Comparison View of Different Technologies Used to Prioritize the Test Case in Regression Testing

[1]Dr. Amit Verma, [2]Rohit Bajaj, [3]Ishadeep Kaur Luthra
[1,2,3]Dept. of CSE, Chandigarh Engineering College, Landran (Mohali), India

Abstract
Testing is the part of the software development and come into use in different methods. Our research is about regression testing and various techniques that have been developed so far. In regression testing various test cases are developed and testing is performed repeatedly on it. The main objective of regression testing is to find errors before the implementation of the system.
Various techniques being developed on regression testing to increase the performance are like Greedy algorithm, Genetic algorithm, Particle Swarm Optimization, clustering approach and many more. In these various techniques the main concept is of prioritization of test cases, so that regression testing is performed on only validated test cases, thus improving the performance and being time efficient.
This can be used for analyzing various tools and techniques that can be implemented with existing algorithm and hybrid model may be proposed

Keywords
ACO, Test Case Prioritization, Regression Testing

I. Introduction

Software Testing is a process to gauge the system with the desire of finding errors before the implementation of the system. It is the action that is performed under the controlled conditions. Once the system has been developed there may be bugs present in the system, to overcome those bugs few changes need to be made in the code developed. These changes made in the code can produce different result, undesired to the result of the system. To ensure that the desired result has not been changed software testing is need to be done. Software testing is oriented toward Error-detection, defects and fundamental weakness of the application code. Software testing has 3 main purposes i.e. Verification, Validation and Defect.

- The purpose of verification confirms that software meets its specification. It ensures that software developed is able to be implemented to perform specific functions.
- The process of validation ensures that the software meets the business requirement. It provides the identifiable activities to the customer.
- A defect is conflict among the actual and expected result. The defect is caused due to the fault introduced in the specification, design or development phase.

Regression testing is performed on software whenever the modification is made in the software, to ensure that the modification made does not lead to non satisfactory results. It provides confidence that the software behaves correctly and has not impacted on the quality of the software. Regression testing is performed by re-testing the whole program of partial functionalities when the problem was first discovered, to sure new changes made to the application have not caused failure to the component. The purpose of regression testing is to catch bugs that have been introduced while programming or while making changes in the software. Such test can be performed manually or on small projects. It is very time consuming process and engineers have to perform whole test again and again, moreover performing whole test repeatedly become expensive.

II. Literature Survey

T As software evolves, software engineers perform regression testing on it to validate new features and detect whether corrections and enhancements have introduced new faults into previously tested code. In practice, engineers often reuse all of the existing test cases to test the modified version of the software system; however, this retest-all approach can be expensive.
Manika et al. [3] proposed 3 phase approach to solve test case prioritization. In the first phase they removed redundant test cases by simple matrix operations. In second phase test cases are selected from test suits such that selected test cases represent the minimal set which covers all faults and also at the minimum execution time. In the third phase Priority is obtained by calculating the ratio of fault coverage to the execution time of test cases, and rest of test cases which are not selected in phase 2, are now added in the test suite in sequential order. The proposed MOPSO outperformed other approaches like No Ordering, Reverse Ordering and Random Ordering as MOPSO achieves maximum fault coverage, maximum value of APFD and minimum execution time. Muthusamy et al. [22] prioritize the test cases depending on the business impact, importance and frequently used functionalities. Their algorithm is based on analysis of the percentage of test cases performed to find the faults and on APFD metric's results. Chen et al. [4] uses classification algorithm for early fault detection rate and to guide the scheduled process based on code change information. During the process of the software development and maintenance, they gathered the code change information and corresponding introduced faults. From that data collected, they found out that in specific program modules, code changes were prone to introduce new faults. Therefore they could conjecture that there exists some specific relationship between the code change and the fault detection probability, and they wanted to use classification algorithms to mine those relationships. Shweta Kanwar et al. [18] presented the analysis of the regression test prioritization technique to reorder test suites in time constraint environment along with the sample runs on various programs. Their analysis concluded that the ACO (Ant Colony Optimization) finds better orderings at higher values of the time constraint (TC). In this paper, they validated the technique proposed in Test Case Prioritization Using Ant Colony optimization and implemented in Ant Colony Optimization for Test Case Selection and Prioritization. Their results achieved, encouraged the facts that; the test suite selection and prioritization technique reduces the size of test suite, the execution time is reduced comparatively, the correctness achieved is very high for most of the test programs, and the ordered test suite potentially enables to discover the faults earlier.
Carlson et al.[5] implemented new prioritization technique that incorporates clustering approach which utilize code coverage, code complexity, and history data on Real faults. They have designed and conducted empirical studies using an industrial software product, Microsoft Dynamic Ax, which contain real faults. R.Krishnamoorthi et al. [6] aims to improve the effectiveness of

