UNIVERSITI PUTRA MALAYSIA

MULTIFACTOR APPROACH TO PRIORITIZE EVENT SEQUENCE TEST CASES

JOHANNA BINTI AHMAD

FSKTM 2018 83

MULTIFACTOR APPROACH TO PRIORITIZE EVENT SEQUENCE TEST CASES

By

JOHANNA BINTI AHMAD

Thesis Submitted to the School of Graduate Studies, Universiti Putra Malaysia, in Fulfillment of the Requirements for the Degree of Doctor of Philosophy

December 2018

COPYRIGHT

DEDICATION

This thesis is dedicated to my beloved mother, husband, son, daughter and my best friend. Thank you for all the prayers.

Abstract of thesis presented to the Senate of Universiti Putra Malaysia in fulfillment of the requirement for the degree of Doctor of Philosophy

MULTIFACTOR APPROACH TO PRIORITIZE EVENT SEQUENCE TEST CASES

By

JOHANNA BINTI AHMAD

December 2018

Chairman : Salmi Baharom, PhD
Faculty : Computer Science and Information Technology

Software testing consumes 40 to 70 percent of the development effort, time, and cost, especially for large software systems. Test Case Prioritization (TCP) is a method to prioritize and schedule test cases in order to run test cases of higher priority to minimize time, cost, and effort during the software testing phase. Numerous TCP techniques have been proposed with a variety of criteria and evaluation metrics. However, in the context of prioritizing the event sequences test cases, most researchers agree that testing the event sequences test cases is more complex than the single event test cases. Meanwhile, one of the most significant issues in TCP is how researchers handle the same priority value issues. Most researchers either applied the random technique or did not provide any information regarding the same priority value issues in their research. Due to that reason, this research proposed Multifactor Weighted Approach (MFWA) using six factors: complexity, redundancy, frequency, permutations, fault matrix, and distance to improve the existing TCP technique that used random technique to break the ties once the same priority value exists during the prioritization process. The aim of this research is to combine all six factors to produce a unique weight for each test case. The ordered suite will be based on the final test case weight. The test case that has the highest weight will be executed first compared to the others. The mutation testing approach is selected to compare and evaluate the random technique and MFWA technique. Both techniques will execute the same test suites and codes. The effectiveness of both techniques is measured in terms of the capability to detect faults using the Average Percentage Faults Detected (APFD). Meanwhile, the efficiency refers to how quickly each technique is capable of detecting faults by referring to the position of the test case that managed to kill all the live mutants. The Jaccard Distance approach was managed to solve same priority value by measuring the similarities of the data state value among test cases. The new ordering of test case after the implementation of Jaccard Distance approach has been proof can detect faults earlier than others based on the Fault Detection table. In fact, the

empirical results indicated that the MFWA technique is more effective and efficient than the random technique.

Abstrak tesis yang dikemukakan kepada Senat Universiti Putra Malaysia sebagai memenuhi keperluan untuk ijazah Doktor Falsafah

PENDEKATAN PELBAGAI FAKTOR UNTUK MENGUTAMAKAN KES UJIAN BAGI RANGKAIAN ACARA

Oleh

JOHANNA BINTI AHMAD

Disember 2018

Pengerusi : Salmi Baharom, PhD
Fakulti : Sains Komputer dan Teknologi Maklumat

Pengujian perisian menggunakan 40 hingga 70 peratus daripada usaha pembangunan, masa dan kos, terutamanya untuk sistem perisian yang besar. Pengutamaan Kes Ujian (TCP) adalah kaedah untuk mengutamakan dan menjadualkan kes-kes ujian dalam usaha untuk menjalankan kes ujian yang berkeutamaan lebih tinggi bagi meminimumkan masa, kos, dan usaha semasa fasa ujian perisian. Banyak teknik TCP yang telah dicadangkan merangkumi pelbagai kriteria dan metrik penilaian. Walau bagaimanapun, dalam konteks mengutamakan siri acara kes ujian, kebanyakan penyelidik bersetuju bahawa menguji rangkaian acara kes ujian adalah lebih kompleks daripada menguji kes ujian secara tunggal. Sementara itu, salah satu isu yang paling penting dalam TCP adalah bagaimana penyelidik mengendalikan isu nilai keutamaan yang sama. Kebanyakan penyelidik menggunakan samada teknik rawak atau tidak memberikan sebarang maklumat mengenai isu nilai keutamaan yang sama dalam penyelidikan mereka. Disebabkan itu, penyelidikan ini mencadangkan Pendekatan Pelbagai Faktor Pemberat/Bebanan (MFWA) menggunakan enam faktor: kerumitan, lebihan, kekerapan, penggantian/pilihatur, kesalahan matrik dan jarak untuk memperbaiki teknik TCP yang sedia ada yang menggunakan teknik rawak untuk memecahkan ikatan sejurus nilai keutamaan yang sama wujud semasa proses pengutamaan. Tujuan penyelidikan ini adalah untuk menggabungkan semua keenam-enam faktor untuk menghasilkan beban unik untuk setiap kes ujian. Set yang dipesan akan berdasarkan bebanan kes ujian akhir. Kes ujian yang mempunyai bebanan tertinggi akan dilaksanakan terlebih dahulu berbanding yang lain. Pendekatan ujian mutasi dipilih untuk membandingkan dan menilai teknik rawak dan teknik MFWA. Kedua-dua teknik akan melaksanakan set ujian dan kod yang sama. Keberkesanan kedua-dua teknik diukur dari segi keupayaan untuk mengesan kesilapan menggunakan kaedah Peratusan Purata Kesalahan Dikesan (APFD). Sementara itu, kecekapan merujuk kepada seberapa cepat setiap teknik mampu mengesan kesilapan dengan merujuk kepada kedudukan kes ujian yang berjaya mematikan semua mutan hidup.

Pendekatan jarak jauh mampu menyelesaikan isu nilai keutamaan yang sama dengan cara mengukur kesamaan nilai keadaan data di kalangan kes ujian. Susunan baru kes ujian selepas pelaksanaan pendekatan jarak jauh telah terbukti dapat mengesan kesilapan lebih awal daripada yang lain berdasarkan jadual Pengesanan Kesilapan. Keputusan empirikal menunjukkan bahawa teknik MFWA lebih berkesan dan cekap daripada teknik rawak.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my beloved supervisor, Dr. Salmi binti Baharom for her guidance, suggestions, support, and encouragement during the research process. I would also like to express my warm and sincere thanks to my co-supervisors, Professor Dr. Hj. Abdul Azim bin Abd Ghani, Associate Professor Dr. Hazura binti Zulzalil, and Dr. Jamilah binti Din for their invaluable guidance.

I would like to express my appreciation to my loving husband, Hasraf Azad, my mother, Isam, my son, Aiman Hakim, and my daughter, Adelia Akhtar. It would have been impossible for me to finish this study without their understanding and support. Not to forget my best friend, Adilah, for always supporting me whenever I face difficulties during my research, and of course, for praying for my success. Thank you all for the support and prayers throughout these years. May Allah (SWT) bless us all.

Finally, I would like to acknowledge every individual for his or her invaluable help, and cooperation throughout my research.

This thesis was submitted to the Senate of the Universiti Putra Malaysia and has been accepted as fulfillment of the requirement for the degree of Doctor of Philosophy. The members of the Supervisory Committee were as follows:

Salmi Baharom, PhD
Senior Lecturer
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Chairman)

Abdul Azim Abd Ghani, PhD
Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Member)

Hazura Zulzalil, PhD
Associate Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Member)

Jamilah Din, PhD
Senior Lecturer
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Member)

ROBIAH BINTI YUNUS, PhD
Professor and Dean
School of Graduate Studies
Universiti Putra Malaysia

Date:

Declaration by graduate student

I hereby confirm that:

- this thesis is my original work;
- quotations, illustrations and citations have been duly referenced;
- this thesis has not been submitted previously or concurrently for any other degree at any institutions;
- intellectual property from the thesis and copyright of thesis are fully-owned by Universiti Putra Malaysia, as according to the Universiti Putra Malaysia (Research) Rules 2012;
- written permission must be obtained from supervisor and the office of Deputy Vice-Chancellor (Research and innovation) before thesis is published (in the form of written, printed or in electronic form) including books, journals, modules, proceedings, popular writings, seminar papers, manuscripts, posters, reports, lecture notes, learning modules or any other materials as stated in the Universiti Putra Malaysia (Research) Rules 2012;
- there is no plagiarism or data falsification/fabrication in the thesis, and scholarly integrity is upheld as according to the Universiti Putra Malaysia (Graduate Studies) Rules 2003 (Revision 2012-2013) and the Universiti Putra Malaysia (Research) Rules 2012. The thesis has undergone plagiarism detection software

Signature: _____ Date: _____

Name and Matric No: Johanna Binti Ahmad, GS43485

Declaration by Members of Supervisory Committee

This is to confirm that:

- the research conducted and the writing of this thesis was under our supervision;
- supervision responsibilities as stated in the Universiti Putra Malaysia (Graduate Studies) Rules 2003 (Revision 2012-2013) were adhered to.

