ISSN:2231-0711

Saakshi Narula| IJCSET(www.ijcset.net) |April 2016 | Vol 6, Issue 4, 126-128

Review Paper on Test Case Selection

Saakshi Narula

The Northcap University, M.tech student
Gurgaon, Haryana
India

***Abstract*-Regression testing is of great significance and costly action that happens each time a program is transformed or modified to ensure that changes do not introduce new bugs into the previous accurate code. It is important to select a relevant subset of test cases from the initial test suite that would minimize the effort and time. Optimization of a test suite is the central in the development process of a software, keeping into account the resource and time constraints. The aim of test selection is to exclude or eliminate the redundant test data, which is a key for definition of the test strategies. This paper is a systematic review on Test Case Selection which is conducted in leading conferences and journals. The most commonly reported techniques include genetic algorithm, adaptive random testing and greedy algorithm.**

Keywords: Regression Testing, Test Cases, Test suite, Test Case Prioritization, Test Case Minimisation, Test Case Selection.

I. INTRODUCTION

To develop a reliable software the principal is testing. We neglect specifications and there are number of barriers to the execution which includes time constraints, inadequate planning, indiscriminate costs and lack of automated tools.

A constructive test depends on the group of detailed and specific conditions that are used in software testing which is known as test cases (TC). Test cases are defined as a set of input data, pre and post conditions and the expected results that are used for testing. A set of one or more test cases is known as test suite. As the software requirements evolve the size of the test suites tend to increase, making execution of the entire suite to be unfeasible, time consuming and resource constraints. Test case selection is a regression testing technique which aims at choosing a group of test cases within a specific area related to the criterion of interest. This technique is performed to decrease the size of the test suite. Its aim is to eliminate the redundant test data and maximization of fault detection. To develop a reliable software the principal is testing. We neglect specifications and there are number of barriers to the execution which includes time constraints, inadequate planning, indiscriminate costs and lack of automated tools.

Regression testing is of great significance and costly action that happens each time a program is transformed or modified to ensure that changes do not introduce new bugs into the previous accurate code. Regression testing is a type of testing that a program has not regressed, that is the features or functionalities that were working in the previous version is still working in the new version [2]. We perform regression testing between two different versions of a software to ensure that the new features introduced in the latest version do not interfere with the already existing features. It uncovers new software bugs or regressions in existing non-functional and functional areas of the system after changes such as patches, enhancements or configuration alterations have been made to them. Regression testing is expensive but it is a necessary task.

The intent of this testing is to ensure no new faults have been introduced due to the changes. The foremost reason to perform this testing is to determine whether a refinement in a part of the software affects the other one. It ensures the quality of the software being developed and approves the revised software. The main strategy is to minimize the test suite and maximize the ability to detect faults. Regression Testing has different approaches- Test Case Selection, Test Case Prioritization, Test Suite Minimization. Test Case Selection technique tries to decrease the number of test cases to be executed, while satisfying the testing requirements [1].

II. LITERATURE SURVEY

This section presents the survey on search-based approaches for the solution of test case selection.

Fischer et al. prepared a test case selection problem with the application of Integer Programming [3]. In this approach there was no study about the variations of the control flow. Agrawal et al. outlined an exclusive strategy on test case selection with a special interpretation to the discrepancies found in the program slicing techniques [4]. Rothermel and Harrold explained regression test case selection techniques based on graph walking of Control, Program Dependence Graphs [5], and, Control Flow Graphs [7], system Dependence Graphs [6].

Benedusi et al. implemented path analysis for test case selection [8]. A testing formation called TestTube was introduced by Chen et al. [9] which made use of a modification-based method for selection of test cases. Leung and White emphasized a firewall technique for regression testing of system integration [10]. Laski and Szemer provided a technique for test case selection which is based on cluster identification technique [11]. In 1997, Baradhi and Mansour applied Genetic Algorithm and Simulated Annealing to minimize execution of time and the number of test cases and to maximize the precision [12].

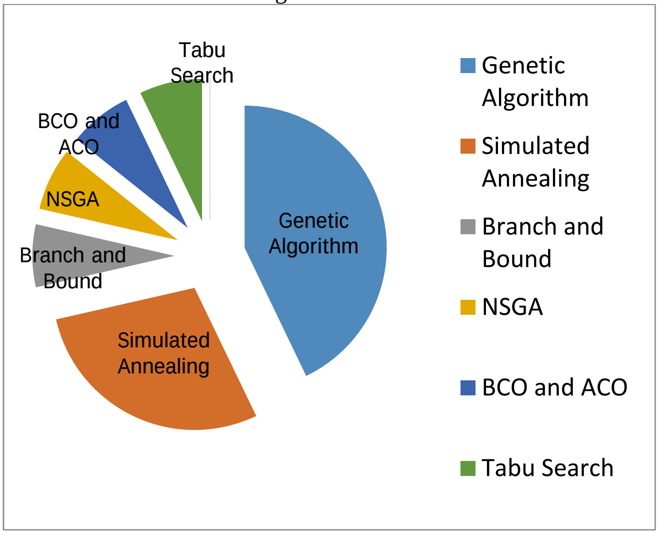
In 1999, Mansour and El-Fakih applied Simulated Annealing, Genetic Algorithm and Branch and Bound to minimize the number of test cases needed for execution [13].

