

A Hybrid Approach for Test Case Prioritization using PSO Based on Software Quality Metrics

K. Senthil Kumar^{1*}, A. Muthukumaravel²

¹Research Scholar, Bharathiar University, Coimbatore

¹Assistant Professor, Department of Information Technology, SRM Institute of Science and Technology, Chennai

²Professor & Head, Department of MCA, Bharath University, Chennai

*Corresponding Author Email: ¹senthil.ku@ktr.srmuniv.ac.in

Abstract

Effective functionality checking of any software application is the crucial event that determines the quality of outcome obtained. Generally, checking scenarios that involves multiple test cases in mixture with multiple components is time consuming and also increases the quality assurance cost. Selection of suitable method/approach for optimization and prioritization of test cases as well as appropriate evaluation of the application would result in reduction of fault detection effort without appreciable information loss and further would also significantly decrease the clearing up cost. In the proposed method, test cases are optimized and then prioritized by Particle Swarm Optimization algorithm (PSO) and Improved Cuckoo Search algorithm (ICSA), respectively. Finally, the result will be evaluated for software quality measures.

Keywords: Regression Testing, Test Case Prioritization, Test Case Optimization, Particle Swarm Optimization, Cuckoo Search, Software metrics, Quantitative measure.

1. Introduction

For attaining high quality software, Software testing is preferred to be one of the main and principal technique. Software testing is prepared to identify presence of errors, which cause software failure [14]. Test case prioritization is an efficient and practical technique of regression testing. It is useful to increase the competence of regression testing by sorting and implementing test cases according to their significance [5]. Test case prioritization has also been employed to decrease the quality assurance cost and for minimizing the fault detection effort [2]. Majority of the techniques for test case prioritization are based on particular coverage criteria. That is based on particular performance goal such as fault coverage, statement coverage, path coverage, branch coverage, and function coverage [3] the test cases are prioritized. Regression testing is the process of verifying modifications brought in a system during software maintenance, i.e., after it was changed and interfaced with other software. It is observed as an expensive, still an imperative process as the size of test suite is very huge, retesting of system consumes huge amount of time and computing resources [6]. In regression testing the set of tests should be selected according to priority. The test cases are selected in a regression test suite in order that the test case maximizes particular objectives which helps in reducing the time and cost needed to maintain service-oriented business applications. The regression testing techniques formulated for such applications concentrate on testing the individual services or workflow programs [9]. Regression test prioritization is the ordering and selection of the test cases either in an growing or diminishing order to achieve several presentation goal. Different criteria can be used to prioritize the regression test set to achieve the presentation goal [13].

The activity carried out inside dissimilar phases are experimenting and validating the part of code. Test Suite Selection, Test Case Prioritization, Test case reduction are considered to be the Tasks of regression testing which guarantees that no intended fault is generated while the code is changed[12]. For example, a software development method which is comprising of components mostly employs the components of black-box, frequently admitted from a third party. As the internal functionality of third party components are unfamiliar to the users, when any changes are produced in those components may be accepted with software system [11]. While the significance of incorporating requirements information during the testing phase has been well known by the requirements engineering community only some researchers have studied the employ of requirements with software testing (black-box testing) [8]. To build the rate of deficiency location experiment prioritization is the process of requesting the implementation of experiments. Expanding the rate of fault discovery can be more input to framework engineers, improving obligation create action and, eventual, programming conveyance. Several presented experiment prioritization procedures admit that tests can be carried out in any request [10]. Employing code coverage information as substitute's coverage based test case prioritization reorganizes test case in ordered to exploit code coverage as early as feasible. On the other hand code coverage itself is not an ample criterion to guarantee the accomplishment of high rate of faults detection [4]. Several test case prioritization techniques have been suggested and most are based mainly on structural coverage metrics such as branch or statement coverage [1]. For test case prioritization, different algorithms such as greedy, additional greedy, Additional Greedy Algorithm, hill climbing, Heuristic Algorithm, particle swarm optimization and Genetic Algorithm can be used. Each algorithm is characterized by their own functionality. The algorithms will vary in terms of the presentation [15].

Particle Swarm Optimization (PSO):

The Particle Swarm Optimization is a population based global optimization algorithm which makes use of heuristic search. The PSO was developed by John Kennedy and Eberhart, in 1995. The social behavior of the group of animals, flock of birds and fishes were the inspiration for the development of this algorithm. Each individual follows the path of the social best particle within its environment e.g. in ocean, during the search of its food source, a school of fish. Every individual fish is observing it's the position and velocity of its neighbors. The values are compared with global best position and velocity. The updates are made by individuals in their position and velocity after choosing the best position and velocity. Hence, each fish come closer towards the best position once its velocity has been modified. It helps each individual to progress faster towards the food.