Regression Testing by ordering the Test Cases so that the most beneficial are executed first. The technique of genetic algorithm is used. The proposed technique prioritizes subsequences of original test suites so that the new suites which are in run within a time constrained execution environment will have a superior rate of fault detection compared with randomly prioritize test suites. The time constrained test case prioritization problem can be reduced to the NP-complete zero/one knapsack problem [7 - 8] which can often be efficiently approximated with a Genetic Algorithm (GA) heuristic search technique. Just as genetic algorithms have been effectively used in other software engineering and programming language problems such as test generation [10], program transformation [11], and software maintenance resource allocation [12], they have demonstrated that they also prove to be effective in creating time constrained test prioritizations. Sbastium et al. [16] proposed to measure the rate of fault detection as quickly as possible within the testing process, an improved rate of fault detection can provide faster feedback on the system under test, and let software engineers begin locating and correcting faults earlier than might otherwise be possible. Their work raised several additional questions. First, they examined only general prioritization", which attempts to select a test case order that will be effective on average over a succession of subsequent versions of the software. In regression testing, they concerned with a particular version of the software, and wished to prioritize test cases in a manner that will be most effective for that version. In this context, they were interested in "version-specific prioritization" and they were interested in the effectiveness of this prioritization relative to versions that contain multiple faults. Secondly, the techniques they examined were all operated at relatively fine granularity i.e., they involved instrumentation analysis and prioritization at the level of source code statements.

III. Competitive Analysis

S no.	Author's Name	Tools used	Technique	Pros	Cons
1	Manika et al.[3]	MATLAB R2009b	Multi objective Particle Swarm Optimization (MOPSO)	Non Complex Self Adaptation of parameters	Not much in Demand Effective for small data.
2.	Muthusamy et al. [22]	JDK 1.6	Average Percentage of Fault Detected (APFD)	Implemented on Working Application	Too many Test Cases. Long and Old Process Time consuming and cost inefficient.
3.	Chen et al. [4]	C programming language	Classification Algorithm	Two Classifiers are used. One to test code change and running result in previous version and second to estimate fault detection probability	Long Process, include lot of steps Different condition need different classifiers. Conclusions may not be applicable for programs written in other programming language
4.	Shweta Singhal et al. [18]	C++	Ant Colony Optimization (ACO)	ACO finds better ordering at higher value of the Time Constraint Provide high reduction in the size of test suites	Each Test case have large difference in Time Constrain. 100% of correctness was not achieved in spite of getting the time constraint of 400 seconds, which is considered very high time constraint.
5.	Carlson et al.[5]	Matlab	Clustering Approach	Test Cases with common properties have similar fault detection ability. Better for small data Clustering approach is applied	All the test cases don't have common property For large data to figure out Test Cases common property is difficult. Fine clusters are not formed.
6.	R.Krishnamoorthi et al. [6]	Java Programming Language	Genetic Algorithm (GA)	Population Based search method Ease of implementation Effective to solve highly non-linear optimization problem	Long time process If a test case is taken it show 6 faults in 9 minutes similarly other test cases also take a lot of time to find faults. So all the test cases cannot be run at a single time.

