International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-3, September 2019

Design of Proposed Test Case Prioritization Model for Test Sequence Generation and Validating Performance against Existing Methods

Priti, Kavita

Abstract: ***Regression testing is an effective tool or technique used for minimizing the efforts and resource consumption when a new version or the modification in the software project is tested. In such project, the testing is performed only on the selected modules or testcases. The sequence of the testcase execution also matter in regression testing. Researchers used various evolutionary, swarm based and mathematical methods for testcase prioritization and optimization. Researchers also used the fault based or coverage based measures for testcase prioritization. In this paper, the functional contribution of the recent research is provided. The research method, projects and the contribution of the work is discussed in this paper. In this paper, the framework of the proposed dynamic programming based weighted method is provided. The paper defined the work stages of the proposed model. The algorithm for test sequence generation is also provided in this paper. The analysis results of the work are provided against the greedy and dynamic programming based methods. The results are derived for cost and the number of testcases processed by the method. The results show that the proposed model improved the performance effectively.***

Index Terms: Dynamic Programming, Fault Based, Prioritization Methods, Regression Testing.

I. INTRODUCTION

The objective of regression is to test the regularly changing software in terms of new versions. The version related changes in software is the demand of customer and the market. The functional changes in the software can occur to fulfill the minor demands of customers. These small changes can affect the higher reflection across the modules of software projects. It is not cost and time effective to test the complete software system again. The regression testing provides the solution for the same problem. In regression testing, the selective testing is performed based on the actual changes happen in the software system. The quality of software project depends on the timely release of software without any possible fault. The regression testing enables the testing accurately in effective time. It identifies the modules and the testcases that requires more consideration and performs the selected testing instead of large scale. The functional behavior of the regression testing is shown in figure 1.

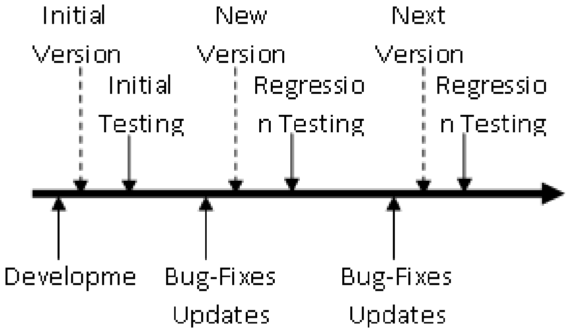

Figure 1 : Functioning of Regression testing

Figure 1 shows the functional behavior with associated stages of software updation and release. The figure shows that the developed software with earlier version is processed and after the initial testing it is available in field for use. Now as the requirement occur or some bug located in the software system, the bug-fixing is done by the development team. Before releasing the software in the field, the regression testing is performed over it for validating the software product. After the regression testing the software product can be released in the field. While using the product, if some other bug identified, the same process is repeated.

In the regression testing, the prior work is to define a testsuite that can test the changed modules effectively and accurately. The test suite must be capable to identify all the possible faults in this modified software system. Various parameters, methods and frameworks are investigated by the researchers for generating the testsuite. The testsuite must be effective to the statement, module or functional coverage. Once the test suite is generated, the next task is to assign the priorities to the test cases. These priorities are assigned based on various factors including fault detecting capabilities, module criticality etc. The priorities defined the consideration required for the specific module or testcases so that the fault free software will be produced. After assigning the priorities, some method is applied for generating the test sequence in which the testing will be carried out. Various optimization and analytical methods are present for taking the decision. After this regression testing the software product can be released.

Manuscript published on November 30, 2019.
* Correspondence Author
Dr. Priti*, Assistant Professor, Computer Science and Application Department, MDU, Rohtak, pritish80@yahoo.co.in.
Kavita, Research Scholar, Computer Science and Application Department MDU, Rohtak, Kavita.dahiya78@gmail.com

The paper has presented a dynamic programming based framework for optimizing the test sequence. The prioritization of the test cases is done using module dependency and fault based measure. In section I, the exploration of regression testing behavior is done with standard model specification.

In section II, the work and contribution of earlier researchers is provided for optimizing the prioritization and test sequence generation. In section III, the research method and contribution is listed in tabular form. In section IV, the graph processed and dynamic programming based method is defined for test sequence optimization. In section V, the results obtained from the work are provided graphically. In section VI, the conclusion and future scope of this paper are provided.

