A systematic literature study of regression test case prioritization approaches

Omdev Dahiya[1] *, Kamna Solanki [2]

[1] *Research Scholar, UIET MD University Rohtak, India*
[2] *Assistant professor, UIET MD University Rohtak, India*
*Corresponding author E-mail: Omdahiya21792@gmail.com

Abstract

Regression testing is about running the entire test ensemble again to ensure that amendments do not negatively affect the system. A popular approach in regression testing is test case prioritization which reorders test cases in such a way that those with higher priorities are run earlier than those with lower priorities based on some criterion. Numerous researchers have worked on different aspects of prioritization approaches. This paper presents the result of the study conducted on different prioritization approaches showing mostly stressed areas by researchers and the areas where there is a future scope. For it, studies related to test case prioritization in regression testing from the year 2004 to 2018 are analyzed by dividing this time period into three slots of five years each. 36 studies were selected from 948 studies to answer the research questions framed for this study. The trends followed in TCP along with the approaches evolved are thus documented to find the current trends and future scope for the researchers to work upon.

Keywords: *Regression Testing; Software Development; Software Testing; Systematic Literature Study; Test Case Prioritization*.

1. Introduction

"Regression testing is selective retesting of a system or component to verify that modifications have not caused undesirable effects and that the system or component still complies with its specified requirements" [1]. In this process, the retesting of a system is performed in which both existing and newly added test cases were executed [2]. The experiment conducted by an industry collaborator showed that more than 50 days are required to test 20,000 lines of code [3]. Hence researchers proposed diverse techniques to improve regression testing performance so as to make it a cost-effective process. In the study conducted by Yoo and Harman, these techniques are categorized as test suite minimization, test case selection, test case prioritization [4].

1.1. Test case minimization

This approach works towards eliminating the test cases, which becomes redundant with regard to coverage of some set of program requirement thereby decreasing the number of test cases in a regression test suite [5]. It states that only a subset of the test suite is economical to use. Minimization is sometimes called "test suite reduction" signifying that the deletion of test cases is everlasting [5].

1.2. Test case selection

These techniques select some of the test cases which focus on testing the changed part of the software. It does not remove test cases; rather it filters the ones which are related to the modified portion of the source code [6].

1.3. Test case prioritization

Prioritization approach states that as the execution of all the test cases is not practical due to resource constraints, therefore the test cases are ordered in a way such that those with higher priorities are run prior to those with lower priorities based on some criterion [7]. These techniques are usually preferred because it deals with the original test suite and no test cases are eliminated from the initial test suite [8].

2. Literature study

Numerous studies were performed by researchers and several prioritization methods were introduced, with an aim to enhance fault detection rate. These methods were categorized into several domains according to the nature of the study conducted by the researchers.

Yoo and Harman conducted a survey in which various regression testing approaches were examined to enhance the significance of the assembled test suite in regression testing. The study performed by them is elaborated correspondingly into three spheres, that is - minimization, selection and prioritization format. TCP approaches were discussed in detail along with their evaluation metrics. The most popular approaches as defined by them are Coverage based, History based, Cost-based, model-based and requirement based and other approaches [4]. Kumar and Singh performed a detailed overview of coverage and cost-based approach. They have elaborated that software development cost will eventually be reduced if the new techniques are implemented in the initial phases of the software development life cycle [9]. Kiran and Chandra Prakash have performed an overall review of regression testing for test case prioritization. They have also discussed several TCP techniques along with

their strategies and algorithms developed so far. They have concluded that survey analysis eases the way of performing TCP. But, the benefits of the various parameters as well as cost factors should be taken care of. The software testing sums to nearly 50% of the whole cost of its development. If the path testing is carried out, then almost 65% of the errors in the software will be detected by itself [10]. Catal and Mishra, have focused on documenting the current trends of test case prioritization in their study and the trend of publication are shown with the types of dataset and metrics used [11]. Singh *et al.* carried out a systematic literature study for regression test case prioritization techniques for the time period of 1997 to 2011. The authors have discussed that in spite of various approaches followed by different techniques, the main aim of prioritizing test cases remains the same i.e. to increase fault detection rate. It is concluded that; as no broad techniques exist, there is a necessity to carry out experimental comparisons amongst the existing techniques and to derive out new and effective techniques [12].

Khatibsyarbini *et al.* performed a systematic literature review on TCP approaches in regression testing for the time period of 1999 to 2016. The authors have concluded that despite the fact the TCP approaches are considerably worked upon by numerous researchers in software testing industry, there are a number of prioritization techniques which have a scope for further improvement in the type of data used and thus the execution process followed by each approach [13]. Omdev *et al.* have performed a literature study of TCP approaches in which they concluded that various approaches can be combined or used as a hybrid so as to gain the advantage of each other in achieving better and remarkable results [14]. Solanki *et al.* have performed a comparative performance evaluation of modified ant colony optimization (m-aco) technique with other existing nature-inspired techniques. The results deduced by them have shown the superiority of m-aco in reaching optimal or near-optimal solutions [15]. Solanki *et al.* have performed an experimental analysis of m-aco technique for TCP with popular software testing problems and have shown the effectiveness of their approach using APFD and PTR Metrics [16].

This study was conducted with an aim to contribute towards the software testing domain by showing the trend which is going on in regression test case prioritization. For it, certain research questions are devised which will help us to focus on research papers which will show us the prevailing trend of publication of research papers in the regression test case prioritization domain.

