

	Publication	Input Data	Algorithm	Coverage										Evaluation						Application																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								
	feature model domain knowledge code artifacts test artifacts product set greedy evolutionary manual selection feature-wise coverage pair-wise coverage 3-wise coverage 4-wise coverage 5-wise coverage 6-wise coverage t-wise coverage statement coverage block coverage requirements coverage no coverage guarantee sampling efficiency testing efficiency effectiveness no tool tool unavailable available tool open-source tool evaluation no evaluation testing type checking data-flow analysis non-functional properties																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											

References

- [1] I. Abal, C. Brabrand, and A. Wasowski. 42 Variability Bugs in the Linux Kernel: A Qualitative Analysis. In *Proc. Int'l Conf. Automated Software Engineering (ASE)*, pages 421–432. ACM, 2014.
- [2] I. Abal, J. Melo, S. Stănculescu, C. Brabrand, M. Ribeiro, and A. Wasowski. Variability Bugs in Highly Configurable Systems: A Qualitative Analysis. *Trans. Software Engineering and Methodology (TOSEM)*, 26(3):10:1–10:34, 2018.
- [3] M. Al-Hajjaji, S. Krieter, T. Thüm, M. Lochau, and G. Saake. IncLing: Efficient Product-line Testing Using Incremental Pairwise Sampling. In *Proc. Int'l Conf. Generative Programming: Concepts & Experiences (GPCE)*, pages 144–155. ACM, 2016.
- [4] M. Al-Hajjaji, S. Lity, R. Lachmann, T. Thüm, I. Schaefer, and G. Saake. Delta-Oriented Product Prioritization for Similarity-Based Product-Line Testing. In *Proc. Int'l Workshop Variability and Complexity in Software Design (VACE)*, pages 34–40. IEEE, 2017.
- [5] M. Al-Hajjaji, J. Meinicke, S. Krieter, R. Schröter, T. Thüm, T. Leich, and G. Saake. Tool Demo: Testing Configurable Systems with FeatureIDE. In *Proc. Int'l Conf. Generative Programming: Concepts & Experiences (GPCE)*, pages 173–177. ACM, 2016.
- [6] M. Al-Hajjaji, T. Thüm, M. Lochau, J. Meinicke, and G. Saake. Effective Product-Line Testing Using Similarity-Based Product Prioritization. *Software and System Modeling*, 2018. To appear.
- [7] P. Arcaini, A. Gargantini, and P. Vavassori. Generating Tests for Detecting Faults in Feature Models. In *Proc. Int'l Conf. Software Testing, Verification and Validation (ICST)*, pages 1–10. IEEE, 2015.
- [8] H. Baller, S. Lity, M. Lochau, and I. Schaefer. Multi-Objective Test Suite Optimization for Incremental Product Family Testing. In *Proc. Int'l Conf. Software Testing, Verification and Validation (ICST)*, pages 303–312. IEEE, 2014.
- [9] A. Cmyrev and R. Reissing. Efficient and Effective Testing of Automotive Software Product Lines. *Int'l J. Applied Science and Technology (IJAST)*, 7(2), 2014.
- [10] M. B. Cohen, M. B. Dwyer, and J. Shi. Coverage and Adequacy in Software Product Line Testing. In *Proc. Int'l Workshop Role of Software Architecture for Testing and Analysis (ROSATEA)*, pages 53–63. ACM, 2006.
- [11] X. Devroey, G. Perrouin, A. Legay, P.-Y. Schobbens, and P. Heymans. Covering SPL Behaviour with Sampled Configurations: An Initial Assessment. In *Proc. Int'l Workshop Variability Modelling of Software-Intensive Systems (VaMoS)*, pages 59:59–59:66. ACM, 2015.
- [12] A. Ensan, E. Bagheri, M. Asadi, D. Gasevic, and Y. Biletskiy. Goal-Oriented Test Case Selection and Prioritization for Product Line Feature Models. In *Proc. Int'l Conf. Information Technology: New Generations (ITNG)*, pages 291–298. IEEE, 2011.
- [13] F. Ensan, E. Bagheri, and D. Gasevic. Evolutionary Search-Based Test Generation for Software Product Line Feature Models. In *Proc. Int'l Conf. Advanced Information Systems Engineering (CAiSE)*, volume 7328, pages 613–628. Springer, 2012.
