A Revised Classification of Product Sampling for Software Product Lines

Sabrina Böhm, Aaron Molt, Tim Jannik Schmidt, Thomas Thüm | March 23 - March 27 2026

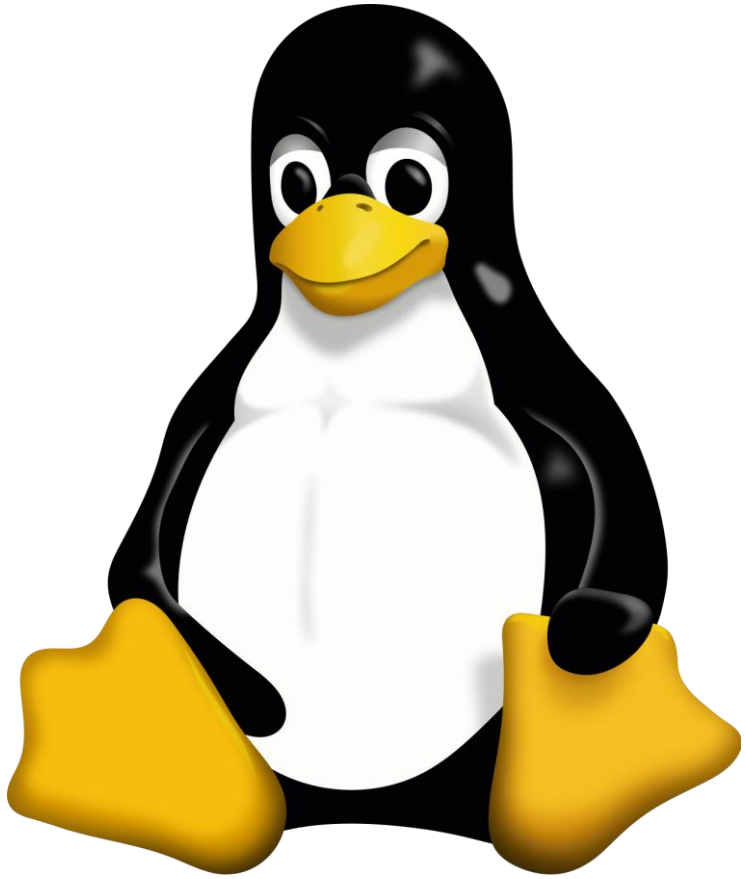
universität ulm

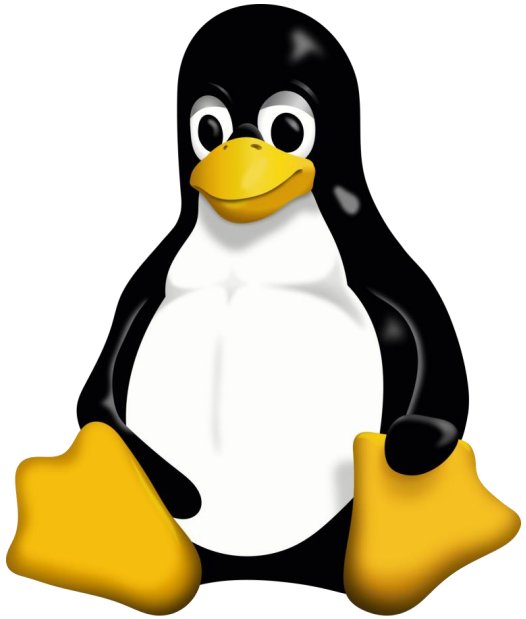
ISF

Technische Universität Braunschweig

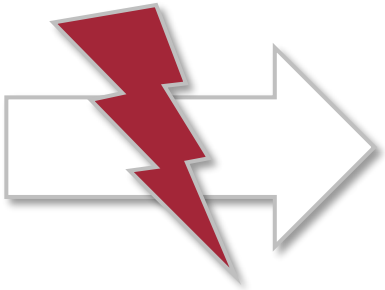
Product Lines

(Software) Product Lines

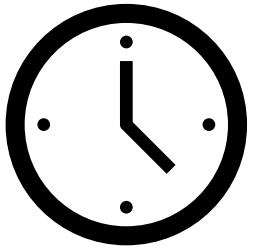


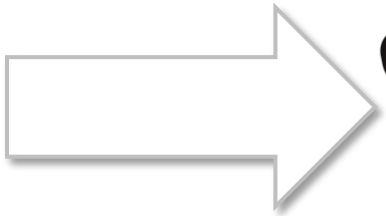


Whole SPL

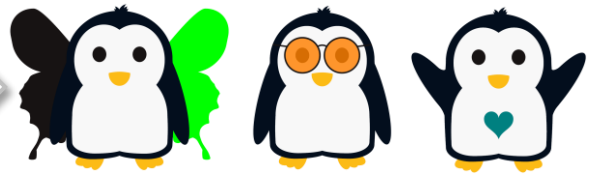


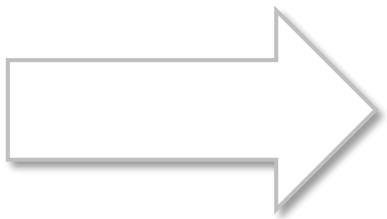
Testing





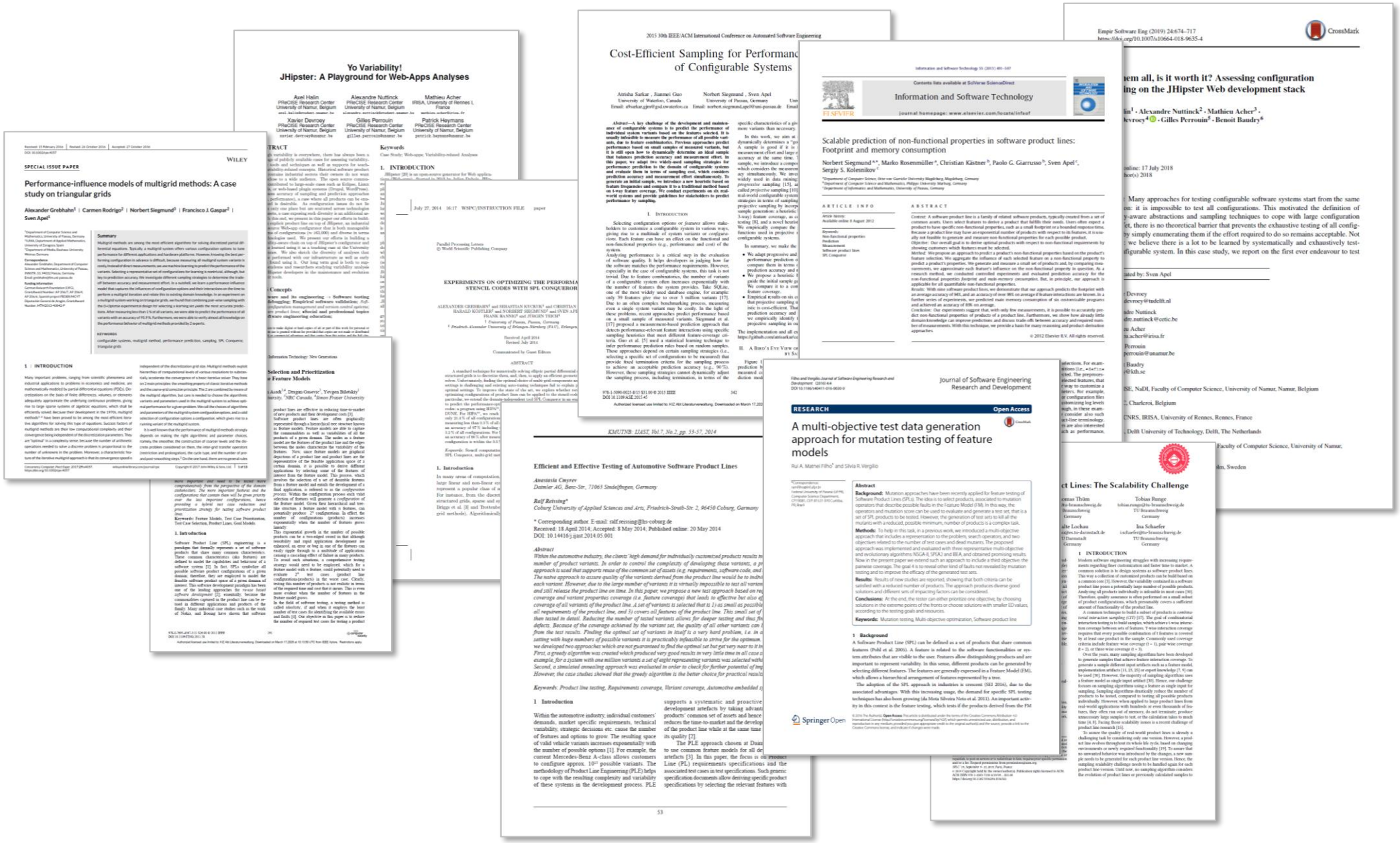
Sample

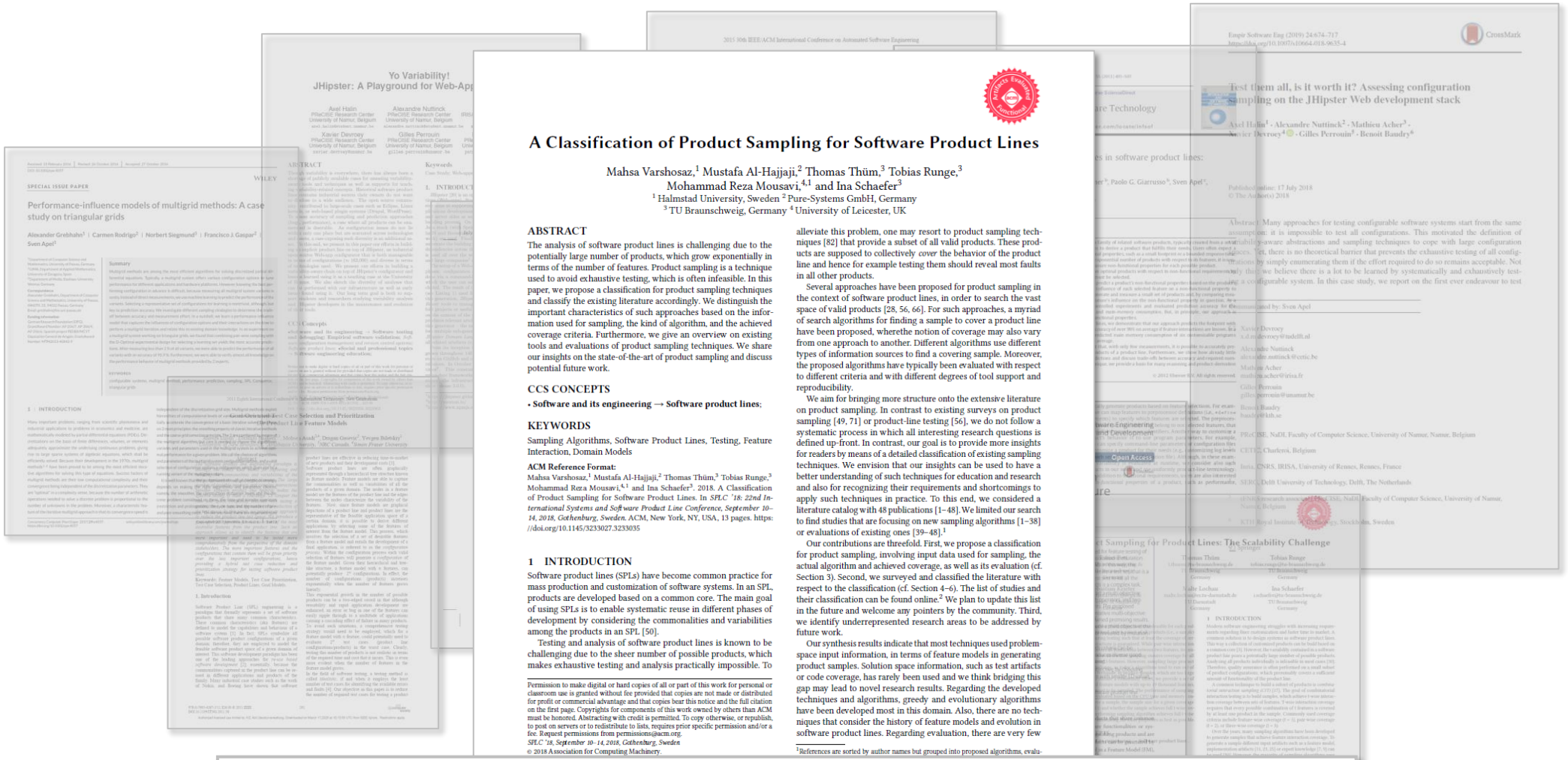




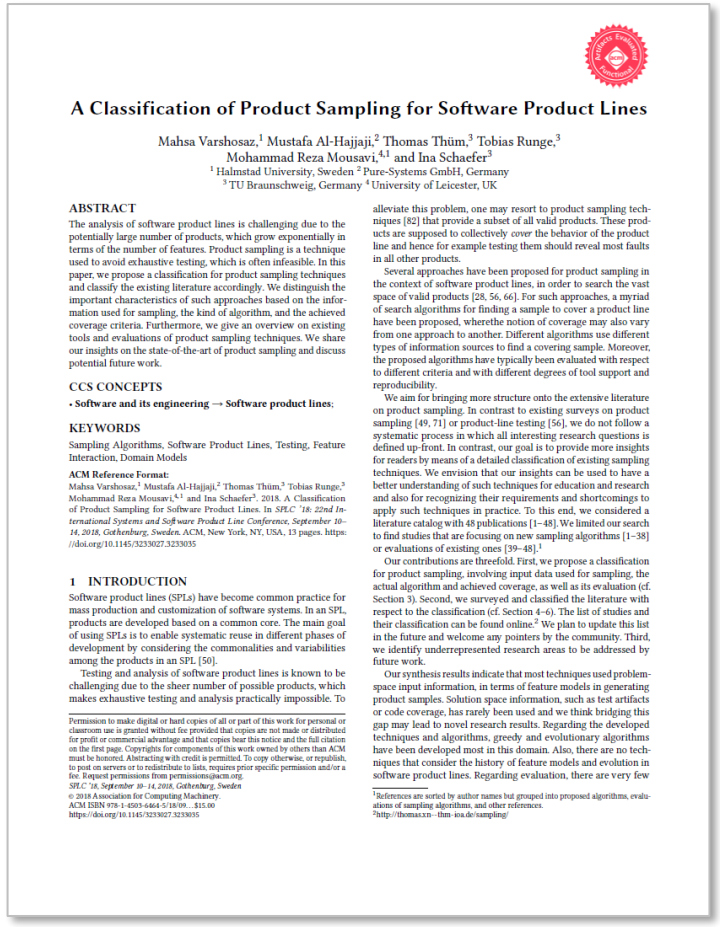
Testing

```
> testsuite@1.0.0 test

PASS ./index.test.js
  my tests
    ✓ are perfect
    ✓ always work

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Time:        0.513 s, estimated 1s
Ran all test suites.
```


Yo Variability!
JHipster: A Playground for Web-Apps Analyses
Performance-influence models of multigrid methods: A case study on triangular grids
Cost-Efficient Sampling for Performanc of Configurable Systems
Scalable prediction of non-functional properties in software product lines: Footprint and memory consumption
Information and Software Technology
A multi-objective test data generation approach for mutation testing of feature models
Journal of Software Engineering Research and Development
Efficient and Effective Testing of Automotive Software Product Lines
Empir Software Eng (2019) 24:674–717
Many approaches for testing configurable software systems start from the same
Modern software engineering struggles with increasing require-



"A Classification of Product Sampling for Software Product Lines"
Varshosaz, Al-Hajjaji, Thüm, Runge, Mousavi, Schaefer
SPLC 2018



A Classification of Product Sampling for Software Product Lines

Mahsa Varshosaz,[1] Mustafa Al-Hajjaji,[2] Thomas Thüm,[3] Tobias Runge,[3]
Mohammad Reza Mousavi,[4,1] and Ina Schaefer[3]
[1] Halmstad University, Sweden [2] Pure-Systems GmbH, Germany
[3] TU Braunschweig, Germany [4] University of Leicester, UK