3. Research method

This study aims to analyze the research work of numerous researchers in developing various test case prioritization approaches used in regression testing field. The guidelines of Kitchenham [17] were followed in presenting this study in a systematic and well-structured format. The given Fig. 1 shows the sequence of steps which were followed.
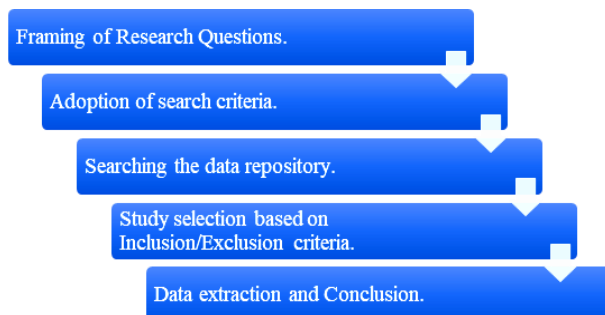

Fig. 1: Steps Taken for Systematic Literature Review.

Figure 1 depicts the steps followed in documenting this study. In the first phase, research questions are formulated which answers the importance and need for this paper review. In the next phase, search criteria are defined so as to find the most relevant papers. After this phase, the most crucial phase is searching the data repository; this study selected Google Scholar for searching the papers as it comprises papers from all data repositories whether it may be IEEE Xplore, Springer, ACM digital library, and Elsevier. Next phase deals with selecting the most relevant papers after following the exclusion and inclusion criteria. After this step 36 studies were finalized which answers the research questions. Subsequently, data extraction is performed to draw some fruitful conclusion regarding current trends and future scope.

3.1. Research questions

The aim of deciding the research questions is to screen the papers on the basis of these questions.
RQ 1 What percentage of TCP techniques are focused/oriented towards capturing user requirements/faults/Cost?
RQ 2 What percentage of test case prioritization techniques developed so far are based on nature-inspired algorithms?
RQ 3 Do some regression testing techniques available that support hybrid of Selection/Minimization and Prioritization?

3.2. Search process

3.2.1. Source of information

As per the guidelines proposed by Kitchenham in [17], for conducting systematic literature study, selection of most appropriate database for paper selection is very important. So, Google Scholar [18] was searched instead of the limited number of journals and conference proceedings in order to cover the most remarkable information, as Google scholar offers a platform where all the others databases are indexed such as

- ACM Digital library (https://dl.acm.org) [19].
- IEEE Explore (https://ieeexplore.ieee.org) [20].
- Elsevier (https://www.elsevier.com) [21].
- Springer (https://www.springer.com/in) [22].

3.2.2. Search criteria

Initially, the phrase Test case prioritization was used as a search query in Google Scholar. The starting year was set to 2004, till 2018. It returned a huge amount of literature with number mounting to 71,600. It was practically not feasible to go through so many papers. So, the search string is modified as "Test case prioritization" with same starting year to till date (April 2018). 3435 number of papers were retrieved, which is still not feasible. So, the search results were narrowed by modifying the search string to "Test case prioritization" <and> "Regression testing" <and> "Software testing". The results obtained were 2132 in number. The search results were further narrowed and finally a search string is adopted, which is, "Test case prioritization" <and> "Regression testing" <and> "Software testing" <and> "Selection" <and> "Minimization". The results obtained were 948 in number.
Then the time period is divided into 3 parts i.e. one is from 2004-2008, second is from 2009-2013 and the third is from 2014-2018, three parts of 5 years each. Then papers were searched following the search string adopted and 71 papers for 2004-2008, 311 papers for 2009-2013 and 566 papers for 2014-2018 were fetched. With this, the total goes to 948 in number.

3.3.3. Study selection

Of the results obtained by using the search string, selection of the relevant papers was performed based on the research questions and relevance to the topic i.e. which fulfills the validity of this systematic study. For the time span of 2004-2008, out of 71 papers, 10 papers were selected, for 2009-2013, out of 311 papers, 13 papers were selected and for 2014-2018, out of 566 papers, 13 papers were selected which are relevant to this study.

4. Selected studies

4.1. Research papers selected for the time span 2004-2008

Qu *et al.* have conducted a study on combinatorial interaction testing for regression testing. The authors have performed CIT on evolving programs with multiple versions [23]. Yoo and Harman have introduced a concept of Pareto efficiency in test case selection. The approach discussed by them takes multiple objectives collectively and then they describe the supremacy of their approach in test case selection [24]. Elbaum *et al.* have performed a study to select a cost-effective technique for test case prioritization. The authors have worked on analyzing the fault detection rate while applying different prioritization technique to various programs and their modified versions [25]. Malishevsky *et al.* proposed a new technique APFDc in their work, which is a cost-based metric for measuring the rate of fault detection of prioritized test cases. They have shown how test cases and fault costs are varying in nature, thus proving the effectiveness of the new metric [26]. Ramasamy *et al.* have worked on incorporating varying requirement priorities and costs in test case prioritization for regression testing. The authors have discussed that through code coverage-based techniques are stressed more till now leaving the scope for requirement-based techniques to enhance user satisfaction, cost, quality as well as the rate of fault detection and have performed their experiment to prove it [27].

Srikant *et al.* have proposed an approach, PORT which is prioritization of requirements for the test. In this system test cases are subdivided into different factors. The study was conducted with an aim to improve the rate of fault detection by incorporating customer priorities as a vital factor [28]. Jeffrey and Gupta have worked to propose a new approach for test suite reduction. They have said that as test suite minimization reduces fault detection effectiveness, so their approach will use additional coverage information regarding test cases and as a consequence of which some more test cases are retained in the reduced test suite. The results show that the fault detection is improved for reduced suites without straining the extent of test suite size reduction [29]. Baudry *et al.* have proposed an automotive test case generation and optimization approach. The authors have followed the adaptive behaviour of bacteria in a large and diverse environment so as to generate and find a global set of test cases which have potential to find faults/ kill mutants [30]. Walcott *et al.* introduced a time-aware prioritization technique that uses the genetic algorithm for ordering test cases. the results obtained by them have higher APFD values which show the superiority of their technique [31]. Hla *et al.* proposed a particle swarm optimization algorithm for automatically prioritizing the test cases according to modified software modules. The results deduced by them shows the effectiveness of their approach [32].