II. RELATED WORK

The regression testing methods are defined for identifying the effective test sequence. Various methods were defined by the researchers for assigning the priority and weights for optimizing the test sequence. In this section, some of the prioritization and path optimization methods defined by the earlier authors are provided.

Zhang et al.[1] defined the adaptive random sequence (ARS) based method for categorizing the test cases and applying the effective prioritization method on these categorized test cases. The fault based analysis on diverse testset cases were applied by the author for optimizing the test sequence. The distance metrics based ordering method was defined by the author for improving the fault detection and optimizing the generation of test sequence. Wu et al.[2] defined the similarity analysis based prioritization method ordering the sequence of program elements and its measuring was performed using execution count. The fault detection based method was defined for improving the coverage and reducing the risk in fault detection. The cost benefit based method was defined for generating the effective test sequence. Pilskalns et al.[3] provided the UML based modeling for performing the regression testing. The design change analysis and inconsistency observation was performed by the author for classifying the test cases. The design change analysis was observed using set of rule for generating the effective test sequence. Author defined the safe and efficient retest method.

Aman et al.[4] defined the hybrid method for recommending the test cases. The history and priority based analysis method was defined for improving the regression testing. The empirical study based method was defined by the author for selection of test data. Sun et al.[5] proposed the test case selection and prioritization based method for optimizing the testcases. The criteria of test case selection and prioritization is defined for dependency evaluation. The coverage ability and troubleshooting capabilities were defined by the author for prioritization of test cases. The method improved the effectiveness of test path generation. Mohapatra et al.[6] defined a new approach for test case reduction and generating the effective test path using genetic approach. The representative set based method was defined for removing the redundant test cases and improving the execution cost of test sequence formation. The testing criteria were defined by the author for optimizing the test sequence. Choudhary et al.[7] proposed the Pareto based harmony search method was defined for test case selection and optimizing the test sequence. The test adequacy based fault coverage analysis was defined for optimizing the test sequence. The Bat and cuckoo search based method was defined for optimizing the test sequence. Ding et al.[8] defined the greedy arithmetic approach for reducing the test suite and generating the pair-wise combination of test cases. Author applied the ant colony optimization method for handling the impurities and generating the effective test sequence with reduced test suite. The method reduced the regression cost and scalability for test sequence optimization.

Jiang et al.[9] defined the change point based minimum set formation for improving the regression testing with application specification. The path coverage testing was defined by the author. The method was defined for reducing the cost and size of regression testing and improving the cost effectiveness. Dobuneh et al.[10] defined the hybrid method with multiple criteria specification for defining the appropriate test sequence. The prioritization method was defined under the fault detection rate. The performance aware prioritization method was defined for improving the sequence of test sequence formation. The fault analysis based prioritization criteria were defined for improving the effectiveness of test sequence generation. Indumathi et al.[11] defined a study for using the prioritization method within genetic algorithm for optimizing the test sequence. The method uses the redundancy reduction method for improving the efficiency of test sequence formation. The maximum coverage and fault rate detection method was defined by the author for effective sequence formation. Zhang et al.[12] used the functional behavior analysis for optimizing the static path formation. The test case prioritization method was defined for defect detection in the regression testing. The prioritization specification and sequence generation algorithm was defined for reducing the cost of regression testing.

Salehie et al.[13] used the requirement based prioritization method for regression test formation. The goal-question metric based method was defined by the author for prioritization of the test cases. The goal driven method was defined for effective test sequence formation. Lawanna et al.[14] proposed the regression test reduction and optimizing the ordering of test sequence. The bug fixing and reduction method is defined for effective sequence formation. Zhang et al.[15] defined the code based regression sequence formation. The change detection, logical verification and pruning phase methods were optimized in this work. Based on this evaluation, the effective test suite generation method was defined by the author. Dhareula et al.[16] defined the exploratory study based regression test selection method for improving the usability and test sequence method. The code based method was defined for categorizing the group and requirement based methods. The code based technique was defined for selection of regression test cases. The method reduced the cost and improved the effectiveness of regression testing. Anwar et al.[17] defined the multi-objective optimization method for regression test sequence generation. The fuzzy rules were defined by the author for improving the test sequence. The size and cost reduction for test sequence was also defined by the author. Han et al.[18]

defined a heuristic model for prioritization of test cases and optimizing the test sequence. The model based method was defined for detection of faults and improving the effectiveness of test sequence formation.