ABSTRACT

The analysis of software product lines is challenging due to the potentially large number of products, which grow exponentially in terms of the number of features. Product sampling is a technique used to avoid exhaustive testing, which is often infeasible. In this paper, we propose a classification for product sampling techniques and classify the existing literature accordingly. We distinguish the important characteristics of such approaches based on the information used for sampling, the kind of algorithm, and the achieved coverage criteria. Furthermore, we give an overview on existing tools and evaluations of product sampling techniques. We share our insights on the state-of-the-art of product sampling and discuss potential future work.

CCS CONCEPTS

• **Software and its engineering → Software product lines**;

KEYWORDS

Sampling Algorithms, Software Product Lines, Testing, Feature Interaction, Domain Models

ACM Reference Format:
Mahsa Varshosaz,[1] Mustafa Al-Hajjaji,[2] Thomas Thüm,[3] Tobias Runge,[3] Mohammad Reza Mousavi,[4,1] and Ina Schaefer[3]. 2018. A Classification of Product Sampling for Software Product Lines. In *SPLC '18: 22nd International Systems and Software Product Line Conference, September 10-14, 2018, Gothenburg, Sweden.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3233027.3233035

1 INTRODUCTION

Software product lines (SPLs) have become common practice for mass production and customization of software systems. In an SPL, products are developed based on a common core. The main goal of using SPLs is to enable systematic reuse in different phases of development by considering the commonalities and variabilities among the products in an SPL [50].

Testing and analysis of software product lines is known to be challenging due to the sheer number of possible products, which makes exhaustive testing and analysis practically impossible. To alleviate this problem, one may resort to product sampling techniques [82] that provide a subset of all valid products. These products are supposed to collectively *cover* the behavior of the product line and hence for example testing them should reveal most faults in all other products.

