ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 8, August 2019

Review on Test Case Prioritization Using Fitness Function

Varsha Rani [1], Umesh Goyal[2]

P.G. Student, Department of Computer Science & Engineering, Delhi Institute of Technology Management and Research, Faridabad, Haryana, India[1]

Assistant Professor, Department of Computer Science & Engineering, Delhi Institute of Technology Management and Research, Faridabad, Haryana, India [2]

ABSTRACT: Present-day programming development requires a powerful relapse testing process so as to continue progressing through a progression of superb refreshed programming renditions. Each such refreshed rendition represents an incalculable development in the number of experiments. In such a circumstance, the key plan to improve the viability of programming testing is experiment prioritization. The current experiment prioritization methodologies decide the execution request of all experiments by considering just the experiment auxiliary inclusion data of the first form, without considering the execution data of experiments for the altered adaptation of the program. Test case prioritization involves scheduling test cases in an order that increases the effectiveness in achieving some performance goals. One of the most important performance goals is the rate of fault detection. Test cases should run in an order that increases the possibility of fault detection and also that detects the most severe faults at the earliest in its testing life cycle. In this paper, we develop and validate requirement based system-level test case prioritization scheme to reveal more severe faults at an earlier stage and to improve customer-perceived software quality using Genetic Algorithm (GA). For this, we propose a set of prioritization factors to design the proposed system. In our proposed technique, we refer to these factors as Prioritization Factors (PF). These factors may be concrete, such as test case length, code coverage, data flow, and fault proneness, or abstract, such as perceived code complexity and severity of faults, which prioritizes the system test cases based on the six factors: customer priority, changes in requirement, implementation complexity, completeness, traceability and fault impact. The goodness of these orderings was measured using an evaluation metric called APFD and PTR that will also be calculated. Regression testing is the process of validating modifications introduced in a system during software maintenance. It is expensive, yet an important process. As the test suite size is very large, system retesting consumes a large amount of time and computing resources. Unfortunately, there may be insufficient resources to allow for the re-execution of all test cases during regression testing. Test case prioritization techniques aim to improve the effectiveness of regression testing, by ordering the test cases so that the most beneficial is executed first with higher priority.

KEYWORDS: Software Testing, Genetic Algorithm, Prioritization Factors, Average Percentage Statement Coverage, Performance Test Report.

I. INTRODUCTION

The concept of adaptive approach or proposed is very interesting and beneficial for the accurate prioritization of test cases during the execution of test cases for the particular updated version of software on which, regression testing is to be performed. Keeping in mind the concept of adaptive approach, we proposed here an adaptive genetic approach of test case prioritization, which is a genetic optimization of the statement level adaptive approach of test case prioritization. This adaptive genetic approach uses the concept of adaptive approach for the test cases until all modified statements are covered at least once. The main motive behind the coverage of modified statements first is based on the fact that the coverage of modification gives us better and earlier detection of faults. After the coverage of modified

DOI: 10.15680/IJIRCCE.2019. 0708005

statements, the probability of faults decreases. So, the next main motive should be the coverage of existing statements, so that the remaining faults can be revealed at faster rate and for the better and faster coverage of existing statements, we used genetic algorithm for the prioritization of remaining test cases in the proposed adaptive genetic approach, because according to Mark Harman, genetic algorithm gives the better code coverage and better fault detection rate in maximum cases than all other static coverage information based test case prioritization approaches.

Test prioritization is used to coordinate and execute the examinations composed to extra cost and time. Trial prioritization is progressively beneficial and comprehensively used by the analyzers. Various researchers exhibited more methodologies for trial prioritization in backslide testing. Test prioritization for early inadequacy revelation has various essential central focuses:

1. First, if a bug is found as it so happens in the backslide testing process, investigating can start earlier and hereafter bugs could be fixed snappier.
2. Minimization of testing costs.
3. Testers may wish to assemble their trust in the unfaltering nature of the structure under test at a faster rate.
4. Identification of high risk relinquishes earlier.
5. Increasing trustworthiness of the item.

