ISSN 2278-3091
Volume 8, No.4, July – August 2019
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse36842019.pdf
https://doi.org/10.30534/ijatcse/2019/36842019

Comparative analysis of Test Case Prioritization Approaches in Regression Testing

Priyanka Paygude[1], Shashank Joshi[2], Debnath Bhattacharyya[3], Tai-hoon Kim[4]
[1]Bharati Vidyapeeth Deemed University College of Engineering, Pune, India, pspaygude@bvucoep.edu.in
[2]Bharati Vidyapeeth Deemed University College of Engineering, Pune, India, sdjoshi@bvucoep.edu.in
[3]Vignan's Institute of Information Technology, Visakhapatnam-530049, India, debnathb@gmail.com
[4]Beijing Jiaotong University, Beijing 100044, P. R. China, taihonn@daum.net

ABSTRACT

Testing is a huge time and cost consuming task in software development. Moving most of the testing strategies from manual to automation has significantly reduced time and efforts a lot. Still, regression testing pays a lot in time, effort and cost for testing the whole software for each change in the code. Thus, test case prioritization is highly researched and improved domain in last decade which covers and prioritizes all the test cases for early fault detection and cost saving. In this paper, 27 scholar articles are reviewed ranging from year 2000 to 2019. Objective of this paper is to review and compare top TCP approaches for their strengths and limitations. This comparative study will be beneficial for beginners and experts in the domain of TCP for further study.

Key words : Coverage based, comparative study, history based, model based, regression testing, test case prioritization

1. INTRODUCTION

Testing is one of the significant phases in SDLC model, where cost, resources and time are the key factors. Traditionally, testing is performed after the coding phase. But nowadays, as the agile became the development approach due to its benefits, testing has become a parallel activity performed along with the coding. Aim of the testing is to deliver the defect free software within estimated cost and schedule, which can be achieved by detecting defects in the early phase of software development.
Testing becomes very crucial when the software undergoes continuous changes/ up-gradations. Regression testing is the approach which confirms that changes done in the code under up-gradation has not affected the already working functionalities of SUT. Regression testing is performed by re-running all the test cases which ensures that no new defects have been introduced in the changed code. However, re-running all the test cases ("Retest All" approach) at each stage of change in the application is impossible in terms of money, time and resources. Thus, researchers have focused on this problem in last two decades and came up with many solutions to reduce the time for testing along with improvement in effective testing. Test Case Prioritization (TCP) is the widely researched area to overcome the limitations of regression testing. TCP ranks the test cases of the test suite in order to achieve one or more objectives, such as increasing the rate of fault detection, reducing time for testing, detecting bugs in early phase of development, prioritization based on user requirements etc. Average Percentage of Faults Detected (APFD) metric is used study the rate of fault detection[2]. Too much detection of faults after doing the modification in the code has direct effect of reliability of software [4].Regression testing is the part of risk management of system under maintenance which distinguishes potential faults in the system before they happen and influence software system risk, thus reliability [5]-[7]. In research work [09], author has frame a software reliability growth model by using the test execution time and code coverage information.
Despite of some literature surveys and studies of introducing and evaluating TCP approaches, there are limited comparative studies available for TCP approaches. In this article, we have reviewed the ten recent survey papers, which results that code coverage, model and history based are most commonalty used TCP approaches. Here, we are building a comparative study of coverage, model and history based TCP approaches. We have surveyed the recent contributory articles under these three approaches and performed a comparative analysis.

2. CLASSIFICATION OF TCP TECHNIQUES AND OBJECTIVES

There are three techniques under regression testing: Test Case Prioritization (TCP), Test Case Selection (TCS) and Test Suite Minimization (TSM). Test Case Selection (TCS) is like subset of Test Case Prioritization, where selected test cases from the test suite are picked for execution. TCS aims to locate the faults under the changed code and execute them to get the confidence that changed code does not have affected unchanged code. TCP re-execute all test cases where TCS execute subset of test cases. TSM goal is to trim down the number test cases from the test suite which are redundant and absolute.
TCP is better approach as compared to TCS and TSM, as it does not negotiate with the quality of testing by reducing the test suite. Every TCP technique follows one or more objectives; so technique can be single or multi objective. Figure 1 shows TCP techniques and objectives.

