ISSN : 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 7, September 2012

A Genetic Algorithm for Regression Test Sequence Optimization

Suman[1], Seema[2]

M.Tech. Student (Computer Engineering)[1], Assistant Professor (Computer Engineering) (Supervisor)[2]

Yadavindra College of Engineering, Punjabi University Guru Kashi Campus, Talwandi Sabo, Bathinda, Punjab, India.

ABSTRACT: Regression testing is the process of validating modified software to assure that changed parts of software behave as intended and unchanged parts of software have not been adversely affected by the modification. The regression test suite is typically large and needs an intelligent method to choose those test cases which will reduce the overall test cost. In this situation, test case prioritization techniques aim to improve the effectiveness of regression testing by ordering the test cases so that the most beneficial are executed first. In this approach, a new Genetic Algorithm to prioritize the regression test suite is introduced that will prioritize test cases dynamically on the basis of complete code coverage. Meanwhile, an approach to generating new test cases is presented using PMX and cyclic crossover and analysis is done on the basis of process cost and test cost. The overall aim of this research is to reduce the number of test cases that need to be run after changes have been made.

Keywords: Regression Testing, Dynamic Prioritization, Fitness Function, Mutation, Cross Over.

I. INTRODUCTION

Regression testing is expensive but an essential activity in software maintenance. Regression testing attempts to validate modified software and ensure that the modified parts of the program do not introduce unexpected errors. It executes an existing test suite on a changed program to assure that the program is not adversely affected by unintended amendments. The time used for regression testing can be assumed approximately half of the software maintenance activities. However, through the use of an effective prioritization sequence, testers can reorder the test cases to reduce the time and cost in the system, allowing corrections to be made earlier and raising overall confidence that the software has been adequately tested. One concern in regression testing is the effectiveness of test suites in finding new faults in successive program versions. A regression test selection technique may help us to select an appropriate number of test cases from this test suite [1]. Test case prioritization techniques concern with identifying an ideal order of test cases according to some criteria, such that test cases with higher priority are executed earlier than ones with lower priority [1]. Several different methods have been developed to reduce the cost of regression testing [2]. These include retest all, test selection, and prioritization techniques.

1) Retest All: **-** Retest all method is one of the conventional methods for regression testing in which all the tests in the existing test suites are performed again. This type of technique is not feasible in most of time as it require more time and budget compared to techniques which will be discussed further.

2) Regression Test Selection: - The retest-all approach, reruns all available tests, this strategy generally consumes excessive time and resources. Regression test selection techniques, in contrast, attempt to reduce the time required to retest a modified program by selecting some subset of the existing test suite. Selective regression testing attempts to reduce the cost.

3) Regression test Prioritization Technique: - Test case prioritization techniques organize the test cases in a test suite by ordering such that the most beneficial are executed first thus allowing for an increase in the effectiveness of testing. It is better than other techniques as it doesn't discard or permanently remove the test cases from test suite. Effectiveness will be measured by the rate of faults detected.

The research implemented a new regression test suite prioritization algorithm that prioritizes using a genetic algorithm with the goal of minimizing the number of test cases that are likely to be found during time-constrained execution.

This paper is organized as follow: Section I gives the Introduction of the Regression testing and its techniques. Section II is helpful to understand the background of related work. Section III explains about Genetic algorithm including crossover and mutation. Section IV explains the algorithm. Section V shows the performance of proposed technique and the last section VI concludes the paper and followed by the references.

II. RELATED WORK

In this section, a brief overview of the previously proposed methods to find the optimal test sequence dynamically while performing the regression testing is discussed. Kaushik et al. [1] proposed a paradigm called Dynamic Prioritization which involves changing the order of test cases during the testing process. Since the test case pool changes through the development cycle, the list of prioritized test cases would change as well. Kaur et al. [3] Proposed a new Genetic Algorithm to prioritize the regression test suite is introduced

ISSN : 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 7, September 2012

that will prioritize test cases on the basis of complete code coverage. The genetic algorithm would also automate the process of test case prioritization. Rothermel et al. [4] describe several techniques for using test execution information to prioritize test cases for regression testing. Do et al. [5] conducted a series of experiments to assess the effects of time constraints on the costs and benefits of prioritization techniques. Kumar et al. [6] presents a combined approach by which the stated problems are resolved in effective manner. Jiang et al. [7] select the test cases that can test part of changes, and then do the reduction for these selected test cases. In addition to the methods mentioned in this section, a large number of methods were proposed in the past. A good survey of test case prioritization methods as well as algorithms for optimal test sequence analysis can be found in [8] and [9].

