*International Journal of Research in Advent Technology, Vol.6, No.6, June 2018*
*E-ISSN: 2321-9637*
*Available online at www.ijrat.org*

# A Pruning Based Regression Testing Approach to Optimize Industrial Software

Ramandeep Kaur[#1], Manpreet Singh[#2], Rishideep Singh[#3], Maninderpal Singh Dhaliwal[*4]

[1]*Research Scholar,* [2,3,4]*Assistant Professor*

[#] *Department of Computer Science & Engineering, North West Institute Of Engineering &Technology, Dhudike, Moga, India*

[*]*Department of Computer Science, Ludhiana College of Engineering and Technology, Katani Kalan, India*

*Email:* [1] ermdhaliwal85@gmail.com, [2] manpreet_cse@northwest.ac.in, [3] hod_it@northwest.ac.in, [4] ermdhaliwal85@gmail.com

**Abstract-** Software testing plays a vital role in the software development life cycle. Software has to pass the various phases from initial development to testing and final usability. In this process, there are a lot of amendments in the modules of software. Testing is the key step to verify the accurate working of software but it is not possible to test entire software each time after amendments. To save time and resources, there is the concept of regression testing that can be applied only on the module/phase of the software that has been updated. This ensures the successful implementation of changes in the software modules. Although regression testing is a successful concept with its subcategories of retest entire module, regression test selection, regression suite minimization and test case prioritization but sometimes it becomes difficult in case of program of large code files and industrial software to detect the specific module that can be consider for regression testing. This may be due to insertion of fresh software module or under-covering of regression. In that case, regression testing becomes expensive, time consuming and risk-oriented. To overcome this problem, we have applied the pruning method to optimize regression testing. In this research work, a mathematical model of pruning regression testing is proposed for the industrial applications to save time, cost and reduce the risk factor. In this pruned regression testing only required modules of the software are tested other are pruned (trimmed). In this proposed mathematical model, two different case scenarios are tested for the five different parameters. The effect is analyzed for the individual change and combinational change in parameters. The efficiency of the proposed concept is evaluated in terms of time, cost and risk factor for the regression with pruning approach and regression testing without pruning approach. From evaluated results, it is noted that regression testing with pruning approach dominates over regression testing without pruning approach for all the parameters of time, cost and risk factor.

**Index Terms-**Regression Testing; Test Case Generation; Software Testing; Software Development; Software Development Life Cycle

## 1. INTRODUCTION

Regression testing is the concept of software testing that ensures the functionality of software after the amendments in software modules, change in software platform, insertion of fresh code phase etc [1]. After each amendment in software, a software tester ensures the working of software by practical observational results. Regression testing is a challenging task for the tester due to increase in number of test cases with insertion of new feature code, costly & time consuming process in case of entire test suite execution, need to cover maximum test software with minimum test suite, increasing test cases & bugs after each modification of software. Regression testing steps also briefed with fig. 1.

Regression testing is not only helpful due to fixing bugs and errors but it also benefits the clients in terms of saving their energy, time and reducing software failure risk factor, save manpower, ensures quality etc. Moreover, it is also beneficial for the industrial and business professional to ensure the quality and productivity of software which is must to file a patent of the software [2] [3].

Regression testing is of four types: test entire case, test case selection [4], test case prioritization [5] and test suite reduction [6]. Testing of entire case based approach is most time consuming and expensive approach as it demands to retest entire software with all the possible test cases. Test case selection approach selects only the useful test cases which are necessary after the modification of program. It selects only the test modules in which modification has been made. Test case prioritization approach prioritizes the useful test cases that ensure the software quality. Test suite reduction approach reduces the number of test cases with useful and effective cases that can ensure maximum software quality with minimum test cases. The insertion of new module based program test codes, large program codes of billion lines makes, presence of software bugs etc makes the regression testing difficult to implement. In this paper, pruned regression testing is applied to uncover these cases in which test cases are not easy to observe. Pruned

regression testing approach trimmed the unusual test cases and considers only the useful test cases. In this approach the hidden cases are uncovered by comparing the test cases available before amendments. The additional test cases after the consideration of new modules are further added. Further, this proposed autonomous mathematical model approach selects Rest of the paper is organized in the following manner: Section 2 presents the literature review on the methods of regression testing, Section 3 defines the motivation and problem statement of research work, only the useful from overall cases and trimmed the test cases that belong to unaffected modules of software. Test cases are observed for the individual and combination effect of five software parameters. The performance of the system is observed before and after the pruning regression testing based on the evaluation terms of time, cost and risk factor.