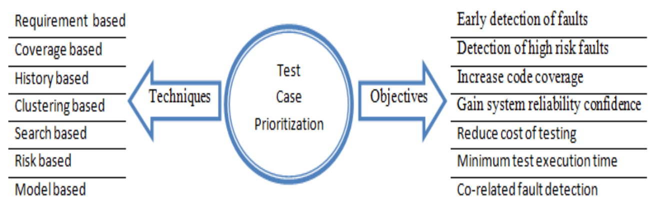

Figure 1: TCP techniques and objectives

3. RESEARCH METHODOLOGY

This section will describe the research methodology we adopted.

3.1 Research Questions

We set up four research questions for this case study. The findings of these research questions will be helpful for researchers who are beginners to conduct research in the domain of TCP approaches of regression testing.

- RQ1: What are the most inspected TCP approaches?

We know that there are many approaches of TCP studied. Our motive is to identify most researched TCP approaches using recent survey and review papers.

- RQ2: What are the strengths of top researched TCP techniques?

Our second research question is to focus on comparative study of top researched TCP techniques for their strength.

- RQ3: What are the limitations of top researched TCP techniques?

We assume that every TCP approach is best suited for some specific programming system. This research question does comparative study of top researched TCP techniques for their weaknesses.

There are many survey papers published on TCP approaches for effective and efficient regression testing, but to the best of our knowledge, there is no comparative study paper on TCP techniques. This encouraged us to prepare the comparative review article on top researched TCP techniques. This paper attempts to make the following contribution:

1. We have extracted around 16 survey papers out of which 09 have been finalized for comparative study. We restricted our study of survey for years from 2012 to 2019. From the studied survey papers on TCP techniques, it has been observed that coverage-based technique is the most researched area followed by history-based and model-based technique. This study is shown in table 1.
2. In depth study of recent published papers on coverage, history and model-based prioritization mechanism in details.
3. The main contribution of paper is the comparative study of top TCP techniques i.e. coverage, history and model-based prioritization based on factor of metric, results, dataset and performance. This comparative study will help researchers to gain the knowledge of advantage and limitations of each compared TCP approaches.

4. REVIEW OF TOP THREE RESEARCHED TCP APPROACHES

4.1 Code Coverage Based Prioritization

It is a code-based fundamental white-box testing approach that uses the structural aspects of the code such as statement, function, condition, path etc. Here, the TCs are mapped with the structural aspects of the code under testing, which then are used for ordering the TCs. The main goal of Coverage Based Prioritization is to achieve the maximum coverage by running minimum number of TCs, which ultimately enhance the rate of fault detection. There are many tools available to capture structural information of a code, which then can be used for mapping with respective TCs.

3.2 Research Contribution

Table 1: Summary of studied survey papers on TCP in regression testing

Study Reference	Year of Publication	Years Covered	Total Studies Reviewed	Top Researched TCP Technique		
Yoo and Harman [1]	2012	1977-2009	159	Code Coverage	Fault based	Model based
Singh et al. [3]	2012	1997-2011	65	Code Coverage	History based	Model based
Catal et al. [10]	2013	2001-2011	120	Code Coverage	Model based	History based
Kumar and Singh [11]	2014	NA	19	Code Coverage	Fault based	Model based
Kiran et al. [12]	2015	NA	90	Code Coverage	Requirement based	Model based
Hao Dan et al. [13]	2016	NA	NA	Code Coverage	Model based	History based
M. Khatibsyarbini et al.[14]	2017	1999-2016	69	Search algorithm	Code Coverage	History based
M. Rajendrani et al. [15]	2018	2001-2018	90	Code Coverage	Requirement based	History based
Saraswat Pavi et al. [16]	2019	2012-2019	34	Code Coverage	Model based	History based

Here, we are highlighting total 7 research publications in coverage-based TCP which have added a significant contribution in this approach. Elbaum et al. [1] proposed a version specific coverage-based TCP technique with the aim of maximum fault detection, which works on statement and function level SUT. Every test case is mapped with underline affected lines of code (known as statement coverage) and function covered (known as function coverage). Author concludes that function level coverage-based TCP needs less instrumentation for gathering TC mapping information compared to statement level coverage. This work is evaluated on space program and seven C programs at Siemens Corporate Research Lab. APFD is the metric used for experimental evaluation of work.