Classified the test case prioritization techniques into 6 broad categories and some other composite approaches by different combinations of these categories, on the basis of the approach used by these techniques for prioritization of test cases, the 6 broad categories are represented in Table.1

Name of the Approach	Description
Coverage based approach	More the coverage achieved by the test suite, more are the chances of revealing the faults earlier in the testing.
Modification based approach	Prioritize test cases based on the modifications done to the program.
Fault based approach	Prioritize test cases based on the fault exposing potential of test cases.
Requirement based approach	Test case prioritization is done based on the requirements.
History based approach	Based on the fact that the historical information may be useful in reducing costs and increasing the effectiveness of regression testing
Genetic based approach	Prioritization is done based on the genetic algorithm.

Table 1: Categories of Test Case Prioritization Approaches

In recent studies unified model of total and additional test case prioritization approaches and provide an empirical evidence that the spectrum of test case prioritization approaches lying in between purely total and additional techniques are more effective than the pure total and additional approaches of test case prioritization. The unifiedapproach updates the priorities of unselected test cases based on the coverage of latest selected test case for the original version of software. Zhang also presents an empirical evaluation of the influence of three internal (model type, fp value, and coverage type) and three external factors (coverage granularity, test case granularity, programming and testing paradigm) on the effectiveness of the prioritization techniques created through the unified approach. The conclusions drawn by the analysis of the effect of internal and external factors on the unified approach are as follows: The unified approach is more beneficial when using method coverage than when using statement coverage, considering the cost and effectiveness of the unified approach. The unified approach is more effective when applied to test cases at the test method level than at the test class level.

II. RELATED WORK

Praveen Ranjan Srivastva-Test case prioritization techniques incorporate arranging over investigations in a solicitation that improves the presentation of backsliding testing. It is inefficient to re-execute every investigation for

DOI: 10.15680/IJIRCCE.2019. 0708005

each program work if once change occurs. Examination prioritization systems mastermind the trials in a test suite by mentioning with the ultimate objective that the most beneficial are executed first hence thinking about a development in the suitability of testing. One of the display destinations, for instance, the inadequacy acknowledgment rate, is an extent of how quickly faults are recognized during the testing strategy. In this paper, he presents another analysis prioritization count, which figures ordinary defects found each minute. He presents the results depicting the ampleness of figuring with the help of APFD metric. The essential purpose of my paper is to choose the sufficiency of sorted out and non-composed case with the help of APFD.

Ashima Singh,Kamal Parkash-Testing is a fundamental bit of any item improvement lifecycle. It requires some venture and cash to make investigations and apply to test. Genetic Algorithms are exhibiting to be a fantastic instrument in improving programming testing. This paper uses the possibility of Genetic counts in overhauling programming testing. In this paper, we have separated genetic counts and focused their feasibility to find the defects and time overhead-based criteria to - arrange tests. The proposed strategy is giving the game plan of analyses sequencing similarly as decline by using an astute powerful system. The proposed structure will make the trials subject to the necessities, which are doled out by the count to examinations dependent on some canny undertakings. A joined change probability (CMP) metric is used to choose the sufficiency of the new trial orderings

Suman, Seema-Regression testing is the path toward supporting modified programming to ensure that changed bits of programming carry on as unaltered bits of programming has not been unfairly affected by the modification. The backslide test suite is usually gigantic and needs a shrewd system to pick those examinations which will reduce the general test cost. In this condition, analyze prioritization frameworks hope to improve the ampleness of backsliding testing by mentioning the examinations with the goal that the most profitable are executed first. In this system, another Genetic Algorithm to sort out the backslide test suite is displayed that will arrange analyzes intensely dependent on complete code incorporation. Meanwhile, an approach to manage to deliver new analyses is presented using PMX and cyclic half and half and examination is done dependent on methodology cost and test cost. The general purpose of this investigation is to diminish the number of analyses that ought to be sought after changes have been made.

Quience Gulati, Arun Prakash Agarwal- When as of late made writing computer programs is modified, there are chances that there may be a piece of mix-ups, which may incite dissatisfaction of the item system. The cost required to test every module of the structure will be amazingly high and even starting there ahead, testing all parts of the system is incredible as there is obliged time given to this stage. Backslide testing is ordinary these days as it helps in testing the modified and changed part, also lessening time and cost. The amount of trials is lessened using the health limit of genetic estimation. Furthermore, the examinations are consigned needs, which help in testing the more fundamental part on need. A dynamic system can be used to achieve proportional.

