JOIV : Int. J. Inform. Visualization, 7(3-2): Empowering the Future: The Role of Information Technology in Building Resilience - November 2023 2126-2134

INTERNATIONAL JOURNAL
ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

Test Case Prioritization for Software Product Line: A Systematic Mapping Study

Muhammad Idham [a], Shahliza Abd Halim [a,*], Dayang Norhayati Abang Jawawi [a], Zalmiyah Zakaria [a], Aldo Erianda [b], Nachnoer Arss [c]

Faculty of Computing, Universiti Teknologi Malaysia, 81310 Johor Bahru, Johor, Malaysia
Department of Information Technology, Politeknik Negeri Padang, Padang, West Sumatera, Indonesia
Faculty of Engineering, Universitas Negeri Padang, Padang, West Sumatera, Indonesia
*Corresponding author: *shahliza@utm.my*

***Abstract*—Combinatorial explosion remains a common issue in testing. Due to the vast number of product variants, the number of test cases required for comprehensive coverage has significantly increased. One of the techniques to efficiently tackle this problem is prioritizing the test suites using a regression testing method. However, there is a lack of comprehensive reviews focusing on test case prioritization in SPLs. To address this research gap, this paper proposed a systematic mapping study to observe the extent of test case prioritization usage in Software Product Line Testing. The study aims to classify various aspects of SPL-TCP (Software Product Line – Test Case Prioritization), including methods, criteria, measurements, constraints, empirical studies, and domains. Over the last ten years, a thorough investigation uncovered twenty-four primary studies, consisting of 12 journal articles and 12 conference papers, all related to Test Case Prioritization for SPLs. This systematic mapping study presents a comprehensive classification of the different approaches to test case prioritization for Software Product Lines. This classification can be valuable in identifying the most suitable strategies to address specific challenges and serves as a guide for future research works. In conclusion, this mapping study systematically classifies different approaches to test case prioritization in Software Product Lines. The results of this study can serve as a valuable resource for addressing challenges in SPL testing and provide insights for future research.**

***Keywords*— Software testing; software product line; system product line; test case prioritization; systematic mapping study.**

Manuscript received 23 Oct. 2022; revised 29 Apr. 2023; accepted 12 Sep. 2023. Date of publication 30 Nov. 2023.

I. INTRODUCTION

Testing is vital in the quality assurance process for Software Product Line Engineering (SPL). Two issues should be handled with testing strategies in SPL: 1) consider testing generation as a systematic selection of products to test, and 2) perform testing on final product functionalities. The first strategy specifies how products are selected from the vast set of possible products to test. This helps to reduce the set of possibilities to a reasonable but representative set of product configurations. The second strategy focuses on how each selected product is tested. This second strategy works in two SPL levels: domain engineering and application engineering, where variable test assets, i.e., test cases and test scenarios, are taken as input to the variability defined for the SPL [1].

Combinatorial explosion is still a commonplace issue when testing an SPL. Since there are many product variations, the number of test cases for each product increases dramatically. One of the techniques in SPL testing (SPLT) used is test case prioritization. Test Case Prioritization (TCP), used to minimize test suites based on regression testing approaches, originated from a Yoo and Harman survey in 2012 [2] that separates it into three classifications: selecting, minimizing, and prioritizing. Numerous variations of this regression testing approach have been used for SPL testing at the domain and application engineering levels. The goal of the SPL prioritization technique is to detect a large number of defects with less effort and adequate test coverage.

This paper aims to systematically map a study to observe how far test case prioritization has been used in Software Product Line Testing. Section 2 will report the related work that other researchers have done in SPLT, especially TCP for SPL. Section 3 describes our research method for the mapping study. Section 4 will discuss and show the results of our work. Section 5 explains threats to the validity of our work—the conclusion of our work in Section 6.

II. MATERIALS AND METHOD

A. Systematic Mapping Study

The literature review is crucial in comprehensively examining and analyzing existing literature on a particular topic. It helps researchers establish the current state of knowledge, identify gaps, and develop research questions or hypotheses. In the context of literature review, two commonly employed approaches are systematic literature review and systematic mapping studies.

Systematic literature reviews aim to comprehensively synthesize existing research on a specific topic, focusing on the analysis and interpretation of primary studies. Numerous systematic literature reviews have been applied, such as the SLR profiling method described by Adwan and Alsaeed [3], the adoption of the PRISMA framework as highlighted by Malatji et al. [4], and the systematic review proposed by Lee et al. [5].

On the other hand, systematic mapping studies have a broader scope, aiming to map and categorize the existing literature rather than providing a detailed analysis of individual studies [6]. We proposed a systematic mapping study that identifies research's extent, nature, and distribution within a specific field. We classify studies based on predefined criteria and extract relevant information.

B. Software Product Line Testing (SPLT)

There have been existing SPLT literature reviews in general. Lee *et al.* [5] conducted a systematic literature review to identify test coverage criteria that have been adopted in SPLT. Lee analyzes the concept of test bases (groups of test basis [5]) and test coverage criterion for SPLT compared to single system testing, then attempts to identify and scrutinize the variability in SPLT. The test coverage criteria, for example, feature combination, feature interaction, model-based coverage, and code-based coverage, were discovered due to limitations in the test coverage criteria for test classification in SPLT. Machado *et al.* [1] in 2014 classified SPLT issues into two categories. The first is to focus on generating and selecting the product to be tested, and the second is to achieve testing on the final product's functionalities. Machado *et al.* help determine which scope of interest to optimize and select a test approach in SPLT with different case studies. Neto *et al.* [7] created a systematic mapping and classified the results based on research focus, research type, and contribution type. Sahid *et al.* [8] specifically used systematic mapping to review Combinatorial Interaction Testing in SPL using a systematic mapping study. Our study would apply classifications from [2] and [5] to identify the method, evaluation technique, and domain of test case prioritization used to solve testing issues in SPL.

In single systems, several works of literature explain test case prioritization. Khatibsyarbini *et al.* [9] examine and classify the current test case prioritization approaches. Lou *et al.* [10] conducted a detailed survey to investigate TCP from six technical aspects systematically. However, no specific mention of test case prioritization applied in SPL testing exists. This research will focus on test case prioritization techniques in SPL testing.