Several approaches have been proposed for product sampling in the context of software product lines, in order to search the vast space of valid products [28, 56, 66]. For such approaches, a myriad of search algorithms for finding a sample to cover a product line have been proposed, wherethe notion of coverage may also vary from one approach to another. Different algorithms use different types of information sources to find a covering sample. Moreover, the proposed algorithms have typically been evaluated with respect to different criteria and with different degrees of tool support and reproducibility.

We aim for bringing more structure onto the extensive literature on product sampling. In contrast to existing surveys on product sampling [49, 71] or product-line testing [56], we do not follow a systematic process in which all interesting research questions is defined up-front. In contrast, our goal is to provide more insights for readers by means of a detailed classification of existing sampling techniques. We envision that our insights can be used to have a better understanding of such techniques for education and research and also for recognizing their requirements and shortcomings to apply such techniques in practice. To this end, we considered a literature catalog with 48 publications [1–48]. We limited our search to find studies that are focusing on new sampling algorithms [1–38] or evaluations of existing ones [39–48].[1]

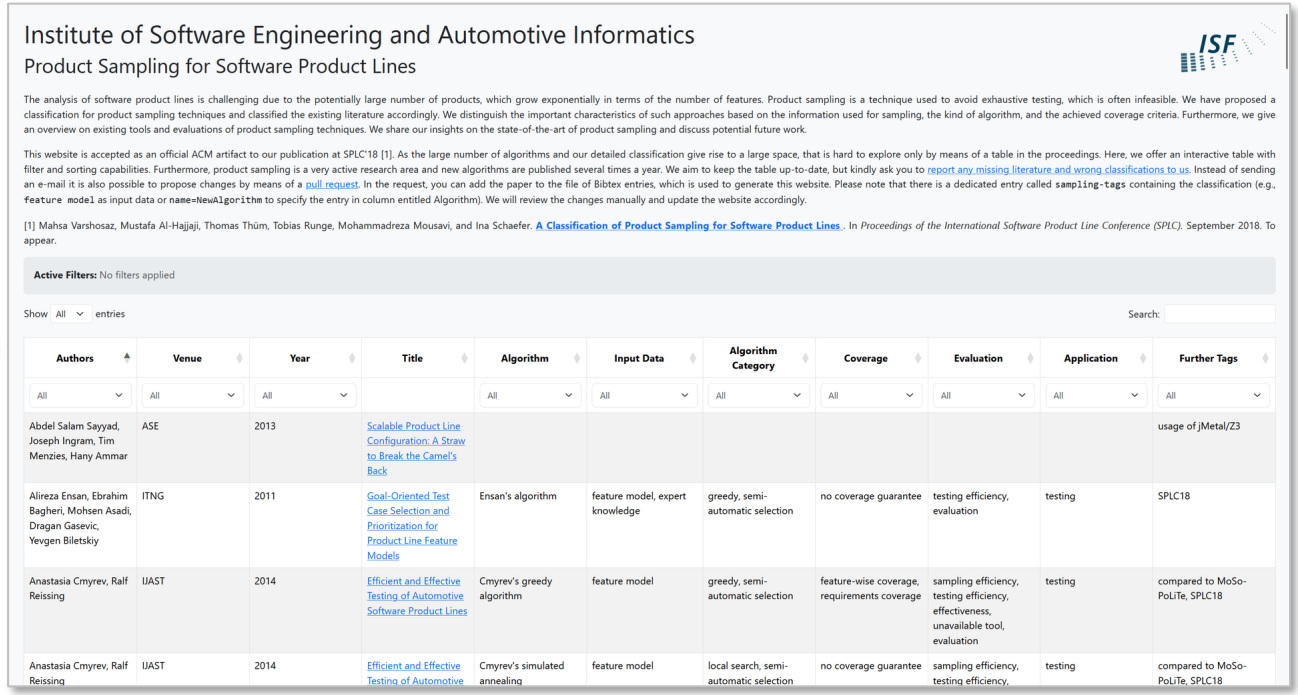
Our contributions are threefold. First, we propose a classification for product sampling, involving input data used for sampling, the actual algorithm and achieved coverage, as well as its evaluation (cf. Section 3). Second, we surveyed and classified the literature with respect to the classification (cf. Section 4–6). The list of studies and their classification can be found online.[2] We plan to update this list in the future and welcome any pointers by the community. Third, we identify underrepresented research areas to be addressed by future work.