In the work proposed by Leon et al. [19], authors have compared the coverage-based TCP techniques which includes total - additional coverage and distribution based TCP includes clustering based coverage. Euclidian distance is calculated between clusters for similarity and dissimilarity measure. TCs which are part of same cluster are more similar than that of another cluster. Authors conducted experiment and used an APFD metric to compare the results, which depicted that distribution based TCP filtering techniques are more efficient than coverage based. Jeffrey et al. [23] used the relevance slice, where slice is the block of affected output of SUT and the underline branches executed. Higher the number of branches, higher is the priority assigned to slice. Authors evaluated the slice concept on Siemens Program and found that proposed technique performs better as compared to additional coverage-based prioritization. Statement based coverage is researched in more depth by Beena et al. [21] where a matrix is constructed that stores the Test Case Coverage (TCC) data mapped across the TCs. Each time the TC gets selected for execution, the TCC matrix gets updated. Getting maximum code coverage by executing minimum number of TCs is the goal behind this approach. Konsaard Patipat and Lachana Raming wong [20] used a condition coverage data which was given as an input to the Genetic Algorithm (GA) and Bee Colony Optimization (BCO) for TCP. Authors have evaluated technique on classic triangle identifier problem and found the result that GA works better as compared to BCO. Improving software reliability by early detection of faults using particle swarm optimization (PSO) is studied in the work [8]. The optimization by PSO can also be improved by combining the technique with clustering [9]. Ant colony optimization (ACO) is another swarm optimization technique that is considered for TCP [18]. Authors [50-24] have proposed a hybrid approach by combining two optimization algorithms: particle Swarm optimization technique and genetic algorithm. Test case execution time is used as a fitness function for evaluation of each generation. APFD metric is used to evaluate the performance of proposed technique. The research work proposes in [52-25] utilize statement coverage information for assigning priority to test cases. Here, authors propose an adaptive approach to select the TCs using APSC metric. The code coverage in combination with hamming distance technique is proposed by Hridoy et al. [22] that detects the existing faults as well as newly introduced at the earlier phase of development. This technique uses the behavior of the SUT and constructs a flow graph, used for mutant code coverage data gathering. Code coverage based TCP approach is summarize in table 2.

The figure 2 shows the general working approach of code coverage based TCP where statement coverage, branch coverage, path coverage and function coverage data works as basic block for deciding the priority of covered TCs. Each TC is aligned with no. statement, path, function and condition it covers for given SUT.

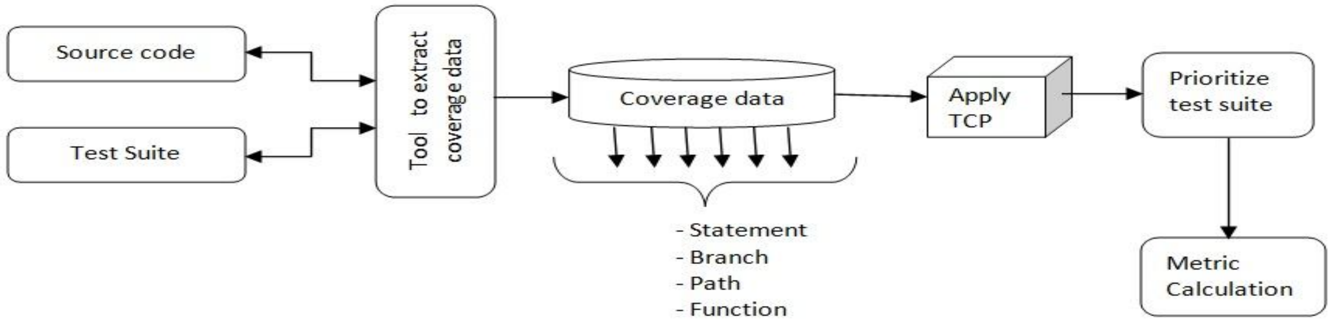

Figure 2: General working of code coverage based TCP technique

Table 2: Summary of code coverage based TCP

Authors and year	Prioritization technique	Remarks	Metric used	Subject Program
Elbaum et al. (2000) [1]	coverage at statement and function level 9 TCP techniques (total, additional etc.) are proposed	Both statement and function level coverage gives efficient result Instrumentation work for function coverage is much lesser compare to function coverage	Average Percentage of Fault Detection (APFD)	seven C programs at Siemens research lab and 1 space program
David Leon and Andy Podgurski	Proposed code coverage based TCP uses total and additional coverage data	distribution based TCP filtering techniques are more efficient than coverage based	APFD	3 compilers are used for GCC, Jikes, and

