International Journal of Engineering Research & Management Technology

ISSN: 2348-4039

July-2019 Volume 6, Issue-4

Email: editor@ijermt.org www.ijermt.org

REGRESSION TESTING USING GENETIC ALGORITHMS

Harendra Singh[1], Arvind Kumar[2]

[1,2]Assistant Professor
Department of MCA
SRMIST, Ghaziabad, India

ABSTRACT

Regression testing is used to test the software against the side effect appeared due to the changes made in the software. The correct functionality inherited from previous version should be validated. But for this testing time and resources are limited. To overcome these constraint test cases are prioritized On the bases of some criteria by which we execute the test case in a sequence by which we can ensure maximum coverage. This paper proposes an algorithm based of genetics which reorders the test cases in a intelligent way that detects maximum faults in less time. This is the use of genetic algoritm in regression testing.

KEYWORDS

Genetic Algorithms, Regression Testing, Test Case Prioritization, Software testing, Algorithms,

INTRODUCTION

Quality software can be developed by efficiently and testing thoroughly. Various changes are made in maintenance phase. When new functionality is added many bugs can arise. Regression testing is done by re-executing the test cases that validates that the software has no added side effect by the change [1]. Regression testing ensures that modification do not affect the previous functionality. After each modification new test cases are added and the obsolete test cases are removed from the test suit but the size of test suit increases test suit goes massive.

To execute all test cases is a costly process in regression testing. Test prioritization is the technique by which we can make sequence of test case execution in regression testing as the time is limited for this testing. In prioritization an order of test cases is made in important test cases are executed first. [2]. Here genetic algorithm is used in the technique of prioritization

This paper gives genetic algorithm based intelligent technique for prioritizing test cases in regression testing. Genetic algorithms are adaptive heuristic search algorithms that are based on Charles Darwin theory of the survival of the fittest [7]. Genetic Algorithms are used to solve optimization problem. The work proposes a technique that prioritizes subsequence of regression test suite in such a way, that its execution has a high rate of fault coverage when compared to rates of randomly prioritized test suite. The work reviews various techniques of test case prioritization.

The paper has been organized as follows. Section two discusses regression testing. Section three reviews various techniques of test case prioritization. Section four explains genetic algorithms. Section five presents the proposed framework and section six concludes.

REGRESSION TESTING

Software undergoes many changes during its maintenance phase. New features reflecting new customer requirements are added and some existing features are updated. Changes in the version of software, requires its retesting to confirm that new features have not affected the functioning of derived features. This is known as regression testing. Regression testing is selective retesting of the system or components to verify that modifications have not caused an unintended effects and the system or component still complies with its specified requirements [5]. Regression testing is required in the following scenarios [6].

- Software code is changed in accordance to changed requirements of customer.
- Enhancement of Functionality of software.

- When fixing of defects in previous versions.
- At removal of obsolete functionality from previous version.

Test cases of regression test suite are classified in three categories [3].

- Test instances that exams all software program functionality.
- Test cases that specializes in the capability that can be stricken by the new adjustments.
- Test instances which can be brought to the take a look at suite to test better software program capability.

As new test cases are added the size of regression test suite grows rapidly. Regression is the most expensive phase of software testing and is required whenever the software functionality is updated. Regression testing may be done manually or can be automated with help of playback tools [6].

Re-executing all the test cases of regression test suite is a costly process. Due to restricted time and resources, subset of test cases is executed. Techniques of regression testing are selection, minimization and prioritization [4]. Test case selection technique selects and re-executes only a subset of test cases that focuses on changed module [4]. Test case minimization technique reduces the test case suite based on some criterion [4]. Test case prioritization technique reorders the test cases based on some criterion in such a way that most important test cases are executed within time and resources [5].

Regression Test

1- Test case selection
2- Test case minimization
3- Test case prioritization.

Test case selection and minimization techniques reduce the number of test cases of regression test suite, while test case prioritization does not alter the number of test cases. Test cases are assigned high, medium and low priorities on the basis of given criterion. Prioritization criteria may be based on test case history [6], coverage [3], fault severity [4], customer requirements [5] or costs [6]. The paper proposes a technique which prioritizes the test cases based on maximum fault coverage by using genetic algorithms.

LITERATURE REVIEW

A detailed review has been carried out in order to find the gap in the existing literature. The review has been carried out on the guidelines of Kitchenham [3]. Some of the work is as below.
Zing Li, et. Al. provided five seek algorithms for test case prioritization of regression suite [4]. Three greedy set of rules strategies and two metaheuristic seek strategies and had been studied. Techniques included hill mountaineering, Genetic Algorithm, greedy, extra grasping and a couple of-most reliable greedy algorithms. The paintings accomplished an empirical study to evaluate 5 algorithms on 6 programs starting from 375-11149 strains of code. Results showed that extra greedy and a couple of- most beneficial set of rules have pleasant universal results. In one of the works, cuckoo seek was used for check case choice and prioritization [5]. Cuckoo search is a unmarried parameter optimization problem that is inspired by means of the obligate brood parasitism of cuckoo species. The paintings prioritized take a look at instances on criteria of number of faults blanketed in minimum time.