The work on test suite minimization is presented by different authors as mentioned. Ahmed et al. [14] have used Cuckoo Search (CS) approach for the test suite minimization and evaluated results with variation in

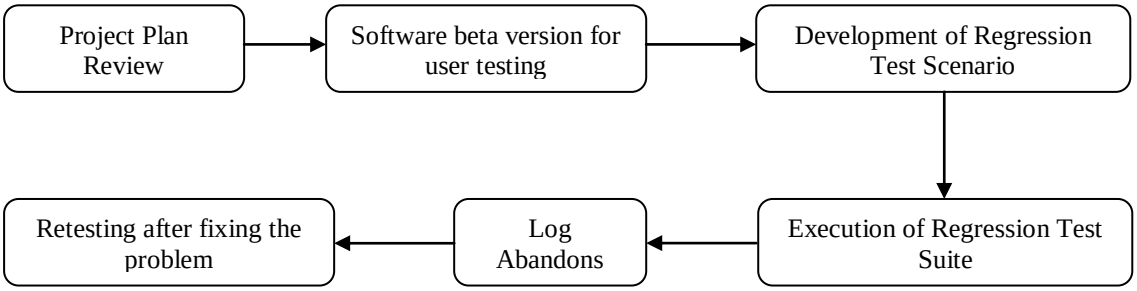


Fig. 1. Flowchart of Regression testing steps

Section 4 presents the research methodology, Section 5 evaluates the result values in terms of cost, time and risk factor and Section 6 concludes the paper with some future directions.

## 2. LITERATURE REVIEW

This section presents the literature review on the existing concepts of different regression testing approaches. Here, some of the latest research for the different regression testing techniques based on regression test selection, test suite minimization and test case prioritization are presented.

The research work on the test case prioritization techniques is presented by following authors. Saraswat and Singhal [7] have hybridized the concepts of genetic algorithm and particle swarm optimization for the test case prioritization. Hettiarachchia et al. [8] have used fuzzy expert system based approach for the test case prioritization. Sharma and Singh [9] have performed the test case selection and prioritization using ant colony optimization. Farooq & Nadeem [10] have used while box technique of mutation testing for the test case prioritization with the coverage of additional mutants killed. Ozturk [11] have proposed IMProved Bat Inspired Test Case Prioritization and observed the superiority of results in terms of APFD as compared to other considered techniques. Abid & Nadeem [12] have improved the multiple criteria based approach with "Additional" strategy for the test case prioritization. Chen et al. [13] have used adaptive random sequence based clustering algorithms for the test case prioritization.

the parameters of CS algorithm. Huang et al. [15] have proposed the concept of fuzzy expert system (FES) with traditional approaches of Greedy algorithm, GRE and Harrold-Gupta-Soffa (HGS) approach. Concept is experimented on nine real time programming codes. Results are evaluated in terms of execution time, fault detection capability etc. Zhang et al, [16] have modified the concept of ant colony optimization with quantum inspired evolutionary approach and quantum gate to reduce the test suite cases with different simulation results. But authors have observed the interruption in local search and slower coverage rate. Liu et al. [17] have used K-medoids clustering approach after observing the drawbacks of k-means clustering algorithm. Further, Indumathi and Madhumathi [18] have used Maximum Frequent Test set for the test suite reduction and genetic algorithm for the test case prioritization. Results are observed in terms of Average Percentage of Fault Detection metric (APFD. Yamuc et al. [19] have used greedy algorithm and genetic algorithm for the test suite minimization. Authors have achieved efficient performance of the proposed approach for suite reduction. Kabir et al. [20] have used modified flower pollination algorithm for the test suite reduction. Here, flower pollination algorithm is modified by updation of step length parameter at each iteration.

Further, research work on regression test selection has been presented. Vedpal and Chauhan [21] have applied slicing technique and OPDG concept for the test case selection of object oriented programs. Chauhan et al. [22] have presented the Program Model based Regression Test case Selector (P-ReTEST) approach for the test case selection and evaluated the

