IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 3 Issue 8, August 2016
ISSN (Online) 2348 – 7968 | Impact Factor (2015) - 4.332
www.ijiset.com

An Hybrid Approach of Test-Case Prioritization: Review

Zubair Rashid Bhat

RN engineering&management
Department of CSE

Mudasir Ahmed Mutto

Assistant professor SSM college of
engineering &Technology

Abstract: Test-Case Prioritization is the method to schedule any execution order of the test with the purpose of maximizing some objects like revealing faults early. In this paper we have proposed the hybrid approach for the purpose of the test case prioritization involving Robust Genetic Algorithm to improve the parameters like APSC and execution time. This technique involves robust approach, parent generation, cross-over and mutation over each test-case and then calculates APSC and execution time.

Introduction: Software testing is one of the important steps in measuring the quality of the software system. As we know that most of the software development cost is for the purpose of its testing and maintenance so it has become a great issue for the developers to reduce the cost of the testing and maintenance.

Test case prioritization was first introduced in the regression testing whose purpose is to test the change in software during its evolution by reusing the test cases of its previous version that is before the modifications of it. In this approach for the purpose of regression testing test case prioritization maintains the schedule for the execution order of the test cases for maximizing some objects like revealing faults.

Test case prioritization is the technique for scheduling the test cases. With the help of scheduling these test cases, the effectiveness can be increased to achieve some performance goals. Test cases help in order to detect the faults in the system and the occurrence of fault rate. Test case prioritization is the expensive technique to be used, but also very necessary for validating and improving the quality of the software. Test case prioritization technique is used to prioritize the test cases in order to test the cases with higher priority than the cases having lower priority. Test cases are being prioritized for achieving cost effectiveness, time and efforts needed for the software system. Test cases must be prioritized in order to increase the fault detection and finding higher severity faults earlier. These test cases are being contained in a particular test suite. Test suite is thus the collection of the test cases.

Various test case prioritization techniques are named as under:-

Total Coverage Prioritization: - The technique to sort the various test cases by the total number of functions and statements covered. Test case prioritization arranges these test cases according to the number of transitions covered and executed.

Total Property Prioritization: - Total property prioritization sorts the test cases according to the number of properties these test cases are relevant to. Each test case is being prioritized on the property bases. The test cases are being tested for the cases that are affected by the property violation.

Additional Coverage Prioritization: - The technique in which the test cases having larger amount of coverage, are tested and executed first. But the time may be consumed in this technique.

Total FEP Prioritization: - The technique is named as Fault Exposing Potential. This technique is applied to arrange the test cases according to the maximum number of faults detected or exposed.

Mutation score technique is used in total FEP prioritization. Mutation score is the ratio of mutants. These mutants are distinguished from original program.

Additional FEP Prioritization: - The technique is similar to additional coverage prioritization. The test cases are being arranged according to the number of additional, but undetected mutants. FEP prioritization technique is more complex than the coverage based technique.

Optimal Prioritization: - This technique is based on the required information of the known mutants, which is not applicable as in practise. The faults are being detected with the help of minimum number of test cases.

Random Prioritization: - The random ordering of the test cases can be applied to test the test cases. Any ordering of the test cases can be determined to prioritize the test cases.

There are many software testing techniques like: Black-Box testing, white box testing, regression is testing.

Genetic Algorithm: Although test case prioritization using genetic algorithm gives satisfactory results, it can be quite time consuming at the same time. This is so because genetic algorithm is an iterative process which involves a population of chromosomes (test cases in this context) being repeatedly evolved to generate a better solution. During each iteration, the fitness function for each individual of the population is calculated andthe more fit individuals are selected for the next iteration. This process continues till the best fit chromosome is achieved. For very large population such as 100 test cases, it can take significant amount of time.

Regression testing

It is a type of testing aimed at finding out new errors after the modification of software. In other words, it assures that no additional errors were introduced in the process of fixing other problems and the software still works as it did before. However, executing the entire test suite for this purpose can be expensive in terms of cost and time. Therefore, it is essential to reduce the expense involved in regression testing. One way to achieve this is by test case prioritization. Test Case Prioritization aims to order the test cases so as to maximize the rate of fault detection. This topic has been a major subject of research for many years and many techniques have been proposed for achieving the same. Popular among these are genetic algorithm, ant colony optimization, bee colony optimization and particle swarm optimization.

