A Systematic Literature Review of Test Case Prioritization Technique on Software Product Line Testing

Siti Hawa Mohamed Shareef[1], Rabatul Aduni Sulaiman[1*], and Abd Samad Hasan Basari[2]
[1,2] Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia,
86400 Parit Raja, Johor, Malaysia
[e-mail: rabatul@uthm.edu.my]
[*]Corresponding author: Rabatul Aduni Sulaiman

Received July 17, 2023; revised February 28, 2024; revised July 30, 2024; accepted October 8, 2024; published October 31, 2024

Abstract

Software product lines (SPL) leverage extensive reuse to enhance quality and competitiveness. However, the inherent high testing risks underscore the need for efficient techniques. Test case prioritization (TCP) emerges as a pivotal strategy for improving defect detection by optimizing the arrangement of test cases, thereby maximizing benefits, particularly when testing is prematurely halted. This paper presents a comprehensive review of the latest TCP techniques, investigating emerging trends, varied approaches, and the associated cost-effectiveness trade-offs. Through a systematic literature review spanning from 2011 to 2023 and employing specific search terms, this study scrutinizes the existing body of research. The analysis reveals a growing demand for TCP in research over the past five years, accompanied by a positive upsurge in SPL testing trends. TCP proves instrumental in orchestrating test cases from high to low priority levels, facilitating early defect detection and subsequent error resolution. Nonetheless, notable gaps persist in terms of time execution and coverage, prompting ongoing research efforts aimed at enhancing the cost-effectiveness of TCP. While TCP serves as a potent regression technique in software testing, existing approaches stand to benefit from the reordering of test cases and the integration of refined methodologies gleaned from current research endeavors.

Keywords: Systematic literature review, Software product line, Software product line testing, Regression testing, Test case prioritization

1. Introduction

Software product line (SPL) development, which involves creating a collection of software products using software engineering methods, tools, and techniques, addresses the increasing quality requirements and competition in a global market [1]. SPL employs a formal procedure for designing modules based on prediction, enabling testers to reuse software assets to address a variety of problems. Additionally, SPL proves beneficial for organizations facing resource constraints. The introduction of SPL facilitates companies in achieving improvements related to time to market, cost, productivity, quality, and other business drivers [2]. Within the realm of SPL, software product line testing (SPLT) encompasses core software, product-specific software, and their interactions. SPLT presents a challenging task within a product line context compared to individual software testing due to the inability to test all SPL products. Consequently, numerous techniques, including the SPL prioritization technique, have been proposed to mitigate testing phase challenges while aiming to detect a large number of defects with reduced effort and ensuring adequate test coverage [3].

Regression testing works to determine that changes are correct and have not affected the unchanged parts in the context of the SPL [4]. As the software developed, the test suite tended to increase in size, which often made it costly to implement.[5]. Yoo and Harman [6] claimed that regression testing is one of the simplest approaches to overcome this problem by executing all existing test cases in the test suite; this is called the retesting all approach. Among the existing methods, regression testing offers the minimization of test artefacts via minimization, selection, and prioritisation techniques. These techniques have been adapted in software testing and in the SPL domain context.

Based on **Table 1**, three testing approaches in relation to compare how the test results is adapted into SPL. Test case minimization is a technique for regression testing where all the assets in an existing test case need to be re-executed. However, this method is very expensive as it requires a lot of time and resources. Second, selection testing is a technique where a series of selected test cases from a test suite are run. This testing technique can affect code changes in the software application. Test cases are divided into two categories, namely reusable test cases and test cases that cannot be used in the next cycle. Finally, test cases are prioritised based on their impact on the business and on critical and frequently used functions. Test cases are selected based on preferences. This method indirectly reduces the number of regression tests.

Table 1. Types of Regression Testing

Component	Types of Regression Testing		
	Minimization	**Selection**	**Prioritization**
Strategy	Eliminate test cases.	Focuses on modifying and selecting relevant test cases.	Reorder test cases based on fault detection rate.
Strength	Reduce testing required.	Effective in faults finding for test cases.	Prioritizes test cases to maximize fault detection with minimal execution time.
Limitation	Minimize a set of test cases to be re-executed.	Tester has limited time to evaluate multiple products.	Ineffective for handling diverse case study sizes.

SPL requires an efficient testing technique to enhance the quality of test cases, utilizing TCP. To implement this technique effectively, the development of a new product is necessary to improve quality standards. The effectiveness of test cases can also be gauged through defect detection rates. In this technique, the generated test cases are typically classified as good quality based on cost and effectiveness measures [2]. This classification allows prioritization of test cases that are not only cost-effective but also efficient in detecting defects, thereby optimizing testing resources and improving overall software quality. These two measurements are widely used to generate good test cases.

TCP is a regression technique that can be used to overcome the problem of performing regression testing when changes are made to existing software by rearranging test cases that contain many elements of product variant changes to achieve the desired criteria [6]. TCP can be related to the process of prioritizing test cases in a test suite based on various factors. This is the process of ordering the test cases that will eventually be executed. TCP is very important to meet two important constraints, which are time and cost, in order to increase the fault detection rate.

Although in the literature there are many approaches to TCP, there is lack of recent advanced literature review describing the current importance of TCP in SPLT. Therefore, this study provides a systematic literature review (SLR) based on existing TCP approaches. SLR is a method of reviewing research to improve research recommendations based on evidence [5]. This SLR is presented as follows. TCP approaches have been reviewed in Section 2. After described an approach, Section 3 explains the plan chosen to conduct this SLR. Then, section 4 covers the results and discussion. Next, Section 5 will explain the research finding. The threat to the validity will be discussed in Section 6. Finally, a conclusion is drawn in Section 7.

2. Existing Studies of Test Case Prioritization

SPL discusses previous studies on TCP in this section. From the collected literature, the authors compile four SLRs related to TCP and SPL, as shown in **Table 2**.

Table 2. Summary of Existing SLR

Type of Study	Study References	Study Aims	Year Publication	Total of Studies Reviewed	Years Covered
SLR	[5]	Regression Test Case Prioritization Approaches	2017	159	1977 - 2007
	[7]	Regression Test Case Prioritization Approaches	2018	36	2004 - 2018
	[8]	Test Case Prioritization	2019	384	1999 - 2018
Mapping	[9]	Prioritization in Automotive Software Testing	2018	25	2012

Following **Table 2**, author [5] and [7] have delved into TCP approaches within the realm of regression testing. In the study by author [5], the focus lies on enhancing the efficiency of software testing within regression testing scenarios. They emphasize the criticality of a robust software testing environment in augmenting the commercial value of a system. Their research, spanning from 1977 to 2007, encompasses a thorough examination of 159 studies. While various approaches have been explored, the overarching goal of TCP remains consistent: enhancing fault detection effectiveness within regression testing contexts.

Similarly, in the work of author [7], a review of 36 studies conducted between 2004 and 2018 sheds light on the retesting process within software systems, encompassing both existing and newly introduced test cases. Consequently, the authors propose several techniques aimed at optimizing the performance of regression testing, rendering it a cost-effective endeavour. Given the continual evolution of prioritization methodologies, it becomes imperative to scrutinize the analyses of prioritization systems to identify the most effective strategies. This SLR seeks to delve into the underlying motives driving the adoption of each TCP approach and its applicability within SPLT, while also addressing pertinent trade-offs between cost and effectiveness.

Furthermore, the SLR defined by [8] offers a comprehensive exploration of the utilization of genetic algorithms in TCP, spanning studies from 1999 to 2018. Their rigorous selection process resulted in the inclusion of 31 full-text studies out of 384 initially identified studies. This SLR not only examines the current landscape of genetic algorithm implementation in TCP but also endeavours to classify and assess its efficacy. In contrast to our proposed SLR, which aims to adapt prioritization techniques specifically for SPLT, this prior study provides insights into the broader landscape of genetic algorithm utilization in TCP.

In contrast, the study by author [9], focuses on a systematic mapping of TCP, with particular emphasis on automotive applications and emerging functions and systems aimed at meeting the market demand for an expanding array of software-intensive functions. Their evaluation, comprising 25 studies in 2012, underscores the importance of prioritization techniques within the context of evolving software systems. In our research, spanning from 2009 to 2023, encompasses numerous studies addressing SPL test prioritization techniques. **Table 3** compares the findings from the related studies to those from the present SLR study.

Table 3. Summary of Related Studies

Study Author	Comparison of Similar Findings	Uncovered Findings Added into This SLR
[5]	Trends in TCP approaches.	Detailed overview of approaches and process involved in TCP.
[7]	Minimize time, cost, and effort during the software testing phase.	Causes of TCP approach trends.
[8]	Reduce time and effort required for regression testing and TCP approaches details overview.	Detailed overview of GA application in TCP technique.
[9]	Minimize time, cost, and effort during the software testing phase, trends in prioritization and method used in TCP.	Uncovered quality evaluation of prioritization.

