A Novel Approach to Multiple Criteria Based Test Case Prioritization

Robeala Abid[1], Aamer Nadeem[2]
[1,2]Department of Computer Science
Capital University of Science and Technology
Islamabad, Pakistan

robealamalik@yahoo.com, anadeem@cust.edu.pk

***Abstract*— When software is modified, it is retested to ensure that no new faults have been introduced in the previously tested code and it still works correctly. Such testing is known as regression testing.The cost of regression testing is high because the original program has large number of test cases. It is not feasible to execute all test cases for regression testing. Test suite minimization, test case selection and test case prioritization are cost commonly used techniques in regression testing to reduce the cost of regression testing. While test suite minimization and test case selection techniques select a subset of test cases, test case prioritization does not eliminate any test case, it only orders the test cases with the objective of increasing the fault detection rate. Prioritization is usually preferred over other two approaches because it does not involve the risk of losing useful test cases. Prioritization techniques assign priority to each test case on the basis of some coverage criteria. A number of different single criterion and multiple criteria based prioritization techniques have been proposed in the literature. Multiple criteria based prioritization techniques perform better than single criterion based prioritization techniques. The existing multiple criteria based prioritization techniques combinethe criteria in such a way that "Additional" strategy cannot be applied on them. In this paper, we propose a new multiple criteria based test case prioritization algorithm that considers two criteria to prioritize test cases using "Additional" strategy. One criterion is considered as primary and other is considered as secondary. Primary criterion is used to prioritize the test cases whereas secondary criterion is used to break the tie among test cases when two or more test cases provide equal coverage of entities of first criterion. Our proposed multiple criteria based prioritization algorithm performs better than the existing prioritization techniques.**

I. INTRODUCTION

Software testing is performed for evaluating quality of software and making improvements in it. The main aim of testing is to identify errors in the software product and to fix them. It additionally guarantees that whether customer‘s requirements are being refined by the software [1]. Testing of software is a critical and expensive phase of software development [2]. Faults can occur at any stage of development process and must be identified and removed to control their further transmission in the next steps of development phases [3].There are different types of software testing [4], which include white box testing, black box testing [5], grey box testing [6], and model based testing [7]. When software is revised, it is retested to ensure that no new faults have occurred in the earlier tested code and it still works correctly. Such testing is known as regression testing [7].

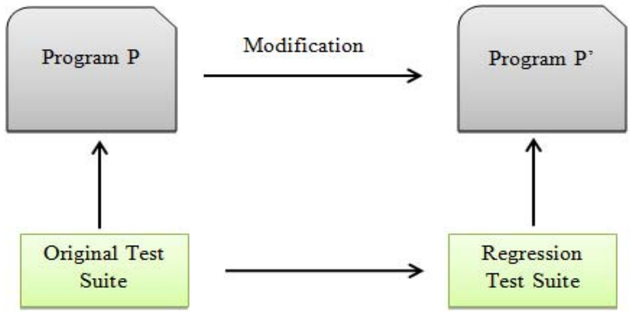

Fig.1. Regression Testing basic Concept

Regression testing is vital for real-world software development projects. It is used to test the new changes to computer programs. Regression testing ensures the validity of a software program upon making some changes in the program [8]. As the software is upgraded or enhanced, the number of test cases increases exponentially. This makes regression testing a complicated, inefficient and costly process. Different techniques have been introduced to reduce the cost of regression testing which are defined as follows.

i. **Test case Selection:** Test case selection selects the correct subset of test cases which are valid for required parts of the software and which traverse the required parts of the software. Remaining test cases are eliminated in this technique [9].

ii. **Test suite Minimization:** Minimization is another technique of regression testing in which test suite is minimized by the removing the redundant test cases. Test cases checking same functionality are removed by this technique. The selected test cases are placed in a set which is called as minimal hitting set [10].

iii. **Test case Prioritization:** Test case prioritization assigns priority to test cases by considering some coverage criteria. This technique is effective than the other two techniques of cost reduction because it does not eliminate any test case from test suite. It assigns priority to test cases in such a way that the test case with highest priority is executed first [8].

Prioritization technique assigns priority to test cases by considering some criteria. Test case with maximum coverage of that criterion is assigned highest priority [11]. These criterions can be the structural entities of source code like statements, branches, function etc. [12] or can be specification based like interaction between graphical events or customer priority etc. [14]. Using different criteria, priority is assigned on the basis of two general strategies named as “Total” and “Additional” proposed by [13]. “Total” strategy considers the total number of entities covered by test cases. This can result in repetition of entities if two or more test cases have same coverage data. Also, if testing is halted prematurely, then some entities can be missed. “Additional” strategy is better than “Total” strategy as it not only considers the number of entities covered by each test case but also the additional entities which remains uncovered by previously selected test case. This strategy reduces the repetition of entities and it will also ensure the coverage of each entity by atleast one test case [13].

