An Adaptive Hybrid Technique approach of Test Case Prioritization

Zubair Rashid Bhat[1], Mudasirahma Dmutto[2]

[1]RN Engineering & Management College Rohtak, Haryana, India
[2]Department of CSE, Assistant Professor, SSM College of Engineering &Technology, Jammu and Kashmir, India

Abstract— *Test-Case Prioritization is the method to schedule any execution order of the test with the purpose of maximizing some objects like revealing faults early. In this paper we have proposed the hybrid approach for the purpose of the test case prioritization involving Robust Genetic Algorithm to improve the parameters like APSC and execution time. This technique involves robust approach, parent generation, cross-over and mutation over each test-case and then calculates APSC and execution time.*

Keywords— Test Case Prioritization, software testing, Regression testing.

I. INTRODUCTION

Software testing is one of the important steps in measuring the quality of the software system. As we know that most of the software development cost is for the purpose of its testing and maintenance so it has become a great issue for the developers to reduce the cost of the testing and maintenance.

Test case prioritization was first introduced in the regression testing whose purpose is to test the change in software during its evolution by reusing the test cases of its previous version that is before the modifications of it. In this approach for the purpose of regression testing test case prioritization maintains the schedule for the execution order of the test cases for maximizing some objects like revealing faults.

Test case prioritization is the technique for scheduling the test cases. With the help of scheduling these test cases, the effectiveness can be increased to achieve some performance goals. Test cases help in order to detect the faults in the system and the occurrence of fault rate. Test case prioritization is the expensive technique to be used, but also very necessary for validating and improving the quality of the software. Test case prioritization technique is used to prioritize the test cases in order to test the cases with higher priority than the cases having lower priority. Test cases are being prioritized for achieving cost effectiveness, time and efforts needed for the software system. Test cases must be prioritized in order to increase the fault detection and finding higher severity faults earlier. These test cases are being contained in a particular test suite. Test suite is thus the collection of the test cases.

Various test case prioritization techniques are named as under:-

Total Coverage Prioritization: - The technique to sort the various test cases by the total number of functions and statements covered. Test case prioritization arranges these test cases according to the number of transitions covered and executed.

Total Property Prioritization: - Total property prioritization sorts the test cases according to the number of properties these test cases are relevant to. Each test case is being prioritized on the property bases. The test cases are being tested for the cases that are affected by the property violation.

Additional Coverage Prioritization: - The technique in which the test cases having larger amount of coverage, are tested and executed first. But the time may be consumed in this technique.

Total FEP Prioritization: - The technique is named as Fault Exposing Potential. This technique is applied to arrange the test cases according to the maximum number of faults detected or exposed. Mutation score technique is used in total FEP prioritization. Mutation score is the ratio of mutants. These mutants are distinguished from original program.

Additional FEP Prioritization: - The technique is similar to additional coverage prioritization. The test cases are being arranged according to the number of additional, but undetected mutants. FEP prioritization technique is more complex than the coverage based technique.

Optimal Prioritization: - This technique is based on the required information of the known mutants, which is not applicable as in practise. The faults are being detected with the help of minimum number of test cases.

Random Prioritization: - The random ordering of the test cases can be applied to test the test cases. Any ordering of the test cases can be determined to prioritize the test cases.

There are many software testing techniques like: Black-Box testing, white box testing, regression is testing.

Genetic Algorithm: Although test case prioritization using genetic algorithm gives satisfactory results, it can

be quite time consuming at the same time. This is so because genetic algorithm is an iterative process which involves a population of chromosomes (test cases in this context) being repeatedly evolved to generate a better solution. During each iteration, the fitness function for each individual of the population is calculated andthe more fit individuals are selected for the next iteration. This process continues till the best fit chromosome is achieved. For very large population such as 100 test cases, it can take significant amount of time.

Regression testing

Itis a type of testing aimed at finding out new errors after the modification of software. In other words, it assures that no additional errors were introduced in the process of fixing other problems and the software still works as it did before. However, executing the entire test suite for this purpose can be expensive in terms of cost and time. Therefore, it is essential to reduce the expense involved in regression testing. One way to achieve this is by test case prioritization. Test Case Prioritization aims to order the test cases so as to maximize the rate of fault detection. This topic has been a major subject of research for many years and many techniques have been proposed for achieving the same. Popular among these are genetic algorithm, ant colony optimization, bee colony optimization and particle swarm optimization.

Although these techniques are able to efficiently prioritize the test cases, but take significant amount of time to do so. For example in case of genetic algorithm, a large population of candidate solutions has to be repeatedly evolved in an iterative manner for reaching the best solution. When the population is large i.e. when there are some 100 or 1000 test cases, then genetic algorithm may consume a large proportion of time to prioritize the test cases. Toremove this drawback, a new hybrid technique has been proposed which speeds up the time taken to prioritize the test cases. This hybrid technique is a combination of adaptive approach and genetic algorithm. The speedup is achieved by using an adaptive approach which schedules the test cases simultaneously during the execution of test cases.