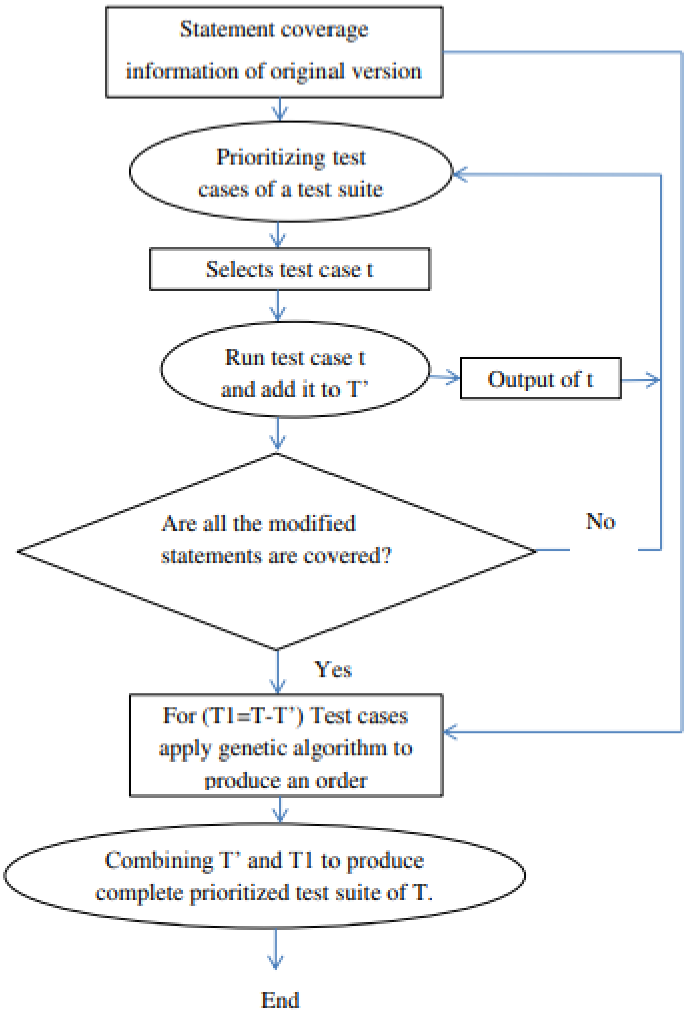
Siripong Roongruangsuwan, Jirapun Daengdej Software testing has been shown that testing, examination, and investigating costs usually use over the portion of the costs related to the headway of gigantic programming systems. Various authorities have found a couple of approaches to manage schedule a solicitation for test execution. Unfortunately, existing test prioritization frameworks are failing to sort out various test suites and examinations with the same need regards. Along these lines, those techniques are inefficient to arrange tests in the huge business structures. They erroneously schedule tests and the cost is attacked during the prioritization methodology. In this way, this paper proposes two new capable prioritization strategies to address the above issues. The central methodology expects to decide the issue of numerous tests doled out a comparative weight regards. The consequent procedure is made to sufficiently sort out various suites. Along these lines, this paper inspects an ability to spare high sort out tests in various suites while restricting a prioritization time.

III. PROPOSED ALGORITHM

The adaptive genetic approach is mainly proposed using an assumption that the output of a test case is expected to change whenever its targeted program statements are modified. The probability of fault detection increases with the increase in the rate of coverage of modifications. The proposed adaptive genetic approach uses the concept of adaptive approach for the test cases until all modified statements are covered to increase the rate of fault detection, and for the remaining test cases, it uses the concept of genetic algorithm for better statement coverage. The test case prioritization process based on adaptive genetic approach, shown in Figure 1, consists of following steps:

1. The criteria for the selection of first test case is same as given by the adaptive approach i.e. based on the statement coverage information of original version of software.
2. Calculate the fault detection capability of all the unselected test cases by considering only the potential of modified statements covered by test cases.
3. Run the test case having the maximum fault detection capability and record its output.
4. Repeat Steps 2 and 3 until all modified statements are covered at least once.
5. For the prioritization of remaining test cases, genetic algorithm is executed.