Signature: _____
Name of Chairman of Supervisory Committee: Dr. Salmi Baharom

Signature: _____
Name of Member of Supervisory Committee: Professor Dr. Abdul Azim Abd Ghani

Signature: _____
Name of Member of Supervisory Committee: Associate Professor Dr. Hazura Zulzalil

Signature: _____
Name of Member of Supervisory Committee: Dr. Jamilah Din

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

APFD	Average Percentage Fault Detected
TCP	Test Case Prioritization
SLR	Systematic Literature Review
IFC	Information Flow Complexity
FP	Functions Point
UCM	Unique Complexity Metric
LOC	Lines of Codes
Mc Cabe	Mc Cabe Complexity
BCS	Basic Control Structure
$DWtc_j$	Dissimilarity weight of test case
No of ds_{tcj}	Number of data state in a test case
No of redundant ds_{tcj}	Number of redundant data state in a test case
$DWts_j$	Dissimilarity weight within test suite
No of ds_{tsj}	Number of data state within test suite
No of redundant ds_{tsj}	Number of redundant data state within test suite
No of $non-redundant\ ds_{tsj}$	Number of non-redundant date state within test suite
CA	Covering array
SCA	Sequence covering array
GUI	Graphical User Interface
AST	Abstract Syntax Trees
LOC	Line of Code
POI	Poor Obfuscation Implementation
MFWA	Multifactor Weighted Approach

CHAPTER 1

INTRODUCTION

Software testing is known as a crucial phase in software development life cycle. In general, the objective of software testing is to provide information about the quality of a product or service to the stakeholders or customers. Software testing involves the process of executing a programme with the intention of discovering as much error as possible in the estimated time. As the size of a system becomes larger, the complexity of the system will also increase. Hence, the significance of the testing phase will also increase due to its importance in revealing potential faults. As a result, some projects would perform exhaustive testing to ensure the quality of the system and to maintain the quality once changes or modifications are made. The exhaustive testing, however, requires for a great deal of cost, effort, and time.

In the last several decades, numerous techniques have been proposed in reducing the cost, effort, and time of the testing phase. Most software testing researches would discuss the issue on the generation of test cases. The research in software testing can be categorized into test case generation, test oracle generation, and test case execution. Test case generation is the most dominant test compared to test oracle generation and test case execution. Test case prioritization (TCP) is one of the techniques in test case generation that is used in reducing the time and cost of testing. TCP technique can be adopted either in the regression or non-regression testing.

1.1 Research Motivation

For a large commercial system, the test suites can contain thousands of test cases and may sometimes be infinite. The issue of the large number of test cases was reported by Rothermel et al. (1999), where one of the industrial collaborators reported that they had to run 200,000 lines of codes for seven weeks. Numerous techniques have been studied to cater the large number of test cases issues: test case minimization, test case selection, and TCP (Yoo & Harman, 2007). A number of TCP techniques have been proposed since 1997 to detect the defects as early as possible, aside from reducing time and cost (Silva Ouriques et al., 2015). With the TCP technique, the more important test case will be executed earlier (Yu & Lau, 2012).

The TCP technique is widely used for single-event test cases, however, only a few studies have focused on event sequence test cases (He & Bai, 2015; Huang et al., 2010; Bryce et al., 2011). This is because event sequences are difficult to handle, whereby there is a possibility of the test cases having a combination of events, with a large input sequence, as well as a large amount of test cases that have considerable degrees of redundancy (Huang et al., 2010; Bryce et al., 2011; Silva Ouriques et al., 2015). Harman (2007) addressed the complexity of testing event sequences due to the large test space, different positions of the events, and various permutations of inputs.

Generally, event sequence test cases have enormous numbers of states and every state should be tested. Changes in one internal data state to another should be taken under consideration since they involve interactions between events (Kumar, 2016).

Throughout the years a huge number of TCP techniques have been proposed with varieties of combinations factors to improve the effectiveness of the test case generation (Tonella et al., 2006; Huang et al., 2010; Jena et al., 2015; He & Bai, 2015; Ammar et al., 2016). However, some of these researches failed to prioritize multiple test suites and test cases with the same priority value (Huang et al., 2010; Ammar et al., 2016). The random technique will be applied once the same priority value exists during the prioritization process. Specifically, the existing works failed to cater to the same priority value for the event sequence test cases. In addition to the general limitations and gaps in prioritizing event sequence test cases as previously mentioned, it also shows that the TCP technique has not been extensively explored for this purpose.

The basic motivation behind the limitations and gaps that exist in the existing work is to produce a unique weight approach to avoid the random technique. This, consequently, can increase the effectiveness of the TCP technique. The random technique is ineffective and can create bias issues (Tonella et al., 2006; Ammar et al., 2016). Furthermore, the random technique can produce a large amount of testing data to be executed, which would result in the increase of time, effort, and cost during the testing phase, as well as causing some of the important test cases to be missed out for during testing.

The issues that exist in the previous TCP technique provide the fundamental motivation in improving the effectiveness and efficiency of the existing TCP technique. Therefore, the focus is on software testing, to determine whether the proposed approach is able to maintain and focus in terms of data testing, and its ability to detect faults earlier.

1.2 Problem Statement

Prior to breakthroughs in 1997, people would randomly generate test cases and select the best test cases based on the results without knowing the quality of the test cases. For this reason, numerous TCP techniques have been proposed to prioritize test cases especially for the large commercial system. There are still a number of important issues neglected by previous researchers. To date, existing TCP technique prioritize the test cases in terms of the basic blocks, number of lines of codes, number of covered methods or execution history on a previous version (Sebastian Elbaum et al., 2004; Mirarab & Tahvildari, 2007; Zhang et al., 2013; Ammar et al., 2016).

Some TCP techniques order the test case execution in accordance with their degree of importance, which is indirectly defined. A weighted approach was proposed as an effective predictor to select the best test case ordering (Sebastian Elbaum et al., 2014). A test case priority is based on the weight given to a test case during the prioritization process (Srivastava et al., 2009; Panichella et al., 2014). The weight of the test case depends on the prioritization algorithm. For example, if the algorithm is based on the number of executed statements, then any test case that has the highest number of executed statements will have higher priority than other test cases.

However, most researchers have reported that the proposed algorithms would end up with the random technique (Lin & Huang, 2009; Bryce et al., 2011). Random technique has been applied to break ties. In other word, break ties mean if more than one test case have the same priority value. Nonetheless, it has been proven as an ineffective choice because there is no guarantee that the chosen technique is the best compared to other choices (He & Bai, 2015; Ammar et al., 2016). The random technique may uncover faults that were missed during the development phase.

For example, a test suite consists of three test cases with different combinations of event sequences: for TC1, a combination of event C and event A; TC2 is a combination of event B, event C, and event A; while TC3 consists of combinations of event B, event B, event A, and event A. Assume that the weight value of each event is as follows: event A = 1, event B = 2, event C = 3. Finally, the summation weight of TC1 is 4, TC2 is 6, and TC3 is 6. The weight values show that TC2 and TC3 have the same priority value. This leads to an important question, *"Which test case should be executed earlier, TC2 or TC3?"* The effectiveness and efficiency in detecting faults is not the same, even if the summation weight is equal (Huang et al., 2010). Therefore, this study is interested in answering this question by proposing an approach that produces a unique weight value for each test case during the prioritization process.

One of the most significant current discussions is in terms of the combination of factors that can influence the effectiveness and efficiency of the TCP technique (Srikanth & Cohen, 2011; Khalilian et al., 2012). Previous researchers believe that these factors are capable of improving the rate of failure detection, as well as the ability to solve the same priority value issue (Chaudhary et al., 2014; Fang et al., 2014; Frolin S. Ocariza et al., 2015). For example, if TC1 and TC2 have the same priority value for branch coverage, each test case has the possibility to exercise different coverage (Khalilian et al., 2012). Furthermore, there is no indicator to show the similarity or differences between test cases that have the same priority value.

Assume that the number of *if-else* conditions in TC1 is greater than in TC2. From the perspective of the complexity measurement theory, it is extremely doubtful to assume that both correspond to the same complexity measurement. Thus, an indicator that could detect between similarities and differences needs to be defined to cater to this issue. Ammar et al. (2016) found that whenever two or more test cases have equal weight, they would mostly cover the same segment of codes. Based on that finding,

they arranged the test cases using the initial orders. Hence, without strong justifications based on findings, as reported by Ammar et al. (2016), we cannot simply assume that all test cases with the same priority value cover the same faults during testing. However, the current issue that this study aims to address is how to define an indicator to prove that test cases with the same priority value have different fault coverage.

