Airo International Research Journal
Volume XVI, ISSN: 2320-3714
April, 2018
Impact Factor 0.75 to 3.19
UGC Approval Number 63012

REGRESSION TEST CASE OPTIMIZATION USING HFFA ALGORITHM AND APFD EVALUATION METRICS

Ms. Kale Sanjivani
Research Scholar Mewar University, Chittorgarh, Rajasthan
Dr. S.N. Agal
Professor Mewar University, Chittorgarh, Rajasthan

ABSTRACT

Testing is executing a system in order to identify any gaps, errors or missing requirements in contrary to the actual desire or requirements. Testing can be defined as "A process of analyzing a software item to detect the differences between existing and required conditions and to evaluate the features of the software item". In software, with the target of finding faults, Software testing is a complete set of activities performed. It is one movement in the software development process planned at assessing a software item, such as system, subsystem and features against a specified set of system necessities. Software testing is a critical with size and complexity of today's software.

Keywords- Regression Testing, Software Testing, Prioritization, Optimization, Artificial Bee colony, Firefly, K-Means

INTRODUCTION

As defined earlier, testing can be defined as "A process of analyzing a software item which detect the differences between existing and required conditions (that is defects/errors/bugs) and evaluate the features of the software item". Software testing is a critical with size and complexity of today's software.

Regression testing is a type of software testing that seeks to uncover new software bugs in areas of a system after changes have been made to them. Whenever a change in a software application is made it is quite possible that other areas within the application have been affected by this change. To verify that a fixed bug hasn't resulted in another functionality or business rule violation I use regression testing. The intent of Regression testing is to ensure that a change, such as a bug fix did not result in another fault being uncovered in the application.

In the presented software, Regression testing performs a very important role while changes happened. During regression testing the modified parts of the system are first tested. Then the whole system needs to be retested using the old test suite to have confidence that the modifications did not introduce new faults into the system. Novel test cases are not produced during Regression testing however formerly created test cases are re-executed. The simplest method is to run all test cases for validating the adapted program.

Test case selection and Test case prioritization are the two major part of Regression Testing. Test case selection works out the problem of choosing the test case which will be helpful to check the adapted part of the software. When the test cases grow in number due to the growth of applications, it turns into very hard to implement all the test cases inside a particular amount of time.

Test case prioritization is significant in regression testing. Test case prioritization techniques are scheduled over test cases in an order to improves the performance of regression testing. Test case prioritization technique schedules the test cases for execution so that the test cases with higher priority executed before lower priority. The objective of test case prioritization is to detect fault as early as possible

LITERATURE SURVEY

Several techniques were proposed by various authors for Regression Testing and a few of them are explained below:

Hyunsook Do *et al* [16] have proposed a series of experiments to assess the effects of time constraints on the costs and benefits of prioritization techniques. Their first experiment has manipulated time constraint levels and also they have shown that time constraints do play a significant role in determining both the cost-effectiveness of prioritization and the relative cost-benefit trade-offs among techniques. Their second experiment has replicated the first experiment, controlling for several threats to validity including numbers of faults present, and also shown that the results generalize to this wider context. Their third experiment has manipulated the number of faults presented in programs to examine the effects of faultiness levels on prioritization and shown that faultiness level affects the relative cost-effectiveness of prioritization techniques. Those results have several implications for test engineers wishing to cost-effectively regression test their software systems.

Md. Imrul Kayes *et al* [17] have proposed a new metric for assessing rate of fault dependency detection and an algorithm to prioritize test cases. Using the new metric the effectiveness of that prioritization was shown comparing it with non-prioritized test case. They have also proposed a metric to measure effectiveness of test case prioritization in regression testing and a prioritization technique which can be used to improve the rate of dependency detection for regression testing. Analysis was done for prioritized and non-prioritized test cases with the help of proposed metric. The metric proposed has considered fault severity and test case execution time to be uniform. Analysis has proved that prioritized test cases are more effective in detecting dependency among faults.