4.2. Research papers selected for the time span 2009-2013

Yoo and Harman conducted a survey in which various regression testing approaches were examined for the significance of the assembled test suite in regression testing. The study performed by them is elaborated correspondingly into three spheres, that is - minimization, selection and prioritization format. TCP approaches were discussed in detail along with their evaluation metrics [4]. Arafeen and Do have performed their work towards showing that, the effectiveness of TCP techniques can be enhanced by grouping test cases associated with the related set of requirements and the regression testing process is effectively managed [33]. Suri and Singhal have worked towards implementing ant colony optimization (ACO) for test case selection and prioritization. The results deduced by them shows that ACO leads them towards obtaining solutions which are in close proximity to optimal solutions [34].

Krishnamoorthi and Mary have proposed a new requirement-based prioritization technique so as to develop good quality, cost-effective software with an improved rate of fault detection for the new test as

well as for regression test [35]. Raju and Uma have proposed a requirement-based prioritization scheme and have shown its effectiveness using APFD and PTR metrics [36]. Kavitha *et al.* have proposed an algorithm for system level test case prioritization based on software requirement specification with an aim to increase fault detection rate and enhanced customer satisfaction and their results prove it [37].

Suri *et al.* have worked in developing a hybrid technique for test case selection based on BCO and GA for regression test suite reduction. They were able to obtain optimum results in minimum time [38]. Suri and Singhal have analyzed test case selection and prioritization techniques using ant colony optimization (ACO). They have concluded that the approach followed by them is able to yield optimum results [39]. Suri and Singhal have performed a survey of ant colony optimization used in the field of software testing arena. They were able to identify 21 such studies and have analyzed them in order to present a useful insight into the field [40]. Salehie *et al.* have worked on prioritizing requirement-based regression test cases and have used the practical issues encountered in the software development to generalize their findings [41]. Catal. C. has conducted a systematic literature review on the use of genetic algorithms in test case prioritization. They studied 120 papers initially and were able to find only six studies which have applicability of genetic algorithms in TCP and concluded that this field is still open for research [42]. Gupta and Chauhan have worked towards developing a hybrid approach to prioritize test cases in regression testing. They have made clusters of test cases which are then prioritized on the basis of requirement priorities thereby ensuring the reduction in the number of test cases to be executed [43]. Kaur and Goyal have proposed a genetic algorithm to prioritize the regression test suite for total fault coverage in a time constraint environment. The effectiveness of their algorithm was shown using the APFD metric [44].

4.3. Research papers selected for the time span 2014-2018

Nardo *et al.* have conducted a case study on industry system of real regression faults using coverage-based regression testing techniques. The authors examined selection, minimization, prioritization and the combination of selection and minimization approaches and deduced their results [45]. Mayan and Ravi have proposed a hybrid algorithm for test sequence selection and test case generation for regression testing. The results thus obtained shows their approach performs better as compared to other approaches [46]. Dobuneh *et al.* have used hybrid criteria for test case prioritization to be used for testing web applications. Their approach yields higher APFD values and its performance in time constrained environment is also remarkable [47]. Wang and Zeng have proposed a history-based test case prioritization based on requirement properties. The results deduced by them shows the efficiency of their approach in terms of better fault detection [48].

Hettiarachchi *et al.* have proposed a requirement risk-based prioritization technique. In their results, they show that the approach proposed by them detects fault early in the components of the system as compared to other existing techniques [49]. Panichella *et al.* have introduced a new diversity based genetic algorithm. In the results of the study conducted by the authors, their algorithm has provided superior results as compared to greedy algorithms and multi-objective genetic algorithms by detecting more faults with the same execution cost [50]. Ma *et al.* have proposed a test case prioritization method based on requirement correlations. In their method users and developers performs prioritization of requirements. In it, the order of test cases is optimized so as to lead high testing efficiency [51]. Srikanth *et al.* have continued their prior work and have done an industrial study by introducing requirement-based TCP approach which uses risk information of the system. In the results deduced by the authors, they have shown the effectiveness of their approach [52]. Silva *et al.* have proposed a hybrid approach for regression test case selection and prioritization. To select the test cases ACO is used and then test cases are prioritized on the basis of their criticality value. They have introduced a method to find which parts of the system are mostly focused by the potential users [53].

Achimugu *et al.* have conducted a systematic literature review of requirement prioritization techniques. They have designed research questions to screen in papers relevant to the decided area. The aim of their work is to study existing techniques along with their limitations so that possible improvements can be suggested in requirement prioritization techniques by means of systematic evaluation of the relevant studies [54]. Gao *et al.* have proposed an algorithm to prioritize the test cases in regression testing using ACO with an aim to find faults in the early stages of the testing process. The authors have shown the effectiveness of their approach using APFD metric [55]. Solanki *et al.* have proposed a modified ant colony optimization approach for test case prioritization. Their algorithm works by

changing the criteria followed by natural ants for selecting their food source. When the algorithm was tested on various problems, it has shown remarkable results [56]. Vescan *et al.* have proposed an approach for test case prioritization in regression testing based on requirements dependencies. The authors have combined both test case selection and prioritization approaches to conquer the drawbacks of both and have shown their findings [57].

The following table presents the summary of selected studies along with the techniques proposed by the researchers involved.