Below the proposed workflow is depicted below:-


Statement coverage information of original version
Prioritizing test cases of a test suite
Selects test case t
Run test case t and add it to T'
Output of t
Are all the modified statements are covered?
No
Yes
For (T1=T-T') Test cases apply genetic algorithm to produce an order
Combining T' and T1 to produce complete prioritized test suite of T.
End

. Figure 1: Proposed Workflow or Scenario

IV. CONCLUSION

This proposition displays that, the precise confirmation aggregated during this assessment has given a firm foundation to future assessments concerning using inherited estimations to sort out test suites that will be continued running in a period constrained execution condition using Genetic Algorithm based on fitness calculation. From the preliminaries in, we can construe that this method for prioritization can give a convincing mentioning of examinations for applications. Extended granularity in the wellbeing limit incited progressively effective examination prioritizations as well. Later on, we will improve the limit of these techniques by improving the ability to normally sort out different gigantic test suites with certifiable business data. In future work, these domains, we might want to outfit programming authorities with reasonable frameworks for improving testing structures through prioritization of analyses.

REFERENCES

[1] Ahmed, M. Shaheen and E. Kosba [2012], "SoftwareTesting Suite Prioritization using Multi-Criteria FitnessFunction".

[2] Anoj Kumar, Shailesh Tiwari, K. K. Mishra, “Generationof Efficient Test Data using Path Selection Strategy with Elitist GA in Regression Testing”, 978-1-4244-5540-9/10 ©2010 IEEE

[3] David Leon, Andy Podgurski [2003],"A Comparison ofCoverage-Based and Distribution-Based Techniques forFiltering and Prioritizing Test Cases".

[4] Emelie Engström, “Regression Test Selection and ProductLine System Testing”, 2010 Third InternationalConference on Software Testing, Verification andValidation 978-0-7695-3990-4/10 $26.00 © 2010 IEEE

[5] Elbaum, Malishevsky, Rothermel [2002],"Test caseprioritization: a family of empirical studies".

[6] F. I. Vokolos and P. G. Frankl. Pythia: a regression test selection tool based on textual differencing. In Proc. of the 3rd Int’l. Conf. on Rel., Quality & Safety of Softw.-Intensive Sys. (ENCRESS ’97), May 1997.

[7] Ghinwa Baradhi and Nashat Mansour[1997], “AComparative Study of Five Regression TestingAlgorithms”.

[8] G. Rothermel and M. Harrold. A safe, efficient regression test selection technique. ACM Trans. on Softw. Eng. and Methodology, 6(2):173–210, Apr. 1997.

[9] H. Leung and L. White. Insights Into Regression Testing. In Proc. of the Conf. on Softw. Maint., pages 60–69, Oct. 1989.

[10] Hyunsook Do, Member, Siavash Mirarab, “The Effectsof Time Constraints On Test Case Prioritization: A Seriesof Controlled Experiments”, IEEE Transactions On Software Engineering, Vol. 36, No. 5,September, 0098-5589/10/ 2010 IEEE.

[11] J. Munson and T. Khoshgoftaar. The detection of fault-prone programs. IEEE Trans. on Softw. Eng., pages 423–433, May 1992.

[12] J. Voas. PIE: A dynamic failure-based technique. IEEE Trans. on Softw. Eng., pages 717–727, Aug. 1992.

[13] King Chun Foo, Zhen Ming Jiang, Bram Adams,“Mining Performance Regression Testing Repositories forAutomated Performance Analysis”, 2010 10thInternational Conference on Quality Software, 1550-6002/10 © 2010 IEEE

[14] K. Onoma, W.-T. Tsai, M. Poonawala, and H. Suganuma. Regression testing in an industrial environment. Comm. of the ACM, 41(5):81–86, May 1988.

[15] L. Ott. An Introduction to Statistical Methods and Data Analysis. PWS-Kent Publishing Company, Boston, MA, third edition, 1988.

[16] L. You and Y. Lu[2012], “A Genetic Algorithm for theTime AwareRegression Testing Reduction Problem”.

[17] Md. Imrul[2011], “Test Case Prioritization for RegressionTesting Based on Fault Dependency”.