code based results using open software platform of Graphviz, Cygwin and Eclipse. Do et al. [23] have applied novel test case selection approach on android applications and presented the results with an example of implementation on android application. Refai et al. [24] have used model based approach of FIGA (Fine Grained Adaption) for the test case selection. Dahiya et al. [25] used the UML model based activity, sequence and class diagram for the test case selection. Hafez et al. [26] have used the concept of cache memory to store the information of potential faulty files and test only those selected files. Legunsen et al. [27] have used the static regression selection technique of STARTS (STAtic Regression Test Selection) which is a Maven plug-in. STARTS approach uses graph based mapping for the selection of test cases by removing faulty cases. Further, Refai et al. [28] have used fuzzy logic based RTS (Regression Test Selection) approach for the selection of test cases. This approach is UML model based approach in which classification is performed on existing test cases to classify reusable and retestable cases as per the changes in the activity diagram. Wongwuttiwat and Lawanna [29] have improved the concept of test case selection by recovering the faulty test cases and reduction of possible test cases. Authors presented the methods of novel proposed model, filtering based selection, code coverage based and original regression test. Romano et al. [30] have used Simple Information Retrieval Regression Test Selection Approach (SPIRITuS) for the test case selection. SPIRITuS outperformed in cost effective and outperformed in comparison with other considered techniques.

## 3. MOTIVATION AND PROBLEM STATEMENT

In software testing, there are various phases of development and amendments as per client requirement. During these phases of amendments, it is not possible to test entire software again and again as it increases cost and effort. But the software working and quality are also important. So, regression testing introduced to reduce the effort and cost with testing only the respective module in which amendments are made. But in case of programs with large codes, it becomes difficult to find the bugs and amendments of code features. Moreover, test suite minimization and increased frequency of regression testing after each updation motivates us to further improve the regression testing. To overcome this problem, pruned regression testing is proposed. In pruned regression testing, only the useful modules and test cases of software testing are considered. Other test cases are trimmed as useless cases as those cases do not guarantee the software quality of new updations. Moreover, the existing concepts discussed in literature review also not much effective in case of large software as test suite minimization & case selection becomes difficult. Test cases can only be prioritized after the successful detection of effectual test cases which is not a cake walk. So, we have proposed an autonomous pruned regression testing bases mathematical model to improve the software quality and reduction in cost, time and risk with the help of five parametric testing.

## 4. RESEARCH METHODOLOGY

This section presets the proposed approach of pruned regression testing. In this pruned regression testing, there are two modules of manual result generation and autonomous pruning. There is the consideration of five parameters in this approach with individual and combinational testing scenarios. The overall work flow of proposed concept is presented in fig. 2. The step by step research methodology of proposed work is presented below.

### 4.1. *Manual result generation*

Step 1: Initialize the system for the manual generation of testing results.
Step 2: Consider the testing parameters and select the parameters that are affected with the amendments.
Step 3: Generate the test cases with the help of available test scenarios of individual and combinational approach.
Step 4: Evaluate the results with the generated test cases.
Step 5: Check the evaluated results and observe the following considerations:

- Observe and show the affected parameters
- Observe and present the results of testing scenario based on individual and combinational scenarios
- Observe and evaluate the generated test cases.

### 4.2. *Autonomous pruning*

In autonomous pruning, test scenario selection (individual or combinational scenario) is a random process for the five parameters. For large organizational software's, there may be need to manipulate the system in autonomous process based on affected modules after amendments. The stepwise autonomous process is explained as below:
Step 1: Initialize the system for the autonomous pruning based testing.
Step 2: Apply the concept of autonomous pruning for the evaluation of results with test cases. In this process, selection of parameters and test scenarios is automatic.

Step 3: Check the evaluated results and observe the following considerations:

- Observe and show the affected parameters
- Observe and present the results of testing scenario based on individual and combinational scenarios
- Observe and evaluate the generated test cases.

