ISSN: 2278 – 7798
International Journal of Science, Engineering and Technology Research (IJSETR)
Volume 2, Issue 11, November 2013

Test Suite Optimization For Regression Testing By Using New Meta-Heuristics Algorithm.

1. Shaik Irfan,M.Tech,CSE 2. Vivek Kulkarni,Assoc Proffesor,CSE 3.K.Hanumantha Rao, Assoc Proffessor,CSE

ABSTRACT

Regression testing is any type of software testing that seeks to uncover new software flaw, or regressions, in existing functional and non-functional areas of a system after changes, such as improvement in quality, patches or configuration changes have been made to them. Assuming that TS={T1,T2,T3…..Tn} are test cases. which satisfies test criteria's C={C1, C2, C3……..Ck}.The project is aimed to reduce the test cases in the test suite such that TS'={T1,T2,T3….Tm},where m<n, satisfying all the test criteria's in 'C'.As the above problem is N-P complete, Heuristic Approach is to be adopted, various appropriate heuristic Algorithms will be used to optimize Regression test suite [2], [3], [4], [5], and we proposed a new Algorithm, which is more efficient. The proposed Meta-Heuristic Algorithm approaches to reduce the test suite by using test case-requirement matrix[3] and greedily selects an optimum test cases into the reduced test suite until all test criteria's are satisfied.

Keywords: *Software Regression testing, Regression test optimization, testing criteria, sparse matrix.*

INTRODUCTION

Software regression testing is a critical activity in the maintenance phase of evolving software. However, it requires large amounts of test cases to test any new or modified functionality within the program. Re-running all existing test cases together with the new ones is often costly and even in-feasible due to time and resource constraints. To address this problem, the research community proposed techniques to optimize regression testing [2], [3], [4], [5],.Re-running test cases that do not Exercise any changed or affected parts of the program makes extra cost and gives no benefit. An effective technique is to permanently discard such redundant or obsolete test cases and retain the most effective ones to reduce the excessive cost of regression testing [2]. Such technique attempts to find a minimal subset of test cases which satisfy all the testing requirements as the original set does . This subset could be found during the test case generation or after creating the test suite. As far as the less the number of test cases the less time it takes to test the program, so the optimization of test cases is done. This consequently improves the successful in producing a desired result of the test process. This technique is commonly known as test suite reduction or test suite minimization

DISCUSSION

One of the main objectives of the proposed algorithm is to select effective test cases in fault detection. However, new testers are not aware of faults and their severities before testing. Thus, they have to estimate faults behavior through some heuristic methods. One way to do this is using the extent of code coverage by each test case. In other words, coverage of testing requirements can provide a reasonable estimate of fault detection capability the proposed algorithm requires the coverage information of test cases for a single criterion. Unlike some existing approaches, the required information can be assemble accurately and full automatically. Moreover, our proposed algorithm exactly determines test cases once they become redundant by selecting a test case into the reduced suite. Thus, it is possible to use other criteria to improve fault detection loss.

EXISTING SYSTEM

In the existing system, the Algorithm has a vector called sumColumns which will be computed which will indicate the number of requirements coverage

overlap of a test case with others and the algorithm repeatedly selects an optimum test case until all testing requirements are satisfied. In each of its iteration. the *selected* vector is updated with respect to the selected test case and cumulative coverage of the reduced suite is updated. Moreover, the diagonal elements of the *multiplied* matrix are updated for unselected test cases. If during the update process, the value of an element becomes zero, it is redundant and will be removed from further considerations. The existing Algorithm increases the complexity. Even for a simple matrix the whole Algorithm should be run. The existing system makes the complexity for a new tester, and even the error in the Algorithm cannot be easily identified.

PROPOSED SYSTEM

Our approach to test suite reduction has been stimulate by the following issue: An
Appropriate test suite reduction technique should select test cases that are both unique in exercising execution paths and effective in fault detection. The first objective attempts to remove as much redundancy from the test suite, and the second one seeks for satisfying the main purpose of software testing which is fault detection.
The proposed approach to test suite reduction uses test case-requirement matrix. This matrix shows the mappings between test cases and testing requirements. The elements consist of 1's and 0's which indicate for satisfying or dissatisfying the requirements by test cases respectively. the test cases are taken as input for the tool and the output will be the reduced test cases.
In this paper, we propose a new algorithm for test suite minimization. The proposed algorithm greedily selects an optimum test case into the reduced suite until all testing requirements are satisfied.

