EVOLUTIONARY APPROACH FOR COMBINATORIAL TESTING OF SOFTWARE PRODUCT LINES

MOHD ZANES BIN SAHID

FSKTM 2020 14

EVOLUTIONARY APPROACH FOR COMBINATORIAL TESTING OF SOFTWARE PRODUCT LINES

By

MOHD ZANES BIN SAHID

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

July 2020

DEDICATION

Tesis ini didedikasi untuk ibu, Wahyunah Yusof; ayah, Sahid Lakiman; dan isteri, Halimatun Saadiah Fardi, yang sentiasa mendoakan, membantu dan menyokong saya mengharungi perjalanan ini.

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

EVOLUTIONARY APPROACH FOR COMBINATORIAL TESTING OF SOFTWARE PRODUCT LINES

By

MOHD ZANES BIN SAHID

July 2020

Chair : Professor Abu Bakar Md. Sultan, PhD
Faculty : Computer Science and Information Technology

Software Product Line (SPL) focuses on common features reusability, formulated for different software products. Complete testing on the entire SPL is known to be unfeasible. This is due to the very large number of possible products to be produced; configured using all possible sets of features in the SPL.

Combinatorial Testing (CT) has been widely used in software testing as it is able to generate test input for a single software product that deviates from exhaustive testing, nevertheless proven to be effective. In SPL testing, to generate minimal test configuration that maximizes t-wise coverage is not trivial, especially when dealing with a huge number of features and when constraints have to be satisfied, which is the case in most SPL systems. Two salient obstacles in SPL testing are identified; (1) time required to cover feature configuration testing for large scale Feature Models and higher strength of t-wise testing, and (2) insufficient coverage of feature interaction towards higher fault detection due to uniform strength of feature combination. In the case of unfeasible higher strength CT-based testing for SPL, this thesis proposed a relaxed version of Covering Array (CA), i.e. Variable-strength CA (VCA). In highly configurable systems, the entire configuration spaces contain several groups of sub-configurations that have different level of risks that can cause failures. By focusing on these group of sub-configurations, the space of testing artefacts can be reduced, while still maintain the results of testing.

Practical Combinatorial Testing is challenging, such that it is hard to decide what level of interaction strength to be applied to which different groups of features. There are two inherent challenges; (1) to decide what value of interaction strength one should apply, and (2) to choose which group or set of features the interaction strength should be applied to. For these reasons, part of this thesis presents and evaluates a new technique aimed at tackling all the mentioned difficulties in the combinatorial testing of SPL system. The approaches are (1) adopting an Estimation of Distribution Algorithm approach to aid the construction of a covering array driven by second order feature dependence, and (2) construct a feature configuration dependence graph to assist in building the variable-strength covering array. These techniques are implemented as COTED (Combinatorial Testing using Bivariate EDA).

Experiments have been done to gauge the performance of COTED from three different facets; (1) the ability to reduce test configuration redundancy, (2) the ability to achieve better rate of interaction coverage and (3) the ability to detect high number of faults. Four empirical studies conducted to compare COTED against other approaches (namely ICPL, LOOKUP and CASA). The statistical analysis of the results obtained from the experiments are promising. For the first facet, COTED is comparable with LOOKUP, but a slightly better redundancy is achieved by ICPL with significant level of 0.01. For the second facet, COTED able to outperform ICPL and LOOKUP with significance level of 0.1. For the third facet, the two sets of VCA test configurations (i.e. (i) 2-wise and partial 3-wise, and (ii) 2-wise, partial 3 and partial 4-wise) generated by COTED are able to achieve competitive fault detection rate (with significance level of 0.05) using less number of test configurations. The result shows that COTED increase the interaction coverage rate, and managed to produce less number of test configurations to detect an almost same number of faults as the other strategies, hence, suggests that it is able to improve Combinatorial Interaction Testing (CIT) for SPL.

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

PENDEKATAN EVOLUSI UNTUK PENGUJIAN KOMBINASI PADA BARISAN PRODUK PERISIAN

Oleh

MOHD ZANES BIN SAHID

Julai 2020

Pengerusi : Profesor Abu Bakar Md. Sultan, PhD
Fakulti : Sains Komputer dan Teknologi Maklumat

Barisan Produk Perisian (SPL) menumpu kepada kitar semula ciri-ciri yang dikongsi oleh produk-produk perisian yang berlainan. Pengujian lengkap pada keseluruhan SPL agak mustahil untuk dilaksana. Ini disebabkan oleh bilangan produk perisian yang mungkin dihasilkan adalah terlalu besar; berdasarkan pengaturan sebahagian atau semua gabungan fitur-fitur yang dibangunkan untuk sesuatu SPL.

Pengujian Kombinasi (CT) telah dibangun dan digunapakai dalam pengujian perisian kerana ia mampu menjana pelbagai input pada produk perisian tunggal dan mampu mengatasi pengujian secara menyeluruh, dan terbukti berkesan. Dalam SPL, untuk menjana configurasi ujian minimum yang memaksimakan liputan *t*-hala bukanlah suatu yang mudah, terutama apabila berdepan dengan SPL yang bersaiz besar dan keperluan mematuhi kekangan sistem. Dua halangan utama pada pengujian SPL dikenalpasti, iaitu; (1) masa yang diperlukan untuk meliputi pengujian konfigurasi semua ciri pada Model Fitur berskala besar dan peringkat tinggi pengujian *t*-hala, dan (2) kekurangan liputan pada interaksi fitur untuk mengesan kecacatan pada perisian disebabkan peringkat kombinasi fitur yang seragam. Sebagai usaha untuk mengatasi cabaran itu, tesis ini membentangkan versi lain kepada Jajaran Liputan (CA) iaitu Jajaran Liputan Kekuatan Bolehubah (VCA). Dalam sistem yang banyak melibatkan konfigurasi, gabungan-gabungan fitur yang berlainan mempunyai risiko yang berbeza terhadap kesan pada kemungkinan wujudnya pepijat. Dengan memfokuskan hanya pada gabungan tertentu, ruang pengujian dapat dikurangkan dan berkemungkinan kualiti pengujian dapat dikekalkan.

CT yang practical adalah mencabar. Ini kerana, adalah sukar untuk menentukan peringkat interaksi mana harus dilaksana ke atas gabungan fitur-fitur yang mana. Atas alasan inilah satu teknik baru dibentang dan dibincangkan dalam tesis ini. Pendekatan yang digunakan ialah (1) mengguna pakai Algoritma Anggaran Teragih (EDA) untuk membantu membina CA berpandukan kepada kebergantungan fitur aras kedua, dan (2) menjana graf kebergantungan konfigurasi fitur sebagai asas kepada VCA.

Eksperimen dibuat bagi menilai prestasi teknik yang dicadangkan dari tiga aspek berbeza iaitu; (1) kemampuan untuk meminimakan kelewahan konfigurasi ujian, (2) kemampuan untuk mencapai liputan interaksi yang lebih baik, dan (3) kemampuan untuk mengesan lebih banyak pepijat. Empat kajian empiris telah dijalankan untuk membandingkan COTED dengan empat pendekatan lain (iaitu ICPL, LOOKUP dan CASA). Analisa statistik ke atas data amat memberangsangkan. Untuk aspek pertama, COTED didapati setara dengan LOOKUP, tetapi pengurangan kelewahan adalah sedikit baik melalui ICPL dengan tahap signifikan 0.01. Untuk aspek kedua, COTED mengatasi ICPL dan LOOKUP dengan tahap signifikan 0.1. Untuk aspek ketiga, kedua-dua set konfigurasi ujian (iaitu (i) 2-hala dan sebahagian 3-hala, dan (ii) 2-hala, sebahagian 3 dan 4-hala) yang dijana COTED mampu mengesan bilangan pepijat yang setanding dengan teknik lain (dengan tahap signifikan 0.05) tetapi menggunakan jauh lebih sedikit bilangan konfigurasi ujian. Kesimpulannya, secara purata, COTED meningkatkan kadar capaian liputan dan mengurangkan bilangan konfigurasi ujian tetapi mengekalkan kebolehan mengesan pepijat, sekaligus menyokong andaian penyelidikan iaitu ia mampu menambahbaik pengujian kombinasi pada SPL.