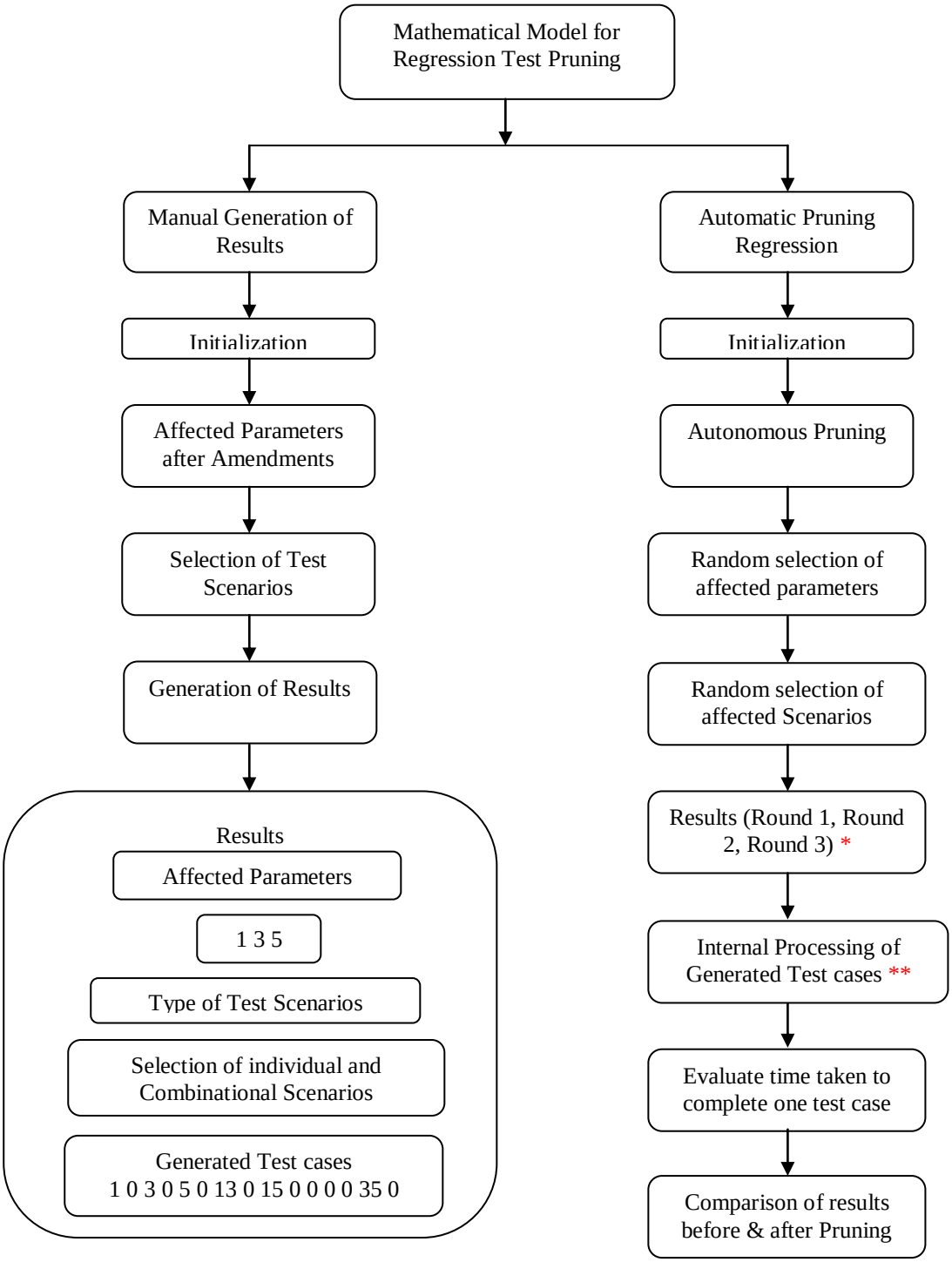


Fig. 1. Work flow of Research Methodology

* Different rounds (Round 1, Round 2, Round 3) of results are presented in Fig. 3.

