HYBRID SEARCH-BASED AND STRING SIMILARITY-BASED PRIORITIZATION TECHNIQUE FOR REGRESSION TESTING

MUHAMMAD IRSYAD BIN KAMIL RIADZ

UNIVERSITI TEKNOLOGI MALAYSIA

HYBRID SEARCH-BASED AND STRING SIMILARITY-BASED PRIORITIZATION TECHNIQUE FOR REGRESSION TESTING

MUHAMMAD IRSYAD BIN KAMIL RIADZ

A thesis submitted in fulfilment of the
requirements for the award of the degree of
Master of Philosophy

School of Computing
Faculty of Engineering
Universiti Teknologi Malaysia

APRIL 2022

ACKNOWLEDGEMENT

In the Name of Allah, The Most Gracious and The Most Merciful

First and foremost, all praise to The Almighty, all praise to Allah for granting me strength and knowledge to complete this thesis. In preparing this thesis, I was in contact with many people, researchers, academicians, and practitioners. They have contributed towards my understanding and thoughts. I wish to express my sincere appreciation to my main thesis supervisor, Prof. Ts. Dr. Dayang Norhayati Abang Jawawi, for encouragement, guidance, criticism, and friendship. I am also very thankful to my co-supervisor Assoc. Prof. Dr Rosbi bin Mamat and Dr. Raja Zahilah Binti Raja Mohd Radzi for their guidance, advice, and motivation. Without their continued support and interest, this thesis would not have been the same as presented here.

I am also indebted to Universiti Teknologi Malaysia (UTM) for funding my Master study. Librarians at UTM, management staff at School of computing also deserve special thanks for their assistance in supplying the relevant literature and submitting my thesis.

My fellow postgraduate student should also be recognised for their support. My sincere appreciation also extends to all my colleagues and others who have assisted at various occasions. Their views and tips are useful, indeed. Unfortunately, it is not possible to list all of them in this limited space.

"My dearest dad, mum, family and friends. This is all for you."

ABSTRACT

Testers have popularly used regression testing in detecting errors encountered after changes were made. Numerous techniques were introduced in maximizing average percentage fault detection (APFD). Based on recent studies, test case prioritization (TCP) technique can give the highest APFD score. However, each approach used in TCP has limitations such as high execution cost and lack of information. Approaches that can cover more than one variable of test suite remained unseen. Thus, there is a need for a hybrid TCP technique to be developed to search for the best test plan that gives a good APFD score while having a good coverage of test cases relevant to the cost execution. Ordering the test cases based on the string similarity is one of the conventional approaches used by researchers. With the usage of string similarity, the study can gain more information regarding the test suite. This study aims to maximize the high rate of fault detection while reducing cost by decreasing the number of test cases. In this research, two TCP techniques which are string similarity-based and search-based were hybridized to form a new hybrid TCP technique and applied with Test Case Selection using weight-based to consider more variables during regression. The whole process begins by calculating string similarity for TCP with an enhanced Jaro-Winkler, then prioritizing test cases using a search-based approach with a genetic algorithm based on fault revealing. Each process generates a test plan, and those test plans will be merged and selected to form a new test plan. The selection process is structured using a weight-based approach. The experimental result showed that the final test plan produced second highest APFD with 89.60%, covers 74.10% of test case coverage and 82% of APFD, covering 77.55% of test case coverage in Siemens dataset and Smart Wheelchair System (SWS) case study. In conclusion, the proposed technique has benefited all approaches applied by getting a good APFD and coverage score. Thus, the proposed technique has proven to be cost-effective, as the APFD and coverage score are significant as the size of test suite decreases.

ABSTRAK

Kebanyakan penguji telah menggunakan ujian regresi dalam mengesan kesilapan yang berlaku setelah perubahan dilakukan. Pelbagai teknik telah diperkenalkan dalam memaksimumkan pengesanan kesalahan peratusan purata (APFD). Berdasarkan kajian baru-baru ini, teknik mengutamakan kes ujian (TCP) dapat memberikan skor APFD tertinggi. Namun, setiap pendekatan yang digunakan dalam TCP mempunyai pembatasan seperti kos pelaksanaan yang tinggi dan kekurangan informasi. Pendekatan yang boleh merangkumi lebih daripada satu pemboleh ubah set ujian hingga kini tidak dapat dikenalpasti. Oleh itu, ada keperluan untuk teknik hibrid diperkenalkan untuk mencari rancangan ujian terbaik yang memberikan skor APFD yang baik disamping mempunyai liputan kes ujian yang baik yang berkaitan dengan kos pelaksanaan. Menyusun kes ujian berdasarkan kesamaan rentetan adalah salah satu pendekatan konvensional yang digunakan oleh penyelidik. Dengan pengunaan kesamaan rentetan, kajian ini dapat memperoleh lebih banyak maklumat daripada set ujian. Tujuan kajian ini adalah untuk memaksimumkan kadar pengesanan kesalahan yang tinggi sekaligus mengurangkan kos dengan mengurangkan jumlah kes ujian. Dalam penyelidikan ini, dua teknik TCP iaitu pendekatan kesamaan rentetan dan pendekatan pencarian telah dihibridisasikan untuk membentuk teknik TCP hibrid baharu dan dilaksanakan bagi teknik Memilih Kes Ujian dengan pendekatan keberatan untuk mempertimbangkan lebih banyak pemboleh ubah semasa regresi. Seluruh proses dimulakan dengan mengira kesamaan rentetan untuk TCP dengan menggunakan algorithma Jaro-Winkler yang ditambahbaik, diikuti dengan, memberi keutamaan set ujian dengan menggunakan pendekatan berdasarkan carian algoritma genetik dalam mengenalpasti ralat. Setiap proses menghasilkan rancangan ujian dan rancangan ujian tersebut akan digabungkan dan dipilih untuk membentuk rancangan ujian baru. Proses pemilihan set ujian telah disusun menggunakan pendekatan keberatan. Hasil eksperimen menunjukkan bahawa rancangan ujian akhir menghasilkan APFD kedua tertinggi dengan 89.60%, merangkumi 74.10% liputan kes ujian dalam set data Siemens dan 82% APFD, merangkumi 77.55% liputan kes ujian bagi kajian kes Smart Wheelchair System (SWS). Kesimpulannya, teknik yang dicadangkan telah menguntungkan semua

pendekatan yang diterapkan dengan mendapatkan APFD dan skor liputan yang baik. Malah, teknik yang dicadangkan juga telah terbukti menjimatkan kos, kerana skor APFD dan liputan adalah signifikan bila keluasan set ujian menurun.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACO	-	Ant Colony Optimization
ANOVA	-	Analysis of variance
BLE	-	Bluetooth Low Energy
CR	-	Change Request
CI	-	Continuous Integration
CS	-	Cosine Similarity
CSA	-	Cuckoo-Search Algorithm
EJW	-	Enhanced Jaro-Winkler
FWER	-	Family-Wise Error Rate
GA	-	Genetic Algorithm
HTML	-	Hyper-text Marking Language
JC	-	Jaccard Index
NN		Neural Network
PSO	-	Particle Swarm Optimization
RTO	-	Regression Test Optimization
SIR	-	Software-architecture Infrastructure Repository
SLR	-	Systematic Literature Review
SMS	-	Systematic Mapping Study
SUT	-	Software Under Test
SWS	-	Systematic Wheelchair System
TCP	-	Test Case Prioritization
TCS	-	Test Case Selection
TP	-	Test Plan
TSM	-	Test Suite Minimization
UML	-	Unified Modelling Language
USEI	-	University Software Engineering Institute
UTM	-	Universiti Teknologi Malaysia
XML	-	Extensible Markup Language

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Overview

Nowadays, every company needs software to assist in their daily works. Software may assist in managing staff, machinery, stock exchange and many more. As the era of software emerges, software testing technology may also have expanded. It is acknowledged that there is a difference in software testing between academic theory and industry work (Rafi et al., 2012). The application of software testing in industry is important because software testing still plays a major role which determines the success of a software.

Software testing is described as an imperative analysis technique aimed to provide stakeholders with information about the quality of a product or service being evaluated. In a software development lifecycle, software development is a critical process, dedicated to ensuring the program meets the quality level, specifications and customer needs. Expansion of the technology has introduced new methods of research, methodologies that bring more obstacles, difficulties, and weaknesses.

As both size and complexity of software systems increase, quality becomes difficult to produce and fragile. Such increasingly complex systems increase the complexity of existing research problems and generate new problems (Tilley and Floss, 2014). Developers are aware of the frustration arising from software bugs and they are determined to solve the issue as one of their responsibilities is to enhance software quality.

Donald Firesmith of the University Software Engineering Institute (SEI) specifically has defined a number of recurrent issues in software creation related to software testing (Firesmith, 2013). There are two types of test problems based on his results, which are general test problems and test type-specific problems. Overall testing concerns were further divided into eight sub-categories: research preparation and scheduling concerns, participation of stakeholders and interaction issues, management related testing issues, testing organizational and integrity issues, testing process issues, testing methods and environmental issues, testing communication issues, and testing related testing issues. However, problems that occur during the implementation phase will be dealt with during regression testing. In addition, a number of techniques are normally applied in regression testing and the most highly anticipated technique is test case prioritization (TCP) due to its fault detection rate.

Other techniques worth mentioning are test case selection (TCS) and test suite minimization (TSM). These techniques, including TCP, are involved not only regression testing but also other testing as well. It is proven that these techniques solve not only regression issues but other software testing issues as well. Nevertheless, it is believed that merging two similar techniques can produce an optimized technique which leads to the introduction of hybrids. Hybrid technique has high probability of delivering better result.

1.2 Background

Regression testing is a validation method. It is a testing method that is run to ensure the modification made does not affect the existing module of a product. It is widely used by the industries before publishing their applications and incorporates four techniques, which are Retest-All, Test Case Selection (TCS), Test Suite Minimization (TSM) and Test Case Prioritization (TCP). Retest-All is the simplest technique where all test cases will be executed without any filtering or ordering. However, it is not practical for a large-scale system due to its large number of test cases, thus, high cost and high time consumption.

Test Suite Minimization (TSM) is a method that removes redundant test cases that do not added tangible value in the mean of test coverage (Akour and Abuwardih, 2018). Its goal is to achieve a representative collection of test suites with at least no test cases that cover all the functionalities and the system's consistency. One of the major challenges of the software testers is the vast number of test cases that are required for generation and execution. This results in a nearly 50% increase in the cost of software development (Akour and Abuwardih, 2018). It is therefore important that the test cases are reduced so that pain handling by the testers is reduced.

Test case selection (TCS) is a process which re-runs the most relevant test cases with regard to system modifications or updates (Elbaum et al., 2003). TCS aims to classify test cases that are important for the most recent software updates (Khatibsyarbini et al., 2018). Finally, giving priority to the identification of the "ideal" series of test cases will optimize desirable properties, such as early detection of defects (Yunja, 2015).

The regression testing techniques are proven to be efficient in their own way, but every technique has its own limitation. Researchers have implemented many different approaches to help the selection process of regression testing such as genetic algorithms and machine learning, etc. The simple, modular, derivative free structure of these algorithms and their ability to escape the local optimum are gaining more popularity than conventional optimization methods (Gupta and Gupta, 2017). There is, however, an analysis that only modified test cases with slow execution time, low inclusion, are revealed by the algorithm (Mansour et al., 2001). Another researcher then proposed a new strategy by using neural network (NN). NN were applied in their study to select test cases that may expose new faults. In experimenting with the algorithm for clonal filtering, Lie et al. (2011) found that it may improve search efficiency by preventing local optimum problems and premature convergence. Although their results indicate that the proposed method has very high effectiveness, the time consumption in preparing the model is not practical for application to a real testing environment. Furthermore, determining the correctness of the model's output for each test input can be costly (Byun et al., 2019).