C. Test Case Prioritization for SPL

A survey explicitly done on SPL-TCP has been done by Sahak *et al.* [11], Sanchez *et al.* [12], and Kumar *et al.* [13]. Sahak *et al.* [11] evaluate the test case prioritization techniques by distributing and comparing the approach, trends, differences, and suggested frameworks based on the research article. Sanchez *et al.* [12] explore the TCP technique by comparing proposed prioritization criteria with established common metrics of the feature model. Kumar *et al.* [13] introduced a survey of the TCP technique for SPL in 2016. However, they do not specifically explain the differences in the method, criterion, or metrics that were used in the most recent study. In summary, no study systematically reviews test case prioritization for SPL. Thus, this research will propose to classify test case prioritization in the software product line using a systematic mapping study.

D. PICOC Criteria

Our study implements the instructions from Kitchenham and Chartes [14] and the bubble plot diagram by Petersen et al. [15] to visualize some research results. The PICOC criteria are described in Table 1.

TABLE I
PICOC CRITERIA TO GENERATE A RESEARCH QUESTION

Population	Software Product Line Testing, System Product Line Testing, Software Product Family Testing, Configurable Systems
Intervention	Test Case Prioritization, SPL Testing issue, SPL Test Basis, Application Domain, SPL Testing strategy, SPLT criteria, SPLT method, test measurement, SPLT domain, SPLT case study, and tools that support for SPL-TCP
Comparison	n/a
Outcome	Variability of test prioritization approaches; selected domain, SPL testing method, and metrics for evaluation; key findings
Context	Specified in Software Product Line Engineering with a focus on using test prioritization approaches

E. Research Questions

To generate and define the research question, we established a framework based on the PICOC criteria (Population, Intervention, Comparison, Outcome, and Context) introduced by Petticrew and Roberts [16]. Research questions are generated based on the PICOC that has been created. The research questions are specified in terms of Test Case Prioritization.

- RQ1: What issues are the most discussed in SPL Test Case Prioritization studies?
- RQ2: What prioritization approach have researchers applied to prioritize test cases for SPL?
- RQ3: What measurements do researchers use to evaluate the test case prioritization techniques in SPL?
- RQ4: In what SPL case study domain does the researcher focus on SPL-TCP?
- RQ5: What tools have researchers used to prioritize test cases in SPL?

Each research question is based on a review of related literature and an evaluation by several researchers. The major goal behind RQ1 is to identify the most discussed issues in SPL testing that the TCP strategy must handle. This RQ is related to Machado *et al.* literature review [1] that separates SPL testing into two issues. RQ2 concentrates on the prioritization approach used in recent primary studies. RQ3 identified the measurement that has been used to evaluate the test case prioritization approach in recent primary studies. In RQ4, we collect the dataset or case study domain used in every primary study. RQ5 identified the tools for evaluating the TCP approach in recent SPL-TCP studies. This research question will generate our mapping study results in Section 6.

F. Search Strategy and Study Selection

A search strategy was accomplished to get the existing primary study. A search string was provoked to look at the key area in the online data library. We use the reference database below:

- SCOPUS
- ScienceDirect
- SpringerLink
- IEEE Xplore
- ACM Digital Library
- Web of Science

The search string query depends on what databases want to discover. The search string query has been generated for each reference database in Table 2. Since several search strings focusing on the SPL-TCP key area were not found in the selected online data library, we first searched the "Software Product Line Testing" key area, then eliminated it based on the title and keywords to classify the research about TCP for SPL.

TABLE II
SEARCH STRINGS AND FILTERING FOR STUDIES

Engine	Search string
SCOPUS	TITLE (test AND prior*) AND TITLE-ABS-KEY ("product line*" OR "product famil*" OR configurable)
ScienceDirect	***Title, abstract, keywords***: test AND (prioritization OR prioritisation) AND ("product line" OR "product lines" OR "product family" OR "product families" OR configurable)
SpringerLink	test AND prior* AND ("product line*" OR "product famil*" OR "configurable")(where title contains "prior*")
IEEE Xplore	("All Metadata":test*) AND ("Document Title":prior*) AND ("All Metadata":product line* OR "All Metadata":product famil* OR "All Metadata":configurable OR "All Metadata":SPL)
ACM Digital Library	Title: test] AND [Title: prior*] AND [[All: "software product line*"] OR [All: "product line*"] OR [All: "product famil*"] OR [All: "spl"] OR [All: configurable]]
Web of Science	TI=(test prior*) AND ALL=("product line*" OR "product famil*" OR configurable OR SPL)

The result of the primary study was based on the search string and then selected by inclusive and exclusive criteria. To determine the primary study and to restrict the primary study to PICOC criteria, inclusive and exclusive criteria are defined in Table 3.

TABLE III
INCLUSION AND EXCLUSION CRITERIA

Inclusion criteria	Year 2012 until 2021 test case prioritization for software product line Studies related to computer science OR software engineering Studies include both journal and conference papers. Only the latest studies if there are duplicate publications
Exclusion criteria	Studies not related to SPL Testing topics. Study discussing TCP-SPL method, approach in a context other than TCP-SPL Studies that related to literature review Study text not in the English language Studies that are not accessible

