IJCSMA

Ekta *et al*, International Journal of Computer Science and Mobile Applications,
Vol.5 Issue. 8, August- 2017, pg. 30-42 **ISSN: 2321-8363**
Impact Factor: 4.123

A Novel Approach towards Test Suite Optimization in Regression Testing

Ekta[1], Sonal[2]
[1]M.Tech Student, [2]Assistant Professor
Department of Computer Science & Engineering
BPS Mahila Vishwavidyalaya Khanpur Kalan, Sonipat
Ektadalal108@gmail.com, sonalkharb@gmail.com

Abstract: The size of a test-suite has a great impact on the cost and the time consumed in software testing[21][22] It is important in case of regression testing specially, where software is again tested after some changes, the size of the test suite is very important[3][4]. Test-suite minimization or reduction techniques attempt to reduce the time of test suite execution by eliminating redundant test cases from test suites[1][2].

This paper suggest a novel approach where test cases created with model checker based techniques are transpose such that redundancy the test – suite is evade, and the overall size is decrease.

Keywords: Regression testing, test suite reduction, test suite minimization techniques, empiric view.

1. INTRODUCTION

Software testing is a growing area. It can define it as "Software testing" is the process of executing the program or a software application with the intention of finding faults[1]. For this purpose there is a need to generate the test cases for the execution so that it can discover the faults[3]. As the software size grows up there are thousands of test cases to be executed. Because time is a key, if need to reduce the size of a test suite especially during regression testing, which is performed whenever a change has been put in any application program. In such cases testers have been used some reduction techniques to minimize the size of a test-suite[23].

Test suite minimization for optimization is the processes of selecting those test cases that satisfies some given requirement or removing all those are become redundant. Parsa et al.[13][18] define the test suite minimization as follows:

Definition: A test suite T with a given set of test requirements like {r1, r2,…., rn}, that must be satisfied and a set of subset of T as T1, T2,…Tn, where every subset is associated with each of the requirement ri such that a test cases tj belonging to subset Ti can be used to achieve each requirement ri. To find a minimized set i.e. a representative set, T' from T that satisfies all ri`s.

Regression testing is performed by the testing team after some modification i.e. a change that was given for some bug fixes or for some new functionality. So whenever any change has been put up in a code or in a whole system, there is a need to perform regression testing[4][12]. It is a testing activity that can be performed with the intention of providing the confidence to the users

or a client that a change will not harm the existing behavior in a software product. Regression testing can be performed effectively in three ways that are: Test suite reduction, test suite selection, test suite prioritization[5][6]. The impact on fault detection capability can be calculated by decreasing one from the ratio of faults in reduced test suite and total number of faults in original test suite[10][24]. The value is calculated as a percentage, so multiply it by 100.
It is an issue that was given by many of researchers. They claim that it could be happened.

2. TEST SUITE MINIMIZATION TECHNIQUES

Regression Testing is a growing area. It is performed whenever any change has been put up into a system. Whenever any testing activity is performed there is a process of executing test cases. Regression testing becomes very difficult with such large suite[3]. For the sake of time and cost there are various techniques are discovered to reduce that size of a suite. Such minimization of size of a test suite is also known as Test suite Optimization[11][12][14]. Regression testing activity can be processing in these approaches as follows:

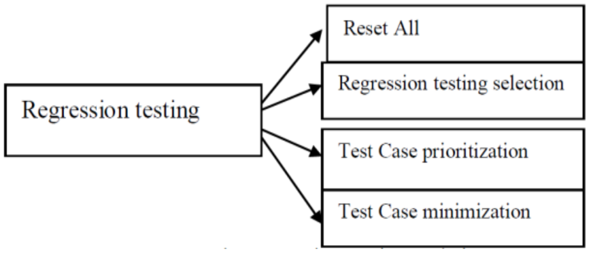

Fig: regression testing techniques

Test suite minimization techniques attempt to reduce the cost of saving and reusing tests during software maintenance[7][8].
Test suite minimization techniques are of various types. These techniques can be categorized as:

- Heuristic based test suite minimization
- Greedy based test suite minimization
- Set theory based test suite minimization[25]
- Model for test suite minimization

Heuristic based techniques are based on three factors: essentialness, redundancy and one to one redundancy.