ACKNOWLEDGEMENTS

I would like to express my gratitude to the supervisory committee led by Professor Dr Abu Bakar Md. Sultan and committee members, Professor Dr Abdul Azim Abd. Ghani and Associate Professor Dr Salmi Baharom for their guidance, motivation and patience supervision throughout my research.

My deepest appreciation is to my parents, wife and children for their support, love and prayers over the past years. Finally, my thanks are also extended to my friends and colleagues, for sharing experiences throughout the years.

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

Abu Bakar Md. Sultan, PhD
Professor
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)

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

ZALILAH MOHD SHARIFF, PhD
Professor and Dean
School of Graduate Studies
Universiti Putra Malaysia

Date: 12 November 2020

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 other 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.: Mohd Zanes Bin Sahid (GS41144)

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) are adhered to.

Signature: _____
Name of Chairman of Supervisory Committee: Professor Dr. Abu Bakar Md. Sultan

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

Signature: _____
Name of Member of Supervisory Committee: Associate Professor Dr. Salmi Baharom

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

APCC	Average Percentage of Covering Array Coverage
BDD	Binary Decision Diagram
BMDA	Bivariate Marginal Distribution Algorithm
CA	Covering Array
CIT	Combinatorial Interaction Testing
CNF	Conjunctive Normal Form
DM	Decision Modelling
EC	Evolutionary Computation
EDA	Estimation of Distribution Algorithm
FCDG	Feature Configuration Dependence Graph
FD	Feature Diagram
FM	Feature Model
GA	Genetic Algorithm
MCA	Mixed-covering Array
OA	Orthogonal Array
OVM	Orthogonal Variability Model
PV	Probability Vector
SA	Simulated Annealing
SAT	Boolean Satisfiability
SBSE	Search-Based Software Engineering
SBT	Search-Based Testing
SMS	Systematic Mapping Study
SPL	Software Product Lines
TC	Test Configuration
VCA	Variable-strength CA
cFCDG	Cumulative FCDG

CHAPTER 1

INTRODUCTION

1.1 Background

Software Product Line (SPL) is a software engineering paradigm that facilitates the development of software products that share common features. The primary motivation of structuring one's systems as a product line is to allow customers to have a system tailored for their variety purposes and needs, while still avoiding redundancy of software artefacts. It is common for customers to have conflicting requirements. In that case, it is not possible to develop and deploy one system for all customers.

In SPL, a unit of system function is represented as a *feature*. Features are explicitly defined as common and variable features, and utilized throughout the SPL development process. One way to model the commonalities and variabilities is an SPL is by using a Feature Model (FM) based on feature modelling technique. A FM sets up the commonalities and variabilities of a product line in a tree, such that configuring the product line proceeds from the root of the tree. The tree representation of FM is known as Feature Diagram (Czarnecki and Eisenecker, 2000). A FM encompasses the constraints linking the features. Feature modelling is a popular way to model SPL variability and it is by far the most commonly used technique in industry (Berger et al., 2013).

In correlation to that, an important activity in SPL that attracts significant attention among researchers is feature configuration. Feature configuration is a process in which two or more features are combined and utilized together in a single software product. This could possibly result in unspecified and unintended system behaviour and might lead to incorrect execution. Hence, it is crucial to test all possible feature configurations in order to reduce the potential misbehaviour of interacting features. But, to test all possible feature configurations is unfeasible. In a most trivial case, small number of features in a FM will results in small number of possible feature configurations. However, the number of feature configurations increase dramatically as the size of FM increases (Kim et al., 2011). Therefore, exhaustively testing all feature configurations especially in large-scale FM is not practical.

1.2 Research Motivation

Software engineering studies based on the SPL paradigm is an active research (Lindohf et al., 2020; Nešić et al., 2019) due to its potential cost reduction, faster deployment and better maintenance, thanks to its intertwine coexistence of feature commonality and variability principles. Despite the significant strengths it brings to software engineering community, it poses a great challenge in various aspects, including testing.

Combinatorial Interaction Testing (CIT) is a prominent and dependable technique towards feature configuration testing. In the context of SPL testing, CIT works based on the construction of an array, containing combination of two or more features. This array is known as t-wise covering array, where *t* refers to the number of features chosen to be considered in the testing. Current state-of-the-art feature configuration testing of SPL mostly deals with low strength (2 and 3) t-wise testing. However, higher strength of t-wise testing is important towards achieving higher fault detection. But, higher strength of t-wise testing results in much more test cases to be executed for each feature configuration. The scenario gets worst in the case of SPL having high number of features, as the number of features to be considered for any combination gets higher. Thus, an efficient and effective strategy to generate t-wise covering array that could satisfy a decent t-wise coverage is deemed necessary.

A number of researchers had proposed a couple of prominent strategies to reduce the combinatorial explosion of feature configuration testing (Roberto E Lopez-Herrejon et al., 2015). Most of current approaches are based on greedy algorithms with only few works leveraged the potential of optimization techniques, which have been widely used in the single software (non-SPL) development testing. Additionally, it is common that software engineers develop an SPL with some concrete or predetermined software products as a subset of its final products (Oster et al., 2010). Employment of conventional meta-heuristics techniques to generate minimized test configuration often requires these predefined valid software products as seeds or initial population. In traditional meta-heuristics (e.g. genetic algorithm), probabilities are implicitly employed in selection and re-production operators to produce offspring. In this sense, by explicitly building a probabilistic model of features distribution out of this seeds, it allows us to estimate the distribution of highly fit features in subsequent candidate solutions. In view of this, and also inspired by current works on search-based SPL testing, this thesis explores the viability and strategy towards probability and statistics approach of search-based SPL testing.

1.3 Problem Statement

To consider all feature configurations towards total coverage of interaction testing is an ideal case, as it could discover all interaction faults. However, exhaustive testing of all feature configurations is not feasible on most SPL systems (C. Henard et al., 2014b; Lackner and Schlingloff, 2017). On one hand, the size of the test configuration needs to be reduced so that it can meet the product release deadlines and cost constraints. On the other hand, insufficiently testing of an SPL system should be avoided as it could affect too many products and customers if the faulty feature(s) left undetected. Therefore, in feature configuration testing, an efficient test configuration generation technique (such as pairwise testing) that could balance between the maximum test coverage and minimum testing cost is required.

Pairwise testing measures the quality of a solution in terms of its ability to fulfil the intended pairwise coverage (Christopher Henard et al., 2013; Marijan et al., 2013; Wang et al., 2013). Observing pairwise coverage is synonym to observing the test configuration size. The smaller the size, the better the solution. However, no conclusion can be made if they have the same *t* size and same minimum test configuration size (B. Chen and Zhang, 2011). Therefore, and based on the lack of works on other evaluation metrics, we found that there is a need to consider (i) test configuration redundancy and (ii) rate of coverage metrics to evaluate the quality of current solutions.

