International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 5 | ISSN : 2456-3307

Enhancing the Performance of Coverage-Based Techniques in Test Case Prioritization

G. Bhavyasri[1], Prof. A. AnandaRao[2], Dr. P. Radhika Raju[3]
[*1]M.Tech Scholar, Department of CSE, JNTUA College of Engineering, Ananthapuramu, Andhra Pradesh, India
[2]Professor, Department of CSE, JNTUA College of Engineering, Ananthapuramu, Andhra Pradesh, India
[3]Ad-hoc Assistant Professor, Department of CSE, JNTUA College of Engineering, Ananthapuramu, Andhra Pradesh, India

ABSTRACT

A presented approach is based on coverage techniques using closed dependency structure. Test cases are grouped in clusters from functional dependency. Closed functional dependency structure is applied to arrange test case in each cluster, Test case prioritization is done from function-coverage information. Results demonstrate that, proposed method performs better when compared to code coverage technique and some other coverage techniques in test case prioritization. Moreover, current approach capitulate 97% of average percentage of fault detection (APFD) and coefficient of determination in correlation as 0.8363, which enhances the performance of coverage-based techniques.

Keywords: Regression Testing, Dependency Structures, Clustered Technique, Test Case Prioritization, Function Coverage Techniques.

I. INTRODUCTION

Test case prioritization is to diminish the cost of regression testing by increasing the fault rate. Test cases are arranged in an order and executed based on priority. To keep the tests in an order, many techniques adopt code coverage information through the process and software execution under test. So, these are mentioned as coverage-based techniques [1, 2, 3, 4]. Coverage-based techniques are not correlated positively with fault rate detection in some programs like Ant, which is an open source program tool from Apache Software Foundation [1, 5, 6].

Many suboptimal solutions such as lexicographical ordering, clustering, dependency structures were implemented [1, 7, 8, 9, 10, 11] in test case prioritization by using code coverage. These three approaches were used to increase the fault rate in test prioritization and to reduce faults at earlier stage in regression testing. The main drawback here is given approaches cannot show the high correlation value in between coverage and fault.

This paper presents the combination of cluster and closed-dependency structure methods based on function coverage techniques to prioritize test cases [8, 9, 10]. Test case prioritization has done from the values of two coverage information factors Total Function Coverage (TFC) and Additional Function Coverage (AFC) [2, 3, 4] from which coverage and fault rate can be obtained and helps in correlation calculation.

The paper is organized as follows, overviews about the related work is presented in section II. Section III introduces the cluster and closed-dependency structures in coverage-based techniques. Section IV shows evaluated results that present the correlation value between coverage and fault rate in coverage-based techniques. At last, section V concludes the work and suggests some possible directions for future work.

II. RELATED WORK

Many approaches are availed to increase the fault rate detection in test case prioritization for regression testing. Sepehr Eghbali and Ladan Tahvildari compared different java programs in Ant, Galileo, Jtopas, Jmeter, Nano and XML among test cases and prioritized them by a method called lexicographical ordering. Also measured the average of fault rate detection by Average Percentage Fault Detection (APFD), it measures how fast the generated order detects the faults [1].

Elbaum stated three groups of test case prioritization techniques that are control, statement, function level techniques to prioritize and increase fault rate detection [2].

Akbar Siami Namin and James H. Andrews experiments stated that, when size is restricted, coverage and effectiveness are correlated with each other. In addition, by making use of size and coverage gives more accurate prediction of effectiveness than size alone [5]. Laura Inozemtseva and Reid Holmes presented, when the number of test case in a suite is restricted, there will be a moderate in between coverage and effectiveness with low correlation [6]. Other author's moreover says it, imply that coverage does not correlate positively with the fault rate detection all the time. Carlos proposes a technique, which incorporates a stratified clustering manner and used the coverage in the aspects of code, complexity and fault history as the inputs in clustering [8].

Miller, T. presented test case prioritization among test cases using dependency structures whichis intently related to relationships among system components [10]. So, by giving a priority to dependent test cases improve early fault finding in testing.