Cuckoo Search Algorithm (CSA):

CSA is meta heuristic algorithm which was proposed by observing the breeding behavior of the cuckoo and alleviates to implement. This algorithm considers a number of nests in cuckoo search. Each egg in the nest is treated as a solution and an egg of cuckoo indicates a fresh solution. The poorest solution in the nest is replaced by the fresh and better solution. The subsequent representation scheme is selected by Cuckoo Search algorithm: Each egg in the nest indicates a solution. A Cuckoo egg indicates a novel solution. The plan here is that the not-so-good egg of Cuckoo is substituted by the novel and probably better egg. On the other hand, in addition to the fundamental case of considering one cuckoo per nest, the approach can be extended by including the property that more than one egg can be available in each nest, which indicates a set of solutions.

2. Literature Survey

Hong Me et al [16] have proposed an approach to prioritize test cases in situations where the coverage information is not available. The proposed approach functions on Java program which is tested using popular class of systems which is named as the JUnit framework. It is popularly called as JUPTA, The JUnit test case Prioritization Techniques operating in the Absence of coverage information. It does the job by analyzing the static call graphs of all the JUnit test cases and also the program which is tested. Thus the capability of each and every test case to accomplish code coverage is estimated. The test cases are arranged according to the estimates. The efficiency of JUPTA is evaluated by conducting a practical research on 19 different descriptions of 4 Java programs with a range of 2K-80K program lines. Different versions of JUPTA are compared with 3 different control techniques, and various dynamic approaches to prioritize coverage-based test cases. The capabilities of the methods to enhance the fault detection rate of test suites are assessed by the comparison.

Bestoun S. Ahmed et al [17] have recommended an approach for User Interface functional testing using Simplified Swarm Optimization (SSO). An optimized test suite is generated by Simplified Swarm Optimization with the help of Event-Interaction Graph (EIG). The approach also handles and repairs the test suites by excluding the needless event sequences. This generation algorithm which is based on SSO is more effective than other algorithms. This approach was applied on standard case study and proved that it can be capable of handling applications in reality.

Ke Zhai et al [18] have suggested a group of metrics and he initialized them to exhibit input-guided techniques and point-of-interest (POI) aware test case prioritization techniques, which are varying if the locality information is used in the predictable output of test cases. It results in a elaborative study on a state full (location-based service) LBS-enabled services, which

demonstrates a comparative study of Point of Interest techniques with the baseline and the input-guided techniques. In the baseline techniques test cases are often reordered arbitrarily. The observation stated that 'cdist', one of the popular Point of Interest-aware technique found to be either the most efficient compared to all other techniques in the evaluated aspects of their evaluation.

Sreedevi Sampath et al [19] have suggested a formalized notion by merging several criterion as a hybrid. The objective was to formulate a uniform representation by merging several criteria. Uniform representation enables unambiguous description and sharing among researchers. They predict that the sharing of uniform representation will permit the researchers to study, implement, evaluate, and extend the hybrids with the use of a general suite of tools and techniques. Three various combinations of hybrid known as Rank, Choice and Merge are formulated. The usefulness of the combinations is exhibited in two different ways. Firstly, they focused on recast, which is in respect of their formulations; Second they made use of earlier outcomes of test case prioritization to generate and evaluate the new hybrid criteria. They found that the hybrid criteria proposed by other researchers can also be illustrated with the use of the formulations namely Merge and Rank, and that the hybrid criterion proposed by them surpassed the essential personal criteria, most frequently.

Ashima Singh [20] has suggested a time overhead-based approach to prioritize the test cases. The approach can be used for ordering or rearranging the test cases and test case reduction using an intelligent dynamic approach. Using this approach, the test cases are generated based on their priorities, which are allotted on account of intelligent operations. The efficiency of the new test case orderings is determined using a cumulative mutation probability (CMP) metric.

Ahlam Shakeel Ahmed Ansari et al [21] have suggested a method to test a quality product, decreasing time and cost for the regression testing. They shows interest in reducing test suites during rigorous regression test on a product so that the same quality product should be produced as the original test suites would produce. Two phase test suite refinement approach is proposed. Test Suite Minimization and Test Suite Prioritization are the two phases of the proposed approach. The Specification based minimization technique was used in Test Suite Minimization and the risk based prioritization technique was used in Test Suite Prioritization. Thus using this approach, quality product can be tested with decreased time and cost for regression testing, as this approach characterized with the generation of reduced test suites which performs as equivalent as the original test suites.