(2003) [19]	distribution based TCP uses clustering approach			javac compilers
Jeffrey Dennis, and Neelam (2006) [23]	Uses the concept of relevance slice which is mapping of branch coverage data with affected output of SUT.	perform better compare to additional code coverage	APFD	Siemens suite
Kaur et al. (2011) [26]	Branch Coverage based TCP	Better result compare to total additional statement coverage based TCP	Average percentage of condition coverage (APCC)	Siemens suite
Beena, Raman, and S. Sarala (2013) [21]	Statement coverage data and affected TC is mapped in test case coverage matrix(TCC)	Technique not evaluated	APFD	NA
Konsaard Patipat, and Lachana Ramingwong (2015) [20]	Comparative study of Genetic Algorithm (GA) and Bee Colony Optimization(BCO) using Total condition coverage as input	GA performs better compare to BCO	APCC	Evaluated on classic triangle problem
Hridoy et al. (2015) [22]	Input code coverage data in combination with hamming distance	Early detection of existing as well as new faults	APFD	NA

4.2. Model based prioritization

As the software undergoes maintenance phase, continuous changes in code indirectly increases the complexity of testing the code and thus testing efforts and cost. Maintaining the data of changed code and underlined test cases needs to be timely upgrade which is time consuming task. Model based TCP suits best in such cases and thus it becomes the one of the top researched TCP approaches due to its benefits. In SDLC, models are created earlier before coding phase; thus help to detect bugs in early phase of development. Unified Modeling Language (UML) models (state diagrams, activity diagram, class diagram etc.) are utilized to depict the behavior of system. In this section we would like to highlight the 6 research publication for their unique contribution under model based TCP.

Approach proposed by Hemmati et al. [30] has followed similarity based test case selection approach. Similarity in association with clustering for unique selection in proposed in [39]. Graph based TCP is exercised by Panigrahi et al. [31] where they construct a graph showing data control dependencies of the changed functionality and obtain the TCs covering the affected graph nodes. Priority is assigned to each node. The empirical evaluation of proposed work is performed on ATM, Library System, Elevator Controller and Vending Machine and found 30% improvement in TCP compared to random prioritization. Research on comparison of model based and code based TCP was carried out by Qin et al. [29] for objected-oriented programming which concludes that UML-based approach is best for TCP in objected-oriented programming. Rathore et al. [33] developed an automated tool which converts sequence and activity diagram into control flow tree which then used for test case generation and prioritization.

Dahiya et al. [28] considers class, sequence and activity diagrams to capture the semantic changes in code for test case selection. But this approach does not consider prioritization of test cases. The approach proposed by Zhang et al. [32] extract C&K metrics from UML class diagram and sequence diagram to estimate error probability and severity. TCs are then prioritized using calculated error probability of each affected functionality. Basa et al. [26] focused on behavior of system before and after changes in system. The changes are captured in revise sequence diagram which then are used to calculate weight of affected nodes of constructed graph. Each node is in association with number of test cases. Authors have evaluated approach on six Java programs. This approach considers only dynamic changes in system behavior, and does not identify static changes at class level, which can be drawback of approach. The work presented in [40] has implemented Markov chain testing strategy for test case selection by improving the rate of fault detection. In [41], author constructs dependency graph by using UML models to track the changes in the original and modified code for affected test case selection. This technique is proposed for D programming language. The model based TCP approaches studied are summarizing in table 3.

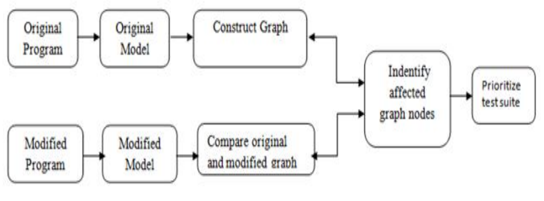

Figure 3: General working flow of Model based TCP

Figure 3 shows the broad working approach of model based TCP. Here, a graph is constructed for original and modified code. The affected nodes are identified based on changed code- model and underline TCs are prioritized.