Greedy based technique used the greedy approach. According to Analysis and Design algorithms in a greedy approach we need to find a feasible solution that either maximize or minimizes a given objective function[25]. A feasible solution that does this is called an optimal solution. So this technique first selects the optimal test cases, means selects those test cases those are satisfying maximum number of requirements[16][17][18].

Set theory based technique used set operations as a tool for minimization. In this approach a Regression matrix is formed where there is an entry for Can that is the union operation of between faults between test cases[13]. The iteration will be continued till the total number of faults will be identified.

A model for test suite minimization that was proposed by Ankur Mudgal [20] is also used as a test suite minimization technique. Here Boolean matrix A=[aid] is formed. Which is of order n×(m+1). This matrix is used for describing a satisfaction relation between test cases and relation. In matrix A, aij is as follows:

$$aij= \begin{cases} 0 \text{ if test case ti can't test requirement rj} \\ 1 \text{ if test case ti can test rj} \end{cases}$$

By using some rules this matrix is updated until then an identity matrix is developed.

3. EFFECTIVENESS CALCULATION

Effectiveness of a test suite minimization technique can be calculated by various factors like number of faults detected by reduced test suite, amount of time taken by a reduced set etc[9][10]. Here some example for doing such evaluation of minimization techniques.

Test suite minimization techniques evaluation:

For evaluation purpose if taking an example of a test suite on which reduction technique will be applied[19][15]. It is of tabular form where the rows entries will taken as test cases and columns will be considers for their satisfying requirements.

The regression Test Suite, T, is as follows:

Test Case	**r 1**	**r 2**	**r 3**	**r 4**	**r 5**	**r 6**	**r 7**	**r 8**	**r 9**	**r 1**
T1	0	1	0	1	0		1	0	1	0
T2	1	0	1	0	0	0	0	0	0	0
T3	1	0	0	0	1	0	1	1	0	0
T4	0	1	0	1	0	0	0	0	1	0
T5	0	0	1	0	0	1	0	0	0	1
T6	1	0	0	0	0	0	1	0	0	0
T7	0	0	1	0	0	1	0		0	0
T8	0	1	0	0	0	0	0	0	0	1

Table 3.1 Regression Test Suite T

Evaluation for Heuristic approach

Heuristic **H**: Firstly select an essential test case that is t3 in this above example. Then next select t1, t2 and t5 those are satisfying most of remaining requirements. So the representative set RS= {t1, t2, t3, t5}

Heuristic **GE** : (Greedy and Essential)

First the essential test case that is t3 after that greedy approach is applied for remaining requirements and we select t1 and t5
RS= {t1, t3, t5}

Heuristic **GRE**: (Essential, Redundant and Greedy)

First remove all redundant that make other remaining essential then apply greedy on remaining set. So selected here t4 and t6 is redundant then select essential test case that is again t3 then after applying greedy we get the representative set RS={t1,t3,t5}

Calculation

Effectiveness of techniques:-

$1-\frac{4}{8}\times 100$ =50% for H

$1-\frac{3}{8}\times 100 = 62.65\%$ for GE and GRE

Between all three heuristic GE and GRE are best to apply for reducing test cases then heuristic H[24]. But in GE and GRE, GRE is best when overlapping between the requirements is lying in between <2 to >15.

Evaluation for Greedy based approach

Greedy approach takes maximum number of requirement satisfying test cases, and then less number than the selected one[10]. By selecting those test cases in such a manner that representative set as:

RS={t1,t3,t5} which is same as above 2 heuristics. I took an example to compare greedy with heuristic based technique as shown below:

Test Case	r1	r2	r3	r4	r5	r6
T1	1	1	1	0	0	0
T2	1	0	0	1	0	0
T3	0	1	0	0	1	0
T4	0	0	1	0	0	1
T5	0	0	0	0	1	0

Table 3.2 Regression Test Suite T

By applying greedy approach the RS= {t1, t2, t3, t4} and heuristic gives RS= {t2, t3, t4}
By looking at those sets are concluding that Heuristic technique is better than Greedy one[25]. Another Heuristic given by Tallam et.[2]. Al named HGS heuristic is also better then Greedy (classical).

Evaluation of Model based technique

It is based on Regression matrix formation model. From table1 selective rules those are defined for this technique.