** The internal Processing of generated test cases is shown in Fig. 4.

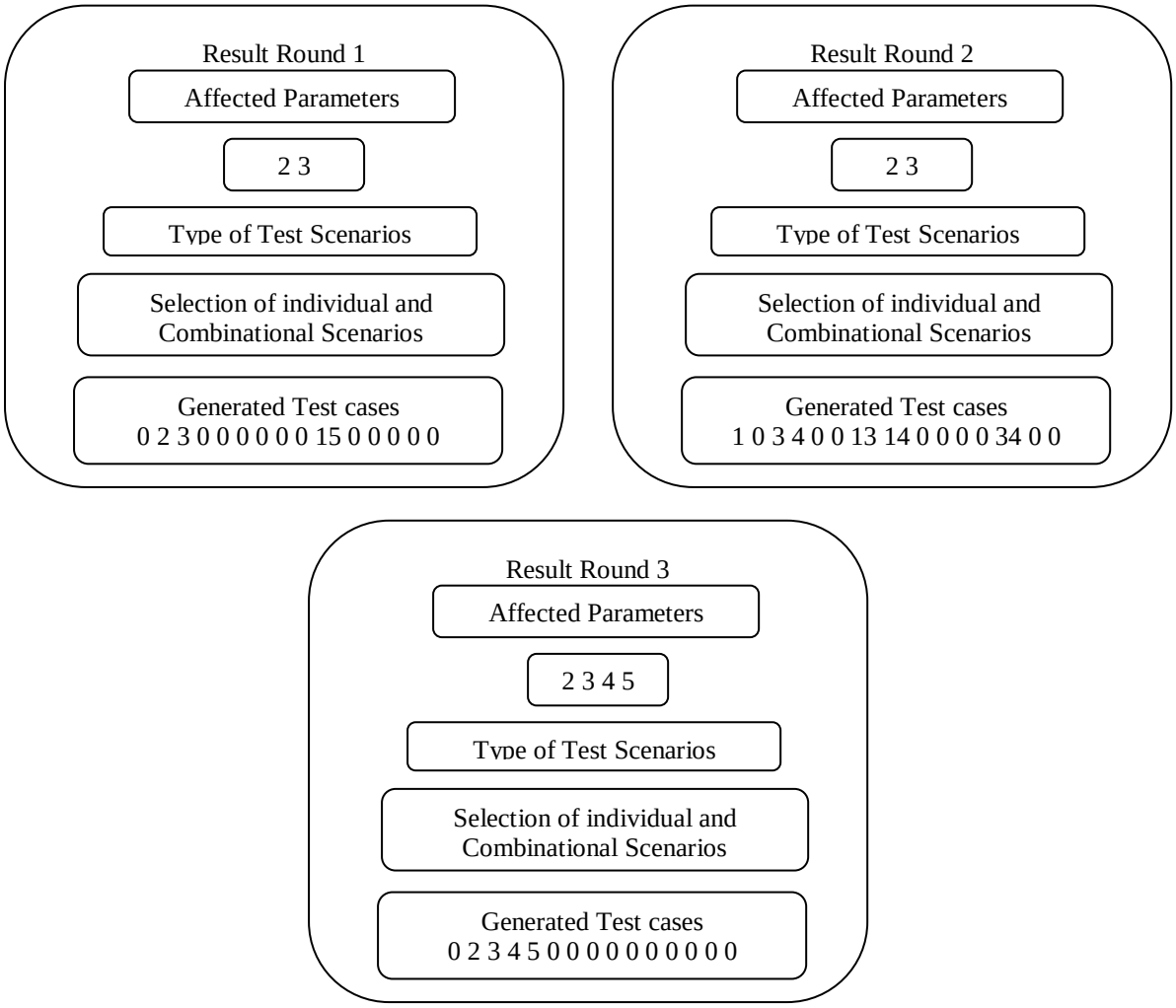


Fig. 3. Results Rounds of Autonomous Pruning Regression Testing

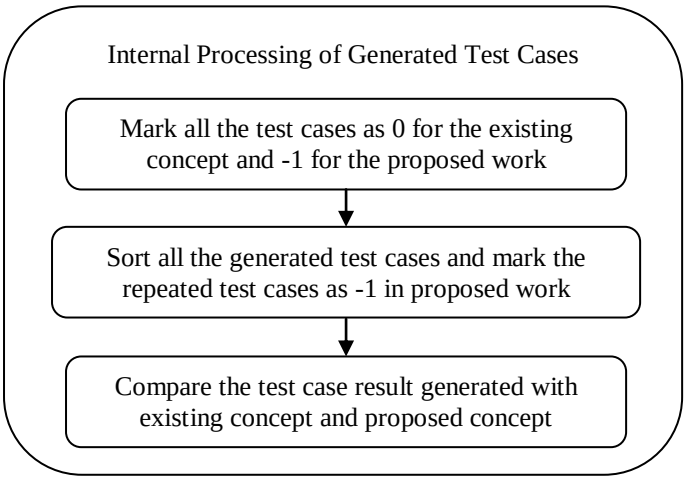


Fig. 4. Workflow of Internal Processing of Generated Test Cases using Autonomous Pruning Regression Testing

## 5. RESULT AND DISCUSSION

In this research work, the proposed autonomous regression testing approach is implemented on MATLAB simulation tool. A GUI based system is generated for the autonomous test case generation based on five parameters and two scenarios. These five parameters and two test case scenarios are presented in table 1. The GUI based system is shown in fig. 5.

Based on the above mentioned parameters and test scenarios, test cases are generated using autonomous pruning approach. Further, results are evaluated in terms of Cost, risk factor and time. Comparison is performed for the use of regression testing approach with and without the use the pruning concept. Time can be defined as the consumption of time for the successful generation of final test cases. Cost is the total wok cost for the effectual test case generation. Risk factor is the risk involves in the software modules after the successful implementation of proposed approach. Lesser the values of these three time, cost & risk indicates the more optimized results of test cases. The evaluated results in terms of cost, time and risk are presented in fig. 6.