Prioritization can be single criterion based or multiple criteria based. In single criterion based prioritization, only one criterion is used to prioritize the test cases whereas; in multiple criteria based prioritization, more than one criterion is used to prioritize the test cases. Test cases are prioritized by combining multiple criteria in different ways. The majority of the existing prioritization techniques are based on single criterion.In single criterion based prioritization techniques; test case is selected randomly if there is an occurrence of a tie in more than one test case. This reduces code coverage to much extent, and the early fault detection rate is also reduced. The existing multiple criteria based prioritization techniques are not appropriate because these techniques are using “Total” strategy to prioritize the test cases. These existing techniques are unable to use “Additional” strategy because these techniques combine the coverage information of all criteria.

In this paper, we propose a new multiple criteria based prioritization algorithm which considers two coverage criteria to prioritize the test cases. The rest of this paper is organized as; Section 2 describes different prioritization techniques. Section 3 includes the proposed algorithm. Section 4 includes an example to explain the working of proposed algorithm and Section 5 concludes the paper.

II. Related Work

Test case prioritization finds an ordering of test cases that provides the maximum benefits to the software testers. The software testers assign priority to each test case in test suite to ensure that the test cases with higher priorities are run earlier in the testing process [8].
Test case prioritization techniques are divided into two categories; Single Criteria based prioritization techniques and Multiple Criteria based prioritization techniques.

1. Single Criterion based prioritization Techniques:

Different white box prioritization techniques use single criterion to prioritize the test cases. Priority is assigned to each test case on the basis of some criterion. Following are different single criterion based white box prioritization techniques.

White Box Prioritization techniques:

In white box testing, detailed information of internal logic of software product is required. Tester needs complete knowledge of source code for testing [5]. In white box prioritization techniques, the test cases are prioritized on the basis of code covered. These techniques are given as below.

Coverage based prioritization strategies:

Two main prioritization strategies were introduced by Rothermel et al. [13] and Elbaum et al. [7]: “Total” and “Additional”. “Total” strategy considers the total number of entities covered by test cases. “Additional” strategy chooses the test case with high coverage degree and covering entities that are not so far covered by previously chosen test case. Followingarethe prioritization techniquesbasedon“Total”and“Additional” strategies.

i) Function coverage prioritization: The criterion used in this technique is number offunctions. This technique can be based on Total or Additional strategy. The criterion used in Total function coverage prioritization is total number of functions covered. The criterion used in Additional function coverage prioritization is based on number of uncovered functions.

ii) Statement coverage prioritization: The criterion used in this technique is number of statements.The criterion used in Total statement coverage prioritization is total number of statements covered. The criterion used in Additional statement coverage prioritization is based on number of uncovered statements.

iii) Branch coverage prioritization: The criterion used in this technique is number of branches.The criterion used in Total branch coverage prioritization is total number of branches covered. The criterion used in Additional branch coverage prioritization is based on number of uncovered branches.

vii) Total FEP based prioritization: In this technique, test cases are assigned prioritizes on the basis of their fault exposing potential. Mutation testing is applied to create mutants of a program and mutant score of a test case for each statement is calculated. Finally, summation is obtained by adding the mutation score of all statements for a specific test case and the test case with maximum mutation score is given highest priority.

viii) Additional FEP based prioritization: Additional FEP based prioritization is same as total FEP with the addition of a new factor called confidence. Confidence C(s) of a statement is the probability that statement s is correct. Value of C(s) is between 0 and 1, inclusive. If we execute a test case on a statement, and the test case does not expose any fault in that statement, then the value of confidence C(s) increases and the new value of confidence will be C'(s) and the additional confidence gained is Caddi(s). Caddi(t) is defined as the additional confidence gained by executing test case t on program P.
Highest priority is assigned to that test case which has maximum Caddi(t) value. After selection of a test case, the value of confidence C(s) for all statements covered by test case t is updated. Also, the Caddi(s) values are updated for remaining statements against remaining test cases. This process continues until all test cases are prioritized.

History based prioritization

Kim and Porter [15] proposed a technique, which is stated as “History-based prioritization”. Historical information is used to select set of test cases for new version of program. Firstly, RTS techniques are used to produce T’ for the test suite T. Afterwards, every test case of T’ is given a selection probability. Previous performance is considered to assign selection probability to each test case. These techniques uses test histories data which is based on execution history, number of entities covered and fault detection rate. Finally, a test case is selected on the basis of probabilities assigned in previous step. This step is repeated until all test cases are ordered.

