

A Study on Different Approaches for Regression Test Selection

T. K. Akila¹, Dr. A. Malathi²

¹Ph.D Research Scholar, Department of Computer Science, Government Arts College, Coimbatore, Tamil Nadu, India

²Assistant Professor, Department of Computer Science, Government Arts College, Coimbatore, Tamil Nadu, India

Abstract: *Software testing plays a major role in the field of software cycle. In which, Regression testing covers an important phase as it is very essential after every new modification done to the software as it helps to ensure the stability of the system not being affected by the new change. In this paper, regression testing, the need for it and the different ways of carrying out the same are discussed.*

Keywords: Regression Test, Test case, Test suite, Genetic Algorithm, Ant Colony Optimization, Bee Colony Optimization

1. Introduction

Software testing is a process of ensuring the software being developed to meet the requirement of the clients. This help to detect the faults that end up in failures and enable the developers to fix the faults before it would create a greater impact on the system. The process of Regression testing is an important testing phase that needs to be mandatorily carried out to ensure the quality of the software. For this, a considerable amount of test cases is required to identify the faults and ensure the system stability. But it is impossible for the testers to check each and every functionality of the software after each modification or enhancement being done by the development team as they will have to run the entire set of test cases of the software in addition to the new set of test cases of the newly added functionality as well. This makes the testing team a tedious task. In order to overcome this difficulty, a technique called test case minimization or test case prioritization is followed so that the redundant test cases are ignored and important test cases are prioritized based on different criteria. There are different algorithms that fulfill this task to some extent. In this paper, three different algorithms called Genetic Algorithm (GA), Ant Colony Optimization algorithm (ACO) and Bee Colony Optimization algorithm(BCO) to be discussed on how they help in minimizing the test cases.

2. Regression Test

A method of testing the software system to confirm that the recent changes done to the software has not disturbed or created new issues to the existing software system. When a bug is identified and fixed by the developer, regression test will check the current functionalities of the software along with the new functionality after the bug fix so that, the dependant functionalities have no effect on it. Either the entire software can be retested to ensure the quality or only a part of the software along with the dependencies be run. This test is triggered after every code change to confirm that it still complies [8] with the specifications. The way of proceeding with the regression test is a kind of art as many issues logged in, are because of the side effects of the last minute bug clearance. In most of the cases, [9] this happen to be an expensive activity. So an effective way of carrying

out this will be very beneficial with respect to cost, time and effort.

2.1 Test Case

IEEE Standard 610(1990) defines test case as follows: “A set of test inputs, execution conditions and expected results developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement” [5].

The set of conditions based on which the tester confirms that the software is working fine is called a test case. It has components like input, event and expected result. A bunch of test cases is called a test suite. The test cases of similar type are put under one test suite. Immediately after the preparation of the test plan, the test cases are written by the testers based on the requirement specified by the customers. A test plan may have any number test suites based on the software size and the test scenarios. A good test case has to be written by keeping the end users in mind on what are the possible ways of using the application by them.

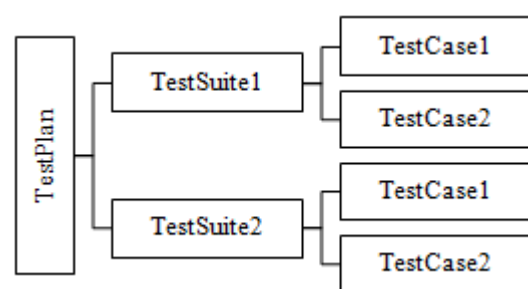


Figure 1: Components for testing

2.2 Test Case Prioritization

The test case prioritization problem as follows – [7] “Let T, be the test suite, PT be the set of all possible permutations of test in T, f be the function from PT, then the test case prioritization problem is defined as finding $T' \in PT$ such that $(\forall T'') (T'' \in PT) (T'' \neq T') [(f(T') \geq f(T''))]$.”

Test case prioritization schedules the test cases based on certain priority put in place. Such priority could be either on the relationship, type or the faults they detect. This helps to

allow the most important test cases to run first so that the defects can be caught earlier. There might be some test cases that have similar properties or possess similar fault detection capability. So, keeping in mind the relationship between the test and by having some knowledge on the test flow, it would much help to improve the process of testing. There have been many techniques followed in order to carry forward regression testing. Every technique has its own way of reordering the test cases based on certain fitness value being calculated according to the algorithm selection.

3. Algorithm

3.1. Genetic Algorithm

Genetic Algorithm is a method based on the principles of natural selection and genetics. In GA, a population is formed from a set of chromosomes. The algorithm is dependent on fitness function that needs to be defined. Based on fitness function of each chromosome, one population is replaced successively by another. The chromosomes with high fitness values are included and the ones with low fitness values are removed. This population happens through the process of cross over and mutation. The following are the steps involved in GA.