Existing studies highlight the drive to minimize time, cost, and effort expended during the software testing phase across various TCP approaches. However, there remains an incomplete understanding of emerging trends within certain TCP methodologies, warranting further investigation. In summary, there are critical insights gleaned from existing literature that merit elucidation, particularly concerning the adaptation of prioritization techniques for SPLT and the ongoing optimization of TCP approaches to address evolving testing challenges.

3. Research Method

To fulfil the main purpose of this SLR, a structured method for reviewing studies on TCP is presented. **Fig. 1** shows how the structured research method will be used in the SLR and there are five main stages of the review protocol which are research question and motivation,

repository selection, search strategy, study selection and data synthesis and extraction. First, the phases identify the research question and motivation to achieve the main goal of reviewing the work. After the research question is developed, the next phase is repository selection to continue the process. The next phase after the selection includes the search strategy, which includes two points, namely the search term and the search process. The result after the search strategy is the study selection, which consists of quality assessment and inclusion and exclusion criteria. Finally, after going through all four phases, the data synthesis and extraction of the primary study are used for this SLR. The systematic and structured methodology employed in this study draws inspiration from Kitchenham seminal work [9] on guidelines for performing systematic literature reviews in software engineering, as well as insights from Achimugu research [10] on systematic review methodologies in a related domain.

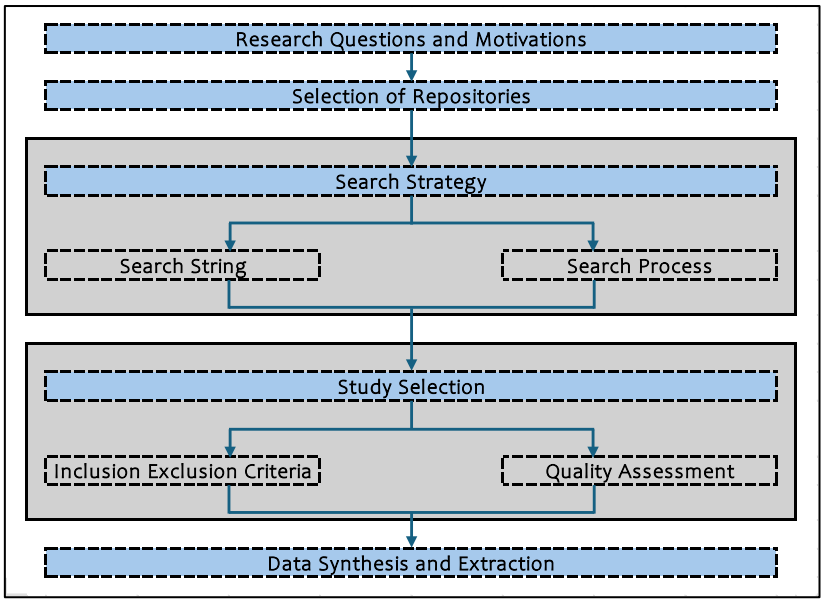

Fig. 1. Phases of Review Protocol

3.1 Research Questions and Their Motivations

A research question is the starting point for any good research. It provides a roadmap for proceeding and identifying and focusing on research gaps. To achieve this goal, three research questions were formulated. Research objectives are actions to answer the listed research questions. This research question serves to answer further findings that will be explored in this SLR. To present this clearly, **Table 4** represents each research question based on corresponding research question for additional insights and their significance. The significance of the outcome for each RQ was also detailed as a guide to achieving the objectives of this SLR study.

Table 4. Research Questions with Significance

Research Questions	Possible Outcomes	Extra Outcomes	Significance of the Outcomes
RQ 1	The trends for each TCP approach.	Discussion of approaches in TCP techniques.	To specify recent insights and trends in TCP development.
RQ 2	The overview of TCP approaches.	Strengths and limitations of existing prioritization approaches.	To provide information on how TCP functions propose ways to improve the said approaches.
RQ 3	Metric to evaluate cost and effectiveness measure for TCP outcome.	Ways to improve cost and effectiveness by existing TCP approaches.	To provide a multi criteria of cost and effectiveness measurement to researcher

3.2 Study Strategy

The research strategy provides the overall direction of the research, including the research process. Primary studies usually determine the SLR value. The strategy for this review is based on these three phases, as follows:

i. Literature of selected repository
ii. Search of strings identification
iii. Study of selection process

3.2.1 Literature of selected repository

The process begins with a few selections of keywords are entered for example, "test case" AND "prioritization" AND " software product line" as a search string with exact phrases on online resources database. Based on the search results, the repository in the TCP domain area was determined to collect the primary research data from identified repositories. The selected repositories are as follows:

a. Google Scholar
b. IEEE Xplore
c. Science Direct
d. Springer Link
e. Scopus

The results after selecting these online databases revealed that Science Direct provides a large number of journal articles, while IEEE Xplore provides more conference-type articles. The remaining repository covered journal articles, conferences, books, and dissertations on TCP obtained from Google Scholar, Springer Link, and Scopus libraries.

3.2.2 Search of Strings Identification

Search string identification is a crucial aspect of conducting a systematic literature review. A search string is a combination of keywords and phrases that are used to identify relevant studies for the review. The process of identifying the search string involves selecting appropriate databases, defining inclusion and exclusion criteria, and refining the search terms to ensure that all relevant studies are captured.

a. RQs are used to determine significant terms.
b. Analyzing significant terms to identify equivalents.
c. Identifying relevant keywords in studies.
d. Use of Boolean operators 'AND' as alternative links between terms.

Fig. 2 is the search string used in each repository. To retrieve potential significant reviews, we used the terms: *"test case"*, *"prioritization"*, and *"software product line"*. In order to create an alternative search term, we used the *'AND'* operator to link "test case" phrase, *"prioritization"* phrase, and *"prioritization"* phrase. From 2011 to 2023, the published years will be covered.

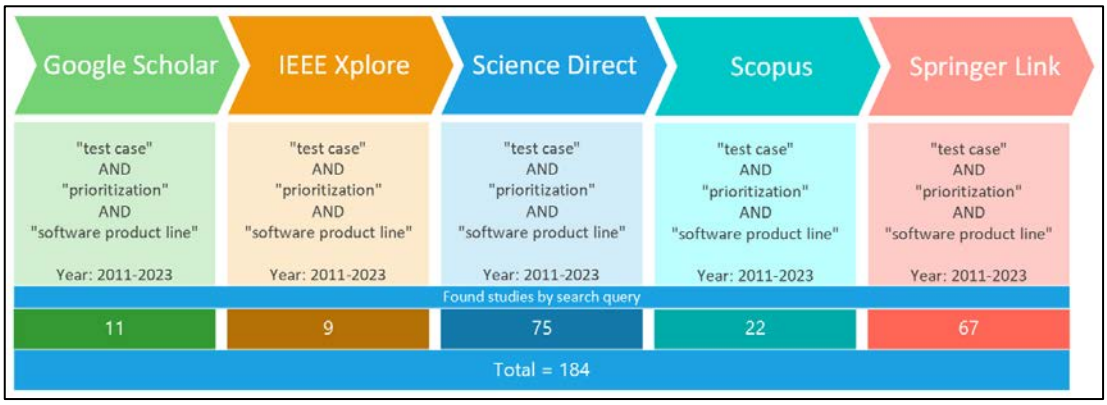

Fig. 2. Search Strings Method from Repositories

3.2.3 Study of Selection Process

Based on **Fig. 3**, 183 papers were successfully identified from the search phase, including Google Scholar 10 papers, IEEE Xplore 9 papers, Science Direct 75 papers, Scopus 22 papers, and Springer Link 67 papers. These selected papers will undergo the selection process. At this stage, inclusion and exclusion criteria must be applied. **Table 5** presents criteria for inclusion and exclusion. This process is important to eliminate duplicate papers in other search databases and unrelated studies. As part of the selection phase, quality assessments are carried out after the inclusion and exclusion phases. After completing the selection phase, 77 primary studies met all the research questions previously posed, which meant that they could be included in the analysis.

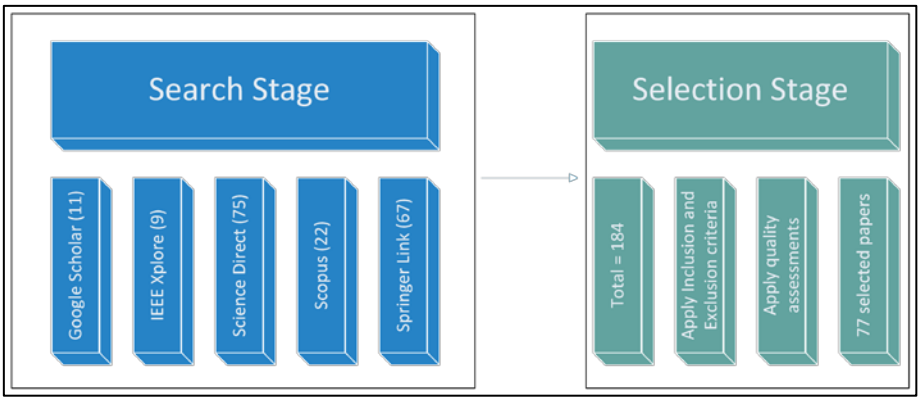

Fig. 3. Search and Selection Process

3.3 Data Synthesis and Extraction Method

The data synthesis starts with data extracted from a collection of studies. The process continues with the process of organizing the information in a way that allows understanding of the study and provides conclusions based on findings. This method is used to identify patterns in the data, including conflicting results and inconsistent data, to answer the defined research questions. Thus, the 77 selected primary studies were subjected to additional scrutiny based on the content assessment step. In order to determine the input from selected studies**,** **Table 6** presented the steps. The evaluation measures for selecting information help summarise the studies and draw conclusions. This method helps in extracting data from papers that include the type of paper nominated and a description for each item. **Table 7** presents an overview of the summarized data that related to the research questions.