Steps:

1) Applying rule#1 on table 1 data we have ai=t4 and aj=t1; we remove t4
2) Also apply rule#1 for ai=t6 and aj=t3; remove t6.update the table.
3) Apply rule#3 on updated table for bi=r4 and bj=r2; r2 is removed
4) Apply rule#3 for bi=r1, bj=r5; r1 is removed.
5) Apply rule#3 for bi=r6,bj=r3; r3 is removed
6) Apply rule#3 for bi=r4 and bj=r7; r7 is removed
7) Apply rule#1 on next updated table data we have ai=t8, aj=t5; t8 is removed and also t2 becomes zero entirely so delete this row.
8) Apply rule#3 on next updated table data, we have bi=r5,bj=r8; r8 is removed
9) Apply rule #3 for bi=r10,bj=r6; r6 is removed. The table is updated.
10) Applying rule#1 for ai=r9, aj=r4; r9 is removed. Update the table. To require the last matrix of iteration as it becomes an identity matrix.

Updated table

Test Case	r4	r5	r10
T1	1	0	0
T3	0	1	0
T5	0	0	1

Table 3.3 next iterated tables

©2017, IJCSMA All Rights Reserved, www.ijcsma.com

Effectiveness of technique: $1-\frac{3}{8} \times 100 = 62.65\%$

Which is same as Heuristic GRE and GE but it is more complex to calculate.
So for large test suite it is better to apply heuristic technique instead of model based technique.

Evaluation of set theory approach

Taking same table1 for this technique but here the faults are taken in place of requirements.
Also the time to execute the given test cases is also considered. Sample data given for this technique is illustrated by the table given below:

Test Case	**F 1**	**F 2**	**F 3**	**F 4**	**F 5**	**F 6**	**F 7**	**F 8**	**F 9**	**F 1**
T1	0	1	0	1	0	0	1	0	1	0
T2	1	0	1	0	0	0	0	0	0	0
T3	1	0	0	0	1	0	1	1	0	0
T4	0	1	0	1	0	0	0	0	1	0
T5	0	0	1	0	0	1	0	0	0	1
T6	1	0	0	0	0	0	1	0	0	0
T7	0	0	1	0	0	1	0		0	0
T8	0	1	0	0	0	0	0	0	0	1

Table 3.4 Algorithm's Input

Test Case	**Number of faults covered**	**Execution Time**
T1	4	7
T2	2	4
T3	4	5
T4	3	4
T5	3	4
T6	2	5
T7	3	4
T8	2	2

Table 3.5 Sample Data with Execution Time

Now take the set theory operation UNION and create a regression matrix of test cases i.e. test cases are put on row and column position. After apply union operation the regression matrix obtained: **Iteration1**

The value of 10 in a cell represents that a test suite that has covered all the faults. To consider that in this iteration no test case suite is generated which covers all the faults. So we have to iterate the procedure further and we further consider only those pairs of test cases that cover faults more than the given threshold value (α=5 faults) shown in table below:

Test case	**T2**	**T3**	**T4**	**T5**	**T7**	**T8**
T1	6	7		7	7	
T3			7		6	6
T4				6	6	

Table 3.6 updated table

Iteration2: In this iteration, combinations of test cases are produced by applying union on selected test cases of iteration 1 and original test suite T

Test Case	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**	**T8**
T1T	X	X	8	6	8		8	7
T1T	X	X	X	7	10	7	9	8
T1T	X	8	10	7	X	7	8	7
T1T	X	8	9	7	8	8	X	8
T3T	7	8	X	X	10	7	9	8
T3T	9	6	X	9	7	6	X	8
T3T	8	7	X	8	8	6	8	
T4T	7	7	10	X	X	8	7	6
T4T	7	7	9	X	7	8	X	7

Table: 3.7 iterated table

X indicates that this value is not significant as R U S U R = R U S

Test Suite	**Number of faults covered**	**Total Execution Time**
T1 T3 T5	10	16
T3 T4 T5	10	13

Table: 3.8 effectiveness of techniques

From the above table to determine the test suite of minimum execution time. So representative set: RS= {t3, t4, t5}

Effectiveness of technique:- $\{ 1-\frac{3}{8} \times 1\}00 = 62.65\%$

This is same as model based technique, but to examine it by the time of test suite execution

Model Based	**Set theory based**
T1 T3 T5	T3 T4 T5
16	13

Table 3.9 comparison b/w model based and set theory based

Also comparison can be represented by **fault detection capability** that is

$$\frac{\textit{Number of faults detected by the reduced test suite}}{\textit{Number of faults detected by the original test suite}} \times 100$$

For model based technique: $1-\frac{11}{23} \times 100 = 52.1\%$

For Set theory based technique: $1-\frac{10}{23} \times 100 = 56.5\%$

So Model based is better then set theory based approach as its reduction in fault detection capability is less.