Harman (2011) believes that multi-objective optimization is sufficient to achieve all the aims in the regression testing. Regression Test Optimization (RTO) is a single objective approach that is computed and formulated to solve the selection problem (Harman, 2011). The method is therefore not realistic, as experiments typically have several different goals, such as high fault detection rate, productivity, and low costs. There are also limitations, such as the different types of dependency between test cases, which have to be taken into account in the entire RTO cycle (Harman, 2011). The plan is to come up with a method that could fulfil multiple objectives, which in this case are cost and effectiveness. It is true that testing is the longest and most expensive process in the life cycle of software development. However, there is a way to achieve both objectives by squeezing two or more certified techniques. Pang's work suggested a methodology of test case classification focused on the k-mean technique of clustering (Pang et al., 2013). Their findings show that the technology is cheaper with a decrease in the number of test cases performed. However, the overall performance of their analysis is less persuasive because during the process, accuracy cannot exceed at least 50%. This indicates there are some gaps in the research to develop in order not just to reduce cost consumption but also to create an effective process. Elbaum's work claimed that the central importance of TCP is at the rate of error detection (Elbaum et al., 2002). Prioritization methods have greatly increased the rate of failure detection in their studies. Nonetheless, the TCP technique is faced with cost problems as the testers must carry out all test cases according to the order.

Size and time would not be applied on small to medium scale system due to their number of test cases, but many have forgotten that most medical-related systems are enormous. Noguchi and Washizaki work with 17000 test cases for their medical system. Their result shows promise, which is to be expected since they applied TCP, excellent fault detection, but high time is consumed due to number of test cases executed (Noguchi and Washizaki, n.d.). To form an effective technique, one technique must be flexible and able to be applied to all scale systems. In terms of cost-effectiveness, TCP is beneficial since the technique reorders test cases such that those are more important are run earlier in the testing process. Do and Mirarab claim that even though the TCP took a long time to finish, it is beneficial since the benefits gained from early fault detection are high enough to compensate for the cost and time incurred (Do and Mirarab, 2010). However, as time passes by, new requirements will exist in a system which leads to the increase of test suite size and with the current economic trend, high fault detection rate can never cope with the high execution cost.

Applying single technique or single approach on regression testing is a traditional way of doing regression. Applying hybrid method is much more conventional due to its strength which is multi objective. With hybrid approach, regression can consider more variables to achieve such as cost, time, effectiveness and more. In recent studies, there are two types of integration in hybridization strategy applied by researchers in their works, which are hybridization and Continuous Integration (CI). Hybridization is an integration where same technique or different algorithms in a same technique execute together to produce a hybridize solution, while CI is a process that executes technique or algorithm in sequence. CI is quite common among researchers as the strategy is less difficult to apply compared to hybridization. Nevertheless, CI are proven to be effective as proven in the work of (Ali et al., 2019; Bach et al., 2017) and for hybridization, this is the evidence to confirm the effectiveness of the strategy (Agrawal and Kaur, 2018; Pandey et al., 2018; Souza et al., 2015).

TCS with clustering method reduces cost consumption while TCP with prioritization technique is an example of hybrid technique which has high rate of fault detection. A collaboration between the techniques could bring a better impact to the community. Malhotra, Kaur & Singh (2010) and Suri & Singal (2011) were among the first few researchers who implemented more than one technique in their case study. Malhotra et al. were able to increase confidence in the correctness of the modified program (Malhotra et al., 2010). Meanwhile, Suri et al. managed to reduce the execution time and were also able to discover the faults earlier than before (Suri and Singhal, 2011). Thus, it is proven that a hybrid technique has high tendency to give a better result during a regression testing.

Arun Prakash from University of Uttar tested a hybrid algorithm method by hybridizing Levy flight and Whale Optimization Algorithm (WOA). The authors proposed hybrid approach by utilizing Levy flight random walk for search agent position update and chaotic map to avoid premature convergence of the whale optimization algorithm. In terms of results, a significant difference in fault detection ability and significant reduction cost in time and efforts without significantly decreasing the fault coverage ability is shown (Agrawal et al., 2020).

Daniel Di Nardo from University of Luxembourg and his researchers have tried working the regression testing by using TCS and TSM. The work proves that the regression testing can be hybrid depending on the requirement of researchers or testers. Their study also claims that the method has reduced the number of test cases but did not provide a significant improvement compared to the test case prioritization method (Nardo et al., 2015). Although the improvement was not significant, it manages to reduce the test cases which lead to cost reduction. Sampath et al. (2013) also argued that multiple criteria could increase the effectiveness of regression test techniques compared to those with a single criterion. The effectiveness of a regression testing can be measured by several variables. In their study, they measured average percentage fault detection (APFD) and based on their findings, the APFD increases as the number of criteria increase (Sampath et al., 2013). It is proven that hybrid approach can optimize the regression testing by reducing the test cases along with time execution and cost while maintaining fault detection capability.

The goal of TCP is to reschedule regression test cases so that certain test criterion (for example, improved fault detection rates) can be reached more rapidly (Alves et al., 2013). Thus, having a good, prioritized test suite, might come in handy since the testers will not be losing much of the testing potential due to resource limitation (e.g., time constraint and cost). According to the systematic literature review on TCP approaches, it can be concluded that each approach has specified potential values, advantages and limitations which means that all TCP approaches have their limitations. For example, a search-based approach tends to minimize coverage, so it can complete its path in a shorter period (Khatibsyarbini et al., 2018). The challenge illustrated by TCP approaches can be solved by considering executing a hybrid approach where it can cover a greater range of challenges.

TCP similarity-based is seen to be a potential approach to be hybridized considering its process does not require high power. Similarity rate between test cluster can minimize the time consumption for the testing process by reducing the number of iterations in the test case searching process (Gokilavani and Bharathi, 2021). This approach can be applied to eliminate cost and time constraint which justified the recommendation for researchers to apply this approach in their hybridization experiment.

Hybrid approach can also be inefficient due to the memory consumed while executing the process. The memory consumption and processing power has been observed and proven to be less efficient when memory consumption and processing time are considered (Banias, 2019). To create an efficient hybrid regression, these variables need to be checked and observed, so that these variables would not affect the results of the testing.

1.3 Research Questions

In an agile background, regression testing is to be carried out at the end of each sprint and release (Kandil et al., 2016). TCP and TCS in regression testing are struggling in solving their problems. First, TCP technique can produce high fault detection rate, but it requires longer time. Second, TCS technique can reduce time execution along with cost (due to decreased test suite size) but the scalability issues are against it. Thus, this study is required to develop a hybrid approach that combines TCP and TCS in producing an optimal result. The general research questions this research tries to answer are as follows:

"How can an optimal result be achieved by hybridizing prioritization with selection technique in regression testing?"

How can string algorithm be enhanced and applied in finding the similarity between test cases?
How can the enhanced prioritization technique be integrated into selection technique?
What is the process in applying the proposed technique?
How can the effectiveness of the proposed technique be evaluated?

1.4 Research Aim

The aim of this research to develop a regression testing hybrid technique that could achieve cost-effectiveness with high fault detection rate and relevant test cases. Furthermore, considering more than one variable as experiment measure has also motivated this research path.

1.5 Research Objectives

From the aim of the study and derived research questions, the main objectives for this research are:

(a) To enhance string algorithm to apply in string similarity-based TCP for use in a hybridized technique in objective (b)

(b) To propose a hybrid TCP technique by hybridizing two prioritization test plans and applying selection

To evaluate the implementation of the proposed technique on primary and secondary experiment with different techniques.

1.6 Scope of Study

In order to produce an optimized hybrid technique with requirement change, this research is focused on the following scope:

(a) The research focuses on small-scale medical equipment

JavaScript will be used for the development of the experimentation setup

Hybrid strategy in regression will be studied together with TCP and TCS technique

Three benchmark programs and a case study will be used to compare the findings of the hybrid prioritization technique with native technique.

1.7 Significant Contribution of Study

The study on a hybrid technique in regression testing is important in the context of software testing. Findings from this research will contribute to our understanding on how regression testing can be performed at a much-reduced cost with reduced test suite size. This study also gives benefits to the software tester community as they can apply the technique while knowing that the fault detection capability in regression testing has increased.

Furthermore, this study will also provide insights in the execution performance of each string algorithm which lead to the introduction of the enhanced string algorithm. The enhanced string algorithm will be evaluated with other algorithms using APFD metric. Next contribution is at the proposed hybridization method where two test plans created by TCP are merged and undergo TCS process. Overall process will be tested on primary and secondary experiment. Statistic test will also be executed to bring a statistical significance (referred to later in Chapter 5).

1.8 Thesis Structure and Organization

The thesis is outlined as follows:

Chapter 1 provides a brief overview of the research. It consists of introduction, challenges, research questions, objective, and scope of study. In this chapter, the proposed technique is also elaborated briefly in terms of its history, statistic and challenges.

Chapter 2 provides an outline of related works on the proposed technique. TCP, TCS and TSM are compared and studied in terms of strength and weaknesses. The hybrid technique is also reviewed with proof from existing studies. String algorithm, search algorithm and selection algorithm are discussed and elaborated in detail. The existing work for the algorithms is compared and tabulated.

Chapter 3 provides a summary on the research framework, also known as research methodologies. This chapter describes the flow of the research study from problem statement until result and discussion. This chapter also briefly introduces the primary experiment and secondary experiment, which will be further discussed in the next chapter.

Chapter 4 provides a detailed process of the proposed technique. The experiment setup is explained in detail, step by step, including the experiment design and interfaces. The chapter continues with elaborations on the application of similarity-based approach on TCP, followed by search-based approach on TCP and lastly, the hybridization process. There are also examples of applying the proposed technique by using the benchmark dataset. This chapter elaborates the process implementation of the proposed technique in every step.

Chapter 5 provides the result from the application of secondary and primary experiment. The initial result uses the same experiment that were prepared for primary experiment which allow readers to have more understanding when applying the proposed technique to a medium-scale dataset and small-scale dataset. The technique supports string similarity based on software traceability in calculating similarities between test cases. The result is compared with existing technique in terms of fault detection rate and test case coverage. This chapter also provides a statistical evaluation of the proposed work on case study. The author utilized Average Percentage of Faults Detected (APFD), and Kruskal Wallis to provide an empirical assessment of the findings of the experiment.

Chapter 6 provides the conclusion, contribution, limitations, and future works of this research.

REFERENCES

Agrawal, A. P., Delhi, N., Choudhary, A., Kaur, A., and Delhi, N. (2020). *An Effective Regression Test Case Selection Using Hybrid Whale Optimization Algorithm*. *11*(1), 53–67. https://doi.org/10.4018/IJDST.2020010105

Agrawal, A. P., and Kaur, A. (2018). *A Comprehensive Comparison of Ant Colony and Hybrid Particle Swarm Optimization Algorithms Through Test Case Selection*. 397–405. https://doi.org/10.1007/978-981-10-3223-3

Akour, M., and Abuwardih, L. (2018). Test Case Minimization using Genetic Algorithm : Pilot Study. *2018 8th International Conference on Computer Science and Information Technology (CSIT)*, 66–70.

Ali, S., Hafeez, Y., Hussain, S., and Yang, S. (2019). Enhanced regression testing technique for agile software development and continuous integration strategies. *Software Quality Journal*.