Our synthesis results indicate that most techniques used problem-space input information, in terms of feature models in generating product samples. Solution space information, such as test artifacts or code coverage, has rarely been used and we think bridging this gap may lead to novel research results. Regarding the developed techniques and algorithms, greedy and evolutionary algorithms have been developed most in this domain. Also, there are no techniques that consider the history of feature models and evolution in software product lines. Regarding evaluation, there are very few

SPLC '18, September 10–14, 2018, Gothenburg, Sweden

ACM ISBN 978-1-4503-6464-5/18/09…$15.00
https://doi.org/10.1145/3233027.3233035

[1] References are sorted by author names but grouped into proposed algorithms, evaluations of sampling algorithms, and other references.
[2] http://thomas.xn--thm-ioa.de/sampling/



Institute of Software Engineering and Automotive Informatics

Product Sampling for Software Product Lines

The analysis of software product lines is challenging due to the potentially large number of products, which grow exponentially in terms of the number of features. Product sampling is a technique used to avoid exhaustive testing, which is often infeasible. We have proposed a classification for product sampling techniques and classified the existing literature accordingly. We distinguish the important characteristics of such approaches based on the information used for sampling, the kind of algorithm, and the achieved coverage criteria. Furthermore, we give an overview on existing tools and evaluations of product sampling techniques. We share our insights on the state-of-the-art of product sampling and discuss potential future work.

This website is accepted as an official ACM artifact to our publication at SPLC'18 [1]. As the large number of algorithms and our detailed classification give rise to a large space, that is hard to explore only by means of a table in the proceedings. Here, we offer an interactive table with filter and sorting capabilities. Furthermore, product sampling is a very active research area and new algorithms are published several times a year. We aim to keep the table up-to-date, but kindly ask you to report any missing literature and wrong classifications to us. Instead of sending an e-mail it is also possible to propose changes by means of a pull request. In the request, you can add the paper to the file of Bibtex entries, which is used to generate this website. Please note that there is a dedicated entry called `sampling-tags` containing the classification (e.g., `feature model` as input data or `name=NewAlgorithm` to specify the entry in column entitled Algorithm). We will review the changes manually and update the website accordingly.

[1] Mahsa Varshosaz, Mustafa Al-Hajjaji, Thomas Thüm, Tobias Runge, Mohammadreza Mousavi, and Ina Schaefer. **A Classification of Product Sampling for Software Product Lines** . In *Proceedings of the International Software Product Line Conference (SPLC)*. September 2018. To appear.

Active Filters: No filters applied

Show All entries Search:

Authors	Venue	Year	Title	Algorithm	Input Data	Algorithm Category	Coverage	Evaluation	Application	Further Tags
All	All	All		All	All	All	All	All	All	All
Abdel Salam Sayyad, Joseph Ingram, Tim Menzies, Hany Ammar	ASE	2013	Scalable Product Line Configuration: A Straw to Break the Camel's Back							usage of jMetal/Z3
Alireza Ensan, Ebrahim Bagheri, Mohsen Asadi, Dragan Gasevic, Yevgen Biletskiy	ITNG	2011	Goal-Oriented Test Case Selection and Prioritization for Product Line Feature Models	Ensan's algorithm	feature model, expert knowledge	greedy, semi-automatic selection	no coverage guarantee	testing efficiency, evaluation	testing	SPLC18
Anastasia Cmyrev, Ralf Reissing	IJAST	2014	Efficient and Effective Testing of Automotive Software Product Lines	Cmyrev's greedy algorithm	feature model	greedy, semi-automatic selection	feature-wise coverage, requirements coverage	sampling efficiency, testing efficiency, effectiveness, unavailable tool, evaluation	testing	compared to MoSo-PoLiTe, SPLC18
Anastasia Cmyrev, Ralf Reissing	IJAST	2014	Efficient and Effective Testing of Automotive	Cmyrev's simulated annealing	feature model	local search, semi-automatic selection	no coverage guarantee	sampling efficiency, testing efficiency,	testing	compared to MoSo-PoLiTe, SPLC18

https://tubs-isf.github.io/

Institute of Software Engineering and Automotive Informatics

Product Sampling for Software Product Lines

The analysis of software product lines is challenging due to the potentially large number of products, which grow exponentially in terms of the number of features. Product sampling is a technique used to avoid exhaustive testing, which is often infeasible. We have proposed a classification for product sampling techniques and classified the existing literature accordingly. We distinguish the important characteristics of such approaches based on the information used for sampling, the kind of algorithm, and the achieved coverage criteria. Furthermore, we give an overview on existing tools and evaluations of product sampling techniques. We share our insights on the state-of-the-art of product sampling and discuss potential future work.

This website is accepted as an official ACM artifact to our publication at SPLC'18 [1]. As the large number of algorithms and our detailed classification give rise to a large space, that is hard to explore only by means of a table in the proceedings. Here, we offer an interactive table with filter and sorting capabilities. Furthermore, product sampling is a very active research area and new algorithms are published several times a year. We aim to keep the table up-to-date, but kindly ask you to report any missing literature and wrong classifications to us. Instead of sending an e-mail it is also possible to propose changes by means of a pull request. In the request, you can add the paper to the file of Bibtex entries, which is used to generate this website. Please note that there is a dedicated entry called `sampling-tags` containing the classification (e.g., `feature model` as input data or `name=NewAlgorithm` to specify the entry in column entitled Algorithm). We will review the changes manually and update the website accordingly.

