.2:Fault Detected with respect to change2.

As shown in figure 2, the automated slicing technique is applied which detect fault from the software. The automated testing is applied to enhance multi-objective model. The experiments are performed on the certain number of changes and 12.42 percent faults are detected with respect to change 2.

Table 2: Compression

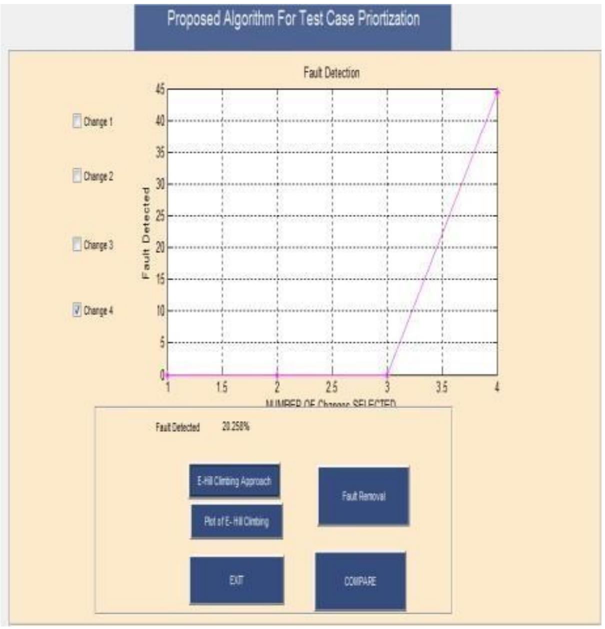
Change Number	Existing Algorithm	Proposed Algorithm
1	5.45	8.90
2	7.89	10.23
3	10.78	12.56
4	19.98	23.45



Fault Detected 14.4892%

Fig.3:Fault Detected with respect to change3.

As shown in figure 3, the enhance multi-objective is applied in which automated slicing technique is applied for the fault prediction. In the proposed approach the 14.4892 percent faults are predicted correspond to change 3.



Fault Detected 20.258%

Fig.4:Fault Detected with respect to change4

As illustrated in figure 4, the automated slicing technique is applied which is enhance multi-objective model. The 20.258 percent faults are predicted correspond to change 4

As shown in table 2, the fault detection value of proposed and existing algorithm is compared It is analyzed that

V. CONCLUSION

In this work, it is concluded that regression testing is the type of testing which is applied to test the project after some changes are being done for future release. The test case prioritization is the technique of regression testing which is being applied to prioritize the test cases according to the defined changes. To analyze the performance of proposed and existing algorithm simulation is being done in MATLAB by considering ten projects with four changes. It is been analyzed that fault detection rate is increased and execution time is reduced by applying automated test case prioritization as compared to manual test case prioritization in regression testing.

VI. REFERENCES

[1]. Hyunsook Do, Siavash Mirarab, landan Tahvildari, and Gregg Rothermel, "The effect of time constrainton test case prioritization", 2010, IEEE TRANSACTION ON SOFTWARE ENGINEERING,VOL.36

[2]. Gregg Rothermel, Roland H.untch,Chengyun Chu, "prioritize the test case for Regression testing", 2001, IEEE Transactions on Software Engineering.

[3]. Hema Srikanthi, Laurie williamsi, Jason Osborne, "system test case prioritization of new regression test case", 2005, IEEE

[4]. Sebastian Elbaum, Gregg Rothermel,y Satya Kanduri,z Alexey G. Malishevsk, "Selecting a Cost-Executive Test Case Prioritization Technique", 2004.

[5]. Lingming Zhang, Ji Zhou, Dan Hao, Lu Zhang, Hong Mei, "Prioritizing JUnit Test Cases in Absence of Coverage Information", 2009, IEEE .

[6]. Paolo Tonella, Paolo Avesani, Angelo Susi, "Using the Case-Based Ranking Methodology for Test Case Prioritization", 2009, 22nd IEEE International Conference on Software Maintenance (ICSM'06).

[7]. Thillaikarasi Muthusamy and Dr. Seetharaman. K, "Effectiveness of Test Case Prioritization based on Regression Testing", International Journal of Software Engineering & Applications (IJSEA), Vol.5, No.6, November 2014

[8]. Mangal, B. S, "Analyzing Test Case Selection using Proposed Hybrid Technique based on BCO and Genetic Algorithm and a Comparison with ACO", IJCA, 2012.

[9]. Suman and 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

[10]. Thillaikarasi Muthusamy and Dr. Seetharaman.K, "A Test Case Prioritization Method with Weight Factors in Regression Testing Based on Measurement Metrics", International Journal of Advanced Research in Computer Science and Software Engg 3(12),December -2013, pp. 390-396

[11]. Shifa-e-Zehra Haidry and Tim Miller, "Using Dependency Structures for Prioritisation of Functional Test Suites", 2012, IEEE.

[12]. Siripong Roongrungsuwan, Jirapun Daengdej, "TEST CASE PRIORITIZATION TECHNIQUES", 2010, Journal of Theoretical and applied information Technology, JATT&LLS.