- 1) **Initialization:** The chromosomes are considered as the test cases, and the possible combinations of test cases are calculated through fitness function. The initial population is the randomly generated set of test cases.
- 2) **Evaluation:** [1,2] After the initial population, the fitness value is determined. Determining the fitness function is the most vital task in genetic algorithm as this is the one helping to achieve the test cases that need to be picked for the selection method.
- 3) **Selection:** [1], [2] In this, chromosomes with high fitness values are chosen for the mechanism.
- 4) **Cross Over:** This involves recombination of parental solutions to populate a better solution.
- 5) **Mutation:** The mutation happens locally within the chromosomes and the modification happens randomly.

The iterations are repeated until the termination condition is reached which is our desired optimal solution to the problem.

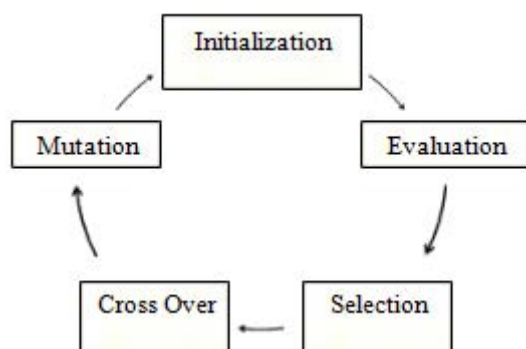


Figure 2: GA flow

3.2. Bee Colony Optimization Algorithm

The BCO algorithm was inspired by the foraging behaviour of the honey bees. The well-organized searching of the honey bees for the food sites and the way the communication

is being passed on from one bee to another is followed to arrive at an optimal solution to the problem. There are three important components involved – employed bees, unemployed and the food source [4]. This follows the neighbour search mechanism to find the best food source. The scout bees going in search collect the load of nectar [3] and bring it to the food hive after which, the employed bee pass on the information of the food path to the onlooker bees by dancing. The onlooker bees follow that independent food path and the sites that have more amount of nectar is remembered and the ones that have negative feedback are ignored. Thus, the honey bees achieve the path of best food sources.

Similarly, the population of the food sites are initialized and the various independent paths are generated [4]. Every independent path would represent a test case. The fitness value of each test case is being determined. The scout bee search for the test case with high fitness value.

The fitness value information is passed on to the onlooker bee and this compares the fitness value with that of the neighbouring test cases and store the test cases with high fitness value in the database and the ones with the less fitness value are ignored. This process is repeated until the termination condition that is the optimal sets of test cases are achieved.

3.3 Ant Colony Optimization Algorithm(ACO)

The ACO algorithm was inspired by the foraging behaviour of the real ants. The method of identifying the shortest path to the food store from the nest by the ants involves two important components – pheromone deposit and the pheromone evaporation [6]. Ants have the habit of depositing pheromone on the path they travel. The other ants follow these paths based on these deposits. Lesser the pheromone, higher the evaporation rate which in turn denotes longer the path. Thus, the ants follow the path with less evaporation and get to choose the shortest path. The algorithm used in regression testing by initializing a set of ants with randomly selected test cases and based on various parameters like time, cost or fault coverage, the new combinations of test cases have been generated and the process is repeated until the optimal solution is arrived.

4. Conclusion

The main objective of test case prioritization technique is to effectively carry out regression testing by identifying the important test cases and allowing only them to run rather than running the entire set of test cases of the software so as to reduce the cost and the time to a greater extent. In this paper, different test case prioritization techniques are discussed. Further investigation and analysis may be carried out on more techniques which will help to propose new ideas, that will be much effective to increase the rate of fault detection in the early stages of testing in short duration of time, leading to deliver high quality of software product.

References

- [1] Kristen R. Walcott Prioritizing,” Regression Test Suites for Time-Constrained Execution Using a Genetic Algorithm”, www.cs.virginia.edu
- [2] T.Prem Jacob and T.Ravi,” Optimization of test case by prioritization”, *Journal of Computer Science* 9 (8): 972-980, 2013 ISSN: 1549-3636 © 2013 Science Publications
- [3] Dušan TEODOROVIĆ, Mauro DELL’ ORCO,” Bee Colony Optimization – A Cooperative Learning Approach to Complex Transportation Problems”
- [4] Navdeep Koundal, Ankur Shukla, Mohsin Rashid Mir,” Test Case Selection Using Bee Colony Optimization”, *International Journal of Science and Research (IJSR)*, Volume 3 Issue 5, May 2014
- [5] Rainer Geisler,” Industrial Software Applications: A Master's Course for Engineers”
- [6] Anjali Choudhary, TarunDalal, “Evaluating the Effectiveness of Regression Test selection using Ant Colony Optimization”, *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*
- [7] Sebastian Elbaum, Alexey G. Malishevsky and Gregg Rothermel. Test Case Prioritization: A Family of Empirical Studies. *IEEE Transactions on Software Engineering*, 2002, Vol. 28, No. 2, pp. 159-182.
- [8] <http://www.webopedia.com/>
- [9] Erik Rogstad, Lionel Briand, Richard Torkar, “Test Case Selection for black-box regression testing of database applications”, *Elsevier-Information and Software Technology*, 2013