III. TEST CASE PRIORITIZATION METHODS

In the recent years, various optimization methods are defined by the researchers for enhancing the capabilities of testcase prioritization and test sequence generation. These optimization methods either achieved the parameter specific enhancement or the effective research methods or frameworks are used. In this section, the research contribution of some recent research methods is defined. Table 1. The table contains the research methods their contribution and the source of projects taken by the researchers. The table shows that most of the researchers used the fault or the coverage based features while assigning the prioritization. For optimization various evolutionary, mathematical and swarm based methods were applied by the researchers. In this table, the functional description of these research methods and relative contribution is provided.

Table 1 : Functional Contribution of Test Case Prioritization and Optimization Methods

Author	Year	Research method	Source	Contribution
Manaswini et al.[19]	2019	Author defined cat swarm optimization method for test case prioritization.	Applied on open source projects like jtopas and jmeter	Improved the detection of faults in the cases accuratelly in lesser time.
Madhavi et al.[20]	2019	Fault adaptive functional coverage was achieved for effective regression testing. Load based analysis was performed for testing.	Clustered testscripts were analyzed	Improved the functional coverage and reduced the fault rate effectively.
Carballo et al.[21]	2019	Biased Random key based genetic algorithm was proposed for test case prioritization.	Conducted on 11 instances of real-world programs	Improved performance over permutational encoding based Gas.
Bruce et al.[22]	2019	Linear regression based prioritization method was defined for test case prioritization.	Conducted on flipkart website	Improved the fault detection, test coverage and significant gain in efficiency.
Sri et al.[23]	2019	Centroid based and hierarchical clustering method were applied for enhancing the prioritization method for regression testing	Numerical data processed for clustering	Distance based measure applied for effective clustering
Azizi et al.[24]	2019	A graph based framework was defined for mapping the prioritization task.	Applied on 20 versions of 4 open source projects	Method improved the statement coverage and efficiency of regression testing.
Musa et al.[25]	2018	Define a tool called HoccDanMafara which was integrated with GAP for optimizing the test case prioritization for object oriented programs	Tested on various object oriented programs	Achieved 27.75% gain in test efforts and 20.93% in efficiency.
Ren et al.[26]	2018	Two-layer model is defined with even handler tree and function calls graph based method defined for test sequence optimization.	Tested on benchmark websites for GUI testing	Effective Defect prediction and accuracy achieved for GUI testing
Fu et al.[27]	2018	Author estimated the risk, fault detection capability and coverage information using integer linear programming method for prioritizing test cases.	Applied on 9 C programs and 2 Java programs	Improved the average percentage of fault detection.

Tulasiraman 1 et al.[28]	2018	Defined the cost-cognigant history based artificial immune system for test case prioritization.	5 sequential versions of hospital management project	Improve average percentage of fault detection upto 20%.
Kerani et al.[29]	2018	Manual and automated slicing based test case prioritization method was defined and processed on maximum number of faults	Applied on 10 web projects	Improved fault detection rate and effectiveness of test prioritization
Ahmad et al.[30]	2018	Ant colony optimization based hybrid method was defined for prioritizing and optimizing test sequence.	Real time projects	Reduced the execution time and improved the fault detection rate
Morozov et al.[31]	2017	The error updation block and location of error propagation based Markov stochastic method was defined for test case prioritization	Applied on Simulink Models	Improved accuracy of fault detection effectively
Vescan et al.[32]	2017	Evolutionary algorithm was defined under fault, cost, requirement features for testcase prioritization	Applied on case study	Reduced cost of fault detection and improved the accuracy of fault detection.
Kim et al.[33]	2017	Failure history and error count based method was defined for prioritization	Applied on Apache, Camel and Tomcat open source projects	Improved accuracy and efficiency effectively

IV. PROPOSED TEST CASE PRIORITIZATION MODEL

In the previous section, various research and optimization methods are discussed with relative work contribution. Most of the researchers used the fault based methods or the statement coverage individually. Various optimization methods such as genetic, ACO, greedy etc. are also applied for optimizing the test prioritization and test sequence generation. In this paper, a dynamic programming based adaptive model is presented for test sequence optimization. The method is able to provide the effective solution for real time projects. The model used the fault analysis and module dependency as the main parameters for prioritizing the test cases. The tasks associated to this model in three main stages. The stages based heuristic model is shown in figure 2.