Hong Mei *et al* [18] have proposed an approach to prioritizing test cases in the absence of coverage information that operates on Java programs tested under the JUnit framework. Their results have shown that the test suites constructed by JUnit test case Prioritization Techniques operating in the Absence of coverage information (JUPTA), were more effective than those in random and untreated test orders in terms of fault-detection effectiveness. Although the test suites constructed by dynamic coverage based techniques have retained fault-detection effectiveness advantages, the fault-detection effectiveness of the test suites constructed by JUPTA was close to that of the test suites constructed by those techniques, and the fault-detection effectiveness of the test suites constructed by some of JUPTA's variants was better than that of the test suites constructed by several of those techniques.

Siavash Mirarab *et al* [19] have proposed an approach for selecting and ordering a predetermined number of test cases from an existing test suite. Their approach has formed an Integer Linear Programming problem using two different coverage-based criteria, and has used constraint relaxation to find many close-to-optimal solution points. Those points were then combined to obtain a final solution using

a voting mechanism. The selected subset of test cases was then prioritized using a greedy algorithm that maximizes minimum coverage in an iterative manner. Their proposed approach has been evaluated and the results shown significant improvements over existing approach for some cases and comparable results for the rest. Their approach has also provided more consistency compared to existing approaches.

ZHANG Zhi-hua *et al* [20] have proposed a set of prioritization algorithms. They have also explained a new exploration for regression testing prioritization technique which oriented function call path. Static paths on function call obtained by analyzing the source code, combined with the dynamic path after executing test cases, the correspondence was built between test cases and the static paths, identifying the changes which software developers modify program to correct defects, given different priority to test case based on path coverage, test cases were selected in accordance with their priorities in regression testing. This set of prioritization algorithms has improved the efficiency of regression testing and guarantee testing adequacy, because only the modified and affected parts of software were tested.

Sreedevi Sampath *et al* [21] have proposed the notion of combining multiple criteria into a hybrid. Their goal was to create a uniform representation of such combinations so that they could be described unambiguously and shared among researchers. They have precisely formulated three hybrid combinations Rank, Merge, Choice and demonstrated their usefulness in two ways. Their findings have suggested that hybrid criteria of others could be described using their Merge and Rank formulations. The hybrid criteria they developed most often outperformed their constituent individual criteria. Finally, they have anticipated that the framework provides a step toward helping researchers to create shared tools and artifacts that use a uniform representation.

Ke Zhai *et al* [22] have proposed a suite of metrics and initialized them to demonstrate input-guided techniques and point-of-interest (POI) aware test case prioritization techniques, differing by whether the location information in the expected outputs of test cases was used. It was reported a case study on a state full LBS-enabled service. The case study has shown that the POI-aware techniques could be more effective and more stable than the baseline, which reorders test cases randomly, and the input-guided techniques. They have also found that one of the POI-aware techniques, cdist, was either the most effective or the second most effective technique among all the studied techniques in my evaluated aspects, although no technique excels in all studied SOA fault classes.

PROPOSED METHODOLOGY

The test case prioritization problem has newly engaged in scheduling test cases for regression testing in an order that raises their efficiency of performance goal. In regression testing this is unsuccessful to re-execute all the test case. Prioritization methods order the test cases for regression testing using information attained from earlier test case execution. In Regression Testing I have proposed to suggest an approach to prioritize test cases. In order to accomplish efficient Test cases, I shall improve a prioritization technique based on fire fly algorithm.

1. Regression Test Case Generation

The principal objective underlying my approach will be to produce efficient test cases. In this phase, test cases will be produced for the input case study. The test case generation is one of the significant steps

applied to find the most important test cases in the study.

2. Factors Identification

In this phase , some factors will be regarded in order to prioritize the test cases. The factors exploited here will be time, trace events, behavioral dependency and responsibility. In order to make out the prioritized test cases, these factors will be used

3. Clustering for Test Case Prioritization

The K-means clustering algorithm will be applied to divide the related test cases from unrelated test cases in this research. Relevant test cases indicate the prioritized test cases which will be in this phase.