Based on the Systematic Literature Review (SLR) evaluation which has been conducted, most of the previous researchers either applied random technique or did not provide any information regarding the same priority value issues in their research. Out of 50 primary studies, 98 percent applied random technique in case if two or more test cases have same priority value during the prioritization process. Detail of the SLR analysis is in Appendix A. Those limitations been a burgeoning interest for this research to propose a unique weight approach to prioritize event sequences test cases.

1.3 Research Objectives

This research objectives that are addressed in this thesis are as follows:

i. To identify and analyze issues of the test case prioritization technique for event sequences test cases.
ii. To propose an approach based on unique weight for prioritizing the event sequence test cases.
iii. To automate the proposed approach by providing supporting tool.
iv. To empirically evaluate and measure the effectiveness and efficiency of the proposed approach. The effectiveness is measured in terms of its capability to detect faults early, while the efficiency is based on the prioritization ranking.

1.4 Research Scope

Numerous TCP techniques have been proposed by many researchers. Most of the proposed techniques use one or more factors to determine the priority value for test cases. Our research proposed a TCP technique for both regression and non-regression testing since no historical data is required for calculating the priority value. However, this research focuses on testing a module that consist of several functions sharing a private data structure (i.e. modules that have memory). For such components, their test cases is in the form of sequence of events, and not a single event. Therefore, the proposed TCP technique is specifically developed to determine the priority value of event sequences test cases.

1.5 Research Contributions

This research aimed to enhance the TCP technique based on the gaps left by previous work in this field. Based on the limitations and gaps, this reseach has made the following contributions:

i. It has defined the issues that have been left out by the existing works in prioritizing the event sequence test cases that occur during the prioritization process (C. Y. Huang et al., 2010; Roongruangsuwan & Daengdej, 2010).
ii. It implemented a tool support for automating the prioritization process suggested by MFWA technique. The MFWA tool can be applied in any related experimental work to test other test cases capability in detecting faults.
iii. The empirical result was gained based on the comparisons and evaluations made between the random technique and the MFWA technique. The measurement used the same test suite and codes, but different approaches. The effectiveness of the proposed approach was measured in terms of its capability to detect faults earlier than other techniques. Meanwhile the efficiency was measured in terms of how quickly each technique is capable of detecting faults by referring to the position of the test case that capable to killed mutants.

1.6 Thesis Organization

This thesis comprises of seven chapters. The first chapter consists of the introduction of this research. Hence, the research problem, the objective and scope, as well as the contributions of this research are also described in the first chapter.

The second chapter presents the detailed study of the existing TCP technique. The research background, which consisted of the issues that have been left out by the existing works, are discussed in detail in this chapter. Apart from that, two key areas are focused on, whereby this research looked into how previous studies handled the same issue of priority value and how they applied the proposed approach in the event sequence test cases. The knowledge gaps left by previous works are also highlighted in this chapter.

The third chapter presents the methodology involved in this research. The research methods, materials or resources, and the deliverables obtained throughout the phases are explained in this chapter. Generally, this chapter highlights the three phases that have been previously defined, namely, the definition of the analysis and problems, the design and development of the tool to support the proposed approach, and finally, the evaluation phase involved in producing the empirical result.

The fourth chapter describes the conceptual design of the MFWA technique using six factors. Details of the reasons for the selection of the six factors and how each of the factors produced the weight to be used in the last stage of the prioritization process are explained in this chapter.

The fifth chapter describes the implementation of the MFWA technique, which involved six factors. This chapter discusses in detail how each factor was processed during the tool execution phase.

The sixth chapter presents the experimental procedure and the results of the experiments using the MFWA technique. It provides the statistical analysis using the Paired T-Test statistical model.

The final chapter is Chapter Seven, which consists of the conclusion and future works for this research. Some suggestions for future work that can be investigated by future researchers are explained in this chapter.

REFERENCES

Aboutaleb, H., & Monsuez, B. (2015). Measuring the complexity of a higraph-based system model: Formalism and metrics. *Procedia Computer Science*, *44*(C), 11–20. https://doi.org/10.1016/j.procs.2015.03.057

Abreu, R., Zoeteweij, P., & Van Gemund, A. J. C. (2009). Spectrum-based multiple fault localization. *ASE2009 - 24th IEEE/ACM International Conference on Automated Software Engineering*, 88–99. https://doi.org/10.1109/ASE.2009.25

Aggarwal, K. K., Singh, Y., & Kaur, A. (2005). A multiple parameter test case prioritization model. *Journal of Statistics and Management Systems*, *8*(2), 369–386. https://doi.org/10.1080/09720510.2005.10701165

Ahmed, B. S. (2015). Test case minimization approach using fault detection and combinatorial optimization techniques for configuration-aware structural testing. *Engineering Science and Technology, an International Journal*, *19*(2), 737–753. https://doi.org/10.1016/j.jestch.2015.11.006

Ahmed, B. S., & Zamli, K. Z. (2011). A variable strength interaction test suites generation strategy using Particle Swarm Optimization. *Journal of Systems and Software*, *84*(12), 2171–2185. https://doi.org/10.1016/j.jss.2011.06.004

Albrecht, A., & Jr, J. G. (1983). Software function, source lines of code, and development effort prediction: a software science validation. *Software Engineering, IEEE …*, (6).

Ammar, A., Baharom, S., Ghani, A. A. A., & Din, J. (2016). Enhanced Weighted Method for Test Case Prioritization in Regression Testing Using Unique Priority Value. In *Information Science and Security (ICISS), 2016 International Conference*. https://doi.org/10.1109/ICISSEC.2016.7885851

Andrews, J. H., Briand, L. C., & Labiche, Y. (2005). Is mutation an appropriate tool for testing experiments? *Proceedings of the 27th International Conference on Software Engineering - ICSE '05*, 402. https://doi.org/10.1145/1062455.1062530

Ansari, A., Khan, A., Khan, A., & Mukadam, K. (2016). Optimized Regression Test Using Test Case Prioritization. *Procedia Computer Science*, *79*, 152–160. https://doi.org/10.1016/j.procs.2016.03.020

Arthi, B., Selvarani, A. G., & Sathya, A. (2016). Software Complexity Measure from Software Requirements Document and Cost Driver Factors, *24*, 6–12. https://doi.org/10.5829/idosi.mejsr.2016.24.IIECS.23132

Azizi, M., & Do, H. (2018). A Collaborative Filtering Recommender System for Test Case Prioritization in Web Applications, (July 2017). https://doi.org/10.1145/3167132.3167299

Baharom, S., & Shukur, Z. (2011). An experimental assessment of module documentation-based testing. *Information and Software Technology*, *53*(7), 747–760. https://doi.org/10.1016/j.infsof.2011.01.005

Belli, F., Eminov, M., & Gokce, N. (2009). A Comparative Soft-Computing Approach and Case Studies. In *32nd Annual German Conference on Advances in Artificial Intelligence (KI'09),* (pp. 425–432).

Belli, F., Eminov, M., Gökçe, N., & Wong, W. eri. (2007). Chapter 1 Process— An Adaptive-Learning-Based. In *Adaptive Control Approach for Software Quality Improvement*.

Beyer, D., & Häring, P. (2014). A formal evaluation of DepDegree based on weyuker's properties. *Proceedings of the 22nd International Conference on Program Comprehension - ICPC 2014*, 258–261. https://doi.org/10.1145/2597008.2597794

Bian, Y., Kirbas, S., Harman, M., Jia, Y., & Li, Z. (2015). Regression Test Case Prioritisation for Guava. In *Springer International Publishing Switzerland 2015* (Vol. 9275, pp. 221–227). https://doi.org/10.1007/978-3-319-22183-0

Blanco, R., García-Fanjul, J., & Tuya, J. (2009). A First Approach to Test Case Generation for BPEL Compositions of Web Services Using Scatter Search. *2009 International Conference on Software Testing, Verification, and Validation Workshops*, 131–140. https://doi.org/10.1109/ICSTW.2009.24

Bradbury, J. S. (2007). *Using Program Mutation for the Empirical Assessment of Fault Detection Techniques: A Comparison of Concurrency Testing and Model Checking*. *PhD Dissertation*.