[1] Mahsa Varshosaz, Mustafa Al-Hajjaji, Thomas Thüm, Tobias Runge, Mohammadreza Mousavi, and Ina Schaefer. **A Classification of Product Sampling for Software Product Lines** . In *Proceedings of the International Software Product Line Conference (SPLC)*. September 2018. To appear.

Active Filters: No filters applied

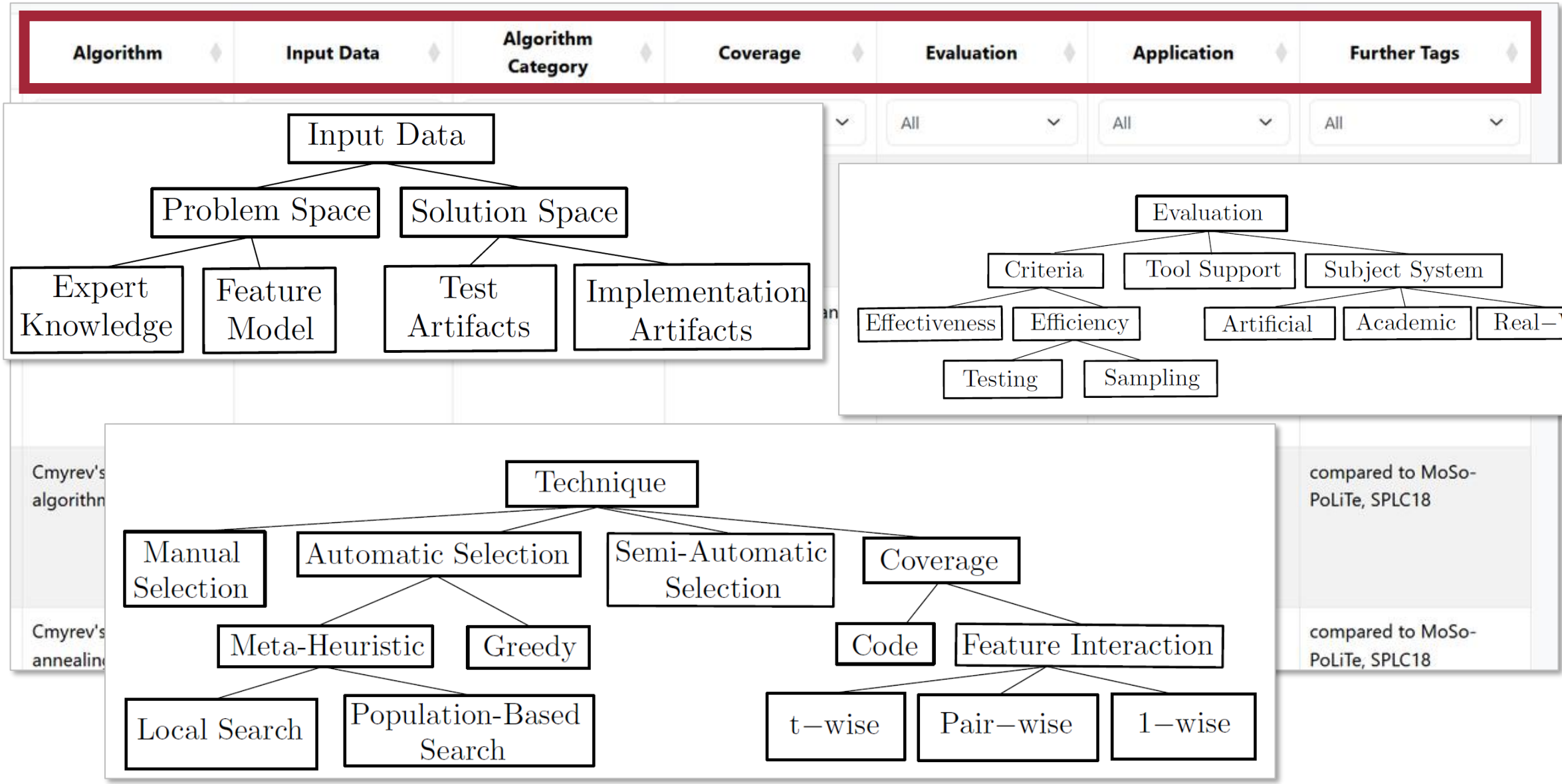
Show All entries Search:

Authors	Venue	Year	Title	Algorithm	Input Data	Algorithm Category	Coverage	Evaluation	Application	Further Tags
All	All	All		All	All	All	All	All	All	All
Abdel Salam Sayyad, Joseph Ingram, Tim Menzies, Hany Ammar	ASE	2013	Scalable Product Line Configuration: A Straw to Break the Camel's Back							usage of jMetal/Z3
Alireza Ensan, Ebrahim Bagheri, Mohsen Asadi, Dragan Gasevic, Yevgen Biletskiy	ITNG	2011	Goal-Oriented Test Case Selection and Prioritization for Product Line Feature Models	Ensan's algorithm	feature model, expert knowledge	greedy, semi-automatic selection	no coverage guarantee	testing efficiency, evaluation	testing	SPLC18
Anastasia Cmyrev, Ralf Reissing	IJAST	2014	Efficient and Effective Testing of Automotive Software Product Lines	Cmyrev's greedy algorithm	feature model	greedy, semi-automatic selection	feature-wise coverage, requirements coverage	sampling efficiency, testing efficiency, effectiveness, unavailable tool, evaluation	testing	compared to MoSo-PoLiTe, SPLC18
Anastasia Cmyrev, Ralf Reissing	IJAST	2014	Efficient and Effective Testing of Automotive	Cmyrev's simulated annealing	feature model	local search, semi-automatic selection	no coverage guarantee	sampling efficiency, testing efficiency,	testing	compared to MoSo-PoLiTe, SPLC18