Additional Spanning Entities Coverage based Prioritization

Marre´ and Bertolino [16] presented an idea to use spanning set of entities as a criterion for prioritization of test cases. Spanning set of entities is obtained from the subsumption relationship between entities. Subsumption relationship between two entities is defined as, given two entities E and E‘, if every complete path which covers E is also covering E‘, it means that E subsumes E‘, but their exists some entities which are not covered by any other entities. Spanning entities are those entities which subsumes other entities without being itself subsumed by other entities. This technique was further divided into Additional Spanning Statements and Additional Spanning Branches. Test cases are ordered on the basis of maximum number of uncovered spanning statements or spanning branches.

2. Multiple Criteria based prioritization techniques:

Regression testing can be improved by using more than one criterion for prioritization of testing. Following are different multiple criteria based prioritization techniques from the literature.

Ahmed [17] proposed a multi criteria based prioritization technique which uses Genetic Algorithm. It uses multi-criteria fitness function for ordering of the test cases. This technique uses three criteria; statement coverage, fault severity and control-flow coverage. Number of statements, conditions and multiple conditions covered by each test case is measured by the control flow coverage metrics. The weight of each metrics is assigned by number of faults revealed and their severity. This technique shows better result than the previous approaches.

Prakash and Gomathi [18] carried out an empirical study to prioritize test cases with multiple criteria. They have used three standard applications (Hospital Management system, Library Management system, Student Registration system) for proposed test case prioritization method and make a comparison with present prioritization techniques. The results of this study demonstrate that performance of proposed method is superior to the previous method.

Kaur et al. [19] carried out an empirical study to prioritize test cases by means of more than one criterion. They have used the bank application for their proposed method. The results of this study show that proposed method is more effective than the existing method. The main goal of this article is to make comparison of the effectiveness of un-prioritized and ordered test cases using APFD, APSD, APBD and APPD.

Khasragi [20] gives an overview of regression testing and prioritization technique. A survey is conducted on different multi-objective test case Prioritization techniques. Through an example, working of single criterion and multiple criteria is discussed. Moreover, single criterion and multiple criteria based prioritization techniques are compared. The decision of this paper shows that multiple criteria based prioritization techniques are better to increase the rate of fault detection than single criterion based techniques.

III. Proposed Algorithm

Our algorithm takes two criteria as input. One is considered as primary and other is considered as secondary. Test cases will be prioritized according to primary criterion. Secondary criterion is applied in case of tie in more than one test case. The inputs of algorithm are:test suite T for program P which consists of test case. EntitiesCov1 is the set of entities of first criteria having "m" number of entities. EntitiesCov2 is the set of entities of second criteria having "n" number of entities. Cov1 (t) is the set of entities of first criterion covered by each test case. Cov2 (t) is the set of entities of second criterion covered by each test case. The output of algorithm is PrT, which is the priority list of test cases.

Procedure for prioritizing regression test cases

Input:

1. *T: Test Suite for P'*
2. *entitiesCov1: Set of entities of criterion1 in P covered by tests in T.*
3. *entitiesCov2: Set of entities of criterion2 in P covered by tests in T.*
4. *cov1 (t): Set of entities of criterion1 covered by executing P against t.*
5. *cov2 (t): Set of entities of criterion2 covered by executing P against t.*

Output:

PrT: A sequence of test cases

Declarartion:

X': Set of Test cases, Y: Set of test cases

Procedure:*PrTest*

```
Step 1:   X'=T. Find t in X' such that
          If Cov1(t) > Cov1(u) for all u in X', u ≠ t
                    Goto step 2
          ElseIf Cov1(t) =Cov1(u) for some u in X', u ≠ t
                    Set Y= <t, u> find t in Y such that
                    Cov2(t) ≥ Cov2(u) for all u in Y, u ≠ t.
                    Set PrT=< t >, X'= X' \ {t},
                    entitiesCov1= entitiesCov1 \ cov1(t),
                    entitiesCov2= entitiesCov2 \ cov2(t)
                    Y= Ø.
Step 3:   Repeat while X' ≠ Ø.
Step 3.1: Compute residual coverage for each test t in X';
          resCov1(t)= entitiesCov1 \ [cov1(t) ∩ entitiesCov1]
Step 3.2: If resCov1(t) > resCov1(u) for all u in X', u ≠ t
          Goto step 3.3
          Else
          If resCov1(t) =resCov1(u) for some u in X', u ≠ t
                    Set Y= <t, u> find t in Y such that
                    resCov2(t) ≥ resCov2(u)for all u in Y, u ≠ t.
Step 3.3: Set PrT= append (PrT , t),  X'=X' \ {t}
          entitiesCov2= entitiesCov2 \ cov2(t)
          entitiesCov1=entitiesCov1 \ cov1(t).
Step 4:   Append to PrT any remaining tests in X' in random
          order.
```

IV. Example of Proposed Algorithm

The following example is used to define the concepts of our proposed multiple criteria based prioritization algorithm. Table 1 shows the mapping between test cases and coverage data.