Table 1: Summary of Selected Studies Along with the Proposed Technique

Sr. No.	Authors	Year	Proposed Technique
1.	Qu *et al.* [23]	2007	Combinatorial interaction regression testing.
2.	Yoo and Harman [24]	2007	Multi-objective test case selection.
3.	Elbaum *et al.* [25]	2004	Cost-Based TCP Technique.
4.	Malishevsky *et al.* [26]	2006	Cost-cognizant TCP Technique.
5.	Ramasamy and Mary [27]	2008	Requirement-based TCP.
6.	Srikanth *et al.* [28]	2005	Requirement-based TCP.
7.	Jeffrey and Gupta [29]	2007	Fault-Based Technique
8.	Baudry *et al.* [30]	2005	Nature-inspired technique for TCP (bacteriologic algorithm).
9.	Walcott *et al.* [31]	2006	Time-aware test suite prioritization (Using Genetic algorithm).
10.	Hla *et al.* [32]	2005	Particle swarm optimization for TCP.
11.	Yoo and Harman [4]	2012	Literature survey of Regression test case selection, minimization, and prioritization.
12.	Arafeen and Do [33]	2013	Requirement-based TCP.
13.	Suri and Singhal [34]	2011	Nature-inspired technique for Requirement-Based TCP (Using Ant colony optimization).
14.	Krishnamoorthi and Mary [35]	2009	Requirements coverage-based TCP.
15.	Raju and Uma [36]	2012	Nature-inspired technique for TCP (Genetic algorithm).
16.	Kavitha *et al.* [37]	2010	Requirement-based TCP.
17.	Suri *et al.* [38]	2011	Nature inspired technique for test case reduction (A Hybrid technique using Bee colony optimization and genetic algorithm).
18.	Suri and Singhal [39]	2011	Nature-inspired technique (Using Ant colony optimization for Selection and Prioritization).
19.	Suri and Singhal [40]	2012	Nature-inspired technique (Usage of ACO in testing).
20.	Salehie *et al.* [41]	2011	Requirement-based TCP.
21.	Catal. C [42]	2012	Nature-inspired technique for TCP (Genetic algorithm).
22.	Gupta and Chauhan [43]	2011	Requirement-based TCP.
23.	Kaur and Goyal [44]	2011	Nature-inspired technique (Fault-Based TCP using genetic algorithm).
24.	Nardo *et al.* [45]	2015	Coverage-Based Regression testing approaches.
25.	Mayan and Ravi [46]	2015	A Hybrid algorithm for test sequence selection.
26.	Dobuneh *et al.* [47]	2014	Hybrid criteria for TCP.
27.	Wang and Zeng [48]	2016	History-Based TCP.
28.	Hettiarachchi *et al.* [49]	2014	Requirements-risk based TCP.
29.	Panichella *et al.* [50]	2015	Nature-inspired technique (Multi-objective test case selection using genetic algorithm)
30.	Ma *et al.* [51]	2016	Requirement-based TCP.
31.	Srikanth *et al.* [52]	2016	Requirement-based TCP.
32.	Silva *et al.* [53]	2016	Hybrid method for test case selection and prioritization.
33.	Achimugu *et al.* [54]	2014	A Systematic literature review of Requirement-based prioritization.
34.	Gao *et al.* [55]	2015	Nature-inspired technique for TCP (Using Ant colony optimization).
35.	Solanki *et al.* [56]	2016	Nature-inspired technique for TCP (Modified Ant colony optimization).
36.	Vescan *et al.* [57]	2017	Requirement-based TCP.

5. Results and analysis

In this study, the existing literature from 2004-2018 was studied. The results will now be generalised from the selected papers by subdividing them according to their potential of answering the research questions.

5.1. What percentage of TCP techniques are focused/oriented towards capturing user requirements/faults/cost? (Research question 1)

Out of the selected studies, for the time slot of 2004-2008, 5 such papers were identified, [23] [25 - 28], which have fulfilled the defined criteria of the research question and therefore included in the study. For the time slot of 2009-2013, 6 such papers were identified, [4] [33] [35] [37] [41] [44], and out of these [44] was able to fulfill criteria of RQ 2 as well, therefore it was included for it also. For the

time slot of 2014-2018, 8 such papers were identified, [47 - 53] [55]. The analysis of these papers shows that the main criterion for the development of diverse techniques is to enhance the rate of fault detection. The main difference between them is the execution sequence followed and the metrics used. In the selected studies requirement-based approach is favored by most of the researchers followed by fault-based and then cost-effectiveness based. From this, we can conclude that exploration of this approach has started and therefore it was chosen by researchers to work on it, as seen by the choice of researchers in selecting this approach to work towards developing new and effective TCP approach.
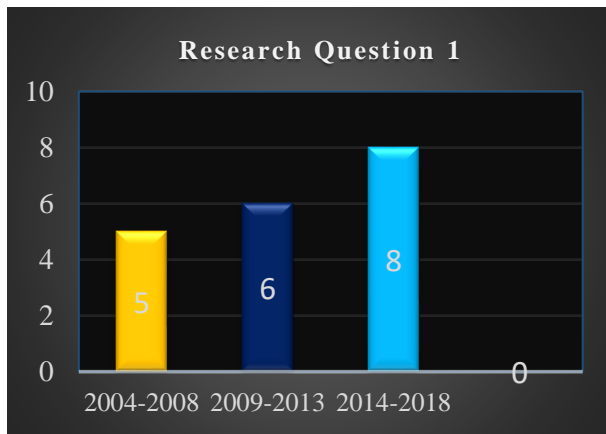
Fig. 2: Graph Depicting Results for Research Question 1.