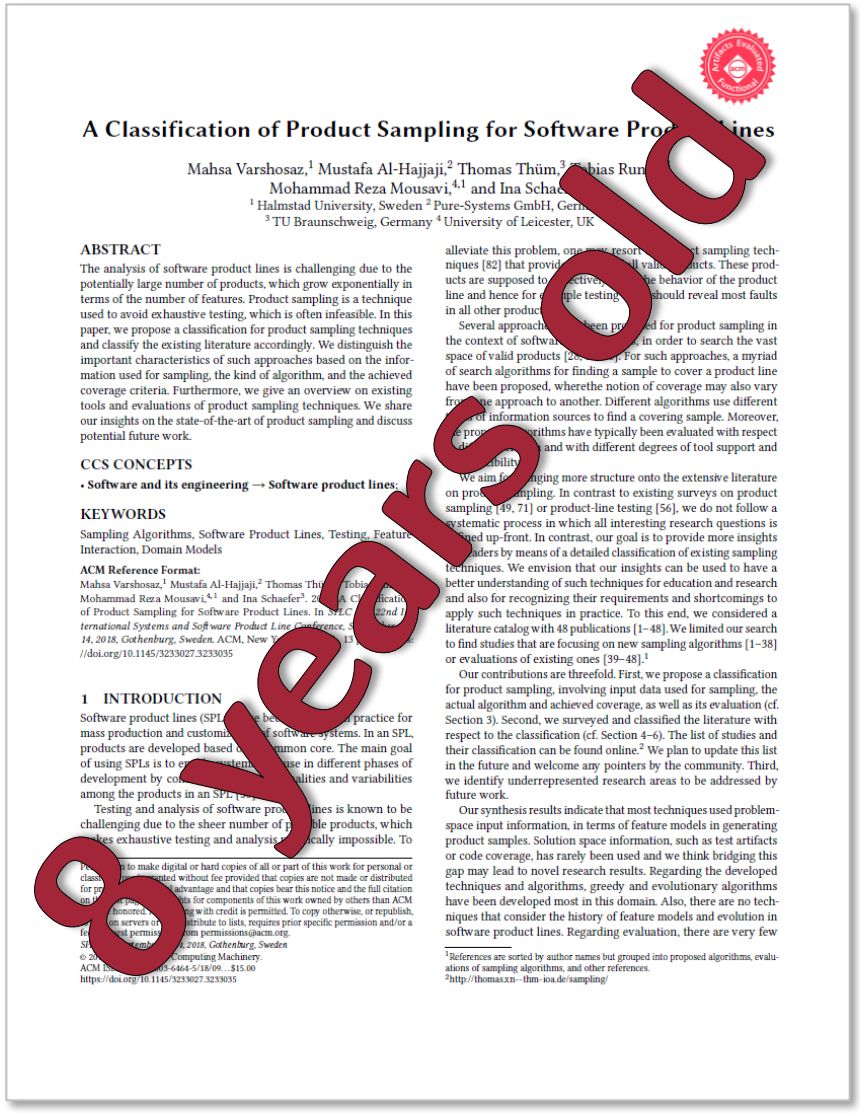
https://tubs-isf.github.io/

Algorithm	Input Data	Algorithm Category	Coverage	Evaluation	Application	Further Tags
All	All	All	All	All	All	All
						usage of jMetal/Z3
Ensan's algorithm	feature model, expert knowledge	greedy, semi-automatic selection	no coverage guarantee	testing efficiency, evaluation	testing	SPLC18
Cmyrev's greedy algorithm	feature model	greedy, semi-automatic selection	feature-wise coverage, requirements coverage	sampling efficiency, testing efficiency, effectiveness, unavailable tool, evaluation	testing	compared to MoSo-PoLiTe, SPLC18
Cmyrev's simulated annealing	feature model	local search, semi-automatic selection	no coverage guarantee	sampling efficiency, testing efficiency,	testing	compared to MoSo-PoLiTe, SPLC18

https://tubs-isf.github.io/



https://tubs-isf.github.io/



A Classification of Product Sampling for Software Prod[illegible] Lines

Mahsa Varshosaz,[1] Mustafa Al-Hajjaji,[2] Thomas Thüm,[3] [illegible]obias Run[illegible]
Mohammad Reza Mousavi,[4,1] and Ina Schae[illegible]
[1] Halmstad University, Sweden [2] Pure-Systems GmbH, Germ[illegible]
[3] TU Braunschweig, Germany [4] University of Leicester, UK

ABSTRACT

The analysis of software product lines is challenging due to the potentially large number of products, which grow exponentially in terms of the number of features. Product sampling is a technique used to avoid exhaustive testing, which is often infeasible. In this paper, we propose a classification for product sampling techniques and classify the existing literature accordingly. We distinguish the important characteristics of such approaches based on the information used for sampling, the kind of algorithm, and the achieved coverage criteria. Furthermore, we give an overview on existing tools and evaluations of product sampling techniques. We share our insights on the state-of-the-art of product sampling and discuss potential future work.

CCS CONCEPTS

• **Software and its engineering → Software product lines**;

KEYWORDS

Sampling Algorithms, Software Product Lines, Testing, Feature Interaction, Domain Models

ACM Reference Format:
Mahsa Varshosaz,[1] Mustafa Al-Hajjaji,[2] Thomas Thü[illegible] Tobia[illegible]
Mohammad Reza Mousavi,[4,1] and Ina Schaefer[3]. 20[illegible]A Classificati[illegible]
of Product Sampling for Software Product Lines. In *SPLC '[illegible]2nd I[illegible]
ternational Systems and Software Product Line Conference, S[illegible]
14, 2018, Gothenburg, Sweden.* ACM, New Y[illegible]13 [illegible]
//doi.org/10.1145/3233027.3233035

1 INTRODUCTION

Software product lines (SPL[illegible]e be[illegible]practice for mass production and customi[illegible]f softwa[illegible]systems. In an SPL, products are developed based [illegible]mmon core. The main goal of using SPLs is to e[illegible]system[illegible]use in different phases of development by co[illegible]alities and variabilities among the products in an SPL [53].

Testing and analysis of software pro[illegible]lines is known to be challenging due to the sheer number of p[illegible]ble products, which makes exhaustive testing and analysis [illegible]ically impossible. To alleviate this problem, one [illegible] resor[illegible]t sampling techniques [82] that provid[illegible]ll vali[illegible]ucts. These products are supposed to [illegible]ective[illegible]e behavior of the product line and hence for [illegible]mple testing [illegible]should reveal most faults in all other product[illegible]

Several approach[illegible]been pr[illegible]ed for product sampling in the context of softwa[illegible] in order to search the vast space of valid products [2[illegible]. For such approaches, a myriad of search algorithms for finding a sample to cover a product line have been proposed, wherethe notion of coverage may also vary fro[illegible]e approach to another. Different algorithms use different [illegible] of information sources to find a covering sample. Moreover, e pro[illegible]orithms have typically been evaluated with respect [illegible] and with different degrees of tool support and [illegible]ibility

We aim fo[illegible]nging more structure onto the extensive literature on pro[illegible]mpling. In contrast to existing surveys on product sampling [49, 71] or product-line testing [56], we do not follow a systematic process in which all interesting research questions is [illegible]ed up-front. In contrast, our goal is to provide more insights [illegible]aders by means of a detailed classification of existing sampling techniques. We envision that our insights can be used to have a better understanding of such techniques for education and research and also for recognizing their requirements and shortcomings to apply such techniques in practice. To this end, we considered a literature catalog with 48 publications [1–48]. We limited our search to find studies that are focusing on new sampling algorithms [1–38] or evaluations of existing ones [39–48].[1]

Our contributions are threefold. First, we propose a classification for product sampling, involving input data used for sampling, the actual algorithm and achieved coverage, as well as its evaluation (cf. Section 3). Second, we surveyed and classified the literature with respect to the classification (cf. Section 4–6). The list of studies and their classification can be found online.[2] We plan to update this list in the future and welcome any pointers by the community. Third, we identify underrepresented research areas to be addressed by future work.