Bradbury, J. S., Cordy, J. R., & Dingel, J. (2005). An Empirical Framework for Comparing Effectiveness of Testing and Property-Based Formal Analysis. *Proc. of the 6th Int. ACM SIGPLAN-SIGSOFT Work. on Program Analysis for Software Tools and Engineering (PASTE 2005)*, 2–5. https://doi.org/10.1145/1108768.1108795

Briand, L., Labiche, Y., & Chen, K. (2013). A Multi-objective Genetic Algorithm to Rank State-Based Test Cases, 66–80. https://doi.org/10.1007/978-3-642-39742-4_7

Brucker, A. D., & Julliand, J. (2014). Editorial: Editorial for the special issue of STVR on tests and proofs volume 2: Tests and proofs for improving the generation time and quality of test data suites. *Software Testing Verification and Reliability*, *24*(8), 591–592. https://doi.org/10.1002/stvr

Bryce, C., Sampath, S., Society, I. C., Memon, A. M., & Society, I. C. (2011). Developing a Single Model and Test Prioritization Strategies for Event-Driven Software, *37*(1), 48–64.

Bryce, R. C., & Memon, A. M. (2007). Test suite prioritization by interaction coverage. *Workshop on Domain Specific Approaches to Software Test Automation in Conjunction with the 6th ESEC/FSE Joint Meeting - DOSTA '07*, 1–7. https://doi.org/10.1145/1294921.1294922

Bryce, R. C., Sampath, S., & Memon, A. M. (2011). Developing a single model and test prioritization strategies for event-driven software. *IEEE Transactions on Software Engineering*, *37*(1), 48–64. https://doi.org/10.1109/TSE.2010.12

Bryce, R. C., Sampath, S., Pedersen, J. B., & Manchester, S. (2011). Test suite prioritization by cost-based combinatorial interaction coverage. *Int. J. Systems Assurance Engineering and Management*, *2*(2), 126–134. https://doi.org/10.1007/s13198-011-0067-4

Cartaxo, E. G., Machado, D. L., & Neto, F. G. O. (2011). On the use of a similarity function for test case selection in the context of model-based testing, (July 2009), 75–100. https://doi.org/10.1002/stvr

Ceylan, B., & Inceoğlu, M. M. (2008). Computational Science and Its Applications – ICCSA 2008. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 5072*(PART 1), 541–554. https://doi.org/10.1007/978-3-540-69839-5

Chaudhary, N., Sangwan, O. P., & Arora, R. (2014). Event-coverage and weight based method for test suite prioritization. *International Journal of Information Technology and Computer Science (IJITCS)*, *6*(12), 61. https://doi.org/10.5815/ijitcs.2014.12.08

Chaudhury, S., Singhal, A., & Sangwan, O. P. (2016). Event- Driven Software Testing – An Overview. *International Journal of Advanced Research in Computer Engineering and Technology*, *5*(4), 1189–1193.

Chen, L., Wang, Z., Xu, L., Lu, H., & Xu, B. (2010). Test case prioritization for web service regression testing. *Proceedings - 5th IEEE International Symposium on Service-Oriented System Engineering, SOSE 2010*, 173–178. https://doi.org/10.1109/SOSE.2010.27

Choe, Y., Jong, C., & Han, S. (2013). Software Cognitive Information Measure based on Relation Between Structures.

Cohen, M. B., Gibbons, P. B., Mugridge, W. B., & Colbourn, C. J. (2003). Constructing Test Suites for Interaction Testing. *Software Engineering, 2003. Proceedings. 25th International Conference On*. https://doi.org/10.1109/ICSE.2003.1201186

Conrad, A. P., Roos, R. S., & Kapfhammer, G. M. (2010). Empirically studying the role of selection operators duringsearch-based test suite prioritization. *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation - GECCO '10*, 1373. https://doi.org/10.1145/1830483.1830735

Coutinho, B., Gadelha, E., & Duarte, P. (2014). Analysis of distance functions for similarity-based test suite reduction in the context of model-based testing. *Software Quality Journal*. https://doi.org/10.1007/s11219-014-9265-z

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

Dario Di Nucci, Panichella, A., Zaidman, A., & Lucia, A. De. (2015). Hypervolume-Based Search for Test Case Prioritization. *Springer International Publishing Switzerland 2015*, *9275*, 157–172. https://doi.org/10.1007/978-3-319-22183-0

DeMillo, R., Lipton, R., & Sayad, F. (1987). Hints on Test Data Selection: Help for the Practising Programmer. *IEEE Computer*, *11*(4), 34–41.

Do, H., & Rothermel, G. (2005a). A controlled experiment assessing test case prioritization techniques via mutation faults. *IEEE International Conference on Software Maintenance, ICSM*, *2005*, 411–420. https://doi.org/10.1109/ICSM.2005.9

Do, H., & Rothermel, G. (2005b). A controlled experiment assessing test case prioritization techniques via mutation faults. *IEEE International Conference on Software Maintenance, ICSM*, *2005*, 411–420. https://doi.org/10.1109/ICSM.2005.9

Do, H., Rothermel, G., & Kinneer, A. (2006). Prioritizing JUnit Test Cases : An Empirical Assessment and Cost-Benefits Analysis, 33–70. https://doi.org/10.1007/s10664-006-5965-8

Dubois, C., Fazlalizadeh, Y., Khalilian, A., Abdollahi Azgomi, M., & Parsa, S. (2009). Incorporating Historical Test Case Performance Data and Resource Constraints into Test Case Prioritization. *Lecture Notes in Computer Science*, *5668*, 43–57. https://doi.org/10.1007/978-3-642-02949-3

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

Elbaum, S., Malishevsky, A. G., & Rothermel, G. (2001). Prioritizing Test Cases For Regression Testing Prioritizing Test Cases For Regression Testing, *27*(10), 929–948.

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

Elbaum, S., Malishevsky, A., & Rothermel, G. (2001). Incorporating varying test costs and fault severities into test case prioritization. *Proceedings of the 23rd International Conference on Software Engineering. ICSE 2001*, 329–338. https://doi.org/10.1109/ICSE.2001.919106

Elbaum, S., Rothermel, G., Elbaum, S., Malishevsky, A. G., & Member, S. (2002). Test Case Prioritization : A Family of Empirical Studies Test Case Prioritization : A Family of Empirical Studies.

Elbaum, S., Rothermel, G., & Kanduri, S. (2004). Selecting a Cost-Effective Test Case Prioritization. *Test*, 185–210.

Elbaum, S., Rothermel, G., & Penix, J. (2014). Techniques for improving regression testing in continuous integration development environments. *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering - FSE 2014*, 235–245. https://doi.org/10.1145/2635868.2635910

Epitropakis, M. G., Yoo, S., Harman, M., & Burke, E. K. (2015). Empirical evaluation of pareto efficient multi-objective regression test case prioritisation. *Proceedings of the 2015 International Symposium on Software Testing and Analysis - ISSTA 2015*, 234–245. https://doi.org/10.1145/2771783.2771788

Fang, C., Chen, Z., Wu, K., & Zhao, Z. (2013). Similarity-based test case prioritization using ordered sequences of program entities. *Software Quality Journal*, 1–27. https://doi.org/10.1007/s11219-013-9224-0

Fang, C., Chen, Z., Wu, K., & 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

Frolin S. Ocariza, J., Li, G., Pattabiraman, K., & Mesbah, A. (2015). Automatic fault localization for client-side JavaScript. *Software Testing, Verification and Reliability*, *Volume 21*(Issue 3), 195–214. https://doi.org/10.1002/stvr

Garg, D. (2013). Parallel Execution of Prioritized Test Cases for Regression Testing of Web Applications, (Acsc), 61–68.

Gautam, & Rahul. (2014). Search based software testing with genetic using fitness function. In *Conference, International Applications, Innovative Intelligence, Computational* (pp. 159–163). https://doi.org/10.1109/CIPECH.2014.7019065

Ghai, S., & Kaur, S. (2017). A Hill-Climbing Approach for Test Case Prioritization, *11*(3), 13–20.

Gupta, S., Raperia, H., Kapur, E., Singh, H., & Kumar, A. (2012). A NOVEL APPROACH FOR TEST CASE. *International Journal of Computer Science, Engineering and Applications*, *2*(3), 53–60.

Hao, D., Zhao, X., & Zhang, L. (2013). Adaptive Test-Case Prioritization Guided by Output Inspection. *2013 IEEE 37th Annual Computer Software and Applications Conference*, 169–179. https://doi.org/10.1109/COMPSAC.2013.31

Harikarthik, S. K., & Palanisamy, V. (2014). Improving Quality of Software Testing Process by Test Case Prioritization, 245–248.

Harman, M. (2006). An Evaluation of Mutation Operators for Equivalent Mutants, (September), 1–99.

Harman, S. M. (2007). Test Data Generation For Event Sequence Graph by.