Table 5. Inclusion and Exclusion Criteria

Inclusion Criteria	Exclusion Criteria
Paper written in English.	Paper written in other languages (non-English).
Paper focusing on TCP approaches.	Paper not related with TCP approaches.
At least one research question can be addressed in the paper.	Studies of the same nature (latest paper is chosen).
Paper in SLR, survey, and systematic mapping.	Paper not in SLR, survey, and systematic mapping.

Table 6. Contents Assessment Measures

Paper Nominated	Description
References	Title, publication year, and sources of the paper
Types	Thesis, book, journal article, conference and proceeding
Aims	Concepts, complications, inspiration, and purposes
Research methodology	Review, literature survey, case study, and experiment
Domain	Situation description
Limitations	Limitations of the study that can be improved in the future

Table 7. Data Collection for Research Questions

Research Questions	Type of Data Extracted
RQ 1	Research trend of TCP in SPL testing.
RQ 2	Type of approaches implemented for TCP.
RQ 3	Trade-off between cost and effectiveness.

4. Result and discussion

In this section, the results of the SLR based on defined research questions will be clearly discussed. This section presents an overview of the primary study, followed by an explanation of each research question.

4.1 Overview of primary study

After going through several processes in the previous section, 77 primary studies were selected. Four types of papers were identified from the primary study, namely 58 journal articles, 13 conference papers, four books, and two dissertations. These papers were selected since the defined criteria of the research question are met. The most searched publication sources were

Google Scholar, IEEE Xplore, Science Direct, Scopus, and Springer Link. The percentage of primary studies is shown in **Fig. 4**. Followed by the number of papers published per year in **Fig. 5**.

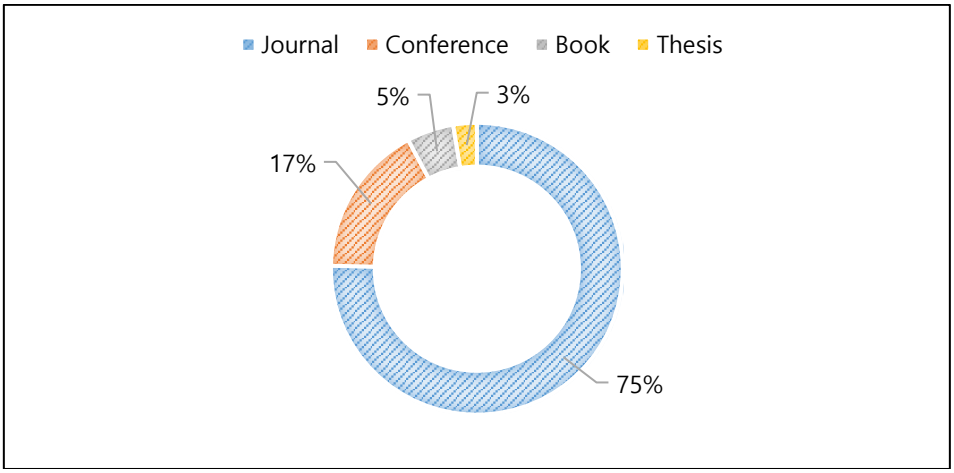

Fig. 4. Percentage of Primary Study

4.2 What is the research trend of TCP in SPL testing? (RQ 1)

Based on the selected studies, the SPL test question determines the research trend for TCP. In this section, we first identified the current trend of studies on TCP techniques and discussed the statistics of the nominated studies, such as the source and distribution of publications**. Fig. 5** shows that the number of research papers increases positively from 2015 to 2017. However, the number slightly decreased after 2017. The graph shows a positive increase in the last five years. Consequently, TCP is an essential part of SPL, according to researchers.

The TCP approach also needs some other approaches, since in TCP it is sometimes quite difficult to identify which tests detect a fault [5]. Therefore, the implementation of test cases in the new approach indirectly contributes to enhancing debugging through various means. This conclusion is further supported by **Fig. 6**, which illustrates that TCP approaches can be categorized into seven main dimensions that are highly sought after by researchers. For the search-based approach, 19 primary studies were found, which is the highest number of approaches among researchers. The second-highest number is the model-based approach, for which 17 primary studies were found. Apart from the number, the third highest number of approaches is similarity-based, as found in 10 primary studies. followed by fault-based approaches in four primary studies and coverage-based approaches in three primary studies.

