Novel Time Aware Regression Testing Technique

Harsh Bhasin
Computergrad.com
i_harsh_bhasin@gmail.com

Praveen
M.Tech. Scholar, BSAITM
Faridabad, India
praveensrcem@gmail.com

Anju Gera
A.P., BSAITM
Faridabad, India
anju.gera@faculty.anangpuria.com

***Abstract*— Regression testing comes into play when changes are made in the software. It is not possible to re-run all the previous test cases therefore, a minimization technique is required in order to reduce the test case suit. The present work proposes a time aware minimization technique to accomplish the task. The technique is verified by taking 3 KLOC professional management system developed by Sahib Soft. The proposed work takes into consideration the shortcomings in the existing techniques and presents a theoretically sound model to handle the anomalies of the existing techniques. The initial results obtained are encouraging.**

Keywords- Regression testing, Test case minimization, Time aware, Cost Aware.

I. INTRODUCTION

Software testing is immensely important as the absence of proper testing process leads to escalation in the maintenance cost and at times non deliverance of the product. Once software is created, many changes are done in its life cycle. This calls for the re-run of the existing test case suit. This is referred to as regression testing. This may lead to exponential increase in the testing time. So reduction in the test case suit is needed. There are three techniques to accomplish the task: Selection, Minimization and Prioritization. Selection technique considers the test cases relevant only to the modification. Minimization technique reduces the test case suit and considers the previous test cases also, whereas prioritization prioritizes the test cases so that maximum coverage can be attained in the limited time.

It may be stated that, heuristics can also be used for the test case reduction problem. The technique proposes the use of an artificial intelligence approach called Genetic Algorithms in order to achieve the goal. So as to get the crux of what has been proposed so far, an extensive literature review has been carried out. This helped in crafting a technique which is theoretically sound and at the same time efficient. The technique is different from the existing ones in terms of the premise use for optimization. It uses a time aware approach with a radical change which helps attaining better optimization. The work is a part of the endeavor to extensively examine various testing techniques and peripherals [1, 2, 15, 16, 17, 18].

The rest of the paper has been organized as follows. Section 2 presents literature review. Section 3 gives an idea of Genetic Algorithms. Section 4 presents proposed technique and Section 5 discuss the conclusion and future scope. The technique is both novel and affective and is bound to change the way the Time Aware premise is perceived.

II. LITERATURE REVIEW

An extensive literature review has been carried out in order to find the gaps in the exiting technique and to propose the new technique. The literature review has been carried out in accordance with guidelines proposed by Kitchenham [3]. It may be stated that the papers of high quality journals have been selected and as far as possible the review has been unbiased. An extensive review of the existing techniques forms the foundation stone of a good research. Therefore an extra effort has been made in order to carry out the appraisal. Table 1 depicts the various techniques for test case suit minimization

SR NO	NAME OF AUTHORS	REF NO	TECHNIQUE	VERIFICATION
1	Scott McMaster at. Al	4	Call-stack coverage criterion.	Three applications from TerpOfficeSuite written in JAVA and European Space Agency Application in C
2	H. You at. Al	5	Modern Integer Liner Programming Solver	Aircraft altitude separation monitor, priority queue scheduler
3	Michael Harder at. Al	6	Dynamic generates operational abstraction(which describe behavior)area and stacking technique for comparing test suit generation	8 C Programs were taken for both faulty and correct version operational abstraction.
4	Wolfgang Grieskamp	7	Finite State Machine, Abstract State Machine	30 slot version of the spec with relevant, then we obtain 24 hyper states
5	Andreas Leitner,	8	Slicing	Eiffel Base, Data struct
6	Shin Yoo at. Al	9	Pareto efficiency	Print tokens, Print tokens2, Schedule, Space
7	Sebastian Elbaum at. Al	10	Tradeoff exist between fine granularity and coarse prioritization Anova Analysis	Eight C programs as subject
8	Gregg Rothermel at. Al	11	An improved rate of fault detection regression test and let debuggers begin their work earlier	7 programs, each with a variety of modified versions
9	W Eric Wong at. Al	12s	A test set is generated block minimized test sets have a size/effectiveness advantage Reduction of large sized test set to a moderate and moderate to minimal	Block coverage constant on the fault detection strength of the resulting minimized test set.

