International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 7 (2017) pp. 1389-1395

A Systematic Literature Review of the Test Case Prioritization Technique for Sequence of Events

Johanna Ahmad
Software Engineering and Information System Department
Universiti Putra Malaysia, 43 400, Serdang, Selangor, Malaysia.

Salmi Baharom
Software Engineering and Information System Department
Universiti Putra Malaysia, 43 400, Serdang, Selangor, Malaysia.

Abstract

Software testing is a set of activities in Software Development Life Cycle (SDLC) carried out with the intent of finding errors before deliver to the customer or top management. Generally, the software testing phase consumes 40 to 70 percent of development effort, time and costs especially for large software systems. Numerous techniques have been proposed to prioritize test cases based on certain coverage criteria with the aim to reduce time and costs of testing. This paper presents the results of a systematic literature review (SLR) of relevant primary studies as evidence of the application of test case prioritization (TCP) in the area of sequence of events. From 115 papers, the researchers identified that a total of 50 papers report complete empirical results. The review reports on different TCP used to guide the search for coverage criteria, type and size of the test cases, techniques under comparison and validity threats of previous works. One of the most significant issues in TCP is how researchers handled the same priority value issues. Most of the papers either applied random technique or did not provide any information regarding the same priority value issues. In addition, 20 percent of research emphasised the effectiveness of their technique by comparing it with the random technique. The aim of this SLR is to identify the background and limitations of previous works, serving as a starting point for guidelines on how the TCP technique can be adopted.

Keywords: test case prioritization; sequence of events; systematic review;

INTRODUCTION

Many researchers agreed that software testing is one of the most critical phases in software development in terms of costs, efforts and time. Test case prioritization (TCP) is an important technique to schedule the order of test execution and may be adopted either in regression or non-regression testing. Before 1997, people generated test cases randomly and selected the best test cases based on those randomly generated without knowing the quality of the test cases. For a large commercial system, test suites can contain thousands of test cases and can sometimes be infinite. A variety of test cases can be formed through a combination of events and input parameters. For industrial collaborators, [1] reported that they needed to run 200,000 lines of codes for seven weeks.

For this reason, numerous TCP techniques have been proposed to prioritize test cases especially for large commercial systems. The first TCP is known as a hybrid technique and combines modification, minimisation, and prioritisation-based selection [2]. Based on the literature, many researchers found techniques that can reduce effort, time and costs during testing. However, there are still a number of important issues neglected by previous researchers. This paper is structured as follows; Section 2 contains details on the systematic review process. In section 3, the extracted information is analyzed to answer the research questions. Section 4 contains discussion of the results.

REVIEW METHOD

The review processes for this SLR follows SLR guidelines for software engineering per [3],[4]. Referring to the guidelines, there are three main phases involved in this SLR: planning the review, conducting a review, and reporting the review, as shown in Fig. 1.

A. Research Questions

Research questions must be defined precisely, as they are the most important part of SLR [3]. The aim of this SLR is to understand and summarize the existing evidence on test case prioritization techniques. Besides that, the researcher attempts to understand how the existing test case prioritization technique caters to the same priority values.

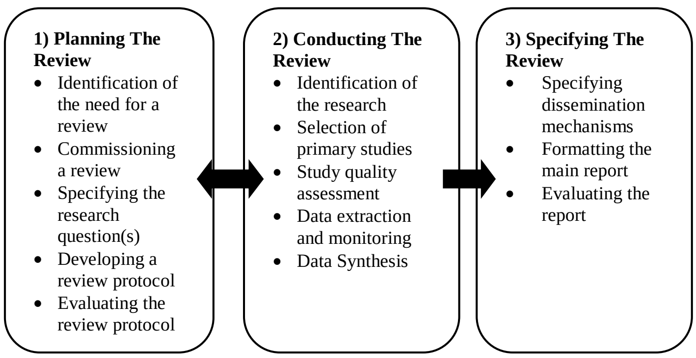

Figure 1. Systematic Literature Review (SLR) phases and stages

The research questions for this SLR comprise five components. These components known as PICOC and were proposed by [4]. Table I shows the criteria and scope of research questions.

