Volume 5, Issue 3, March 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Prioritization of Minimized Regression Test Suite Using Genetic Algorithm

Ekta Tyagi*, Yashika Sharma
Computer Science & Engineering, MRCE,
Haryana, India

Abstract— Regression testing is a process of rerunning the test cases against modified parts of the already tested code of the software and ensuring that modification has introduced no new errors to it. For this type of testing, it is always desirable to spend as little time as possible doing testing without reducing the probability of detecting new failure in already tested code. Test suite minimization technique address this issue by removing redundant test cases and Test case prioritization technique by scheduling test cases in an order that enhance the efficiency of attaining some performance criteria. This paper presents a new approach for regression testing by combining these two techniques. The approach is to first minimize the test suite by using greedy approach and then prioritize this minimized test suite using genetic algorithm. Proposed approach supports tester by minimizing the test suite while ensuring all the requirement coverage and minimum execution time. The overall aim of this research is to make the testing process time and cost effective by reducing the number of test cases that need to run after changes have been made.

Keywords— Regression Testing, Genetic, Initial Population, Test cases minimization, Prioritization.

I. INTRODUCTION

Regression testing is expensive but an essential activity in software maintenance [1]. It is performed to ensure the validity of the modified software. To test the modified software, new test cases are added to already existing test suite to test the changed requirements, which increase the size of the test-suite, cost and time constraints. So, to enhance the efficiency of software testing, improvements in the regression testing would help in reducing the cost of the software. To reduce the cost of regression testing, prioritization of test cases becomes essential. Other concern in regression is the effectiveness of test suites in finding new faults in successive program versions.

Several different techniques are available to reduce the cost of regression testing [2].

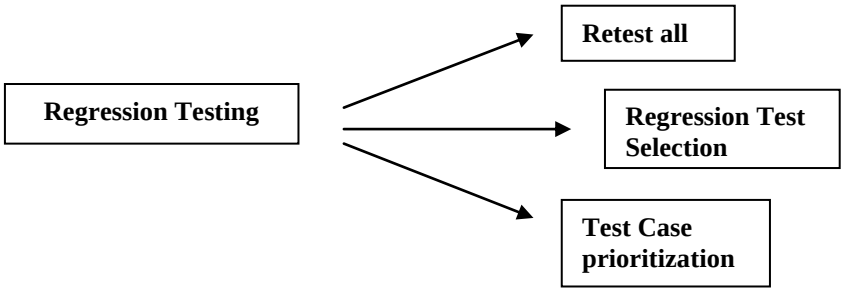

Fig. 1. Regression Testing Techniques

1) Retest all – It is one of the conventional techniques for regression testing in which each and every test case in the existing test suite are rerunned. This technique is not feasible most of time as it require more time and budget.
2) Regression Test Selection (RTS) - Here, RTS allows us to omit some of the test cases. RTS is beneficial only if the cost of selecting some of the test cases is less than the cost of executing the complete test suite. In this technique only part of test cases in test suite is selected to rerun. RTS techniques are further classified into three categories [3] :- Coverage Technique , Minimization Technique and Safe Technique.
3) Test Case Prioritization – Test case prioritization techniques arranges test cases in a most beneficial order thus making the testing process more effective. There are 18 different test case prioritizations techniques [4] numbered P1-P18 which is divided into three groups: - Comparator Techniques, Statement Level Techniques, and Function Level Techniques.

The proposed technique is a combination of minimization techniques which is a sub- technique of Regression Test selection and Test case prioritization with the goal of removing redundant test cases and then prioritizing them especially for time constraint execution.

II. RELATED WORK

Several techniques have been proposed for efficiently computing solution to test case minimization and test case prioritization problem. Chavatal [5] proposes the greedy approach that selects test case one by one that covers the most yet to be covered requirements until all the requirements are covered. In case of tie , one test case is ramdomly selected.

Xiaofang Zhang , Changhai Nie, Baowwen Xu. Bo Qu discussed prioritization of test cases in regression testing and presented a new test case prioritization technique [6] and a metric to measure its effectiveness. He illustrates that the rate of "units of testing requirement priority satisfed per unit test case cost" can be increased using that technique, and hence the testing quality and customert satisfaction can be improved.

Vishnu Raja.P , Murali Bhaskaran.V demonstrated that if entire population is considered it reduces the efficiency of the algorithm, because the initial population for algorithm has good as well as bad individuals. If the population has only good individuals or one cam make it so by removing bad individuals, the performance is improved better [7].