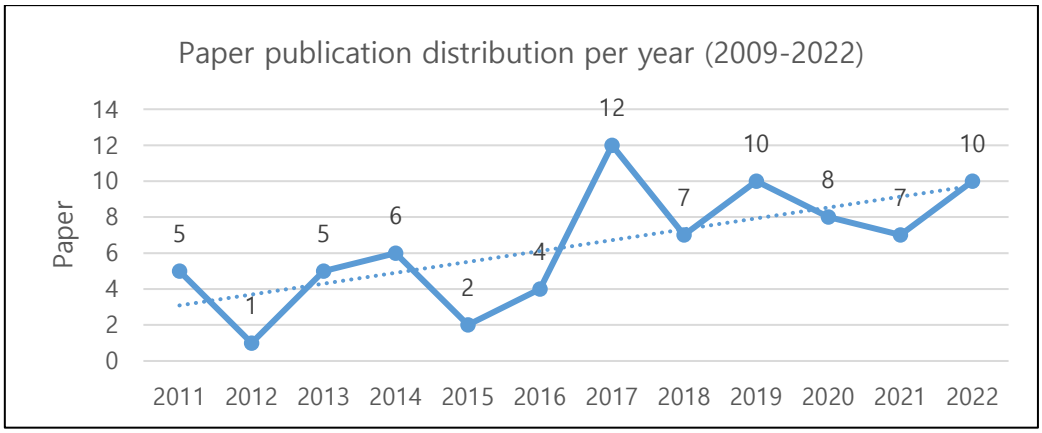

Fig. 5. Number of Papers Published Per Year

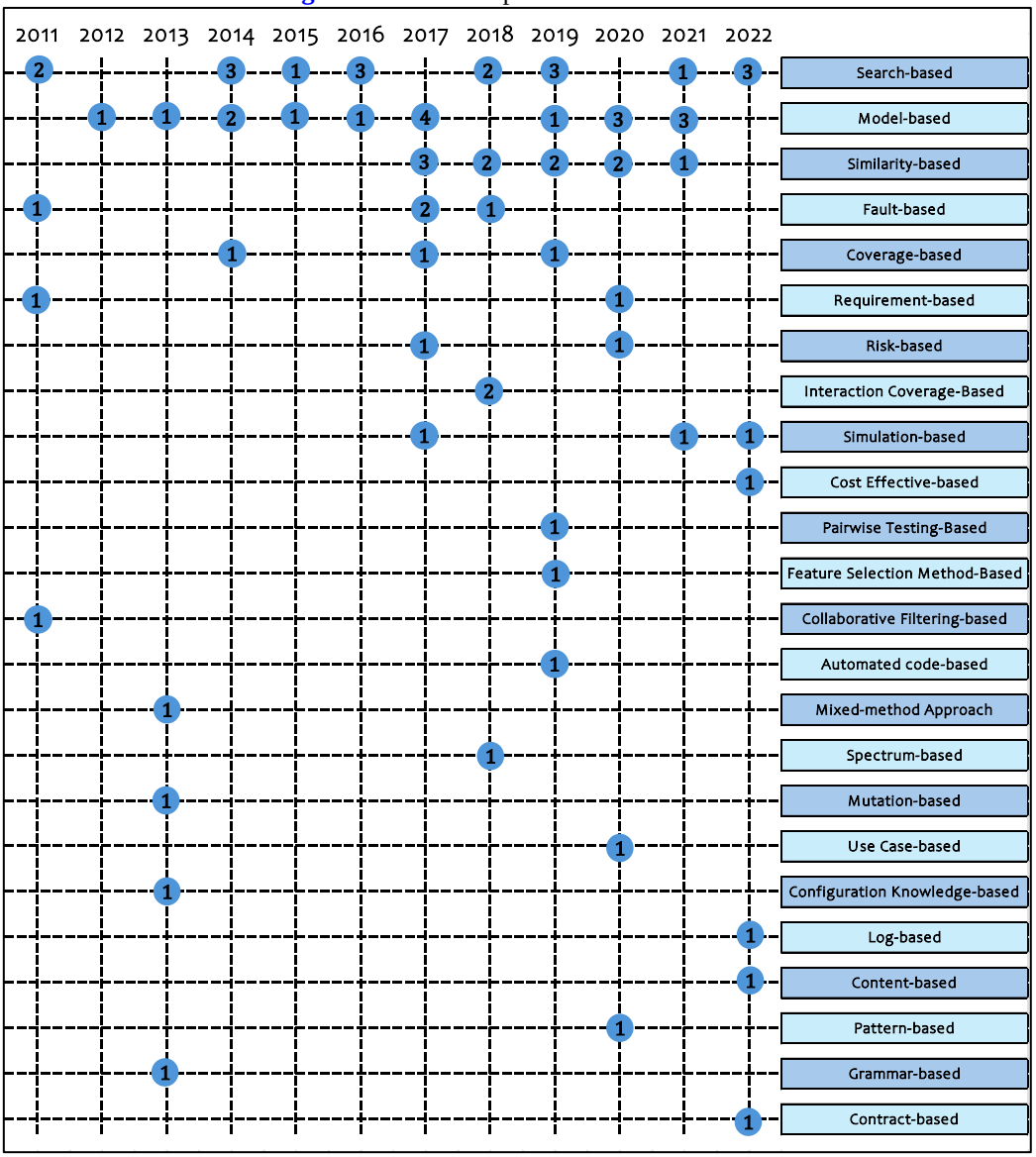

Fig. 6. Bubble Plot of Taxonomy of TCP

However, some approaches were grouped under the "Other" dimension, which were low in demand by researchers, including requirements-based, risk-based, interaction coverage-based, simulation-based, pairwise testing-based, feature selection methods-based, collaborative filtering-based, automated code-based, mixed methods-based, spectrum-based, mutation-based, use case-based, configuration knowledge-based, log-based, content-based, pattern-based, grammar-based, and contract-based approaches. Each approach provides different procedures and data sets for performing regression testing.

4.3 What are the approaches implemented for TCP in SPL Testing? (RQ 2)

The next aspect is discussed for RQ 2. This is related to the approach implemented for TCP in SPLT proposed by previous researchers. **Fig. 7** shows the percentage of primary study approaches for TCP. As outlined in **Table X1** in the **Appendix**, citations are listed for each TCP approach as well.

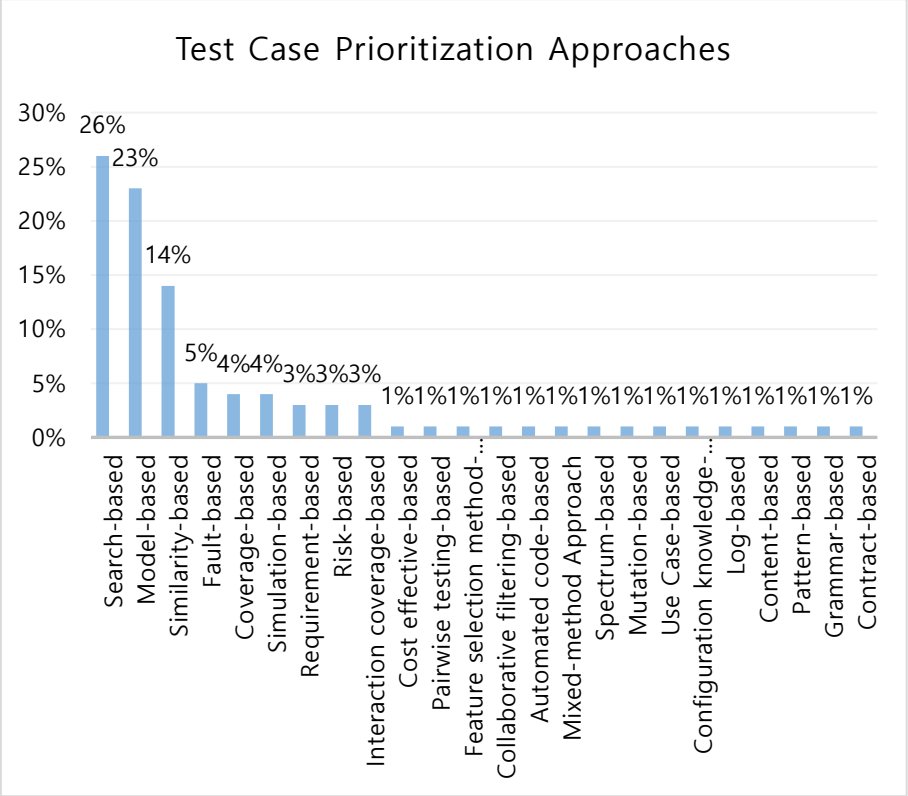

Fig. 7. Percentages of Primary Study Based on TCP Approaches

The result showed the search-based TCP is the approach most used by previous researchers, with 19 papers in the studies [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29]. Search-based TCP mostly focus on several algorithms used to include Genetic Algorithm (GA) [11], [13], [14], [15], [18], [19], [22], [23], [24], [25], [26], GA-ANFIS and PSO-ANFIS [12], Computational Intelligence (CI) as a Service (CIaaS) [16], Automated Analysis of Feature Models (AAFM) [17], Indicator-based Evolutionary Algorithm (IBEA) [15], and Adaptive Fitness Function Selection (AFFS) [21]. The search-based approach is a common method used by researchers to implement and organize their

results.

There are 17 papers [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46] reporting model-based TCP as the second most common approach in the primary survey. A model-based TCP has been proposed to help test managers improve communication about testing and use testing resources efficiently. To address the approach, some authors also highlight an approach based on dissimilarity and string-based distance. This technique can be used to evaluate the distance between test cases [30]. In addition, sampling and prioritization techniques have been proposed to create ordered lists of products based on coverage criteria or feature weights [41]. Based on the model, this technique does not consider the use of product behaviour as a source of preference. In this work, this idea has motivated further model-based publications to improve the approach in terms of time and resources as the researchers evaluate the feasibility of integrating the usage model into the testing process to obtain a statistical testing approach for SPL.

Similarity-based TCP become a third largest number most utilized approach reported in the primary study, with 10 papers [47], [48], [49], [50], [51], [52], [53], [54], [55], [56]. Researchers propose improved string spacing and prioritization algorithms that can reorder test cases resulting in higher fault detection rates [47]. Others study agree early damage detection can also be selected as a key performance goal for researcher [49]. Apart from the previously proposed techniques, TCP also proposed an equality testing approach for role-based access control (RBAC) systems and compared it to simple inequality and random preference. This RBAC equation is a technique for combining the degree of dissimilarity of test case pairs with respect to the RBAC policy being tested to maximise the diversity of tests and their constraint coverage [54].

In fourth place comes the fault-based TCP presented in the four papers of the primary study [57], [58], [59], [60]. This approach proposes the use of metamorphic tests to automate test case generation from a feature model that consists of tools. This approach is used to overcome the Oracle issue in SPL testing [57]. In addition to the metamorphosis test, the effectiveness is also compared by measuring the percentage of pairs protected, the percentage of weight protected, and the damage detection rate [58]. This technique can make it easier for researchers to find defects that should be detected.

Coverage-based and simulation-based TCP were highlighted in three primary study [61], [62], [63], [64], [65], [66]. This is the fifth-highest number of primary study approaches. To solve this problem, various suggestions have been made to minimize the number of assets to be tested while still aiming to achieve good coverage. The authors proposed five priority criteria based on similar metrics for feature models. They also compared the proposed approach in terms of early fault detection rate [62].

Requirement-based, risk-based, and interaction coverage-based shows the same percentages, with two of paper the primary study [1], [67], [68], [69], [70], [71]. While for other approaches which is cost effective-based, pairwise testing-based, feature selection method-based, collaborative filtering-based, automated code-based, mixed-method approach, spectrum-based, mutation-based, use case-based, configuration knowledge-based, log-based, content-based, pattern-based, grammar-based, and contract-based the percentages show 1 of paper in the primary study [2], [4], [72], [73], [74], [75], [76], [77], [78], [79], [80], [81], [82], [83], [84], which did not seem to be popular.

4.4 How researchers handle trade-off issues of cost and effectiveness in TCP for SPL? (RQ 3)

To answer and discuss this research question, this section addresses the issue of TCP cost and

effectiveness for SPL. For any proposed TCP approach, it is important to implement metric measurements to evaluate its effectiveness in TCP techniques. Evaluation metrics are important to measure the effectiveness of a TCP approach in prioritizing test cases and to compare its effectiveness with other existing approaches.