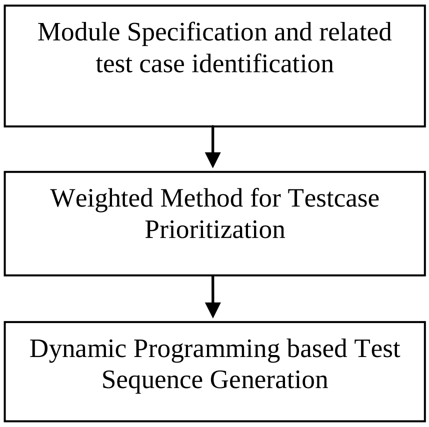

Figure 2 : Heuristic Model for Test Sequence Optimization

Figure 2 shows the three main stages of the proposed model. In the first stage of this model, the software projects are represented in the form of individual modules and the relationship between these modules is also defined. The control flow graph is generated for identifying the dependency between these modules. The connected flow of module is done for effective representation. Once the graph is generated, the testcases for each module are identified. The testcases are defined specific to the module individually. The categorization of the testcases is also done in the same stage based on the functional work.

In second stage, the testcases were analyzed respective to fault and dependency evaluation. The weights are assigned for generating the single prioritization vector. The impact of module and test cases was analyzed while assigning the priority to the test cases. Once the priorities are evaluated, these priorities are also integrated with control flow graph of the software project. In the final stage, these priority adaptive control flow graph is taken as input by the dynamic prioritization method and applied the back tracked evaluation for generating the optimized test sequence. The back tracked method performed the stage specific reanalysis for obtaining the more effective decision. The method is effective to reduce the cost of test sequence generation and improved the effectiveness over the existing model. The consideration of fault and dependency as the single measure improved the reliability and effectiveness of the proposed model.

The detailed flowchart of the work with all the process stages is shown in Figure 3.

Figure 3 shows that the proposed sequence generation method accepts the project modules as the main input. The testcases associated to the project are also listed. Based on this, the module specific testcases are identified. Now each of the testcase is analyzed under the fault and dependency features.

The fault features include the fault type, frequency and testcase criticality. The dependency features include the