Yu-Chi Huang et. al has proposed a cost effective test cost prioritization technique based on the use of historic records and genetic algorithm [8]. Controlled experiment is performed to evaluate the proposed techniques's effectiveness.

Md. Imrul Kayes proposed test cases priortization for regression testing based on fault dependency [9]. He also presented a metric APFDD which measure fault dependency detection rate and presented an algorithm to improve APFDD.

III. GENETIC ALGORITHM

The Genetic algorithm is an evolutionary algorithm and population based search method. The selection takes place from the available population using fitness function; genetic operators are applied to obtain an optimal solution, followed by termination.

A. Initialization

Genetic algorithm starts by generating many individual solutions which form an initial population. Nature of the problem determines the population size and performance of Genetic Algorithm can be improved by Reducing the Population Size [7].

B. Selection

Individuals are selected amongst the population on the basis of their fitness value. The selection can be using roulette wheel selection or tournament selection method. Roulette wheel selection is done on the basis of fitness of each individual while in tournament selection, the individuals are selected randomly.

C. Genetic Operators

The genetic operators are crossover and mutation. The crossover operator is a binary variation operator, applied on two parents to obtain an optimal solution; solution in this case is a child. There are types of crossover: 1-point crossover, 2-point crossover, uniform crossover and multipoint crossover. Mutation operator is a unary variation operator, applied to one parent and delivers a modified mutant.

D. Termination

This process of selection and applying operator is repeated until a specified termination condition has been reached. Some of the common terminating conditions are Fixed number of generations reached, solution is found that satisfies minimum criteria, Manual inspection, Allocated budget (computation time/cost) reached.

IV. TEST CASE PRIORITIZATION

Formal defination for Test Case Prioritization Problem [10] is as follow:-

Given: T, a prepared test suite for testing a code, PT, the set of all the possible ordering (permutations) of T, and f, a function from PT to a real number.

Problem: Find $T' \in PT$ such that $(\forall T'')\ (T'' \in PT)\ (T'' \neq T')\quad [f(T') \geq f(T'')]$.

In above definition, PT represents all the permutation of T, and f is a function that when applied to any permutation, assign an award value for that permutation.

Number of metrics is available to measure the level of effectiveness of prioritized Test Suite like APFD metric [11] by Elbaum where calculation is performed by taking the weighted average of the number of faults detected during the run of the test suite, PTR metric.

V. PROPOSED APPROACH

In this section we present technique to make regession testing efficient.

Let's say a program , P has a test suite T in the form of Test case- requirement matrix ,representing the requirements each test case is covering as shown in Table 1.

Table 1. Test Suite for program P (TestCases.txt)

Test cases	**Requirements**						
	R1	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**

T1	**1**	**0**	**1**	**1**	**0**	**0**	**0**
T2	**0**	**1**	**1**	**1**	**1**	**0**	**0**
T3	**1**	**1**	**1**	**1**	**0**	**0**	**0**
T4	**0**	**0**	**0**	**0**	**1**	**0**	**1**
T5	**1**	**0**	**0**	**1**	**0**	**1**	**0**
T6	**1**	**0**	**1**	**0**	**0**	**0**	**0**
T7	**0**	**1**	**0**	**1**	**0**	**1**	**0**
T8	**0**	**0**	**0**	**0**	**0**	**0**	**0**
T9	**0**	**1**	**0**	**1**	**1**	**1**	**0**