Although these techniques are able to efficiently prioritize the test cases, but take significant amount of time to do so. For example in case of genetic algorithm, a large population of candidate solutions has to be repeatedly evolved in an iterative manner for reaching the best solution. When the population is large i.e. when there are some 100 or 1000 test cases, then genetic algorithm may consume a large proportion of time to prioritize the test cases. To remove this drawback, a new hybrid technique has been proposed which speeds up the time taken to prioritize the test cases. This hybrid technique is a combination of adaptive approach and genetic algorithm. The speedup is achieved by using an adaptive approach which schedules the test cases simultaneously during the execution of test cases.

Robust Genetic Algorithm is the Hybrid proposed test-case prioritization approach in this approach we ordering the test case and find the average percentage of statement coverage for hundred test cases in java . First we measure the APSC of Robust approach and ordering the test case.

In Robust approach we order the tpest case like until our statement not covers if test cases left or we can say failure test cases those are unable to

cover any statement its means the statement coverage is not done perfectly. We take that Left test cases after applying robust approach and perform genetic algorithm on these test case. In Genetic algorithm we apply three main techniques to order the test case like this our APSC improved as compared to robust approach.

We apply these techniques in genetic algorithm to giving the order to each test case

- Robust Approach
- Parent Generations
- Cross Over
- Mutations
- Measures APSC
- Execution time

LITERATURE REVIEW:

Maxim Buzdalov and Arina Buzdalova[1] this paper demonstrates a technique for adaptive selection of partner targets in transformative calculations, which was already connected to model issues just, is connected to era of experiments for programming challenge errands. The strategy depends on support learning. Tests demonstrate that the proposed strategy performs similarly all around contrasted with the best assistant destinations chose by hand.

L. L. Rosen[2] A standout amongst the most troublesome issue zones experienced in cutting edge flying frameworks is the confirmation by testing that the conveyed item meets the consented to and determined configuration necessities. This paper proposes an answer for that inquiry for those parameters, basic to the exactness of the framework, which have execution indicated by RMS values. The idea exhibited could, with legitimate scientific changes, be reached out to any arrangement of different parameters. The key thought is to indicate, in the acquirement detail, the criteria by which basic parameters will be checked. This paper shows a factual strategy, the aftereffects of which might be set in an acquisition determination, to build up a fair purchaser/vender hazard on parameters basic to framework execution. The method is intended to adjust the acknowledgment prerequisite to the genuine assembling process in a way which permits the merchant greatest adaptability without influencing the purchaser's danger. Methodology and illustration bend for execution are displayed. A method is displayed in this paper which utilizes outfit test information to refine beginning suspicions and has as its essential objective that the genuine purchaser and merchant dangers be as equivalent as could be expected under the circumstances at the finish of the creation contract. In this paper, the genuine purchaser's danger is . . . Given that frameworks with genuine execution in overabundance of the legally binding detail are made, what is the combined likelihood of these frameworks being acknowledged? In a like way, the genuine dealer's danger is . . . Given that frameworks with execution equivalent to or superior to the authoritative determination are made, what is the combined likelihood of these frameworks being rejected? As will be demonstrated later in this paper, the proposed test is reasonable to the merchant, since, over the long haul, he is ensured of offering a framework for each framework he assembles (tests) which has execution tantamount to or superior to anything authoritative necessities. The method is additionally reasonable to the purchaser since the troupe of frameworks he really purchases will perform superior to the determination prerequisite. The strategy displayed could likewise permit the control of other criteria. A standout amongst the most conspicuous of these other criteria would be affirmation that toward the culmination of the creation program, the RMS execution capacity of the troupe of conveyed frameworks is in any event on a par with the RMS execution necessity in the acquisition detail. Undoubtedly, there are numerous other such ideas. In any case, the author trusts that the idea of genuine, equivalent purchaser/dealer hazard has the most claim, since it has a tendency to stifle the purchaser's apprehension of "pooch frameworks" while

holding a reasonableness in yield proportion for the vender. In like manner, this paper will deliver itself just to the idea of fairness of purchaser and vender's danger as characterized previously. In the quest for this or whatever other idea including the outline of an acknowledgment test, it must be noticed that the acquisition determination requires execution