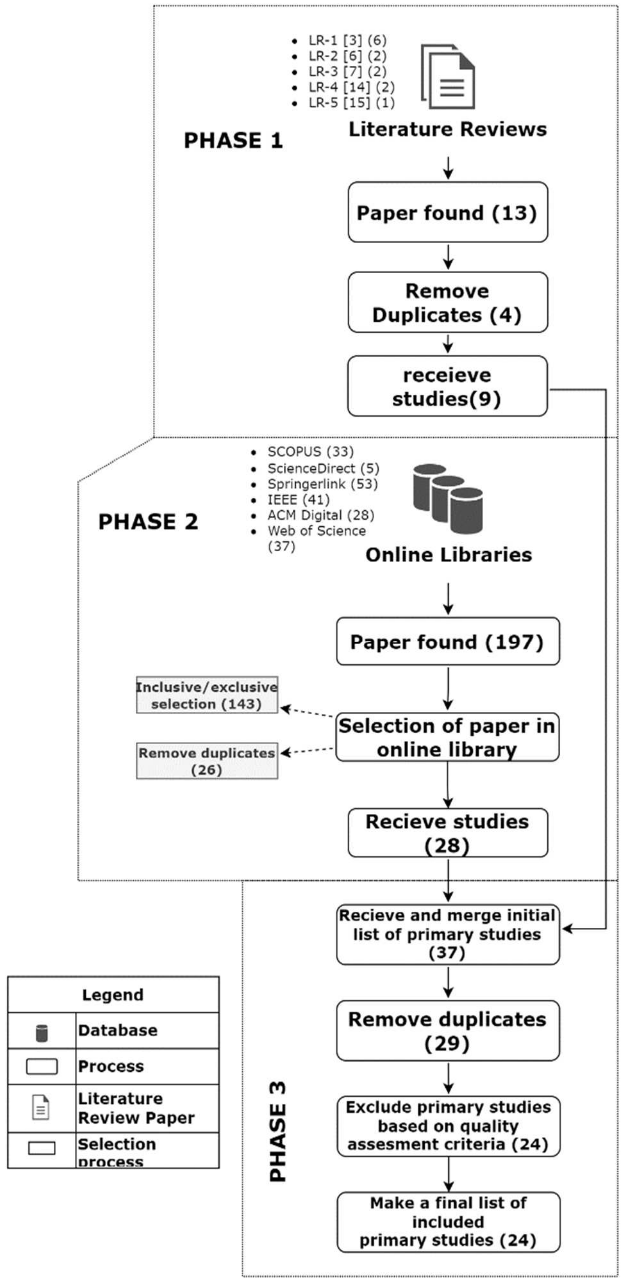

Fig. 1 Step of study selection for Systematic Mapping Study

Study selection in this research has three phases. Since this is a mapping study, the planning stage only focuses on mapping the data based on our findings from the primary study. Fig.1 describes each part step by step. In the first phase, we analyzed the data based on the latest literature review. Several literature reviews have been conducted [5], [9], [10], [17], [18]. At least six potential studies have been accepted. The studies were selected using both inclusive and exclusive criteria. Most of the literature reviews cited identical research studies. The second phase, search strategy, and study selection method are used to search articles in the online data library and collect the most recent publications. We scrutinized studies from 2012 to 2021 to extract more information focused on SPL and TCP. Lastly, the third phase was used to gather all studies based on phases 1 and 2, then remove duplicate studies. The quality assessment generated in these phases will produce better results. During these phases, the requirements for reading the abstract and full text are fulfilled, and the final selected studies, called primary studies, consisting of journal articles and conference articles, are generated.

G. Quality Assessment Criteria

Quality assessment criteria (QAC) from Kitchenham *et al.* [14] help to produce more detailed inclusive and exclusive criteria based on selected primary studies and asserting individual primary studies when the results are being selected (see Fig. 1). QAC aims to guide the development of primary studies based on quality instruments called checklists. Checklists were formed from the questions created. This allowed us to generate points for individual primary studies to confirm whether a given study meets our criteria. When the questions are answered, a point is added with three different values: 1 for satisfied, 0,5 for partially satisfied, and 0 for not satisfied. The questions and the points of feedback are described in Table 4.

TABLE IV
QAC CHECKLIST

ID	QAC	Feedback
QC1	Does the study focus on the software product line testing domain?	Yes=1; Partially=0,5; No=0
QC2	Does the primary study explicitly focus on of test case prioritization approach for SPL?	Yes=1; Partially=0,5; No=0
QC3	Does the primary study not relate to survey, literature review, or comparison study?	Yes=1; Partially=0,5; No=0

H. Data Extraction and Synthesis

After generating the criteria, the collection of primary studies was extracted to collect and classify the detailed information. The data extracted in the reference manager tool from selected studies consisted of these attributes: title, authors, year of publications, name of journal or conference, publisher, keywords, abstract, and notes. This extracted data was gathered in order to provide answers to all research questions. The extracted properties are described in Table 5.

Data synthesis aims to combine the information from the selected studies to answer the research questions collectively. The data synthesized from this study includes both quantitative and qualitative data. Quality assessment criteria have been used to synthesize the objectives and refine the primary studies. As a result of the synthesizing process, at least three primary studies were removed. The results of the synthesized data were a list of various SPL-TCP primary studies, as shown in Table 6.

TABLE V
DATA EXTRACTION PROPERTIES

Data Properties	Descriptions
Citation Data	Title, authors, year of publications, name of journal/conference, publisher, keywords, abstract, and notes.
SPL Testing Interest	Classification of issues in SPL Testing those researchers should handle
SPLT Phases	Domain Engineering or Application Engineering
SPL Testing Basis	Lee [5] separates test basis into 3: specification-based, design-based, and program-based, originating from ISO/IEC/IEEE 29119-1:2013
SPL Testing Strategy	categorization of SPL test strategies derived from [5] research study: selection product (SLPR), Commonality & Reuse Strategy (CMRS), and Sample Application Strategy (SAMP)
Prioritization Criterion	Variability of criteria to appraise test-case fault-detection capability in prioritization either in test selection or test of final product
Prioritization Algorithm	List of algorithms(method) to manage Test Case Prioritization in Software Product Line
Research Case Study	Several Case Studies have been used to evaluate the prioritization method in Software Product Line Testing.
Supporting Tools	Tools that used to support the experiment

TABLE VI
SEARCH RESULT AND REMOVE UNWANTED STUDY

Engine	Results		
	Raw	Incl/Excl	extracted
SCOPUS	33	15	9 journals, 6 conferences
ScienceDirect	5	4	4 journals
SpringerLink	53	2	2 journals
IEEE Xplore	41	7	1 journal, 6 conferences
ACM Digital Library	28	8	8 conferences
Web of Science	37	20	12 Journals, 8 conferences

I. Documents Retrieval

As explained before, the step-by-step process of study selection is described in Fig. 1 in three phases. In Phase 1, we generate studies from several SPL testing literature reviews. We found at least nine primary study candidates to collect. In phase 2, we conducted the primary study search in the online library that was described in the search strategy and study selection from 2012 to 2021. Previously, in Table 5, the selected primary studies were filtered based on inclusive and exclusive criteria and extracted by reading the title, keywords, and abstracts. A total of 54 papers were collected after inclusion and exclusive criteria were applied. Several papers have some duplication in two or more databases. Therefore, we removed the duplicate studies and delivered 28 research papers related to TCP-SPL. The results are already shared in

Fig. 2, and the extracted results of each database are described in Table 6. Finally, in phase 3, after evaluating the primary study by QAC and reading the paper, we accepted 24 primary studies to be reviewed and synthesized into a systematic mapping study related to SPL testing and prioritizing the test cases.

III. Results and Discussion

A. RQ1 – Distribution of SPL Testing Issue on Test Case Prioritization Approach

Software Product Line Testing (SPLT) needs to identify a strategy to test the product lines. Machado *et al.* [2] classified two issues that SPLT strategies must employ. In this case, two issues that TCP in SPL should handle are (1) selecting products before testing and (2) testing the final product. We have distributed the results since 2012 to determine how far the TCP strategy has been used to prioritize the test cases based on both testing interests. The results displayed in Fig. 2 show that the first issue is the most discussed in recent literature, with a total of 14 primary studies, and the second issue has nine primary studies.