Elbaum et. Al. [6] proposed various check case prioritization techniques to improve the fee of fault detection.14 take a look at casse prioritization techniques were categorised into 3 companies i.E control strategies, declaration degree techniques and function level strategies.

The paintings resulted that first-rate granuality strategies are better than direction granularity strategies. Cagatay Catal discussed various crucial troubles for regression testing [3]. The work brought ten quality practices for take a look at case prioritization and their position in successful software trying out. Yoo and Harman classified take a look at case prioritization strategies into 9 groups [4]. They had been coverage based totally, distribution primarily based human based, probabilistic technique, records based totally, requirement based, and version based totally, value conscious technique and others.

Genetic algorithms have been utilized by Konsaard and Ramingwong to perform the undertaking of take a look at case prioritization [4]. The work reordered take a look at cases consistent with standards of maximum code coverage. The paintings as compared performance of proposed technique with five different strategies on the basis of average percent of condition protected and execution time. In the paintings with the aid of Harsh Bhasin and Manoj, a Genetic Algorithm based prioritization approach became proposed [4]. The method used coupling quantity calculator to assign health cost to test cases. Test cases prioritized the use of this approach led to excessive fault detection charge in assessment to others.

GENETIC ALGORITHM

Genetic algorithm (GA) is heuristic seek algorithm stimulated by means of Charles Darwin concept of herbal evolution. GA is an shrewd seek method and is used to resolve optimization problems [5]. It performs random searches via a pool of answer and pursuits to find best opportunity in step with a few given criterion. Basic steps of a GA are as follows.

Step 1: Selection of initial populace- The first step is to pick out a finite set of individuals (chromosomes) for a given problem. These individuals represent possible solution. This results in the generation of preliminary populace.

Step 2: Evaluation of every person- Each member of the population is evaluated in line with some objective function. The goal feature (also referred to as health characteristic) assigns a health cost to each individual member of populace based on a few given criterion.

Step 3: Generate new populace- A new populace of solution is generated by way of applying GA operators. These are reproduction, crossover and mutation [6].

- Reproduction operator- Chromosomes are selected from initial population and are entered to mating process by the usage of duplicate operator. The choice strategies are Roulette wheel selection, Rank choice, Tournament selection and Boltzmann choice [6].
- Crossover operator- crossover operator takes decided on chromosomes as inputs and generates new offspring. It mimics the organic recombination of chromosomes. For two given chromosomes X=x1, x2…… xn and Y=y1, y2……yn and a crossover factor say k, crossover operator generates two new chromosomes X1=x1……xk ,yk+1…….Yn and Y1=y1……yk , xk+1…….Xn. Crossover operator is classed as simple crossover, double crossover and N-point crossover [6].
- Mutation operator- Mutation operator modifies the genetic makeup of offspring. It randomly replaces some of the bits of child chromosome. Different strategies of mutation operator are flipping, interchanging and reversing [6].

Step 4: Stopping criteria- Duplicates are removed from the new generation of chromosomes. If optimized solution is obtained stop, else go to step 2. Stopping conditions for GA are maximum generations, Elapsed time and unchanged fitness [6].

PROPOSED WORK

The work proposes use of genetic set of rules in regression trying out. Test instances in regression check case matc become colossal. Re-executing all the test cases is not possible inside confined time and resources. In such scenarios, check instances need to be prioritized. Test instances of regression take a look at suite are intelligently reordered so that most effective crucial take a look at instances are re- carried out without affecting the overall quality of checking out. This can be useful if the trying out has to be stopped in advance due to restrained price and resources. The paintings prioritizes take a look at instances the use of genetic algorithm on the basis of rate of fault detection. Fault detection depicts how fast a test case can hit upon most range of faults inside trying out technique assuming that all the faults are of identical severity. This facilitates in imparting.

A remarks in an early level thereby helping the developers to fix them on time. Steps of proposed work are as follows.

Test instances for every module are generated. The mission can be computerized the usage of take a look at era tools. This bureaucracy the initial populace of solution. Each test case is then assigned a health fee on the premise of wide variety of fault insurance standards. The take a look at case which covers more faults is assigned a more fitness price than the take a look at case which covers less quantity of faults. On the basis of health price, best check cases are decided on to shape a take a look at suite in this kind of way that execution of suite ends in general fault insurance in a module. In order to create new era of prioritized take a look at cases genetic operators are implemented. Crossover operator applied to test fits generates randomness in solution. In order to feature variation in new technology, mutation operator is implemented. Duplicate take a look at instances are eliminated and prioritized test cases are received. The method is summarized beneath.