Numerous approaches have been proposed based on the Combinatorial Interaction Testing and Search-Based Testing. Based on evidences from recent works, hybrid approach of Combinatorial Interaction Testing and Search-Based Testing have been perceived as the most dominating and promising technique to realize feature configuration testing of SPL. However, most of them are focusing on lower strength (less than 4) of t-wise combinations of features (Hasan et al., 2020; C. Henard et al., 2014b). The main limitation that is causing this phenomenon is due to the expensive computation time required to cover higher strength t-wise testing (Borazjany et al., 2012; Myra B. Cohen et al., 2003a; C. Henard et al., 2014b; M. F. Johansen et al., 2012a). Some empirical results for non-SPL problem suggested that higher strengths are important in detecting more faults (Petke et al., 2013). However, only few attempts have been reported to deal with higher strength t-wise feature combinations (Borazjany et al., 2012; C. Henard et al., 2014b), which makes the practical gain of SPL testing using current techniques questionable (C. Henard et al., 2014b). Thus, further work on test configuration generation that are based on higher strength of t-wise testing is deemed necessary so that more faults can be detected.

We have conducted a systematic mapping study on 44 primary studies which cover diverse CA generation technique. Among others, we found that the most investigated strength is pairwise (2-wise) strength, which accounted for 55% (24) from all reviewed work. 13 studies (30%) have empirically evaluated their solutions for up to 3-wise covering array. Three studies (7%) reported the evaluation for up to 4- wise covering array, and only two works managed to scale their work for up to 6-wise strength. In addition to that, we also found that only fixed-strength Covering Array (CA) is being employed. There is also an attempt to investigate the effect of considering the size of feature model in deciding the optimum level of fixed-strength CA (Huang et al., 2018), however, one might argue that the problem space is not associated by the feature count, as the size of valid feature configuration is non-deterministic.

In the case of unfeasible higher strength CIT-based testing, a relaxed version of Covering Array, i.e. Variable-strength CA (VCA) is deemed viable (Yilmaz et al., 2006). Yilmaz et al. suggested that in highly configurable systems, the entire configuration spaces contain several groups of sub-configurations that have different level of risks that can cause failures. This can be represented and modelled as a Covering Array with multiple t-wise strength. However, the generation of VCA is challenging, such that it is hard to decide what level of interaction strength to be applied to which different groups of features (Yilmaz et al., 2014). There are two inherent challenges; (i) to decide what value of interaction strength, t_s, where s in {2,3,4,5,6}, one should apply, and (ii) to choose which groups or sets of features the t_s should be applied to.

1.4 Research Objectives

The main objective of this thesis is to enhance current Combinatorial Interaction Testing for SPL, by leveraging the estimation of feature distribution to aid in generating the Variable-Strength Covering Array. In order to fulfil the main objective, three sub-objectives are defined as follows:

1. To propose a test configuration generation strategy, by adopting an Estimation of Distribution Algorithm (EDA)-based technique that can fulfil pairwise coverage and accelerate the interaction coverage.

2. To propose a Variable-strength Covering Array (VCA) generation strategy that can increase fault detection rate, driven by the feature dependency information obtained from EDA-based pairwise testing.

3. To empirically evaluate the effectiveness of the proposed approaches in improving Combinatorial Interaction Testing of SPL.

1.5 Scope of Work

The following are four scoping aspects of this research relevant to SPL testing technique:

1. *Variability Modelling*. Many works have been established on SPL domain by employing a model-based variability modelling approach, which is known as Feature Modelling. The central artefact in Feature Modelling is FM, which is defined, analysed, configured and tested by researchers in most research works as well as by practitioners in industries. Due to its widely used, the proposed technique in our work is being formulated based on the attributes and characteristics of FM only, not for other type of Variability Modelling such as Orthogonal Variability Model and Decision Modelling.

2. *Testing on the implementation phase of Domain Engineering*. There are two main phases of SPL software engineering paradigm, and both of them are well established and accepted in practice. The two phases are Domain Engineering and Application Engineering. In the Domain Engineering phase, the Domain Implementation process will enable the reusable artefacts or core assets to be developed that correspond to the specified features. This phase and these assets are the main objects of testing; as thorough testing is required to avoid much higher testing efforts in later phases.

3. *Combinatorial Interaction Testing strategy*. Findings from investigation of related literatures highlighted that the testing in SPL demands more works towards scalable testing especially on higher strength of t-wise testing. Numerous evidences are available that have been formulated for uniform strength of t-wise testing, but none are building solution explicitly based on variable strength of t-wise testing. The work in this thesis has been formulated towards a viable SPL testing with respect to complete pairwise testing combined with partial higher strength of t-wise testing.

1.6 Contributions of the Thesis

Generally, towards the end, this thesis presents a strategy and empirical evaluation of a new approach that provides a viable Combinatorial Testing-based approach without having to compute all higher strength t-wise. Specifically, the contributions of this research are as follows:

- We devise a set of algorithms based on bivariate marginal distribution in SPL context to generate minimum test configuration that fully satisfy pairwise coverage. This approach is perceived as a lightweight variant of estimation of distribution algorithm, specifically (and evolutionary search, generally), in which only the statistical value of the population is maintained across generation, instead of the actual population of individual.

- We introduce the notion of Feature Configuration Dependence Graph (FCDG), which contains the dependency information between pairs of features, extracted using statistical computation.

- We devise a strategy towards Variable-strength Covering Array generation using the constructed FCDG.

- We implement the proposed approach using Java programming language and packaged as a standalone command line application. Empirical studies are conducted to gauge the performance of the proposed approach against state-of-the-art Combinatorial Testing tool for SPL.

1.7 Organization of the Thesis

This thesis is organized as follows; Chapter 1 introduces the background of the study, followed by research motivation and research direction. Chapter 2 discuss and review existing literature that spans across different but related domains including SPL, software testing and some notable categories of works. Chapter 3 discuss the methodology used throughout the entire study. Chapter 4 elaborate and discuss the requirements, algorithms and processes to achieve the first research objective. Chapter 5 elaborate and discuss the requirements, algorithms and processes to meet the second objective. Chapter 6 describes the empirical process, the results and findings. Finally, Chapter

7 discuss the conclusion, relation to other works and suggestion for future works.

1.8 Summary

SPL Testing demands new mechanisms due to its nature of feature commonality and variability. Ideally, complete testing is required. However, exhaustive testing is unfeasible.

REFERENCES

Adamo, D., Bryce, R., & King, T. M. (2018). Randomized Event Sequence Generation Strategies for Automated Testing of Android Apps *Information Technology-New Generations* (pp. 571-578): Springer.

Agrawal, A., Fu, W., & Menzies, T. (2018). What is wrong with topic modeling? And how to fix it using search-based software engineering. *Information and Software Technology, 98*, 74-88.

Ahmed, B. S., Gambardella, L. M., Afzal, W., & Zamli, K. Z. (2017). Handling constraints in combinatorial interaction testing in the presence of multi objective particle swarm and multithreading. *Information and Software Technology, 86*, 20-36.

Ahmed, B. S., & Zamli, K. Z. (2011). *Comparison of metahuristic test generation strategies based on interaction elements coverage criterion.* Paper presented at the Industrial Electronics and Applications (ISIEA), 2011 IEEE Symposium on.

Ahmed, B. S., Zamli, K. Z., & Lim, C. P. (2012). Application of particle swarm optimization to uniform and variable strength covering array construction. *Applied Soft Computing, 12*(4), 1330-1347.

Aho, A. V., & Hopcroft, J. E. (1974). *The design and analysis of computer algorithms*: Pearson Education India.

Al-Hajjaji, M., Thüm, T., Meinicke, J., Lochau, M., & Saake, G. (2014). *Similarity-based prioritization in software product-line testing.* Paper presented at the Proceedings of the 18th International Software Product Line Conference.

Alazzawi, A. K., Rais, H. M., & Basri, S. (2019). *Hybrid Artificial Bee Colony Algorithm for t-Way Interaction Test Suite Generation.* Paper presented at the Computer Science On-line Conference.