Fig. 8 shows the TCP evaluation metric divided into two groups, namely cost and effectiveness measures. After the cost group, which uses the most time of execution with a distribution of 79%. Followed by test suite size (20%) and average percent failure detection cost (APFDc) (1%). Execution time is clearly the most commonly used metric preferred by researchers in primary studies [1], [2], [4], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [35], [36], [38], [39], [41], [43], [44], [46], [47], [48], [49], [50], [51], [52], [53], [54], [55], [57], [58], [60], [61], [62], [63], [64], [65], [66], [67], [68], [69], [70], [71], [72], [73], [74], [76], [77], [78], [79], [80], [81], [82], [84]. Execution time is primarily used in search-based TCP approaches. This metric is used to test minimization of time based on the effectiveness of a proposed algorithm. The authors present a pairwise algorithm that uses ConstRaints, Order, and Weight (PROW) to handle constraints and preferences for pairwise coverage [25]. Further supporting the importance of trade-offs, size of test suites shows 20% [4], [12], [21], [27], [29], [37], [43], [44], [47], [51], [61], [66], [80], [81], [82], [83], [84]. It is the ratio between the total size of the test suite and the coverage of the reconstructed test suite that detects all faults or satisfies all requirements.

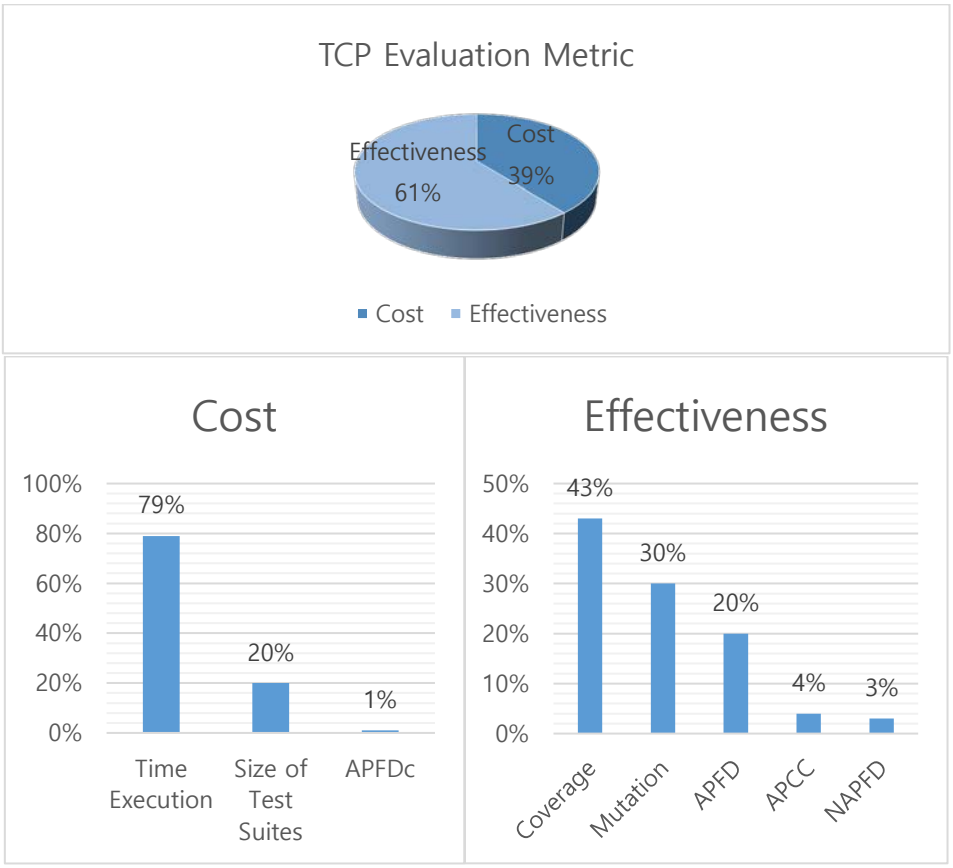

Fig. 8. TCP Evaluation Metric

In the effectiveness group, coverage can have 43%, mutation 30%, average percentage faults detected (APFD) 20%, average percentage of wise combinations covered (APCC) 4% and normalized average percentage faults detected (NAPFD) 3%. Coverage comes as the second largest number of evaluations metric with 43% in TCP evaluation metrics [2], [4], [12], [14], [15], [20], [21], [22], [24], [25], [26], [27], [29], [30], [32], [33], [35], [36], [37], [38], [41], [42], [43], [44], [45], [46], [49], [50], [51], [52], [53], [54], [55], [56], [58], [59], [60], [61], [62], [63], [64], [65], [66], [68], [69], [70], [71], [72], [73], [74], [76], [77], [78], [80], [82], [83], [84]. Most coverage is for TCP approaches, search-based, model-based, and similarity-based. This metric shows the multiple coverage for each approach, among them is defined some author, with higher coverage requirements will be more effective for reduction in a regression test suite, the experimental result already compared with existing techniques [12].

Apart from that, mutation ranks second in the effectiveness group with 30% [2], [11], [12], [13], [14], [15], [16], [17], [19], [20], [21], [22], [23], [24], [26], [27], [29], [30], [32], [36], [38], [43], [46], [48], [50], [54], [55], [57], [58], [63], [65], [66], [68], [69], [70], [71], [78], [80], [81], [84] fault-based criteria, such as those based on mutation tests, have been studied for variability tests using the FM [14]. In addition, some authors concluded that mutation testing is used in FSM-based test methods to derive test cases from FSM models and evaluate whether the behaviour of a SUT conforms to its specification [54]. This trade-off is often used in search-based approaches.

In the context of evaluating TCP in SPLT, APFD metric plays a significant role. This metric considers both the coverage and order of execution of test cases, providing insights into the effectiveness of TCP approaches. Trade-offs arise when assessing TCP effectiveness using APFD, typically involving the balancing of objectives such as maximizing fault detection while minimizing testing time or resources. The percentages show 20% from trade-off evaluation [2], [27], [30], [37], [42], [43], [47], [49], [50], [51], [52], [53], [54], [58], [60], [61], [62], [63], [65], [68], [69], [70], [71], [78], [80], [82]. Most of the trade-off problems were inherited from APFD, which has since become APFDc [63], APCC [42], [50], [61], [70], [71] and NAPFD [2], [58], [69], [80] to satisfy various TCP objectives. This trade-off problem is commonly used in similarity-based approaches.

5. Research Findings

SPL is used to handle the commonality and variability of business applications to meet the specific needs or goals of a particular market. In SPL, TCP was introduced to provide an approach to executing important test cases first and then producing the desired results. In this context, the impact of TCP in SPLT is very important and needs to be highlighted.

In answering RQ1 regarding TCP research trends in SPL, 77 studies were evaluated after inclusion and exclusion stages were applied. All the primary study are used to answer this question. Result from the study selection process, obtained four types of paper was found, which is journal, conferences, book, and thesis **Fig. 5** shows that the number of articles published per year increased positively from 2012 to 2017. Although the graph shows a slight decrease from 2017 until 2022, the demand for TCP in research is still growing. Based on the results, TCP approaches were identified and grouped into seven dimensions. Based on the results obtained, TCP approaches are very important and are still used by researchers. New approaches in TCP continue to be introduced among researchers until now.

RQ2 discusses TCP issue approaches in SPL testing. Referring **Fig. 7**, the percentages of primary study approaches are showing three approaches sitting in the highest ranking, which

is search-based, model-based and similarity-based. Moreover, in determining the advantages and limitations, the input and the type of data set also play an important role in this issue. For example, the search-based approach, the author reports the effect of different approach characteristics, such as search space size, tuning budget, tuning algorithm, and number of classes to tune, on the results [15]. What can be concluded here is that the TCP approach serves as motivation for any change in the future in SPLT. The results of the study found that there are some techniques that are not widely used among researchers, the technique is cost effective-based, pairwise testing-based, feature selection method-based, collaborative filtering-based, automated code-based, mixed-method approach, spectrum-based, mutation-based, use case-based, and configuration knowledge-based. However, these techniques can still be improved by researchers.

In the search-based approach with the highest ranking in the primary study according to **Fig. 7**, there are several techniques that are highlighted, among them are the Genetic Algorithm (GA), AAFM, AFFS, GA-ANFIS and PSO-ANFIS, the Drag and Drop approach, and the Evolutionary Algorithm (IBEA). These techniques have been able to resolve the issue in TCP, where GA-ANFIS and PSO-ANFIS can be effectively used for reduction in test suites AAFM is a mature topic and has stimulated several other topics such as product configuration, derivation, testing, development, and reverse engineering. GA is a technique widely used by researchers; this technique has succeeded solve TCP problem by automatically generating valid feature selections that optimize desired product properties.

For RQ3, trade-off issues of cost and effectiveness in for TCP in SPL testing was discussed. Referring the **Fig. 8** TCP evaluation metric which has been divided into two groups namely, cost and effectiveness. In cost measure, three types of metrics are found, which is time execution, size of test suite and APFDc. While the effectiveness group indicates five types, which is coverage, mutation, APFD, APCC and NAPFD. The evaluation metrics used in this primary study and the reasons for their creation are discussed. Execution time is still the primary metric used in all TCP approaches. In order to support multiple objectives in separate studies, APFD has introduced and adopted several new metrics.