However, in the proposed, by using clustering method and closed-dependency, test cases grouped and ordered based on the function-coverage information of each test case. This is applied to improve fault rate detection that leads to enhance the performance of coverage-based techniques.

III. PROPOSED WORK

Initially, test cases are grouped in the form of clusters by using the condition of functional dependencies among test cases in test suites. Functional dependencies are defined as connections and associations between system performance and formative of their sequence. It is also defined as, a function cannot obtain until another function occurs. When an interaction of a test case tc1 needs to execute before tc2, it is to be concluded that there is a dependency of tc2 on tc1. Take the example of placing an item order on an E-Commerce site, in this scenario without registration and login functionality; the user cannot place the order for the desired product. Like this, there are so many test cases developed to test these kind of scenarios. Same will be consigned to test cases through inheritance.

There are two types of dependency structures, they are open and closed dependency. Open dependency structure determines the condition as the test case tc1 and test case tc2 specifies that tc1 need to be executed tc2 at some point and closed dependency structure determines that tc1 must be executed tc2 immediately before. In this work, test cases are based on closed dependency using clustered method.

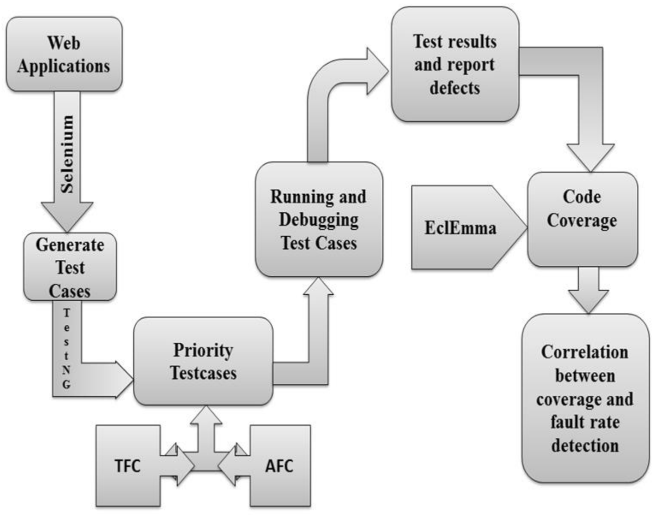
Web Applications
Selenium
Generate Test Cases
TestNG
Priority Testcases
TFC
AFC
Running and Debugging Test Cases
Test results and report defects
EclEmma
Code Coverage
Correlation between coverage and fault rate detection

Figure 1: Framework

Figure.1 describes the entire process of proposed work. Initially, the web application has to be tested by using Selenium IDE and generate the test cases. By using TestNG framework those test cases are clustered and placed using functional dependency and closed dependency structure then prioritized based on coverage information. Test results and defect reports are generated by executing the test cases in priority. To calculate the value of correlation in addition with fault rate there is a necessity of coverage value which is generated by using EclEmma tool.

A. *Test Case Prioritization*

To keep the test cases into an order and improve the fault rate detection is defined as Test case prioritization. It is consider to deciding the speed or efficiency of the application software or code. Test case prioritization techniques [1, 2, 3, 4] are relies on coverage information of test cases and are effective to develop the fault rate detection. The technique used to prioritize test cases is Function coverage technique.

B. Test Case Prioritization Based on Function Coverage Techniques

Clustering Approach enhances the overall performance of test case prioritization with the help of function coverage techniques. The coverage information obtained from two function coverage-based techniques TFC and AFC is gathered to all the test cases which are arranged in clustered and closed-dependency structure form, from this information, test cases are tested priorily based on the condition of "Functions that are covered less are given higher priority" [1].This condition says that, the priority will be given to the test case which covers less number of functions. The performance of coverage-based techniques depends on the high correlation value in between coverage and fault rate detection. However, this is not done in some cases and that lead to obtain undesirable performance. With this main idea of function coverage, test cases are prioritized and tested. If the functions that are covered less in a test case having same number, the priority will be given to the test case that having less fault rate. In this study, clustering approach used closed-dependency to arrange the test cases. In closed-dependency structure, the dependent test cases t1, t2 describes t1 ought to be completed without delay earlier than t2. Here clustering approach expose a high-value correlation among fault rate detection and coverage that concluded as enhancing the overall performance of coverage-based techniques [15]. Test cases are not prioritized at the initial state and tested with no priority. The closest test cases are in respect of functional dependency are formed as the first cluster than the pair-wise similarities re-computed with new clusters. After the advent of clusters, test cases are arranged by using closed-dependency structure. A priority of test cases will be generated in ascending order on a test suite depends on following steps.