Meanwhile, one primary study focuses on both [P10]. The first issue had a significant number of studies in 2014 and 2017, with the highest number in 2017 (5 primary studies), followed by 2014 with four studies, and then in 2019, there were two primary studies. In 2016, 2020, and 2021, only one primary study will focus on the first issue. Meanwhile, in 2015, no researcher focused on the first issue. The distribution of the second issue in recent studies has been limited to no more than two primary studies every year, with two studies in 2014, 2020, and 2021 and one primary study in 2015, 2016, and 2019. However, no primary studies were discovered in 2012, 2013, or 2018. Table 7 shows the references for primary studies.

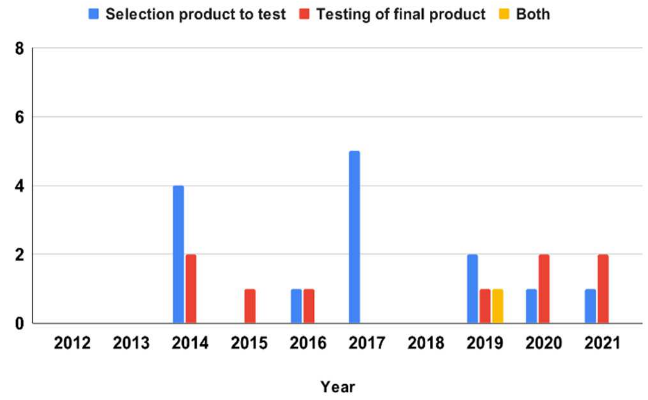

Fig. 2 Distribution of SPL Testing Interests

TABLE VII
Reference of Primary Study in RQ1

Issue in SPL	Primary Study
Selection of Product to test	P3, P5, P7, P8, P11-P13, P14, P15, P16, P21-P24
Testing of end-product	P1, P2, P4, P6, P9, P17, P18, P19, P20
Both	P10

The first issue focuses on strategies using Model-based statistical prioritization [P12, P23], feature prioritization for integration testing based on the feature model to reduce and prioritize test cases called PINE (Prioritized Integration testiNg in software product linE based on FM) [P15], combinatorial interactions testing (CIT) with similarity-based prioritization algorithm [P8, P14, P23], randomized and search-based with similarity-based TCP [P22], combination CIT with search-based metaheuristic approach to solve prioritized pairwise problem [P16, P21], prioritize product using collaborative filtering-based combined with K-Nearest Neighbor algorithm (KNN) [P7], and cluster-based prioritization with similarity [P14]. On the other hand, [P10] used a delta-oriented test case prioritization approach with string distance for the reorder feature model. In several studies, the definition of a product to test addressed "prioritizing test cases" instead of "prioritizing product configuration before testing." However, this does not show any differences since a product configuration is also defined as a test case. Prioritizing a product configuration is a popular option to reduce costs and increase effectiveness before testing a software product line.

The second issue focuses on prioritizing test cases from the final product using a delta-oriented approach for integration testing [P17, P18], search-based prioritization [P9, P19, P20] to solve a multi-objective problem, automated test prioritization for use-case-driven testing [P6], dynamic test prioritization with similarity-based [P2], and model-based with dissimilarity [P1]. In [P10], the delta-oriented TCP approach with weighted and string-distance criteria is also used to evaluate code-based testing. Although not many studies focus on this issue, this leaves largely unexplored research in terms of SPL-TCP.

B. RQ2 – SPL Test Case Prioritization Approaches

RQ2 results focus on the methods and prioritization criteria used in the recent primary study from 2012 until 2021. Several criterion and prioritization algorithms have been used for TCP approaches. Fig. 3 shows that similarity-based using the string distance criterion is the highest prioritization approach in TCP-SPL. The second highest is the Search-based prioritization approach with a weight-based criterion, while the third is Delta-oriented prioritization with a variability criterion using similarity and weight-based.

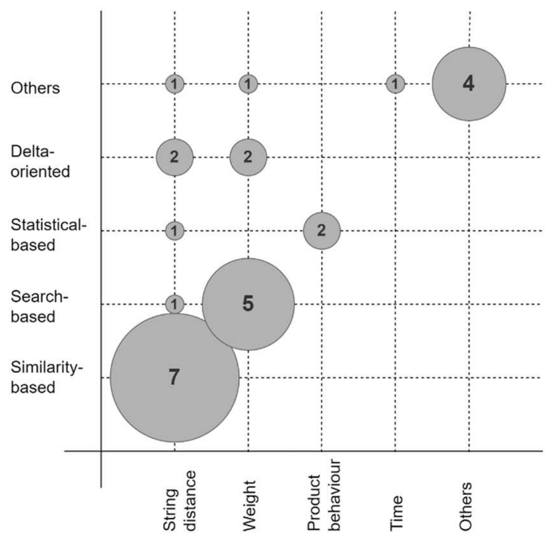

Fig. 3 Result of SPL-TCP criterion and method

At least seven studies used a similarity-based prioritization approach with a string distance criterion to prioritize test cases or product-before-test in SPL. Several methods have been used for SPL test case prioritization, like All-yes Config [P10, P13, P24] and Local Minimum Distance and Global Minimum Distance [P22]. On the other hand, [P14] combined a similarity-based approach with a cluster-based configuration. Moreover, some studies used the improved All-yes-Config method [P10]. Then, [P1] used the Last Minimal for Local Minimum Distance, and [P2] proposed a dynamic test prioritization approach by prioritizing product and test case with static and dynamic test prioritization algorithms. These approaches used string distance as a prioritization criterion. Second, at least five primary studies used a search-based TCP approach [P9, P16, P19–P21]. Different heuristic approaches have been proposed, such as genetic algorithm [P20, P21], local-search algorithm [P9, P20], Evolutionary Algorithm (EA) [P16, P20], and Greedy [P19]. These studies used a weight-based or fitness-function criterion as a prioritization criterion.