Alsariera, Y. A., Majid, M. A., & Zamli, K. Z. (2015, 19-21 Aug. 2015). *SPLBA: An interaction strategy for testing software product lines using the Bat-inspired algorithm.* Paper presented at the Software Engineering and Computer Systems (ICSECS), 2015 4th International Conference on.

Apel, S., Batory, D., Kästner, C., & Saake, G. (2013a). Analysis of Software Product Lines *Feature-Oriented Software Product Lines: Concepts and Implementation* (pp. 243-282). Berlin, Heidelberg: Springer Berlin Heidelberg.

Apel, S., & Kästner, C. (2009). An Overview of Feature-Oriented Software Development. *Journal of Object Technology, 8*(5), 49-84.

Apel, S., Rhein, A. v., Wendler, P., Größlinger, A., & Beyer, D. (2013b). *Strategies for product-line verification: Case studies and experiments.* Paper presented at the Proceedings of the International Conference on Software Engineering.

Arrieta, A., Wang, S., Sagardui, G., & 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.

Atkinson, C., & Muthig, D. (2002). Component-Based Product-Line Engineering with the UML. In C. Gacek (Ed.), *Software Reuse: Methods, Techniques, and Tools* (Vol. 2319, pp. 343-344): Springer Berlin Heidelberg.

Bäck, T., Fogel, D. B., & Michalewicz, Z. (1997). *Handbook of evolutionary computation*: CRC Press.

Bäck, T., Fogel, D. B., & Michalewicz, Z. (2018). *Evolutionary computation 1: Basic algorithms and operators*: CRC press.

Bagheri, E., & Gasevic, D. (2011). Assessing the maintainability of software product line feature models using structural metrics. *Software Quality Journal, 19*(3), 579-612.

Baller, H., Lity, S., Lochau, M., & Schaefer, I. (2014). *Multi-objective test suite optimization for incremental product family testing.* Paper presented at the Software Testing, Verification and Validation (ICST), 2014 IEEE Seventh International Conference on.

Baluja, S., & Davies, S. (1998). *Fast probabilistic modeling for combinatorial optimization.* Paper presented at the AAAI/IAAI.

Batory, D. (2005). *Feature models, grammars, and propositional formulas.* Paper presented at the International Conference on Software Product Lines.

Benavides, D., Segura, S., Trinidad, P., & Cortés, A. R. (2007). FAMA: Tooling a Framework for the Automated Analysis of Feature Models. *VaMoS, 2007*, 01.

Benavides, D., Segura, S., Trinidad, P., & Ruiz-Cortés, A. (2006). A first step towards a framework for the automated analysis of feature models. *Proc. Managing Variability for Software Product Lines: Working With Variability Mechanisms*, 39-47.

Berger, T., Rublack, R., Nair, D., Atlee, J. M., Becker, M., Czarnecki, K., & Wąsowski, A. (2013). *A survey of variability modeling in industrial practice.* Paper presented at the Proceedings of the Seventh International Workshop on Variability Modelling of Software-intensive Systems.

Bondy, J. A., & Murty, U. S. R. (1976). *Graph theory with applications* (Vol. 290): Macmillan London.

Borazjany, M. N., Yu, L., Lei, Y., Kacker, R., & Kuhn, R. (2012). *Combinatorial Testing of ACTS: A Case Study.* Paper presented at the Proceedings of the 2012 IEEE Fifth International Conference on Software Testing, Verification and Validation.

Bron, C., & Kerbosch, J. (1973). Algorithm 457: finding all cliques of an undirected graph. *Commun. ACM, 16*(9), 575-577. doi:10.1145/362342.362367

Bryce, R. C., Lei, Y., Kuhn, D. R., & Kacker, R. (2010). Combinatorial testing *Handbook of Research on Software Engineering and Productivity Technologies: Implications of Globalization* (pp. 196-208): IGI Global.

Bryce, R. C., & Memon, A. M. (2007). *Test suite prioritization by interaction coverage.* Paper presented at the Workshop on

Domain specific approaches to software test automation: in conjunction with the 6th ESEC/FSE joint meeting.

Calvagna, A., Gargantini, A., & Vavassori, P. (2013). *Combinatorial testing for feature models using citlab.* Paper presented at the Sixth International Conference on Software Testing, Verification and Validation Workshops (ICSTW).

Chen, B., & Zhang, J. (2011). *Tuple density: a new metric for combinatorial test suites (NIER track).* Paper presented at the Proceedings of the 33rd International Conference on Software Engineering.

Chen, J., Nair, V., & Menzies, T. (2018). Beyond evolutionary algorithms for search-based software engineering. *Information and Software Technology, 95*, 281-294.

Clements, P., & Northrop, L. (2002). *Software product lines: practices and patterns* (Vol. 59): Addison-Wesley Reading.

Cohen, D. M., Dalal, S. R., Parelius, J., & Patton, G. C. (1996). The combinatorial design approach to automatic test generation. *IEEE software, 13*(5), 83.

Cohen, M. B., Colbourn, C. J., & Ling, A. C. H. (2003a). *Augmenting Simulated Annealing to Build Interaction Test Suites*. Paper presented at the Proceedings of the 14th International Symposium on Software Reliability Engineering.

Cohen, M. B., Dwyer, M. B., & Shi, J. (2006). *Coverage and adequacy in software product line testing.* Paper presented at the Proceedings of the ISSTA 2006 workshop on Role of software architecture for testing and analysis.

Cohen, M. B., Dwyer, M. B., & Shi, J. (2007a). *Interaction testing of highly-configurable systems in the presence of constraints.* Paper presented at the Proceedings of the International Symposium on Software Testing and Analysis.

Cohen, M. B., Dwyer, M. B., & Shi, J. (2007b). *Interaction testing of highly-configurable systems in the presence of constraints.* Paper presented at the Proceedings of the 2007 international symposium on Software testing and analysis.

Cohen, M. B., Dwyer, M. B., & Shi, J. (2008). Constructing interaction test suites for highly-configurable systems in the presence of constraints: A greedy approach. *IEEE Transactions on Software Engineering, 34*(5), 633-650. doi:10.1109/TSE.2008.50

Cohen, M. B., Gibbons, P. B., Mugridge, W. B., Colbourn, C. J., & Collofello, J. S. (2003b, 3-6 Nov. 2003). *A variable strength interaction testing of components.* Paper presented at the Proceedings 27th Annual International Computer Software and Applications Conference. COMPAC 2003.

Czarnecki, K., & Eisenecker, U. W. (2000). *Generative programming: methods, tools, and applications*: ACM Press/Addison-Wesley Publishing Co.

Czarnecki, K., Grünbacher, P., Rabiser, R., Schmid, K., & Wąsowski, A. (2012). *Cool features and tough decisions: a comparison of variability modeling approaches.* Paper presented at the

Proceedings of the Sixth International Workshop on Variability Modeling of Software-Intensive Systems, Leipzig, Germany.

De Bonet, J. S., Isbell Jr, C. L., & Viola, P. A. (1997). *MIMIC: Finding optima by estimating probability densities.* Paper presented at the Advances in neural information processing systems.

Devroey, X., Perrouin, G., Cordy, M., Samih, H., Legay, A., Schobbens, P.-Y., & Heymans, P. (2017). Statistical prioritization for software product line testing: an experience report. *Software & Systems Modeling, 16*(1), 153-171. doi:10.1007/s10270-015-0479-8

Devroey, X., Perrouin, G., & Schobbens, P.-Y. (2014). *Abstract test case generation for behavioural testing of software product lines.* Paper presented at the Proceedings of the 18th International Software Product Line Conference: Companion Volume for Workshops, Demonstrations and Tools - Volume 2, Florence, Italy.