4. RESULT ANALYSIS

Technique Name	Effectiveness in % age
1) Heuristic Based	
I. H	50%
II. GE & GRE	62.65%
2) Greedy Based	50%
3) Model based	62.65%
4) Set Theory Based	62.65%
5) Hybrid Approach(BCO+GA)	50%

Table 4.1 Effectiveness of techniques

Analysis for Heuristic technique: From above to consider that among Heuristic based techniques (H, GE & GRE) the GRE approach is more effective. As researchers said that between all three heuristic Heuristic GE and GRE are best to apply for reducing test cases than heuristic H. But in GE and GRE, GRE is best when overlapping between the requirements is lying in between <2 to >15.

Analysis b/w Greedy and Heuristic technique: As can see from above Greedy is not more effective than Heuristic.
Result: Heuristic is better.

Analysis b/w Model based technique and Set theory based technique: Effectiveness of both techniques is same. So can use another factor: Fault detection capability reduction, comparison can be represented by **fault detection capability** that is

$$\frac{\text{Number of faults detected by the reduced test suite}}{\text{Number of faults detected by the original test suite}} \times 100$$

For model based technique: $1-\frac{11}{23} \times 100 = 52.1\%$

For Set theory based technique: $1 - \frac{10}{23} \times 100 = 56.5\%$

Technique	Fault detection capability reduction in %age
Model based	52.1%
Set theory based	56.5%

Table 4.2 Fault Detection capability of techniques

Result: So Model based is better then set theory based approach as its reduction in fault detection capability is less.

Analysis b/w Heuristic and Model based technique: Between both techniques the application to get reduced test suite this is observed that the technique Model based is much vaster and complex[22][23].To apply than the heuristic one that gives the representative set in a easiest way and not much complex as model based technique is Model based technique takes more effort and time when applying on even a smaller test-suite.
Result: Heuristic is better.

Analysis b/w Hybrid and Heuristic technique: To consider that all three factors here for analyzing which one is best between both.

Minimization Technique	Effectiveness (%age)	Execution Time	Fault Detection Capability Reduction
Heuristic approach	62.65	13	52.1
Hybrid Approach	50	18	43.47

Table 4.3 Comparison between Heuristic and Hybrid

Result: From above table overall than conclude that Heuristic technique is better than Hybrid technique
Final Result: Heuristic technique is the best one to apply for test suite minimization.

5. CONCLUSION

Regression testing is a kind of testing that helps developers make sure that there are no defects after the application has been changed. The overview of regression testing is designed to provide information about regression testing as a whole. There are numerous techniques used to in regression testing. Since testing is complex and there is no direct measure of fault revelation likelihood and there are many different types of cost involved. A certain regression testing technique cannot be used at each and every scenario and there exists a need to find a technique that can be used at a particular scenario every time the scenario is obtained. Thus measured the effectiveness of different test suite minimization techniques by using some formulas and facts.

In this work, to compare the four minimization techniques

i. Heuristic based
ii. Greedy based
iii. Model based
iv. Set theory based techniques.

Heuristic technique is the best one to apply for test suite minimization.

REFERENCES

[1] T.Y. Chen, M.F. Lau, “On the divide-and-conquer approach towards test suite reduction,” Information Science, Vol.152, 2003, PP. 89-119.

[2] S.Tallam, N.Gupta, “A Concept Analysis Inspired Greedy Algorithm for Test Suite reduction”, 2005 ACM 1595932399/05/ 0009, 2003.

[3] D.Jeffrey, N.Gupta, “Test Suite Reduction with Selective Redundancy”, Proceedings of the 21st IEEE International Conference on Software Maintenance (ICSM’05), 1063-6773/05 $20.00, 2005 IEEE

[4] S. Sprenkle, S. Sampath, E. Gibson, A. Souter, L. Pollock, ”An Empirical Comparison of Test Suite Reduction Techniques for User-session-based Testing of Web Applications,” Technical Report 2005-009, Computer and Information Sciences, University of Delaware, November 2004.