Alves, E. L. G., Machado, P. D. L., Massoni, T., and Santos, S. T. C. (2013). A refactoring-based approach for test case selection and prioritization. *2013 8th International Workshop on Automation of Software Test, AST 2013 - Proceedings*, 93–99. https://doi.org/10.1109/IWAST.2013.6595798

Arrieta, A., Wang, S., Markiegi, U., Arruabarrena, A., Etxeberria, L., and Sagardui, G. (2019). Pareto efficient multi-objective black-box test case selection for simulation-based testing. *INFSOF*, *114*(November 2018), 137–154. https://doi.org/10.1016/j.infsof.2019.06.009

Arrieta, A., Wang, S., Sagardui, G., and Etxeberria, L. (2019). Search-Based test case prioritization for simulation-Based testing of cyber-Physical system product lines. *Journal of Systems and Software*, *149*, 1–34. https://doi.org/10.1016/j.jss.2018.09.055

Bach, T., Andrzejak, A., and Pannemans, R. (2017). Coverage-Based Reduction of Test Execution Time : Lessons from a Very Large Industrial Project. *2017 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, *Section III*, 3–12. https://doi.org/10.1109/ICSTW.2017.6

Bach, T., Pannemans, R., and Schwedes, S. (2018). Effects of an economic approach for test case selection and reduction for a large industrial project. *Proceedings - 2018 IEEE 11th International Conference on Software Testing, Verification and Validation Workshops, ICSTW 2018*, 374–379. https://doi.org/10.1109/ICSTW.2018.00076

Bajaj, A., and Sangwan, O. P. (2021). Tri-level regression testing using nature-inspired algorithms. *Innovations in Systems and Software Engineering*, *17*(1), 1–16. https://doi.org/10.1007/s11334-021-00384-9

Banias, O. (2019). Test case selection-prioritization approach based on memoization dynamic programming algorithm. *Information and Software Technology*, *115*(June), 119–130. https://doi.org/10.1016/j.infsof.2019.06.001

Boyce, B. R. (1990). Concepts of information retrieval and automatic text processing: The transformation analysis, and retrieval of information by computer. *Journal of the American Society for Information Science*, *41*(2), 150–151. https://doi.org/10.1002/(sici)1097-4571(199003)41:2<150::aid-asi12>3.0.co;2-8

Byun, T., Sharma, V., Vijayakumar, A., Rayadurgam, S., and Cofer, D. (2019). Input prioritization for testing neural networks. *Proceedings - 2019 IEEE International Conference on Artificial Intelligence Testing, AITest 2019*, 63–70. https://doi.org/10.1109/AITest.2019.000-6

Coutinho, A. E. V. B., Cartaxo, E. G., and Machado, P. D. de L. (2016). Analysis of distance functions for similarity-based test suite reduction in the context of model-based testing. In *Software Quality Journal* (Vol. 24, Issue 2). https://doi.org/10.1007/s11219-014-9265-z

de Oliveira Neto, F. G., Ahmad, A., Leifler, O., Sandahl, K., and Enoiu, E. (2018). Improving continuous integration with similarity-based test case selection. *AST*, 39–45. https://doi.org/10.1145/3194733.3194744

Do, H., and Mirarab, S. (2010). *The Effects of Time Constraints on Test Case Prioritization : A Series of Controlled Experiments*. *36*(5), 593–617.

Doane, D. P., Seward, L. E., Doane, D. P., and Seward, L. E. (2017). *Measuring Skewness : A Forgotten Statistic ? Measuring Skewness : A Forgotten Statistic ?* *1898*(December). https://doi.org/10.1080/10691898.2011.11889611

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

Elbaum, S., Malishevsky, A. G., and Rothermel, G. (2002). Test case prioritization: A family of empirical studies. *IEEE Transactions on Software Engineering*, *28*(2), 159–182. https://doi.org/10.1109/32.988497

Engström, E., Runeson, P., and Ljung, A. (2011). Improving regression testing transparency and efficiency with history-based prioritization - An industrial case study. *Proceedings - 4th IEEE International Conference on Software Testing, Verification, and Validation, ICST 2011*, 367–376. https://doi.org/10.1109/ICST.2011.27

Engström, E., Runeson, P., and Skoglund, M. (2010). A systematic review on regression test selection techniques. In *Information and Software Technology* (Vol. 52, Issue 1, pp. 14–30). https://doi.org/10.1016/j.infsof.2009.07.001

Fang, C., Chen, Z., Wu, K., and Zhao, Z. (2014). Similarity-based test case prioritization using ordered sequences of program entities. *Software Quality Journal*, *22*(2), 335–361. https://doi.org/10.1007/s11219-013-9224-0

Gokilavani, N., and Bharathi, B. (2021). Microprocessors and Microsystems Multi-Objective based test case selection and prioritization for distributed cloud environment. *Microprocessors and Microsystems*, *82*(January), 103964. https://doi.org/10.1016/j.micpro.2021.103964

Gomma, W. H., and Fahmy, A. A. (2013). A Survey of Text Similarity Approaches. *International Journal of Computer Applications*, *68*(13), 13–18.

Gupta, D., and Gupta, V. (2017). Test Suite Prioritization Using Nature Inspired Meta-Heuristic Algorithms. In A. M. Madureira, A. Abraham, D. Gamboa, and P. Novais (Eds.), *Intelligent Systems Design and Applications* (Vol. 557, pp. 216–226). Springer International Publishing. https://doi.org/10.1007/978-3-319-53480-0_22

Habtemariam, G. M., and Mohapatra, S. K. (2019). A Genetic Algorithm-Based Approach for Test Case Prioritization. In *Communications in Computer and Information Science* (Vol. 1026). Springer International Publishing. https://doi.org/10.1007/978-3-030-26630-1_3

Hall, P. A. V., and Dowling, G. R. (1980). Approximate String Matching. *ACM Computing Surveys (CSUR)*, *12*(4), 381–402. https://doi.org/10.1145/356827.356830

Harman, M. (2011). Making the case for MORTO: Multi objective regression test optimization. *Proceedings - 4th IEEE International Conference on Software Testing, Verification, and Validation Workshops, ICSTW 2011*, 111–114. https://doi.org/10.1109/ICSTW.2011.60

Hettiarachchi, C., Do, H., and Choi, B. (2016). Risk-based test case prioritization using a fuzzy expert system. *Information and Software Technology*, *69*, 1–15. https://doi.org/10.1016/j.infsof.2015.08.008

Jahan, H., Feng, Z., and Mahmud, S. M. H. (2020). Risk-Based Test Case Prioritization by Correlating System Methods and Their Associated Risks. *Arabian Journal for Science and Engineering*, *45*(8), 6125–6138. https://doi.org/10.1007/s13369-020-04472-z

Jaro, M. A. (1989). Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. *Journal of the American Statistical Association*, *84*(406), 414–420. https://doi.org/10.1080/01621459.1989.10478785

Jiang, B., Zhang, Z., Chan, W. K., Tse, T. H., and Chen, T. Y. (2012). How well does test case prioritization integrate with statistical fault localization? *Information and Software Technology*, *54*(7), 739–758. https://doi.org/10.1016/j.infsof.2012.01.006

Jung, P., Kang, S., and Lee, J. (2019). Automated code-based test selection for software product line regression testing. *JSS*, *158*. https://doi.org/10.1016/j.jss.2019.110419

Kandil, P., Moussa, S., and Badr, N. (2016). *Cluster-based test cases prioritization and selection technique for agile regression testing*. https://doi.org/10.1002/smr

Kapfhammer, G. M. (2011). Empirically evaluating regression testing techniques: Challenges, solutions, and a potential way forward. *Proceedings - 4th IEEE International Conference on Software Testing, Verification, and Validation Workshops, ICSTW 2011*, 99–102. https://doi.org/10.1109/ICSTW.2011.88

Kazmi, R., Jawawi, D. N. A., Mohamad, R., and Ghani, I. (2017). Effective regression test case selection: A systematic literature review. *ACM Computing Surveys*, *50*(2). https://doi.org/10.1145/3057269

Kazmi, S. R. H. S. (2017). *EFFECTIVE REGRESSION TEST CASE SELECTION TECHNIQUE USING WEIGHTED AVERAGE SCORING*. Universiti Teknologi Malaysia.

Khatibsyarbini, M., Isa, M. A., and Jawawi, D. N. A. (2017). A hybrid weight-based and string distances using particle swarm optimization for prioritizing test cases. *Journal of Theoretical and Applied Information Technology*, *95*(12), 2723–2732.

Khatibsyarbini, M., Isa, M. A., Jawawi, D. N. A., Hamed, H. N. A., and Mohamed Suffian, M. D. (2019). Test Case Prioritization Using Firefly Algorithm for Software Testing. *IEEE Access*, *7*, 132360–132373. https://doi.org/10.1109/ACCESS.2019.2940620

Khatibsyarbini, M., Isa, M. A., Jawawi, D. N. A., and Tumeng, R. (2018). Test case prioritization approaches in regression testing: A systematic literature review. *Information and Software Technology*, *93*, 74–93. https://doi.org/10.1016/j.infsof.2017.08.014

Kruskal, W. H., and Wallis, W. A. (1952). *Journal of the American Use of Ranks in One- Criterion Variance Analysis*. *August 2014*, 37–41. https://doi.org/10.1080/01621459.1952.10483441

Ledru, Y., Petrenko, A., Boroday, S., and Mandran, N. (2012). Prioritizing test cases with string distances. *Automated Software Engineering*, *19*(1), 65–95. https://doi.org/10.1007/s10515-011-0093-0

Li, B., and Han, L. (2013). Distance weighted cosine similarity measure for text classification. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *8206 LNCS*, 611–618. https://doi.org/10.1007/978-3-642-41278-3_74

Li, M., Chen, X., Li, X., Ma, B., and Vitányi, P. M. B. (2004). The similarity metric. *IEEE Transactions on Information Theory*, *50*(12), 3250–3264. https://doi.org/10.1109/TIT.2004.838101

Magalhães, C., Andrade, J., Perrusi, L., Mota, A., Barros, F., and Maia, E. (2020). HSP: A hybrid selection and prioritisation of regression test cases based on information retrieval and code coverage applied on an industrial case study.

Journal of Systems and Software, *159*, 110430. https://doi.org/10.1016/j.jss.2019.110430

Malhotra, R., Kaur, A., and Singh, Y. (2010). *A Regression Test Selection and Prioritization Technique*. *6*(2), 235–252.

Mansour, N., Bahsoon, R., and Baradhi, G. (2001). Empirical comparison of regression test selection algorithms. *Journal of Systems and Software*, *57*(1), 79–90. https://doi.org/10.1016/S0164-1212(00)00119-9

Markiegi, U., Arrieta, A., Etxeberria, L., and Sagardui, G. (2019). Test case selection using structural coverage in software product lines for time-budget constrained scenarios. *SIGAPP*, 2362–2371. https://doi.org/10.1145/3297280.3297512

Mishra, D. B., Panda, N., Mishra, R., and Acharya, A. A. (2019). Total fault exposing potential based test case prioritization using genetic algorithm. *International Journal of Information Technology (Singapore)*, *11*(4), 633–637. https://doi.org/10.1007/s41870-018-0117-0

Misri, A. D. (2021). *Smart Wheelchair System*. Universiti Teknologi Malaysia.

Mittelmark, M. B. (2007). New IUHPE research report series gives graduate students a new avenue to disseminate their research globally. *Promotion & Education, 14*(1). https://doi.org/10.1177/10253823070140010201

Mondal, D., Hemmati, H., and Durocher, S. (2015). Exploring test suite diversification and code coverage in multi-objective test case selection. *2015 IEEE 8th International Conference on Software Testing, Verification and Validation, ICST 2015 - Proceedings*, 1–10. https://doi.org/10.1109/ICST.2015.7102588

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

Nagar, R., Kumar, A., Kumar, S., and Baghel, A. S. (2014a). Implementing test case selection and reduction techniques using meta-heuristics. *Proceedings of the 5th International Conference on Confluence 2014: The Next Generation Information Technology Summit*, 837–842. https://doi.org/10.1109/CONFLUENCE.2014.6949377

Nagar, R., Kumar, A., Kumar, S., and Baghel, A. S. (2014b). Implementing Test Case Selection and Reduction Techniques using Meta-Heuristics. *2014 5th International Conference - Confluence The Next Generation Information Technology Summit (Confluence)*, 837–842.

Nagar, R., Kumar, A., Singh, G. P., and Kumar, S. (2015). Test case selection and prioritization using cuckoos search algorithm. *2015 1st International Conference on Futuristic Trends in Computational Analysis and Knowledge Management, ABLAZE 2015*, 283–288. https://doi.org/10.1109/ABLAZE.2015.7155012

Narciso, E. N., Delamaro, M. E., and De Lourdes Dos Santos Nunes, F. (2014). Test case selection: A systematic literature review. *International Journal of Software Engineering and Knowledge Engineering*, *24*(4), 653–676. https://doi.org/10.1142/S0218194014500259

Nardo, D. Di, Alshahwan, N., Briand, L., and Labiche, Y. (2015). *Coverage-based regression test case selection , minimization and prioritization : a case study on an industrial system*. *April*, 371–396. https://doi.org/10.1002/stvr

Noguchi, T., and Washizaki, H. (n.d.). *History-Based Test Case Prioritization for Black Box Testing on a New Product using Ant Colony Optimization*. *1*, 1–2.