TABLE I. CRITERIA AND SCOPE OF RESEARCH QUESTIONS

Criteria	**Scope**
Population	Sequence Based, Event Based, Search Based, State Based
Intervention	Test case prioritization technique
Comparison	NA
Outcomes	Structure of TCP technique applied in Sequence Based, Event Based, Search Based, State Based
Context	Review(s) of any empirical studies of the test case prioritization

To achieve the aim of this study, three primary questions and seven secondary questions have been defined as follows:

RQ1: How has test case prioritization been adapted and applied in prioritizing test cases in the form of sequence of events?
RQ1.1: What techniques are used to prioritize test case?
RQ1.2: Are test cases with the same priority values handled explicitly?
RQ2: How are the proposed techniques evaluated?
RQ2.1: Are the set of test cases for the experimental study based on industry, benchmark or case study?
RQ2.2: What is the number of test cases utilized for the experiments?
RQ2.3: What are the techniques under comparison?
RQ2.4: What are the evaluation metrics used to measure the performance of the proposed techniques?
RQ2.5: What are the main threats to the validity in the domain of test case prioritization?
RQ3: Is there any test case prioritization technique conducted in the area of sequence of events?

B. Data Sources

Ten online databases have been chosen as data sources: which are ACM Digital Library, Emerald, Elsevier, Google Scholar, IEEExplore Digital Library, ScienceDirect, Scopus, SpringerLink, Taylor & Francais and Wiley. The selection of online databases was based on the list available online database subscribed by the University Putra Malaysia's Library under the "Computer Science" subject category.

C. Search Strategy

Since the search process distinguishes systematic literature reviews from the traditional review, it is necessary to follow the search strategy. Trial searches with combinations of terms derived from the research questions need to be done before finalizing the search string. The following steps have been taken to formulate the search query for this SLR:

- Derivation of major terms from the research question.
- Identifying synonyms of major terms.
- Identifying keywords in relevant papers or books.
- Using Boolean OR for the alternatives synonyms or variants of each keyword.
- Using Boolean AND to link major terms.

The researcher attempted a number of search strings before arrive the conclusion that "*software testing*" as an expression is not suitable as a term, because many papers do not use these two words together even if related to software testing [5]. Nevertheless, the researcher would find too many unrelated papers if using the term "*testing*" alone. Based on that reason and number of trials, the researcher decided to use the term software and test linked together both terms with a Boolean AND. Using the terms *"software"* and *"test",* the researcher will find almost all papers related to software testing, and expression of *"test case prioritization"* as second major term [5]. The resulting search string is described as follows:

(("Software" AND "Test") OR "Test Case Prioritization") OR ("Test Case Prioritization" AND "Sequence Based") OR ("Test Case Prioritization" AND "Search Based") OR ("Test Case Prioritization" AND "Event Based") OR ("Test Case Prioritization" AND "State Based")

The researcher performed a manual check of the results of the search string and realised that some of the online databases needed advanced settings such as IEEExplore and Scopus. The researcher needs to add alternative words and expressions to the search string.

D. Study Selection

In this research, study selection involves selection of the online databases and identification of the search string, and applies inclusion and exclusion as identified in the early stage. The aim at this stage is to ensure the comprehensive of the selection of papers for this research.

E. Inclusion and Exclusion Criteria

Inclusion and exclusion criteria for this SLR is based on research question [3]. Since this SLR is focused on the test case prioritization technique, in order to select only relevant papers, it is necessary to define inclusion and exclusion criteria. The inclusion criteria are as follows:

- All papers must be published in English.
- All papers must be published from 1 January 2005 to 25 November 2015.
- All papers must be focused on test case generation and test case prioritization.

Each of the papers is filtered by exclusion criteria before being accepted for the next stage. The exclusion criteria for this SLR were as follows:

- Papers not published in English.
- Duplicate study areas.
- Papers that only contains opinion pieces, viewpoints, progress research or incomplete results.
- Theses.
- Papers with less than three pages.
- Papers that do not report any empirical study in their paper.