Our synthesis results indicate that most techniques used problem-space input information, in terms of feature models in generating product samples. Solution space information, such as test artifacts or code coverage, has rarely been used and we think bridging this gap may lead to novel research results. Regarding the developed techniques and algorithms, greedy and evolutionary algorithms have been developed most in this domain. Also, there are no techniques that consider the history of feature models and evolution in software product lines. Regarding evaluation, there are very few

SP[illegible]embe[illegible], 2018, Gothenburg, Sweden

ACM ISB[illegible]03-6464-5/18/09…$15.00
https://doi.org/10.1145/3233027.3233035

[1] References are sorted by author names but grouped into proposed algorithms, evaluations of sampling algorithms, and other references.
[2] http://thomas.xn--thm-ioa.de/sampling/

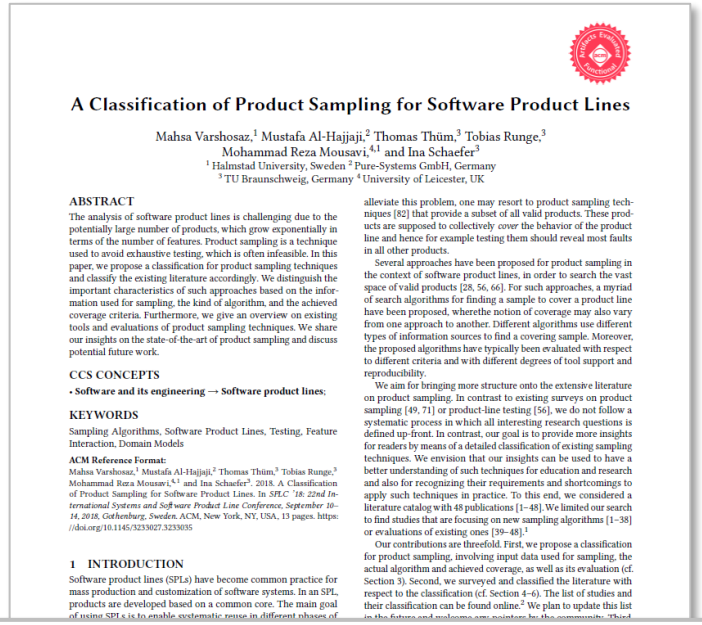
What are the latest trends?

?

Did the application areas change?

Is this classification sufficient?

Golden Set



A Classification of Product Sampling for Software Product Lines

Mahsa Varshosaz,[1] Mustafa Al-Hajjaji,[2] Thomas Thüm,[3] Tobias Runge,[3]
Mohammad Reza Mousavi,[4,1] and Ina Schaefer[3]
[1] Halmstad University, Sweden [2] Pure-Systems GmbH, Germany
[3] TU Braunschweig, Germany [4] University of Leicester, UK

ABSTRACT

The analysis of software product lines is challenging due to the potentially large number of products, which grow exponentially in terms of the number of features. Product sampling is a technique used to avoid exhaustive testing, which is often infeasible. In this paper, we propose a classification for product sampling techniques and classify the existing literature accordingly. We distinguish the important characteristics of such approaches based on the information used for sampling, the kind of algorithm, and the achieved coverage criteria. Furthermore, we give an overview on existing tools and evaluations of product sampling techniques. We share our insights on the state-of-the-art of product sampling and discuss potential future work.

CCS CONCEPTS

• **Software and its engineering → Software product lines**;

KEYWORDS

Sampling Algorithms, Software Product Lines, Testing, Feature Interaction, Domain Models

ACM Reference Format:

Mahsa Varshosaz,[1] Mustafa Al-Hajjaji,[2] Thomas Thüm,[3] Tobias Runge,[3] Mohammad Reza Mousavi,[4,1] and Ina Schaefer[3]. 2018. A Classification of Product Sampling for Software Product Lines. In *SPLC '18: 22nd International Systems and Software Product Line Conference, September 10–14, 2018, Gothenburg, Sweden.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3233027.3233035

1 INTRODUCTION

Software product lines (SPLs) have become common practice for mass production and customization of software systems. In an SPL, products are developed based on a common core. The main goal of using SPLs is to enable systematic reuse in different phases of

alleviate this problem, one may resort to product sampling techniques [82] that provide a subset of all valid products. These products are supposed to collectively *cover* the behavior of the product line and hence for example testing them should reveal most faults in all other products.

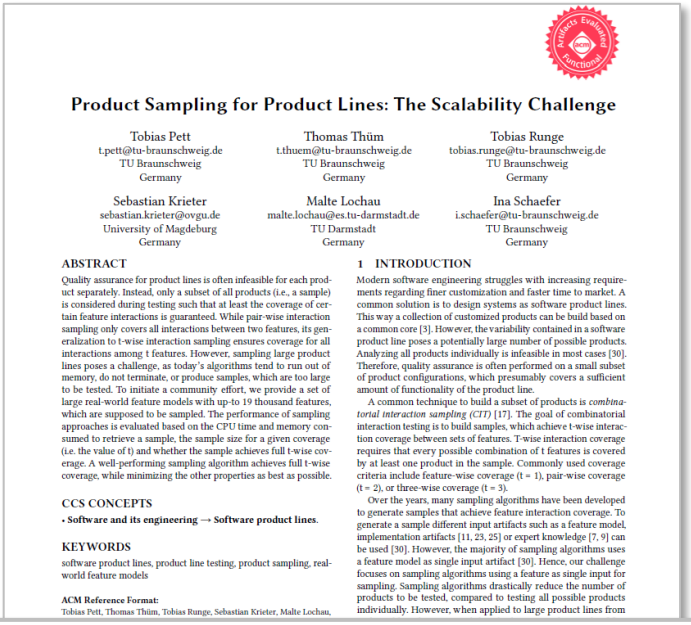
Several approaches have been proposed for product sampling in the context of software product lines, in order to search the vast space of valid products [28, 56, 66]. For such approaches, a myriad of search algorithms for finding a sample to cover a product line have been proposed, wherethe notion of coverage may also vary from one approach to another. Different algorithms use different types of information sources to find a covering sample. Moreover, the proposed algorithms have typically been evaluated with respect to different criteria and with different degrees of tool support and reproducibility.

We aim for bringing more structure onto the extensive literature on product sampling. In contrast to existing surveys on product sampling [49, 71] or product-line testing [56], we do not follow a systematic process in which all interesting research questions is defined up-front. In contrast, our goal is to provide more insights for readers by means of a detailed classification of existing sampling techniques. We envision that our insights can be used to have a better understanding of such techniques for education and research and also for recognizing their requirements and shortcomings to apply such techniques in practice. To this end, we considered a literature catalog with 48 publications [1–48]. We limited our search to find studies that are focusing on new sampling algorithms [1–38] or evaluations of existing ones [39–48].[1]

Our contributions are threefold. First, we propose a classification for product sampling, involving input data used for sampling, the actual algorithm and achieved coverage, as well as its evaluation (cf. Section 3). Second, we surveyed and classified the literature with respect to the classification (cf. Section 4–6). The list of studies and their classification can be found online.[2] We plan to update this list in the future and welcome any pointers by the community. Third,

"A Classification of Product Sampling for Software Product Lines"
Varshosaz et al.
(SPLC 2018)