Noguchi, T., and Washizaki, H. (2015). History-Based Test Case Prioritization for Black Box Testing using Ant Colony Optimization. *2015 IEEE 8th International Conference on Software Testing, Verification and Validation (ICST)*, 1–2. https://doi.org/10.1109/ICST.2015.7102622

Palma, F., Abdou, T., Bener, A., Maidens, J., and Liu, S. (2018). An improvement to test case failure prediction in the context of test case prioritization. *ACM International Conference Proceeding Series*, 80–89. https://doi.org/10.1145/3273934.3273944

Pan, C., Lu, M., Xu, B., and Gao, H. (2019). *applied sciences An Improved CNN Model for Within-Project Software Defect Prediction*. *1*, 1–28.

Panda, S., and Mohapatra, D. P. (2017). Regression test suite minimization using integer linear programming model. *Software - Practice and Experience*, *47*(11), 1539–1560. https://doi.org/10.1002/spe.2485

Pandey, A., Studies, E., and Banerjee, S. (2018). *Test Suite Minimization in Regression Testing Using Hybrid Approach of ACO and GA*. *9*(3). https://doi.org/10.4018/IJAMC.2018070105

Pang, Y., Xue, X., and Namin, A. S. (2013). Identifying effective test cases through K-means clustering for enhancing regression testing. *Proceedings - 2013 12th International Conference on Machine Learning and Applications, ICMLA 2013*, *2*, 78–83. https://doi.org/10.1109/ICMLA.2013.109

Panigrahi, C. R., and Mall, R. (2010). Model-based regression test case prioritization. *Communications in Computer and Information Science*, *54*(6), 380–385. https://doi.org/10.1007/978-3-642-12035-0_39

Paygude, P. (2020). *Fault Aware Test Case Prioritization in Regression Testing*. *8*(5), 2112–2117.

Qiu, D., Li, B., Ji, S., and Leung, H. (2014). Regression testing of web service: A systematic mapping study. *ACM Computing Surveys*, *47*(2). https://doi.org/10.1145/2631685

Qu, B., Nie, C., and Xu, B. (2008). Test case prioritization for multiple processing queues. *2008 International Symposium on Information Science and Engineering, ISISE 2008*, *2*, 646–649. https://doi.org/10.1109/ISISE.2008.106

Quan Do, Thanh Pham, Wei Liu, and K. R. (2017). *WTEN: An Advanced Coupled Tensor Factorization Strategy for Learning from*. *10569*, 537–552. https://doi.org/10.1007/978-3-319-68783-4

Rafi, D. M., Moses, K. R. K., Petersen, K., and Mäntylä, M. V. (2012). Benefits and limitations of automated software testing: Systematic literature review and practitioner survey. *2012 7th International Workshop on Automation of Software Test, AST 2012 - Proceedings*, 36–42. https://doi.org/10.1109/IWAST.2012.6228988

Rogstad, E., Briand, L., and Torkar, R. (2013). Test case selection for black-box regression testing of database applications. *INFSOF*, *55*(10), 1781–1795. https://doi.org/10.1016/j.infsof.2013.04.004

Rothermel, G., Harrold, M. J., Ostrin, J., and Hong, C. (1998). Empirical study of the effects of minimization on the fault detection capabilities of test suites. *Conference on Software Maintenance*, 34–43. https://doi.org/10.1109/icsm.1998.738487

Rothermel, G., Untcn, R. H., Chu, C., and Harrold, M. J. (2001). Prioritizing test cases for regression testing. *IEEE Transactions on Software Engineering*, *27*(10), 929–948. https://doi.org/10.1109/32.962562

Saber, T., Delavernhe, F., Papadakis, M., Neill, M. O., and Ventresque, A. (2018). A Hybrid Algorithm for Multi-objective Test Case Selection. *CEC*, 1–8.

Sahak, M. (2018). *Effective similarity based test case prioritization technique for software product lines*. universiti teknologi malaysia.

Sampath, S., Bryce, R., and Memon, A. M. (2013). A uniform representation of hybrid criteria for regression testing. *IEEE Transactions on Software Engineering*, *39*(10), 1326–1344. https://doi.org/10.1109/TSE.2013.16

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

Sapna, P. G., and Mohanty, H. (2010). Automated test scenario selection based on Levenshtein distance. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 5966 LNCS*, 255–266. https://doi.org/10.1007/978-3-642-11659-9_28

Saraswat, P., Singhal, A., and Bansal, A. (2019). A review of test case prioritization and optimization techniques. In *Advances in Intelligent Systems and Computing* (Vol. 731). Springer Singapore. https://doi.org/10.1007/978-981-10-8848-3_48

Shi, A., Yung, T., Gyori, A., and Marinov, D. (2015). *Comparing and Combining Test-Suite Reduction and Regression Test Selection*. 237–247.

Souza, L. S. De, Bastos, R., Prudêncio, C., and Barros, F. A. De. (2015). *Open Access A hybrid particle swarm optimization and harmony search algorithm approach for multi-objective test case selection*. https://doi.org/10.1186/s13173-015-0038-8

Spieker, H., Gotlieb, A., Marijan, D., and Mossige, M. (2017). Reinforcement learning for automatic test case prioritization and selection in continuous integration. *ISSTA 2017 - Proceedings of the 26th ACM SIGSOFT International Symposium on Software Testing and Analysis*, 12–22. https://doi.org/10.1145/3092703.3092709

Su, W., Li, Z., Wang, Z., and Yang, D. (2020). A Meta-heuristic Test Case Prioritization Method Based on Hybrid Model. *Proceedings - 2020 International Conference on Computer Engineering and Application, ICCEA 2020*, 430–435. https://doi.org/10.1109/ICCEA50009.2020.00099

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

Suri, B., and Singhal, S. (2014). Development and validation of an improved test selection and prioritization algorithm based on ACO. *International Journal of Reliability, Quality and Safety Engineering*, *21*(6), 1–13. https://doi.org/10.1142/S0218539314500326

Thada, V., and Jaglan, V. (2013). Comparison of jaccard, dice, cosine similarity coefficient to find best fitness value for web retrieved documents using genetic algorithm. *International Journal of Innovations in Engineering and Technology*, *2*(4), 202–205.

Tilley, S., and Floss, B. (2014). *Hard Problems in Software Testing*. http://www.morganclaypool.com/doi/pdf/10.2200/S00587ED1V01Y201407SWE002

Tyagi, M., and Malhotra, S. (2014). Test case prioritization using multi objective particle swarm optimizer. *2014 International Conference on Signal Propagation and Computer Technology, ICSPCT 2014*, 390–395. https://doi.org/10.1109/ICSPCT.2014.6884931

Wagner, R. A., and Fischer, M. J. (1974). The String-to-String Correction Problem. *Journal of the ACM (JACM)*, *21*(1), 168–173. https://doi.org/10.1145/321796.321811

Whitley, D. (1994). A genetic algorithm tutorial. *Statistics and Computing*, *4*(2), 65–85. https://doi.org/10.1007/BF00175354

William W Cohen, Pradeep Ravikumar, and Stephen. (2003). A Comparison of String Distance Metrics for. *Proceedings of the IJCAI-2003 Workshop On*, 73--78.

Winkler, W. E. (1999). The State of Record Linkage and Current Research Problems. *Statistical Research Division US Census Bureau*, 1–15. https://doi.org/10.1.1.39.4336

Yan, S., Chen, Z., Zhao, Z., Zhang, C., and Zhou, Y. (2010). A dynamic test cluster sampling strategy by leveraging execution spectra information. *ICST 2010 - 3rd International Conference on Software Testing, Verification and Validation*, 147–154. https://doi.org/10.1109/ICST.2010.47

Yap, B. W. (2011). *Power Comparisons of Shapiro-Wilk , Kolmogorov-Smirnov , Lilliefors and Anderson-Darling Tests. January*.

Yunja, C. (2015). Regression testing minimization, selection and prioritization: a survey. *Software Testing, Verification and Reliability*, *Volume 21*(Issue 3), 195–214. https://doi.org/10.1002/stvr

Zhang, C., Chen, Z., Zhao, Z., Yan, S., Zhang, J., and Xu, B. (2010). An improved regression test selection technique by clustering execution profiles. *Proceedings - International Conference on Quality Software*, 171–179. https://doi.org/10.1109/QSIC.2010.16

Zhou, Z. Q. (2010). Using Coverage Information to Guide Test Case Selection in Adaptive Random Testing. *2010 IEEE 34th Annual Computer Software and Applications Conference Workshops*, 208–213. https://doi.org/10.1109/COMPSACW.2010.43

Zhou, Z. Q., Sinaga, A., and Susilo, W. (2012). On the fault-detection capabilities of adaptive random test case prioritization: Case studies with large test suites. *Proceedings of the Annual Hawaii International Conference on System Sciences*, 5584–5593. https://doi.org/10.1109/HICSS.2012.454

APPENDIX A

(Systematic Mapping Study)

Abstract

In recent years, software testing has been expanding its horizon with new tools and techniques, becoming more efficient with time. However, it will take a long time for large scale system to get past the regression testing process. If there are requirement changes done within a system, a regression testing needs to be implemented in ensuring the system is free of error. Rerunning all existing test cases is a safe but costly option. Test case selection (TCS) is one of the ways to reduce cost as it only executes test cases that are related to the requirement changes. Since TCS has been widely used by researchers, this study will conduct a systematic mapping on TCS approaches. There are a total of 91 papers identified which come from conference, journals, symposiums and workshops. From the whole, there are 27 journal articles, 49 conference papers, 5 workshop articles, and 10 symposium articles. In addition, 13 regression test selection approaches were identified. As for the result, TCS is quite popular among research due to the increasing number of articles published. Besides that, approaches in TCS are improving over time, especially model-based and coverage-based, due to a large number of publications. We discovered that the approaches were widely used since the year 2012. The variations of approaches in TCS have also benefited the researchers in applying it to their case study. In conclusion, test case selection has already been discussed and applied to not only software testing but other testing as well. The variety of approaches have helped researchers in many ways in their research. In this study, the pros and cons in each approach were not discussed. Due to that, we will provide extensive research on the advantages and disadvantages of approaches through a systematic literature review in the coming days. We also believed a study on the method used in approaches is in order.

1. Introduction

Regression testing is a verification method where it is run only if there are new requirement changes. Regression testing is conducted to confirm that the new changes will not affect existing functionality. A variety of regression testing approaches were proposed, and three major branches of the most widely used techniques have been identified: test suite minimization (TSM), test case selection (TCS), and test case prioritization (TCP). Yoo and Harman [1] wrote a survey on the approaches of minimization, selection and prioritization for regression testing. They claimed that minimization and prioritization are not as safe as they were claimed since they were forced to surrogate metrics for real fault-detection capability while selection specifically focuses on modifications between two versions of system under test

(SUT). The scope chosen for this systematic mapping study (SMS) is TCS technique. The reason why we want to explore TCS is because it is practically a safe way without jeopardizing the current state of the system.

This paper presents an SMS on TCS following the guidelines presented by [2]. SMS (or mapping study) is designed to provide a wide overview of a research area, establishing research evidence on a topic, and provide an indication of the quantity of the evidence. It involves searching the literature to know what topics have been covered, and where the papers were published [2]. The results of a mapping analysis are an inventory of papers on the subject, mapped to a classification [3]. A mapping study thus provides an overview of the area's scope and allows research gaps and trends to be discovered [2].

The research area that we are focusing on is TCS approaches. Each testing process will undoubtedly consume cost but TCS approach comes with a lower cost as it did not execute all test cases. The selection can be done using various techniques, with each technique having its strengths and weaknesses. Not all techniques are applicable to the system, it depends on the specific environment or system. Each invocation of a service, for example, may incur certain costs in web services, which reduces the number of payable invocations, making effective regression testing more prominent for web services [4]. So, in this paper, our main role is to deliver a statistical analysis on TCS not only on regression testing but other software testing as well. A study proposed an effective metrics on black box testing using TCS [5]. They claimed that the metrics performed significantly better than traditional techniques, thus proving that TCS is also applicable to other software testing.