F. Data Extraction and Quality Assessment

The quality assessment for this study covered quantitative and qualitative studies since there is no restriction in terms of experimental design. A quality assessment study checklist is used to ensure the data extraction process meets the quality criteria [6]. The quality assessment checklist has been designed following guidelines proposed by [3]. Table II shows a list of general questions to measure the quality of selected studies. Three scale are coded for the quality assessment checklist and given a score; Yes =1; Partially = 0.5; No = 0. Based on the item checklist, each article ranged from 0 (very poor) to 5 (very good).

TABLE II. QUALITY ASSESSMENT CHECKLIST

No	Item	Answer
SQ 1	Are the aims and objectives of the research clearly stated?	Yes/No
SQ 2	Is the research design clearly specified?	Yes/No/Partially
SQ 3	Have the researcher(s) properly carried out the process of data collection?	Yes/No/Partially
SQ 4	Have the researcher(s) given enough data to support their results and conclusions?	Yes/No/Partially
SQ 5	Is there involved comparison of other technique in the experiment?	Yes/No

FINDINGS

From the first stage of the search process, the researcher found 2,294 papers using the defined search terms. After screening titles and abstracts, only 115 were potentially relevant. Each of the papers was filtered for referring inclusion and exclusion criteria before being accepted for the synthesis of evidence. At this stage, irrelevant studies and duplicate studies were eliminated. The researcher read the full articles if the titles and abstracts were not sufficient to categorize the paper relevant or not to the research area. Finally, 50 studies were selected for providing answers to the formulated research questions. Fig. 2 shows results of the search and selection papers process.

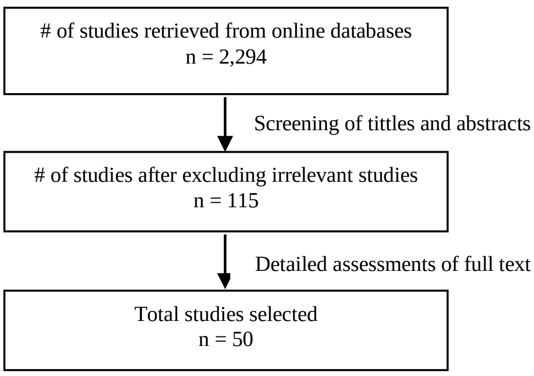

Figure 2. Search and Selection Papers Stage

A. Quality of Factors

Table III indicates the quality assessment scores for final identified papers consisting of 50 studies. 6 studies (12 percent) are fair, 9 studies (18 percent) are good and 35 studies (70 percent) are very good quality. None of the paper rated as poor quality. Therefore, all selected papers were included in the next phase for further analysis.

TABLE III. QUALITY ASSESSMENT SCORES

Quality Scale	**Very poor (>=1)**	**Poor (>=2)**	**Fair (>=3)**	**Good (>=4)**	**Very Good (=5)**	**Total**
Number of studies	0	0	6	9	35	50
Percentage (%)	0	0	12	18	70	100

DISCUSSIONS

This section presents and discusses the results related to the research questions. A detailed description of the findings will be presented in this section with the research questions and its sub-questions.

RQ1: How has test case prioritization been adapted and applied in prioritizing test cases in the form of sequence of events?

The purpose of this research question is to investigate and understand test case prioritization in the current situation. Researchers have shown increased interest in test case prioritization techniques, since it is very important to reduce time and cost during the testing phase. However, only a few research works have used the test case prioritization techniques for populations such as sequence based, event based, search based and state based. The RQ1 is divided into two sub-questions.

RQ1.1: What are techniques to prioritize test cases?

From the SLR studies, a majority of the papers (18 percent) implemented code coverage to prioritize test cases. Some of the papers combined techniques such as combination of code coverage, requirement coverage, and execution cost. Researchers have stated that they combined more than one technique to maximize the number of discovered faults [7]. Previous research has ruled out the execution cost. There could be reasons why some researchers applied more than one technique [5]. From the literature, some researchers agreed that using more than one coverage criteria can break ties and solve same priority value issues [8]. Based on the SLR evaluations, 20 percent combined more than one technique. Table IV depicts the most popular and utilized techniques based on SLR evaluations.