Table-1 Technique for test case Minimization

III. GENETIC ALGORITHMS

Genetic algorithms are used when search is to be carried out in a large space. The search algorithms in such huge spaces become computationally inapt. So, a heuristic algorithm is needed which achieves optimization. It may be stated, though, that robustness is compromised in order to accomplish the goal. Genetic Algorithms are adaptive heuristic search algorithms which are based on Charles Darwin theory of the survival of the fittest [13]. The basic building block of GA is a chromosome. Each chromosome represents a solution to the problem and is composed of a string of cells of finite length [14]. There are many types of chromosomes, the work uses binary chromosomes. Each chromosome has a fitness value associated with it. The fitness value determines the aptness to the problem in hand. The numbers of cells in a chromosome depend on the problem at hand [15]. The number of chromosome can be an optimal number large enough to contain a feasible solution [16]. GAs are known to imitate nature. The natural process of reproduction involves the amalgamation of features of a male and a female to produce another creature which may fare better in the society due to this clubbing of attributes. In GAs this task is accomplished by crossover. Now, imagine a society with just one kind of inhabitants. It will be a lame society. Such society will enrich only if a person with radical ideas is born. He is referred to as a mutant. In genetic algorithms this task is carried by mutilation operator.

The process of GA can be summarized as follows:
Step 1: Randomly generate an initial population.
Step 2: Compute fitness value f (p) for each individual.

Step 3: Select two members from the current population. The chance of being selected is proportional to the chromosomes fitness value.
Step 4: Perform mutation.
Step 5: from amongst the above chromosomes select the chromosomes having fitness greater than the threshold.
Step 6: Repeat step 2 until satisfying solution is obtained or ineffectual value is reached.

IV. PROPOSED WORK

In the section that follows, the following representations are used.
The test cases are represented by T, T={t1, t2,, t3.... tn}.
The Requirements are represented by R, R={r1, r2, r3……. rn}.
Every test case t maps to some r such that the whole test suit satisfy all the requirements.
As per the literature review the relation R×T is depicted by matrix. If tk requires rj then row j and column k has value 1.It may be stated that the running time of each test case t is known .The Time -Aware Regression Testing Problem has been defined by as follows:
Minimize $\sum_{j=1}^{n} c_i x_j$
Subject to $\sum_{n} S_{ij} x_j \geq 1$,i=1,2…..m , j=1,2….n
The aim of the work is to minimize the total running time and optimize the cost. There is a problem, however, in the premise. It may be stated that minimizing cost does not always lead to best test cases. The work proposes amalgamation of prioritization techniques in order to achieve the best of both worlds. The cost will now be defined as
$\lambda = K1 \times Ci + K2 \times Pi$
where Ci represents the cost in terms of running time. Pi represents the number depicting the priority of the test case. It may be stated that Pi directly proportional to the priority .K1 and K2 may be determined by the statistical regression analysis of the history data. For each chromosome, the fitness value may be calculated as
$Fit(i) = 1/1+e^{-\lambda}$
Now, the problem reduces to an optimization problem where by the search space is large and fitness value of each chromosome is known. Thus, it becomes apt for applying genetic algorithms.
The process applied for regression test minimization is as follows: We will start by first Mapping Test Case a chromosome and then evaluating the fitness of that particular chromosome. Then we will proceed by applying Crossover Operator on resultant data and applying Mutation. Thereafter selecting the children as per fitness value and applying Roulette Wheel Selection in order to select the chromosome and repeating of this process is done until coverage is attained. The process is depicted in Figure 1.