Proposed _ Algorithm

```
Begin
{
    Initialize the Data Set Ds
    For Ds=1, Ds< Dl, Ds++
    {
      Apply Acquisition
    }
     Perform Clustering Using K-Mean
      If Ds ϵ Sc
                  //Where clustered data
     Apply Ndt
    //Where is Ndt Novel Density Function
     Calculate Accuracy and Convergence
}
```

In this approach, we can work on improving the test case prioritization on the basis of clustering approach. A novel density based k-means clustering approach can be used to make clusters of different test cases on the basis of statement coverage. Then, prim's algorithm can be used to find out the minimum path between different test cases according to their coverage information. Test cases are selected from every cluster; which may have maximum coverage information. According to Prim's algorithm, we can find the tree of test cases; this technique reduces the test cases numbers. Only those test cases will be selected which have maximum coverage information. It will reduce the effort, cost and time also.

IV. Conclusion

In this paper we compare the different technologies use to prioritizing the test cases for regression testing. The pros & cons have been discussed of different technologies been used till date many different algorithms have been implemented to reduce the effort of prioritizing the test cases and deducting default in specific time constraint. Though there are few cons in new technology but better result have also been produce.

References

[1] G. Antoniol, M. D. Penta, M. Harman,"Search-based techniques applied to optimization of project planning for a massive maintenance project", In Proc. of the 21st ICSM, Washington, DC, USA, 2005, pp. 240-249.

[2] Y. Shi, R.C. Eberhart,"A modified particle swarm optimizer," Proceedings of the IEEE World Congress on Computational Intelligence pp. 69-73, 1998.

[3] Manika Tyagi, Sonika Malhotra,"Test Case Prioritization using Multi Objective Particle Swarm Optimizer", 2014 IEEE

[4] Xiang Chen, Zhaofei Tan, Jian Xia, Pengfei He,"Optimizing Test Case Execution Schedule using Classifiers", Journal of Software, Vol. 9, No. 10, October 2014

[5] Ryan Carlson, Hyunsook Do, Anne Denton,"A Clustering Approach to Improving Test Case Prioritization: An Industrial Case Study", 2011 27th IEEE International Conference on Software Maintenance (ICSM) 47

[6] R.Krishnamoorthi, S.A.Sahaaya Arul Mary, "Regression Test Suite Prioritization using Genetic Algorithms", Department of Computer Science and Engineering, Bharathidasan Institute of Technology Anna University, Trichy-24, India.

[7] M. R. Garey, D. S. Johnson,"Computers and Intractability: A Guide to the Theory of NP-Completeness", W. H. Freeman & Co., New York, NY, USA, 1979.

[8] G. Rothermel, R. J. Untch, C. Chu,"Prioritizing test cases for regression testing", IEEE Transaction on Software. Eng., Vol. 27, Issue 10, 2001, pp. 929-948.

[9] P. Chu, J. Beasley,"A genetic algorithm for the multidimensional knapsack problem. Journal of Heuristics", Vol. 4, Issue 1, 1998, pp. 63-86.

[10] R. P. Pargas, M. J. Harrold, R. R. Peck,"Test-data generation using genetic algorithms", Software Testing, Verification and Reliability, Vol. 9, Issue 4, 1999, pp. 183-282.

[11] D. Fatiregun, M. Harman, R. M. Hierons,"Evolving transformation sequences using genetic algorithms", In Proc. of 4th SCAM, 2004 , pp. 66-75.

[12] G. Antoniol, M. D. Penta, M. Harman,"Search-based techniques applied to optimization of project planning for a massive maintenance project", In Proc. of the 21st ICSM, Washington, DC, USA, 2005, pp. 240-249.

[13] Thillaikarasi Muthusamy, Dr. Seetharaman.K, "Effectiveness of Test Case Prioritization Techniques based on Regression Testing", International Journal of Software Engineering & Applications (IJSEA), Vol. 5, No. 6, November 2014.

[15] K. Mumtaz, Dr. K. Duraiswamy,"A Novel Density based improved k-means Clustering Algorithm – Dbkmeans", (IJCSE) International Journal on Computer Science and Engineering, Vol. 02, No. 02, 2010, pp. 213-218

[16] Sebastian Elbaum et al,"Prioritizing Test Cases for Regression Testing", Technical Report # TR-UNL-CSE-2000-0005, Also presented in International Symposium of Software Testing and Analysis, 102-112, August 2000.

[18] Bharti Suri, Shweta Singhal,"Analyzing Test Case Selection & Prioritization using ACO", ACM SIGSOFT Software Engineering Notes, Vol. 36, No. 6, November 2011.