There are other prioritization methods, such as delta-oriented prioritization approach, statistical prioritization, and so on. Lachmann *et al.* used delta-oriented prioritization using weight-based criterion [P18] and similarity-based criterion [P17], while Hajjaji *et al.* [P13] combined delta-oriented with the All-yes config prioritization method. Devroey *et al.* [P12, P23] used a statistical prioritization approach based on product behavior (Featured Transition Systems). Meanwhile, Akbari *et al.* [P15] proposed PINE (Prioritized Integration Testing in Software Product Line), Kumar *et al.* combined collaborative filtering based on KNN features [P7], and K-means with Principal Component Analysis [P5]. Kumar et al. [P3] also proposed cost-effective product prioritization, which reorders product lines based on product cost. Hajri *et al.* [P4] introduced automated test prioritization in use-case-driven testing.

According to recent studies, the most used criterion is the string distance criterion. String distance criterion is used in the similarity-based prioritization approach and in delta-oriented prioritization [P13, P17] and collaborative filtering with KNN [P7]. There are several proposed string distance criteria, i.e., Hamming distance, Jaccard distance, and Djaro-Winkler. Other string distance criteria can be used, such as Euclidean and dice [19]. The second most used is a weight-based criterion such as CIT prioritization weight (measured value, rank-based value, and random value), fitness function, and coverage component, while others used product behavior [P10, P19] and domain engineer knowledge [P13]. Table 8 enumerates the approaches from the recent study.

TABLE VIII
OVERVIEW OF SPL TEST PRIORITIZATION APPROACHES IN SELECTED STUDY

Prioritization Method	**Criterion**	**Studies**
Similarity-based	String distance	P1, P2, P8, P10, P14, P22, P24
Search-based	Weight-based	P9, P16, P19, P20, P21
Delta-oriented	String distance	P13, P17
	Weight-based	P17, P18
Statistical based	Product behavior	P12, P23
Others		P3–P7, P11, P15

C. *RQ3 – Measurement to Evaluate Test Case Prioritization Result*

In this section, we collect all measurements that every primary study used to measure and prove the experiment results. Fault detection rate is the most used to evaluate Test Case Prioritization experiments, with APDF (Average Percentage of Fault Detection) as a common metric for measurement [P1-P3, P5, P6, P9-P11, P15, P16, P18, P24]. On the other hand, several studies introduced their fault detection measurement such as Rate of Fault Detection (RFD) [P21], Average Percentage Change Coverage (APCC) [P19, P20], Test Efficiency Rate (TER) [P17], Normalized APFD (NAPFD) and Rank of Failing Test Cases (RFTC) [P5], Meanwhile, Henard *et al.* [20] use the 2-wise and t-wise result to measure their evaluation. Several primary studies generated their measurement and execution time to measure the results.

Then, several studies used time parametric to evaluate Test Case Prioritization for SPL experiments. [P2, P10, P22-P25] consider execution time to achieve time reduction to find defects. Other time parametric have been used, such as Fault Detection Time (FDT) [P10], Simulation Time [P10], functional/non-functional covering time [P10], and Normalized Time Reduction (NTR) [P5]. Meanwhile, others such as [P19 & P20] used Average Percentage Change Coverage (APCC), [P17] used Test Case Reduction Rate (TCRR), and [P5] used Rank of Failing Test Cases (RFTC).

D. *RQ4 - Application Domain to Prioritize Test Case*

In a recent primary study, there are numerous domains (case studies) to evaluate test case prioritization for SPL. An academic case study is the most used domain for SPLT [P1, P3, P5, P7, P8, P11, P114, P15, P19, P21, P22, P24]. The second most used is software products, i.e., Linux, FreeBSD, SQL, and Cisco Video Conference [P10, P12, P14, P20-P23]. Then, several studies used automotive case studies [P2, P7, P10, P15, P19, P20], such as automated cruise control, body comfort systems, and ABS systems. Following that, an industrial case study [P2, P4, P9, P10, P12, P16], such as an elevator, mine pump, DRUPAL, industrial tank, or SSH library product. Lastly, [P17] used a case study based on Learning Management Systems (LMS). All results are described in Fig. 4.

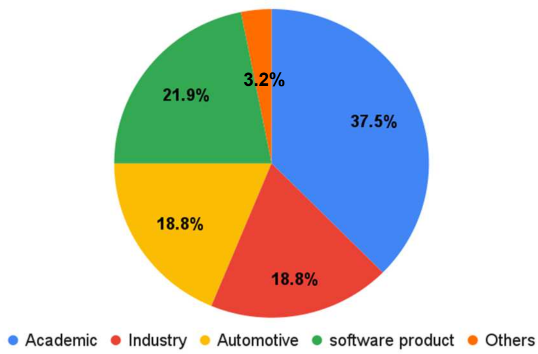

Fig. 4 Result of SPL Domain (case study)

E. *RQ5 – SPL Testing Tools*

In the primary study, many tools are used to help evaluate the TCP approach. More of them are unique. FeatureIDE is the dominant tool for all selected primary studies in SPL testing [P20, P8, P7]. For tools to generate feature models, [P18] and [P13] used SPLOT. [P1] and [P14] used an IDE such as NetBeans or Eclipse. [P15] use ADL DELTARX. [P17] used SPL Conqueror to generate feature model samplings.

F. *Internal Validity*

This mapping study systematically maps primary studies in test prioritization for Software Product Lines. This study is intolerant of the existing, unclear primary study. The search strategy is based on advanced search based on the title of the journal or conference paper. It means that there may be an incomplete selection. To complete our search process, our study precisely determined the search strategy based on PICOC criteria that have been determined before. Then, the search strategies were generated by determining the keywords, formatting the search strings, and implementing the search strategy based on guidance from online libraries. After that, the inclusion and exclusion criteria are implemented for further selection. This study adds more quality criteria to focus on the selected primary study. Primary studies that are survey, review, or comparison studies are excluded from this study to prevent biases between research papers and review papers.

G. *External Validity*

This study aims to classify test case prioritization techniques based on previous research on SPL testing in general. Since only one survey has been found to review test case prioritization in software product lines, we believed that our mapping study would systematically explain the variation of TCP techniques and what metrics to measure the technique.

IV. CONCLUSION

A systematic mapping study is a process to discover and classify some research areas. Regarding SPL test case prioritization, it aims to examine how far test case prioritization is used for software product line testing and answer recent research questions. The test case prioritization approach has recently been used to solve the combinatorial explosion problem by reordering the test cases. This research is systematically mapping the primary study from 2012 to 2021. The recent study primarily focused on prioritizing products before testing, like in CIT testing and model-based testing, rather than the second issue of prioritizing test cases from end-product testing. Although recent studies have mainly focused on the first issue to achieve effectiveness and reduce cost, the second issue is still an exciting option for further research. Similarity-based test prioritization is the most common approach that has been used in recent studies. It is mostly used to prioritize products using a variety of methods. A search-based TCP, a metaheuristic algorithm, is the second-most-used approach, followed by a weight-based fitness function criterion.