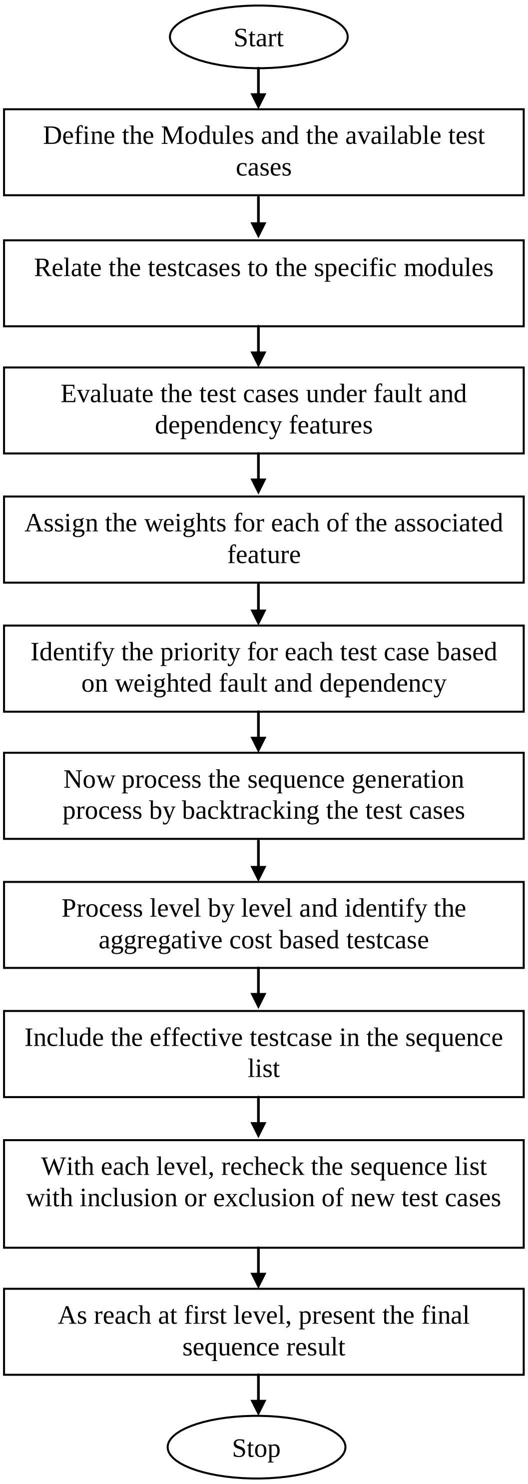

Figure 3 : Flowchat of Optimized Test Sequence Generation

level of module and its interconnection with previous and next modules. Each of the feature is assigned with specific weight and based on it the overall priority of the test of test is identified. Once the priorities are assigned, the process of test sequence generation begins. A reverse level based mapping is performed and the aggregative cost is identified. The minimum aggregative cost based test sequence is identified at each level. The process is continued till the first level and after first level the optimized test sequence is obtained.

Figure 3 shows the detailed view of the stages within the test sequence generation method. The complete work of this model is divided in two stages. In first stage, the test case prioritization is done. The algorithm for test case prioritization is provided in Table 2.

Table 2 : Algorithm for Testcase Prioritization

Step 1:	Define the Modules and the associated Testcases
Step 2 :	Analyze theTestcases specific for Each module and Identify the Dependency and Fault Features
Step 3 :	DependF=Testcase.prevconn*w1+ Testcase.nextconn*w2+ (MaxLevel-Testcase.Level)*w3 [Dependency is computed based on its connectivity with next and previous levels]
Step 4:	Estimate the Fault features including fault type, frequency and testcase criticality
Step 5:	TestcasePriority=DependF*w1+FaultF*w2 [Calculate the Priority of testcases based on the fault and dependency feature analysis]

Table 2 provided the detailed process for computing testcase scoring and priority assignment. The algorithm shows that the modules are the related testcases are obtained. The further analysis on these testcases is done based on the fault and dependency parameters. The dependency is further based on the level and the interaction with next and previous stage. The fault is described by fault criticality, fault type and frequency. These two parameters are processed collectively and the priority to the testcases is computed.

After the priority assignment, the final stage of this work is to generate the optimized test sequence. The dynamic programming based algorithm is defined for generating the sequence. The algorithm for test sequence generation is shown in Table 3.

Table 3 : Algorithm for Dynamic Programming based Test Sequence Generation

Step 1:	Accept the Testcases with assigned priority as input to the sequence generation method
Step 2:	Trace the Modules level based from last to first
Step 3 :	Identify the testcases for each module and levels
Step 4 :	Identify the testcase at the particular level with minimum priority
Step 5 :	Evaluate the cost of the inclusion of each testcases present at that level
Step 6 :	Identify the aggregative minimum cost testcase of that level by analyzing all previous levels
Step 7:	Add the minimum cost testcase in the optimized test sequence
Step 8 :	Evaluate the cost of test sequence and present it as the final result

Algorithm defined in Table 3 accepts the prioritized testcases as input and applied the dynamic programming method for effective sequence generation. The method backtraces the levels and the associated testcases. The cost of all the testcases are identified present at the same level. The aggregative cost is identified by combining the current cost and the previous levels minimum cost. The testcase with aggregative minimum cost is considered as the effective test case for that level. This process is repeated till the first level not arrived. With the first level, the optimized test sequence is obtained.

V. RESULTS

The proposed priority adaptive test sequence optimization method is investigated on five real time projects. These projects are object oriented projects with different number of modules and representing the general applications. These projects are Library system with 79 testcases, Hospital management system with 116 testcases, Inventory with 62 testcases, Exam Processing with 86 testcases and hotel management with 74 testcases. The proposed model is applied on these projects and the performance of the proposed model is analyzed in terms of number of testcases processed and the cost of the method. The comparative analysis is done against the greedy based and dynamic programming based methods. These methods are applied individually with fault based or dependency based prioritization method. The test case processed by these methods is shown in Figure 4.