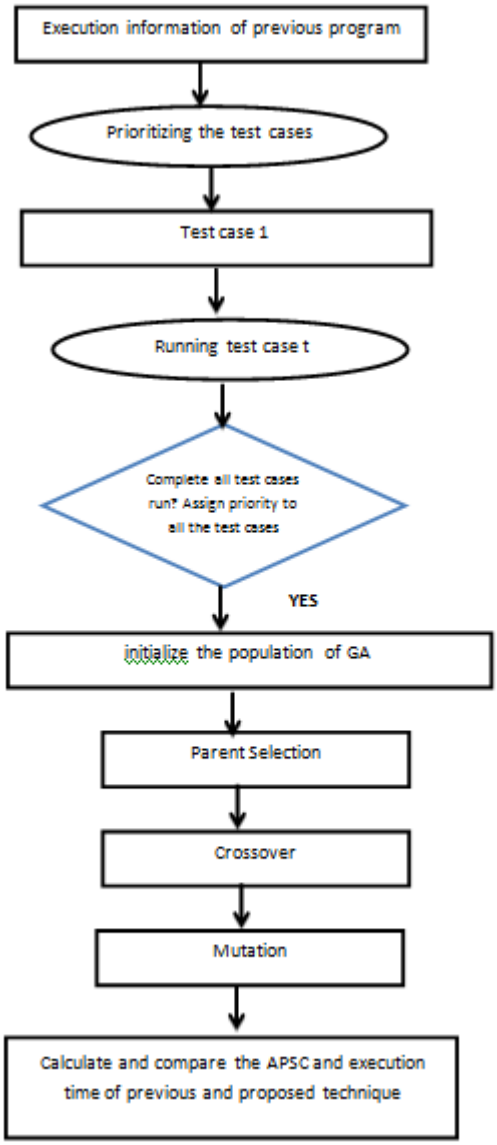
Robust Genetic Algorithm is the Hybrid proposed test-case prioritization approach in this approach we ordering the test case and find the average percentage of statement coverage for hundred test cases in java . First we measure the APSC of Robust approach and ordering the test case.

In Robust approach we order the tpest case like until our statement not covers if test cases left or we can say failure test cases those are unable to cover any statement its means the statement coverage is not done perfectly. We take that Left test cases after applying robust approach and perform genetic algorithm on these test case. In Genetic algorithm we apply three main techniques to order the test case like this our APSC improved as compared to robust approach.

We apply these techniques in genetic algorithm to giving the order to each test case

- Robust Approach
- Parent Generations
- Cross Over
- Mutations
- Measures APSC
- Execution time

II. FLOWCHART FOR THE PROPOSED APPROACH



Following are the hardware and software requirements for my thesis work:

Hardware Requirements:

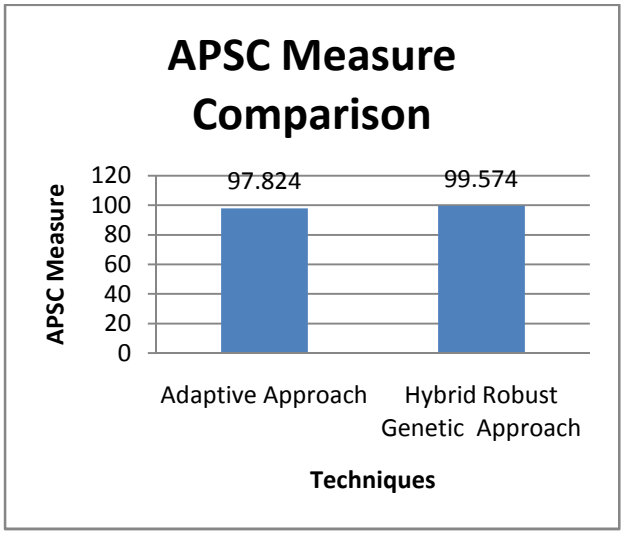
- Processor: Pentium IV of Higher
- RAM: 256 MB or Higher
- Hard Disk: 10GB or Higher

Software Requirements:

- Platform: Microsoft windows 7 or higher.
- JAVA Runtime Environment
- JAVA Development Kit
- Netbeans Integrated Development Environment
- MySQL Relational Database
- JAVA Object Oriented Programming Language

APSC Measure98.65189837291837
EXECUTION TIME: 1144ms
APSC of the Left test cases: 97.82458040863276

Test case ordering results using New Improved Genetic Approach and the execution time and the Average percentage statement coverage given below



The above graph shows that APSC of the proposed approach Hybrid Robust Genetic performs betters than the previous approach. APSC of proposed technique is 99.574% and the adaptive approach is 97.824%