Gurinder Singh et al [22] have proposed an test case reduction hybrid technique on the basis of the Genetic algorithms(GA) and the Ant colony optimization (ACO). Genetic Algorithm was an evolutionary algorithms (EA), wherein the results to optimization problems which use the techniques motivated through natural evolution, namely inheritance, crossover, selection and mutation. Ant Colony Optimization was a swarm intelligence algorithm. ACO adopts the ants' behavior to find solution for a given problem. It is capable of providing the optimum results in minimal time. Thus it is proved to be an optimistic technique.

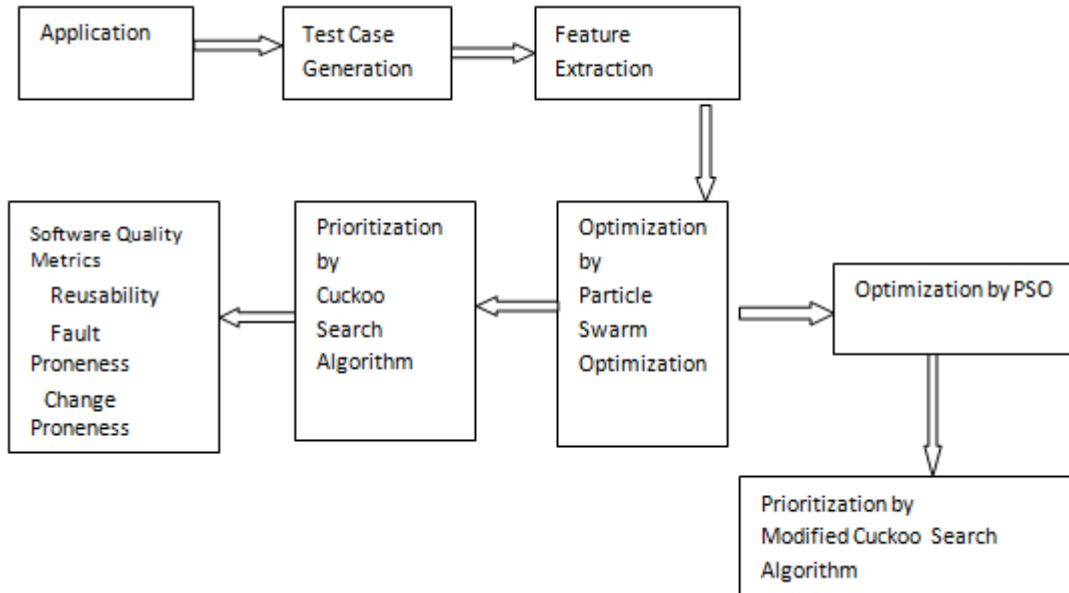
3. Limitations of Existing Systems

- Physically examining the test results can be extremely time consuming for test suites.
- A breakdown of publishing error may take place if the explanation of the service is wrong or the consumption is problematic.
- To decrease the quality assurance cost which comprises both the testing and debugging cost
- Checking scenarios are made difficult if multiple test cases are employed in mixture with multiple components.

- Number of errors identified during non-overlapping intervals is independent of each other.
- Alteration probabilities affect the system dependability estimations drastically.
- Decreasing fault detection effort may lead to the loss of information, which in turn leads to raise of the clearing up cost.

4. Proposed Methodology

The main aim of this paper is to achieve high quality software by using a hybrid approach to prioritize test case. In this approach,



5. Conclusion

ACO, Genetic Algorithm etc in terms of effectiveness and efficiency. The Cuckoo Search Algorithm is more efficient in finding the best solution, So the combination of Particle Swarm Optimization (PSO) and Cuckoo Search Algorithm (CSA) will produce highly Optimized and Prioritized Test Cases which will improve the quality of the Software.