Table 3: Summary of code coverage based

Authors and year	**Prioritization technique**	**Remarks**	**Metric used**	**Subject Program**
Hemmati et al. (2013) [30]	similarity based test case selection approach	Test case diversity increases scalability of model based testing	APFD	video conference system, safety critical system
Panigrahi and Mall (2013) [31]	Object oriented features were taken into account for constructing dependency graph showing control and data dependencies	Almost 30% improvement in bug detection	APFD	ATM, library system, elevator controller and vending machine
Qin Mengqiu, and Haini Cai. (2015) [19]	Compared to model-based and code-based and found earlier is much efficient	UML-based approach is best for RT in OOP	NA	only proposed approach
Rathore et al. (2015) [23]	Developed an automated tool which convert sequence diagram and activity diagram into control flow tree which then used for test case generation and prioritization	Compare to non prioritization approach, proposed approach shows better performance	APFD	travel reservation agency
Dahiya S., Bhatia R. K., and Rattan D. (2016) [28]	TC selection using class, sequence and activity diagrams based on before and after changes in semantic of operations	improvement in finding re-testable and reusable TCs	APFD	ATM, library system and student enrollment system
Zhang et al. (2018) [32]	Extracting C&K metrics from UML class diagram and sequence diagram to estimate error probability and severity. TCs are then prioritized using calculated error probability	Error detection rate increases effectively compare to traditional code coverage and additional coverage techniques	APFD	unmanned aerial vehicles (UAV) fight control system
Basa et al. (2018) [27]	A graph generated from sequence diagram is used for calculating weights of affected nodes	improvement in APFD metric compared to traditional TCP approach	APFD	evaluated on 6 java implemented systems

4.3 History based TCP

History based TCP is one of the widely researched field in prioritization. Under this, history of fault information such as test case raising the fault, fault severity, time stamp, number of executions etc. are stored in the memory which then used for next regression cycle. Fault matrix shown in fig. 4 is the most commonly used storage way to maintain the fault history information. Here, TC1, TC2 etc. are test cases from the test suite and F1, F2 etc. are faults found in the system. The mapping depicts faults detected by each TC from the test suite and is shown in table 4. In this subsection of paper, we would like to show up the 5 research papers which reflect history based TCP approach.

The concept of fault age was established in paper written by Kim and Porter [34]. Author calculates the fault age by counting the test run which could not detect the faults. Probability for each test case is calculated based on its position in test suite execution order and historical performance data of TC. Existing method consider all faults with same severity, which became the point of criticism for many further researchers. The extension of this work is proposed by Gupta et al. [35] which unlike Kim and Porter [34] does not calculate TC based probability; instead consider modified lines of code for probability calculation used for ordering test cases. Author found result improvement compared to Kim and Porter [33].

Gupta et al. [35] studied history based TCP approach that uses data of modified lines of code. The test cases covering modified lines of code were given first priority for execution. The work presented Wang and Zeng [36] uses the fault detection history of TCs and importance of requirements. Requirement properties are obtained in requirement phase of SDLC from the developer and customer. These properties are matched with TCs that are further considered for assigning priority to each requirement. The calculated priority against requirement is mapped with underline TC deciding the order of TCs execution in TCP. This approach is evaluated on industrial CPMISS- web application which concludes that the approach works far better compared to traditional TCP approach. An artificial immune system based, clonal selection algorithm is proposed to find an optimal test case ordering in research article by Tulasiraman et al. [37].

Table 4 :Test cases and its detected faults (History Based)

Test Cases	**F1**	**F2**	**F3**	**F4**	**F5**	**F6**	**F7**	**F8**
TC1	X			X			X	
TC2		X	X		X		X	X
.								
TCn	X		X			X		

Hasan et al. [38] have come up with new edge of TCP technique where they consider the dissimilarity between the test cases based on historical data of fault detection. Test cases belonging to different clusters of affected functionality are given high priority for execution. Authors have evaluated technique on well known dataset - Defects4j and obtained efficient result in APFD metric compared to random TCP approach. Table 5 summaries the studied scholar articles of history based TCP.

Table 5: Summary of history based TCP approach

Authors and year	**Prioritization technique**	**Remarks**	**Metric used**	**Subject Program**
Kim, J.M., Porter, A (2002) [34]	Fault age is calculated using historical data of TC execution	Do not consider fault severity factor for TCP	APFD	Web application
Gupta et al. (2015) [35]	mapping of TC against modified lines of code	Improve result compare to [31] technique	APFD	ten programs of Java, C and C++
Wang et al. (2016) [36]	TCP using fault detection history with requirement classification and importance	proposed approach is superior compared to random prioritization	APFD and Fault detection rate	CPMISS- web application
Tulasiraman Megala, and Vivekanandan Kalimuthu. (2017) [37]	cost cognizant history based prioritization of test cases using clonal selection algorithm (CC-CSA)	perform better compared to Random prioritization, total functional coverage, genetic algorithm and cost cognizant prioritization of test cases	APFDc	hospital management application
Hasanet et al. (2017) [38]	dissimilarity test case clustering using historical data analysis to detect system faults at the earliest	dissimilarity algorithm performs better than untreated, random and similarity based TCP	APFD	JodaTime, Closure, and Chart from well reputed Defects4j datasets

5. CONCLUSION OF COMPARATIVE STUDY OF TCP TECHNIQUES

We have studied and analyzed total 27 papers which include 9 survey papers of TCP approaches in regression testing and 18 research papers covering recent advancement in TCP approach. Those 18 papers contains 7 code coverage based, 6 model based and 5 history based TCP studies. The comparative analysis of these approaches is written in table 6. The main goal of most of the TCP techniques is the early detection of faults with minimum cost. Maximum code coverage, minimum execution time, severe fault detection and customer requirement priority based prioritization are also major objectives of many research TCP techniques.