5.2. What percentage of test case prioritization techniques developed so far are based on nature-inspired algorithms? (Research question 2)

Out of the selected studies, for the time slot of 2004-2008, 3 such papers were identified, [30 - 32], which have worked on test case prioritization using the nature-inspired algorithm. For the time slot of 2009-2013, 7 such papers were identified, [34] [36] [38] [39] [40] [42] [44] and out of these [34] [38] [39] were able to fulfil criteria of RQ 3 as well, therefore it was included for it also and [44] also fulfils criteria for RQ 1, therefore it was included there as well. For the time slot of 2014-2018, 2 such papers were identified, [56] [57] which have fulfilled the defined criteria of the research question. The reason stated by them for using nature-inspired algorithms is about their high efficiency in reaching the optimal solution to real-world problems by using successful characteristics of the biological system. From this, one can conclude that the usage of nature-inspired algorithms in developing TCP techniques has gained popularity in recent years. Earlier, it was an unexplored area, but now it was favoured by the researchers seeing the potential and effectiveness of these algorithms in solving problems in computer science.
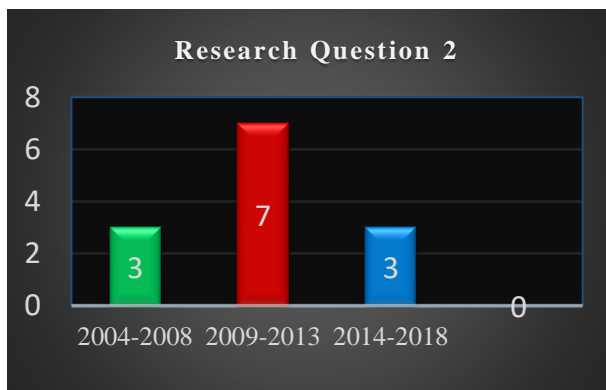

Fig. 3: Graph Depicting Results for Research Question 2.

5.3. Do some regression testing techniques available that supports hybrid of Selection/Minimization and Prioritization? (Research question 3)

Out of the selected studies, for the time slot of 2004-2008, 2 such papers were identified, [24] [29] which fulfills the defined criteria of the research question and therefore included in the study. For the time slot of 2009-2013, 4 such papers were identified, [34] [38] [39] [43] and out of these [34] [38] [39] were able to answer RQ 2 as well, therefore, it was also included there. For the time slot of 2014-2018, 3 such studies were identified, [45] [46] [54] which have answered the research question. From this, it can be deduced that every approach has its own advantages and limitations, if one is used in the combination or hybrid with the other, the researchers will be able to exploit the strengths of combined approaches as well

as in overcoming their shortcomings so that effectiveness and potential of finding faults within limited time and cost can be achieved, subsequently benefitting the entire software testing community.
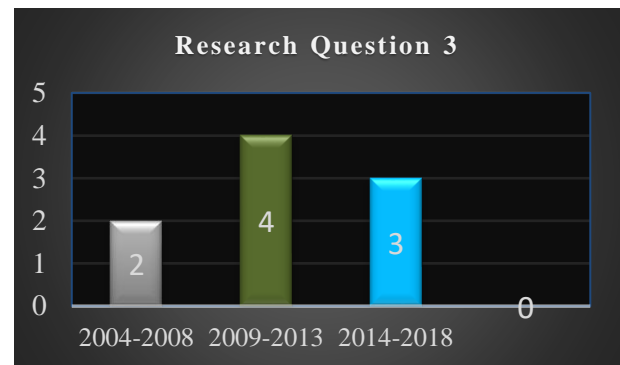

Fig. 4: Graph Depicting Results for Research Question 3.

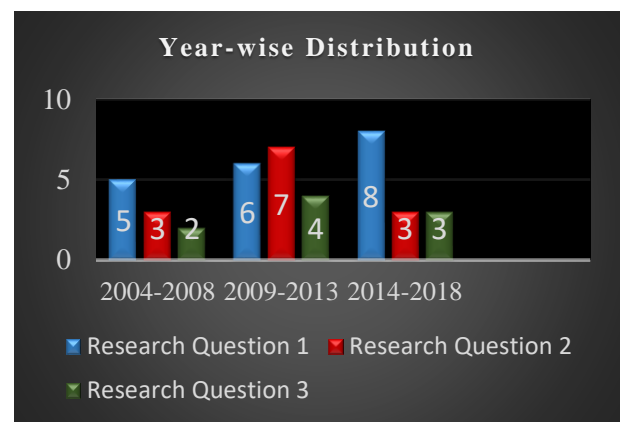Following is the year wise distribution of the selected studies for the respective years.

Fig. 5: Graph Depicting Year Wise Distribution of Studies Fulfilling Criteria of Research Questions.

6. Research findings

During the testing phase, the prioritization of test cases is an essential activity [58]. A proper combination of test plan and test cases are necessary for the development of error-free quality software within planned budget and time. At a time when client hopes are high, effective software testing will ensure delivery of reliable and quality-oriented software product, more accurate results, lower maintenance costs and more satisfied users. However, ineffective testing will lead to the opposite results; low-quality products, inadequate results, unreliable product, higher maintenance costs, unhappy users. For this implementation of TCP technique is very beneficial so that faults may be detected at the earliest and changes can be made accordingly. Therefore, the role of test case prioritization in regression testing is much more emphasized.

While performing the analysis of the studies included for answering the Research question 1, it can be concluded that though there are diverse approaches of TCP developed by numerous researchers with the time being, a shift in the trend was observed, which was noticed during the study selection phase. In the starting period search-based approaches were utilized the most, then coverage based, then fault-based, then history based and then requirement based, followed by other approaches. It is observed that while a lot of work has been done on most of the popular approaches, now requirement-based prioritization holds a scope to work upon. As a system is built on its requirements, so by capturing maximum user requirement and testing those functionalities first which are related to those requirements will lead to delivery of the reliable product. Of the selected studies, 12 were based on requirement-based TCP which is approximately 33%.
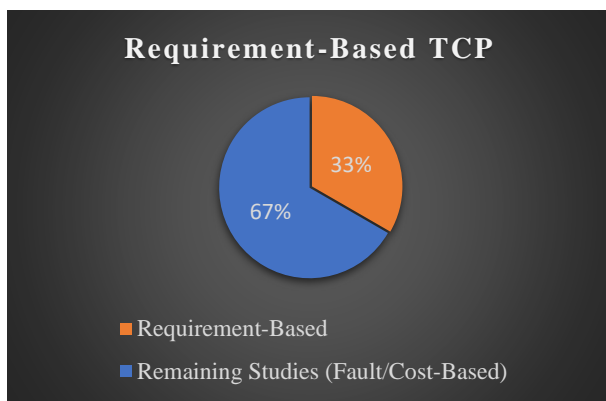
Fig. 6: Pie Chart Depicting Usage of Requirement-Based TCP Approaches in the Selected Studies.