Do, T. B. N., Kitamura, T., Nguyen, V. T., Hatayama, G., Sakuragi, S., & Ohsaki, H. (2013). *Constructing test cases for N-wise testing from tree-based test models.* Paper presented at the Proceedings of the Fourth Symposium on Information and Communication Technology.

Eén, N., & Sörensson, N. (2003). *An extensible SAT-solver.* Paper presented at the International conference on theory and applications of satisfiability testing.

Eiben, A. E., & Smith, J. E. (2015). *Introduction to Evolutionary Computing*: Springer Publishing Company, Incorporated.

Ensan, F., Bagheri, E., & Gašević, D. (2012a). *Evolutionary search-based test generation for software product line feature models.* Paper presented at the Advanced Information Systems Engineering.

Ensan, F., Bagheri, E., & Gašević, D. (2012b). Evolutionary Search-Based Test Generation for Software Product Line Feature Models. In J. Ralyté, X. Franch, S. Brinkkemper, & S. Wrycza (Eds.), *Advanced Information Systems Engineering* (Vol. 7328, pp. 613-628): Springer Berlin Heidelberg.

Etxeberria, R., & Larranaga, P. (1999). *Global optimization using Bayesian networks.* Paper presented at the Second Symposium on Artificial Intelligence (CIMAF-99).

Ferrante, J., Ottenstein, K. J., & Warren, J. D. (1987). The program dependence graph and its use in optimization. *ACM Transactions on Programming Languages and Systems (TOPLAS), 9*(3), 319-349.

Ferreira, T. N., Vergilio, S. R., & de Souza, J. T. (2017). Incorporating user preferences in search-based software engineering: A systematic mapping study. *Information and Software Technology, 90*, 55-69.

Fischer, S., Lopez-Herrejon, R. E., Ramler, R., & Egyed, A. (2016, 16-17 May 2016). *A Preliminary Empirical Assessment of Similarity for Combinatorial Iteraction Testing of Software Product Lines.* Paper

presented at the 2016 IEEE/ACM 9th International Workshop on Search-Based Software Testing (SBST).

Galindo, J. A., Turner, H., Benavides, D., & White, J. (2014). Testing variability-intensive systems using automated analysis: an application to Android. *Software Quality Journal*, 1-41. doi:10.1007/s11219-014-9258-y

Garvin, B. J., Cohen, M. B., & Dwyer, M. B. (2009). *An improved meta-heuristic search for constrained interaction testing.* Paper presented at the First International Symposium on Search Based Software Engineering.

Garvin, B. J., Cohen, M. B., & Dwyer, M. B. (2011). Evaluating improvements to a meta-heuristic search for constrained interaction testing. *Empirical Software Engineering, 16*(1), 61-102. doi:10.1007/s10664-010-9135-7

Haraty, R. A., Mansour, N., & Zeitunlian, H. (2018). Metaheuristic algorithm for state-based software testing. *Applied Artificial Intelligence, 32*(2), 197-213.

Harik, G. R., Lobo, F. G., & Sastry, K. (2006). Linkage Learning via Probabilistic Modeling in the Extended Compact Genetic Algorithm (ECGA). In M. Pelikan, K. Sastry, & E. CantúPaz (Eds.), *Scalable optimization via probabilistic modeling* (pp. 39-61). Berlin, Heidelberg: Springer Berlin Heidelberg.

Harman, M., Jia, Y., Krinke, J., Langdon, W. B., Petke, J., & Zhang, Y. (2014). *Search based software engineering for software product line engineering: a survey and directions for future work.* Paper presented at the Proceedings of the 18th International Software Product Line Conference - Volume 1, Florence, Italy.

Harman, M., & Jones, B. F. (2001). Search-based software engineering. *Information and Software Technology, 43*(14), 833-839.

Harman, M., Mansouri, S. A., & Zhang, Y. (2012). Search-based software engineering: Trends, techniques and applications. *ACM Comput. Surv., 45*(1), 1-61. doi:10.1145/2379776.2379787

Hasan, I. H., Ahmed, B. S., Potrus, M. Y., & Zamli, K. Z. (2020). Generation and Application of Constrained Interaction Test Suites Using Base Forbidden Tuples with a Mixed Neighborhood Tabu Search. *International Journal of Software Engineering and Knowledge Engineering, 30*(03), 363-398.

Haslinger, E. N., Lopez-Herrejon, R. E., & Egyed, A. (2013a). Improving CASA runtime performance by exploiting basic feature model analysis. *arXiv preprint arXiv:1311.7313*.

Haslinger, E. N., Lopez-Herrejon, R. E., & Egyed, A. (2013b). *Using feature model knowledge to speed up the generation of covering arrays.* Paper presented at the Proceedings of the Seventh International Workshop on Variability Modelling of Software-intensive Systems.

Hauschild, M., & Pelikan, M. (2011). An introduction and survey of estimation of distribution algorithms. *Swarm and Evolutionary Computation, 1*(3), 111-128. doi:http://dx.doi.org/10.1016/j.swevo.2011.08.003

Henard, C., Papadakis, M., & Le Traon, Y. (2014a). Mutation-based generation of software product line test configurations *Search-Based Software Engineering* (pp. 92-106): Springer.

Henard, C., Papadakis, M., Perrouin, G., Klein, J., Heymans, P., & le Traon, Y. (2014b). Bypassing the Combinatorial Explosion: Using Similarity to Generate and Prioritize T-Wise Test Configurations for Software Product Lines. *Software Engineering, IEEE Transactions on, 40*(7), 650-670. doi:http://doi.org/10.1109/TSE.2014.2327020

Henard, C., Papadakis, M., Perrouin, G., Klein, J., & Traon, Y. L. (2013). *Multi-objective test generation for software product lines.* Paper presented at the Proceedings of the 17th International Software Product Line Conference.

Hervieu, A., Baudry, B., & Gotlieb, A. (2011, Nov. 29 2011-Dec. 2 2011). *PACOGEN: Automatic Generation of Pairwise Test Configurations from Feature Models.* Paper presented at the IEEE 22nd International Symposium on Software Reliability Engineering (ISSRE).

Horcas, J.-M., Pinto, M., & Fuentes, L. (2019). *Software product line engineering: a practical experience.* Paper presented at the Proceedings of the 23rd International Systems and Software Product Line Conference-Volume A.

Huang, R., Chen, J., Zhang, T., Wang, R., & Lu, Y. (2013). *Prioritizing variable-strength covering array.* Paper presented at the 2013 IEEE 37th Annual Computer Software and Applications Conference.

Huang, R., Zong, W., Chen, T. Y., Towey, D., Chen, J., Zhou, Y., & Sun, W. (2018). *On the selection of strength for fixed-strength interaction coverage based prioritization.* Paper presented at the 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC).

Jamil, M. A., Alhindi, A., Arif, M., Nour, M. K., Abubakar, N. S. A., & Aljabri, T. F. (2019). *Multiobjective Evolutionary Algorithms NSGA-II and NSGA-III for Software Product Lines Testing Optimization.* Paper presented at the 2019 IEEE 6th International Conference on Engineering Technologies and Applied Sciences (ICETAS).

Jan, S., Panichella, A., Arcuri, A., & Briand, L. (2019). Search-based multi-vulnerability testing of XML injections in web applications. *Empirical Software Engineering, 24*(6), 3696-3729.

Jeffrey, D., & Gupta, N. (2007). Improving fault detection capability by selectively retaining test cases during test suite reduction. *IEEE Transactions on Software Engineering, 33*(2).

Johansen, M., Haugen, Ø., & Fleurey, F. (2011). Properties of Realistic Feature Models Make Combinatorial Testing of Product Lines Feasible. In J. Whittle, T. Clark, & T. Kühne (Eds.), *Model Driven Engineering Languages and Systems* (Vol. 6981, pp. 638-652): Springer Berlin Heidelberg.