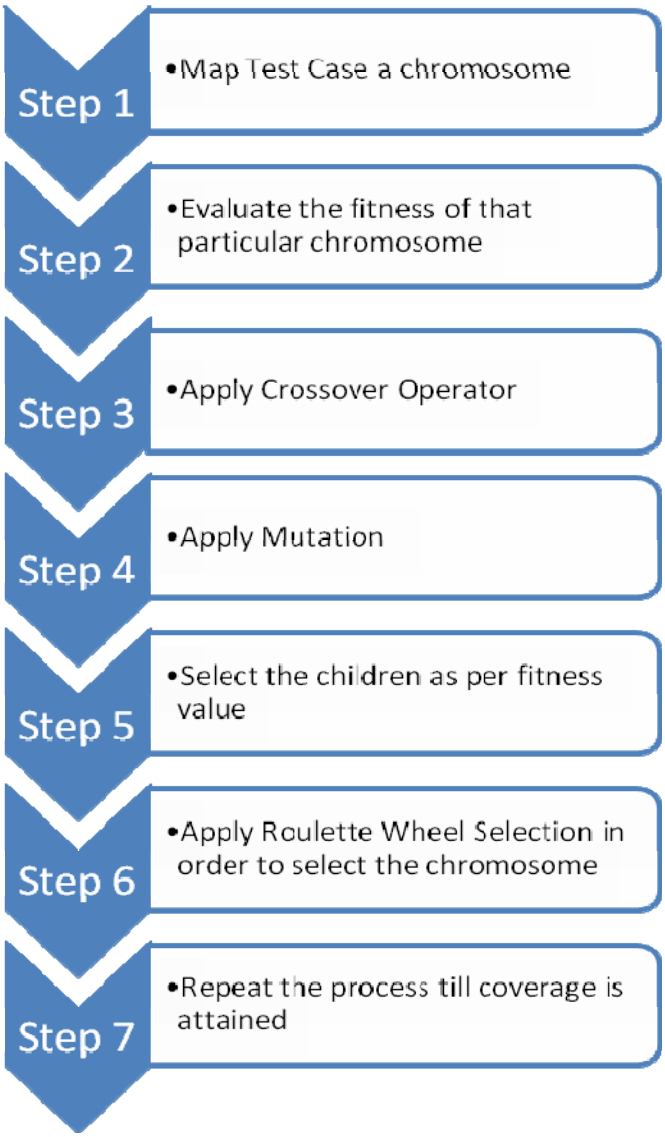

Figure 1: Proposed Work: Regression Testing

V. RESULTS AND CONCLUSIONS

It was found that the cost in terms of time required to execute a test case is an incorrect premise and hence requisite modifications were needed in order to rectify the technique. The work proposed has accomplished the task. The proposed model is being verified on an Enterprise resource planning system developed by Sahib Soft. The results obtained so far are encouraging and paves way for the further enhancement of the technique in order to fill the remaining gaps. A set of 10 or programs have been selected for further modification.

VI. REFERENCES

[1] Harsh Bhasin , Manoj, Regression Testing Using Coupling and Genetic ,IJCSIT, Vol. 3 (1) , 2012, 3255 – 3259.
[2] Harsh Bhasin, Ankush Goyal, Deepika Goyal, Software Architecture Based Regression Testing, IJCSE, Vol. 5 No. 04, 2013.
[3] Emelie Engstrom,Per Runeson,Mats Skoglund, A systematic review on regression test selection techniques, Journal Information and Software Technology, Volume 52 Issue 1, January, 2010, Pages 14-30.
[4] Scott McMaster and Atif M. Memon, Call Stack Coverage for GUI Test-Suite Reduction, In Proceedings of the 17th IEEE International Symposium on Software Reliability Engineering (ISSRE 2006), Nov. 2006.
[5] Hwa-You Hsu and Alessandro Orso, MINTS: A General Framework and Tool for Supporting Test-suite Minimization, ICSE '09 Proceedings of the 31st International Conference on Software Engineering, pages 419-429 .
[6] Michael Harder, Michael Harder, Michael D. Ernst, Improving Test Suites via Operational Abstraction, n Proceedings of the 25th International Conference on Software Engineerin, 2003.
[7] Wolfgang Grieskamp, Yuri Gurevich, Wolfram Schulte, Margus Veanes, Generating Finite State Machines from Abstract State Machines,.

[8] Andreas Leitner, Manuel Oriol, Andreas Zeller,Efficient Unit Test Case Minimization. In ASE '07 Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering, 2007.
[9] Shin Yoo,Mark Harman, Pareto efficient multi-objective test case selection, Proceedings of the 2007 international symposium on Software testing and analysis,Pages 140-150.
[10] Sebastian Elbaum, Alexey G. Malishevsky, Gregg Rothermel, Prioritizing test cases for regression testing, Proceedings of the 2000 ACM SIGSOFT international symposium on Software testing and analysis , Pages 102-112.
[11] Gregg Rothermel, Roland H. Untch, Chengyun Chu, Mary Jean Harrold, Test Case Prioritization: An Empirical Study, Proceedings of the International Conference on Software Maintenance, Oxford, UK,1999.
[12] W. Eric Wong, Joseph R.Horgan,Aditya P. Mathur,Alberto Pasquini, Test Set Size Minimization and Fault Detection Effectiveness: A Case Study in a Space Application, Proceedings of the 21st International Computer Software and Applications Conference.
[13] Harsh Bhasin, Rohan Mahajan, Genetic Algorithms Based Solution To Maximum Clique Problem,IJCSE, Vol. 4 No.,2012
[14] Harsh Bhasin and Neha Singla. Genetic based Algorithm for N - Puzzle Problem, International Journal of Computer Applications , August 2012. Published by Foundation of Computer Science, New York, USA.
[15] Harsh Bhasin et. al., Test Data Generation usinfg Artificial Life, International Journal of Computer Applications, Vol. 67, issue 12, 2013.
[16] Harsh Bhasin et. al., Regression testing using Fuzzy Logic, International journal of Computer Science and information technology, Vol. 4, issue 2, 2013.
[17] Harsh Bhasin et. al., Implemenattion of Regression testing Using Fuzzy Logic, International Journal of Application or Innovation in Engineering and Management, Vol. 2, Issue 4, 2013.
[18] Harsh Bhasin et. al., Orthogonal Testing Using genetic Algorithms, International Journal of Computer Science and Information Technology, Vol. 4, Issue 2, 2013,