In the proposed approach, the genetic algorithm is used to reduce the time and cost of regression testing along with it will provide simple testing flow as the test cases will be minimized. Designed system uses the Partially mapped crossover and swap mutation to optimize the test sequences.

III. GENETIC APPROACH

Genetic algorithm is a population-based search method. Genetic algorithms are acknowledged as good solvers for tough problems. Genetic algorithms require several parameters including the following [2]:

1. Maximum number of generations, G,
2. Population size, P,
3. Crossover rate, pc,
4. Mutation rate, pm,
5. Convergence criterion.

There are two main concepts in genetic algorithm viz: crossover and mutation.

A. Crossover

A binary variation operator is called recombination or crossover. This operator merges information from two parent genotypes into one or two offspring genotypes. Similarly to mutation, crossover is a stochastic operator: the choice of what parts of each parent are combined, and the way these parts are combined, depends on random drawings. The principle behind crossover is simple: by mating two individuals with different but desirable features, we can produce an offspring which combines both of those features.

Child1 = c*parent1 + (1-c)*parent2

Child2 = (1-c)*parent1 + c*parent2

B. Mutation

A unary variation operator is called mutation. It is applied to one genotype and delivers a modified mutant, the child or offspring of it. In general, mutation is supposed to cause a random unbiased change. Mutation has a theoretical role: it can guarantee that the space is connected.

A simple piece of code:

```
child = generateNewChild();
```

The optimization problems are solved by GA's recombination and replacement operators, where recombination is key operator and frequently used, whereas, replacement is optional and applied for solving optimization of problem [2].

IV. ALGORITHM

1. Accept the N Test Cases in form of TestCost Matrix Such as
   TestCost(i1,j1)=Cost1
   TestCost(i2,j2)=Cost2
   TestCost(i3,j3)=Cost3
   |
   |
   |
   TestCost(iN,jN)=Cost
2. Define the Initial Population Size called PopSize
3. Generated the Random Population Set to represent the possible test sequences
   Population=RandPermutation (TestSequnce,PopSize)
4. Define FitnessFunction called Min(TestCost)
5. For i=1 to MaxIterations
   [Repeat Steps 6 to 10]
6. Select two Random Parents called P1 and P2 from Popultation Set
7. Perform PMX crossover to generate new Child
8. Perform Mutation Operation
   Child=Mutation(Child)
9. Population=Population U Child
10. Return Optimized Test Sequence

The optimal solution is searched in GA on the basis of desired population which further can be replaced with the new set of population. Depend upon the problem, the generation and initialization of test cases (population) is done. The fitness function will help in selecting suitable population for problem. The overall fitness of the population generally improves with each generation. Further, the genetic operations are performed. In the beginning, PMX crossover recombines the two individual. Then mutation randomly swaps the individuals. Thirdly, the redundant individuals are removed. Finally, the solution is checked for optimization. If solution is not optimized, then, the new population is reproduced and genetic operators are applied.

ISSN : 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 7, September 2012

V. SIMULATION AND EXPERIMENTAL RESULTS

The complete work has been drawn on different scenarios of genetics. We have implemented the work with some constant stages and implemented the variations at one stage called Cross algorithm. We have implemented two main selection algorithms

1. Partially Mapped Crossover (PMX)
2. Cyclic Crossover

A. *Scenario* 1

The basic properties of this Scenario is

Table 1: Scenario 1 Parameter

Parameter	**Value**
Population	Random
Fitness Function	Min Test Cost
Number of Iterations	100
Population Size	10
Crossover	Cyclic

As we can see in this scenario we have implemented the optimization of test cost for a software project with 10 test cases, as the general case we have assigned the random cost to each test case and perform the analysis based on this random cost assignment. The output driven based on this assignment is shown as under.