The study and analysis we did, is conclude in following points:

1. Our research question 1 is about identifying mostly used TCP approach which is code coverage based approach due to its high accuracy of locating faults at the earliest. But it suffers an overhead of continuous updation of data for underline coding parameters such as lines of code, functions, path, decisions, conditions etc.

Table 6: Comparative study of Code Coverage, Model and History Based Software TCP approaches

Parameter	**Code Coverage**	**History Based**	**Model Based**
Data used	Code coverage data	History of faults and test cases	UML model, OOP models
Overhead	Continuous updation of code data such as lines of code, conditions, decisions etc.	Maintaining history of fault and test covering faults	updation of model as per new code churn
Bug detection in	testing phase	testing phase	design phase
Access to source code	required	sometimes required	not required
APFD	High	High	Moderate
Time requirement	High	Moderate	Less
Rate of detecting severe faults	Moderate	High	Moderate
Application best suited for	Small	Moderate	High

2. When testing a huge application, maintaining the code coverage data becomes a tedious task. As it needs to be updated after each single change in the code. Model based TCP approach works best in case of huge application. As the changes in the model directly highlights the underline test case data.

3. Code coverage based TCP technique dominantly gives better APFD results compared to model and history based TCP approaches.

4. Early detection of severe faults can easily be achieved in history based TCP approach. As the history of fault is detected, its severity- priority and other details related to

faults are maintained over the evaluation of testing process. This approach also helps to predict the probable faults and helps to take the corrective measures.

5. When time and cost are the main concerns of testing, model based TCP in combination with history of fault data will result better compared to code coverage TCP approach which needs much instrumentation for maintaining code updates. We hope this article will be helpful for researchers who have interest in TCP approaches in regression testing.

REFERENCES

[1] Yoo, Shin, and Mark Harman. **Regression testing minimization, selection and prioritization: a survey**. Software Testing, Verification and Reliability vol. 22, no. 2, pp. 67-120, 2012. https://doi.org/10.1002/stv.430

[2] Han Moi Sim, D. Sai Teja Reddy, **Survey on Test Case Prioritization and Measuring Test Cases Using APFD**, Asia-pacific Journal of Convergent Research Interchange, SoCoRI, Vol.1, No.2, pp. 11-17, June 2015. https://doi.org/10.21742/apjcri.2015.06.02

[3] Singh, Yogesh, et al. **Systematic literature review on regression test prioritization techniques** *Informatica* Vol.36, no. 4, 2012.

[4] G. Chandrika, **Study on Software Reliability and Reliability Testing**, Asia-pacific Journal of Convergent Research Interchange, SoCoRI, Vol.1, No.1, pp. 7-20, March 2015 https://doi.org/10.21742/apjcri.2015.03.02

[5] Seong Ho Sung, Pattan Zinna Khan, **Quantitative and Qualitative Approach for IT Risk Assessment**, Asia-pacific Journal of Convergent Research Interchange, SoCoRI, Vol.1, No.1, pp. 29-35, March 2015 https://doi.org/10.21742/apjcri.2015.03.04

[6] Gowtham Kumar Pullagujju, **Risk Utilization in Quantitative Approach**, Asia-pacific Journal of Convergent Research Interchange, SoCoRI, Vol.1, No.1, pp. 21-27, March 2015 https://doi.org/10.21742/apjcri.2015.03.03

[7] Dong Ju Kim, P. Lakshmi Manjusha, **Assessment of Risks in Management Factors**, Asia-pacific Journal of Convergent Research Interchange, SoCoRI, Vol.1, no.2, , pp. 1-10, June 2015 https://doi.org/10.21742/apjcri.2015.06.01

[8] Su Min Shin, Sk. Uroosa, **Predicting Software Reliability Using Particle SWARM Optimization Technique**, Asia-pacific Journal of Convergent Research Interchange, SoCoRI, Vol.1, No.3, pp. 17-30, September 2015 https://doi.org/10.21742/apjcri.2015.09.02