Most studies used APFD as a test measurement, with execution time as the second-most standard test measurement. Several studies propose their measurement, such as APCC for delta-oriented prioritization [P14, P15], cost and effective measures [P16], and FRCT/NFRCT [P7]. Various application domains (case studies) have been used in recent studies, for instance, automotive, industrial, academic, software products, and so on. To conclude, Fig. 6 summarizes the systematic mapping study as a taxonomy.

ACKNOWLEDGMENT

This work was supported/funded by the Ministry of Higher Education under the Fundamental Research Grant Scheme (FRGS/1/2020/ICT02/UTM/03/1).

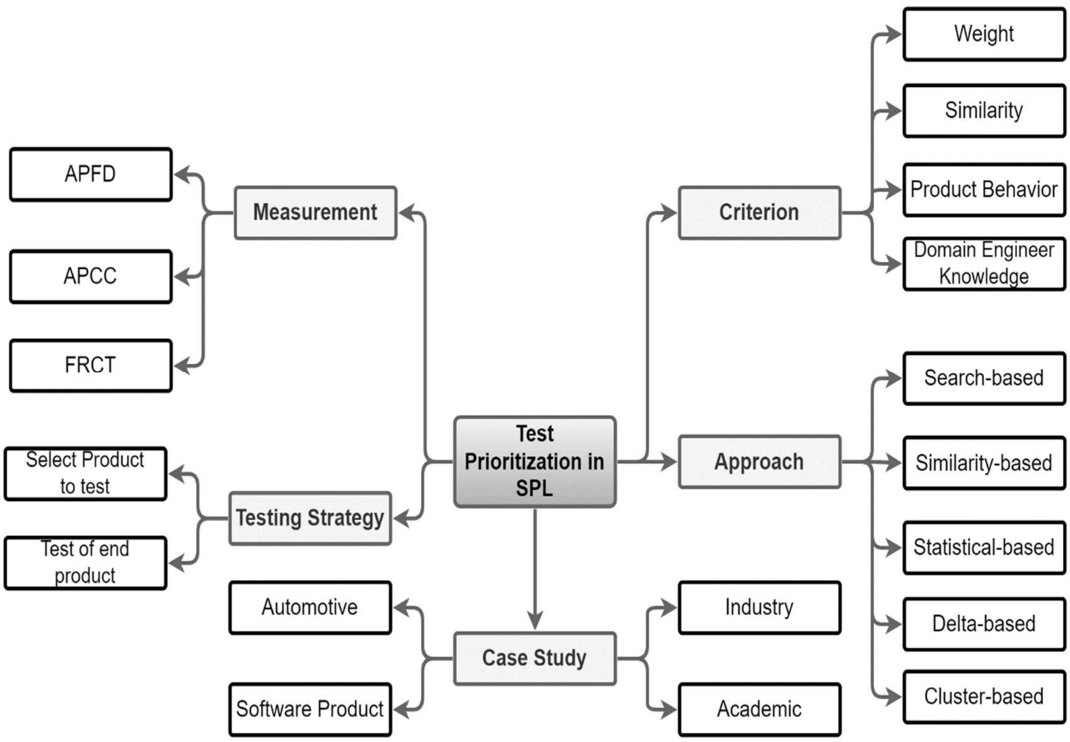

Fig. 6 Taxonomy of the SMS on Test Case Prioritization for Software Product Line

TABLE IX
LIST OF PRIMARY STUDIES

ID	REF	Title	Year	Author
P01	[21]	A dissimilarity with dice-jaro-winkler test case prioritization approach for model-based testing in software product line	2021	R. A. Sulaiman, D. N. A. Jawawi, and S. A. Halim
P02	[22]	Dynamic test prioritization of product lines: An application on configurable simulation models	2021	U. Markiegi, A. Arrieta, L. Etxeberria, and G. Sagardui
P03	[23]	Cost-effective product prioritisation technique for software product line testing	2021	S. Kumar, M. Mittal, and V. K. Yadav
P04	[24]	Learning-based prioritization of test cases in continuous integration of highly-configurable software,	2020	J. A. P. Lima, W. D. Mendonça, S. R. Vergilio, and W. K. Assunção,
P05	[25]	A hybrid approach to perform test case prioritisation and reduction for software product line testing,	2020	S. Kumar, R. Kumar, and M. Mittal
P06	[26]	Automating system test case classification and prioritization for use case-driven testing in product lines	2020	I. Hajri, A. Goknil, F. Pastore, and L. C. Briand
P07	[27]	Collaborative Filtering-based Test Case Prioritization and Reduction for Software Product-Line Testing	2019	S. Kumar, Rajkumar, and M. Rani
P08	[28]	Similarity distance measure and prioritization algorithm for test case prioritization in software product line testing	2019	Halim, S. A., Jawawi, D. N. A., & Sahak, M.
P09	[29]	Search-Based test case prioritization for simulation-Based testing of cyber-Physical system product lines	2019	A. Arrieta, S. Wang, G. Sagardui, and L. Etxeberria
P10	[30]	Effective product-line testing using similarity-based product prioritization	2019	M. Al-Hajjaji, T. Thüm, M. Lochau, J. Meinicke, and G. Saake
P11	[31]	A Prioritization Method for SPL Pairwise Testing based on User Profiles	2017	H. Akimoto, Y. Isogami, T. Kitamura, N. Noda, T. Kishi
P12	[32]	Statistical prioritization for software product line testing: an experience report	2017	X. Devroey *et al*
P13	[33]	Delta-Oriented Product Prioritization for Similarity-Based Product-Line Testing	2017	M. Al-Hajjaji, S. Lity, R. Lachmann, T. Thüm, I. Schaefer, and G. Saake
P14	[34]	Efficient product-line testing using cluster-based product prioritization	2017	M. Al-Hajjaji, J. Kruger, S. Schulze, T. Leich, and G. Saake
P15	[35]	A Method for Prioritizing Integration Testing in Software Product Lines Based on Feature Model	2017	Akbari, Z., Khoshnevis, S., & Mohsenzadeh, M.
P16	[36]	Multi-objective test case prioritization in highly configurable systems: A case study	2016	J. A. Parejo, A. B. Sánchez, S. Segura, A. Ruiz-Cortés, R. E. Lopez-Herrejon, and A. Egyed
P17	[37]	Fine-grained test case prioritization for integration testing of delta-oriented software product lines	2016	Lachmann, R., Lity, S., Al-Hajjaji, M., Fürchtegott, F., & Schaefer, I.
P18	[38]	Delta-oriented test case prioritization for integration testing of software product lines	2015	Lachmann, R., Lity, S., Lischke, S., Beddig, S., Schulze, S., & Schaefer, I.
P19	[39]	Test Suite Prioritization by Switching Cost	2014	H. Y. Wu, C. H. Nie, and F. C. Kuo
P20	[40]	Multi-objective test prioritization in software product line testing: An industrial case study	2014	Wang, S., Buchmann, D., Ali, S., Gotlieb, A., Pradhan, D., & Liaaen, M.
P21	[41]	A parallel evolutionary algorithm for prioritized pairwise testing of software product lines	2014	Lopez-Herrejon, R. E., Javier Ferrer, J., Chicano, F., Haslinger, E. N., Egyed, A., & Alba, E.
P22	[20]	Bypassing the combinatorial explosion: Using similarity to generate and prioritize t-wise test configurations for software product lines	2014	Henard, C., Papadakis, M., Perrouin, G., Klein, J., Heymans, P., & Le Traon, Y.
P23	[42]	Towards Statistical Prioritization for Software Product Lines Testing Categories and Subject Descriptors	2014	Devroey, X., Perrouin, G., Cordy, M., Schobbens, P. Y. et al.
P24	[43]	Similarity-based prioritization in software product-line testing	2014	Al-Hajjaji, M., Thüm, T., Meinicke, J., Lochau, M., & Saake, G.