He, Z., & Bai, C.-G. (2015). GUI Test Case Prioritization by State-coverage Criterion. *2015 IEEE/ACM 10th International Workshop on Automation of Software Test*. https://doi.org/10.1109/AST.2015.11

Hettiarachchi, C., Do, H., & 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

Hsu, H., & Orso, A. (2009). MINTS : A General Framework and Tool for Supporting Test-suite Minimization. In *2009 31st International Conference on Software Engineering (ICSE 2009)* (pp. 419–429). https://doi.org/http://doi.ieeecomputersociety.org/10.1109/ICSE.2009.5070541

Huang, C.-Y., Chen, C.-S., & Lai, C.-E. (2016). Evaluation and analysis of incorporating Fuzzy Expert System approach into test suite reduction. *Information and Software Technology*, *79*, 79–105. https://doi.org/http://dx.doi.org/10.1016/j.infsof.2016.07.005

Huang, C. Y., Chang, J. R., & Chang, Y. H. (2010). Design and analysis of GUI test-case prioritization using weight-based methods. *Journal of Systems and Software*, *83*(4), 646–659. https://doi.org/10.1016/j.jss.2009.11.703

Huang, R., Chen, J., Towey, D., Chan, A. T. S., & Lu, Y. (2015). Aggregate-strength interaction test suite prioritization. *Journal of Systems and Software*, *99*, 36–51. https://doi.org/10.1016/j.jss.2014.09.002

Huang, R., Chen, J., Zhang, T., Wang, R., & Lu, Y. (2013). Prioritizing Variable-Strength Covering Array. *2013 IEEE 37th Annual Computer Software and Applications Conference*, 8–11. https://doi.org/10.1109/COMPSAC.2013.84

Huang, Y.-C., Huang, C.-Y., Chang, J.-R., & Chen, T.-Y. (2010). Design and Analysis of Cost-Cognizant Test Case Prioritization Using Genetic Algorithm with Test History. *2010 IEEE 34th Annual Computer Software and Applications Conference*, 413–418. https://doi.org/10.1109/COMPSAC.2010.66

Huang, Y.-C., Peng, K.-L., & Huang, C.-Y. (2012). A history-based cost-cognizant test case prioritization technique in regression testing. *Journal of Systems and Software*, *85*(3), 626–637. https://doi.org/10.1016/j.jss.2011.09.063

Indumathi, C. P., & Selvamani, K. (2015). Test Cases Prioritization Using Open Dependency Structure Algorithm. *Procedia Computer Science*, *48*(Iccc), 250–255. https://doi.org/10.1016/j.procs.2015.04.178

Islam, M. M., Marchetto, A., Susi, A., & Scanniello, G. (2012). A Multi-Objective Technique to Prioritize Test Cases Based on Latent Semantic Indexing. *2012 16th European Conference on Software Maintenance and Reengineering*, (3), 21–30. https://doi.org/10.1109/CSMR.2012.13

Jalbert, K., & Bradbury, J. S. (2012). *Predicting mutation score using source code and test suite metrics*. *2012 1st International Workshop on Realizing AI Synergies in Software Engineering, RAISE 2012 - Proceedings*. https://doi.org/10.1109/RAISE.2012.6227969

Jena, A. K., Swain, S. K., & Mohapatra, D. P. (2015). Test Case Generation And Prioritization Based, *78*(3), 336–352.

Jiang, B., & Chan, W. K. (2013). Bypassing code coverage approximation limitations via effective input-based randomized test case prioritization. *Proceedings - International Computer Software and Applications Conference*, (123512), 190–199. https://doi.org/10.1109/COMPSAC.2013.33

Jiang, B., & Chan, W. K. (2015). Input-based adaptive randomized test case prioritization: A local beam search approach. *Journal of Systems and Software*, *105*, 91–106. https://doi.org/10.1016/j.jss.2015.03.066

Jiang, B., Chan, W. K., & Tse, T. H. (2015). PORA: Proportion-Oriented Randomized Algorithm for Test Case Prioritization. *2015 IEEE International Conference on Software Quality, Reliability and Security*, (61202077), 131–140. https://doi.org/10.1109/QRS.2015.28

Jiang, B., Zhai, K., Chan, W. K., Tse, T. H., & Zhang, Z. (2013). On the adoption of MC/DC and control-flow adequacy for a tight integration of program testing and statistical fault localization. *Information and Software Technology*, *55*(5), 897–917. https://doi.org/10.1016/j.infsof.2012.10.001

Jiang, B., Zhang, Z., Chan, W. K., & Tse, T. H. (2009). Adaptive Random Test Case Prioritization. *2009 IEEE/ACM International Conference on Automated Software Engineering*, 233–244. https://doi.org/10.1109/ASE.2009.77

Jiang, B., Zhang, Z., Chan, W. K., Tse, T. H., & 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

Juristo, N., & Moreno, A. M. (2003). *Lecture Notes on Empirical Software Engineering* (Series on). World Scientific Publishing Co. Pte. Ltd.

Kearney, J. P., Sedlmeyer, R. L., Thompson, W. B., Gray, M. A., & Adler, M. A. (1986). Software complexity measurement. *Communications of the ACM*, *29*(11), 1044–1050. https://doi.org/10.1145/7538.7540

Khajeh-Hosseini, A., Greenwood, D., Smith, J., & Sommerville, I. (2012). The Cloud Adoption Toolkit: supporting cloud adoption decisions in the enterprise. *Software - Practice and Experience*, *43*(4), 447–465. https://doi.org/10.1002/spe

Khalilian, A., Abdollahi Azgomi, M., & Fazlalizadeh, Y. (2012). An improved method for test case prioritization by incorporating historical test case data. *Science of Computer Programming*, *78*(1), 93–116. https://doi.org/10.1016/j.scico.2012.01.006

Khan, S. U. R., Lee, S. P., Ahmad, R. W., Akhunzada, A., & Chang, V. (2016). A survey on Test Suite Reduction frameworks and tools. *International Journal of Information Management*, *36*(6), 963–975. https://doi.org/10.1016/j.ijinfomgt.2016.05.025

Khandai, S., Acharya, A. A., & Mohapatra, D. P. (2012). Prioritizing Test Cases Using Business Criticality Test Value. *International Journal of Advanced Computer Science and Applications*, *3*(5), 103–110.

Kim, S., & Baik, J. (2010). An effective fault aware test case prioritization by incorporating a fault localization technique. *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement - ESEM '10*, 1. https://doi.org/10.1145/1852786.1852793

Kintis, M., Papadakis, M., & Malevris, N. (2010). Evaluating mutation testing alternatives: A collateral experiment. *Proceedings - Asia-Pacific Software Engineering Conference, APSEC*, 300–309. https://doi.org/10.1109/APSEC.2010.42

Kitchenham, B., & Charters, S. (2007). *Guidelines for performing Systematic Literature Reviews in Software Engineering*. *Engineering* (Vol. 2). https://doi.org/10.1145/1134285.1134500

Klammer, C., Ramler, R., & Stummer, H. (2016). Harnessing Automated Test Case Generators for GUI Testing in Industry. *2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 227–234. https://doi.org/10.1109/SEAA.2016.60

Korel, B., Koutsogiannakis, G., & Tahat, L. H. (2007). Model-based test prioritization heuristic methods and their evaluation. *Proceedings of the 3rd International Workshop on Advances in Model-Based Testing - A-MOST '07*, 34–43. https://doi.org/10.1145/1291535.1291539

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

Kuhn, D. R., Higdon, J. M., Lawrence, J. F., Kacker, R. N., & Lei, Y. (2012). Combinatorial methods for event sequence testing. *Proceedings - IEEE 5th International Conference on Software Testing, Verification and Validation, ICST 2012*, 601–609. https://doi.org/10.1109/ICST.2012.147

Kumar, O., Bhargavi, P. K., & Kumar, V. (2013). A Single Model for Event-Driven Software. *International Journal on Advanced Computer Theory and Engineering (IJACTE) Example*, *2*, 31–36.

Kumar, S. (2016). A Literature Survey on Finite State Testing of Graphical User Interface, *11*(4), 2942–2954.

Kun, Chunrong, Zhengyu, Z. (2012). Test Case Prioritization Incorporating Ordered Sequence of Program Elements. *Saudi Med J*, *33*, 3–8. https://doi.org/10.1073/pnas.0703993104

Kushwaha, D. S., & Misra, A. K. (2006). Robustness Analysis of Cognitive Information Complexity Measure using Weyuker Properties. *ACM SIGSOFT Software Engineering Notes*, *31*(1). https://doi.org/10.1145/1108768.1108775

Lawanna, A. (2014). The Improvement of Test Case Selection for the Process Software Maintenance, *10*(1), 73–81.