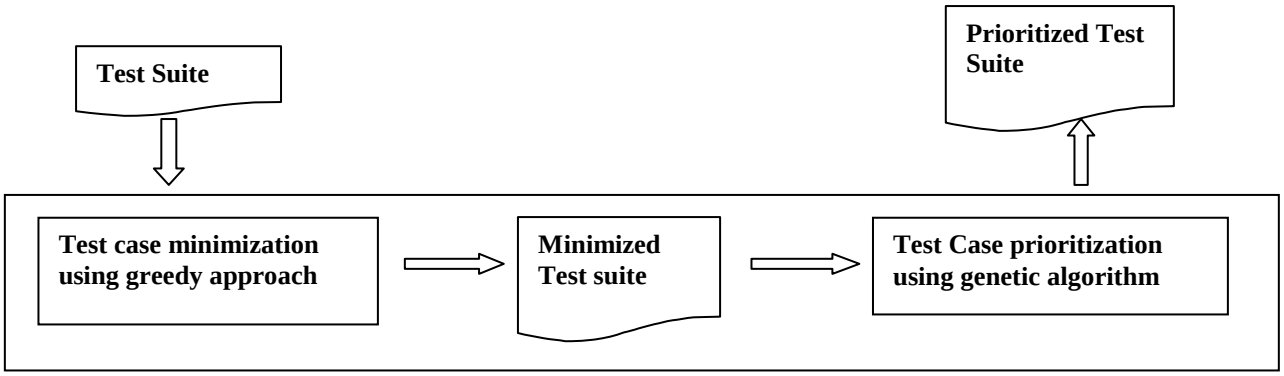

Fig. 2. Overall view of our approach

Prepared test suite is input for Test case minimization module , works by start selecting test cases in order of requirements they are covering means it first select the test case covering maximum requirement, then the test case covering second maximum requirements and so on, until all the requirements are covered and ignore rest of the test cases. Input for Test Case Prioritization module is this minimized test suite . This module will prioritize test cases of this minimized Test Suite according to specified goal of prioritization i.e the fitness value.

Benefit of minimization of test suite before prioritization is , this will reduce the test cases hence the intial population for genetic algorithm used for Test Case prioritization. Generally, using genetic algorithm , for identifying the best population entire population is considered. So , to improve the efficiency of the algorithm bad individuals should be identified and eliminated [7].

In our problem , bad population is redundant test cases which Test case minimization module can eliminate. Hence, improve the efficency of Test Case prioritization using Genetic Algorithm**.**

VI. EXPERIMENTAL RESULTS

We are defining a new approach to minimize and then priorities the test cases using genetic algorithm. For this purpose we have used the MATLAB simulator.

On applying Minimization technique as proposed on Test Suite shown in Tabel 1. and Fig 3, Test cases T1,T2,T3, T4,T5,T6,T9 are selected as shown in Fig 4.

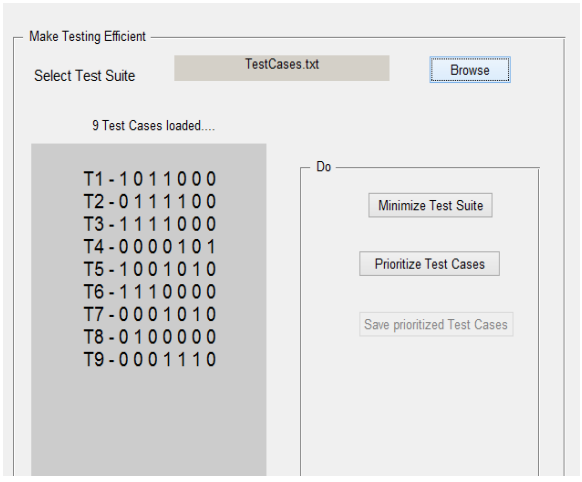

Fig. 3. Prepared Test Suite

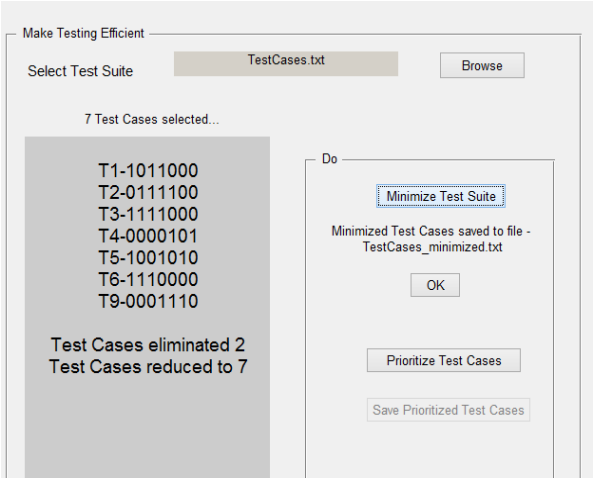

Fig. 4. Minimized Test Suite

Now, For prioritizing these minimized Test Suite a random cost is assigned to each test case . Prioritized output for this random assignment is shown in fig 5. The basis paramenters defined while performing Genetic Algorithm for prioritization are:-

Parameter	Value
Number of Test Cases	7
Fitness Function	Minimization
Generation	100
Test cost	0 to 1
Crossover	PMX