REFERENCES

[1] I. D. C. Machado, J. D. McGregor, Y. C. Cavalcanti, and E. S. De Almeida, "On strategies for testing software product lines: A systematic literature review," *Inf Softw Technol*, vol. 56, no. 10, pp. 1183–1199, 2014, doi: 10.1016/j.infsof.2014.04.002.

[2] S. Yoo and M. Harman, "Regression testing minimization, selection and prioritization: a survey," *Software Testing, Verification and Reliability*, pp. 1–22, 2010, doi: 10.1002/stvr.

[3] E. J. Adwan and B. A. Alsaeed, "Cloud Computing Adoption in the Financial Banking Sector- A Systematic Literature Review (2011-2021)," *International Journal of Advanced Science Computing and Engineering*, vol. 4, no. 1, pp. 48–55, Apr. 2022, doi: 10.30630/ijasce.4.1.73.

[4] W. R. Malatji, R. VanEck, and T. Zuva, "A Systematic Review of the Adoption of eHealth Cloud-based Technology Applications during COVID-19," *International Journal of Advanced Science Computing and Engineering*, vol. 3, no. 2, pp. 53–64, Oct. 2021, doi: 10.30630/ijasce.3.2.44.

[5] J. Lee, S. Kang, and P. Jung, "Test coverage criteria for software product line testing: Systematic literature review," *Inf Softw Technol*, vol. 122, no. December 2019, p. 106272, 2020, doi: 10.1016/j.infsof.2020.106272.

[6] R. S. Wahono, "A Systematic Literature Review of Software Defect Prediction: Research Trends, Datasets, Methods and Frameworks," *Journal of Software Engineering*, vol. 1, no. 1, pp. 1–16, 2015.

[7] P. A. Da Mota Silveira Neto, I. Do Carmo MacHado, J. D. McGregor, E. S. De Almeida, and S. R. De Lemos Meira, "A systematic mapping study of software product lines testing," *Inf Softw Technol*, vol. 53, no. 5, pp. 407–423, 2011, doi: 10.1016/j.infsof.2010.12.003.

[8] M. Z. Sahid, A. Azim, A. Ghani, and S. Baharom, "Combinatorial Interaction Testing of Software Product Lines : A Mapping Study," 2016, doi: 10.3844/jcssp.2016.379.398.

[9] 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. Elsevier B.V., pp. 74–93, Jan. 01, 2018. doi: 10.1016/j.infsof.2017.08.014.

[10] Y. Lou, J. Chen, L. Zhang, and D. Hao, *A Survey on Regression Test-Case Prioritization*, 1st ed., vol. 113. Elsevier Inc., 2019. doi: 10.1016/bs.adcom.2018.10.001.

[11] M. Sahak, S. A. Halim, D. N. A. Jawawi, and M. A. Isa, "Evaluation of software product line test case prioritization technique," *Int J Adv Sci Eng Inf Technol*, vol. 7, no. 4–2 Special Issue, pp. 1601–1608, 2017, doi: 10.18517/ijaseit.7.4-2.3403.

[12] A. B. Sanchez, S. Segura, and A. Ruiz-Cortes, "A Comparison of test case prioritization criteria for software product lines," *Proceedings - IEEE 7th International Conference on Software Testing, Verification and Validation, ICST 2014*, pp. 41–50, 2014, doi: 10.1109/ICST.2014.15.

[13] S. Kumar and Rajkumar, "Test case prioritization techniques for software product line: A survey," *Proceeding - IEEE International Conference on Computing, Communication and Automation, ICCCA 2016*, pp. 884–889, 2017, doi: 10.1109/CCAA.2016.7813841.

[14] S. Kitchenham, B & Charters, "Guidelines for performing systematic literature reviews in software engineering," *Technical report, Ver. 2.3 EBSE Technical Report. EBSE*, vol. 1, no. January 2007, pp. 1–54, 2007, [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.117.471&rep=rep1&type=pdf

[15] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic Mapping Studies in Software Engineering," in *BASE - Revista de Administração e Contabilidade da Unisinos*, Jun. 2008, pp. 1–10. doi: 10.14236/ewic/EASE2008.8.

[16] M. Petticrew and H. Roberts, *Systematic Reviews in the Social Sciences*, vol. 29, no. 2. Wiley, 2006. doi: 10.1002/9780470754887.

[17] 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, doi: 10.1109/ACCESS.2019.2938260.

[18] R. Mukherjee and K. S. Patnaik, "A survey on different approaches for software test case prioritization," *Journal of King Saud University - Computer and Information Sciences*, vol. 33, no. 9, pp. 1041–1054, 2021, doi: 10.1016/j.jksuci.2018.09.005.

[19] 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 Special Issue, pp. 177–185, 2017.