In 2001, applied Simulated Annealing was applied by Mansour et al. on test cases to minimize the number of selected test cases, to reduce the precision and to reduce the time of execution [14]. In 2002, Mansour and Bahsoon applied Simulated Annealing technique to maximize test code coverage. In 2007, Yoo and Harman applied NSGA-II on test cases to maximize code coverage, to minimize the cost and to minimize the execution [15]. In 2012, Isha Mangal and Bharti Suri applied three techniques namely Genetic Algorithm, Bee Colony Optimization, Ant Colony Optimization. These were applied to achieve the fitness function of reduction is test suites, reducing the cost, reducing the execution time and minimizing the subset of test cases [16]. In 2013, Tanvi Agarawal, Arun P. Agrawal applied Genetic Algorithm and Tabu Search on test cases to minimize the time of execution and for the effective quality value [17]. Yoo and Harman [18] wrote a survey on the approaches of minimization, prioritization and test case selection for regression testing. They discussed about the methods that were used to minimize the test suite, relations between the approaches, as well as the features, advantages and disadvantages of each related topic. Chen et al. [19] presented an overview of important theoretical studies which is related to adaptive random testing. Researchers discuss about the relevant topics, such as faulty patterns, theoretical limits, fault-based testing and random sequences. Kasurinen et al. [21] presented a survey on how companies choose test cases selection in their software development research projects. Researchers classified their strategies into two groups: (1) risk-based which helps to focused on testing parts of the software considered too costly to repair after the final distribution; and (2) design-based, in which testing is intended to ensure that the software is able to perform the tasks for which it was designed, in terms of features. In 2014, Kevilienuo Kire, Neha Malhotra applied Simulated Annealing and Genetic Algorithm on test cases to minimize the time of execution [20].

Table I- Comprehensive Report

Year	**TEST CASE SELECTION**		
	Authors	**Technique**	**Fitness Function**
1977	Fischer et al.	Integer Programming	--
1988	Benedusi et al.	Path Analysis	--
1993	Agrawal et al.	Program slicing technique	Interpretation to discrepancies
1994,1997	Rothermel and Harrold	Techniques based on graph walking of Control, Program Dependence Graphs, and, Control Flow Graph, system dependence Graphs	Explanation of regression test selection techniques
1997	Baradhi and Mansour	Genetic Algorithm, Simulated Annealing	1.Minimize Execution Time and the number of test cases, 2. Maximize the Precision
1999	Mansour and El-Fakih	Simulated Annealing, Genetic Algorithm, Branch and Bound	Minimize number of test cases needed.
2001	Mansour et al.	Simulated Annealing	1. Minimize number of selected test cases and execution time. 2. Maximize Precision
2002	Mansour and Bahsoon	Simulated Annealing	Maximize test code coverage
2007	Yoo and Harman	NSGA-II	1. Maximize coverage 2. Minimize execution time 3. Minimize cost
2012	Isha Mangal and Bharti Suri	Genetic Algorithm, Ant Colony Optimization, Bee Colony Optimization	1. Reduction in test suites. 2. Minimize cost 3. Minimize execution time 4. Minimum subset of test cases.
2013	Tanvi Agarawal, Arun P. Agrawal	Tabu Search, Genetic Algorithm	1. Less execution time 2. Better quality value
2014	Kevilienuo Kire, Neha Malhotra	Simulated Annealing, Genetic Algorithm	Less execution time

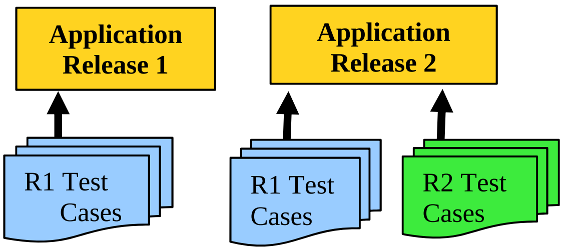
Figure 1: Distribution of various metaheuristics applied for solving Test Case Selection



III. Need For Test Case Selection

As the number of test case are increasing so, it becomes impossible for the tester's to test all the generated test cases because of limited cost, time and resource.

Figure 2



IV. Objectives Of Test Case Selection

Test Case Selection has the following two objectives: Maximum Code Coverage and Minimum Cost.

Code Coverage is the measurement done when automated tests are running which calculates how many lines/arcs or blocks of code are executed while the tests run. Maximum code coverage tells how much maximum code is covered under test. It tells upto what extent the set of test cases cover the source code.