TABLE IV. MOST POPULAR AND UTILIZED TECHNIQUES

Techniques	Authors
Code Coverage	[9], [7], [10], [11], [12], [13], [14], [15], [16], [17]
Requirement Coverage	[18], [19], [20], [21], [22]
Fault Coverage	[23], [24], [25]
Interaction Coverage	[26], [27], [15], [28]

RQ1.2: Are test cases with same priority values handled explicitly?

In recent years, there has been increasing interest in test case prioritization. Some of the research can prioritize multiple test suites and test cases. Unfortunately, only 1 study reported that they handled same priority value cases. "*Break ties*" and "*same weight value*" term referred to same priority value issues. The study described in detail how their practical weight prioritization factors can solve same priority value issues. Four factors identified as the best factors to maximize the number of discovered faults: cost factors, time factors, defect factors and complex factors. Nevertheless, the subject program is not in the form of sequence of events. Most studies (98 percent) either applied random technique or did not provide any information regarding same priority value issues. Random methods can be categorized as poor weight algorithm and can caused wrong prioritization and bias issues [19]. This evidence means that there is a need to improve the ability of test case prioritization technique especially for the test cases in the form of sequence of events.

RQ2: How are the proposed techniques being evaluated?

In accessing the effectiveness and efficiency of existing technique, the reporting of the empirical study is very important. To avoid missing any information regarding the empirical study for each of the papers, RQ2 was further divided into five sub-questions as follows:

RQ2.1: Are the set of test cases for the experimental study based on industry, benchmark or case study?

Type of test cases used for the experiment is an important factor to validate the effectiveness of the proposed approach. In this SLR, the researchers defined three categories for type of test cases, industry-based, benchmark or case study. Most of the papers that used a case study agreed to apply real software system or industry for their future work [7],[19],[29],[30]. Of 50 papers, 25 reported that they used a benchmark as their test cases. 19 papers reported their test cases from a case study. The remaining 5 papers reported they applied real software system test cases for their experiment. Only one paper reported that they combined two industry projects and five case studies to determine that the proposed prioritization technique positively increased numbers of faults detection [20].

RQ2.2: Number of test cases utilized for the experiments.

Test case prioritization is considered an emerging research area. There are already wide sets of data and case studies that have been developed. Some of them have been published publicly for the use of others. With regards to the size of data sets, only 34 percent coming for large data sets in this review. The large data sets have up to 300,000 lines of codes and 200,000 test cases. 18 percent comes from medium data sets with up to 14,437 lines of codes and 5,000 test cases. A majority of the data sets at 48 percent come from small data sets. Small data sets have less than 1,000 lines of codes and fewer than 100 test cases. Most of the papers found it easy to test and manage if they applied small test sizes in the experiment. Small test sizes, means less effort, time, and cost for the experiment [19].

RQ2.3: Techniques under comparison.

Many techniques have been developed by numerous researchers since 1997. In order to validate the effectiveness of their TCP technique, 20 percent of researches compared their technique with random technique. The experimental results shows that their TCP technique is more effective than a random technique. However, existing TCP technique failed to resolves some issues which are considered as limitations to their research. The limitations will be used as a basis to improve the TCP technique. In summary, Table V depicts major limitations of existing TCP techniques.

TABLE V. MAJOR TCP TECHNIQUES LIMITATIONS

Limitations	Authors
Failed to prioritize multiple suites	[23], [21]
Failed to prioritize test case with same priority value	[7], [23], [29], [31], [32], [33], [30], [34], [12], [35], [13], [26], [20], [36], [8], [37], [38], [24], [39], [40], [41], [42], [25], [17], [43], [44], [45], [46], [47], [28], [48], [49], [50], [51], [52], [53]
Ignore the practical weight factors	[23], [33]
Test cases from small data sets	[7], [29], [18], [9], [54], [55], [27], [28], [21], [40], [56], [16], [57], [45], [47], [50], [22], [51]

RQ2.4: What is the evaluation metrics used to measure the performance of the proposed techniques?