6. Threats to Validity

Several risks to validity may arise in the implementation of SLR. To ensure the quality of SLR, the validity threats must be eliminated. In this section, the threat is analysed and minimised.

i. Internal validity:

When conducting research, the internal validity of the study can be one of the major threats to the study conducted. This threat can be minimized by some procedures, such as formulating a good search term. The keywords for the search term were carefully formulated in a database of online resources. Then, the researcher conducted a database search, which included searches in four reliable databases, and collected relevant literature. Since the four reliable databases do not contain all sources in this research area, the limitation is that we do not know if we found all methods of historical preference. However, we tried to perform the following steps well. An evaluation of quality and inclusion and exclusion criteria was used to select studies irrelevant studies were excluded from consideration. To improve the quality of the search and reduce the risk of overlooking relevant studies, data synthesis and extraction of primary studies were performed for this SLR.

ii. External validity:

The first external validity is that the research is general, and the conclusion is also general. We cannot capture reasonable sources, but we have taken practical steps to minimize the impact of this threat, such as creating research questions and finding their motivations. Research questions are the starting point of any good research. They provide the roadmap for moving forward and identifying and focusing on research gaps in the SLR.

iii. Conclusion validity:

The validity of conclusions refers to hazards that affect the ability to draw correct conclusions from the research conducted. One potential risk to the validity of conclusions is the reliability of the data extraction strategy. To minimize this risk, data extraction features are designed to allow researchers to focus on research questions formulated to extract the correct features from the literature.

7. Conclusion

This SLR paper contains the primary studies of SPLT based on TCP that have been published between 2009 and 2023. In order to obtain a finding, the review process has been conducted. A process was conducted for the purpose of identifying and evaluating relevant primary studies. A synchronization process was performed on the data extracted from the study. Several findings can be drawn from the study, as follows:

i. It is still necessary to improve a number of prioritization techniques.
ii. Further exploration is necessary to classify data in TCP, encompassing diverse sources like requirements, system models, and source code. This deeper analysis enables researchers to better understand how varied data types of influence TCP effectiveness, aiding informed decision-making in software testing.
iii. Clarifying the implementation timeline for TCP approaches is crucial, offering insights into practical considerations and resource needs. This documentation aids practitioners and researchers in planning and executing TCP strategies effectively within their specific contexts.
iv. The proposed algorithm for evaluating TCP time reduction, especially in the search-based TCP approach, significantly addresses gaps in timing execution and coverage. Through systematic assessment of TCP effectiveness in reducing testing time, this algorithm offers crucial guidance for optimizing testing processes and enhancing overall efficiency in software development workflows.
v. To compare TCP's effectiveness with existing approaches and measure its effectiveness in prioritizing test cases, evaluation metrics are necessary.
vi. The proposed algorithm evaluates TCP time reduction using a search-based TCP approach, addressing gaps in timing execution and coverage.

Acknowledgement

This research was supported by Ministry of Higher Education (MOHE) through Fundamental Research Grant Scheme (FRGS/1/2022/ICT01/UTHM/03/2).

Appendix

This appendix section contains **Table X1**.

Table X1. Existing Approaches and Their Citation Indexes

No.	Approaches	Source
1	Search-based	[11]–[29]
2	Model-based	[30]–[46]
3	Similarity-based	[47]–[56]
4	Fault-based	[57]–[60]
5	Coverage-based	[61]–[63]
6	Requirement-based	[1], [67]
7	Risk-based	[68]–[69]
8	Interaction Coverage-Based	[70]–[71]
9	Simulation-based	[64]–[66]
10	Cost Effective-based	[2]
11	Pairwise Testing-Based	[3]
12	Feature Selection Method-Based	[72]
13	Collaborative Filtering-based	[73]
14	Automated code-based	[74]
15	Mixed-method Approach	[75]
16	Spectrum-based	[76]
17	Mutation-based	[77]
18	Use Case-based	[78]
19	Configuration Knowledge-based	[79]
20	Log-based	[80]
21	Content-based	[81]
22	Pattern-based	[82]
23	Grammar-based	[83]
24	Contract-based	[84]

References

[1] R. Rabiser, P. O'Leary, and I. Richardson, "Key activities for product derivation in software product lines," *Journal of Systems and Software*, vol.84, no.2, pp.285-300, Feb. 2011. Article (CrossRef Link)

[2] J. A. Prado Lima, W. D. F. Mendonça, S. R. Vergilio, and W. K. G. Assunção, "Cost-effective learning-based strategies for test case prioritization in continuous integration of highly-configurable software," *Empir Softw Eng.*, vol.27, no.6, Jul. 2022. Article (CrossRef Link)

[3] M. Idham, S. A. Halim, D. N. A. Jawawi, Z. Zakaria, A. Erianda, and N. Arss, "Test Case Prioritization for Software Product Line: A Systematic Mapping Study," *JOIV: International Journal on Informatics Visualization*, vol.7, no.3-2, pp.2126-2134, 2023. Article (CrossRef Link)

[4] P. Jung, S. Kang, and J. Lee, "Automated code-based test selection for software product line regression testing," *Journal of Systems and Software*, vol.158, Dec. 2019. Article (CrossRef Link)

[5] M. Khatibsyarbini, M. A. Isa. D. N.A. Jawawi, and R. Tumeng, "Test case prioritization approaches in regression testing: A systematic literature review," *Information and Software Technology*, vol.93, pp.74-93, 2018. Article (CrossRef Link)

[6] S. Yoo and M. Harman, "Regression testing minimization, selection and prioritization: a survey," *Software Testing, Verification and Reliability*, vol.22, no.2, pp.67-120, 2012. Article (CrossRef Link)
[7] O. Dahiya and K. Solanki, "A systematic literature study of regression test case prioritization approaches," *International Journal of Engineering and Technology*, vol.7, no.4, pp.2184-2191, 2018. Article (CrossRef Link)
[8] A. Bajaj and O. P. Sangwan, "A Systematic Literature Review of Test Case Prioritization Using Genetic Algorithms," *IEEE Access*, vol.7, pp.126355-126375, 2019. Article (CrossRef Link)
[9] A. Dadwal, H. Washizaki, Y. Fukazawa, T. Iida, M. Mizoguchi, and K. Yoshimura, "Prioritization in Automotive Software Testing: Systematic Literature Review," in *Proc. of 6th International Workshop on Quantitative Approaches to Software Quality (QuASoQ 2018)*, 2018. Article (CrossRef Link)
[10] P. Achimugu, A. Selamat, R. Ibrahim, and M. N. Mahrin, "A systematic literature review of software requirements prioritization research," *Inf Softw Technol.*, vol.56, no.6, pp.568-585, 2014. Article (CrossRef Link)
[11] J. Guo, J. White, G. Wang, J. Li, and Y. Wang, "A genetic algorithm for optimized feature selection with resource constraints in software product lines," *Journal of Systems and Software*, vol.84, no.12, pp.2208-2221, Dec. 2011. Article (CrossRef Link)
[12] Z. Anwar, H. Afzal, N. Bibi, H. Abbas, A. Mohsin, and O. Arif, "A hybrid-adaptive neuro-fuzzy inference system for multi-objective regression test suites optimization," *Neural Comput & Applic.*, vol.31, no.11, pp.7287-7301, Nov. 2019. Article (CrossRef Link)
[13] W. K. G. Assunção, T. E. Colanzi, S. R. Vergilio, and A. Pozo, "A multi-objective optimization approach for the integration and test order problem," *Information Sciences*, vol.267, pp.119-139, May 2014. Article (CrossRef Link)
[14] R. A. Matnei Filho and S. R. Vergilio, "A multi-objective test data generation approach for mutation testing of feature models," *Journal of Software Engineering Research and Development*, vol.4, no.1, Dec. 2016. Article (CrossRef Link)
[15] S. Zamani and H. Hemmati, "A pragmatic approach for hyper-parameter tuning in search-based test case generation," *Empir Software Eng.*, vol.26, no.6, Nov. 2021. Article (CrossRef Link)
[16] P. Oliveira, M. Souza, R. Braga, R. Britto, R. L. Rabelo, and P. S. Neto, "Athena: A Visual Tool to Support the Development of Computational Intelligence Systems," in *Proc. of 2014 IEEE 26th International Conference on Tools with Artificial Intelligence ICTAI*, pp.950-959, 2014. Article (CrossRef Link)
[17] J. A. Galindo, D. Benavides, P. Trinidad, A.-M. Gutiérrez-Fernández, and A. Ruiz-Cortés, "Automated analysis of feature models: Quo vadis?," *Computing*, vol.101, no.5, pp.387-433, May 2019. Article (CrossRef Link)
[18] B. J. Garvin, M. B. Cohen, and M. B. Dwyer, "Evaluating improvements to a meta-heuristic search for constrained interaction testing," *Empir Software Eng.*, vol.16, no.1, pp.61-102, Feb. 2011. Article (CrossRef Link)
[19] Hitesh and A. C. Kumari, "Feature Selection Optimization in SPL using Genetic Algorithm," *Procedia Comput Sci.*, vol.132, pp.1477-1486, 2018. Article (CrossRef Link)
[20] Y. Xue et al., "IBED: Combining IBEA and DE for optimal feature selection in software product line engineering," *Applied Soft Computing*, vol.49, pp.1215-1231, Dec. 2016. Article (CrossRef Link)
[21] H. Almulla and G. Gay, "Learning how to search: generating effective test cases through adaptive fitness function selection," *Empir Software Eng.*, vol.27, no.2, Mar. 2022. Article (CrossRef Link)
[22] S. Wang, D. Buchmann, S. Ali, A. Gotlieb, D. Pradhan, and M. Liaaen, "Multi-objective test prioritization in software product line testing: an industrial case study," in *Proc. of SPLC '14: Proceedings of the 18th International Software Product Line Conference*, pp.32-41, Association for Computing Machinery, Sep. 2014. Article (CrossRef Link)