References

- [1] Matt Staats, Pablo Loyola, Gregg Rothermel, "Oracle-Centric Test Case Prioritization", In proceeding of IEEE International Symposium on Software Reliability Engineering, pp.311-320,2012.
- [2] E. Ashraf, A. Rauf, and K. Mahmood, "Value based Regression Test Case Prioritization", Proceedings of the World Congress on Engineering and Computer Science, vol.1, 2012.
- [3] Prakash N, Rangaswamy T.R, "Weighted Method For Coverage Based Test Case Prioritization", Journal of Theoretical and Applied Information Technology, vol.56,no.2,pp.235-243,2013.
- [4] Kun Wu, Chunrong Fang, Zhenyu Chen, and Zhihong Zhao, "Test case prioritization incorporation ordered sequence of program elements", In proceeding of IEEE International Workshop on Automation of Software Test, pp.124-130, 2012.
- [5] ZHANG Zhi-hua MU Yong-min TIAN Ying-a, "Test Case Prioritization for Regression Testing Based on Function Call Path", IEEE International Conference on Computational and Information Sciences, pp.1372-1375, 2012.
- [6] Amr AbdelFatah Ahmed, Dr. Mohamed Shaheen, and Dr. Essam Kosba, "Software Testing Suite Prioritization Using Multi criteria Fitness Function", IEEE International Conference on Computer Theory and Applications,pp.160-166, 2012.
- [7] Daniel Di Nardo, Nadia Alshahwan, Lionel Briand, AND Yvan Labiche, "Coverage Based Test Case Prioritization: An Industrial Case Study", In proceeding of IEEE International Conference on Software Testing, Verification and Validation,pp.302-311,2013.
- [8] Md. Junaid Arafeen and Hyunsook Do, "Test Case Prioritization Using Requirements-Based Clustering", In proceeding of IEEE International Conference on Software Testing, Verification and Validation, pp.312-321,2013.
- [9] S. Raju, and G. V. Uma, "Factors Oriented Test Case Prioritization Technique in Regression Testing using Genetic Algorithm", European Journal of Scientific Research, vol.74, no.3, pp.389-402, 2012.
- [10] P.Bharath Kumar C.Lakshminatha Reddy, and V.Surendra Gupta, "Creating a Test Case Prioritization Technique Using Reliance Estimation of Functional Requirement", IEEE International Journal of Engineering Trends and Technology, vol.18, no.2, pp.88-92, 2014.
- [11] Albert Pravin and Subramaniam Srinivasan, "Effective Test Case Selection And Prioritization in Regression Testing", Journal of Computer Science, vol.9, no.5, pp.654-659, 2013.
- [12] Nitika Sharma and Neha Malhotra, "Regression Testing Prioritization, Selection and Reduction using Hybrid Criteria", In proceeding of IEEE International Journal of Computer Applications, vol.95,no.7,pp.38-46,2014.
- [13] Neha Sethi, Shaveta Rani, and Paramjeet Singh, "Ants Optimization for Minimal Test Case Selection and Prioritization as to Reduce the Cost of Regression Testing", IEEE International Journal of Computer Applications, vol.100,no.17,pp.48-54,2014.
- [14] Bansal, Priyanka. "A critical review on test case prioritization and Optimization using soft computing techniques."In proceeding of IEEE International Conference on Role of Technology in Nation Building, pp.74-77, 2013.
- [15] Pravin, A., and S. Srinivasan. "An efficient algorithm for reducing the test cases which is used for performing regression testing." Proceedings of IEEE International Conference on Computational Techniques and Artificial Intelligence, pp.194-197, 2013.
- [16] Hong Mei, Dan Hao, Lingming Zhang,Lu Zhang, Ji Zhou, and Gregg Rothermel, "A Static Approach to Prioritizing JUnit Test Cases", IEEE transactions on software engineering, vol. 38, no. 6, pp.1258-1275,2012.
- [17] Bestoun S. Ahmed, Mouayad A. Sahib, Moayad Y. Potrus, "Generating combinatorial test cases using Simplified Swarm Optimization (SSO) algorithm for automated GUI functional testing", Engineering Science and Technology, an International Journal, vol.17, pp.218-226, 2014.

- [18] Ke Zhai, Bo Jiang, and W.K. Chan, "Prioritizing Test Cases for Regression Testing of Location-Based Services: Metrics, Techniques, and Case Study", *IEEE transactions on services computing*, vol., 7no.1, pp.54-67, 2014.
- [19] Sreedevi Sampath, Rene 'e Bryce and Atif M. Memon, "A Uniform Representation of Hybrid Criteria for Regression Testing", *IEEE transactions on software engineering*, vol.39,no.10,pp.1326-1344,2013.
- [20] Ashima Singh "Prioritizing Test Cases in Regression testing using Fault Based Analysis", *IEEE International Journal of Computer Science Issues*, Vol. 9, NO.6, PP.414-420, 2012.
- [21] Ahlam Shakeel Ahmed Ansari, Prof. K. K. Devadkar, Dr. Prachi Gharpure, "An optimized technique for test suite refinement in regression test", *IEEE International Journal of Advancements in Research & Technology*, vol.2,no.7,pp.263-266,2013.
- [22] Gurinder Singh, Dinesh Gupta "An Integrated Approach to Test Suite Selection Using ACO and Genetic Algorithm", *International Journal of Advanced Research in Computer Science and Software Engineering* vol.3,no.6,pp.1770-1778,2013.