Step-1: Arrange the test cases within each cluster using closed dependency.
Step-2: Consider test case prioritization technique, here in proposed, perform the testing on clustering test cases by following factors are used:

Total Function Coverage (TFC) [1]

Additional Function Coverage (AFC) [1]

C. Algorithm to Cluster the Test Cases Using Functional Dependencies

Input: Test Suite T={TC_1, TC_2,…,TC_n}
Algorithm#1:

```
Begin
        Set T' empty
        For each test case
        Cluster $C_i$ = $TC_i$ using Fn
        End for
        Sort $TC_i$ in $C_i$, as ascending order based on $CF_n$.
        T = $C_i$
        Let T^'=T
End
```

Output: Clustered Test Cases T'

D. Algorithm to Prioritize Clustered Test Cases Using Function Coverage Techniques

The set of rules depend totally on inputs as given from Test Suite T. When these are known, then the output will be generated in a prioritized order of test cases.
Input: T_{TFC}, T_{AFC} from Clustered Test Suite T'
Algorithm#2:

```
Begin
        Set T'' empty
        For each $C_i$
        Sort Ti in $C_i$, as ascending order based on TFC
        & AFC in the form of large test suite.
End for
        T = $C_i$
        Let T^''=T
End
```

Output: Prioritized Test Cases T''

IV. RESULTS

Clustered test cases depend on functional dependency and closed-dependency is shown in table1. Test cases are generated by automation using selenium tool and the framework used is TestNG to execute each test case. Clustered test cases are prioritized primarily based on the coverage information i.e., TFC and AFC and these will be calculated as:

Total Function Coverage = Total variety of functions executed.
Additional Function Coverage = Total quantity of

additional functions covered.
Cluster Defects=Total number of defects in cluster.

Table 1: Clustered Test Cases

Clusters (C_i)	C_1	C_2	C_3	C_4	C_5
Test Cases (TC_i)	TC_1 TC_2 TC_3 TC_4	TC_5 TC_6	TC_7 TC_8	TC_9 TC_{10} TC_{11} TC_{12} TC_{13} TC_{14}	TC_{15} TC_{16} TC_{17} TC_{18} TC_{19} TC_{20} TC_{21} TC_{22}

The test cases prioritized based on the function-coverage technique. In the Table 2, TFC and AFC are calculated for all the test cases that are already arranged in clustered and closed-function dependency structure. Using this information, priority assigned on the condition of "Functions that are covered less are given higher priority". With this main idea of function coverage, test cases are prioritized and tested. If the functions that are covered less in a test case having same number, the priority will be given to the test case that having less fault rate. By covering additional function test cases in regression testing, high pass rate is achieved when compared with previous coverage techniques. Following are clustered test cases that provide function coverage information.

Table 2: Resulted values for clustered test cases with function coverage techniques TFC, AFC.

Test Cases (TC_i)	T_{TFC}	T_{AFC}	Clustering (Ci) Defects
Test $Case_1$	9	-	-
Test $Case_2$	15	-	-
Test $Case_3$	12	1(f_2)	C_1
Test $Case_4$	16	1(f_{12})	C_1
Test $Case_5$	17	1(f_{14})	C_2
Test $Case_6$	18	-	-
Test $Case_7$	8	-	-
Test $Case_8$	5	1(f_3)	C_3
Test $Case_9$	15	-	-
Test $Case_{10}$	7	1(f_7)	C_4
Test $Case_{11}$	9	-	-
Test $Case_{12}$	8	-	-
Test $Case_{13}$	6	-	-
Test $Case_{14}$	7	-	-
Test $Case_{15}$	6	-	-
Test $Case_{16}$	5	-	-
Test $Case_{17}$	7	1(f_6)	C_5
Test $Case_{18}$	5	1(f_5)	C_5
Test $Case_{19}$	4	-	-
Test $Case_{20}$	3	-	-
Test $Case_{21}$	4	1(f_1)	C_5
Test $Case_{22}$	6	1(f_2)	C_5