For research question 2, it is observed that nature-based algorithms were not so popular earlier, to solve problems in computer science. A paradigm shift has been observed and various researchers have started using these techniques due to their high efficiency in reaching towards the optimal solution. Out of the selected studies, 13 have focused on the usage of nature-inspired algorithms to work on TCP techniques which are approximately 36%. Another inference which can be drawn from the analysis of the studies is that earlier only simple versions of nature-inspired techniques were in trends, such as ACO, BCO, and GA. The coming years has shown the trend of developing the modified versions of these techniques, as well as their usage in hybrid form. The reason for modifying these is to achieve efficient results in form of enhanced fault detection as compared to its previous versions. So, it can be said that nature-inspired techniques are also open to improvement.
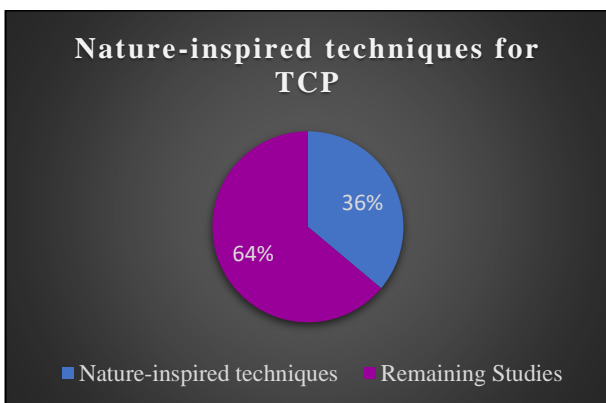

Fig. 7: Pie Chart Depicting Usage of Nature-Inspired TCP Approaches in the Selected Studies.

For research question 3, it is observed that though TCP techniques are favoured by researchers to work upon, they are also used as a hybrid with other regression testing techniques such as selection and minimization. Out of the selected studies, 9 were based on the usage of the hybrid approach. The reason to be concluded for the usage of the hybrid approach is to maximize as well as early fault detection so that improvements can be made. In these two or more approaches are used in such a conjunction that researchers were able to exploit the strengths of the combined approaches and their shortcomings were suppressed by each other. This has led the testing community to work upon it so that delivery of the fault-free and reliable product can be ensured.

7. Threats to validity

This study has some recognized limitations that may jeopardize its validity. Similar to previous reviews there are various potential threats which are-

- There might be a possibility that the authors were not able to access all the studies related to TCP which have answered all

the research questions. There could be two reasons for it. Firstly, the search string used is too narrow. Secondly, as the authors are well versed in the English language, therefore, they have selected studies which are in the English language only. So, some relevant non-English studies could have been missed though it cannot be avoided.

- The probability of occurrence of the problem of overfitting cannot be ruled out while studies were selected for answering the particular research question, though cross-checking of studies was done so as to avoid inclusion of inaccurate studies.

8. Conclusion

Test case prioritization (TCP) is one of the most popular approaches in regression testing. Diverse techniques of TCP have been proposed by numerous researchers. This study has worked to analyze the approaches based on the research questions framed. This study was performed by searching, classifying, analyzing and deducing outcomes from the selected studies. The motive behind this study is to find which of the approach, area or field holds a potential scope of improvement or holds future scope to work upon. From the conducted study, it can be deduced that the current research trend lies in developing a requirement-based TCP technique using the nature-inspired algorithm. This study has defined a base for future work by providing a deep insight to the testing community by exposing the latest trends. This will benefit the potential researchers in developing new techniques to find loopholes in the software system so that a time-bound, within budget, a reliable and robust product can be delivered to the users.

References

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

[2] Rothermel, G., Untch, R. H., Chu, C., & Harrold, M. J. (1999). Test case prioritization: An empirical study. In *Software Maintenance, 1999. (ICSM'99) Proceedings. IEEE International Conference on* (pp. 179-188). IEEE. https://doi.org/10.1109/ICSM.1999.792604.

[3] Rothermel, G., Untch, R. H., Chu, C., & Harrold, M. J. (2001). Prioritizing test cases for regression testing. *IEEE Transactions on software engineering*, *27*(10), 929-948. https://doi.org/10.1109/32.962562.

[4] Yoo, S., & Harman, M. (2012). Regression testing minimization, selection and prioritization: a survey. *Software Testing, Verification and Reliability*, *22*(2), 67-120. https://doi.org/10.1002/stv.430.

[5] Solanki, K., & Singh, Y. (2014). Importance of Selecting Test Cases for Regression Testing. *IOSR Journal of Computer Engineering (IOSRJCE) e-ISSN*, 2278-0661. https://doi.org/10.9790/0661-16444351.

[6] Elbaum, S., Kallakuri, P., Malishevsky, A., Rothermel, G., & Kanduri, S. (2003). Understanding the effects of changes on the cost-effectiveness of regression testing techniques. *Software testing, verification and reliability*, *13*(2), 65-83. https://doi.org/10.1002/stvr.263.

[7] Elbaum, S., Malishevsky, A. G., & Rothermel, G. (2002). Test case prioritization: A family of empirical studies. *IEEE transactions on software engineering*, *28*(2), 159-182. https://doi.org/10.1109/32.988497.