IRFAN ALGORITHM- 1

Step-1 Find Number of zero's in All the rows

Step-2 If there are no zero's in any of the row , then the test case satisfies all the criteria's and store the test cases in the database

Step-3 Otherwise take the row(test case) with min zero's take unary operation with another test case with minimum zero's and if it satisfies all the criteria's then those two test cases stores in database.(t1Ut2)

Step-4 If doesn't satisfy, and then apply unary operation with first test case and next minimum test case. (t1Ut3) stores in database

Step-5 If this (t1Ut3) satisfy all the criteria then its ok or apply the unary operation for

(t1Ut2) Ut3.

Step-6 ((t1U2) Ut3)) Ut5 ok store in database otherwise

Step-7 ((t1Ut2) Ut3)) Ut6 ok store in database otherwise

Step-8(t5Ut6) ok store in database

Step-9 ((t1ut2) Ut3) U (t5Ut6) after execution of Algorithm-1, check solution set and look for the minimum test cases. If it is 30% optimized don't go for

IRFAN ALGORITHM-2

Step-1 Take Row(test case) with minimum zero's take Unary operation with another test case with minimum zero's and if satisfies All the criteria then store these two test cases in data base and store in database.

Step-2 Which is Mathematically represented as (r1 U r2)

Step-3 And if this won't satisfy then we are applying unary operator for test case first and another test suite which is having minimum zero so that it satisfy all the criteria then store it in the database or else apply till it, till r n which satisfies all the criteria and store it in database

Step-4 if won't satisfy then go for the second test case which is having minimum zero and again proceed the same procedure and this will be done till

r n-1 U r n.

Step-5 If it satisfies then Ok otherwise with step 1 apply unary operation to next test case so that this combination will satisfy all the criteria. and the same procedure should be followed till it satisfy all the criteria and the all combinations will be

(r n-2 U r n) U (r n)

Step-6 And if this combination wont satisfy then the combination should be increased and test till it satisfy with the combination

((r n-3 U r n-2) U r n-1) U r n)