[9] Amol K. Kadam, S.D. Joshi, Debnath Bhattacharyya and Hye-Jin Kim.**Diagnosis of Software using Testing Time and Testing Coverage.** International Journal of Hybrid Information Technology. Vol. 9. No. 9. Sep. 2016 https://doi.org/10.14257/ijhit.2016.9.9.08

[10] Catal, Cagatay, and Deepti Mishra. **Test case prioritization: a systematic mapping study.** Software Quality Journal, Vol. 21, no.3, pp: 445-478, 2013 https://doi.org/10.1007/s11219-012-9181-z

[11] Kumar, Amit, and Karambir Singh. **A Literature Survey on test case prioritization**. Compusoft, Vol. 3, no. 5, 2014.

[12] Kiran, R. Surya. **A literature survey on TCP-test case prioritization using the RT-regression techniques.** Global Journal of Research In Engineering, 2015.

[13] Hao, Dan, Lu Zhang, and Hong Mei. **Test-case prioritization: achievements and challenges.** Frontiers of Computer Science, Vol. 10, no. 5 pp. 769-777, 2016. https://doi.org/10.1007/s11704-016-6112-3

[14] Khatibsyarbini, M., Isa, M. A., Jawawi, D. N., & Tumeng, R. **Test case prioritization approaches in regression testing: A systematic literature review.** Information and Software Technology, Vol. 93, pp. 74-93, 2018. https://doi.org/10.1016/j.infsof.2017.08.014

[15] Mukherjee, Rajendrani, and K. Sridhar Patnaik. **A survey on different approaches for software test case prioritization.** Journal of King Saud University-Computer and Information Sciences, 2018. https://doi.org/10.1016/j.jksuci.2018.09.005

[16] Saraswat, Pavi, Abhishek Singhal, and Abhay Bansal. **A Review of Test Case Prioritization and Optimization Techniques**. Software Engineering. Springer, Singapore, pp. 507-516, 2019. https://doi.org/10.1007/978-981-10-8848-3_48

[17] Jin Wang, Yu Gao, Wei Liu, Arun Kumar Sangaiah, Hye-Jin Kim, **An Improved Routing Schema with Special Clustering using PSO Algorithm for Heterogeneous Wireless Sensor Network**, Sensors, vol.19, no.3, Feb. 2019 https://doi.org/10.3390/s19030671

[18] Jin Wang, Jiayi Cao, R. Simon Sherratt, Jong Hyuk Park, **An improved ant colony optimization-based approach with mobile sink for wireless sensor networks**, Journal of Supercomputing,Vol. 74, No.12, pp.6633-6645, Dec. 2018. https://doi.org/10.1007/s11227-017-2115-6

[19] Leon, David, and Andy Podgurski. **A comparison of coverage-based and distribution-based techniques for filtering and prioritizing test cases**. 14th International Symposium on Software Reliability Engineering, 2003. ISSRE 2003,. IEEE, 2003.

[20] Konsaard, Patipat, and Lachana Ramingwong. **Total coverage based regression test case prioritization using genetic algorithm**. In 2015 12th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), pp. 1-6. IEEE, 2015 https://doi.org/10.1109/ECTICon.2015.7207103

[21] Beena, Raman, and Subramani Sarala. **Code coverage based test case selection and prioritization**. arXiv preprint arXiv:1312.2083, 2013.

[22] Hridoy, Syed Akib Anwar, Faysal Ahmed, and Md Shazzad Hosain. "**Regression Testing based on Hamming Distance and Code Coverage"**. International Journal of Computer Applications, Vol. 975, 2015

[23] Jeffrey, Dennis, and Neelam Gupta. **Test case prioritization using relevant slices**. In 30th Annual International Computer Software and Applications Conference (COMPSAC'06), vol. 1, pp. 411-420. IEEE, 2006.
https://doi.org/10.1109/COMPSAC.2006.80

[24] Deepti Arora and Anurag Singh Baghel.**A Hybrid Meta-heuristics Technique for Finding Optimal Path by Software Test Case Reduction**.International Journal of Hybrid Information Technology. Vol. 8. No. 4. Apr. 2015
https://doi.org/10.14257/ijhit.2015.8.4.05

[25] Sarabjit Kaur and Bhanu Priya.**Test-Case Prioritization Using Adaptive Approach.** International Journal of Hybrid Information Technology. Vol. 9. No. 5. May. 2016.
https://doi.org/10.14257/ijhit.2016.9.5.05

[26] Kaur, Arvinder, and Shubhra Goyal. **A genetic algorithm for regression test case prioritization using code coverage**. International journal on computer science and engineering 3, no. 5, pp: 1839-1847, 2011.