- Generate test cases for each module(initial population)
- Assign fitness number to each test case based on fault coverage criteria
- Select best test cases leading to total fault coverage and create test suits
- Apply crossover operator
- Apply mutation operator
- Remove duplicate test cases
- prioritized test cases

CONCLUSION

Modification of software requires re-execution of all the check instances to validate that adjustments in a single module have now not changed accurate capability of others. Due to limited sources, all of the check instances can not be re-carried out within the improvement duration. Prioritization of take a look at cases is needed if you want to preserve quality of regression checking out. Test instances want to be prioritized in this kind of way that vital check cases are finished before final touch of software. This results in cost powerful regression checking out within restrained time and sources. The work prioritizes test cases on the basis of excessive fault coverage rate. The paper provides use of genetic set of rules that optimizes the mission of regression test case prioritization. The technique might be beneficial to both researchers and practitioners if the checking out needs to be stopped upfront due to loss of assets.

REFERENCES

1. Chauhan, N. 2010. Software Testing principles and practices. Oxford University Press.
2. Rothermel, G., et. al. 2001. Prioritizing Test Cases for Regression Testing. IEEE Transactions on Software Engineering. vol. 27, no. 10, pp. 929-948, DOI
3. Yoo, S., Harman, M. 2012. Regression testing minimization, selection and prioritization: a survey. Software Testing, Verification & Reliability. John Wiley and Sons Ltd. Volume 22 Issue 2, Pp. 67-120.
4. Srivastava, P., R. 2008. Test Case Prioritization. Journal of Theoretical and Applied Information Technology.pp 178-181
5. Engström, E., et. al. 2011. Improving Regression Testing Transparency and Efficiency with History-Based Prioritization -- An Industrial Case Study. Software Testing, Verification and Validation (ICST). IEEE Fourth International Conference on 21-25 March 2011 Page(s):367 - 376 Berlin,IEEE, DOI:10.1109/ICST.2011.27.
6. David Leon and Andy Podgurski. 2003. A Comparison of Coverage-Based and Distribution-Based Techniques for Filtering and Prioritizing Test Cases. In Proceedings of the 14th International Symposium on Software Reliability Engineering (ISSRE '03). IEEE Computer Society, Washington, DC, USA, 442
7. Sejun Kim and Jongmoon Baik. 2010. An Effective Fault Aware Test Case Prioritization by Incorporating a Fault Localization Technique. In Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM '10).
8. ACM, New York, NY, USA, Article 5, 10 pages
9. R. Kavitha, V.R. Kavitha, N. Suresh Kumar. 2010. Requirement Based Test Case Prioritization. In Proceedings of International Conference on Communication Control and Computing Technologies, 826–829.
10. Ramasamy, K., Mary, S.A., 2008.Incorporating varying requirement priorities and costs in test case prioritization for new and regression testing IEEE, Computing, Communication and Networking.
11. Kitchenham, B.A. et. al. 2010. Systematic literature reviews in software engineering .A tertiary study, Information &

Software Technology .INFSOF , vol. 52, no. 8, pp. 792-805, 2010

12. Li. Z., et. al. 2007. Search Algorithms for Regression Test Case Prioritization. Software Engineering, IEEE transactions vol 33 issue 4.
13. Nagar, R., et. al. 2015. Test case selection and prioritization using cuckoos search algorithm. International conference on Furistic Trends on Computational Analysis and Knowledge Management. IEEE.
14. Elbaum, S., et. al. 2000. Prioritizing test cases for Regression testing. Presented in international symposium of software testing and analysis. 102-112. Catal, C. 2012. The Ten Best Practices for Test Case Prioritization. ICIST. Springer. pp 452-459.
15. Konsaard, P., Ramingwong, L. 2015. Total coverage based regression test case prioritization using genetic algorithm. 12th International conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology. IEEE.
16. Shivanandam, S. N., Deepa, S. N. 2012. Principles of Soft Computing. Second edition. Wiley India.
17. Bhasin, H., Manoj. 2012. Regression Testing Using Coupling and Genetic Algorithms. International Journal of Computer Science and Information Technologies. Vol. 3(1).
18. IEEE std. definition of Regression Testing.
19. Rajal, J. S., Sharma, S. 2015. A Review on Various Techniques for Regression Testing and Test Case Prioritization. International Journal of Computer Applications. Volume 116, No. 16.
20. Pressman, R.S. 2010. Software engineering: a practitioner's approach. McGraw-Hill Higher Education.