[20] C. Henard, M. Papadakis, G. Perrouin, J. Klein, P. Heymans, and Y. Le Traon, "Bypassing the combinatorial explosion: Using similarity to generate and prioritize t-wise test configurations for software product lines," *IEEE Transactions on Software Engineering*, vol. 40, no. 7, pp. 650–670, 2014, doi: 10.1109/TSE.2014.2327020.

[21] 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, 2021, doi: 10.3837/tiis.2021.03.007.

[22] U. Markiegi, A. Arrieta, L. Etxeberria, and G. Sagardui, *Dynamic test prioritization of product lines: An application on configurable simulation models*, no. 0123456789. Springer US, 2021. doi: 10.1007/s11219-021-09571-0.

[23] S. Kumar, M. Mittal, and V. K. Yadav, "Cost-effective product prioritisation technique for software product line testing," *Engineering Systems Modelling and Simulation*, vol. x, no. x.

[24] J. A. P. Lima, W. D. F. Mendonça, S. R. Vergilio, and W. K. G. Assunção, "Learning-based prioritization of test cases in continuous integration of highly-configurable software," in *Proceedings of the 24th ACM Conference on Systems and Software Product Line: Volume A - Volume A*, New York, NY, USA: ACM, Oct. 2020, pp. 1–11. doi: 10.1145/3382025.3414967.

[25] S. Kumar, R. Kumar, and M. Mittal, "A hybrid approach to perform test case prioritisation and reduction for software product line testing," *International Journal of Vehicle Autonomous Systems*, vol. 15, no. 3–4, pp. 197–224, 2020, doi: 10.1504/IJVAS.2020.116439.

[26] 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," *Empir Softw Eng*, vol. 25, no. 5, pp. 3711–3769, 2020, doi: 10.1007/s10664-020-09853-4.

[27] S. Kumar, Rajkumar, and M. Rani, "Collaborative Filtering-based Test Case Prioritization and Reduction for Software Product-Line Testing," *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, vol. 2019-Octob, pp. 498–503, 2019, doi: 10.1109/TENCON.2019.8929705.

[28] 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. 1, no. 1, pp. 57–75, 2019.

[29] A. Arrieta, S. Wang, G. Sagardui, and L. Etxeberria, "Search-Based test case prioritization for simulation-Based testing of cyber-Physical system product lines," *Journal of Systems and Software*, vol. 149, pp. 1–34, Mar. 2019, doi: 10.1016/j.jss.2018.09.055.

[30] 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, 2019, doi: 10.1007/s10270-016-0569-2.

[31] H. Akimoto, Y. Isogami, T. Kitamura, N. Noda, and T. Kishi, "A Prioritization Method for SPL Pairwise Testing Based on User Profiles," *Proceedings - Asia-Pacific Software Engineering Conference, APSEC*, vol. 2019-Decem, pp. 118–125, 2019, doi: 10.1109/APSEC48747.2019.00025.

[32] X. Devroey *et al.*, "Statistical prioritization for software product line testing: an experience report," *Softw Syst Model*, vol. 16, no. 1, pp. 153–171, 2017, doi: 10.1007/s10270-015-0479-8.

[33] M. Al-Hajjaji, S. Lity, R. Lachmann, T. Thüm, I. Schaefer, and G. Saake, "Delta-Oriented Product Prioritization for Similarity-Based Product-Line Testing," *Proceedings - 2017 IEEE/ACM 2nd International Workshop on Variability and Complexity in Software Design, VACE 2017*, pp. 34–40, 2017, doi: 10.1109/VACE.2017.8.

[34] M. Al-Hajjaji, J. Kruger, S. Schulze, T. Leich, and G. Saake, "Efficient product-line testing using cluster-based product prioritization," *Proceedings - 2017 IEEE/ACM 12th International Workshop on Automation of Software Testing, AST 2017*, pp. 16–22, 2017, doi: 10.1109/AST.2017.7.

[35] Z. Akbari, S. Khoshnevis, and M. Mohsenzadeh, "A Method for Prioritizing Integration Testing in Software Product Lines Based on Feature Model," *International Journal of Software Engineering and Knowledge Engineering*, vol. 27, no. 4, pp. 575–600, 2017, doi: 10.1142/S0218194017500218.

[36] J. A. Parejo, A. B. Sánchez, S. Segura, A. Ruiz-Cortés, R. E. Lopez-Herrejon, and A. Egyed, "Multi-objective test case prioritization in highly configurable systems: A case study," *Journal of Systems and Software*, vol. 122, pp. 287–310, 2016, doi: 10.1016/j.jss.2016.09.045.

[37] 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," *FOSD 2016 - Proceedings of the 7th International Workshop on Feature-Oriented Software Development, co-located with SPLASH 2016*, vol. 1, no. 212, pp. 1–10, 2016, doi: 10.1145/3001867.3001868.

[38] R. Lachmann, S. Lity, S. Lischke, S. Beddig, S. Schulze, and I. Schaefer, "Delta-oriented test case prioritization for integration testing of software product lines," *Proceedings of the 19th International Conference on Software Product Line*, pp. 81–90, 2015, doi: 10.1145/2791060.2791073.

[39] H. Wu, C. Nie, and F. C. Kuo, "Test suite prioritization by switching cost," *Proceedings - IEEE 7th International Conference on Software Testing, Verification and Validation Workshops, ICSTW 2014*, pp. 133–142, 2014, doi: 10.1109/ICSTW.2014.15.

[40] 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," *ACM International Conference Proceeding Series*, vol. 1, pp. 32–41, 2014, doi: 10.1145/2648511.2648515.

[41] R. E. Lopez-Herrejon, J. Ferrer, F. Chicano, E. N. Haslinger, A. Egyed, and E. Alba, "A parallel evolutionary algorithm for prioritized pairwise testing of software product lines," *GECCO 2014 - Proceedings of the 2014 Genetic and Evolutionary Computation Conference*, pp. 1255–1262, 2014, doi: 10.1145/2576768.2598305.

[42] X. Devroey *et al.*, "Towards Statistical Prioritization for Software Product Lines Testing Categories and Subject Descriptors," *VaMoS '14 Proceedings of the Eighth International Workshop on Variability Modelling of Software-Intensive Systems*, p. Article No. 10, 2014.

[43] M. Al-Hajjaji, T. Thüm, J. Meinicke, M. Lochau, and G. Saake, "Similarity-based prioritization in software product-line testing," *ACM International Conference Proceeding Series*, vol. 1, pp. 197–206, 2014, doi: 10.1145/2648511.2648532.