Multi-Objective Test Case Prioritization Techniques for Regression Testing: A Review

Anita Saini¹ Dr. Sanjay Tyagi²
¹,²Department of Computer Science & Applications
¹²Kurukshetra University, Kurukshetra

Abstract—Regression testing is an expensive and important process of validating modification introduced in a system during software maintenance. A number of regression testing techniques namely Retest all, Regression test selection, Test case reduction and Test case prioritization have been proposed to improve the effectiveness of regression testing. The important concern in regression testing is its cost and execution time. The most familiar technique in regression testing is Test case prioritization that is used to deal with this problem. Test case prioritization techniques order the execution of test cases in an organized manner so as to achieve an objective function with greater efficiency. Many prioritization techniques have been proposed in the literature. This paper attempts to review various multi-objective based test case prioritization techniques.

Key words: Regression testing, Test cases, Test case prioritization, APFD.

I. INTRODUCTION

Software maintenance is an important phase of software development life cycle. It is a broad activity that includes error corrections, improvement of capabilities, deletion of obsolete capabilities, and optimization [9]. Software maintenance is becoming very expensive with the changing requirements and trends. After changes have been made, regression tests are applied to the modified parts of the software. This assures that a modified part has not affected the quality of the other parts of software [2]. Regression testing consumes cost and time for executing the test cases for the modified system codes. Many approaches have been proposed multi-criteria to reduce cost and time during regression testing, including test suite reduction techniques and test case prioritization techniques. The most popular technique to perform regression testing that is test case prioritization [11].

A. Regression testing

Regression Testing [7] is the process of executing the set of test cases which have passed on the previous build or release of the application under test in order to validate that the original features and functions are still working as they were previously. Figure 1 describes the types of regression testing technique. These are Retest all, Regression test selection, Test case minimization and Test case prioritization.

Retest all technique means re-execution of all available test suites, which are critical and moreover cost effective [7]. Regression test selection technique is used to select the test cases for testing based on the information about modified program [6]. Test case minimization technique helps in removing duplicate test cases in test suite by using the information about the program under test. The advantage of this technique is reduced cost of validating and managing test suites and disadvantage is that it reduces the fault detection capability with reduction of test suite size [4]. Retest all, Regression test selection and Test case minimization techniques are time consuming and expensive. To overcome this problem the developers incorporate prioritization of test cases for regression testing.

B. Test case prioritization

The test case prioritization problem [1] was first introduced by Wong et al. in 1997 as a flexible method of software regression testing. It was defined as finding a permutation of test cases to maximize an award function, which reflects the prioritization goal [1]. In this technique, each test case is assigned a priority. According to some criterion the priority has been assigned and test cases with highest priority are scheduled first. It adds an advantage to previous regression testing techniques that it doesn’t remove the test cases from test suite.

Several techniques have been proposed for prioritizing the existing test cases to accelerate the rate of fault detection in regression testing [4]. Techniques have also been proposed for Multi-Objective test case prioritization. This paper studies the different multi-objective based test case prioritization techniques.

II. RELATED STUDY

In the frequent paragraphs, some of the studies have been discussed, which had been previously undertaken in the field of Test Case Prioritization for regression testing.

Yoo et al. [13] and Mohanty et al. [12] described the existing techniques of test case prioritization, which mostly used code and functional coverage criteria to prioritize test cases with respect to the system code executed by test cases.

Elbaum et al. [3] proposed test case prioritization technique which used two major attributes: test cost and fault severity. The cost of a test case might be measured in terms of test execution, setup and test validation. The approach measured the severity of a fault in any one of the two ways i.e. the time required to locate the fault and correct the fault & the impact of failures that are caused by the faults. The effectiveness of proposed technique is less as compared to genetic algorithm based prioritization techniques.