Johansen, M. F., Haugen, Ø., & Fleurey, F. (2012a). *An algorithm for generating t-wise covering arrays from large feature models.* Paper presented at the Proceedings of the 16th International Software Product Line Conference-Volume 1.

Johansen, M. F., Haugen, Ø., & Fleurey, F. (2012b). *Bow tie testing: a testing pattern for product lines.* Paper presented at the Proceedings of the 16th European Conference on Pattern Languages of Programs.

Johansen, M. F., Haugen, Ø., Fleurey, F., Carlson, E., Endresen, J., & Wien, T. (2012c). A technique for agile and automatic interaction testing for product lines *Testing Software and Systems* (pp. 39-54): Springer.

Johansen, M. F., Haugen, Ø., Fleurey, F., Eldegard, A. G., & Syversen, T. (2012d). *Generating better partial covering arrays by modeling weights on sub-product lines*: Springer.

Kacker, R. N., Kuhn, D. R., Lei, Y., & Lawrence, J. F. (2013). Combinatorial testing for software: An adaptation of design of experiments. *Measurement, 46*(9), 3745-3752.

Kalaee, A., & Rafe, V. (2019). Model-based test suite generation for graph transformation system using model simulation and search-based techniques. *Information and Software Technology, 108*, 1-29.

Khari, M., & Kumar, P. (2019). An extensive evaluation of search-based software testing: a review. *Soft Computing, 23*(6), 1933-1946.

Kim, C. H. P., Batory, D. S., & Khurshid, S. (2011). *Reducing combinatorics in testing product lines.* Paper presented at the Proceedings of the tenth international conference on Aspect-oriented software development, Porto de Galinhas, Brazil.

Kitchenham, B. A., & Charters, S. (2007). *Guidelines for performing Systematic Literature Reviews in Software Engineering*. Retrieved from

Knüppel, A., Thüm, T., Mennicke, S., Meinicke, J., & Schaefer, I. (2017). *Is there a mismatch between real-world feature models and product-line research?* Paper presented at the Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering.

Kowal, M., Schulze, S., & Schaefer, I. (2013). *Towards efficient SPL testing by variant reduction.* Paper presented at the Proceedings of the 4th international workshop on Variability & composition.

Kuhn, D. R., Wallace, D. R., & AM Gallo, J. (2004). Software fault interactions and implications for software testing. *Software Engineering, IEEE Transactions on, 30*(6), 418-421.

Kuhn, R., Kacker, R., Lei, Y., & Hunter, J. (2009). Combinatorial Software Testing. *Computer, 42*(8), 94-96. doi:10.1109/MC.2009.253

Kumar, L., Rath, S., & Sureka, A. (2017). *An empirical analysis on effective fault prediction model developed using ensemble methods.* Paper presented at the 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC).

Lackner, H., & Schlingloff, B.-H. (2017). Advances in testing software product lines *Advances in computers* (Vol. 107, pp. 157-217): Elsevier.

Lamancha, B. P., Polo, M., & Piattini, M. (2013). Systematic review on software product line testing *Software and Data Technologies* (pp. 58-71): Springer.

Lamancha, B. P., Polo, M., & Piattini, M. (2015). PROW: A Pairwise algorithm with constRaints, Order and Weight. *Journal of Systems and Software, 99*, 1-19. doi:10.1016/j.jss.2014.08.005

Larrañaga, P., Karshenas, H., Bielza, C., & Santana, R. (2012). A review on probabilistic graphical models in evolutionary computation. *Journal of Heuristics, 18*(5), 795-819. doi:10.1007/s10732-012-9208-4

Larrañaga, P., & Lozano, J. A. (2001). *Estimation of distribution algorithms: A new tool for evolutionary computation* (Vol. 2): Springer Science & Business Media.

Le Berre, D., & Parrain, A. (2008). *On SAT technologies for dependency management and beyond.*

Lee, K., Kang, K., & Lee, J. (2002). Concepts and Guidelines of Feature Modeling for Product Line Software Engineering. In C. Gacek (Ed.), *Software Reuse: Methods, Techniques, and Tools* (Vol. 2319, pp. 62-77): Springer Berlin Heidelberg.

Lei, Y., Kacker, R., Kuhn, D. R., Okun, V., & Lawrence, J. (2007). *IPOG: A general strategy for t-way software testing.* Paper presented at the 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'07).

Li, X., Wong, W. E., Gao, R., Hu, L., & Hosono, S. (2018). Genetic algorithm-based test generation for software product line with the integration of fault localization techniques. *Empirical Software Engineering, 23*(1), 1-51.

Lindohf, R., Krüger, J., Herzog, E., & Berger, T. (2020). Software Product-Line Evaluation in the Large. *Empirical Software Engineering.*

Lochau, M., Oster, S., Goltz, U., & Schürr, A. (2012). Model-based pairwise testing for feature interaction coverage in software product line engineering. *Software Quality Journal, 20*(3-4), 567-604. doi:10.1007/s11219-011-9165-4

Lopez-Herrejon, R. E., Chicano, F., Ferrer, J., Egyed, A., & Alba, E. (2013). *Multi-objective optimal test suite computation for software product line pairwise testing.* Paper presented at the International Conference on Software Maintenance (ICSM).

Lopez-Herrejon, R. E., Ferrer, J., Chicano, F., Egyed, A., & Alba, E. (2014a). *Comparative analysis of classical multi-objective evolutionary algorithms and seeding strategies for pairwise testing of software product lines.* Paper presented at the Congress on Evolutionary Computation (CEC).

Lopez-Herrejon, R. E., Fischer, S., Ramler, R., & Egyed, A. (2015). *A first systematic mapping study on combinatorial interaction testing for software product lines.* Paper presented at the Eighth

International Conference on Software Testing, Verification and Validation Workshops (ICSTW).

Lopez-Herrejon, R. E., Javier Ferrer, J., Chicano, F., Haslinger, E. N., Egyed, A., & Alba, E. (2014b). *A parallel evolutionary algorithm for prioritized pairwise testing of software product lines.* Paper presented at the Proceedings of the 2014 conference on Genetic and evolutionary computation.

Marijan, D., Gotlieb, A., Sen, S., & Hervieu, A. (2013). *Practical pairwise testing for software product lines.* Paper presented at the Proceedings of the 17th international software product line conference.

Markiegi, U., Arrieta, A., Sagardui, G., & Etxeberria, L. (2017). *Search-based product line fault detection allocating test cases iteratively.* Paper presented at the Proceedings of the 21st International Systems and Software Product Line Conference-Volume A.

Martins, M. S., Delgado, M. R., Santana, R., Lüders, R., Gonçalves, R. A., & Almeida, C. P. d. (2016). *HMOBEDA: Hybrid Multi-objective Bayesian estimation of distribution algorithm.* Paper presented at the Proceedings of the Genetic and Evolutionary Computation Conference 2016.

McGregor, J. (2001). *Testing a software product line*. Retrieved from

Meinicke, J., Thüm, T., Schröter, R., Benduhn, F., Leich, T., & Saake, G. (2017). Feature Modeling *Mastering Software Variability with FeatureIDE* (pp. 43-62). Cham: Springer International Publishing.

Mendenhall, W., Beaver, R. J., & Beaver, B. M. (2012). *Introduction to probability and statistics*: Cengage Learning.

Mendonca, M., Branco, M., & Cowan, D. (2009a). *SPLOT: software product lines online tools.* Paper presented at the Proceedings of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications.

Mendonça, M., Cowan, D., Malyk, W., & Oliveira, T. (2008). Collaborative Product Configuration. *Journal of Software, 3*(2), 69.