Even though numerous TCS approaches exist, no latest mapping study has been made into the topic which leads to the importance of this work. Thus, it is hoped that the research gap will be closed at the end of this work.

2. Related Work

Our paper is a tertiary study which follows the guidelines of SMS. In this section, all published tertiary studies will be identified and researched to improve and close the gap left by other researchers. Tertiary studies must be conducted within the research areas that have a large number of publications [3]. We will collect and review all published papers and analyse data obtained to find the research gap. In our study, several articles are similar to ours and worth mentioning.

In the work of [6], the authors have reviewed and evaluated available regression test selection techniques. Out of 2923 papers analysed, 28 papers were selected to undergo the evaluation process. Their study concluded that there is no technique that supports or provides empirical evidence, except for a small group of related techniques [6].

Besides that, in the work of [7], the authors have published a systematic literature review on TCS in regression testing. Their study reviewed 449 published articles before shortlisting the number to 32 quality papers which met their research objectives. Their research showed adaptive random selection, genetic algorithm, and greedy algorithm are the most widely mentioned methods. Not only that, most approaches rely on heuristics such as test case diversity, and coverage of code or model. The paper also addressed the principles and approaches available for TCS, as well as field of application and evaluation metrics [7].

The only SMS that was similar to our study was presented in the work of [4]. The authors provided a qualitative analysis on TCS. It is an SMS focusing specifically on web services for regression testing. The authors managed to collect a total of 60 papers on TCP, TCS and TSM. They claimed that a large number of theoretical and experimental regression test techniques were not proved, limited to large-scale systems. They thought that their survey found holes in the literature and provided new insights into future research [4].

Kazmi's work [8] came out with a systematic literature review on regression test selection in 2017. The study reviewed related papers between 2007 and 2015. The papers were then categorized according to standard SLR procedure. Their study shows mining and learning-based regression TCS covers at least 39% of the literature, while

18% involves unit level testing and 26% include object-oriented environment (Java) [8].

On the other hand, in the work of [1], the authors surveyed each area of regression testing technique which includes minimization, selection and prioritization. Based on the study, it is evident that TCP is becoming an increasingly important regression testing technique. The trend shows a rising number of publications in this field which provides evidence that support the popularity of TCP, TCS and TSM. Their survey also provided evidence indicating that there is a preponderance of empirical work that draws upon small scale subjects [1].

All existing papers related to this topic are tabulated and shown in Table 1. It is clear that existing TCS studies only focus on the fundamental of selected approaches. The most recent study by [8] focused on revealing methods used by researchers. The pros and cons of the method were elaborated in the study. The analysis made by the authors is mainly focusing on the existing methods and guidelines for applying TCS. They managed to discover five distinct families of ways which are associated with TCS techniques. Besides that, the author also studied the effectiveness of existing TCS techniques. The efficiency was measure based on cost, coverage and faults. However, they never intended to explore the variety of existing approaches. So, we aim to study existing approaches using systematic mapping approach.

Table 1. Summary of related studies in regression testing.

Study Type	**Study Reference**	**Study Focus**	**Year of Publication**	**Total Studies Reviewed**	**Years Covered**
SLR	Everton et al. (Narciso et al., 2014)	TCS	2014	32	2006 - 2012
SMS	Dong et al. (Qiu et al., 2014)	Regression Testing	2014	30	2000 - 2013
SLR	Rafaqut et al. (R. Kazmi et al., 2017)	TCS	2017	47	2007 - 2015
SLR	Emelie et al. [6]	TCS	2008	27	1969 - 2006

Survey	Yoo et al. [1]	Regression Testing	2010	159	1977 - 2009

3. Research Method

Figure 1 below demonstrates a standardized process for producing a SMS. This method was implemented to answer all research questions listed. This method is inspired by [9]. In the work of [2], the authors study and make a comparison on every systematic mapping guideline, and the result showed that [10] are one of the best practices available in this time. In addition, their works were also proven to be highly reliable since 2162 people cited their paper. Due to that fact, we were not only inspired by their work but also confident with our work when applying their practices in our study.

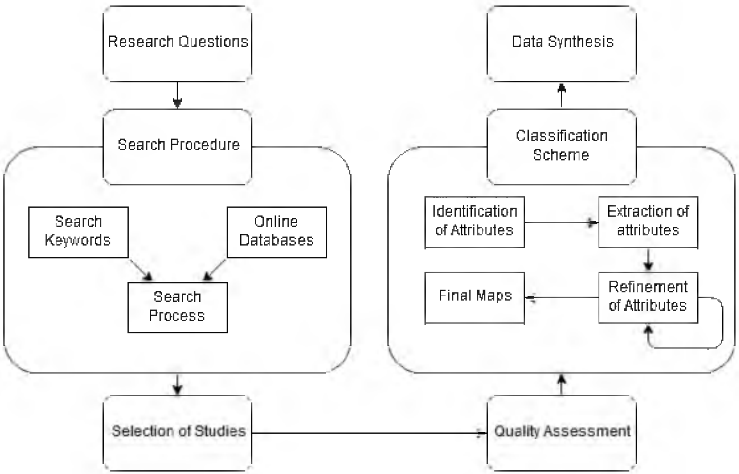

Figure 1. Structured SMS method [9].

The aim of this SMS is to map and identify the current trend of using TCS technique. SMS aims to help other researchers envision research differences in TCS. Therefore, four research questions with respective motivations are presented in Table 2.

Table 2. Research questions and motivations.

Research questions	**RQ statement**	**Motivations**
RQ 1	What are the most popular approaches used in test case selection?	These research questions aim to recognize the areas most frequently covered by TCS
RQ 1.1	Do the approaches used improve over time?	
RQ 2	Which publication published the most test case selection papers?	This research question aims to identify where the papers related to regression testing originate

RQ 3	How frequent is test case selection applied in each software testing?	This research question aims to identify the novelty of TCS in other software testing phase
RQ 4	What are the case studies commonly used by researchers?	This research question aims to identify the most frequent case studies used

SMS's strategy is to focus on the width or broadness of a topic, in this case, test case selection. The procedure consists of search strings, online databases, search process and selection of studies. Search has been done by using specific search queries namely "Regression Test Selection", "Test Case Selection" and "Testing". It is highly likely that when the authors searched the exact queries alone which we mentioned earlier, it will return a large number of papers that includes mapping, review, comparison which are unrelated in this study. To solve the issue, 'AND' operator was used to link all search queries.

We searched for related publications from five online repositories including ACM Digital Library, IEEExplore Science Direct, Web of Science and Scopus. The search results will provide the title, abstract and keywords of the publications. Papers from all library (excluding Scopus) are likely to be redundant with results from Scopus.

There are reasons why we select these online databases. IEEE Xplore provides a variety of conferences papers and symposium articles. Also, the ACM digital library offers more reports from workshops used for the primary studies. Other digital libraries remained important since they host journal articles related to our studies. From our findings, journals were also identified in IEEE Explore and ACM Digital Library.

Initially, we found 463 papers from the five online databases. Next, to select the final papers, the inclusion and exclusion criteria need to be applied. The applied principles decrease the number of manuscripts from 463 articles to 91 articles. The papers were extracted from the online databases, and the number of papers from each database is shown below in Figure 2. The criteria are as follows: (1) Paper must be written in English; (2) Paper must be able to correspond to at least one research question; (3) Paper must be focusing on TCS approaches; (4) Paper must provide approach description and other associated metrics; and (5) Paper must have bibliographic information.

4. Result and Discussion

In this section, we present our findings, answer to the research questions along with a discussion.

RQ 1. Which are the most popular approaches used in test case selection?

To identify the most frequent approaches used, we map a bubble chart with the X-axis corresponding to related approaches and Y-axis corresponding to number of papers as in Figure 3. We also presented a pie chart that shows the percentage of approaches published in Figure 4.

From this result we can deduce that model-based and coverage-based approaches have been quite popular since 2011. There are 13 types of approaches researchers often use within 1998–2019. History-based, control-based, fault-based and extraction-based methods can be considered as new approaches since they had been only introduced less than 5 years ago. 24 papers applied model-based approach while its competitor, coverage-based approach, featured 23 papers. Based on Figure 4, 27% of the studies use model-based approach while coverage-based approach is used by 25% of the studies. Combination of the two approaches would cover more than 50% of the total approaches. This evidence indicates that model-based and coverage-based approaches are the most used approaches. Code-based and search-based approaches are worth mentioning since they cover 12% and 11% of the studies, respectively.

These approaches have provided researchers with many solutions depending on their study. We believed that these approaches do not apply to every scenario given. Many researchers used TCS because it can reduce test cases; thus, time and cost will decrease. However, the researchers have to decide on which approach would benefit the most in their experiment.

RQ 1.1 Do the approaches used improve over time?

The line graph in Figure 5 shows the increasing number of publications throughout the years. In answering RQ 1.1, we agreed that the existing approaches improved over the years due to the rising number of papers published. Aside from that, numbers for model-based and coverage-based approaches have been growing since they have been around since 2012. The result from their experiment [11][12][13] also proves that the approaches have improved over time.

RQ 2. Which publication published the most test case selection papers?

Based on Figure 6, 13 publications published papers related to TCS, where each publication published at least 2 papers. There are many existing publications but they only publish one paper related to it. International Conference on Software Testing, Verification and Validation (ICST) and Information and Software Technology (INFSOF) have published the highest number of papers on TCS with 6 papers from 1998 to 2019. INFSOF produce Q2 journals which summarize the quality of the publications. Second best publication is International Conference on Software Engineering (ICSE) and Journal of Systems and Software (JSS) with 5 papers each. JSS produces quality Q1 journals. All the abbreviations in Figure 6 are listed in Appendix.

Based on Figure 7, there are 4 types of publication among the papers collected. Most of the studies were conferences papers and the result shows that conference covers 54% of the total papers. Researchers are likely to publish their works in conferences rather than symposiums or workshops, each recorded only 11% and 5%, respectively. Journal publication covers at least 30% – this is because it tends to have generous page limits compared to conference paper but requires more elaborations on the work. It is proven that papers in well-recognized journals tend to have more prestige than papers in well-recognized conferences.

RQ 3. How frequent is test case selection applied in each software testing?

TCS is popular in regression testing as that was what it was first introduced to do. The technique is proven and can be applied on other testing such as system testing and model-based testing. We distributed the collected papers into related testing as shown in Figure 8.

From the result, we can deduce that TCS is less popular in other testing as we can only collect a total of 23 papers that are not related to regression testing. However, we believed that the numbers will increase due to the popularity in recent years. There are

7 papers which applied TCS to perform general testing. In other words, the researchers want to enhance the efficiency of all types of testing by applying TCS.

RQ 4. What are the case studies commonly used by researchers?

It is true that TCS is widely used by researchers since its introduction. However, we are curious about the types of case studies that use this technique.

Pie chart in Figure 9 shows the case studies commonly used by researchers. The question leads to a lot of case studies but only case studies that used at least 5 times made into the chart. From the pie chart it is clear that the majority of researchers applied case studies from Software-artefacts Infrastructure Repository (SIR) with 43%. Simple program and SourceForge project come second as both have same number of 15% followed by automated teller machine (ATM) with 11%. SIR and SourceForge project are open-source case studies that can be downloaded from [14] and [15]. Simple program is a program that was built by the authors just to make a point that their framework or proposed technique works. It can be a code-based program like finding the maximum number or just a Unified Modeling Language (UML) diagram. The smallest minority goes to power window controller and industry collaboration with only 8%.

In conclusion, since SIR is the most popular choice, it is clear that this technique has greatly contributed to the advancement in TCS field. In the future, we can expect to see more researchers to apply SIR in their studies. We believe that these findings will benefit researchers to utilize various types of case studies available to be used in their research.