Numerous evaluation metrics have been proposed in the selected primary studies that employ test case prioritization technique. The most frequently employed evaluation metric is Average of the Percentage of Faults Detected (APFD) with 58 percent. Average Percentage Statement Coverage (APSC) and Average Percentage of Faults Detected per Cost (APFDc), both were counted for 8 percent. Furthermore, 6 percent were counted for the usage of Normalize Percentage of Faults Detected (NAPFD). The remaining evaluation metrics reached as 2 percent, as shown in Table VI. In terms of the number of evaluation metric techniques employed, 12 primary studies used more than one evaluation metric.

TABLE VI. EVALUATION METRICS

Evaluation Metrics	Count of Usage	Percentage
Average of the Percentage of Faults Detected (APFD)	29	58%
APSC (Average Percentage Statement Coverage)	4	8%
Average Percentage of Faults Detected per Cost (APFDc)	4	8%
Normalized Percentage of Faults Detected (NAPFD)	3	6%
Remaining Evaluation Metrics	1	2%

RQ2.5: What are the main threats to the validity in the domain of test case prioritization?

In order to answer this research question, the researcher looked at the validity in terms of the effectiveness of the measurement [5]. The most frequently observed threats were found when researchers modified the codes and performed code inspection and testing to reduce failure bugs [36]. Besides that, most of the papers stated that they applied APFD only as a metric; therefore, they may get other results if other metrics are used, as APFD is not the only metric available [8], [36], [33], [48]. Table VII shows the most frequently threats researchers cater in their research. Out of 50 papers, 20 papers do not mention how they catered to threats in their experiments.

TABLE VII. VALIDITY THREATS

Threats	Number of papers	Percentage
Internal	29	58%
External	26	52%
Construct	23	46%
Conclusion	2	4%
NA	20	40%

RQ3: Is there any test case prioritization technique conducted in the area of sequence of events?

There has been a lot of research done into TCP since its proposal in 1997 [19]. Based on the SLR conducted, only 36 percent applied sequence of event as their subject programs in their TCP technique. However, none of the papers catered to the same priority value issue in their experiments. As mentioned in RQ 1.2, various TCP techniques applied random techniques to solve same priority value issues, even though most of the researchers agree that the random technique is ineffective.

CONCLUSIONS

The aim of this SLR was to identify the status of research on TCP in the area of sequence of events. Solving the same priority value issue is one of the concerns of the research into this SLR. The analysis in this SLR is based on the constructed research question. A total of 50 primary studies were selected and analyzed. Results of this SLR show that many papers have still not reported their empirical results completely. However, the results have shown that TCP research continues to increase annually as the TCP technique is increasingly important [14]. The essence of this SLR may be used to identify areas for possible improvement in TCP technique for sequences of events.

ACKNOWLEDGMENT

The authors appreciate the efforts of the community of Software Engineering Universiti Putra Malaysia (UPM) for supporting this research.

REFERENCES

[1] G. Rothermel, R. H. Untch, and M. J. Harrold, “Test Case Prioritization 1 Introduction,” *Test*, no. December, pp. 1–32, 1999.

[2] W. E. Wong, J. R. Horgan, S. London, and H. Agrawal, “A Study of Effective Regression Testing in Practice*,” *Softw. Reliab. Eng. 1997. Proceedings., Eighth Int. Symp.* , pp. 264–274, 1997.

[3] B. Kitchenham and S. Charters, “Guidelines for performing Systematic Literature Reviews in Software Engineering,” *Engineering*, vol. 2, p. 1051, 2007.

[4] M. Petticrew and H. Roberts, *Systematic Reviews in the Social Sciences: A Practical Guide*. 2006.

[5] S. Ali, L. C. Briand, H. Hemmati, and R. K. Panesar-Walawege, “A systematic review of the application and empirical investigation of search-based test case generation,” *IEEE Trans. Softw. Eng.*, vol. 36, no. 6, pp. 742–762, 2010.

