ABSTRACT

Software Testing is an essential part of Software Engineering in order to verify the product whether it meets the expectations of customers and clients from the software development team. A variety of testing is to be conducted to verify the quality attributes of the system and their levels of quality in the developed system. Hence, the software testing team should have a perfect testing plan to enable the testing phase to use the resources in a perfect manner.

The prepared Software Testing plan should give a clear picture about the resource allocation and resource utilization and the tests to be conducted for verifying the quality attributes. All quality attributes in a software system cannot be verified from a single test. The appropriate testing types should be identified to verify and measure the hidden quality attributes. Among the different test types, the ‘Regression Testing’ is used to verify the changes made in the existing source code when recommendations are made for modifying the source code or behaviour of the system or a system component. Due to this, the ‘Regression Testing’ may consume huge amount of resources in order to verify the changed and unchanged parts in the system to test.

The total cost of testing can be reduced only when the number of test cases is reduced to execute with the system. But, the test case reduction or test suite reduction should not affect the performance of the testing. So, a prominent test suite reduction technique is to be used for reducing the number of test cases existing in the final test suite. The reduced test suite should have maximum requirements coverage and it should be an optimal one. For having such test suites, here two algorithms are applied to the problem domain to satisfy the goals of software testing.

The test suite reduction technique is proposed using test case classification by using ‘Reusability’ level of test cases. This attribute of test cases can be found only when the historical data are maintained in the test case repository. The fuzzy set with a Boolean AND operation is applied and it was tuned with the ‘Reusability’ parameter to classify and to select the test case for final test suite. The ‘Risk’ factor of the test cases is combined with ‘Reusability’ factor to have a proper test case in the reduced test suite. The performance of the ‘Tuned Fuzzy Logic’ is better than the state art algorithms ‘BOG’ (Bi-Objective Greedy) and ‘HGS’ (Harrold, Gupta and Soffa). But, the test suite requirements coverage percentage is to be improved. So, a test suite optimization algorithm is derived from the metaheuristics techniques.

The ‘Firefly’ algorithm from metaheurisitics techniques is proposed for optimizing the test suite by using the same ‘Reusability’ factor. Hence, the nature of ‘Firefly’ algorithm is tuned using ‘Reusability’ level of test cases by measuring the sub attributes of ‘Reusability’. In this ‘Fine Tuned Firefly Algorithm’, the following attributes ‘Age’ and ‘Number of Versions’ of test cases are computed and applied to ‘Firefly algorithm’ to promise the level of ‘Reusability’.

The ‘Reusability’ of a test case can be used to decide that whether a test case is an optimal one or not. If a test case is used a certain number of times in its lifetime and in testing different applications, then its reuse frequency will be high. The reuse frequency is considered when the test case is able to satisfy the functional specification to be tested. So, here the test cases are classified according to their age and at the next level, those test cases will be classified according to their number of versions produced in different periods. The performance of ‘Fine Tuned Firefly’ algorithm on software test suite reduction is compared with ‘Genetic Algorithm’ and ‘Artificial Bee Colony Optimization Algorithm’ in order to find out the applicability of the

proposed algorithm. The obtained results show that the ‘Fine Tuned Firefly Algorithm’ may consume less execution time compared to ‘GA’, ‘ABC’ and ‘Tuned Fuzzy Logic’ algorithms to produce an optimized and reduced test suite.

The ‘Reusability’ framework is applied on test case pool to classify the test cases in order to speed up the selection of optimal test case(s). The proposed framework helps to divide the main test case pool into two different parts such as ‘Highly Reusable Test cases’ and ‘Low Reusable Test cases’. The classified test case pools will be able to reduce the time on selecting or searching a test case for a particular requirement specification to test. The test cases are sorted in descending order according to their Age and ‘Average Performance of Faults Detection’ metric to reduce the time of searching a suitable test case. A simple Linear Searching algorithm is applied to prove the time difference between accessing a whole library and a part of the library. Therefore, the proposed ‘Reusability’ framework is having multiple faces in the software testing domain for reducing the cost of testing and obtaining better results.