4. Optimization of Prioritized Test Cases

Finally in last phase, in order to optimize it with Hybrid Fire Fly Algorithm (HFFA) I will reflect on only this related test cases resultant from the clustering algorithm.

The firefly algorithm is a meta-heuristic algorithm; it is excellent in flashing behavior of fireflies. The unique work for a firefly's flash is to perform as a signal system to draw other fireflies. HFFA efforts by the hybridization of Artificial Bee Colony (ABC) Algorithm and Fire Fly (FF) Algorithm. ABC algorithm is a swarm based meta-heuristic algorithm which is motivated by the sharp foraging behavior of the honey bees. It contains three components namely, employed bees, onlooker bees and scout bees. The number of food sources indicates the probable solutions of optimization problem and the nectar amount of a food source indicates the quality of the solution. The FF will be practiced inside the scout bee component of ABC in my HFFA Optimization Algorithm, which leads to rapid convergence and restricted search space controlled based optimization. As a result I will get Effective prioritized test cases.

My approach will be implemented on Java program with a hospital management case study as an input and the presentation will be assessed with different evaluation metrics. For assessing the superiority Average of the Percentage of Faults Detected (APFD) will be the major evaluation metric applied. Performing the test cases based on my prioritization algorithm will significantly give efficient test cases. The suggested technique will be competent in prioritizing the regression test cases. The implementation will be done in JAVA.

CONCLUSION

A hybrid firefly-ABC based prioritization with four phase's test case generation, factors identification, test case prioritization and test case optimization. The test case generation techniques aim to generate test cases which maximize cover for each scenario. From that test case I am performing factors identification, the factors utilized here will be time, trace events, behavioral dependency and responsibility. These factors will be employed to identify the prioritized test cases. Then the test cases are prioritized by using a hybrid firefly-ABC technique. The evaluation measures of APFD were evaluated for our proposed method. The test case prioritization time and taken memory space are discussed and is very high by presenting very good outcomes and also the prioritization of test case is gives very accurate outcomes. From the outcomes, I have showed that the hybrid firefly-ABC utilized in my proposed work outperforms the other classifiers by facilitated very good accuracy. Thus, I have observe that my proposed work is better than other existing works for the regression test case prioritization

REFERENCES

[1]. Siripong Roongruangsuwan and Jirapun Daengdej,"Test Case Prioritization Techniques", In proceeding of the Journal of Theoretical and Applied Information Technology, pp. 45-60, 2010.

[2]. Sanjukta Mohanty, Arup Abhinna Acharya and Durga Prasad Mohapatra, "A Survey on Model Based Test Case Prioritization", In proceeding of IEEE International Journal of Computer Science and Information Technologies, Vol. 2, No. 3, pp. 1042-1047, 2011.

[3]. Xiaofang Zhang, Changhai Nie, Baowen Xu and Bo Qu, "Test Case Prioritization based on Varying Testing Requirement Priorities and Test Case Costs", In proceeding of IEEE International Conference on Quality Software, pp. 15-24, Oct 2007.

[4]. Praveen Rajan Srivastava, "Test Case Prioritization", In proceeding of the Journal of Theoretical and Applied Information Technology, pp. 178-181, 2008.

[5]. Lin chen, Ziyuan Wang, Lei Xu, Hongmin Lu and Baowen Xu, "Test Case Prioritization for Web Service Regression Testing", In proceeding of IEEE International Symposium on Service Oriented System Engineering, pp. 173-178, Jun 2010.

[6]. Ruchika Malhotra, Arvinder Kaur and Yogesh Singh, "A Regression Test Selection and Prioritization Technique", In proceeding of the Journal of Information Processing Systems, Vol. 6, No. 2, pp. 235-252, June 2010.