Table.1. Example Data

Test case(t)	Entities of Criterion 1	Cov1(t)	Entities of Criterion 2	Cov2(t)
t1	1,2,3,4,5,6	6	1,2	2
t2	4,6,8,9	4	2,4	2
t3	1,4,8,10	4	4	1
t4	1,2,3,4,8,9	6	2,3,4	2
t5	2,4,7,10	4	2,3	2

Single criteria based prioritization algorithm can be explained using test cases mentioned in above table. Assume that we want to prioritize the test cases using coverage of criterion 1. The entities covered by each test case are identified. Test case covering maximum entities of coverage is given highest priority. The next step is to choose the test case with high coverage degree and covering entities that are not so far covered by previously choosed test case. When multiple test cases have same coverage entities, one test case is selected randomly for that entity. Considering this definition, Test case t1 and t4 have same coverage value. So tie occurs between these test cases. So, we will select randomly (t1 in this case). This test case is placed in the prioritized list. The prioritiy list becomes PrT= {t1} Next step is to calculate the residual coverage of all remaining test cases while test suite T does not get empty. Test case t2, t3, t4 and test case t5 have same value of residual coverage, so tie occurs between these test cases. In this step, test case will be selected randomly (t5 in this case). The priority list becomes PrT= {t1, t5}. This step is repeated until all test cases will be prioritized and all the entities of criterion will be covered by atleast one test case. The final priority list will be PrT = {t1, t5, t2, t3, t4}.

The same example will now be explained for our proposed multiple criteria based prioritization algorithm. Assume that the first criterion is primary and second is secondary. Primary criterion is used to prioritize the test cases and secondary criterion is used to break the tie between two or more test cases. Test case t1 and t4 have same coverage value. So tie occurs between these test cases. Test case t4 will be selected according to coverage of criterion 2 as it covers maximum entities of criterion 2. The selected test case will be added to priority list and remaining test suite will be prioritized. EntitiesCov1 and EntitiesCov2 will also be updated to show the remaining uncovered entities. The prioritiy list becomes PrT= {t4}. Next step is to calculate the residual coverage of all remaining test cases while test suite T does not get empty. Test case t1 and test case t5 have same value of residual coverage, so tie occurs between these two test cases. This tie will be broken by calculating the residual coverage of t1 and t5 according to second criterion. Test case t1 will be selected as it has minimum value of residual coverage. This test case will be added to priority list and test suite T will be updated. EntitiesCov1 and EntitiesCov2 will also be updated to show the remaining uncovered entities. The priority list becomes PrT= {t4, t1}. This step is repeated until all test cases will be prioritized and all the entities of both criterions will be covered by atleast one test case. The final priority list will be PrT = {t4, t1, t5, t2, t3}.

From the above two scenarios, it can be seen that multiple criteria based prioritization approach has greater coverage than single criterion based prioritization approach. In case of a tie between two test cases, single criterion based approaches arbitrarily selects a test case to break the tie and therefore, the test case covering more entities may be removed loosing desired code coverage which decreases the fault detection rate to much extent. Multiple criteria based approach has high rate of fault detection because it achieves desirable code coverage by using multiple criteria.

V. Conclusion

Regression testing is important because in maintenance phase, software continuously undergoes changes and requirements are also changed which can result in addition of some new faults. Different cost reduction techniques are used to reduce the cost of regression testing. The most effective technique is test case prioritization as it does not eliminate any test case. Different criteria are used to assign priority to each test case on the basis of maximum coverage of those criteria.Test case prioritization uses "Total" and "Additional" as two different strategies to prioritize the test cases.Based on these strategies, prioritization techniques can be single criterion or multiple criteria based. Multiple criteria based prioritization techniques are better than single criterion based prioritization techniques because these techniques are using more than one criterion to prioritize the test cases and hence increasing the rate of fault detection.In existing single criterion based prioritization techniques, test case selection is random in case of occurrence of a tie between two or more test cases when two or more test cases provide equal coverage of entities of first criterion. This reduces the coverage of criterion and the fault detection rate is also decreased.Existing multiple criteria based prioritization techniques are not suitable because of some issues. These techniques use "Total" strategy to prioritize the test cases. "Total" strategy considers the total number of entities covered by each test case, rather than considering the entities which remains uncovered in the previous selection. Our proposed multiple criteria based prioritization algorithm will perform better than existing single criterion and multiple criteria based test case prioritization techniques because it prioritizes the test cases using "Additional" strategy and also it reduces the randomness of test cases to much extent.