[8] Raju, S., & Uma, G. V. (2012). Factors oriented test case prioritization technique in regression testing using genetic algorithm. *European Journal of Scientific Research*, *74*(3), 389-402.

[9] Kumar, A., & Singh, K. (2014). A Literature Survey on test case prioritization. *Compusoft*, *3*(5), 793.

[10] Kiran, R. S. (2015). A literature survey on TCP-test case prioritization using the RT-regression techniques. *Global Journal of Research in Engineering*.

[11] Catal, C., & Mishra, D. (2013). Test case prioritization: a systematic mapping study. *Software Quality Journal*, *21*(3), 445-478. https://doi.org/10.1007/s11219-012-9181-z.

[12] Singh, Y., Kaur, A., Suri, B., & Singhal, S. (2012). Systematic literature review on regression test prioritization techniques. *Informatica*, *36*(4).

[13] Khatibsyarbini, M., Isa, M. A., Jawawi, D. N., & Tumeng, R. (2017). Test case prioritization approaches in regression testing: A systematic literature review. *Information and Software Technology*.

[14] Omdev, Solanki, K., Dalal, S., A Literature study of various regression testing approaches. InComputing for Sustainable Global Development (INDIACom), 2018 5th International Conference on 2018 Mar 15. IEEE. (In Press).

[15] Solanki, K., Singh, Y., & Dalal, S. (2016). A Comparative Evaluation of "m-ACO" Technique for Test Suite Prioritization. *Indian Journal of science and technology*, 9(30). https://doi.org/10.17485/ijst/2016/v9i30/86423.

[16] Solanki, K., Singh, Y., & Dalal, S. (2016). Experimental analysis of m-ACO technique for regression testing. *Indian Journal of Science and Technology*, 9(30). https://doi.org/10.17485/ijst/2016/v9i30/86588.

[17] Kitchenham, B. (2004). Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33(2004), 1-26.

[18] https://scholar.google.co.in.

[19] https://dl.acm.org.

[20] https://ieeexplore.ieee.org.

[21] https://www.elsevier.com/en-in.

[22] https://www.springer.com/in.

[23] Qu, X., Cohen, M. B., & Woolf, K. M. (2007, October). Combinatorial interaction regression testing: A study of test case generation and prioritization. In *Software Maintenance, 2007. ICSM 2007. IEEE International Conference on* (pp. 255-264). IEEE. https://doi.org/10.1109/ICSM.2007.4362638.

[24] Yoo, S., & Harman, M. (2007, July). Pareto efficient multi-objective test case selection. In *Proceedings of the 2007 international symposium on Software testing and analysis* (pp. 140-150). ACM.

[25] Elbaum, S., Rothermel, G., Kanduri, S., & Malishevsky, A. G. (2004). Selecting a cost-effective test case prioritization technique. *Software Quality Journal*, 12(3), 185-210. https://doi.org/10.1023/B:SQJO.0000034708.84524.22.

[26] Malishevsky, A. G., Ruthruff, J. R., Rothermel, G., & Elbaum, S. (2006). Cost-cognizant test case prioritization. *Department of Computer Science and Engineering, University of Nebraska-Lincoln, Techical Report*.

[27] Ramasamy, K., & Mary, S. A. (2008, December). Incorporating varying requirement priorities and costs in test case prioritization for new and regression testing. In *Computing, Communication and Networking, 2008. ICCCn 2008. International Conference on* (pp. 1-9). IEEE.

[28] Srikanth, H., Williams, L., & Osborne, J. (2005, November). System test case prioritization of new and regression test cases. In *Empirical Software Engineering, 2005. 2005 International Symposium on* (pp. 10-pp). IEEE.

[29] Jeffrey, D., & Gupta, N. (2007). Improving fault detection capability by selectively retaining test cases during test suite reduction. *IEEE Transactions on software Engineering*, 33(2). https://doi.org/10.1109/TSE.2007.18.

[30] Baudry, B., Fleurey, F., Jézéquel, J. M., & Le Traon, Y. (2005). Automatic test case optimization: A bacteriologic algorithm. *ieee Software*, 22(2), 76-82. https://doi.org/10.1109/MS.2005.30.

[31] Walcott, K. R., Soffa, M. L., Kapfhammer, G. M., & Roos, R. S. (2006, July). Timeaware test suite prioritization. In *Proceedings of the 2006 international symposium on Software testing and analysis* (pp. 1-12). ACM. https://doi.org/10.1145/1146238.1146240.

[32] Hla, K. H. S., Choi, Y., & Park, J. S. (2008, July). Applying particle swarm optimization to prioritizing test cases for embedded real time software retesting. In *Computer and Information Technology Workshops, 2008. CIT Workshops 2008. IEEE 8th International Conference on* (pp. 527-532). IEEE.

[33] Arafeen, M. J., & Do, H. (2013, March). Test case prioritization using requirements-based clustering. In *Software Testing, Verification and Validation (ICST), 2013 IEEE Sixth International Conference on* (pp. 312-321). IEEE.

[34] Suri, B., & Singhal, S. (2011). Implementing ant colony optimization for test case selection and prioritization. *International journal on computer science and engineering*, 3(5), 1924-1932.

[35] Krishnamoorthi, R., & Mary, S. S. A. (2009). Factor oriented requirement coverage based system test case prioritization of new and regression test cases. *Information and Software Technology*, 51(4), 799-808. https://doi.org/10.1016/j.infsof.2008.08.007.

[36] Raju, S., & Uma, G. V. (2012). Factors oriented test case prioritization technique in regression testing using genetic algorithm. *European Journal of Scientific Research*, 74(3), 389-402.