Table 3: Prioritized Test Cases

Clusters (C_i)	Prioritized Test Cases(T'')
C_1	TC_1,TC_3,TC_2,TC_4
C_2	TC_5,TC_6
C_3	TC_8,TC_7
C_4	TC_{14}, TC_{10}, TC_{13}, TC_{12}, TC_9, TC_{11}
C_5	TC_{20},TC_{18},TC_{22},TC_{19},TC_{17}, TC_{15},TC_{16},TC_{21}

Fault rate for clustered test cases is given in the Table 4.

Table 4: Fault Rate for Clustered Test Cases

Clusters (C_i)	Failed Test cases	Fault Rate
C_1	TC_3, TC_4	0.009
C_2	TC_5	0.018
C_3	TC_8	0.024
C_4	TC_9	0.020
C_5	TC_{18},TC_{22},TC_{17}, TC_{21}	0.029

A. Average Percentage Fault Detection (APFD):

$$APFD = 1 - \frac{TF_1 + \dots + TF_m}{nm} + \frac{1}{2n} \qquad (1)$$

APFD is a metric to measure the rate of fault detection with a particular ordering [1]. Fast fault detection indicated by the high value of APFD. Equation (1) is used to degree APFD and the terms to be considered are: n variety of test cases are represented with T, ith fault of a test case is referred as TF_i. Number of faults in a test suite is m.

$$APFD = 1 - \frac{2+1+1+1+4}{(22)(9)} + \frac{1}{44}$$

APFD = 0.9772

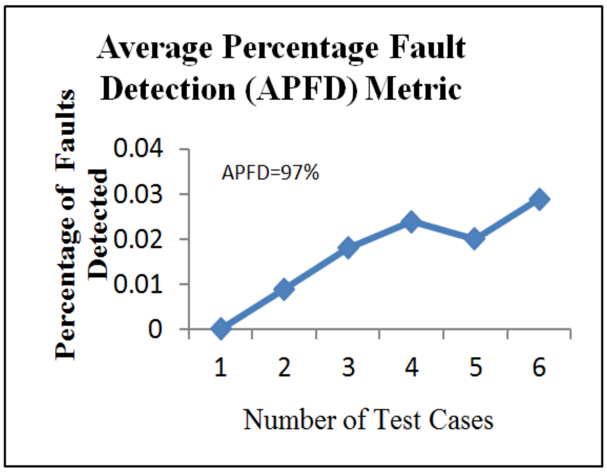

Figure 2 : APFD Graph for Clustered and Prioritized Test Cases

B. Correlation among Coverage value and Fault Rate

Coverage information is calculated for each test case in a test suite. The tool called EcLemma in automation of test suites does this. There is a possibility to show a positive linear correlation in between coverage and fault rate detection [17, 18]. After calculation of code coverage and fault rate from the executed clustered test cases, the correlation is to be calculated mathematically by pearson correlation method. The equation (2) shows the correlation between the inputs coverage Cv and fault rate f.

$$r = \frac{\sum_i (Cv_i - \overline{Cv})(f_i - \overline{f})}{\sqrt{\sum_i (Cv_i - \overline{Cv})^2}\sqrt{\sum_i (f_i - \overline{f})^2}} \qquad (2)$$

Table 5: Coverage Value for Different Coverage Types

Coverage Type	Coverage (%)
Branch	47.7
Statement	50.0
Complexity	65.7
Function	77.7

Table 6: Fault Rate in Clusters

Cluster Fault Rate				
C_1	C_2	C_3	C_4	C_5
0.009	0.018	0.024	0.020	0.029

Correlation in between Coverage and Fault rate is shown in figure 3.