B. Narendra Kumar Rao; A. RamaMohan Reddy [3] This paper gives a diagram of the foundational issues identified with case-based thinking, depicting a portion of the main methodological methodologies inside the field, and embodying the present state through pointers to a few frameworks. The structure impacts the late techniques for learning level portrayals of wise frameworks. The strategies for case recovery reuse, arrangement testing, and learning are condensed, and acknowledgment is examined with few case frameworks that speak to various CBR approaches. Relapse testing happens amid the support phase of the product life cycle, be that as it may, it requires a lot of experiments to guarantee the fulfillment of a specific level of value. In this way, test suite sizes may become essentially. This paper concentrates essentially on use of CBR to test suite enhancement.

Dai Yue Ming; Wu Yi Ting; Wu Ding Hui [4] With a specific end goal to enhance the effectiveness and nature of programming experiment programmed era, a sort of molecule swarm improvement was proposed. It had adaptive streamlining taking into account the grouping thought. The calculation separated the populace into two sorts which were primary molecule and optional molecule when the calculation was executed. They utilized distinctive hunt procedures so that the calculation extended the pursuit extent of particles to accelerate the calculation running. The test result demonstrates that the proposed calculation has a greater number of focal points and is more powerful than the other contrastive calculations in the product experiment programmed era.

Dan Hao, Xu Zhao, Lu Zhang[5] Test-case prioritization is to plan the execution request of test cases to amplify some goal (e.g., uncovering deficiencies early). The current test-case prioritization approaches separate the procedure of test-case prioritization and the procedure of test-case execution by displaying the execution request of all test cases before software engineers begin running test cases. As the execution data of the altered project is not accessible for the current test-case prioritization approaches, these methodologies for the most part depend on just the execution data of the past system before alteration. To address this issue, we exhibit an adaptive test-case prioritization approach, which decides the execution request of test cases at the same time amid the execution of test cases. Specifically, the adaptive methodology chooses test cases taking into account their issue recognition capacity, which is computed in view of the yield of chose test cases. When a test case is chosen and runs, the shortcoming recognition ability of each unselected test case is altered by yield of the latest chose test case. To assess the viability of the proposed adaptive methodology, we directed a trial study on eight C projects and four Java programs. The exploratory results demonstrate that the adaptive methodology is normally essentially superior to the aggregate test-case prioritization methodology and aggressive to the extra test-case prioritization approach. Besides, the adaptive methodology is superior to the extra approach on a few subjects (e.g, supplant and plan).

Bo Jiang; W. K. Chan[6] This paper proposes a novel group of LBS strategies. They make adaptive tree-based randomized investigations with an adaptive randomized competitor test set system to differentiate the investigations among the branches of the investigation trees developed by the test inputs in the test suite. They dispose of the supposition on the chronicled connection of code scope between system variants. Our systems can be connected to programs with or with no past

forms, and thus are more broad than numerous current test case prioritization strategies. The exact study on four prevalent UNIX utility benchmarks demonstrates that, as far as APFD, our LBS procedures can be as successful as a portion of the best code scope based voracious prioritization methods ever proposed. We likewise demonstrate that they are essentially more productive and adaptable than the last methods.

References:

[1] Adaptive selection of helper-objectives for test case generation Maxim Buzdalov; Arina Buzdalova 2013 IEEE Congress on Evolutionary Computation Year: 2013 Pages: 2245 - 2250, DOI: 10.1109/CEC.2013.6557836 Cited by: Papers (4) IEEE Conference Publications

[2] An adaptive acceptance test design for an RMS parameter L. L. Rosen Adaptive Processes (9th) Decision and Control, 1970. 1970 IEEE Symposium on Year: 1970 Pages: 246 - 246, DOI: 10.1109/SAP.1970.270040 IEEE Conference Publications