[37] Kavitha, R., Kavitha, V. R., & Kumar, N. S. (2010, October). Requirement based test case prioritization. In *Communication Control and Computing Technologies (ICCCCT), 2010 IEEE International Conference on* (pp. 826-829). IEEE.

[38] Suri, B., Mangal, I., & Srivastava, V. (2011). Regression test suite reduction using an hybrid technique based on BCO and genetic algorithm. *Special Issue of International Journal of Computer Science & Informatics (IJCSI), ISSN (PRINT)*, 2231-5292.

[39] Suri, B., & Singhal, S. (2011). Analyzing test case selection & prioritization using ACO. *ACM SIGSOFT Software Engineering Notes*, 36(6), 1-5. https://doi.org/10.1145/2047414.2047431.

[40] Suri, B., & Singhal, S. (2012, September). Literature survey of ant colony optimization in software testing. In *Software Engineering (CONSEG), 2012 CSI Sixth International Conference on* (pp. 1-7). IEEE.

[41] Salehie, M., Li, S., Tahvildari, L., Dara, R., Li, S., & Moore, M. (2011, March). Prioritizing requirements-based regression test cases: A goal-driven practice. In *Software Maintenance and Reengineering (CSMR), 2011 15th European Conference on* (pp. 329-332). IEEE. https://doi.org/10.1109/CSMR.2011.46.

[42] Catal, C. (2012, September). On the application of genetic algorithms for test case prioritization: a systematic literature review. In *Proceedings of the 2nd international workshop on evidential assessment of software technologies* (pp. 9-14). ACM. https://doi.org/10.1145/2372233.2372238.

[43] Gupta, V., & Chauhan, D. S. (2011, December). Hybrid regression testing technique: a multi layered approach. In *India Conference (INDICON), 2011 Annual IEEE* (pp. 1-5). IEEE. https://doi.org/10.1109/INDCON.2011.6139363.

[44] Kaur, A., & Goyal, S. (2011). A genetic algorithm for faultbased regression test case prioritization. *International Journal of Computer Applications*, 32(8), 975-8887.

[45] Di Nardo, D., Alshahwan, N., Briand, L., & Labiche, Y. (2015). Coverage-based regression test case selection, minimization and prioritization: A case study on an industrial system. *Software Testing, Verification and Reliability*, 25(4), 371-396. https://doi.org/10.1002/stvr.1572.

[46] Mayan, J. A., & Ravi, T. (2015). Structural software testing: hybrid algorithm for optimal test sequence selection during regression testing. *International Journal of Engineering and Technology (IJET)*, 7(1).

[47] Dobuneh, M. R. N., Jawawi, D. N., Ghazali, M., & Malakooti, M. V. (2014, September). Development test case prioritization technique in regression testing based on hybrid criteria. In *Software Engineering Conference (MySEC), 2014 8th Malaysian* (pp. 301-305). IEEE.

[48] Wang, X., & Zeng, H. (2016, May). History-based dynamic test case prioritization for requirement properties in regression testing. In *Continuous Software Evolution and Delivery (CSED), IEEE/ACM International Workshop on* (pp. 41-47). IEEE.

[49] Hettiarachchi, C., Do, H., & Choi, B. (2014, June). Effective regression testing using requirements and risks. In *Software Security and Reliability (SERE), 2014 Eighth International Conference on* (pp. 157-166). IEEE.

[50] Panichella, A., Oliveto, R., Di Penta, M., & De Lucia, A. (2015). Improving multi-objective test case selection by injecting diversity in genetic algorithms. *IEEE Transactions on Software Engineering*, 41(4), 358-383. https://doi.org/10.1109/TSE.2014.2364175.

[51] Ma, T., Zeng, H., & Wang, X. (2016, May). Test case prioritization based on requirement correlations. In *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2016 17th IEEE/ACIS International Conference on* (pp. 419-424). IEEE.

[52] Srikanth, H., Hettiarachchi, C., & Do, H. (2016). Requirements based test prioritization using risk factors: An industrial study. *Information and Software Technology*, 69, 71-83. https://doi.org/10.1016/j.infsof.2015.09.002.

[53] Silva, D., Rabelo, R., Campanha, M., Neto, P. S., Oliveira, P. A., & Britto, R. (2016, July). A hybrid approach for test case prioritization and selection. In *Evolutionary Computation (CEC), 2016 IEEE Congress on* (pp. 4508-4515). IEEE. https://doi.org/10.1109/CEC.2016.7744363.

[54] Achimugu, P., Selamat, A., Ibrahim, R., & Mahrin, M. N. R. (2014). A systematic literature review of software requirements prioritization research. *Information and software technology*, 56(6), 568-585. https://doi.org/10.1016/j.infsof.2014.02.001.

[55] Gao D, Guo X, Zhao L. (2015). Test case prioritization for regression testing based on ant colony optimization. *InSoftware Engineering and Service Science (ICSESS), 6th IEEE International Conference on* (pp. 275-279). IEEE. https://doi.org/10.1109/ICSESS.2015.7339054.

[56] Solanki, K., Singh, Y., Dalal, S., & Srivastava, P. R. (2016). Test case prioritization: An approach based on modified ant colony optimization. In *Emerging Research in Computing, Information, Communication and Applications* (pp. 213-223). Springer, Singapore.

[57] Vescan, A., Șerban, C., Chisăliță-Cretu, C., & Dioșan, L. (2017, September). Requirement dependencies-based formal approach for test case prioritization in regression testing. In *Intelligent Computer Communication and Processing (ICCP), 2017 13th IEEE International Conference on* (pp. 181-188). IEEE. https://doi.org/10.1109/ICCP.2017.8117002.

[58] Schwartz, A., & Do, H. (2016). Cost-effective regression testing through Adaptive Test Prioritization strategies. *Journal of Systems and Software*, *115*, 61-81. https://doi.org/10.1016/j.jss.2016.01.018.