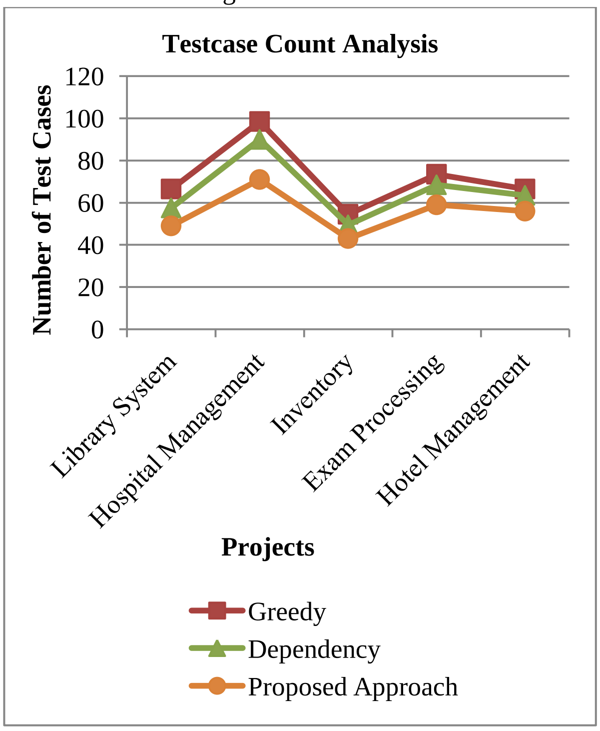

Figure 4 : Processed Test case Analysis

Figure 4 is showing the analysis results for observing the performance of the proposed model in terms of number of testcases processed. The average testcases processed by the Greedy and dependency based method is also provided in the figure. Here x axis represents the projects and y axis represents the number of testcases processed by these methods for each project. The line chart clearly shows that the number of testcases processed by the proposed method is much lesser than existing methods. The existing methods also used the single fault or dependency based analysis for testcase prioritization. Another analysis performed in this work is based cost based parameter. The cost based comparative analysis is provided in Figure 5.

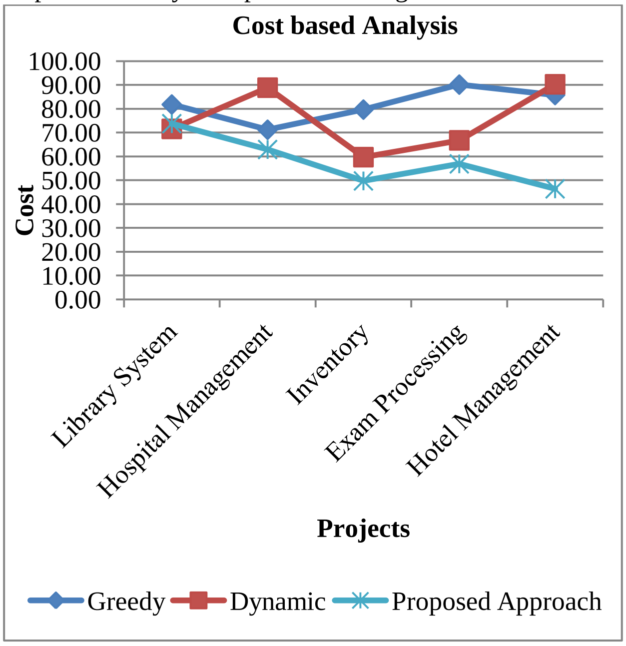

Figure 5 : Cost based Analysis

Figure 5 provides performance analysis of proposed model in terms of cost. The figure shows that the cost of test sequence generation is lesser than average cost of Greedy and dynamic programming methods. The line graphs shows that the proposed method optimized the regression testing method by effective test sequence generation.

VI. CONCLUSION

In this paper, detailed exploration of regression testing and its functional behavior is presented. Various research methods and the research contribution is listed in the form of table. The prioritization measures and the optimization methods used by the researchers in recent work are provided in this paper. The paper also provided a dynamic programming based model for optimization of test sequence. The three stage model is defined in this paper with exploration of functional behavior of each of the associated stage. The algorithm for each stage is defined for effective test sequence generation. The model is applied on five real time projects and the results shows that the proposed method reduced the cost and the number of test cases. The proposed method improved the performance of regression testing over greedy and dynamic programming methods.

REFERENCES

1. X. Zhang, X. Xie and T. Y. Chen, "Test Case Prioritization Using Adaptive Random Sequence with Category-Partition-Based Distance," 2016 IEEE International Conference on Software Quality, Reliability and Security (QRS), Vienna, 2016, pp. 374-385.
2. K. Wu, C. Fang, Z. Chen and Z. Zhao, "Test case prioritization incorporating ordered sequence of program elements," 2012 7th International Workshop on Automation of Software Test (AST), Zurich, 2012, pp. 124-130.