Leblanc, R., Dingle, A., Hagar, J. D., & Knight, J. (2012). *Software Metrics Fundamentals of Dependable Computing for Software Engineers*. https://doi.org/10.1201/b17461

Ledru, Y., Petrenko, A., & Boroday, S. (2009). Using string distances for test case prioritisation. *ASE2009 - 24th IEEE/ACM International Conference on Automated Software Engineering*, (0), 510–514. https://doi.org/10.1109/ASE.2009.23

Lei, Y., Kacker, R., Kuhn, D. R., Okun, V., & Lawrence, J. (2008). IPOG / IPOG-D : efficient test generation for multi-way combinatorial testing, (November 2007), 125–148. https://doi.org/10.1002/stvr

Li, S., Bian, N., Chen, Z., You, D., & He, Y. (2010). A Simulation Study on Some Search Algorithms for Regression Test Case Prioritization. *2010 10th International Conference on Quality Software*, 72–81. https://doi.org/10.1109/QSIC.2010.15

Li, Z., Bian, Y., Zhao, R., & Cheng, J. (2013). A fine-grained parallel multi-objective test case prioritization on GPU. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *8084 LNCS*, 111–125. https://doi.org/10.1007/978-3-642-39742-4_10

Li, Z., Harman, M., & Hierons, R. M. (2007). Search algorithms for regression test case prioritization. *IEEE Transactions on Software Engineering*, *33*(4), 225–237. https://doi.org/10.1109/TSE.2007.38

Lima, L., Iyoda, J., Sampaio, A., & Aranha, E. (2009). Test case prioritization based on data reuse an experimental study. *2009 3rd International Symposium on Empirical Software Engineering and Measurement*, 279–290. https://doi.org/10.1109/ESEM.2009.5315980

Lin, J. W., & Huang, C. Y. (2009). Analysis of test suite reduction with enhanced tie-breaking techniques. *Information and Software Technology*, *51*(4), 679–690. https://doi.org/10.1016/j.infsof.2008.11.004

Luo, Q., Moran, K., & Poshyvanyk, D. (2016). A Large-Scale Empirical Comparison of Static and Dynamic Test Case Prioritization Techniques. *ACM.978-1-4503-4218-6/16/11*, 559–570. https://doi.org/http://dx.doi.org/10.1145/2950290.2950344

Madi, A., Zein, O. K., & Kadry, S. (2013). On the improvement of cyclomatic complexity metric. *International Journal of Software Engineering and Its Applications*, *7*(2), 67–82.

Malishevsky, A. G., Rothermel, G., & Elbaum, S. (2002). Modeling the cost-benefits tradeoffs for regression testing techniques. *Software Maintenance, 2002. Proceedings. International Conference On*, 204–213. https://doi.org/10.1109/ICSM.2002.1167767

Marchetto, A., Islam, M., Asghar, W., Susi, A., & Scanniello, G. (2015). A Multi-Objective Technique to Prioritize Test Cases, *5589*(c), 1–22. https://doi.org/10.1109/TSE.2015.2510633

Mariani, L., Pezzè, M., Riganelli, O., & Santoro, M. (2011). AutoBlackTest: a tool for automatic black-box testing. *Software Engineering (ICSE), 2011 33rd International Conference On*, 1013–1015. https://doi.org/10.1145/1985793.1985979

Mei, L., Chan, W. K., Tse, T. H., & Merkel, R. G. (2009). Tag-based techniques for black-box test case prioritization for service testing. *Proceedings - International Conference on Quality Software*, 21–30. https://doi.org/10.1109/QSIC.2009.12

Memon, M., Pollack, M. E., & Soffa, L. (1999). Using a Goal-driven Approach to Generate Test Cases for GUIs. *Proceedings of the 1999 International Conference on Software Engineering*, 257–266.

Memon, Pollack, M. E., & Soffa, M. L. (2001). Hierarchical GUI test case generation using automated planning. *IEEE Transactions on Software Engineering*, *27*(2), 144–155. https://doi.org/10.1109/32.908959

Mendes, E., Rodriguez, P., & Freitas, V. (2017). Towards improving decision making and estimating the value of decisions in value-based software engineering : the VALUE framework. *Software Quality Journal*. https://doi.org/10.1007/s11219-017-9360-z

Miranda, B., & Bertolino, A. (2016). Scope-aided Test Prioritization, Selection and Minimization for Software Reuse. *Journal of Systems and Software*, *0*, 1–22. https://doi.org/10.1016/j.jss.2016.06.058

Mirarab, S., & Tahvildari, L. (2007). A Prioritization Approach for Software Test Cases Based on Bayesian Networks. *Fundamental Approaches to Software Engineering. Springer Berlin Heidelberg*, *4422*, 276–290. https://doi.org/10.1007/978-3-540-71289-3

Mishra, P. K. (2013). Analysis of Test Case Prioritization in Regression Testing using Genetic Algorithm, *75*(8), 1–10.

Misra, S., & Akman, I. (2008). A unique complexity metric. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *5073 LNCS*(PART 2), 641–651. https://doi.org/10.1007/978-3-540-69848-7_52

Mohamed, S. I., & Wahba, A. M. (2008). Value estimation for software product management. In *Proceedings of the 2008 IEEE IEEM 7-Correctness* (pp. 2196–2200). https://doi.org/10.1109/IEEM.2008.4738261

Moore, I. (2001). Jester-a JUnit test tester. *Proc. of 2nd XP*, 84–87. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.25.1223&rep=rep1&type=pdf

Moreno-Leon, J., Robles, G., & Roman-Gonzalez, M. (2016). Comparing computational thinking development assessment scores with software complexity metrics. *IEEE Global Engineering Education Conference, EDUCON*, *10–13–Apri*(April), 1040–1045. https://doi.org/10.1109/EDUCON.2016.7474681

Mr. Anil Mor. (2014). Evaluate the Effectiveness of Test Suite Prioritization Techniques Using APFD Metric. *IOSR Journal of Computer Engineering (IOSR-JCE)*, *16*(4), 47–51.

Muthusamy, T., & Seetharaman, K. (2014). Effectiveness of Test Case Prioritization Techniques based on Regression Testing. *International Journl of Software Engineering & Application*, *5*(6), 113–123.

Nawar, M. N., & Ragheb, M. M. (2014). Multi-heuristic Based Algorithm for Test Case Prioritization, 449–460.

Nayak, S., Kumar, C., & Tripathi, S. (2016). Effectiveness of prioritization of test cases based on Faults. *2016 3rd International Conference on Recent Advances in Information Technology, RAIT 2016*, 657–662. https://doi.org/10.1109/RAIT.2016.7507977

Ng, P., & Fung, R. Y. K. (2008). Model-based test suite reduction with concept lattice. *Proceedings of the 2008 Advanced Software Engineering and Its Applications, ASEA 2008*, 3–8. https://doi.org/10.1109/ASEA.2008.27

Nguyen, B. N., & Memon, A. M. (2014). An observe-model-exercise* Paradigm to test event-driven systems with undetermined input spaces. *IEEE Transactions on Software Engineering*, *40*(3), 216–234. https://doi.org/10.1109/TSE.2014.2300857

Pandey, A. K., & Shrivastava, V. (2011). Early fault detection model using integrated and cost-effective test case prioritization. *International Journal of System Assurance Engineering and Management*, *2*(1), 41–47. https://doi.org/10.1007/s13198-011-0056-7

Panichella, A., Oliveto, R., Di Penta, M., & De Lucia, A. (2014). Improving Multi-Objective Test Case Selection by Injecting Diversity in Genetic Algorithms. *IEEE Transactions on Software Engineering*, *41*(4), 358–383. https://doi.org/10.1109/TSE.2014.2364175

Panigrahi, C. R., & Mall, R. (2015). Regression test size reduction using improved precision slices. *Innovations in Systems and Software Engineering*, *12*(2), 153–159. https://doi.org/10.1007/s11334-015-0262-6

Panthi, V., & Mohapatra, D. P. (2016). Generating and evaluating effectiveness of test sequences using state machine. *International Journal of System Assurance Engineering and Management*, (Jorgensen 2008). https://doi.org/10.1007/s13198-016-0419-1

Papadakis, M., & Malevris, N. (2010). Automatic Mutation Test Case Generation Via Dynamic Symbolic Execution. https://doi.org/10.1109/ISSRE.2010.38

Parashar, P., Kalia, A., & Bhatia, R. (2012). Pair-Wise Time-Aware Test Case Prioritization, 176–186.

Parejo, J. A., Sánchez, A. B., Segura, S., Ruiz-Cortés, A., Lopez-Herrejon, R. E., & Egyed, A. (2016). Multi-Objective Test Case Prioritization in Highly Configurable Systems: A Case Study. *Journal of Systems and Software*, *122*, 287–310. https://doi.org/10.1016/j.jss.2016.09.045

Petticrew, M., & Roberts, H. (2006). *Systematic Reviews in the Social Sciences: A Practical Guide*. *Cebma.Org*. https://doi.org/10.1027/1016-9040.11.3.244

Pradeepa, R., & K.VimalaDevi. (2013). Effectiveness of Testcase Prioritization using APFD Metric : Survey. *International Journal of Computer Applications*, *0975-8887*, 1–4.