Table 1. Test Case Scenario and Parameters

| Parameters | Individual Test Scenario | Combinational Test Scenario |
|---|---|---|
| P1 Parameter | P1 Parameter | P1 Parameter, P2 Parameter |
| P2 Parameter | P2 Parameter | P2 Parameter, P3 Parameter |
| P3 Parameter | P3 Parameter | P3 Parameter, P4 Parameter |
| P4 Parameter | P4 Parameter | P4 Parameter, P5 Parameter |
| P5 Parameter | P5 Parameter | P1 Parameter, P3 Parameter |
| | | P1 Parameter, P4 Parameter |
| | | P1 Parameter, P5 Parameter |
| | | P2 Parameter, P4 Parameter |
| | | P2 Parameter, P5 Parameter |
| | | P3 Parameter, P5 Parameter |

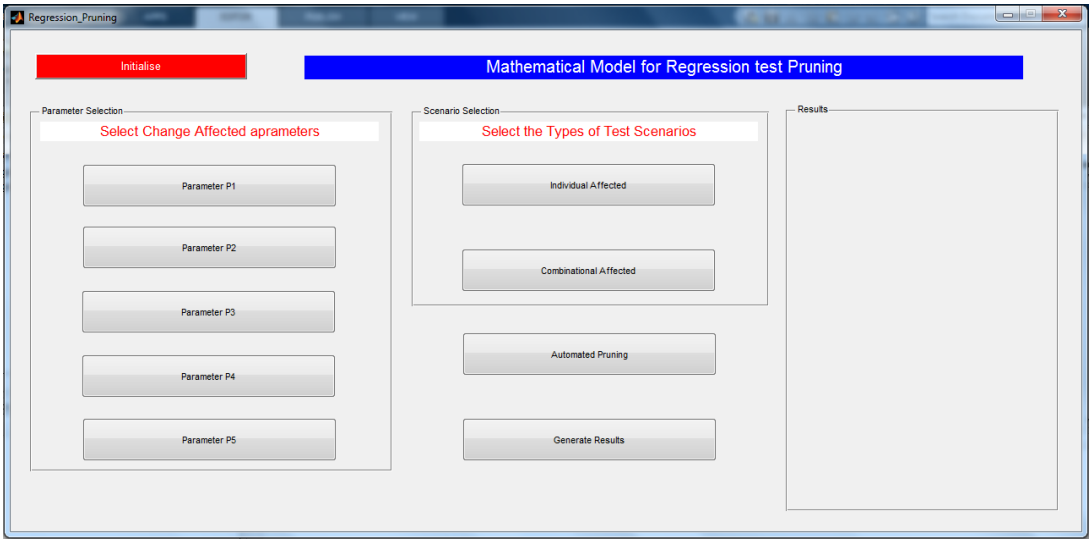


Fig. 5. GUI of Proposed Pruning Regression Testing

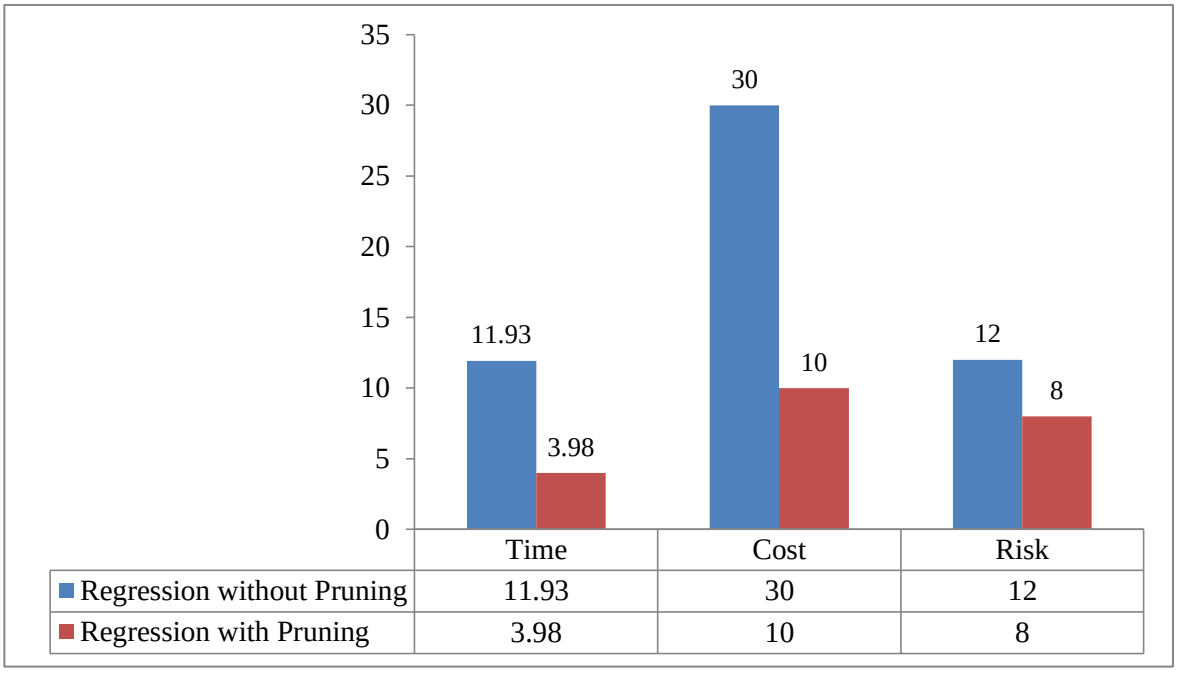


Fig. 6. Comparison of Regression Testing Approach with Pruning Concept (Proposed Concept) and Without Pruning Concept (Existing Concept)

As shown in fig. 6, the time consumed by the proposed approach is 3.98 which are much less than time consumption by existing concept which is 11.93. The cost of proposed concept is 10 which is also lesser than cost of existing concept which is 30. Risk factor of proposed concept is also reduced value of 8. From evaluated results in terms of time, cost and risk, it can be observed that proposed pruned regression testing approach is efficient enough for the autonomous efficient test case generation and quality assurance.

## 6. CONCLUSION AND FUTURE SCOPE

Regression testing has been an important phase of software development to test and assure the successful implementation of amendments in software code and modules. The categories of regression testing are efficient enough to retest the successful working of software. But, sometimes it becomes difficult in case of industrial large code software to find the useful test case. For those cases, we have proposed pruned regression testing approach. In this approach, the useless test cases are trimmed after the evaluation of useful test cases that can ensure the quality and successful working of software. Five parameter based two test case scenarios has been used and autonomous results are evaluated. Results for the generated test cases are evaluated in terms of time, cost and risk factor. From evaluated results, it can be noted that proposed concept is capable to reduce the time, risk and cost as compare to existing concept in which pruning is not considered.