3. O. Pilskalns, G. Uyan and A. Andrews, "Regression Testing UML Designs," 2006 22nd IEEE International Conference on Software Maintenance, Philadelphia, PA, 2006, pp. 254-264.
4. H. Aman, T. Nakano, H. Ogasawara and M. Kawahara, "A Topic Model and Test History-Based Test Case Recommendation Method for Regression Testing," 2018 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), Vasteras, 2018, pp. 392-397.
5. S. Sun, X. Hou, C. Gao and L. Sun, "Research on optimization scheme of regression testing," 2013 Ninth International Conference on Natural Computation (ICNC), Shenyang, 2013, pp. 1628-1632.
6. S. K. Mohapatra and M. Pradhan, "Finding representative test suit for test case reduction in regression testing," 2015 International Conference on Computer, Communication and Control (IC4), Indore, 2015, pp. 1-6.
7. A. Choudhary, A. P. Agrawal and A. Kaur, "An Effective Approach for Regression Test Case Selection Using Pareto Based Multi-Objective Harmony Search," 2018 IEEE/ACM 11th International Workshop on Search-Based Software Testing (SBST), Gothenburg, 2018, pp. 13-20.
8. W. Ding, J. Kou, K. Li and Z. Yang, "An Optimization Method of Test Suite in Regression Test Model," 2009 WRI World Congress on Software Engineering, Xiamen, 2009, pp. 180-183.
9. B. Jiang, Y. Mu and Z. Zhang, "Research of Optimization Algorithm for Path-Based Regression Testing Suit," 2010 Second International Workshop on Education Technology and Computer Science, Wuhan, 2010, pp. 303-306.
10. M. R. Nejad Dobuneh, D. N. A. Jawawi, M. Ghazali and M. V. Malakooti, "Development test case prioritization technique in regression testing based on hybrid criteria," 2014 8th. Malaysian Software Engineering Conference (MySEC), Langkawi, 2014, pp. 301-305.
11. C. P. Indumathi and S. Madhumathi, "Cost aware test suite reduction algorithm for regression testing," 2017 International Conference on Trends in Electronics and Informatics (ICEI), Tirunelveli, 2017, pp. 869-874.
12. Z. Zhang, Y. Mu and Y. Tian, "Test Case Prioritization for Regression Testing Based on Function Call Path," 2012 Fourth International Conference on Computational and Information Sciences, Chongqing, 2012, pp. 1372-1375.
13. M. Salehie, S. Li, L. Tahvildari, R. Dara, S. Li and M. Moore, "Prioritizing Requirements-Based Regression Test Cases: A Goal-Driven Practice," 2011 15th European Conference on Software Maintenance and Reengineering, Oldenburg, 2011, pp. 329-332.
14. A. Lawanna, "Extended regression test model for test suite reduction," 2017 IEEE 6th Global Conference on Consumer Electronics (GCCE), Nagoya, 2017, pp. 1-5.
15. Z. Zhang, J. Huang, B. Zhang, J. Lin and X. Chen, "Regression Test Generation Approach Based on Tree-Structured Analysis," 2010 International Conference on Computational Science and Its Applications, Fukuoka, 2010, pp. 244-249.
16. P. Dhareula and A. Ganpati, "Identification of attributes for test case reusability in regression test selection techniques," 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, 2016, pp. 1144-1147.
17. Z. Anwar and A. Ahsan, "Multi-objective regression test suite optimization with Fuzzy logic," INMIC, Lahore, 2013, pp. 95-100.
18. X. Han, H. Zeng and H. Gao, "A Heuristic Model-Based Test Prioritization Method for Regression Testing," 2012 International Symposium on Computer, Consumer and Control, Taichung, 2012, pp. 886-889.
19. Manaswini B, Rama Mohan Reddy A; "A Cat Swarm Optimization Based Test Case Prioritization Technique to Perform Regression Testing", International Journal of Recent Technology and Engineering, Volume-8, Issue-1,pp 2677-2682, 2019
20. Karanam Madhavi, G. Bhavyasri, G. Ramesh, Ch. Mallikarjuna Rao, Lavanya Gottemukkala, "Load Effectiveness on Coverage-Technique for Test case Prioritization in Regression Testing", International Journal of Innovative Technology and Exploring Engineering, Volume-8 Issue-7, pp 711-715,, 2019