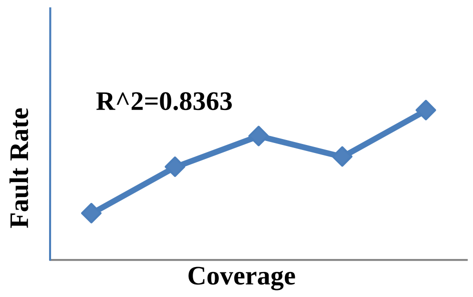

Figure 3: Correlation between Coverage and Fault Rate.

Figure.3 says that the coverage value correlates with fault rate as 0.9145 by the value of correlation, this is a strong positive correlation, which means that high coverage value goes with high fault rate value. The value of R^2, the coefficient of determination, is 0.8363. It shows the graph for four different types of coverage and the fault rate for clustered test cases based on functional dependency along with function level techniques. In this combination, high correlation value says that stronger coverage types provide some more information of the best in clusters.

V. CONCLUSIONS & FUTURE WORK

This work proposed such a way for test case prioritization by clusters and closed-dependency structure along with function level techniques. The results provide positive correlation value between coverage and fault rate and proved experimentally in terms of performance in coverage-based techniques, because in existing work, the coverage and fault are not correlates based on the test cases with lower coverage find more faults. With this, coverage-based techniques will consist of random nature. This is the overcome in current system.

This is made in the form of clusters that depends on closed functional dependency. Assessments took with APFD, the priority elevated the fault rate and coverage strongly correlated with the fault rate. It is viable to enhance the capacity of scheduled test cases in test case prioritization to meet some performance goal in the confidence in reliability to increase rate along with fault detection and other coverage techniques in future.

Using Steepest Descent Hill Climbing Algorithm in test case prioritization on ordering of test suite based on function coverage techniques. Hill Climbing is a simple and cheap to consider and implement as future work.

VI. REFERENCES

[1]. Eghbali, S., &Tahvildari, L. (2016). "Test Case Prioritization Using Lexicographical Ordering". IEEE Transactions on Software Engineering, 42(12), 1178-1195.

[2]. Elbaum, S., Malishevsky, A. G., &Rothermel, G. (2000). "Prioritizing test cases for regression testing". (Vol. 25, No. 5, pp. 102-112).ACM.

[3]. Elbaum, S., Malishevsky, A. G., &Rothermel, G. (2002). "Test case prioritization: A family of empirical studies". IEEE transactions on software engineering, 28(2), 159-182.

[4]. Rothermel, G., Untch, R. H., Chu, C., &Harrold, M. J. (2001). "Prioritizing test cases for regression testing. IEEE Transactions on software engineering", 27(10), 929-948.

[5]. Akbar SiamiNamin, James H. Andrews, "The Influence of Size and Coverage on Test Suite Effectiveness" Department of Computer Science", Texas Tech University at Abilene, University of Western Ontario London, Ontario, Canada.

[6]. Inozemtseva, L., & Holmes, R. (2014, May). "Coverage is not strongly correlated with test suite effectiveness."In Proceedings of the 36th International Conference on Software Engineering (pp. 435-445).ACM.

[7]. Srikanth, H., Williams, L., & Osborne, J. (2005, November). "System test case prioritization of new and regression test cases".In Empirical Software Engineering, 2005.2005 International Symposium on (pp. 10-pp).IEEE.

[8]. Carlson, R., Do, H., & Denton, A. (2011, September). "A clustering approach to improving test case prioritization: An industrial case study". In Software Maintenance (ICSM), 2011 27th IEEE International Conference on (pp. 382-391).IEEE.

[9]. Arafeen, M. J., & Do, H. (2013, March). "Test case prioritization using requirements-based clustering". In Software Testing, Verification and Validation (ICST), 2013 IEEE Sixth International Conference on (pp. 312-321). IEEE.

[10]. Miller, T. (2013). "Using dependency structures for prioritization of functional test suites". IEEE Transactions on Software Engineering, 39(2), 258-275.

[11]. Indumathi, C. P., &Selvamani, K. (2015). "Test Cases Prioritization Using Open Dependency Structure Algorithm". Procedia Computer Science, 48, 250-255.