Mendonca, M., Wąsowski, A., & Czarnecki, K. (2009b). *SAT-based analysis of feature models is easy*. Paper presented at the Proceedings of the 13th International Software Product Line Conference, San Francisco, California, USA.

Nešić, D., Krüger, J., Stănciulescu, Ş., & Berger, T. (2019). *Principles of feature modeling.* Paper presented at the Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering.

Neto, P. A. D. M. S., do Carmo Machado, I., McGregor, J. D., De Almeida, E. S., & de Lemos Meira, S. R. (2011). A systematic mapping study of software product lines testing. *Information and Software Technology, 53*(5), 407-423.

Nie, C., & Leung, H. (2011). A survey of combinatorial testing. *ACM Computing Surveys (CSUR), 43*(2), 1-29. doi:10.1145/1883612.1883618

Nie, C., Xu, B., Shi, L., & Dong, G. (2005). Automatic test generation for n-way combinatorial testing *Quality of Software Architectures and Software Quality* (pp. 203-211): Springer.

Olaechea, R., Rayside, D., Guo, J., & Czarnecki, K. (2014). *Comparison of exact and approximate multi-objective optimization for software product lines.* Paper presented at the Proceedings of the 18th International Software Product Line Conference-Volume 1.

Oster, S., Markert, F., & Ritter, P. (2010). Automated Incremental Pairwise Testing of Software Product Lines. In J. Bosch & J. Lee (Eds.), *Software Product Lines: Going Beyond* (Vol. 6287, pp. 196-210): Springer Berlin Heidelberg.

Oster, S., Zink, M., Lochau, M., & Grechanik, M. (2011a). *Pairwise feature-interaction testing for SPLs: potentials and limitations.* Paper presented at the Proceedings of the 15th International Software Product Line Conference.

Oster, S., Zorcic, I., Markert, F., & Lochau, M. (2011b). *MoSo-PoLiTe: tool support for pairwise and model-based software product line testing.* Paper presented at the Proceedings of the 5th Workshop on Variability Modeling of Software-Intensive Systems.

Ottenstein, K. J., & Ottenstein, L. M. (1984). The program dependence graph in a software development environment. *SIGPLAN Not., 19*(5), 177-184. doi:10.1145/390011.808263

Papadakis, M., Henard, C., & Traon, Y. L. (2014). *Sampling program inputs with mutation analysis: Going beyond combinatorial interaction testing.* Paper presented at the Software Testing, Verification and Validation (ICST), 2014 IEEE Seventh International Conference on.

Patel, S., Gupta, P., & Shah, V. (2013). *Feature interaction testing of variability intensive systems.* Paper presented at the 4th International Workshop on Product Line Approaches in Software Engineering (PLEASE).

Pelikan, M., Goldberg, D. E., Cant, E. E., #250, & -paz. (2000). Linkage Problem, Distribution Estimation, and Bayesian Networks. *Evol. Comput., 8*(3), 311-340. doi:10.1162/106365600750078808

Pelikan, M., Hauschild, M., & Lobo, F. (2015). Estimation of Distribution Algorithms. In J. Kacprzyk & W. Pedrycz (Eds.), *Springer Handbook of Computational Intelligence* (pp. 899-928): Springer Berlin Heidelberg.

Pelikan, M., & Mühlenbein, H. (1998). *Marginal distributions in evolutionary algorithms.* Paper presented at the Proceedings of the International Conference on Genetic Algorithms Mendel.

Pelikan, M., & Mühlenbein, H. (1999). The bivariate marginal distribution algorithm *Advances in Soft Computing* (pp. 521-535): Springer.

Pérez Lamancha, B., & Polo Usaola, M. (2010). Testing Product Generation in Software Product Lines Using Pairwise for

Features Coverage. In A. Petrenko, A. Simão, & J. Maldonado (Eds.), *Testing Software and Systems* (Vol. 6435, pp. 111-125): Springer Berlin Heidelberg.

Perrouin, G., Oster, S., Sen, S., Klein, J., Baudry, B., & Le Traon, Y. (2012). Pairwise testing for software product lines: comparison of two approaches. *Software Quality Journal, 20*(3-4), 605-643. doi:10.1007/s11219-011-9160-9

Perrouin, G., Sen, S., Klein, J., Baudry, B., & le Traon, Y. (2010, 6-10 April 2010). *Automated and Scalable T-wise Test Case Generation Strategies for Software Product Lines.* Paper presented at the Software Testing, Verification and Validation (ICST), 2010 Third International Conference on.

Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M. (2008). *Systematic mapping studies in software engineering.* Paper presented at the 12th International Conference on Evaluation and Assessment in Software Engineering (EASE). http://ewic.bcs.org/content/ConWebDoc/19543

Petke, J., Cohen, M. B., Harman, M., & Yoo, S. (2015). Practical combinatorial interaction testing: Empirical findings on efficiency and early fault detection. *IEEE Transactions on Software Engineering, 41*(9), 901-924.

Petke, J., Yoo, S., Cohen, M. B., & Harman, M. (2013). *Efficiency and early fault detection with lower and higher strength combinatorial interaction testing.* Paper presented at the Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering.

Pohl, K., Böckle, G., & Van Der Linden, F. (2005). Software product line engineering. *Springer, 10,* 72-86.

Qi, R.-Z., Wang, Z.-J., & Li, S.-Y. (2016). A Parallel Genetic Algorithm Based on Spark for Pairwise Test Suite Generation. *Journal of Computer Science and Technology, 31*(2), 417-427. doi:10.1007/s11390-016-1635-5

Ramirez, A., Romero, J. R., & Simons, C. L. (2018). A systematic review of interaction in search-based software engineering. *IEEE Transactions on Software Engineering, 45*(8), 760-781.

Ramirez, A., Romero, J. R., & Ventura, S. (2019). A survey of many-objective optimisation in search-based software engineering. *Journal of Systems and Software, 149*, 382-395.

Reis, S., Metzger, A., & Pohl, K. (2007). Integration Testing in Software Product Line Engineering: A Model-Based Technique. In M. Dwyer & A. Lopes (Eds.), *Fundamental Approaches to Software Engineering* (Vol. 4422, pp. 321-335): Springer Berlin Heidelberg.

Sabharwal, S., & Aggarwal, M. (2015). *Variable strength interaction test set generation using Multi Objective Genetic Algorithms.* Paper presented at the Advances in Computing, Communications and Informatics (ICACCI), 2015 International Conference on.

Saeed, A., Ab Hamid, S. H., & Mustafa, M. B. (2016). The experimental applications of search-based techniques for model-based

testing: Taxonomy and systematic literature review. *Applied Soft Computing, 49*, 1094-1117.

Sánchez, A. B., Segura, S., Parejo, J. A., & Ruiz-Cortés, A. (2015). Variability testing in the wild: the Drupal case study. *Software & Systems Modeling*, 1-22. doi:10.1007/s10270-015-0459-z

Sánchez, A. B., Segura, S., & Ruiz-Cortés, A. (2014a). *A comparison of test case prioritization criteria for software product lines.* Paper presented at the Seventh International Conference on Software Testing, Verification and Validation (ICST).

Sánchez, A. B., Segura, S., & Ruiz-Cortés, A. (2014b). *The Drupal framework: a case study to evaluate variability testing techniques.* Paper presented at the Proceedings of the Eighth International Workshop on Variability Modelling of Software-Intensive Systems.

Santana, R. (2005). Estimation of distribution algorithms with Kikuchi approximations. *Evolutionary Computation, 13*(1), 67-97.

Santana, R., Larrañaga, P., & Lozano, J. A. (2010). Learning factorizations in estimation of distribution algorithms using affinity propagation. *Evolutionary Computation, 18*(4), 515-546.