- [14] T. d. N. Ferreira, J. N. Kuk, A. Pozo, and S. R. Vergilio. Product Selection Based on Upper Confidence Bound MOEA/D-DR for Testing Software Product Lines. In *Proc. Congress Evolutionary Computation (CEC)*, pages 4135–4142. IEEE, 2016.
- [15] T. N. Ferreira, J. A. P. Lima, A. Strickler, J. N. Kuk, S. R. Vergilio, and A. Pozo. Hyper-Heuristic Based Product Selection for Software Product Line Testing. *Comp. Intell. Mag. (CIM)*, 12(2):34–45, 2017.
- [16] H. L. J. Filho, J. A. P. Lima, and S. R. Vergilio. Automatic Generation of Search-Based Algorithms Applied to the Feature Testing of Software Product Lines. In *Proc. Brazilian Symposium Software Engineering (SBES)*, pages 114–123. ACM, 2017.
- [17] R. A. M. Filho and S. R. Vergilio. A Mutation and Multi-Objective Test Data Generation Approach for Feature Testing of Software Product Lines. In *Proc. Brazilian Symposium Software Engineering (SBES)*, pages 21–30. IEEE, 2015.
- [18] S. Fischer, R. E. Lopez-Herrejon, R. Ramler, and A. Egyed. A Preliminary Empirical Assessment of Similarity for Combinatorial Interaction Testing of Software Product Lines. In *Proc. Int'l Workshop on Search-Based Software Testing (SBST)*, pages 15–18. ACM, 2016.
- [19] B. J. Garvin, M. B. Cohen, and M. B. Dwyer. Evaluating Improvements to a Meta-Heuristic Search for Constrained Interaction Testing. *Empirical Software Engineering (EMSE)*, 16(1):61–102, 2011.

- [20] A. Grebhahn, S. Kuckuk, C. Schmitt, H. Köstler, N. Siegmund, S. Apel, F. Hannig, and J. Teich. Experiments on Optimizing the Performance of Stencil Codes with SPL Conqueror. *Parallel Processing Letters*, 24(3), 2014.
- [21] A. Grebhahn, C. Rodrigo, N. Siegmund, F. J. Gaspar, and S. Apel. Performance-Influence Models of Multigrid Methods: A Case Study on Triangular Grids. *Concurrency and Computation: Practice and Experience*, 29(17), 2017.
- [22] J. Guo, K. Czarnecki, S. Apel, N. Siegmund, and A. Wasowski. Variability-Aware Performance Prediction: A Statistical Learning Approach. In *Proc. Int’l Conf. Automated Software Engineering (ASE)*, pages 301–311. IEEE, 2013.
- [23] J. Guo, D. Yang, N. Siegmund, S. Apel, A. Sarkar, P. Valov, K. Czarnecki, A. Wasowski, and H. Yu. Data-Efficient Performance Learning for Configurable Systems. *Empirical Software Engineering (EMSE)*, 2018. To appear.
- [24] A. Halin, A. Nuttinck, M. Acher, X. Devroey, G. Perrouin, and B. Baudry. Test Them All, Is It worth It? A Ground Truth Comparison of Configuration Sampling Strategies. Technical Report arXiv:1710.07980, Cornell University Library, 2017.
- [25] E. N. Haslinger, R. E. Lopez-Herrejon, and A. Egyed. Using Feature Model Knowledge to Speed Up the Generation of Covering Arrays. In *Proc. Int’l Workshop Variability Modelling of Software-Intensive Systems (VaMoS)*, pages 16:1–16:6. ACM, 2013.
- [26] C. Henard, M. Papadakis, and Y. Le Traon. Mutation-Based Generation of Software Product Line Test Configurations. In C. Le Goues and S. Yoo, editors, *Search-Based Software Engineering*, volume 8636 of *Lecture Notes in Computer Science*, pages 92–106. Springer International Publishing, 2014.
- [27] C. Henard, M. Papadakis, G. Perrouin, J. Klein, P. Heymans, and Y. L. Traon. Bypassing the Combinatorial Explosion: Using Similarity to Generate and Prioritize T-Wise Test Configurations for Software Product Lines. *IEEE Trans. Software Engineering (TSE)*, 40(7):650–670, 2014.
- [28] C. Henard, M. Papadakis, G. Perrouin, J. Klein, and Y. L. Traon. Multi-Objective Test Generation for Software Product Lines. In *Proc. Int’l Software Product Line Conf. (SPLC)*, pages 62–71. ACM, 2013.