Minimum cost is the objective which ensures that test cases that are selected are on the basis of execution time, cost of running the test cases, cost of analysis, etc.

V. Conclusion

This paper gives a summary of the techniques applied to Regression Test Case Selection. Further investigation and analysis may be carried out on more techniques which will help to propose new ideas that will give maximum code coverage and minimum execution cost, leading to deliver a product of high quality.

References

[1] Shin Yoo and Mark Harman, King's College, London, *Pareto Efficient Multi-Objective Test Case Selection*.

[2] S. Yoo, M. Harman, *Regression Testing Minimisation, Selection and Prioritisation: A Survey*. SOFTWARE TESTING, VERIFICATION AND RELIABILITY Softw. Test. Verif. Reliab. 2007; 00:1–7 (DOI: 10.1002/000) Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/000.

[3] Fischer K. *A test case selection method for the validation of software maintenance modifications*. Proceedings of International Computer Software and Applications Conference, IEEE Computer Society Press, 1977; 421–426.

[4] Agrawal H, Horgan JR, Krauser EW, London SA. *Incremental regression testing. Proceedings of the International Conference on Software Maintenance* (ICSM 1993), IEEE Computer Society, 1993; 348–357.

[5] Rothermel G, Harrold MJ. *Selecting tests and identifying test coverage requirements for modified software*. Proceedings of International Symposium on Software Testing and Analysis (ISSTA 1994), ACM Press, 1994; 169–184.

[6] Rothermel G, Harrold MJ. *A safe, efficient regression test selection technique*. ACM Transactions on Software Engineering and Methodology April 1997; 6(2):173–210.

[7] Rothermel G, Harrold MJ. *Experience with regression test selection*. Empirical Software Engineering: An International Journal 1997; 2(2):178–188.

[8] Benedusi P, Cmitile A, De Carlini U. *Post-maintenance testing based on path change analysis*. Proceedings of the International Conference on Software Maintenance (ICSM 1988), IEEE Computer Society Press, 1988; 352–361.

[9] Chen YF, Rosenblum D, Vo KP. *Testtube: A system for selective regression testing*. Proceedings of the 16th International Conference on Software Engineering (ICSE 1994), ACM Press, 1994; 211–220.

[10] Leung HKN, White L. *Insights into testing and regression testing global variables*. Journal of Software Maintenance 1990; 2(4):209–222.

[11] Laski J, Szermer W. *Identification of program modifications and its applications in software maintenance*. Proceedings of the International Conference on Software maintenance (ICSM 1992), IEEE Computer Society Press, 1992; 282–290.

[12] Mansour, N., Bahsoon, R., and Baradhi, G. (2001). *Empirical Comparison of Regression Test Selection Algorithms*. Journal of Systems and Software, 57(1), 79–90.

[13] Mansour, N. and El-Fakih, K. (1999). *Simulated Annealing and Genetic Algorithms for Optimal Regression Testing*. Journal of Software Maintenance: Research and Practice, 11(1), 19–34.

[14] Mansour, N. and Bahsoon, R. (2002). *Reduction-based Methods and Metrics for Selective Regression Testing*. Information and Software Technology, 44(7), 431–443.

[15] Yoo, S. and Harman, M. (2007). *Pareto Efficient Multi-Objective Test Case Selection*. In Proceedings of the 2007 International Symposium on Software Testing and Analysis (ISSTA '07), pages 140–150, London, England. ACM.

[16] Tanvi Agrawal, Arun P. Agrawal. *Regression Test Selection Using Metaheuristics*. International Journal of Innovative Technology and Exploring Engineering (IJITEE)ISSN: 2278-3075,Volume-3, Issue-1,June 2013.

[17] Bharti Suri, Isha Mangal. *Analyzing Test Case Selection using Proposed Hybrid Technique based on BCO and Genetic Algorithm and a Comparison with ACO*. 2012, IJARCSSE All Rights Reserved Page | 206Volume 2, Issue 4,April 2012.ISSN: 2277 128XInternational Journal of Advanced Research in Computer Science and Software Engineering Research Paper.

[18] S. Yoo and M. Harman, *Regression testing minimization, selection and prioritization: A survey*, Journal of Software Testing, Verification and Reliability 22 (2012) 67–120.

[19] T. Y. Chen, F. C. Kuo, R. G. Merkel and T. H. Tse, *Adaptive random testing: The ART of test case diversity*. Journal of Systems and Software 83 (2010) 60–66.

[20] Kevilienuo Kire, Neha Malhotra. *Study of Test Case Selection and Prioritization*. International Journal of Computer Applications (0975 – 8887) Volume 85 – No 5, January 2014.

[21] J. Kasurinen, O. Taipale and K. Smolander, *Test case selection and prioritization: Risk based or design-based* in Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ACM, New York, USA, 2010).