Scheidemann, K. D. (2006). *Optimizing the Selection of Representative Configurations in Verification of Evolving Product Lines of Distributed Embedded Systems.* Paper presented at the Proceedings of the 10th International on Software Product Line Conference.

Segura, S., Galindo, J. A., Benavides, D., Parejo, J. A., & Ruiz-Cortés, A. (2012). *BeTTy: benchmarking and testing on the automated analysis of feature models.* Paper presented at the Proceedings of the Sixth International Workshop on Variability Modeling of Software-Intensive Systems.

Segura, S., Parejo, J. A., Hierons, R. M., Benavides, D., & Ruiz-Cortés, A. (2014). Automated generation of computationally hard feature models using evolutionary algorithms. *Expert Systems with Applications, 41*(8), 3975-3992.

Sen, S., & Gotlieb, A. (2013). *Testing a data-intensive system with generated data interactions.* Paper presented at the Advanced Information Systems Engineering.

Shakya, S., Brownlee, A., McCall, J., Fournier, F., & Owusu, G. (2009). *A fully multivariate DEUM algorithm.* Paper presented at the Evolutionary Computation, 2009. CEC'09. IEEE Congress on.

Shakya, S., & Santana, R. (2012). A Review of Estimation of Distribution Algorithms and Markov Networks. In S. Shakya & R. Santana (Eds.), *Markov Networks in Evolutionary Computation* (Vol. 14, pp. 21-37): Springer Berlin Heidelberg.

She, S., Lotufo, R., Berger, T., Wąsowski, A., & Czarnecki, K. (2011). *Reverse engineering feature models.* Paper presented at the Proceedings of the 33rd International Conference on Software Engineering.

Silva, R. A., de Souza, S. d. R. S., & de Souza, P. S. L. (2017). A systematic review on search based mutation testing. *Information and Software Technology, 81*, 19-35.

Simon, D. (2013a). Estimation of Distribution Algorithms *Evolutionary Optimization Algorithms* (pp. 313-347): John Wiley & Sons.

Simon, D. (2013b). *Evolutionary Optimization Algorithms*: Wiley.

Sisodia, R. S., & Channakeshava, V. (2009). *Combinatorial approach for automated platform diversity testing.* Paper presented at the Fourth International Conference on Software Engineering Advances (ICSEA).

Song, C., Porter, A., & Foster, J. S. (2013). iTree: efficiently discovering high-coverage configurations using interaction trees. *IEEE Transactions on Software Engineering, 40*(3), 251-265.

Sorensson, N., & Een, N. (2005). Minisat v1. 13-a sat solver with conflict-clause minimization. *SAT, 2005*(53), 1-2.

Steffens, M., Oster, S., Lochau, M., & Fogdal, T. (2012). *Industrial evaluation of pairwise spl testing with moso-polite.* Paper presented at the Proceedings of the Sixth International Workshop on Variability Modeling of Software-Intensive Systems.

Strasser, S., Sheppard, J., & Butcher, S. (2017). *A formal approach to deriving factored evolutionary algorithm architectures.* Paper presented at the 2017 IEEE Symposium Series on Computational Intelligence (SSCI).

Thaker, S., Batory, D., Kitchin, D., & Cook, W. (2007). *Safe composition of product lines.* Paper presented at the Proceedings of the 6th international conference on Generative programming and component engineering.

Thiel, S., & Peruzzi, F. (2000). *Starting a product line approach for an envisioned market: research and experience in an industrial environment.* Paper presented at the Proceedings of the first conference on Software product lines : experience and research directions, Denver, Colorado, USA.

Thüm, T., Kästner, C., Benduhn, F., Meinicke, J., Saake, G., & Leich, T. (2014). FeatureIDE: An extensible framework for feature-oriented software development. *Science of Computer Programming, 79*, 70-85.

Tsukiyama, S., Ide, M., Ariyoshi, H., & Shirakawa, I. (1977). A new algorithm for generating all the maximal independent sets. *SIAM Journal on Computing, 6*(3), 505-517.

Wang, S., Ali, S., & Gotlieb, A. (2013). *Minimizing test suites in software product lines using weight-based genetic algorithms.* Paper presented at the Proceedings of the 15th annual conference on Genetic and evolutionary computation, Amsterdam, The Netherlands.

Wang, S., Ali, S., & Gotlieb, A. (2015a). Cost-effective test suite minimization in product lines using search techniques. *Journal of Systems and Software, 103*, 370-391. doi:10.1016/j.jss.2014.08.024

Wang, S., Ali, S., & Gotlieb, A. (2015b). Cost-effective test suite minimization in product lines using search techniques. *Journal of Systems and Software, 103*(0), 370-391. doi:http://dx.doi.org/10.1016/j.jss.2014.08.024

Wang, S., Ali, S., Gotlieb, A., & Liaaen, M. (2014). A systematic test case selection methodology for product lines: results and insights from an industrial case study. *Empirical Software Engineering*, 1-37. doi:10.1007/s10664-014-9345-5

Weiss, D. M., & Lai, C. T. R. (1999). *Software product-line engineering: a family-based software development process*: Addison-Wesley Longman Publishing Co., Inc.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M., Regnell, B., & Wesslén, A. (2012a). Experiment Process *Experimentation in Software Engineering* (pp. 73-81): Springer Berlin Heidelberg.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2012b). *Experimentation in software engineering*: Springer Science & Business Media.

Ye, H., & Liu, H. (2005). Approach to modelling feature variability and dependencies in software product lines. *IEE Proceedings-software, 152*(3), 101-109.

Yilmaz, C., Cohen, M. B., & Porter, A. A. (2006). Covering arrays for efficient fault characterization in complex configuration spaces. *IEEE Transactions on Software Engineering, 32*(1), 20-34.

Yilmaz, C., Fouche, S., Cohen, M. B., Porter, A., Demiroz, G., & Koc, U. (2014). Moving forward with combinatorial interaction testing. *Computer, 47*(2), 37-45.

Yu, L., Duan, F., Lei, Y., Kacker, R. N., & Kuhn, D. R. (2014a). *Combinatorial Test Generation for Software Product Lines Using Minimum Invalid Tuples.* Paper presented at the 15th International Symposium on High-Assurance Systems Engineering (HASE).

Yu, L., Duan, F., Lei, Y., Kacker, R. N., & Kuhn, D. R. (2014b). *Combinatorial test generation for software product lines using minimum invalid tuples.* Paper presented at the High-Assurance Systems Engineering (HASE), 2014 IEEE 15th International Symposium on.

Yu, T.-L., Goldberg, D. E., Yassine, A., & Chen, Y.-P. (2003). Genetic Algorithm Design Inspired by Organizational Theory: Pilot Study of a Dependency Structure Matrix Driven Genetic Algorithm. In E. Cantú-Paz, J. A. Foster, K. Deb, L. D. Davis, R. Roy, U.-M. O'Reilly, H.-G. Beyer, R. Standish, G. Kendall, S. Wilson, M. Harman, J. Wegener, D. Dasgupta, M. A. Potter, A. C. Schultz, K. A. Dowsland, N. Jonoska, & J. Miller (Eds.), *Genetic and Evolutionary Computation — GECCO 2003: Genetic and Evolutionary Computation Conference Chicago, IL, USA, July 12–16, 2003 Proceedings, Part II* (pp. 1620-1621). Berlin, Heidelberg: Springer Berlin Heidelberg.

Zhao, X., Li, Y., Liu, L., Zheng, J., Liu, Y., & He, X. (2017). A Combination Test Suite Generation Method Based on Adaptive

Simulated Annealing Genetic Algorithm for Software Product Line Testing. *DEStech Transactions on Engineering and Technology Research*(apetc).

Zobel, J. (2004). *Writing for computer science* (Vol. 8): Springer.