- [29] M. F. Johansen, Ø. Haugen, and F. Fleurey. Properties of Realistic Feature Models Make Combinatorial Testing of Product Lines Feasible. In *Proc. Int’l Conf. Model Driven Engineering Languages and Systems (MODELS)*, pages 638–652. Springer, 2011.
- [30] M. F. Johansen, Ø. Haugen, and F. Fleurey. An Algorithm for Generating T-Wise Covering Arrays from Large Feature Models. In *Proc. Int’l Software Product Line Conf. (SPLC)*, pages 46–55. ACM, 2012.
- [31] M. F. Johansen, Ø. Haugen, F. Fleurey, A. G. Eldegard, and T. Syversen. Generating Better Partial Covering Arrays by Modeling Weights on Sub-Product Lines. In *Proc. Int’l Conf. Model Driven Engineering Languages and Systems (MODELS)*, pages 269–284. Springer, 2012.
- [32] C. H. P. Kim, D. Batory, and S. Khurshid. Eliminating Products to Test in a Software Product Line. In *Proc. Int’l Conf. Automated Software Engineering (ASE)*, pages 139–142. ACM, 2010.
- [33] C. H. P. Kim, D. Batory, and S. Khurshid. Reducing Combinatorics in Testing Product Lines. In *Proc. Int’l Conf. Aspect-Oriented Software Development (AOSD)*, pages 57–68. ACM, 2011.
- [34] C. H. P. Kim, E. Bodden, D. Batory, and S. Khurshid. Reducing Configurations to Monitor in a Software Product Line. In *Proc. Int’l Conf. Runtime Verification (RV)*, pages 285–299. Springer, 2010.
- [35] C. H. P. Kim, D. Marinov, S. Khurshid, D. Batory, S. Souto, P. Barros, and M. D’Amorim. SPLat: Lightweight Dynamic Analysis for Reducing Combinatorics in Testing Configurable Systems. In *Proc. Europ. Software Engineering Conf./Foundations of Software Engineering (ESEC/FSE)*, pages 257–267. ACM, 2013.
- [36] M. Kowal, S. Schulze, and I. Schaefer. Towards Efficient SPL Testing by Variant Reduction. In *Proce. Int’l workshop on Variability & composition (VariComp)*, pages 1–6. ACM, 2013.
- [37] J. Liebig, A. von Rhein, C. Kästner, S. Apel, J. Dörre, and C. Lengauer. Scalable Analysis of Variable Software. In *Proc. Europ. Software Engineering Conf./Foundations of Software Engineering (ESEC/FSE)*, pages 81–91. ACM, 2013.
- [38] R. E. Lopez-Herrejon, J. Ferrer, F. Chicano, A. Egyed, and E. Alba. Comparative Analysis of Classical Multi-Objective Evolutionary Algorithms and Seeding Strategies for Pairwise Testing of Software Product Lines. In *Proc. Congress Evolutionary Computation (CEC)*, pages 387–396. IEEE, 2014.
- [39] D. Marijan, A. Gotlieb, S. Sen, and A. Hervieu. Practical Pairwise Testing for Software Product Lines. In *Proc. Int’l Software Product Line Conf. (SPLC)*, pages 227–235. ACM, 2013.
- [40] F. Medeiros, C. Kästner, M. Ribeiro, R. Gheyi, and S. Apel. A Comparison of 10 Sampling Algorithms for Configurable Systems. In *Proc. Int’l Conf. Software Engineering (ICSE)*, pages 643–654. ACM, 2016.

- [41] S. Oster, F. Markert, and P. Ritter. Automated Incremental Pairwise Testing of Software Product Lines. In *Proc. Int'l Software Product Line Conf. (SPLC)*, pages 196–210. Springer, 2010.
- [42] S. Oster, M. Zink, M. Lochau, and M. Grechanik. Pairwise Feature-interaction Testing for SPLs: Potentials and Limitations. In *Proc. Int'l Software Product Line Conf. (SPLC)*, pages 6:1–6:8. ACM, 2011.
- [43] G. Perrouin, S. Oster, S. Sen, J. Klein, B. Baudry, and Y. Le Traon. Pairwise Testing for Software Product Lines: Comparison of Two Approaches. *Software Quality Journal (SQJ)*, 20(3-4):605–643, 2012.
- [44] G. Perrouin, S. Sen, J. Klein, B. Baudry, and Y. Le Traon. Automated and Scalable T-Wise Test Case Generation Strategies for Software Product Lines. In *Proc. Int'l Conf. Software Testing, Verification and Validation (ICST)*, pages 459–468. IEEE, 2010.