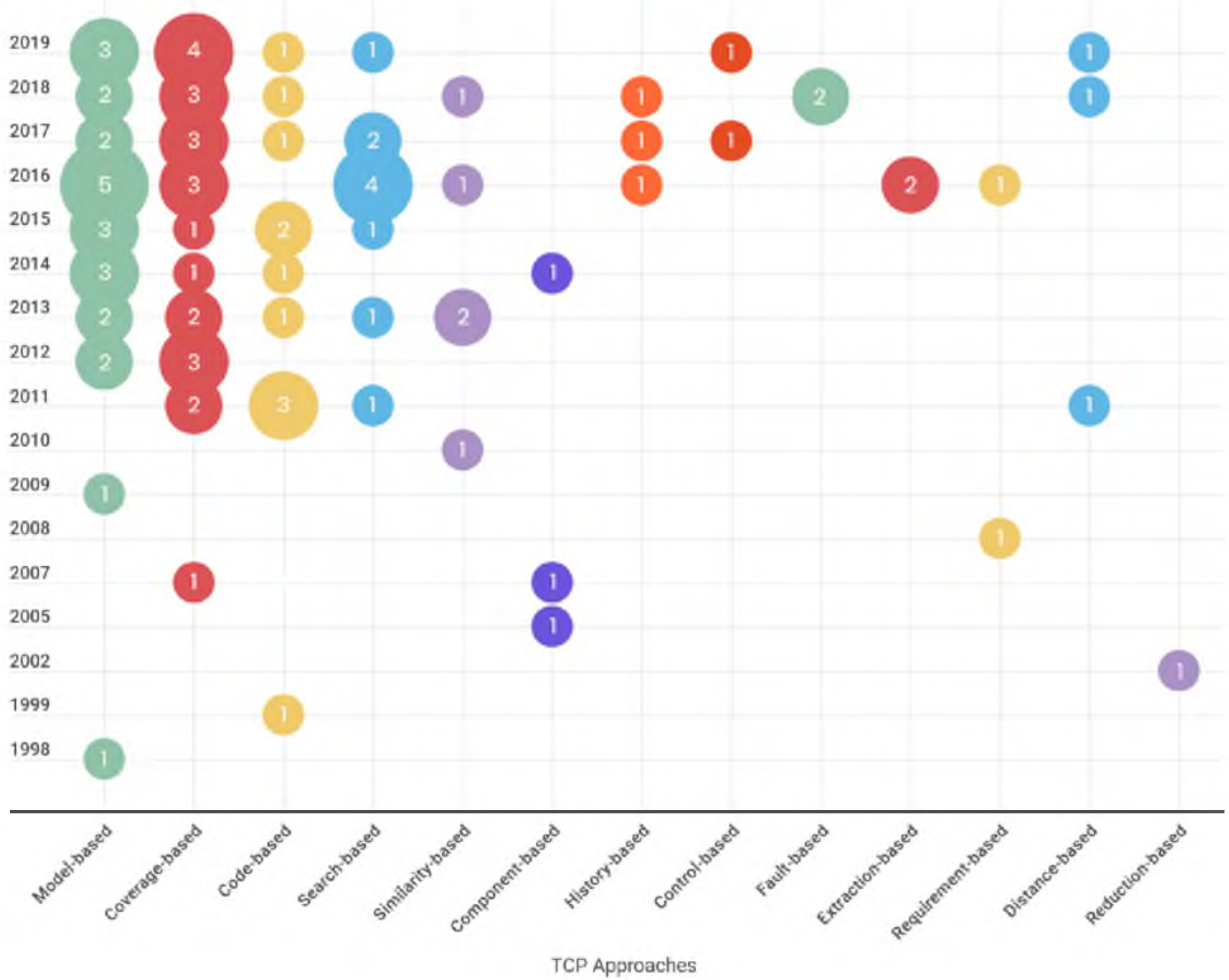

Figure 3. Percentage of collated study approaches

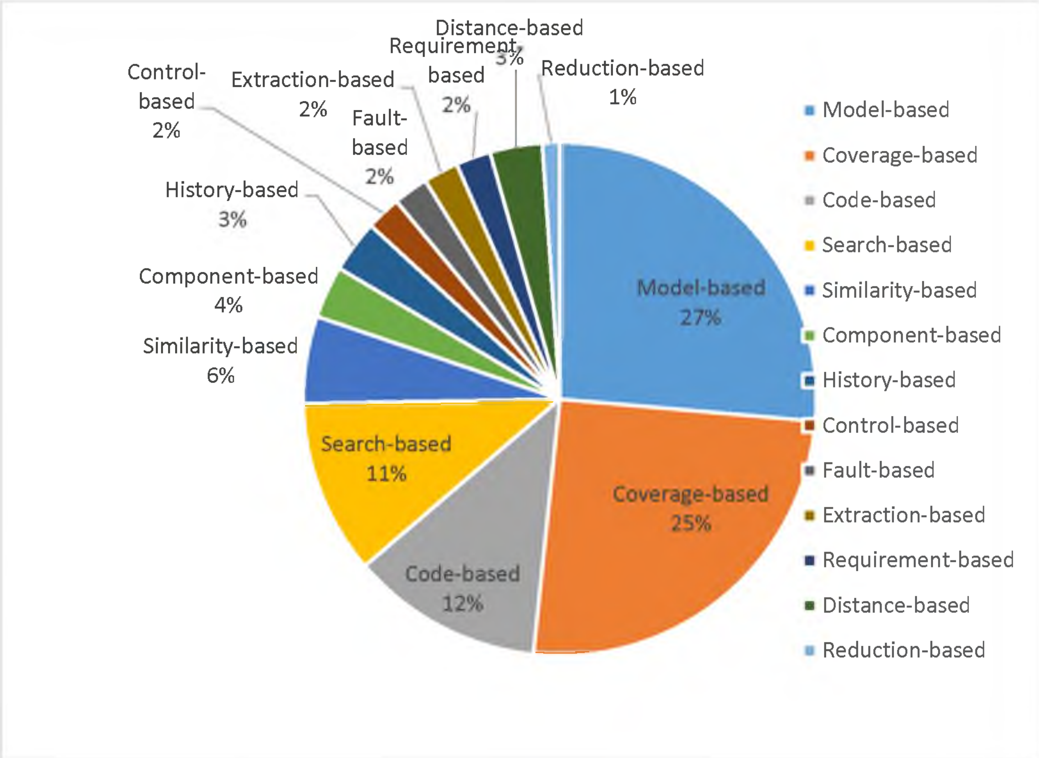

Figure 4. Percentage of collated study approaches.

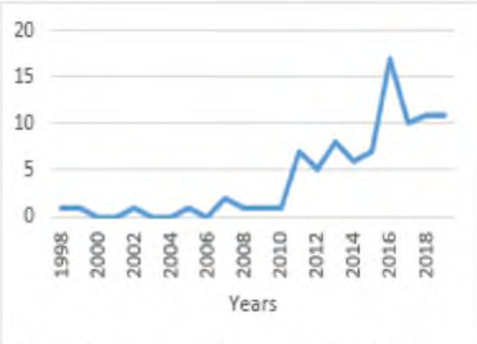

Figure 5. Number of papers published each year.

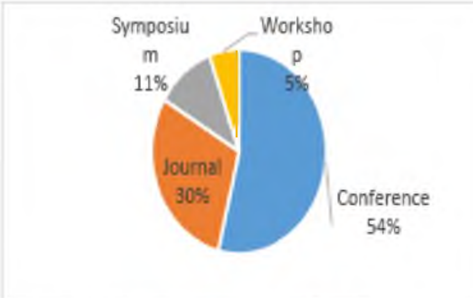

Figure 7. Percentage of publication types.

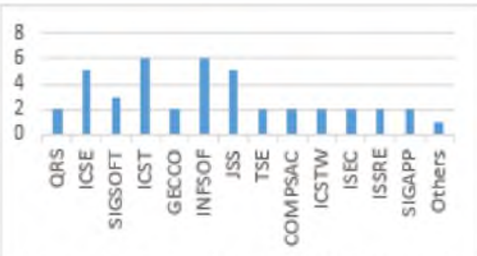

Figure 6. Distribution of publication series

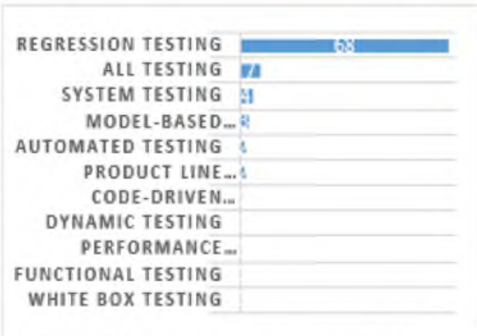

Figure 8. Distribution of software testing application

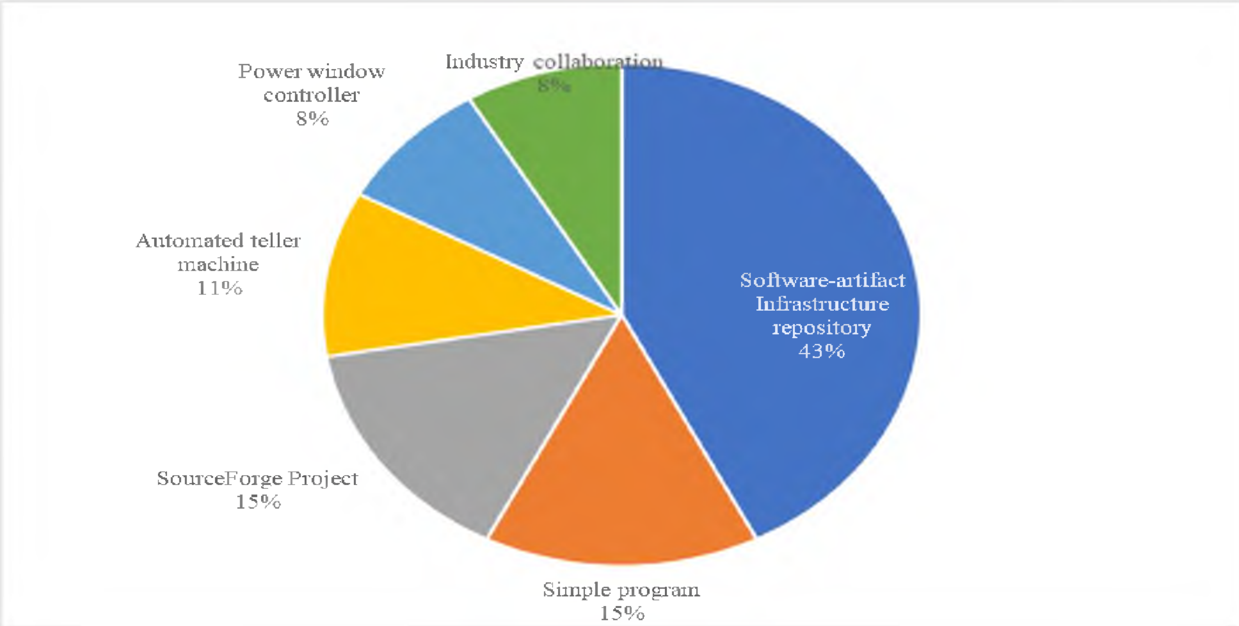

Figure 9. Distribution of case studies.

5. Conclusion and Future Work

This paper presented an overview of existing research related to TCS. Relevant papers on this topic were collected and tabulated. We used systematic mapping to assist in proving our research objectives. Our study revealed that model-based approaches are widely used by researchers until now. It was also revealed that ICST and INFSOF have the highest number of publications in this area. Lastly, the selected papers also proved that TCS is not only applied for regression testing, but also other types of software testing.

This paper shows several potential future works for researchers in this field. The most popular approaches used are likely to be applied in other studies as well. However, we did not discuss on the benefits of the approaches. We believe this gap can be explored in future studies. Thus, a systematic literature review on the approaches will be one of the agenda in our future works.

Appendix

List of Publication Series

Acronym	**Publication Series Title**
QRS	International Conference on Software Security and Reliability
ICSE	International Conference on Software Engineering
SIGSOFT	International Symposium on Foundations of Software Engineering
ICST	International Conference on Software Testing, Verification and Validation
GECCO	Genetic and Evolutionary Computation Conference
INFSOF	Information and Software Technology
JSS	Journal of Systems and Software
TSE	Transactions on Software Engineering
COMPSAC	International Conference on Computer Software & Applications
ICSTW	International Conference on Software Testing, Verification and Validation Workshops
ISEC	India Software Engineering Conference
ISSRE	International Symposium on Software Reliability Engineering
SIGAPP	Symposium On Applied Computing

Acknowledgement

The authors wish to acknowledge Universiti Teknologi Malaysia for grant UTM-TDR Grant Vot No. 06G23, and Ministry of Higher Education (MOHE) for FRGS Grant Vot No. 5F117, which have made this research possible. The authors would also like to express their warmest appreciation for the continuous support and input provided by members of Embedded Real Time Software Engineering Laboratory.