a) The obtained Test Sequence of this random cost assignment is given as

3 7 5 4 8 10 9 2 1 6

b) The Process cost driven from the genetic on initial cost assignment is given as

Process Cost = 2.037950

c) The cost driven after implementation of optimized test sequence is given as

Test Cost = 3.10426

5.2 Scenario-2

The basic properties of this Scenario is

Table 2: Scenario-2 Parameter

Parameter	**Value**
Population	Random
Fitness Function	Min Test Cost
Number of Iterations	100
Population Size	10
Crossover	PMX

As we can see in this scenario we have implemented the optimization of test cost for a software project with 10 test cases, as the general case we have assigned the random cost to each test case and perform the analysis based on this random cost assignment. The output driven based on this assignment is shown as under.

a) The obtained Test Sequence of this random cost assignment is given as

6 7 4 3 8 5 10 9 2 1

b) The Process cost driven from the genetic on initial cost assignment is given as

Process Cost = 1.979647

c) The cost driven after implementation of optimized test sequence is given as

Test Cost = 3.64043

C. Analysis(Scenario-1 and scenario-2)

Here the analysis of the two crossover approaches is shown that are crossover and the PMX crossover. We have implemented these two crossovers on a set of 10 test cases. The analysis here driven in the form of performance and the test cost. The analysis graph is shown in Fig. 1 and Fig. 2.

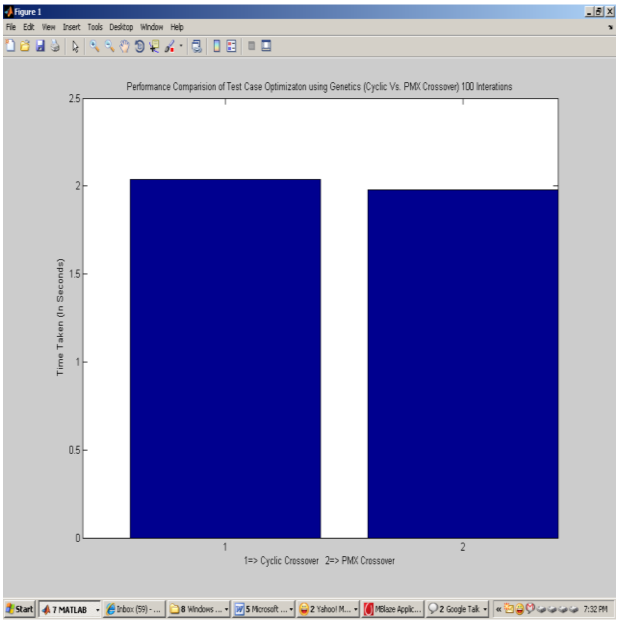

Fig. 1 Performance Comparison (100 Generation 10 Test Case)

As we can see Fig. 1 is showing the performance comparison of PMX and the Cyclic Crossover in case of a suite of 10 test cases. We have implemented it for 100 generation. As we can see the cyclic crossover is less effective in terms of time.

ISSN : 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 7, September 2012

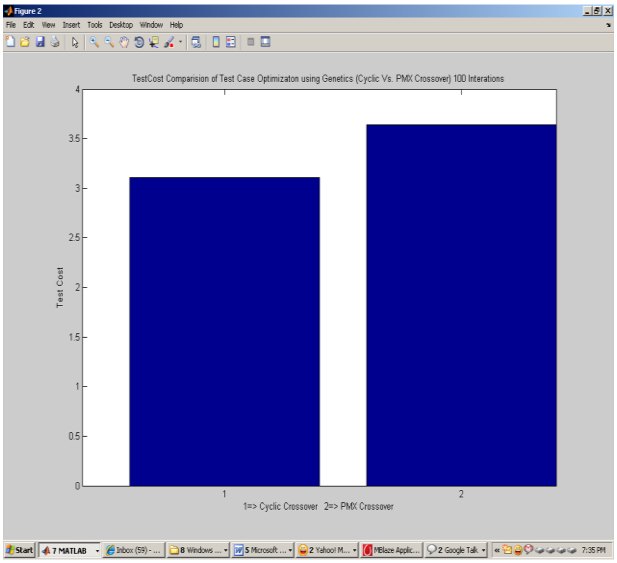

Fig. 2 TestCostComparison (100 Generation 10 Test Case)

As we can see Fig. 2 is showing the performance comparison of PMX and the Cyclic Crossover in case of a suite of 10 test cases. We have implemented it for 100 generation. As we can see the cyclic crossover is more effective in terms of test cost optimization.