[23] M. W. Mkaouer, M. Kessentini, S. Bechikh, M. Ó Cinnéide, and K. Deb, "On the use of many quality attributes for software refactoring: a many-objective search-based software engineering approach," *Empir Software Eng*, vol.21, no.6, pp.2503-2545, Dec. 2016. Article (CrossRef Link)

[24] H. L. Jakubovski Filho, T. N. Ferreira, and S. R. Vergilio, "Preference based multi-objective algorithms applied to the variability testing of software product lines," *Journal of Systems and Software*, vol.151, pp.194-209, May 2019. Article (CrossRef Link)

[25] B. Pérez Lamancha, M. Polo, and M. Piattini, "PROW: A Pairwise algorithm with constRaints, Order and Weight," *Journal of Systems and Software*, vol.99, pp.1-19, Jan. 2015. Article (CrossRef Link)

[26] T. Ferreira, S. R. Vergilio, and M. Kessentini, "Variability testing of software product line: A preference-based dimensionality reduction approach," *Inf Softw Technol.*, vol.152, Dec. 2022. Article (CrossRef Link)

[27] C. Birchler, S. Khatiri, P. Derakhshanfar, S. Panichella, A. Panichella, "Single and Multi-objective Test Cases Prioritization for Self-driving Cars in Virtual Environments," *ACM Transactions on Software Engineering and Methodology*, vol.32, no.2, pp.1-30, Apr. 2023. Article (CrossRef Link)

[28] K. Pohl and A. Metzger, "Software Product Lines," *The Essence of Software Engineering*, pp.185-201, 2018. Article (CrossRef Link)

[29] T. K. Akila and A. Malathi, "Test case prioritization using modified genetic algorithm and ant colony optimization for regression testing," *International Journal of Advanced Technology and Engineering Exploration*, vol.9, no.88, pp.384-400, Mar. 2022. Article (CrossRef Link)

[30] R. A. Sulaiman, D. N. A. Jawawi, and S. A. Halim, "A Dissimilarity with Dice-Jaro-Winkler Test Case Prioritization Approach for Model-Based Testing in Software Product Line," *KSII Transactions on Internet and Information Systems*, vol.15, no.3, pp.932-951, Mar. 2021. Article (CrossRef Link)

[31] F. Damiani, D. Faitelson, C. Gladisch, and S. Tyszberowicz, "A novel model-based testing approach for software product lines," *Softw Syst Model.*, vol.16, no.4, pp.1223-1251, Oct. 2017. Article (CrossRef Link)

[32] H. Lackner and B.-H. Schlingloff, "Advances in Testing Software Product Lines," *Advances in Computers*, vol.107, pp.157-217, 2017. Article (CrossRef Link)

[33] S. Segura, A. B. Sánchez, and A. Ruiz-Cortés, "Automated variability analysis and testing of an E-commerce site.: an experience report," in *Proc. of ASE '14: Proceedings of the 29th ACM/IEEE International Conference on Automated Software Engineering*, pp.139-150, Association for Computing Machinery, Inc, 2014. Article (CrossRef Link)

[34] H. Andersson, E. Herzog, and J. Ölvander, "Experience from model and software reuse in aircraft simulator product line engineering," *Inf Softw Technol.*, vol.55, no.3, pp.595-606, Mar. 2013. Article (CrossRef Link)

[35] C. D. N. Damasceno, M. R. Mousavi, and A. da S. Simao, "Learning by sampling: learning behavioral family models from software product lines," *Empir Software Eng.*, vol.26, no.1, Jan. 2021. Article (CrossRef Link)

[36] L. Mariani, M. Pezzè, and D. Zuddas, "Recent Advances in Automatic Black-Box Testing," *Advances in Computers*, vol.99, pp.157-193, 2015. Article (CrossRef Link)

[37] P. Runeson and E. Engström, "Regression Testing in Software Product Line Engineering," *Advances in Computers*, vol.86, pp.223-263, 2012. Article (CrossRef Link)

[38] S. Lity, M. Nieke, T. Thüm, and I. Schaefer, "Retest test selection for product-line regression testing of variants and versions of variants," *Journal of Systems and Software*, vol.147, pp.46-63, Jan. 2019. Article (CrossRef Link)

[39] R. Lindohf, J. Krüger, E. Herzog, and T. Berger, "Software product-line evaluation in the large," *Empir Software Eng.*, vol.26, no.2, Mar. 2021. Article (CrossRef Link)

[40] M. Lochau, J. Bürdek, S. Hölzle, and A. Schürr, "Specification and automated validation of staged reconfiguration processes for dynamic software product lines," *Softw Syst Model.*, vol.16, no.1, pp.125-152, Feb. 2017. Article (CrossRef Link)

[41] X. Devroey et al., "Statistical prioritization for software product line testing: an experience report," *Softw Syst Model.*, vol.16, no.1, pp.153-171, Feb. 2017. Article (CrossRef Link)
[42] R. Lachmann, S. Lity, M. Al-Hajjaji, F. Fürchtegott, and I. Schaefer, "Fine-grained test case prioritization for integration testing of delta-oriented software product lines," in *Proc. of FOSD 2016: Proceedings of the 7th International Workshop on Feature-Oriented Software Development*, pp.1-10, Association for Computing Machinery, Inc, Oct. 2016. Article (CrossRef Link)
[43] R. A. B. Sulaiman, "Cost-effective model-based test case generation and prioritization for software product line," PhD thesis, Universiti Teknologi Malaysia, Faculty of Engineering - School of Computing, 2020. Article (CrossRef Link)
[44] F. Di Silvestro, "Improving testing reusability and automation for software product lines," M.S. thesis, Dept. Innovation Design and Eng., Vasteras, Sweden, 2020. Article (CrossRef Link)
[45] H. Samih, H. Le Guen, R. Bogusch, M. Acher, and B. Baudry, "An Approach to Derive Usage Models Variants for Model-Based Testing," in *Proc. of 26th IFIP WG 6.1 International Conference, ICTSS 2014, Testing Software and Systems*, LNPSE, vol.8763, pp.80-96, 2014. Article (CrossRef Link)
[46] M. Felderer, W. Hasselbring, H. Koziolek, F. Matthes, L. Prechelt, R. Reussner, B. Rumpe, and I. Schaefer Eds, Ernst Denert Award for Software Engineering 2020, Springer Cham, 2020. Article (CrossRef Link)
[47] S. A. Halim, D. N. A. Jawawi, and M. Sahak, "Similarity Distance Measure and Prioritization Algorithm for Test Case Prioritization in Software Product Line Testing," *Journal of Information and Communication Technology*, vol.18, no.1, pp.57-75, 2019. Article (CrossRef Link)
[48] T. Lavoie, M. Mérineau, E. Merlo, and P. Potvin, "A case study of TTCN-3 test scripts clone analysis in an industrial telecommunication setting," *Inf Softw Technol.*, vol.87, pp.32-45, Jul. 2017. Article (CrossRef Link)
[49] S. A. Razak, M. A. Isa, and D. N. A. Jawawi, "A Comparison on Similarity Distances and Prioritization Techniques for Early Fault Detection Rate," *Journal of Telecommunication, Electronic and Computer Engineering*, vol.9, no.3-3, pp.89-94, 2017. Article (CrossRef Link)
[50] R. Huang et al., "Abstract Test Case Prioritization Using Repeated Small-Strength Level-Combination Coverage," *IEEE Transactions on Reliability*, vol.69, no.1, pp.349-372, Mar. 2020. Article (CrossRef Link)
[51] M. Sahak, D. N. A. Jawawi, and S. A. Halim, "An Experiment of Different Similarity Measures on Test Case Prioritization for Software Product Lines," *Journal of Telecommunication, Electronic and Computer Engineering*, vol.9, no.3-4, pp.177-185, 2017. Article (CrossRef Link)
[52] M. Al-Hajjaji, T. Thüm, M. Lochau, J. Meinicke, and G. Saake, "Effective product-line testing using similarity-based product prioritization," *Softw Syst Model.*, vol.18, no.1, pp.499-521, Feb. 2019. Article (CrossRef Link)
[53] N. Gokilavani and B. Bharathi, "Multi-Objective based test case selection and prioritization for distributed cloud environment," *Microprocessors and Microsystems*, vol.82, Apr. 2021. Article (CrossRef Link)
[54] C. D. N. Damasceno, P. C. Masiero, and A. Simao, "Similarity testing for role-based access control systems," *Journal of Software Engineering Research and Development*, vol.6, no.1, Jan. 2018. Article (CrossRef Link)
[55] D. Flemström, P. Potena, D. Sundmark, W. Afzal, and M. Bohlin, "Similarity-based prioritization of test case automation," *Software Quality Journal*, vol.26, no.4, pp.1421-1449, Dec. 2018. Article (CrossRef Link)
[56] S. Khoshmanesh and R. Lutz, "Does Link Prediction Help Detect Feature Interactions in Software Product Lines (SPLs)?," *arXiv:2009.07392*, 2020. Article (CrossRef Link)
[57] S. Segura, R. M. Hierons, D. Benavides, and A. Ruiz-Cortés, "Automated metamorphic testing on the analyses of feature models," *Inf Softw Technol.*, vol.53, no.3, pp.245-258, Mar. 2011. Article (CrossRef Link)