[27] Basa, S. S., S. K. Swain, and D. P. Mohapatra. **Model-based Test Case Prioritization.** pp. 373-381, 2011.
https://doi.org/10.26438/ijcse/v6i10.373381

[28] Dahiya, S., Bhatia, R. K., & Rattan, D. **Regression test selection using class, sequence and activity diagrams**. IET Software, Vol. 10, no. 3, pp:72–80, 2014 .
https://doi.org/10.1049/iet-sen.2014.0241

[29] Qin, Mengqiu, and Haini Cai. **Comparison and Analysis of Three Regression Testing Methods**. In 2015 2nd International Conference on Electrical, Computer Engineering and Electronics. Atlantis Press, 2015.
https://doi.org/10.2991/icecee-15.2015.265

[30] Hemmati, H., Arcuri, A., Briand, L., **Achieving scalable model-based testing through test case diversity**. ACM Trans. Software Eng. Methodol. 22, pp. 1–42, 2016
https://doi.org/10.1145/2430536.2430540

[31] Panigrahi, C.R., Mall, R. **A heuristic-based regression test case prioritization approach for object-oriented programs**. Innovations Syst. Softw. Eng. Vol. 10, pp. 155– 163, 2014.
https://doi.org/10.1007/s11334-013-0221-z

[32] Zhang, Tianning, Xingqi Wang, Dan Wei, and Jinglong Fang. **Test Case Prioritization Technique Based on Error Probability and Severity of UML Models**. International Journal of Software Engineering and Knowledge Engineering, Vol. 28, no.06, pp. 831-844, 2018.
https://doi.org/10.1142/S0218194018500249

[33] Rathore, Lokesh Kumar, and Neelabh Sao. **An integrated model based test case prioritization using uml sequence and activity diagram,** 2015.

[34] Kim, J.-M., Porter, A. **A history-based test prioritization technique for regression testing in resource constrained environments**. In: Proceedings of the 24th IEEE International Conference on Software Engineering. ICSE, pp. 119–129, 2002.
https://doi.org/10.1145/581339.581357

[35] Gupta, Avinash, Nayneesh Mishra, Aprna Tripathi, Manu Vardhan, and Dharmender Singh Kushwaha. **An improved history-based test prioritization technique technique using code coverage.** In Advanced Computer and Communication Engineering Technology, pp. 437-448. Springer, Cham, 2015.
https://doi.org/10.1007/978-3-319-07674-4_43

[36] Wang, Xiaolin, and Hongwei Zeng. **History-based dynamic test case prioritization for requirement properties in regression testing**. In Proceedings of the International Workshop on Continuous Software Evolution and Delivery, pp. 41-47. ACM, 2016.
https://doi.org/10.1145/2896941.2896949

[37] Tulasiraman, Megala, and Vivekanandan Kalimuthu. **Cost Cognizant history based prioritization of test case for regression testing using immune algorithm.** Journal of Intelligent Engineering Systems Vol. 11, no.1. pp.221-228, 2018.
https://doi.org/10.22266/ijies2018.0228.23

[38] Hasan, Md Abu, Md Abdur Rahman, and Md Saeed Siddik. "**Test Case Prioritization Based on Dissimilarity Clustering Using Historical Data Analysis**. International Conference on Information, Communication and Computing Technology. Springer, Singapore, 2017

[39] Yu, Jun; Tao, Dapeng; Li, Jonathan; Cheng, Jun.: **Semantic preserving distance metric learning and applications.** INFORMATION SCIENCES, pp. 281: 674-686, 2014.
https://doi.org/10.1016/j.ins.2014.01.025

[40] Jin Jiangang, Sun Shibao and Bao Xiaoan.**Research of Software Testing Model based on Correlation Defect.** International Journal of Hybrid Information Technology. Vol. 9. No. 2. Feb. 2016.
https://doi.org/10.14257/ijhit.2016.9.2.25

[41] Nitesh Chouhan, Maitreyee Dutta and Mayank Singh. **A Program Model based Regression Test Selection Technique for D Programming Language**. International Journal of Hybrid Information Technology. Vol. 8. No. 6. Jun. 2015.
https://doi.org/10.14257/ijhit.2015.8.6.33