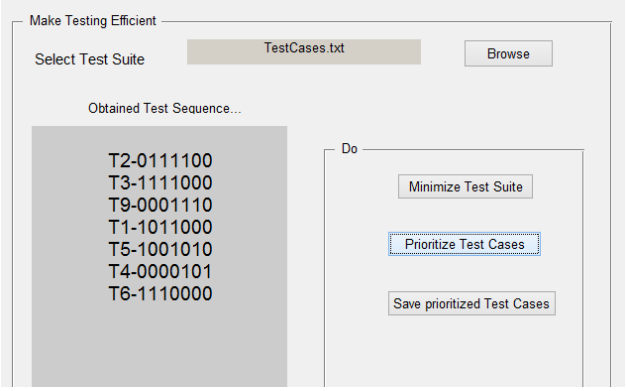

Fig. 4. Prioritized Test Sequence

The output based on this random cost assignement is :-

a) The obtained Test Sequence of this assignement is
   2 3 9 1 5 4 6
b) The Process cost driven from the genetic on initial cost assigment is
   Process cost = 9.479643
c) The cost driven after implementation of optimized est sequence is
   Test Cost = 1.2563

VII. CONCLUSION

In this paper we proposed an approach which combine the Minimization technique and prioritization techniques of Regression Testing. Proposed approach is definitely better than using these two techniques separately in term of time as well as cost. Test case with highest fitness value is assigned highest priority. Genetic algorithm is best choice for prioritization as using this algorithm fairly large number of time we will get optimum solution.

ACKNOWLEDGEMENT

I express my sincere and deep gratitude to my guide Mrs. Yashika Sharma, Assistant Professor, Department of Computer Science & Engineering , Manav Rachna College of Engineering, Faridabad, for her guidance, support and encouragement.

REFERENCES

[1] Clarke. J., Dolado, J. J., Harman, M., Hierons, R. M., Jones, B. and M. Shepperd, "Reformulating, Software Engineering as a Search Problem," IEEE Proceedings - Software, Volume 150, Issue. .3, June 2003, Pages 161-175.

[2] K. R Walcott , "Prioritizing regression test suites for time constrainted execution using genetic algorithm". [online] available at www.cs.virginia.edu/~krw7c/gaprioritization.pdf

[3] Todd L. Graves, Mary Jean Harrold, Jung – Min Kim and Adam Porter, Gregg Rothermal ," An Empirical Study of Regression Test Selection Selection Technique ", "ACM Transactions on Software Engineering and Methodology", Volume 10, Issue. 2, Pages 184–208, April 2001.

[4] Sebastian Elbaum, , Alexey G. Malishevsky, and Gregg Rothermel,, "Test case prioritization: A family of empirical studies," IEEE Transactions on Software Engineering, Volume 28, Issue.2, pages 159-182, Feb.2002.

[5] V. chavatal, "A greedy heuristic for set covering problem " ,Mathematics of operations Research, 4(3) , Pages 233-235,August , 1979.

[6] Md. Imrul Kayes ,"Test Case Prioritization for regression Testing Based on Fault Dependency", IEEE ,ICECT, 2011, 3rd International Conference, Volume 5,Pages 48-52.

[7] Xiaofang Zhang , Chaghai Nie, Haowen X. Bo Qu,"Test Case Prioritization based on varying Testing Requirement Priorities and Test Case Costs", IEEE, QSIC'07, 7th International Conference , 2007,Pages 15-24.

[8] Yu-Chi Huang, Chin-Yu Huang, Jun-Ru Chang and Tsan-Yuan chen "Design and Analysis of Cost-cognizant Prioritization Using Genetic Algorithm with test history". IEEE 34th Annual Computer Software and Application Conference 2010, Pages 413-418.

[9] Vishnu Raja.P , Murali Bhaskaran.V. "Improving the Performance of Genetic Algorithm by Reducing the Population Size", International Journal of Emerging Technology and Advanced Engineering , Volume 3, Issue 8, August 2013.

[10] Rothermel, Gregg; Untch, Roland H.; Chengyun chu; and Harrold Mary Jean,."Prioritizing Test Cases for Regression Testing", IEEE Transactions on Software Engineering, Volume. 27, Issue. 10, Pages. 929-948, October 2001.

[11] Himanshu Ghetia , Shantanu Santold, Vairamuthu S , "Test case Prioritization based on testing requirement Priorities and fault dependency", International journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 6, June 2014.