References

Agrawal, A. P., Delhi, N., Choudhary, A., Kaur, A., and Delhi, N. (2020). *An Effective Regression Test Case Selection Using Hybrid Whale Optimization*

Algorithm. *11*(1), 53–67. https://doi.org/10.4018/IJDST.2020010105

Agrawal, A. P., and Kaur, A. (2018). *A Comprehensive Comparison of Ant Colony and Hybrid Particle Swarm Optimization Algorithms Through Test Case Selection*. 397–405. https://doi.org/10.1007/978-981-10-3223-3

Akour, M., and Abuwardih, L. (2018). Test Case Minimization using Genetic Algorithm : Pilot Study. *2018 8th International Conference on Computer Science and Information Technology (CSIT)*, 66–70.

Ali, S., Hafeez, Y., Hussain, S., and Yang, S. (2019). Enhanced regression testing technique for agile software development and continuous integration strategies. *Software Quality Journal*.

Alves, E. L. G., Machado, P. D. L., Massoni, T., and Santos, S. T. C. (2013). A refactoring-based approach for test case selection and prioritization. *2013 8th International Workshop on Automation of Software Test, AST 2013 - Proceedings*, 93–99. https://doi.org/10.1109/IWAST.2013.6595798

Arrieta, A., Wang, S., Markiegi, U., Arruabarrena, A., Etxeberria, L., and Sagardui, G. (2019). Pareto efficient multi-objective black-box test case selection for simulation-based testing. *INFSOF*, *114*(November 2018), 137–154. https://doi.org/10.1016/j.infsof.2019.06.009

Arrieta, A., Wang, S., Sagardui, G., and Etxeberria, L. (2019). Search-Based test case prioritization for simulation-Based testing of cyber-Physical system product lines. *Journal of Systems and Software*, *149*, 1–34. https://doi.org/10.1016/j.jss.2018.09.055

Bach, T., Andrzejak, A., and Pannemans, R. (2017). Coverage-Based Reduction of Test Execution Time : Lessons from a Very Large Industrial Project. *2017 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, *Section III*, 3–12. https://doi.org/10.1109/ICSTW.2017.6

Bach, T., Pannemans, R., and Schwedes, S. (2018). Effects of an economic approach for test case selection and reduction for a large industrial project. *Proceedings - 2018 IEEE 11th International Conference on Software Testing, Verification and Validation Workshops, ICSTW 2018*, 374–379. https://doi.org/10.1109/ICSTW.2018.00076

Bajaj, A., and Sangwan, O. P. (2021). Tri-level regression testing using nature-inspired algorithms. *Innovations in Systems and Software Engineering*, *17*(1), 1–16. https://doi.org/10.1007/s11334-021-00384-9

Banias, O. (2019). Test case selection-prioritization approach based on memoization dynamic programming algorithm. *Information and Software Technology*, *115*(June), 119–130. https://doi.org/10.1016/j.infsof.2019.06.001

Boyce, B. R. (1990). Concepts of information retrieval and automatic text processing: The transformation analysis, and retrieval of information by computer. *Journal of the American Society for Information Science*, *41*(2), 150–151. https://doi.org/10.1002/(sici)1097-4571(199003)41:2<150::aid-asi12>3.0.co;2-8

Byun, T., Sharma, V., Vijayakumar, A., Rayadurgam, S., and Cofer, D. (2019). Input prioritization for testing neural networks. *Proceedings - 2019 IEEE International Conference on Artificial Intelligence Testing, AITest 2019*, 63–70. https://doi.org/10.1109/AITest.2019.000-6

Coutinho, A. E. V. B., Cartaxo, E. G., and Machado, P. D. de L. (2016). Analysis of distance functions for similarity-based test suite reduction in the context of model-based testing. In *Software Quality Journal* (Vol. 24, Issue 2). https://doi.org/10.1007/s11219-014-9265-z

de Oliveira Neto, F. G., Ahmad, A., Leifler, O., Sandahl, K., and Enoiu, E. (2018). Improving continuous integration with similarity-based test case selection. *AST*, 39–45. https://doi.org/10.1145/3194733.3194744

Do, H., and Mirarab, S. (2010). *The Effects of Time Constraints on Test Case Prioritization : A Series of Controlled Experiments*. *36*(5), 593–617.

Doane, D. P., Seward, L. E., Doane, D. P., and Seward, L. E. (2017). *Measuring Skewness : A Forgotten Statistic ? Measuring Skewness : A Forgotten Statistic ? 1898*(December). https://doi.org/10.1080/10691898.2011.11889611

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

Elbaum, S., Malishevsky, A. G., and Rothermel, G. (2002). Test case prioritization: A family of empirical studies. *IEEE Transactions on Software Engineering*, *28*(2), 159–182. https://doi.org/10.1109/32.988497

Engström, E., Runeson, P., and Ljung, A. (2011). Improving regression testing transparency and efficiency with history-based prioritization - An industrial case study. *Proceedings - 4th IEEE International Conference on Software Testing,* *Verification, and Validation, ICST 2011*, 367–376. https://doi.org/10.1109/ICST.2011.27

Engström, E., Runeson, P., and Skoglund, M. (2010). A systematic review on regression test selection techniques. In *Information and Software Technology* (Vol. 52, Issue 1, pp. 14–30). https://doi.org/10.1016/j.infsof.2009.07.001

Fang, C., Chen, Z., Wu, K., and Zhao, Z. (2014). Similarity-based test case prioritization using ordered sequences of program entities. *Software Quality Journal*, *22*(2), 335–361. https://doi.org/10.1007/s11219-013-9224-0

Gokilavani, N., and Bharathi, B. (2021). Microprocessors and Microsystems Multi-Objective based test case selection and prioritization for distributed cloud environment. *Microprocessors and Microsystems*, *82*(January), 103964. https://doi.org/10.1016/j.micpro.2021.103964

Gomma, W. H., and Fahmy, A. A. (2013). A Survey of Text Similarity Approaches. *International Journal of Computer Applications*, *68*(13), 13–18.

Gupta, D., and Gupta, V. (2017). Test Suite Prioritization Using Nature Inspired Meta-Heuristic Algorithms. In A. M. Madureira, A. Abraham, D. Gamboa, and P. Novais (Eds.), *Intelligent Systems Design and Applications* (Vol. 557, pp. 216–226). Springer International Publishing. https://doi.org/10.1007/978-3-319-53480-0_22

Habtemariam, G. M., and Mohapatra, S. K. (2019). A Genetic Algorithm-Based Approach for Test Case Prioritization. In *Communications in Computer and Information Science* (Vol. 1026). Springer International Publishing. https://doi.org/10.1007/978-3-030-26630-1_3

Hall, P. A. V., and Dowling, G. R. (1980). Approximate String Matching. *ACM Computing Surveys (CSUR)*, *12*(4), 381–402. https://doi.org/10.1145/356827.356830

Harman, M. (2011). Making the case for MORTO: Multi objective regression test optimization. *Proceedings - 4th IEEE International Conference on Software Testing, Verification, and Validation Workshops, ICSTW 2011*, 111–114. https://doi.org/10.1109/ICSTW.2011.60

Hettiarachchi, C., Do, H., and Choi, B. (2016). Risk-based test case prioritization using a fuzzy expert system. *Information and Software Technology*, *69*, 1–15. https://doi.org/10.1016/j.infsof.2015.08.008

Jahan, H., Feng, Z., and Mahmud, S. M. H. (2020). Risk-Based Test Case

Prioritization by Correlating System Methods and Their Associated Risks. *Arabian Journal for Science and Engineering*, *45*(8), 6125–6138. https://doi.org/10.1007/s13369-020-04472-z

Jaro, M. A. (1989). Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. *Journal of the American Statistical Association*, *84*(406), 414–420. https://doi.org/10.1080/01621459.1989.10478785

Jiang, B., Zhang, Z., Chan, W. K., Tse, T. H., and Chen, T. Y. (2012). How well does test case prioritization integrate with statistical fault localization? *Information and Software Technology*, *54*(7), 739–758. https://doi.org/10.1016/j.infsof.2012.01.006

Jung, P., Kang, S., and Lee, J. (2019). Automated code-based test selection for software product line regression testing. *JSS*, *158*. https://doi.org/10.1016/j.jss.2019.110419

Kandil, P., Moussa, S., and Badr, N. (2016). *Cluster-based test cases prioritization and selection technique for agile regression testing*. https://doi.org/10.1002/smr

Kapfhammer, G. M. (2011). Empirically evaluating regression testing techniques: Challenges, solutions, and a potential way forward. *Proceedings - 4th IEEE International Conference on Software Testing, Verification, and Validation Workshops, ICSTW 2011*, 99–102. https://doi.org/10.1109/ICSTW.2011.88

Kazmi, R., Jawawi, D. N. A., Mohamad, R., and Ghani, I. (2017). Effective regression test case selection: A systematic literature review. *ACM Computing Surveys*, *50*(2). https://doi.org/10.1145/3057269

Kazmi, S. R. H. S. (2017). *EFFECTIVE REGRESSION TEST CASE SELECTION TECHNIQUE USING WEIGHTED AVERAGE SCORING*. Universiti Teknologi Malaysia.

Khatibsyarbini, M., Isa, M. A., and Jawawi, D. N. A. (2017). A hybrid weight-based and string distances using particle swarm optimization for prioritizing test cases. *Journal of Theoretical and Applied Information Technology*, *95*(12), 2723–2732.

Khatibsyarbini, M., Isa, M. A., Jawawi, D. N. A., Hamed, H. N. A., and Mohamed Suffian, M. D. (2019). Test Case Prioritization Using Firefly Algorithm for Software Testing. *IEEE Access*, *7*, 132360–132373. https://doi.org/10.1109/ACCESS.2019.2940620

Khatibsyarbini, M., Isa, M. A., Jawawi, D. N. A., and Tumeng, R. (2018). Test case prioritization approaches in regression testing: A systematic literature review. *Information and Software Technology*, *93*, 74–93. https://doi.org/10.1016/j.infsof.2017.08.014

Kruskal, W. H., and Wallis, W. A. (1952). *Journal of the American Use of Ranks in One- Criterion Variance Analysis*. *August 2014*, 37–41. https://doi.org/10.1080/01621459.1952.10483441

Ledru, Y., Petrenko, A., Boroday, S., and Mandran, N. (2012). Prioritizing test cases with string distances. *Automated Software Engineering*, *19*(1), 65–95. https://doi.org/10.1007/s10515-011-0093-0

Li, B., and Han, L. (2013). Distance weighted cosine similarity measure for text classification. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *8206 LNCS*, 611–618. https://doi.org/10.1007/978-3-642-41278-3_74

Li, M., Chen, X., Li, X., Ma, B., and Vitányi, P. M. B. (2004). The similarity metric. *IEEE Transactions on Information Theory*, *50*(12), 3250–3264. https://doi.org/10.1109/TIT.2004.838101

Magalhães, C., Andrade, J., Perrusi, L., Mota, A., Barros, F., and Maia, E. (2020). HSP: A hybrid selection and prioritisation of regression test cases based on information retrieval and code coverage applied on an industrial case study. *Journal of Systems and Software*, *159*, 110430. https://doi.org/10.1016/j.jss.2019.110430

Malhotra, R., Kaur, A., and Singh, Y. (2010). *A Regression Test Selection and Prioritization Technique*. *6*(2), 235–252.