We further can change population from 100 to 200 and 300, and conclude the results for both cyclic and PMX crossover.

VI. CONCLUSION

In this present work we have used the Genetics approach to find the optimal test sequence dynamically while performing the regression testing. From this complete work we can conclude the genetic can be used effectively to perform the work on test case generation but it can be used effectively if there is large number of test cases and large number of possible test sequences. The work has generated an optimal test sequence.

REFERENCES

[1] N. Kaushik, M. Salehie, L. Tahvildari, S. Li, M. Moore (2011) "Dynamic prioritization in regression testing" IEEE fourth international conference on software testing, verification and validation workshops ,pp: 135-138

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

[3] A. Kaur, S. Goyal (2011) "A genetic algorithm for regression test case prioritization using code coverage", International journal on computer science and engineering, vol. 3, pp:1839-1847

[4] G. Rothermel, R. H. Untch, C. Chu , M. J. Harrold (2001) "Prioritizing test cases for regression testing" IEEE transactions on software engineering (to appear),pp:1-33

[5] H. Do, S. Mirarab, L. Tahvildari, G. Rothermel (2010) "The effects of time constraints on test case prioritization: a series of controlled experiments" IEEE transactions on software engineering, vol. 36, No. 5, pp: 593-617

[6] A. Kumar, S. Tiwari, K. K. Mishra, A.K. Misra (2010) "Generation of efficient test data using path selection strategy with elitist GA in regression testing", IEEE, pp:389-393

[7] A. Jiang, Y. Mu, Z. Zhang (2010) "Research of optimization algorithm for path-based regression testing suit", 2ND IEEE International workshop on education technology and computer science, , pp:303-306

[8] R. Kavitha, N. S. Kumar (2010) "Test case prioritization for regression testing based on severity of fault" (ijcse) international journal on computer science and engineering Vol. 02, No. 05, pp: 1462-1466

[9] S. Raju, G. V. Uma (2012) "Factors oriented test case prioritization technique in regression testing using genetic algorithm" European Journal of Scientific Research, Vol.74, No.3, pp. 389-402

[10] R. Malhotra, A. Kaur, Y. Singh (2010) "A regression test selection and prioritization technique" Journal of Information Processing Systems, Vol.6, No.2, pp: 235-252

Biography

Suman received her B.Tech degree in computer science and engineering from GuruGobind Singh College of Engg. & Tech.(GGSCET), Talwandi Sabo(Bathinda), Punjab, India, in 2009, and pursuing M.Tech degree in computer science and engineering from Yadavindra college of engineering, Punjabi University Guru kashi campus, Talwandi sabo(Bathinda), Punjab, India. Her research interests include Software Testing and Image processing.

Mrs. Seema is presently working as Assistant Professor in Computer Engineering at Yadavindra College of Engineering, Punjabi University Guru Kashi Campus, Talwandi Sabo (Distt Bathinda) Punjab w.e.f. year 2008. She has almost seven year of teaching experience of teaching B.Tech. (CSE), MCA and M.Tech. (CE) Classes. She previously worked as Senior Lecturer & Head, Department of Computer Science & Engineering at Bhai Maha Singh College of Engineering, Muktsar and Lecturer (Computer Science & Engineering) at Government Polytechnic, Bathinda. She completed her Bachelor of Technology (CSE) from Institute of Engineering & Technology, Bhaddal, Ropar in year 2004 holding 9th merit position in the University. She completed Master of Technology in Computer Engineering in year 2007 from Punjabi University, Patiala, Punjab. She has guided almost 15 M.Tech. dissertations and a number of B.Tech. Projects. Her research areas include Digital Image Processing, MANET, Adhoc Networks, Video compression and Enhancement, Optimization using GA & PSO etc.