A value-driven approach to system-level test case prioritization was presented by HemaSrikanth et al. [8]. It is also called Prioritization of Requirements for Test (PORT).
The PORT algorithm used the four factors for prioritization of test cases namely requirements, complexity of implementation and fault proneness of the requirements and customer priority. The approach used factor value and factor weight to calculate a Prioritization Factor Value (PFV) for every requirement. It improved the rate of fault detection. Test case prioritization using multi-criteria fitness function was developed by AmrAbdelFatah Ahmed et al. [14]. The approach used multiple control flow coverage metrics for prioritization. The calculation of metrics is based on conditions and statements that covered the test case. The technique has used the concept of genetic algorithm to obtain best optimal order of test cases. The disadvantage of proposed method is that it has not calculated the cost and execution time of test cases, which affects the effectiveness of regression testing.

N. Prakash and T.R. Rangaswamy [15] described modular based multiple test case prioritizations. In the proposed method, program was divided into multiple modules & the numbers of test cases are prioritized according to each module. In the second stage, the individual modular based prioritized test suites have been combined together and further prioritized for the whole program. Each module is made by number of test cases and fault coverage of test cases. The effectiveness of proposed method is high as compared to greedy algorithm and additional greedy algorithm, but less than genetic algorithm. The method considers only the fault coverage criteria prioritization and ignores all other criteria like code coverage, cost, and time etc.

Mahfuzul Islam et al. [16] identified multi-objective techniques to prioritize test cases based on latent semantic indexing. Latent semantic indexing means Information Retrieval (IR) approach [10]. The proposed technique counted the coverage of source code & application requirements and the cost of execute test cases. An IR-based traceability recovery approach has been applied to link software artifacts (i.e., requirements specifications) with source code. The order of test case was determined by using multi-objective optimization, implemented in terms of NSGA-II (Non-dominated Sorting Genetic Algorithm II) [5]. The technique has been evaluated in a preliminary case study on two small Java applications.

Sudhir Kumar Mohapatra and Srinivas Prasad [17] worked on an evolutionary approach search for test case prioritization. The proposed algorithm is based on testing time and potential code coverage information to schedule a test suite using genetic algorithm. The prioritization technique explained that test case generated by genetic algorithm takes less time as compared to test case generated by random test and optimal prioritization. The algorithm during time constraint execution has been shown to have detected maximum number of faults detection. The fitness function of proposed algorithm was more complex.

XiaoLin Wang and HongWeiZeng [18] proposed dynamic test case prioritization method based on multi-objective. The proposed method had calculated the prioritization values of test case separately based on five criteria namely coverage, fault, requirement, history information & time and calculated the weighted sum of their optimization results, then was test suite were sorted according to the prioritization value that calculated by weighted sum. It is tested on a small experiment instance. The proposed method is more complex and consumes more time during prioritization of test cases.

Test case prioritization using multi objective PSO (Particle Swarm Optimizer) was proposed by ManikaTyagi and SonaMalhotra [19]. The proposed algorithm has used the maximum fault coverage and minimum execution time for prioritization of test cases. The algorithm used 3-phase approach to solve the test case prioritization problem. First phase consists of redundant testcase removal. Second phase consists of using Multi Objective PSO for selecting test cases from test suite based on two objective functions: fault coverage and execution time. In the third phase, test case prioritization is performed & priority is assigned to the test cases which are obtained from phase 2. Multi objective PSO approach is better than other approaches like no ordering, reverse ordering and random ordering.

Multi-objective test prioritization via a genetic algorithm was described by MitrabindaRay et al. [20]. The proposed method first prioritizes components in accordance with their impact on the reliability of the system under test and then a test case selection approach to select a fixed number of test cases out of a pool of test cases. The test case selection approach is a multi-objective optimization problem and it is used to minimize the Deviation and maximize the Fitness of test suite for selection of test cases. The effectiveness of approach has been calculated by using simulation experiments.