- [45] D. Reuling, J. Bürdek, S. Rotärmel, M. Lochau, and U. Kelter. Fault-Based Product-Line Testing: Effective Sample Generation Based on Feature-Diagram Mutation. In *Proc. Int'l Software Product Line Conf. (SPLC)*, pages 131–140. ACM, 2015.
- [46] A. Sarkar, J. Guo, N. Siegmund, S. Apel, and K. Czarnecki. Cost-Efficient Sampling for Performance Prediction of Configurable Systems. In *Proc. Int'l Conf. Automated Software Engineering (ASE)*, pages 342–352. IEEE, 2015.
- [47] A. S. Sayyad, J. Ingram, T. Menzies, and H. Ammar. Scalable product line configuration: A straw to break the camel's back. In *Proc. Int'l Conf. Automated Software Engineering (ASE)*, pages 465–474. IEEE, 2013.
- [48] J. Shi, M. B. Cohen, and M. B. Dwyer. Integration Testing of Software Product Lines Using Compositional Symbolic Execution. In *Proc. Int'l Conf. Fundamental Approaches to Software Engineering (FASE)*, pages 270–284. Springer, 2012.
- [49] N. Siegmund, A. Grebhahn, S. Apel, and C. Kästner. Performance-Influence Models for Highly Configurable Systems. In *Proc. Europ. Software Engineering Conf./Foundations of Software Engineering (ESEC/FSE)*, pages 284–294. ACM, 2015.
- [50] N. Siegmund, S. S. Kolesnikov, C. Kästner, S. Apel, D. Batory, M. Rosenmüller, and G. Saake. Predicting Performance via Automated Feature-Interaction Detection. In *Proc. Int'l Conf. Software Engineering (ICSE)*, pages 167–177. IEEE, 2012.
- [51] N. Siegmund, M. Rosenmüller, C. Kästner, P. G. Giarrusso, S. Apel, and S. Kolesnikov. Scalable Prediction of Non-functional Properties in Software Product Lines. In *Proc. Int'l Software Product Line Conf. (SPLC)*, pages 160–169. IEEE, 2011.
- [52] N. Siegmund, M. Rosenmüller, C. Kästner, P. G. Giarrusso, S. Apel, and S. S. Kolesnikov. Scalable Prediction of Non-functional Properties in Software Product Lines: Footprint and Memory Consumption. *J. Information and Software Technology (IST)*, 55(3):491–507, 2013.
- [53] N. Siegmund, M. Rosenmüller, M. Kuhlemann, C. Kästner, S. Apel, and G. Saake. SPL Conqueror: Toward Optimization of Non-functional Properties in Software Product Lines. *Software Quality Journal (SQJ)*, 20(3-4):487–517, 2012.
- [54] N. Siegmund, M. Rosenmüller, M. Kuhlemann, C. Kästner, and G. Saake. Measuring Non-functional Properties in Software Product Lines for Product Derivation. In *Proc. Asia-Pacific Software Engineering Conference (APSEC)*, pages 187–194. IEEE, 2008.
- [55] N. Siegmund, S. Sobernig, and S. Apel. Attributed Variability Models: Outside the Comfort Zone. In *Proc. Europ. Software Engineering Conf./Foundations of Software Engineering (ESEC/FSE)*, pages 268–278. ACM, 2017.
- [56] N. Siegmund, A. von Rhein, and S. Apel. Family-Based Performance Measurement. In *Proc. Int'l Conf. Generative Programming: Concepts & Experiences (GPCE)*, pages 95–104. ACM, 2013.
- [57] R. Tartler, C. Dietrich, J. Sincero, W. Schröder-Preikschat, and D. Lohmann. Static Analysis of Variability in System Software: The 90,000 #Ifdefs Issue. In *Proc. USENIX Annual Technical Conference (ATC)*, pages 421–432. USENIX Association, 2014.
- [58] R. Tartler, D. Lohmann, C. Dietrich, C. Egger, and J. Sincero. Configuration Coverage in the Analysis of Large-Scale System Software. *ACM SIGOPS Operating Systems Review*, 45(3):10–14, 2012.
- [59] M. H. ter Beek, H. Muccini, and P. Pelliccione. Assume-Guarantee Testing of Evolving Software Product Line Architectures. In *Proc. Int'l Conf. Software Engineering for Resilient Systems (SERENE)*, pages 91–105. Springer, 2012.