Mansour, N., Bahsoon, R., and Baradhi, G. (2001). Empirical comparison of regression test selection algorithms. *Journal of Systems and Software*, *57*(1), 79–90. https://doi.org/10.1016/S0164-1212(00)00119-9

Markiegi, U., Arrieta, A., Etxeberria, L., and Sagardui, G. (2019). Test case selection using structural coverage in software product lines for time-budget constrained scenarios. *SIGAPP*, 2362–2371. https://doi.org/10.1145/3297280.3297512

Mishra, D. B., Panda, N., Mishra, R., and Acharya, A. A. (2019). Total fault exposing potential based test case prioritization using genetic algorithm. *International Journal of Information Technology (Singapore)*, *11*(4), 633–637. https://doi.org/10.1007/s41870-018-0117-0

Misri, A. D. (2021). *Smart Wheelchair System*. Universiti Teknologi Malaysia.

Mittelmark, M. B. (2007). New IUHPE research report series gives graduate students a new avenue to disseminate their research globally. *Promotion & Education*, *14*(1). https://doi.org/10.1177/10253823070140010201

Mondal, D., Hemmati, H., and Durocher, S. (2015). Exploring test suite diversification and code coverage in multi-objective test case selection. *2015 IEEE 8th International Conference on Software Testing, Verification and Validation, ICST 2015 - Proceedings*, 1–10. https://doi.org/10.1109/ICST.2015.7102588

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

Nagar, R., Kumar, A., Kumar, S., and Baghel, A. S. (2014a). Implementing test case selection and reduction techniques using meta-heuristics. *Proceedings of the 5th International Conference on Confluence 2014: The Next Generation Information Technology Summit*, 837–842. https://doi.org/10.1109/CONFLUENCE.2014.6949377

Nagar, R., Kumar, A., Kumar, S., and Baghel, A. S. (2014b). Implementing Test Case Selection and Reduction Techniques using Meta-Heuristics. *2014 5th International Conference - Confluence The Next Generation Information Technology Summit (Confluence)*, 837–842.

Nagar, R., Kumar, A., Singh, G. P., and Kumar, S. (2015). Test case selection and prioritization using cuckoos search algorithm. *2015 1st International Conference on Futuristic Trends in Computational Analysis and Knowledge Management, ABLAZE 2015*, 283–288. https://doi.org/10.1109/ABLAZE.2015.7155012

Narciso, E. N., Delamaro, M. E., and De Lourdes Dos Santos Nunes, F. (2014). Test case selection: A systematic literature review. *International Journal of Software Engineering and Knowledge Engineering*, *24*(4), 653–676. https://doi.org/10.1142/S0218194014500259

Nardo, D. Di, Alshahwan, N., Briand, L., and Labiche, Y. (2015). *Coverage-based regression test case selection , minimization and prioritization : a case study on an industrial system*. *April*, 371–396. https://doi.org/10.1002/stvr

Noguchi, T., and Washizaki, H. (n.d.). *History-Based Test Case Prioritization for*

Black Box Testing on a New Product using Ant Colony Optimization. *1*, 1–2.

Noguchi, T., and Washizaki, H. (2015). History-Based Test Case Prioritization for Black Box Testing using Ant Colony Optimization. *2015 IEEE 8th International Conference on Software Testing, Verification and Validation (ICST)*, 1–2. https://doi.org/10.1109/ICST.2015.7102622

Palma, F., Abdou, T., Bener, A., Maidens, J., and Liu, S. (2018). An improvement to test case failure prediction in the context of test case prioritization. *ACM International Conference Proceeding Series*, 80–89. https://doi.org/10.1145/3273934.3273944

Pan, C., Lu, M., Xu, B., and Gao, H. (2019). *applied sciences An Improved CNN Model for Within-Project Software Defect Prediction*. *1*, 1–28.

Panda, S., and Mohapatra, D. P. (2017). Regression test suite minimization using integer linear programming model. *Software - Practice and Experience*, *47*(11), 1539–1560. https://doi.org/10.1002/spe.2485

Pandey, A., Studies, E., and Banerjee, S. (2018). *Test Suite Minimization in Regression Testing Using Hybrid Approach of ACO and GA*. *9*(3). https://doi.org/10.4018/IJAMC.2018070105

Pang, Y., Xue, X., and Namin, A. S. (2013). Identifying effective test cases through K-means clustering for enhancing regression testing. *Proceedings - 2013 12th International Conference on Machine Learning and Applications, ICMLA 2013*, *2*, 78–83. https://doi.org/10.1109/ICMLA.2013.109

Panigrahi, C. R., and Mall, R. (2010). Model-based regression test case prioritization. *Communications in Computer and Information Science*, *54*(6), 380–385. https://doi.org/10.1007/978-3-642-12035-0_39

Paygude, P. (2020). *Fault Aware Test Case Prioritization in Regression Testing*. *8*(5), 2112–2117.

Qiu, D., Li, B., Ji, S., and Leung, H. (2014). Regression testing of web service: A systematic mapping study. *ACM Computing Surveys*, *47*(2). https://doi.org/10.1145/2631685

Qu, B., Nie, C., and Xu, B. (2008). Test case prioritization for multiple processing queues. *2008 International Symposium on Information Science and Engineering, ISISE 2008*, *2*, 646–649. https://doi.org/10.1109/ISISE.2008.106

Quan Do, Thanh Pham, Wei Liu, and K. R. (2017). *WTEN: An Advanced Coupled Tensor Factorization Strategy for Learning from*. *10569*, 537–552.

https://doi.org/10.1007/978-3-319-68783-4

Rafi, D. M., Moses, K. R. K., Petersen, K., and Mäntylä, M. V. (2012). Benefits and limitations of automated software testing: Systematic literature review and practitioner survey. *2012 7th International Workshop on Automation of Software Test, AST 2012 - Proceedings*, 36–42. https://doi.org/10.1109/IWAST.2012.6228988

Rogstad, E., Briand, L., and Torkar, R. (2013). Test case selection for black-box regression testing of database applications. *INFSOF*, *55*(10), 1781–1795. https://doi.org/10.1016/j.infsof.2013.04.004

Rothermel, G., Harrold, M. J., Ostrin, J., and Hong, C. (1998). Empirical study of the effects of minimization on the fault detection capabilities of test suites. *Conference on Software Maintenance*, 34–43. https://doi.org/10.1109/icsm.1998.738487

Rothermel, G., Untcn, R. H., Chu, C., and Harrold, M. J. (2001). Prioritizing test cases for regression testing. *IEEE Transactions on Software Engineering*, *27*(10), 929–948. https://doi.org/10.1109/32.962562

Saber, T., Delavernhe, F., Papadakis, M., Neill, M. O., and Ventresque, A. (2018). A Hybrid Algorithm for Multi-objective Test Case Selection. *CEC*, 1–8.

Sahak, M. (2018). *Effective similarity based test case prioritization technique for software product lines*. universiti teknologi malaysia.

Sampath, S., Bryce, R., and Memon, A. M. (2013). A uniform representation of hybrid criteria for regression testing. *IEEE Transactions on Software Engineering*, *39*(10), 1326–1344. https://doi.org/10.1109/TSE.2013.16

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

Sapna, P. G., and Mohanty, H. (2010). Automated test scenario selection based on Levenshtein distance. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *5966 LNCS*, 255–266. https://doi.org/10.1007/978-3-642-11659-9_28

Saraswat, P., Singhal, A., and Bansal, A. (2019). A review of test case prioritization and optimization techniques. In *Advances in Intelligent Systems and Computing* (Vol. 731). Springer Singapore. https://doi.org/10.1007/978-981-10-8848-3_48

Shi, A., Yung, T., Gyori, A., and Marinov, D. (2015). *Comparing and Combining Test-Suite Reduction and Regression Test Selection*. 237–247.

Souza, L. S. De, Bastos, R., Prudêncio, C., and Barros, F. A. De. (2015). *Open Access A hybrid particle swarm optimization and harmony search algorithm approach for multi-objective test case selection*. https://doi.org/10.1186/s13173-015-0038-8

Spieker, H., Gotlieb, A., Marijan, D., and Mossige, M. (2017). Reinforcement learning for automatic test case prioritization and selection in continuous integration. *ISSTA 2017 - Proceedings of the 26th ACM SIGSOFT International Symposium on Software Testing and Analysis*, 12–22. https://doi.org/10.1145/3092703.3092709

Su, W., Li, Z., Wang, Z., and Yang, D. (2020). A Meta-heuristic Test Case Prioritization Method Based on Hybrid Model. *Proceedings - 2020 International Conference on Computer Engineering and Application, ICCEA 2020*, 430–435. https://doi.org/10.1109/ICCEA50009.2020.00099

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

Suri, B., and Singhal, S. (2014). Development and validation of an improved test selection and prioritization algorithm based on ACO. *International Journal of Reliability, Quality and Safety Engineering*, *21*(6), 1–13. https://doi.org/10.1142/S0218539314500326

Thada, V., and Jaglan, V. (2013). Comparison of jaccard, dice, cosine similarity coefficient to find best fitness value for web retrieved documents using genetic algorithm. *International Journal of Innovations in Engineering and Technology*, *2*(4), 202–205.

Tilley, S., and Floss, B. (2014). *Hard Problems in Software Testing*. http://www.morganclaypool.com/doi/pdf/10.2200/S00587ED1V01Y201407SWE002

Tyagi, M., and Malhotra, S. (2014). Test case prioritization using multi objective particle swarm optimizer. *2014 International Conference on Signal Propagation and Computer Technology, ICSPCT 2014*, 390–395. https://doi.org/10.1109/ICSPCT.2014.6884931

Wagner, R. A., and Fischer, M. J. (1974). The String-to-String Correction Problem.

Journal of the ACM (JACM), *21*(1), 168–173. https://doi.org/10.1145/321796.321811

Whitley, D. (1994). A genetic algorithm tutorial. *Statistics and Computing*, *4*(2), 65–85. https://doi.org/10.1007/BF00175354

William W Cohen, Pradeep Ravikumar, and Stephen. (2003). A Comparison of String Distance Metrics for. *Proceedings of the IJCAI-2003 Workshop On*, 73--78.

Winkler, W. E. (1999). The State of Record Linkage and Current Research Problems. *Statistical Research Division US Census Bureau*, 1–15. https://doi.org/10.1.1.39.4336

Yan, S., Chen, Z., Zhao, Z., Zhang, C., and Zhou, Y. (2010). A dynamic test cluster sampling strategy by leveraging execution spectra information. *ICST 2010 - 3rd International Conference on Software Testing, Verification and Validation*, 147–154. https://doi.org/10.1109/ICST.2010.47

Yap, B. W. (2011). *Power Comparisons of Shapiro-Wilk , Kolmogorov-Smirnov , Lilliefors and Anderson-Darling Tests. January.*

Yunja, C. (2015). Regression testing minimization, selection and prioritization: a survey. *Software Testing, Verification and Reliability, Volume 21*(Issue 3), 195–214. https://doi.org/10.1002/stvr

Zhang, C., Chen, Z., Zhao, Z., Yan, S., Zhang, J., and Xu, B. (2010). An improved regression test selection technique by clustering execution profiles. *Proceedings - International Conference on Quality Software*, 171–179. https://doi.org/10.1109/QSIC.2010.16

Zhou, Z. Q. (2010). Using Coverage Information to Guide Test Case Selection in Adaptive Random Testing. *2010 IEEE 34th Annual Computer Software and Applications Conference Workshops*, 208–213. https://doi.org/10.1109/COMPSACW.2010.43

Zhou, Z. Q., Sinaga, A., and Susilo, W. (2012). On the fault-detection capabilities of adaptive random test case prioritization: Case studies with large test suites. *Proceedings of the Annual Hawaii International Conference on System Sciences*, 5584–5593. https://doi.org/10.1109/HICSS.2012.454

LIST OF PUBLICATIONS

Index

Kamil, R. M. I., Jawawi, D. N., Mamat, R., & Jamal, N. N. (2020, May). Indoor Smart Wheelchair: Systematic Mapping. In *IOP Conference Series: Materials Science and Engineering* (Vol. 864, No. 1, p. 012070). IOP Publishing.

Non-Index

Kamil, R. M. I., Jawawi, D. N., Mamat, R., & Jamal, N. N. (2020, December). An Enhanced String Algorithm of Prioritizing Technique. In *ICONI'20 Conference Proceedings: ISSN: 2093-0542.*