For future work, the proposed concept can be further considered for the autonomous sequence recognition system. The proposed concept can also be improved using the computational intelligence concepts of fuzzy logic, neural network etc. for the optimization of time, cost and risk factor.

## REFERENCES

[1]. Leung, H. K., & White, L. (1989): Insights into regression testing (software testing). In Proceedings: Conference on Software Maintenance, IEEE, pp. 60-69.

[2]. Onoma, A. K., Tsai, W. T., Poonawala, M., & Suganuma, H. (1998): Regression testing in an industrial environment. Communications of the ACM, **41**(5), pp. 81-86.

[3]. Wong, W. E., Horgan, J. R., London, S., & Agrawal, H. (1997): A study of effective regression testing in practice. In Proceedings: 1997 The Eighth International Symposium on Software Reliability Engineering, IEEE, pp. 264-274.

[4]. Graves, T. L., Harrold, M. J., Kim, J. M., Porter, A., & Rothermel, G. (2001): An empirical study of regression test selection techniques. ACM Transactions on Software Engineering and Methodology (TOSEM), **10**(2), pp. 184-208.

[5]. Elbaum, S., Malishevsky, A. G., & Rothermel, G. (2000): Prioritizing test cases for regression testing **25**(5), pp. 102-112.

[6]. Rothermel, G., Harrold, M. J., Von Ronne, J., & Hong, C. (2002): Empirical studies of test-suite reduction. Software Testing, Verification and Reliability, **12**(4), pp. 219-249.

[7]. Saraswat, P., & Singhal, A. (2016): A hybrid approach for test case prioritization and optimization using meta-heuristics techniques, In Proceedings: 2016 1st India International Conference on Information Processing (IICIP), IEEE, pp. 1-6.

[8]. Hettiarachchi, C., Do, H., & Choi, B. (2016): Risk-based test case prioritization using a fuzzy expert system. Information and Software Technology, **69**, pp. 1-15.

[9]. Sharma, S., & Singh, A. (2016): Model-based test case prioritization using ACO: A review. In Proceedings: 2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC), pp. 177-181.

[10]. Farooq, F., & Nadeem, A. (2017): A Fault Based Approach to Test Case Prioritization. In Proceedings: 2017 International Conference on Frontiers of Information Technology (FIT), pp. 52-57.