21. P. Carballo, P. Perera, S. Rama and M. Pedemonte, "A Biased Random-Key Genetic Algorithm for Regression Test Case Prioritization," *2018 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, Gudalajara, Mexico, 2018, pp. 1-6.
22. John Bruce. E, T. Sasi Prabha, "Lrpt- Linear Regression Based Prioritization Techniques for Early Fault Detection", International Journal of Engineering and Advanced Technology, Volume-8, Issue-2S2, pp 237-241, 2019
23. Gayam Rupa Sri, Addepalli Leela Supriya, Enukonda Raja Avinash Reddy, Venkata Naresh Mandhal, "An Efficient Test Case Prioritization using Hierarchical Clustering for Enhancing Regression Testing", International Journal of Innovative Technology and Exploring Engineering, Volume-8 Issue-5, pp 914-917, 2019
24. M. Azizi and H. Do, "Graphite: A Greedy Graph-Based Technique for Regression Test Case Prioritization," *2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, Memphis, TN, 2018, pp. 245-251.
25. Samaila Musa, Abu Bakar Md Sultan, Abdul Azim Abd Ghani, and Salmi Baharom, "Empirical evaluation of test effort efficiency of software GA–based regression test case prioritization strategy", AIP Conference Proceedings, pp 1-7, 2018
26. Y. Ren, B. Yin and B. Wang, "Test Case Prioritization for GUI Regression Testing Based on Centrality Measures," *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, Tokyo, 2018, pp. 454-459
27. W. Fu, H. Yu, G. Fan, X. Ji and X. Pei, "A Regression Test Case Prioritization Algorithm Based on Program Changes and Method Invocation Relationship," *2017 24th Asia-Pacific Software Engineering Conference (APSEC)*, Nanjing, 2017, pp. 169-178.
28. Megala Tulasiraman1, Vivekanandan Kalimuthu, "Megala Tulasiraman1* Vivekanandan Kalimuthu", International Journal of Intelligent Engineering & Systems, Vol 11, Issue 1, pp 221-228, 2018
29. Mampi Kerani, Sharmila, "Novel Technique for the Test Case Prioritization in Regression Testing", International Conference on Advances in Computing and Data Sciences, pp 362-371, 201
30. Sheikh Fahad Ahmad, Deepak Kumar Singh, Preetam Suman; "Prioritization For Regression Testing Using Ant Colony Optimization Based On Test Factors", Intelligent Communication, Control And Devices, Pp 1353-1360, 2018
31. A. Morozov, K. Ding, T. Chen And K. Janschek, "Test Suite Prioritization For Efficient Regression Testing Of Model-Based Automotive Software," *2017 International Conference On Software Analysis, Testing And Evolution (Sate)*, Harbin, 2017, Pp. 20-29.
32. A. Vescan, C. Şerban, C. Chisăliţă-Cretu And L. Dioşan, "Requirement Dependencies-Based Formal Approach For Test Case Prioritization In Regression Testing," *2017 13th Ieee International Conference On Intelligent Computer Communication And Processing (Iccp)*, Cluj-Napoca, 2017, Pp. 181-188.
33. Jeongho Kim, Hohyeon Jeong, And Eunseok Lee. 2017. Failure History Data-Based Test Case Prioritization For Effective Regression Test. In*proceedings Of The Symposium On Applied Computing* (Sac '17). Acm, New York, Ny, Usa, 1409-1415

AUTHORS PROFILE

Dr. Priti Sharma MCA, Ph.D.(Computer Science) is working as an Assistant Professor in the Department of Computer Science & Applications, M.D. University, Rohtak. She has published more than 50 publications in various journals/ magzines of national and international repute. She is guiding 3 Ph.D. research scholars from various research areas. She is engaged in teaching and research from last 12 years. Her area of research includes Data mining, Big data, Software Engineering, Machine Learning.

Kavita Dahiya received the MCA and M.Tech degree from Maharshi Dayanand University, Rohtak in 2004 and 2009 respectively. She received M.phill (Computer Science) Degree in 2005 from Madurai Kamaraj University Madurai. She Worked as an Assistant Professor in Matu Ram Institute of Engineering & Management Rohtak in 2011 to 2013. She worked as a lecturer from 2009 to 2010 in Sat Priya Group of Institution Rohtak. Currently She is a Research Scholar from Maharshi Dayanand University.