VI. References

[1] R. Kaur Chauhan and I. Singh, "Latest Research and Development on Software Testing Techniques and Tools", International Journal of Current Engineering and Technology, vol. 4, no. 4, ISSN 2347 - 5161, 2014.

[2] D. Jeffrey and N. Gupta, "Experiments with test case prioritization using relevant slices", Journal of Systems and Software, vol. 81, no. 2, pp. 196-221, 2008.

[3] L. Baresi and M. Pezzè,"An introduction to software testing", Electronic Notes in Theoretical Computer Science, vol. 148, no. 1, pp. Pages 89-111, 2006.

[4] I. Hooda and R. Singh Chhillar, "Software Test Process, Testing Types and Techniques", International Journal of Computer Applications, vol. 111, no. 13, pp. 10-14, 2015.

[5] M. Ehmer and F. Khan, "A Comparative Study of White Box, Black Box and Grey Box Testing Techniques", International Journal of Advanced Computer Science and Applications, vol. 3, no. 6, 2012.

[6] R. Saxena and M. Singh, "Gray Box Testing: Proactive Methodology for the Future Design of Test Cases to Reduce Overall System Cost", Journal of Basic and Applied Engineering Research, vol. 1, 8, no. 2350-0255, 2014.

[8] S. Elbaum, A. Malishevsky and G. Rothermel, "Test case prioritization: a family of empirical studies", IEEE Transactions on Software Engineering, vol. 28, no. 2, pp. 159-182, 2002.

[8] S. Yoo and M. Harman, “Regression testing minimization, selection and prioritization: A survey”, in Software Testing, Verification, Rel., vol. 22, no. 2, pp. 67–120, Mar. 2012.

[9] G. Rothermel and M. Harrold, "A framework for evaluating regression test selection techniques", Proceedings of 16th International Conference on Software Engineering, 1994.

[10] G.M.R, “Computers and Intractability: A Guide to the Theory of NP-Completeness”,W.H.Freeman and company: New York, 44(2), pp.340, 1979.

[11] Y. Fazlalizadeh, A. Khalilian, M. Azgomi and S. Parsa, "Prioritizing test cases for resource constraint environments using historical test case performance data", 2009 2nd IEEE International Conference on Computer Science and Information Technology, 2009.

[12] G. Rothermel, R. Untch, Chengyun Chu and M. Harrold, "Test case prioritization: an empirical study", Proceedings IEEE International Conference on Software Maintenance - 1999 (ICSM'99). 'Software Maintenance for Business Change' (Cat. No.99CB36360), 1999.

[13] G. Rothermel, R. Untch, Chengyun Chu and M. Harrold, "Prioritizing test cases for regression testing", IEEE Transactions on Software Engineering, vol. 27, no. 10, pp. 929-948, 2001.

[14] C. Henard, M. Papadakis, M. Harman, Y. Jia and Y. Le Traon, "Comparing white-box and black-box test prioritization", Proceedings of the 38th International Conference on Software Engineering - ICSE '16, 2016.

[15] J. Kim and A. Porter, "A history-based test prioritization technique for regression testing in resource constrained environments", Proceedings of the 24th international conference on Software engineering - ICSE '02, 2002.

[16] M. Marre and A. Bertolino, "Using spanning sets for coverage testing", IEEE Transactions on Software Engineering, vol. 29, no. 11, pp. 974-984, 2003.

[17] A. Ahmed, M. Shaheen and E. Kosba, "Software testing suite prioritization using multi-criteria fitness function", 2012 22nd International Conference on Computer Theory and Applications (ICCTA), 2012.

[18] N. Prakash and K. Gomathi, "Improving Test Efficiency through Multiple Criteria Coverage Based Test Case Prioritization", International Journal of Scientific & Engineering Research, vol. 5, 4, no. 2229-5518, 2014.

[19] N. Kaur, M. Mahajan, “Regression testing with multiple criteria based test case prioritization”, vol 2(Isuue V), pp.ISSN 2321-9653, 2014.

[20] AB. Khasragi, “Comparison between Multi objective Test Case Prioritization Techniques”, 2th International conference in new research on Electrical Engineering & Computer Science, 2016.