Prakash, N. (2013). Potentially Weighted Method for Test Case Prioritization, *18*(2), 7147–7156. https://doi.org/10.12733/jcis5860

Prakash, N., & Gomathi, K. (2014). Improving Test Efficiency through Multiple Criteria Coverage Based Test Case Prioritization. *International Journal of Scientific & Engineering Research*, *5*(4), 430–435.

Qiu, D., Li, B., Ji, S., & Leung, H. (2014). Regression Testing of Web Service: A Systematic Mapping Study. *ACM Computing Surveys*, *47*(2), 1–46. https://doi.org/10.1145/2631685

Qu, X., Cohen, M. B., & Woolf, K. M. (2007). Combinatorial Interaction Regression Testing: A Study of Test Case Generation and Prioritization. *2007 IEEE International Conference on Software Maintenance*, 255–264. https://doi.org/10.1109/ICSM.2007.4362638

R.W.Hamming. (1950). The Bell system technical journal. *Journal of the Franklin Institute*, *196*(4), 147–160. https://doi.org/10.1016/S0016-0032(23)90506-5

Raju, S., & Uma, G. V. (2012). Factors Oriented Test Case Prioritization Technique in Regression Testing using Genetic Algorithm. *European Journal of Scientific Research*, *74*(3), 1450–216. Retrieved from http://www.europeanjournalofscientificresearch.com

Roongruangsuwan, S., & Daengdej, J. (2010). Test case prioritization techniques. *Journal of Theoretical and Applied Information Technology*, 45–60.

Rothermel, G., Untch, R. H., Chu, C. C. C., & Harrold, M. J. (1999). Test case prioritization: an empirical study. *Proceedings IEEE International Conference on Software Maintenance - 1999 (ICSM'99). "Software Maintenance for Business Change" (Cat. No.99CB36360)*. https://doi.org/10.1109/ICSM.1999.792604

Rothermel, G., Untch, R. H., & Harrold, M. J. (1999). Test Case Prioritization 1 Introduction. *Test*, (December), 1–32.

Sabharwal, S., Sibal, R., & Sharma, C. (2011). A genetic algorithm based approach for prioritization of test case scenarios in static testing. *2011 2nd International Conference on Computer and Communication Technology, ICCCT-2011*, 304–309. https://doi.org/10.1109/ICCCT.2011.6075160

Saha, R. K. (2015). An Information Retrieval Approach for Regression Test Prioritization Based on Program Changes. *Proceeding of the 37th International Conference on Software Engineering (ICSE 2015)*, 268–279. https://doi.org/10.1109/ICSE.2015.47

Said, S. K., Othman, R. R., & Zamli, K. Z. (2011). Prioritizing interaction test suite for t-way testing. *2011 5th Malaysian Conference in Software Engineering, MySEC 2011*, 292–297. https://doi.org/10.1109/MySEC.2011.6140686

Sampath, S., & Bryce, R. C. (2012). Improving the effectiveness of test suite reduction for user-session-based testing of web applications. *Information and Software Technology*, *54*(7), 724–738. https://doi.org/10.1016/j.infsof.2012.01.007

Sampath, S., Bryce, R. C., Viswanath, G., Kandimalla, V., & Koru, a. G. (2008). Prioritizing User-Session-Based Test Cases for Web Applications Testing. *2008 International Conference on Software Testing, Verification, and Validation*, 141–150. https://doi.org/10.1109/ICST.2008.42

Sanchez, A. B., Segura, S., & Ruiz-Cortes, A. (2014). A Comparison of Test Case Prioritization Criteria for Software Product Lines. *Software Testing, Verification and Validation (ICST), 2014 IEEE Seventh International Conference*, 41–50. https://doi.org/10.1109/ICST.2014.15

Schneider, G. P., Ph, D., & Shipp, L. (2010). *Ninth Edition*. https://doi.org/10.1136/bmj.1.5802.756-b

Silva Ouriques, J. F., Cartaxo, E. G., & Lima Machado, P. D. (2015). Revealing influence of model structure and test case profile on the prioritization of test cases in the context of model-based testing. *Journal of Software Engineering Research and Development*, *3*(1), 1. https://doi.org/10.1186/s40411-014-0015-5

Simon baker, emilia mendes. (2010). Assessing the Weighted Sum Algorithm for Automatic Generation of Probabilities in Bayesian Networks. In *Proceedings of the 2010 IEEE International Conference on Information and Automation June 20 - 23, Harbin, China* (pp. 867–873).

Singh, S. S. and A. (2016). Model Based Test Case Prioritization Using Greedy Approach. *International Journal of Emerging Trends in Engineering and Development*, *6*(5), 80–88. https://doi.org/10.17632/CPRWV2HPFM.1

Smith, A. M., & Kapfhammer, G. M. (2009). An empirical study of incorporating cost into test suite reduction and prioritization. *Proceedings of the 2009 ACM Symposium on Applied Computing - SAC '09*, *1*, 461. https://doi.org/10.1145/1529282.1529382

Soid, S. N., Amir, S. A., Ismail, M. A., Hamid, M. N., Amzari, M. M., & Said, M. F. M. (2015). Simulation Studies on the Performance of Small Engine Fuelled by Methane and the Effect of Various Valve Timings. *Indian Journal of Science and Technology*, *8*(November). https://doi.org/10.17485/ijst/2015/v8i30/

Souza, L. S. De, Miranda, P. B. C. De, Prudencio, R. B. C., & Barros, F. D. a. (2011). A Multi-objective Particle Swarm Optimization for Test Case Selection Based on Functional Requirements Coverage and Execution Effort. *2011 IEEE 23rd International Conference on Tools with Artificial Intelligence*, 245–252. https://doi.org/10.1109/ICTAI.2011.45

Srikanth, H., & Banerjee, S. (2012). Improving test efficiency through system test prioritization. *Journal of Systems and Software*, *85*(5), 1176–1187. https://doi.org/10.1016/j.jss.2012.01.007

Srikanth, H., Cashman, M., & Cohen, M. B. (2016). Test case prioritization of build acceptance tests for an enterprise cloud application: An industrial case study. *Journal of Systems and Software*, *119*, 122–135. https://doi.org/10.1016/j.jss.2016.06.017

Srikanth, H., & Cohen, M. B. (2011). Regression testing in Software as a Service: An industrial case study. *2011 27th IEEE International Conference on Software Maintenance (ICSM)*, 372–381. https://doi.org/10.1109/ICSM.2011.6080804

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

Srivastava, P. R., Ray, M., Dermoudy, J., Kang, B., & Kim, T. (2009). Test Case Minimization and Prioritization Using CMIMX Technique *. In *International Conference on Advanced Software Engineering and Its Applications* (Vol. 333031, pp. 25–33).

Srivastava, P. R., Vijay, A., Bariikha, B., Senear, P. S., & Sharma, R. (2009). An optimized technique for test case generation and prioritization using “tabu” search and data clustering. *Proceedings of the 4th Indian International Conference on Artificial Intelligence, IICAI 2009*, 30–46. Retrieved from http://www.scopus.com/inward/record.url?eid=2-s2.0-84857607266&partnerID=tZOtx3y1

Staats, M., Loyola, P., & Rothermel, G. (2012). Oracle-Centric Test Case Prioritization. *2012 IEEE 23rd International Symposium on Software Reliability Engineering*, 311–320. https://doi.org/10.1109/ISSRE.2012.13

Sultan, A. B. M., Abdul Ghani, A. A. Bin, Baharom, S., & Musa, S. (2014). An Evolutionary Regression Test Case Prioritization based on Dependence Graph and Genetic Algorithm for. *Icetet*, 22–26.

Sultan, A., Baharom, S., Abdul Ghani, A. A., Din, J., & Zulzalil, H. (2015). Adopting Genetic Algorithm To Eenhance State - Sensitivity Partitioning. In *Proceedings of the 5th International Conference on Computing and Informatics, ICOCI 2015* (pp. 280–286).

Sultan, A. M. (2017). *An Optimized Test Case Generation Technique For Enhancing State-Sensitivity Partitioning*. Universiti Putra Malaysia.

Tahat, L., Korel, B., Koutsogiannakis, G., & Almasri, N. (2016). State-based models in regression test suite prioritization. *Software Quality Journal*, 1–40. https://doi.org/10.1007/s11219-016-9330-x

Tanwani, L., & Waghire, A. (2016). Test Case Prioritization for Regression Testing of GUI. *International Acadmey of Engineering and Medical Research*, *13*(1), 686–692. Retrieved from http://www.iaemr.com/wp-content/uploads/2016/11/test-case-prioritization-regression-testing-gui.pdf

Thirumalai, C., R.R., S., & L, R. R. (2017). An Assessment of Halstead and COCOMO Model for Effort Estimation. *International Conference on Innovations in Power and Advanced Computing Technologies [i-PACT2017]*, (April), 1–4.