[7]. Albert Pravin and Subramaniam Srinivasan, "Effective Test Case Selection and Prioritization in Regression Testing", In proceeding of the Journal of Computer Science, Vol. 9, No. 5, pp. 654-659, 2013.

[8]. Lijun Mei, Zhenyu Zhang, W.K.Chan and T.H.Tse, "Test Case Prioritization for Regression Testing of Service-Oriented Business Applications", In proceeding of the Journal of Service Oriented Development, pp. 901-910, 2009.

[9]. Aravind Kumar Upadhyay and A.K.Misra, "Prioritizing Test Suites Using Clustering Approach in Software Testing", In proceeding of IEEE International Journal of Soft Computing and Engineering, Vol. 2, No. 4, pp. 222-226, Sep 2012.

[10]. Thillaikarasi Muthusamy and Dr. Seetharaman.K, "A Test Case Prioritization Method with Weight Factors in Regression Testing Based on Measurement Metrics", In proceeding of IEEE International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 3, No. 12, pp. 390-396, Dec 2013.

[11]. Amandeep Kaur and Sheena Singh, "A New Technique for Test Case Prioritization", In proceeding of IEEE International Journal of Computer Science and Mobile Computing, Vol. 2, No. 9, pp. 107-109, Sep 2013.

[12]. Deepak Garg and Amitava Datta, "Test Case Prioritization due to Database Changes in Web Applications", In proceeding of IEEE International Conference on Software Testing, Verification and Validation, pp. 726-730, April 2012.

[13]. Adipat Laeprattanakul and Taratip Suwannasart, "An Approach for Regression Test Case Selection using Object Dependency Graph", In proceeding of IEEE International Conference on Intelligent Networking and Collaborative Systems, pp. 617-621, 2013.

[14]. Zheng Li, Mark Harman and Robert M.Hierons, "Search Algorithms for Regression Test Case Prioritization", IEEE Transactions on Software Engineering, Vol. 33, No. 4, pp. 225-237, April 2007.

[15]. Hyunsook Do, Siavash Mirarab, Ladan Tahvildari and Gregg Rothermel,

"The Effects of Time Constraints on Test Case Prioritization: A Series of Controlled Experiments", IEEE Transaction on Software Engineering, Vol. 36, No. 5, pp. 593-617, Sep 2010.

[16]. Md. Imrul Kayes, "Test Case Prioritization for Regression Testing Based on Fault Dependency", In Proceeding of IEEE International Conference on Electronics Computer Technology, Vol.5, pp. 48-52, Apr 2011.

[17]. Hong Mei, Dan Hao, Lingming Zhang, Lu Zhang, Ji Zhou and Gregg Rothermel, "A Static Approach to Prioritizing JUnit Test Cases", IEEE Transaction on Software Engineering, Vol. 38, No. 6, pp. 1258-1275, Nov 2012.

[18]. Siavash Mirarab, Soroush Akhlaghi and Ladan Tahvildari, "Size-Constrained Regression Test Case Selection Using Multicriteria Optimization", IEEE Transaction on Software Engineering, Vol. 38, No. 4, pp. 936-956, Jul 2012.

[19]. ZHANG Zhi-hua, MU Yong-min and TIAN Ying-ai, "Test Case Prioritization for Regression Testing Based on Function Call Path", In Proceeding of IEEE International Conference on Computational and Information Sciences, pp. 1372-1375, Aug 2012.

[20]. Sreedevi Sampath, Renee Bryce and Atif M. Memon, "A Uniform Representation of Hybrid Criteria for Regression Testing", IEEE Transaction on Software Engineering, Vol. 39, No. 10, pp. 1326-1344, Oct 2013.

[21]. Ke Zhai, Bo Jiang and W.K. Chan, "Prioritizing Test Cases for Regression Testing of Location-Based Services: Metrics, Techniques, and Case Study", Transaction on Service Computing, Vol. 7, No. 1, pp. 54-67, Jan 2014.