The above discussed multi-objective test case prioritization techniques are tabulated below:

<table>
<thead>
<tr>
<th>Author</th>
<th>Description</th>
<th>Objective</th>
<th>Metrics</th>
<th>Approach Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>S. Elbaum, A. Malishevsk y and G. Rothermel [3]</td>
<td>It is used the minimum cost and maximum fault severity attributes for prioritization of test cases in test suite.</td>
<td>Statement coverage, Test cost and Fault severity.</td>
<td>APFDe(Average Percentag e of Faults Detected per Cost)</td>
<td>Greed y approach</td>
</tr>
<tr>
<td>HemaSrika nth, Laurie Williams and Jason Osborne[8]</td>
<td>It is system level test case prioritization. It improved rate of fault detection at the system level.</td>
<td>Requireme nt, Complexit y of impleme ntation, fault proneness of requireme nt and Customer priority</td>
<td>TSFD(Total Severity Fault Detected) and ASFD(Average Severity Fault Detected)</td>
<td>Value - driven approach</td>
</tr>
<tr>
<td>Authors</td>
<td>Case Prioritization Method</td>
<td>Metric Used</td>
<td>Authors</td>
<td>Prioritization Method</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Amr Abdel Fahmy Ahmed, Dr. Mohamed Shaheen and Dr. Essam K. Osba [14]</td>
<td>It is used multiple control flow coverage metrics for prioritization. These metrics are weighted by number of faults detection and their severity.</td>
<td>Condition coverage, Statement coverage and Fault severity.</td>
<td>Xiaolin Wang and Hongwei Zheng [18]</td>
<td>It integrated several traditional single-objective technologies.</td>
</tr>
<tr>
<td>N. Prakash and T.R. Rangaswamy [15]</td>
<td>The program was divided into multiple modules. Each module was prioritized by maximum fault coverage.</td>
<td>Fault coverage.</td>
<td>Manika Tya and Sona Malhotra [19]</td>
<td>The proposed algorithm was prioritization of test cases based on maximum fault coverage and minimum execution time</td>
</tr>
<tr>
<td>M.M. Islam, A. Marchetto, A. Susi and G. Scanniello [16]</td>
<td>It is multi-objective technique and deters mines test case orderings to maximize the number of faults detection.</td>
<td>Code coverage, Requirement coverage and Execution cost.</td>
<td>Mitrabinda Ray and Durga Prasad Mohapatra [20]</td>
<td>The algorithm was prioritized the components for testing according to reliability of the system under test cases.</td>
</tr>
<tr>
<td>Sudhir Kumar Mohapatra and Srinivas Prasad [17]</td>
<td>This algorithm used multi fitness function for prioritization of test case.</td>
<td>Code coverage and Execution time.</td>
<td>🌳</td>
<td>APFD</td>
</tr>
</tbody>
</table>

### III. THE METRIC FOR QUANTIFIES PRIORITIZATION

Many of the metrics are used to measure the prioritization of test cases. APFD (Average Percentage of Fault Detection) [4] metric is one of important metric to measure the test cases prioritization based on the fault detection, because the main purpose of the prioritization is to early identify the fault.

\[
\text{APFD} = 1 - \frac{\sum_{i=1}^{n} TF_i}{nm} + \frac{1}{2n}
\]

Where \( m \) represents number of faults, \( n \) represents number of test cases and \( TF_i \) is the position of the first test in \( T \) that exposes the fault \( i \). The APFD metric provides better results than the other metrics.

### IV. CONCLUSION

This paper provides a detailed review on multi-objective based test case prioritization for regression testing. It has
been found that best optimal order of test cases could be obtained by using concept of genetic algorithm. It has been concluded further that most of the researchers are working on code coverage, requirement coverage and fault detection for test case prioritization and little or no work has been done on the concept of complexity for prioritize test cases. Hence complexity criteria should be explored in detail. This study provides a direction to researchers and practitioners whereby they can extend multi objective test case prioritization technique via using genetic algorithm.

REFERENCES