[6] N. H. Hassan, Z. Ismail, and N. Maarop, “INFORMATION SECURITY CULTURE : A SYSTEMATIC,” no. 205, pp. 456–463, 2015.

[7] M. M. Islam, A. Marchetto, A. Susi, and G. Scanniello, “A Multi-Objective Technique to Prioritize Test Cases Based on Latent Semantic Indexing,” *2012 16th Eur. Conf. Softw. Maint. Reengineering*, no. 3, pp. 21–30, 2012.

[8] A. Khalilian, M. Abdollahi Azgomi, and Y. Fazlalizadeh, “An improved method for test case prioritization by incorporating historical test case data,” *Sci. Comput. Program.*, vol. 78, no. 1, pp. 93–116, 2012.

[9] R. Huang, J. Chen, T. Zhang, R. Wang, and Y. Lu, “Prioritizing Variable-Strength Covering Array,” *2013 IEEE 37th Annu. Comput. Softw. Appl. Conf.*, pp. 8–11, 2013.

[10] R. Blanco, J. García-Fanjul, and J. Tuya, “A First Approach to Test Case Generation for BPEL Compositions of Web Services Using Scatter Search,” *2009 Int. Conf. Softw. Testing, Verif. Valid. Work.*, pp. 131–140, 2009.

[11] S. Li, N. Bian, Z. Chen, D. You, and Y. He, “A Simulation Study on Some Search Algorithms for

Regression Test Case Prioritization," *2010 10th Int. Conf. Qual. Softw.*, pp. 72–81, 2010.

[12] P. K. Mishra, "Analysis of Test Case Prioritization in Regression Testing using Genetic Algorithm," vol. 75, no. 8, pp. 1–10, 2013.

[13] P. R. Srivastava, A. Vijay, B. Bariikha, P. S. Senear, and R. Sharma, "An optimized technique for test case generation and prioritization using 'tabu' search and data clustering," *Proc. 4th Indian Int. Conf. Artif. Intell. IICAI 2009*, pp. 30–46, 2009.

[14] J. Frolin S. Ocariza, G. Li, K. Pattabiraman, and A. Mesbah, "Automatic fault localization for client-side JavaScript," *Softw. Testing, Verif. Reliab.*, vol. Volume 21, no. Issue 3, pp. 195–214, 2015.

[15] D. D. N. B, A. Panichella, A. Zaidman, and A. De Lucia, "Search-Based Software Engineering," vol. 9275, pp. 157–172, 2015.

[16] M. N. Nawar and M. M. Ragheb, "Multi-heuristic Based Algorithm for Test Case Prioritization," pp. 449–460, 2014.

[17] J. F. Silva Ouriques, E. G. Cartaxo, and P. D. Lima Machado, "Revealing influence of model structure and test case profile on the prioritization of test cases in the context of model-based testing," *J. Softw. Eng. Res. Dev.*, vol. 3, no. 1, p. 1, 2015.

[18] L. S. De Souza, P. B. C. De Miranda, R. B. C. Prudencio, and F. D. a. Barros, "A Multi-objective Particle Swarm Optimization for Test Case Selection Based on Functional Requirements Coverage and Execution Effort," *2011 IEEE 23rd Int. Conf. Tools with Artif. Intell.*, pp. 245–252, 2011.

[19] S. Roongruangsuwan and J. Daengdej, "Test case prioritization techniques," *J. Theor. Appl. Inf. Technol.*, pp. 45–60, 2010.

[20] R. Krishnamoorthi and S. a. Sahaaya Arul Mary, "Factor oriented requirement coverage based system test case prioritization of new and regression test cases," *Inf. Softw. Technol.*, vol. 51, no. 4, pp. 799–808, 2009.

[21] K. K. Aggarwal, Y. Singh, and A. Kaur, "A multiple parameter test case prioritization model," *J. Stat. Manag. Syst.*, vol. 8, no. 2, pp. 369–386, 2005.

[22] A. P. Conrad, R. S. Roos, and G. M. Kapfhammer, "Empirically studying the role of selection operators duringsearch-based test suite prioritization," *Proc. 12th Annu. Conf. Genet. Evol. Comput. - GECCO '10*, p. 1373, 2010.