Conclusion: The outcome of proposed technique is to provide the higher efficiency as much as possible. Code coverage will be applied to count the total number of lines of codes in our software and by using it, we can check and prioritize the test cases according to the number of lines of codes and the execution time of the software code. In this Research we proposed an approach that improves APSC (average percentage of statement coverage). Our work is extension into the adaptive approach for APFD (average percentage of fault detection) into adaptive genetic algorithm hybrid approach from which we conclude that our proposed approach improved the APSC.We take hundred java test cases package of apache server to evaluate our approach. First we apply adaptive approach and calculate APSC. Than we apply our proposed algorithm robust genetic algorithm hybrid approach than we calculate APSC than we found that our approach gives better results than adaptive approach for APSC only. Basically in this research we focused on APSC only but while we calculate Execution time for both approach we found that our proposed approach take large time to execute as compare to adaptive approach. But as the tester view our main aim to cover all statements of the code for better quality. So, we considering this work as our next future work and we believe that if we apply any other technique we can improve execution time as well APSC together. And we take small data set in our research while in future we take large data set of test cases for efficient results

REFERENCES

[1] Solanki and Y. Singh, "Novel classification of test case prioritization techniques," International Journal of Computer Applications, 2014.
[2] A. Jatain and G. Sharma, "A systematic review of techniques for test case prioritization," International Journal of Computer Applciations, 2013.
[3] D. A. Kaur and S. Goyal, "A Systematic Review of Techniques for Test Case Prioritization," International Journal of Computer Applications, 2013.
[4] D. M. Hashini, "Clustering approach to test case prioritization using code coverage metric," International journal of engineering and computer science, 2014.
[5] J. Badwal and H. Raperia, "Test case prioritization using clustering," International journal of current engineering and technology, 2013.
[6] T. P. Jacob and D. T. Ravi, "Optimal regression test case prioritization using genetic algorithm," life Science Journal, 2013.
[7] M. Athar and I. Ahmad, "Maximize the code coverage for test suit by genetic algorithm," International journal of computer science and informatin technologies, vol. 5(1), pp. 431-435, 2014.
[8] M. M. B. Patel, "Test case prioiritzation for regression testing using ant colony optimization," International journal for scientific research & development, vol. 2, no. 04,2014, pp. 632-636, 2014.
[9] N. Sethi, S. Rani and P. Singh, "Ants optimization for minimal test case selection and prioritization as to reduce the cost of regression testing," International journal of computer applications, vol. 100, pp. 48-54, 2014.
[10] I. Sharma, J. Kaur and M. Sahni, "A test case prioritization approach in regression testing," International journal of computer science and mobile computing, vol. 3, no. 7, pp. 607-614, 2014.
[11] M. Shahid and S. Ibrahim, "An Evaluation of Test Coverage Tools in Software Testing," International

Conference on Telecommunication Technology and Applications, vol. 5, no. 2011, pp. 216-222, 2011.

[12] P. R. Srivastava, "Test case prioritization," Journal of theoretical and applied information technology, 2008.

[13] Yuejian Li, Nancy J. Wahl, "An Overview of Regression Testing", 1999 ACM SIGSOFT SoftwareEngineering Notes volume 24 no 1.

[14] S. Yoo, M. Harman, "Regression testing minimization, selection and prioritization: a survey", 2010 Software Testing, Verification and Reliability.

[15] Gregg Rothermel, Roland H. Untch, Chengyun Chu, Mary Jean Harrold, "Test Case Prioritization: An Empirical Study" 1999 IEEE International Conference on Software Maintenance, (ICSM '99) Proceedings.

[16] HemaSrikanth, Laurie Williams, Jason Osborne, "System Test Case Prioritization of New and Regression Test Cases"

[17] Paolo Tonella, Paolo Avesani, Angelo Susi, "Using the Case-Based Ranking Methodology for Test Case Prioritization" 2006 22nd IEEE International Conference on Software Maintenance.

[18] Xiaofang Zhang,Changhai Nie,Baowen Xu, Bo Qu, "Test Case Prioritization based on Varying Testing Requirement Priorities and Test Case Costs" 2007Seventh International Conference on Quality Software.

[19] Yu-Chi Huang, Chin-Yu Huang, Jun-Ru Chang Tsan-Yuan Chen, "Design and Analysis of Cost-Cognizant Test Case Prioritization UsingGenetic Algorithm with Test History"2010 IEEE 34th Annual Computer Software and Applications Conference.

[20] R. Kavitha, V.R. Kavitha, Dr. N. Suresh Kumar, "Requirement Based Test Case Prioritization" 2010 IEEE International Conference on Communication Control and Computing Technologies (ICCCCT).

[21] Yogesh Singh, ArvinderKaur, BhartiSuri, "Test Case Prioritization using Ant Colony Optimization" 2010 ACM SIGSOFT Software Engineering Notes Page 1 Volume 35 Number 4.