Tonella, P., Avesani, P., & Susi, A. (2006). Using the Case-Based Ranking Methodology for Test Case Prioritization. *22nd IEEE International COnference on Software Maintenance (ICSM'06)*, 123–132. https://doi.org/10.1109/ICSM.2006.74

Tyagi, M., & 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

Vijayakumar, E., & Punithavalli, M. (2013). Enhanced Approaches To Improve Graphical User. *ARPN Journal of Engineering and Applied Sciences*, *8*(7), 519–524.

Wang, S., Buchmann, D., Ali, S., & Liaaen, M. (2014). Multi-Objective Test Prioritization in Software Product Line Testing : An Industrial Case Study. *SPLC '14 Proceedings of the 18th International Software Product Line Conference*, 32–41. https://doi.org/10.1145/2648511.2648515

Weyuker, E. J. (1988). Evaluating software complexity measures. *IEEE Transactions on Software Engineering*, *14*(9), 1357–1365. https://doi.org/10.1109/32.6178

Wohlin, Runeson, Martin, Magnus, Bjorn, A. (2012). *Experimentation in Software Engineering*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-29044-2

Ye, N., Chen, X., Ding, W., Jiang, P., Bu, L., & Li, X. (2012). Regression Test Cases Generation Based on Automatic Model Revision. *2012 Sixth International Symposium on Theoretical Aspects of Software Engineering*, 127–134. https://doi.org/10.1109/TASE.2012.31

Yoo, S., & Harman, M. (2007). Regression Testing Minimisation, Selection and Prioritisation : A Survey. *Test. Verif. Reliab*, *00*, 1–7. https://doi.org/10.1002/000

Yoo, S., Harman, M., Tonella, P., & Susi, A. (2009). Clustering test cases to achieve effective and scalable prioritisation incorporating expert knowledge. *Proc. ISSTA*, 201–212. https://doi.org/10.1145/1572272.1572296

Yu, Y. T., & Lau, M. F. (2012). Fault-based test suite prioritization for specification-based testing. *Information and Software Technology*, *54*(2), 179–202. https://doi.org/10.1016/j.infsof.2011.09.005

Yuan, F., Bian, Y., Li, Z., & Zhao, R. (2015). Search-Based Software Engineering, *9275*, 109–124. https://doi.org/10.1007/978-3-319-22183-0

Yuan, X., Cohen, M., & Memon, A. M. (2007). Covering Array Sampling of Input Event Sequences for Automated Gui Testing. *Proceedings of the Twenty-Second IEEE/ACM International Conference on Automated Software Engineering*, 405–408. https://doi.org/10.1145/1321631.1321695

Zanoni, M., Perin, F., Fontana, F. A., & Viscusi, G. (2014). Pattern detection for conceptual schema recovery in data-intensive systems. *Journal of Software: Evolution and Process*, *26*(12), 1172–1192. https://doi.org/10.1002/smr

Zhai, K., Jiang, B., & Chan, W. K. (2014). Prioritizing test cases for regression testing of location-based services: Metrics, techniques, and case study. *IEEE Transactions on Services Computing*, *7*(1), 54–67. https://doi.org/10.1109/TSC.2012.40

Zhang, L., Hao, D., Zhang, L., Rothermel, G., & Mei, H. (2013). Bridging the gap between the total and additional test-case prioritization strategies. *Proceedings - International Conference on Software Engineering*, 192–201. https://doi.org/10.1109/ICSE.2013.6606565

Zhang, L., Zhou, J., Hao, D., Zhang, L., & Mei, H. (2009). Jtop: Managing JUnit test cases in absence of coverage information. *ASE2009 - 24th IEEE/ACM International Conference on Automated Software Engineering*, 677–679. https://doi.org/10.1109/ASE.2009.22

Zhang, W., Wei, B., & Du, H. (2014). Test Case Prioritization Based on Genetic Algorithm and Test-Points Coverage Evaluation of Test Case Prioritization, 644–654.

Zhang, X., Xie, X., & Chen, T. Y. (2016). Test Case Prioritization Using Adaptive Random Sequence with Category-Partition-Based Distance. *2016 IEEE International Conference on Software Quality, Reliability and Security (QRS)*, 374–385. https://doi.org/10.1109/QRS.2016.49

BIODATA OF STUDENT

Johanna Ahmad was born on July 20, 1982 in Johor, Malaysia. She started her primary education in Sekolah Kebangsaan Kampung Melayu, Kluang, Johor for six years. Then, she attended Sekolah Tinggi Segamat and Sekolah Tinggi Kluang to complete her secondary education. In 2000, she was accepted to pursue her diploma in Computer Science at Universiti Teknologi Mara. She was promoted to pursue her degree in 2003, and then received a Bachelor's degree in Information System Engineering in October 2006. She worked in the software sector from 2007 to 2009 as a Software Engineer. One year later, she joined the government sector as an Information Technology Officer at the Prime Minister's Department, Malaysia. Her job scope was focused on database management. She was also pursuing a postgraduate degree at Universiti Putra Malaysia and graduated with an MSc. in Computer Science (Software Engineering) in 2012. In September 2015, she enrolled as a full-time student at Universiti Putra Malaysia, where she pursued her PhD in Software Engineering. She is currently an Information Technology Officer at the Prime Minister's Department of Malaysia, Putrajaya.

LIST OF PUBLICATIONS

Ahmad, J., & Baharom, S. (2017). A Systematic Literature Review of the Test Case Prioritization Technique for Sequence of Events. *International Journal of Applied Engineering Research*, *12*(7), 1389–1395.

Ahmad, J., & Baharom, S. (2017). Comparison of Software Complexity Metrics in Measuring the Complexity of Event Sequences. In *Lecture Notes in Electrical Engineering* (Vol. 424, pp. 615–624). Springer. https://doi.org/10.1007/978-981-10-4154-9

Ahmad, J., & Baharom, S. (2018). Factor Determination in Prioritizing Test Cases for Event Sequences : A Systematic Literature Review. *Journal of Telecommunication, Electronic and Computer Engineering*, *10*(Xe-ISSN: 2289-8131), 1–6.

Ahmad, J., Baharom, S., & Sapaat, M. A. (2018). Test Case Prioritization Technique For Event Sequence Test Cases Based On Redundancy, *96*(18), 6041–6052.

Ahmad, J., Baharom, S., Abdul Ghani, A. A., Zulzalil, H., & Din, J. (2019). *Intelligent Computing* (Vol. 857). Springer International Publishing. https://doi.org/10.1007/978-3-030-01177-2

Ahmad, J., Baharom, S., Abdul Ghani, A. A., Zulzalil, H., & Din, J, "Prioritizing Event Sequence Test Cases Based on Faults".(Submitted for GCMT KL 2018)

Ahmad, J., Baharom, S., Abdul Ghani, A. A., Zulzalil, H., & Din, J, "A Review on Test Case Prioritization Technique for Event Sequence Test Cases", 4th International Postgraduate Conference on Enginering, Science and Technology , IPCEST 2018, 2018.

UNIVERSITI PUTRA MALAYSIA

STATUS CONFIRMATION FOR THESIS / PROJECT REPORT AND COPYRIGHT

ACADEMIC SESSION : _____

TITLE OF THESIS / PROJECT REPORT :

MULTIFACTOR APPROACH TO PRIORITIZE EVENT SEQUENCE TEST CASES

NAME OF STUDENT: JOHANNA BINTI AHMAD

I acknowledge that the copyright and other intellectual property in the thesis/project report belonged to Universiti Putra Malaysia and I agree to allow this thesis/project report to be placed at the library under the following terms:

1. This thesis/project report is the property of Universiti Putra Malaysia.
2. The library of Universiti Putra Malaysia has the right to make copies for educational purposes only.
3. The library of Universiti Putra Malaysia is allowed to make copies of this thesis for academic exchange.

I declare that this thesis is classified as :

*Please tick (√)

☐	**CONFIDENTIAL**	(Contain confidential information under Official Secret Act 1972).
☐	**RESTRICTED**	(Contains restricted information as specified by the organization/institution where research was done).
☐	**OPEN ACCESS**	I agree that my thesis/project report to be published as hard copy or online open access.

This thesis is submitted for :

☐ **PATENT** Embargo from _____ (date) until _____ (date)

Approved by:

_____ (Signature of Student)
New IC No/ Passport No.:

Date :

_____ (Signature of Chairman of Supervisory Committee)
Name:

Date :

[Note : If the thesis is CONFIDENTIAL or RESTRICTED, please attach with the letter from the organization/institution with period and reasons for confidentially or restricted.]