[23] M. Tyagi and S. Malhotra, "Test case prioritization using multi objective particle swarm optimizer," *2014 Int. Conf. Signal Propag. Comput. Technol. (ICSPCT 2014)*, pp. 390–395, 2014.

[24] Y. T. Yu and M. F. Lau, "Fault-based test suite prioritization for specification-based testing," *Inf. Softw. Technol.*, vol. 54, no. 2, pp. 179–202, Feb. 2012.

[25] P. Parashar, A. Kalia, and R. Bhatia, "Pair-Wise Time-Aware Test Case Prioritization," pp. 176–186, 2012.

[26] C. Y. Huang, J. R. Chang, and Y. H. Chang, "Design and analysis of GUI test-case prioritization using weight-based methods," *J. Syst. Softw.*, vol. 83, no. 4, pp. 646–659, 2010.

[27] R. Huang, J. Chen, D. Towey, A. T. S. Chan, and Y. Lu, "Aggregate-strength interaction test suite prioritization," *J. Syst. Softw.*, vol. 99, pp. 36–51, Jan. 2015.

[28] R. C. Bryce, S. Sampath, J. B. Pedersen, and S. Manchester, "Test suite prioritization by cost-based combinatorial interaction coverage," *Int. J. Syst. Assur. Eng. Manag.*, vol. 2, no. 2, pp. 126–134, 2011.

[29] B. Jiang, W. K. Chan, and T. H. Tse, "PORA: Proportion-Oriented Randomized Algorithm for Test Case Prioritization," *2015 IEEE Int. Conf. Softw. Qual. Reliab. Secur.*, no. 61202077, pp. 131–140, 2015.

[30] L. Chen, Z. Wang, L. Xu, H. Lu, and B. Xu, "Test case prioritization for web service regression testing," *Proc. - 5th IEEE Int. Symp. Serv. Syst. Eng. SOSE 2010*, pp. 173–178, 2010.

[31] H. Srikanth and M. B. Cohen, "Regression testing in Software as a Service: An industrial case study," *2011 27th IEEE Int. Conf. Softw. Maint.*, pp. 372–381, 2011.

[32] R. K. Saha, "An Information Retrieval Approach for Regression Test Prioritization Based on Program Changes," *Proceeding 37th Int. Conf. Softw. Eng. (ICSE 2015)*, pp. 268–279, 2015.

[33] M. Staats, P. Loyola, and G. Rothermel, "Oracle-Centric Test Case Prioritization," *2012 IEEE 23rd Int. Symp. Softw. Reliab. Eng.*, pp. 311–320, 2012.

[34] Z. Li, M. Harman, and R. M. Hierons, "Search Algorithms for Regression Test Case Prioritization," *IEEE Trans. Softw. Eng.*, vol. 33, no. 4, pp. 225–237, 2007.

[35] S. Khandai, A. A. Acharya, and D. P. Mohapatra, "Prioritizing Test Cases Using Business Criticality Test Value," *Int. J. Adv. Comput. Sci. Appl.*, vol. 3, no. 5, pp. 103–110, 2012.

[36] B. Jiang and W. K. Chan, "Input-based adaptive randomized test case prioritization: A local beam search approach," *J. Syst. Softw.*, vol. 105, pp. 91–106, 2015.

[37] S. Sampath and R. C. Bryce, "Improving the effectiveness of test suite reduction for user-session-based testing of web applications," *Inf. Softw. Technol.*, vol. 54, no. 7, pp. 724–738, Jul. 2012.

[38] Y.-C. Huang, K.-L. Peng, and C.-Y. Huang, "A history-based cost-cognizant test case prioritization technique in regression testing," *J. Syst. Softw.*, vol. 85, no. 3, pp. 626–637, 2012.

[39] H. Srikanth and S. Banerjee, "Improving test efficiency through system test prioritization," *J. Syst. Softw.*, vol. 85, no. 5, pp. 1176–1187, 2012.