[5] S.Rehman Khan, A. Nadeem, “Test Filter”: “A Statement-Coverage Based Test Case Reduction Technique”, 1-4244-0794-X/06/$20 .00, 2006 IEEE

[6] A. Smith, J. Geiger, M.L Soffa, Test Suite Reduction and Prioritization with Call Trees, ACM 978-1-59593-882-4/07/0011, 2007.

[7] Z. Chen, X.Zhang and B. Xu, “A Degraded ILP Approach for Test Suite Reduction, National Natural Science Foundation of China (60425206, 60773104, 60403016, 60633010), 2008, PP. 390-394.

[8] Jun-Wei Lin, Chin-Yu Huang, "Analysis of test suite reduction with enhanced tie-breaking techniques", Information and Software Technology", 0950-5849/$, 2008 Elsevier

[9] S.Yoo, M.Harman, "Using hybrid algorithm for Pareto efficient multi-objective test suite minimization", The Journal of Systems and Software 83, 2010, PP.689-701.

[10] S.Nachiyappan, A.Vimaladevi, C.B. Selva, "An Evolutionary Algorithm for Regression Test Suite Reduction", Proceedings of the International Conference on Communication and Computational Intelligence, 2010, PP. 503-508.

[11] S. Yoo, M. Harman, "Regression Testing Minimization, Selection and Prioritization : A Survey", King's College London, Centre for Research on Evolution, Search & Testing, Verification and reliability, Vol. 22, No. 2, 2010, PP.67-120.

[12] B.Galeebathullah, C.P.Indumathi, "A Novel Approach for Controlling a Size of a Test Suite with Simple Technique", (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 03, 2010, PP.614-618.

[13] **Parsa et al,** *"A new algorithm for test suite minimization"*, International Journal of Engineering Research & Technology (IJERT), 2010.

[14] L.Zhang, D. Marinov, Lu Zhang, S.Khurshid, "An Empirical Study of JUnit Test-Suite Reduction", 2011 22nd IEEE International Symposium on Software Reliability Engineering, 2011 IEEE

[15] ***S.* Kadry,** "*A New Proposed Technique to Improve Software Regression Testing cost",* international Journal of Security and its Applications, 2011.

[16] B.Suri, I.Mangal & Varun Srivastava,"Regression Test Suite Reduction using an Hybrid Technique Based on BCO and Genetic Algorithm", Special Issue of International Journal of Computer Science & Informatics (IJCSI), ISSN (PRINT): 2006, 2231-5292, Vol. II, No. 1,2

[17] S. Xu, H. Miao, H. Gao, "Test Suite Reduction Using Weighted Set Covering Techniques", 2012 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2012 IEEE

[18] **Jose Campos,** Mestrado "*Regression Testing with GZoltar: Techniques for Test Suite Minimization, Selection, and Prioritization"***,** Integrado An Engenharia Information Computation: Rui Abreu (PhD), 2012.

[19] **Shengwei Xu, Huaikou Miao, Honghao Gao,** "*Test Suite Reduction Using Weighted Set Covering Techniques"*, 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, IEEE**,** 2012

[20] A.Mudgal, "Proposed model for minimization of test suite (PMMTS)", Journal of Nature Inspired Computing (JNIC) 34 Vol. 1, No. 2, 2013.

[21] R.Singh, M.Santosh, "Test Case Minimization Techniques: A Review" International Journal of Engineering Research & Technology (IJERT), Vol. 2, No.12, 2013, PP. 1048-1056.

[22] I Mangal, D. Bajaj, P.Gupta, "Regression Test Suite Minimization using Set Theory", International Journal of Advanced Research in Computer Science and Software Engineering, 2014. Volume 4, Issue 5, May 2014, PP. 502-506

[23] Renu, R. Singh, "Test Suite Minimization using Evolutionary Optimization Algorithms: Review", International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181, Vol. 3 Issue ,2014.

[24] **Usman Farooq, "***Model Based Test Suite Minimization Using met heuristics***",** Proceedings of the third IASTED Conference on Advances in Computer Science and Technology, ISBN: 978-0-88986-656-0, 2014.

[25] P. Harris, N.Raju, "A Greedy Approach for Coverage-Based Test Suite Reduction", The International Arab Journal of Information Technology, Vol. 12, No.1, 2015, PP. 17-23.