[11]. Öztürk, M. M. (2017): Adapting code maintainability to bat-inspired test case prioritization. In 2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA), IEEE, pp. 67-72,

[12]. Abid, R., & Nadeem, A. (2017): A novel approach to multiple criteria based test case prioritization. In 2017 13th International Conference on Emerging Technologies (ICET), pp. 1-6.

[13]. Chen, J., Zhu, L., Chen, T. Y., Towey, D., Kuo, F. C., Huang, R., & Guo, Y. (2018): Test case prioritization for object-oriented software: An adaptive random sequence approach based on clustering. Journal of Systems and Software, **135**, pp. 107-125.

[14]. Ahmed, B. S., Abdulsamad, T. S., & Potrus, M. Y. (2015): Achievement of minimized combinatorial test suite for configuration-aware software functional testing using the cuckoo search algorithm. Information and Software Technology, **66**, pp. 13-29.

[15]. Huang, C. Y., Chen, C. S., & Lai, C. E. (2016): Evaluation and analysis of incorporating Fuzzy Expert System approach into test suite reduction. Information and Software Technology, **79**, pp. 79-105.

[16]. Zhang, Y. N., Yang, H., Lin, Z. K., Dai, Q., & Li, Y. F. (2017): A Test Suite Reduction Method Based on Novel Quantum Ant Colony Algorithm. In 2017 4th International Conference on Information Science and Control Engineering (ICISCE), pp. 825-829.

[17]. Liu, F., Zhang, J., & Zhu, E. Z. (2017): Test-Suite Reduction Based on K-Medoids Clustering Algorithm. In 2017 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), pp. 186-192.

[18]. Indumathi, C. P., & Madhumathi, S. (2017): Cost aware test suite reduction algorithm for regression testing. In 2017 International Conference on Trends in Electronics and Informatics (ICEI), pp. 869-874.

[19]. Yamuç, A., Cingiz, M. Ö., Biricik, G., & Kalıpsız, O. (2017): Solving test suite reduction problem using greedy and genetic algorithms. In 2017 9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), pp. 1-5.

[20]. Kabir, M. N., Ali, J., Alsewari, A. A., & Zamli, K. Z. (2017): An adaptive flower pollination algorithm for software test suite minimization. In 2017 3rd International Conference on Electrical Information and Communication Technology (EICT), pp. 1-5.

[21]. Chauhan, N. (2015): Regression test selection for object oriented systems using OPDG and slicing technique. In 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom), pp. 1372-1378.

[22]. Chauhan, N., Dutta, M., & Singh, M. (2015): A Program Model Based Regression Test Selection Technique for Object-Oriented Programs. In 2015 Fifth International Conference on Communication Systems and Network Technologies (CSNT) pp. 918-924.

[23]. Do, Q., Yang, G., Che, M., Hui, D., & Ridgeway, J. (2016): Regression test selection for android applications. In 2016 IEEE/ACM International Conference on Mobile Software Engineering and Systems (MOBILESoft) pp. 27-28.

[24]. Al-Refai, M., Ghosh, S., & Cazzola, W. (2016): Model-based regression test selection for validating runtime adaptation of software systems. In 2016 IEEE International Conference on Software Testing, Verification and Validation (ICST) pp. 288-298.

[25]. Dahiya, S., Bhatia, R. K., & Rattan, D. (2016): Regression test selection using class, sequence and activity diagrams. IET Software, **10**(3), pp. 72-80.

[26]. Hafez, S., ElNainay, M., Abougabal, M., & ElShehaby, S. (2016): Potential-fault cache-based regression test selection. In 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), pp. 1-8.

[27]. Al-Refai, M., Cazzola, W., & Ghosh, S. (2017): A Fuzzy Logic Based Approach for Model-Based Regression Test Selection. In 2017 ACM/IEEE 20th International Conference on Model Driven Engineering Languages and Systems (MODELS), pp. 55-62.

[28]. Legunsen, O., Shi, A., & Marinov, D. (2017): STARTS: STAtic regression test selection. In Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering pp. 949-954.

[29]. Wongwuttiwat, J., & Lawanna, A. (2017): Performance improvement model of regression test selection. In 2017 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), pp. 49-53.

[30]. Romano, S., Scanniello, G., Antoniol, G., & Marchetto, A. (2018): SPIRITuS: a SimPle Information Retrieval regressIon Test Selection approach. Information and Software Technology, **99**, pp. 62-80.