[40] A. Khajeh-Hosseini, D. Greenwood, J. Smith, and I. Sommerville, "The Cloud Adoption Toolkit: supporting cloud adoption decisions in the enterprise," *Softw. - Pract. Exp.*, vol. 43, no. 4, pp. 447–465, 2012.

[41] S. Elbaum, P. Kallakuri, A. Malishevsky, G. Rothermel, and S. Kanduri, "Understanding the effects of changes on the cost-effectiveness of regression testing techniques," *Softw. Testing, Verif. Reliab.*, vol.

13, no. 2, pp. 65–83, 2003.
[42] F. Yuan, Y. Bian, Z. Li, and R. Zhao, "Search-Based Software Engineering," vol. 9275, pp. 109–124, 2015.
[43] L. Briand, Y. Labiche, and K. Chen, "A Multi-objective Genetic Algorithm to Rank State-Based Test Cases," pp. 66–80, 2013.
[44] C. Dubois, Y. Fazlalizadeh, A. Khalilian, M. Abdollahi Azgomi, and S. Parsa, "Incorporating Historical Test Case Performance Data and Resource Constraints into Test Case Prioritization," *Lect. Notes Comput. Sci.*, vol. 5668, pp. 43–57, 2009.
[45] A. K. Pandey and V. Shrivastava, "Early fault detection model using integrated and cost-effective test case prioritization," *Int. J. Syst. Assur. Eng. Manag.*, vol. 2, no. 1, pp. 41–47, 2011.
[46] C. Fang, Z. Chen, K. Wu, and Z. Zhao, "Similarity-based test case prioritization using ordered sequences of program entities," *Softw. Qual. J.*, pp. 1–27, 2013.
[47] W. Zhang, B. Wei, and H. Du, "Test Case Prioritization Based on Genetic Algorithm and Test-Points Coverage Evaluation of Test Case Prioritization," pp. 644–654, 2014.
[48] D. Hao, X. Zhao, and L. Zhang, "Adaptive Test-Case Prioritization Guided by Output Inspection," *2013 IEEE 37th Annu. Comput. Softw. Appl. Conf.*, pp. 169–179, 2013.
[49] L. Zhang, J. Zhou, D. Hao, L. Zhang, and H. Mei, "Jtop: Managing JUnit test cases in absence of coverage information," *ASE2009 - 24th IEEE/ACM Int. Conf. Autom. Softw. Eng.*, pp. 677–679, 2009.
[50] D. Garg, "Parallel Execution of Prioritized Test Cases for Regression Testing of Web Applications," no. Acsc, pp. 61–68, 2013.
[51] S. Kim and J. Baik, "An effective fault aware test case prioritization by incorporating a fault localization technique," *Proc. 2010 ACM-IEEE Int. Symp. Empir. Softw. Eng. Meas. - ESEM '10*, p. 1, 2010.
[52] S. Yoo, M. Harman, P. Tonella, and A. Susi, "Clustering test cases to achieve effective and scalable prioritisation incorporating expert knowledge," *Proc. ISSTA*, pp. 201–212, 2009.
[53] M. G. Epitropakis, S. Yoo, M. Harman, and E. K. Burke, "Empirical evaluation of pareto efficient multi-objective regression test case prioritisation," *Proc. 2015 Int. Symp. Softw. Test. Anal. - ISSTA 2015*, pp. 234–245, 2015.
[54] F. B. Universit, "KI 2009: Advances in Artificial Intelligence," vol. 5803, no. October, 2009.
[55] C. P. Indumathi and K. Selvamani, "Test Cases Prioritization Using Open Dependency Structure Algorithm," *Procedia Comput. Sci.*, vol. 48, no. Iccc, pp. 250–255, 2015.
[56] Z. Li, Y. Bian, R. Zhao, and J. Cheng, "A fine-grained parallel multi-objective test case prioritization on GPU," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8084 LNCS, pp. 111–125, 2013.
[57] Y. B. B, S. Kirbas, M. Harman, Y. Jia, and Z. Li, "Search-Based Software Engineering," vol. 9275, pp. 221–227, 2015.