[3] Case based reasoning approach for adaptive test suite optimization B. Narendra Kumar Rao; A. RamaMohan Reddy Computing Communication & Networking Technologies (ICCCNT), 2012 Third International Conference on Year: 2012 Pages: 1-5, DOI: 10.1109/ICCCNT.2012.6395870 IEEE Conference Publications

[4] Particle swarm optimization algorithm for test case automatic generation based on clustering thought Dai Yue Ming; Wu Yi Ting; Wu Ding Hui Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2015 IEEE International Conference on Year: 2015 Pages: 1479 1485, DOI: 10.1109/CYBER.2015.7288163 IEEE Conference Publications

[5] Adaptive Test-Case Prioritization Guided by Output Inspection Dan Hao, Xu Zhao, Lu Zhang Institute of Software, School of Electronics Engineering and Computer Science Key Laboratory of High Confidence Software Technologies (Peking University), Ministry of Education Peking University, Beijing, 100871, China

[6] Bypassing Code Coverage Approximation Limitations via Effective Input-Based Randomized Test Case Prioritization Bo Jiang; W. K. Chan Computer Software and Applications Conference (COMPSAC), 2013 IEEE 37th Annual Year: 2013 Pages: 190-199, DOI: 10.1109/COMPSAC.2013.33 Cited by: Papers (2) IEEE Conference Publications

[7] Solanki and Y. Singh, "Novel classification of test case prioritization techniques," International Journal of Computer Applications, 2014.

[8] A. Jatain and G. Sharma, "A systematic review of techniques for test case prioritization," International Journal of Computer Applciations, 2013.

[9] D. A. Kaur and S. Goyal, "A Systematic Review of Techniques for Test Case Prioritization," International Journal of Computer Applications, 2013.

[10] D. M. Hashini, "Clustering approach to test case prioritization using code coverage metric," International journal of engineering and computer science, 2014.

[11] J. Badwal and H. Raperia, "Test case prioritization using clustering," International journal of current engineering and technology, 2013.

[12] T. P. Jacob and D. T. Ravi, "Optimal regression test case prioritization using genetic algorithm," life Science Journal, 2013.

[13] M. Athar and I. Ahmad, "Maximize the code coverage for test suit by genetic algorithm," International journal of computer science and informatin technologies, vol. 5(1), pp. 431-435, 2014.

[14] M. M. B. Patel, "Test case prioiritzation for regression testing using ant colony optimization," International journal for scientific research & development, vol. 2, no. 04,2014, pp. 632-636, 2014.

[15] N. Sethi, S. Rani and P. Singh, "Ants optimization for minimal test case selection and prioritization as to reduce the cost of regression testing," International journal of computer applications, vol. 100, pp. 48-54, 2014.

[16] I. Sharma, J. Kaur and M. Sahni, "A test case prioritization approach in regression testing," International journal of computer science and mobile computing, vol. 3, no. 7, pp. 607-614, 2014.

[17] M. Shahid and S. Ibrahim, "An Evaluation of Test Coverage Tools in Software Testing," International Conference on Telecommunication Technology and Applications, vol. 5, no. 2011, pp. 216-222, 2011.

[18] P. R. Srivastava, "Test case prioritization," Journal of theoretical and applied information technology, 2008.

[19]Yuejian Li, Nancy J. Wahl, “An Overview of Regression Testing”, 1999 ACM SIGSOFT SoftwareEngineering Notes volume 24 no 1.

[20] S. Yoo, M. Harman, “Regression testing minimization, selection and prioritization: a survey”, 2010 Software Testing, Verification and Reliability.

[21] Gregg Rothermel, Roland H. Untch, Chengyun Chu, Mary Jean Harrold, “Test Case Prioritization: An Empirical Study” 1999 IEEE International Conference on Software Maintenance, (ICSM '99) Proceedings.

[22] HemaSrikanth, Laurie Williams, Jason Osborne, “System Test Case Prioritization of New and Regression Test Cases”

[18] Paolo Tonella, Paolo Avesani, Angelo Susi, “Using the Case-Based Ranking Methodology for Test Case Prioritization” 2006 22nd IEEE International Conference on Software Maintenance.