[58] M. Aggarwal and S. Sabharwal, "Combinatorial Test Set Prioritization Using Data Flow Techniques," *Arab J Sci Eng.*, vol.43, no.2, pp.483-497, Feb. 2018. Article (CrossRef Link)

[59] A. B. Sánchez and S. Segura, "SmarTest: A Test Case Prioritization Tool for Drupal," in *Proc. of SPLC '17: Proceedings of the 21st International Systems and Software Product Line Conference*, vol.B, pp.9-12, Association for Computing Machinery, Sep. 2017. Article (CrossRef Link)

[60] A. B. Sánchez, S. Segura, J. A. Parejo, and A. Ruiz-Cortés, "Variability testing in the wild: the Drupal case study," *Softw Syst Model.*, vol.16, no.1, pp.173-194, Feb. 2017. Article (CrossRef Link)

[61] M. Sahak, S. Abd Halim, D. N. A. Jawawi, and M. A. Isa, "Evaluation of Software Product Line Test Case Prioritization Technique," *International Journal on Advanced Science Engineering Information Technology*, vol.7, no.4-2, pp.1601-1608, 2017. Article (CrossRef Link)

[62] A. B. Sanchez, S. Segura, and A. Ruiz-Cortes, "A Comparison of Test Case Prioritization Criteria for Software Product Lines," in *Proc. of 2014 IEEE Seventh International Conference on Software Testing, Verification and Validation*, pp.41-50, 2014. Article (CrossRef Link)

[63] H. Hemmati, "Advances in Techniques for Test Prioritization," *Advances in Computers*, vol.112, pp.185-221, 2019. Article (CrossRef Link)

[64] A. Arrieta, G. Sagardui, L. Etxeberria, and J. Zander, "Automatic generation of test system instances for configurable cyber-physical systems," *Software Quality Journal*, vol.25, no.3, pp.1041-1083, Sep. 2017. Article (CrossRef Link)

[65] U. Markiegi, A. Arrieta, L. Etxeberria, and G. Sagardui, "Dynamic test prioritization of product lines: An application on configurable simulation models," *Software Quality Journal*, vol.29, no.4, pp.943-988, Dec. 2021. Article (CrossRef Link)

[66] C. Birchler, S. Khatiri, B. Bosshard, A. Gambi, and S. Panichella, "Machine learning-based test selection for simulation-based testing of self-driving cars software," *Empirical Software Engineering*, vol.28, Apr. 2023. Article (CrossRef Link)

[67] I. Reinhartz-Berger and M. Kemelman, "Extracting core requirements for software product lines," *Requirements Engineering*, vol.25, no.1, pp.47-65, Mar. 2020. Article (CrossRef Link)

[68] R. Lachmann, S. Beddig, S. Lity, S. Schulze, and I. Schaefer, "Risk-based integration testing of software product lines," in *Proc. of VaMoS '17: Proceedings of the 11th International Workshop on Variability Modelling of Software-Intensive Systems*, pp.52-59, 2017. Article (CrossRef Link)

[69] H. Jahan, Z. Feng, and S. M. H. Mahmud, "Risk-Based Test Case Prioritization by Correlating System Methods and Their Associated Risks," *Arabian Journal for Science and Engineering*, vol.45, no.8, pp.6125-6138, Aug. 2020. Article (CrossRef Link)

[70] R. Huang, Q. Zhang, T. Y. Chen, J. Hamlyn-Harris, D. Towey, and J. Chen, "An Empirical Comparison of Fixed-Strength and Mixed-Strength for Interaction Coverage Based Prioritization," *IEEE Access*, vol.6, pp.68350-68372, 2018. Article (CrossRef Link)

[71] R. Huang et al., "On the Selection of Strength for Fixed-Strength Interaction Coverage Based Prioritization," in *Proc. of 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, pp.310-315, 2018. Article (CrossRef Link)

[72] H. Akimoto, Y. Isogami, T. Kitamura, N. Noda, and T. Kishi, "A Prioritization Method for SPL Pairwise Testing Based on User Profiles," in *Proc. of 2019 26th Asia-Pacific Software Engineering Conference (APSEC)*, pp.118-125, Dec. 2019. Article (CrossRef Link)

[73] A. Ensan, E. Bagheri, M. Asadi, D. Gasevic, and Y. Biletskiy, "Goal-Oriented Test Case Selection and Prioritization for Product Line Feature Models," in *Proc. of 2011 8th International Conference on Information Technology: New Generations, ITNG 2011*, pp.291-298, 2011. Article (CrossRef Link)

[74] S. Kumar, Rajkumar, and M. Rani, "Collaborative Filtering-based Test Case Prioritization and Reduction for Software Product-Line Testing," in *Proc. of IEEE Region 10 International Conference (TENCON)*, pp.498-503, Oct. 2019. Article (CrossRef Link)

[75] A. K. Thurimella and B. Brügge, "A mixed-method approach for the empirical evaluation of the issue-based variability modeling," *Journal of Systems and Software*, vol.86, no.7, pp.1831-1849, 2013. Article (CrossRef Link)
[76] A. Arrieta, S. Segura, U. Markiegi, G. Sagardui, and L. Etxeberria, "Spectrum-based fault localization in software product lines," *Inf Softw Technol.*, vol.100, pp. 18-31, Aug. 2018. Article (CrossRef Link)
[77] F. C. Ferrari, A. Rashid, and J. C. Maldonado, "Towards the practical mutation testing of AspectJ programs," *Science of Computer Programming*, vol.78, no.9, pp.1639-1662, Sep. 2013. Article (CrossRef Link)
[78] I. Hajri, A. Goknil, F. Pastore, and L. C. Briand, "Automating system test case classification and prioritization for use case-driven testing in product lines," *Empirical Software Engineering*, vol.25, no.5, pp.3711-3769, Sep. 2020. Article (CrossRef Link)
[79] L. Teixeira, P. Borba, and R. Gheyi, "Safe composition of configuration knowledge-based software product lines," *Journal of Systems and Software*, vol.86, no.4, pp.1038-1053, Apr. 2013. Article (CrossRef Link)
[80] Z. Chen et al., "Exploring Better Black-Box Test Case Prioritization via Log Analysis," *ACM Transactions on Software Engineering and Methodology*, vol.32, no.3, pp.1-32, Apr. 2023. Article (CrossRef Link)
[81] T. Landsberg, C. Dietrich, and D. Lohmann, "TASTING: Reuse Test-case Execution by Global AST Hashing," in *Proc. of the 17th International Conference on Software Technologies (ICSOFT 2022)*, vol.1, pp.33-45, 2022. Article (CrossRef Link)
[82] S. Ali et al., "Towards Pattern-Based Change Verification Framework for Cloud-Enabled Healthcare Component-Based," *IEEE Access*, vol.8, pp.148007-148020, 2020. Article (CrossRef Link)
[83] H.-F. Guo and Z. Qiu, "Automatic Grammar-Based Test Generation," in *Proc. of 25th IFIP WG 6.1 International Conference, ICTSS 2013, Testing Software and Systems*, LNPSE, vol.8254, pp.17-32, 2013. Article (CrossRef Link)
[84] G. Goos et al., Fundamental Approaches to Software Engineering, 2022. Article (CrossRef Link)

SITI HAWA MOHAMED SHAREEF currently studying in the field of master's degree (Information Technology) at Universiti Tun Hussein Onn, Malaysia. She holds a B.S in Computer Science (Web Technology). She was an IT Engineer at OK Blockchain Centre Sdn. Bhd. His current research interest on software engineering and software testing.

R. ADUNI SULAIMAN received PhD degree in computer science from Universiti Teknologi Malaysia, Johor, Malaysia. She was a system engineer at Infineon Technologies Malaysia before she joined the Department of Software Engineering at Universiti Tun Hussein Onn. Currently, her research interest on software engineering, software testing and soft computing.

ABD SAMAD HASAN BASARI is a Professor at Software Engineering Department, Universiti Tun Hussein Onn, Malaysia. He holds a B.S. in Mathematics from Universiti Kebangsaan Malaysia, an M.S. in Universiti Teknologi Malaysia, and PhD in Information and Communication Technology from Universiti Teknikal Malaysia Melaka, Malaysia. His current research interests are in Artificial Intelligence (AI) and Soft Computing.