MATRIX REPRESENTATION

flow chart for finding minimum test suite

	C1	C2	C3	C4	C5	C6
T1	1	1	0	0	1	1
T2	1	0	0	1	0	1
T3	1	1	1	0	0	1
T4	0	0	1	1	1	1
T5	0	0	1	1	1	1
T6	1	0	1	1	1	0

Fig: matrix with test cases and criteria

Fig: 1 matrix representation

FLOW CHART

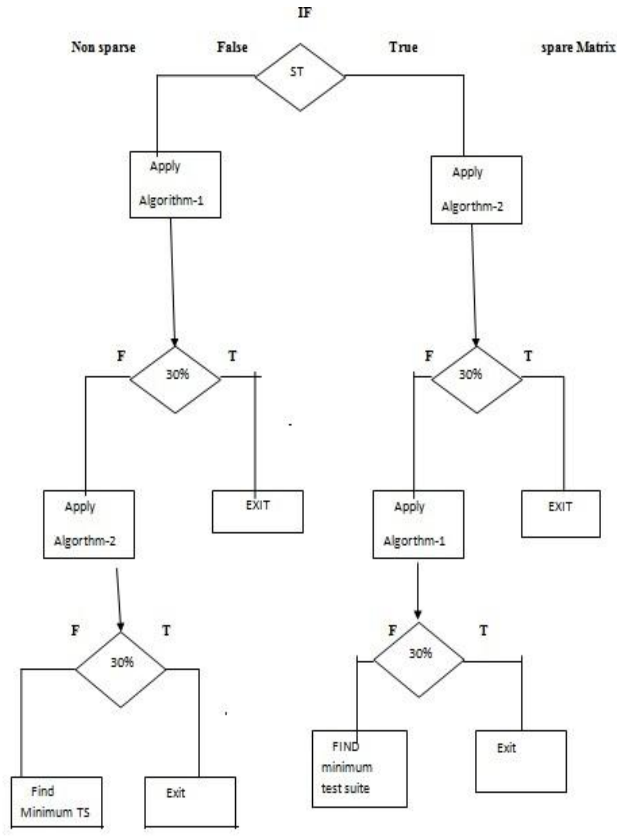
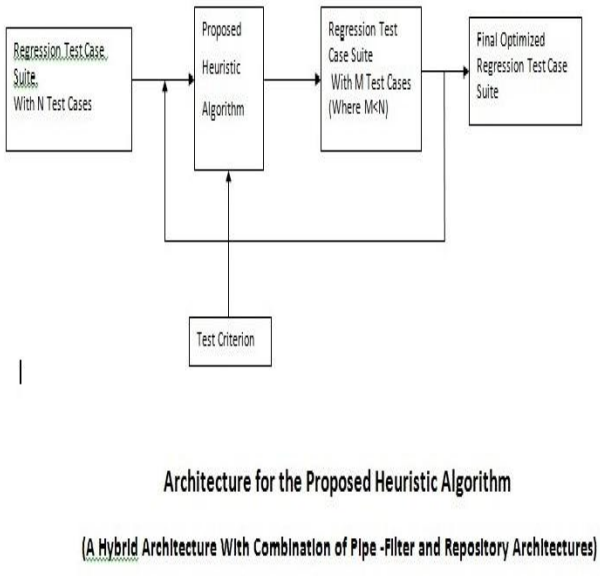


Fig: 2 flow chart

PROJECT ARCHITECTURE



Architecture for the Proposed Heuristic Algorithm

(A Hybrid Architecture With Combination of Pipe -Filter and Repository Architectures)

Fig:3 project Architecture

First the regression test cases (N) will be given to the Algorithm and the algorithm will reduces it to

regression test cases(M) and M<N. while the input as regression test cases (N) the other internal input the test criteria's should be given. The output should not be taken as the final test cases, it should be again given as input and again should compared for the first output and the second output, if both are equal then should be taken as the final output otherwise should continue the procedure till the outputs become equal.

The Architecture is said to be Hybrid Architecture because it is a combination of pipe-filter and repository Architectures.

CONCLUSION

We have presented a new algorithm for test suite minimization. The new algorithm considers test suite minimization as an optimization problem with two objectives. The first objective is fault detection capability which should be maximized. The second objective is number of test cases which should be minimized. The new algorithm is needed to be applicable and effective in reducing suites with significant suite size reduction and improved fault detection capability. This observation is evidenced by the results obtained from the experiments similar to prior studies which compare our algorithm with the best existing approach on typical benchmarks.

REFERENCES

[1] A methodology for controlling the size of a test suite - Harrold, Gupta, et al. - 1993

[2] Analyzing Regression Test Selection Techniques - Rothermel, Harrold - 1996

[3] Effect of Test Set Minimization on the Fault Detection Effectiveness of the All-Uses Criterion - Wong, Horgan, et al. - 1995

[4] The Category-Partition Method for Specifying and Generating Functional Tests - Ostrand, Balcer - 1988

[5] Experiments to assess the costbenefits of test-suite reduction - Rothermel, Harrold, et al. – 1999

First Author:

Shaik Irfan, completed engineering from Deccan college of Engineering and technology Affilated to Osmania University in 2011,presently persuing masters in Technology from Aurora Scientific Technological Research Academy Affilated to Jawaharlal Nehru Hyderabad.

Second Author:

Vivek Kulkarni, 18 years of experience in teaching, presently Head of Department of CSE in Aurora Scientific Technological Research Academy Bandlaguda(Hyderabad) Affilated to Jawaharlal Nehru Technological University. he haspublished paper in 12 National journals conferences and 7 international journal conferences.

Third Author:

K.Hanumantha Rao 9 years of Experience in Teaching, from 2006 to 2011(5 yrs) he was Assist professor in Sri Indu college of Engineering and Technology(Hyderabad) Affilated to Jawaharlal Nehru Technological University and presently Associate Professor in Computer Science and Engineering in Aurora Scientific Technological Research Academy Bandlaguda(Hyderabad) Affilated to Jawaharlal Nehru Technological University.