+



Product Sampling for Product Lines: The Scalability Challenge

Tobias Pett
t.pett@tu-braunschweig.de
TU Braunschweig
Germany

Thomas Thüm
t.thuem@tu-braunschweig.de
TU Braunschweig
Germany

Tobias Runge
tobias.runge@tu-braunschweig.de
TU Braunschweig
Germany

Sebastian Krieter
sebastian.krieter@ovgu.de
University of Magdeburg
Germany

Malte Lochau
malte.lochau@es.tu-darmstadt.de
TU Darmstadt
Germany

Ina Schaefer
i.schaefer@tu-braunschweig.de
TU Braunschweig
Germany

ABSTRACT

Quality assurance for product lines is often infeasible for each product separately. Instead, only a subset of all products (i.e., a sample) is considered during testing such that at least the coverage of certain feature interactions is guaranteed. While pair-wise interaction sampling only covers all interactions between two features, its generalization to t-wise interaction sampling ensures coverage for all interactions among t features. However, sampling large product lines poses a challenge, as today's algorithms tend to run out of memory, do not terminate, or produce samples, which are too large to be tested. To initiate a community effort, we provide a set of large real-world feature models with up-to 19 thousand features, which are supposed to be sampled. The performance of sampling approaches is evaluated based on the CPU time and memory consumed to retrieve a sample, the sample size for a given coverage (i.e. the value of t) and whether the sample achieves full t-wise coverage. A well-performing sampling algorithm achieves full t-wise coverage, while minimizing the other properties as best as possible.

CCS CONCEPTS

• **Software and its engineering → Software product lines**.

KEYWORDS

software product lines, product line testing, product sampling, real-world feature models

ACM Reference Format:

Tobias Pett, Thomas Thüm, Tobias Runge, Sebastian Krieter, Malte Lochau,

1 INTRODUCTION

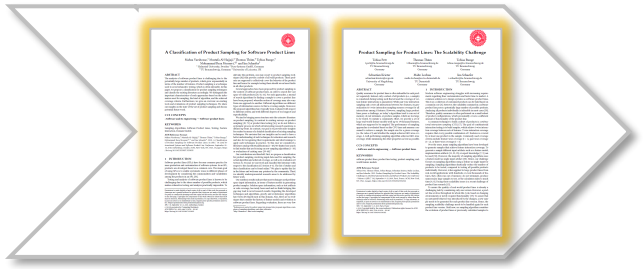
Modern software engineering struggles with increasing requirements regarding finer customization and faster time to market. A common solution is to design systems as software product lines. This way a collection of customized products can be build based on a common core [3]. However, the variability contained in a software product line poses a potentially large number of possible products. Analyzing all products individually is infeasible in most cases [30]. Therefore, quality assurance is often performed on a small subset of product configurations, which presumably covers a sufficient amount of functionality of the product line.

A common technique to build a subset of products is *combinatorial interaction sampling (CIT)* [17]. The goal of combinatorial interaction testing is to build samples, which achieve t-wise interaction coverage between sets of features. T-wise interaction coverage requires that every possible combination of t features is covered by at least one product in the sample. Commonly used coverage criteria include feature-wise coverage (t = 1), pair-wise coverage (t = 2), or three-wise coverage (t = 3).

Over the years, many sampling algorithms have been developed to generate samples that achieve feature interaction coverage. To generate a sample different input artifacts such as a feature model, implementation artifacts [11, 23, 25] or expert knowledge [7, 9] can be used [30]. However, the majority of sampling algorithms uses a feature model as single input artifact [30]. Hence, our challenge focuses on sampling algorithms using a feature as single input for sampling. Sampling algorithms drastically reduce the number of products to be tested, compared to testing all possible products individually. However, when applied to large product lines from

"Product Sampling for Product Lines: The Scalability Challenge"
Pett et al.
(SPLC 2019)

Golden Set



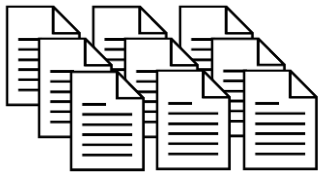
Golden Set

Forward Snowballing

Golden Set

Forward Snowballing

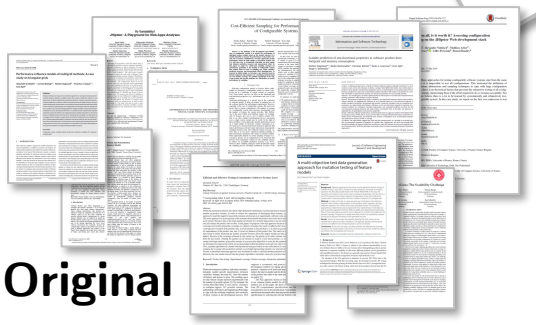
Set of Papers



Golden Set

Forward Snowballing

Set of Papers



Papers from Original Survey

Golden Set

Forward Snowballing

Set of Papers

Inclusion & Exclusion Criteria

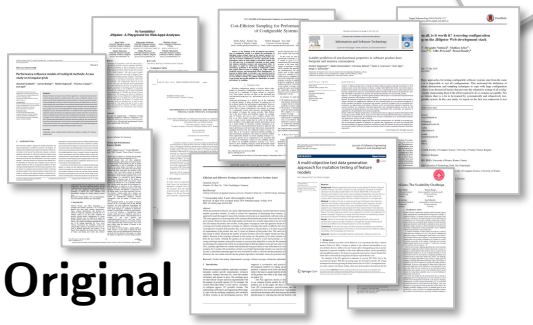
Technical Exclusion

- Conference/Journal
- English
- Available "for free" online

Content Inclusion

- Introduces, Compares, or Evaluates Sampling Approach(es)
- Topic: Product Sampling or Interaction Testing
- Feature Model, Boolean Algebra, or Decision Model

Papers from Original Survey



Inclusion & Exclusion Criteria

Technical Exclusion

- Conference/Journal
- English
- Available “for free” online

Content Inclusion

- Introduces, Compares, or Evaluates Sampling Approach(es)
- Topic: Product Sampling or Interaction Testing
- Feature Model, Boolean Algebra, or Decision Model

Golden Set

Forward Snowballing

Set of Papers

Inclusion & Exclusion Criteria

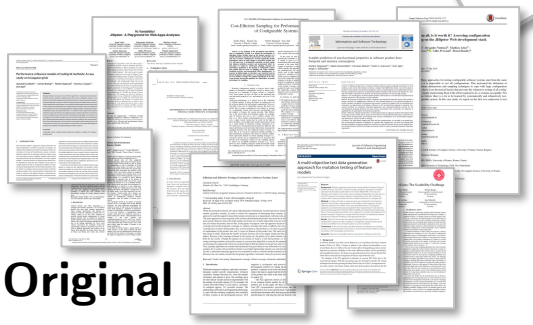
Technical Exclusion

- Conference/Journal
- English
- Available "for free" online

Content Inclusion

- Introduces, Compares, or Evaluates Sampling Approach(es)
- Topic: Product Sampling or Interaction Testing
- Feature Model, Boolean Algebra, or Decision Model

Papers from Original Survey



Golden Set

Forward Snowballing

Set of Papers

Inclusion & Exclusion Criteria

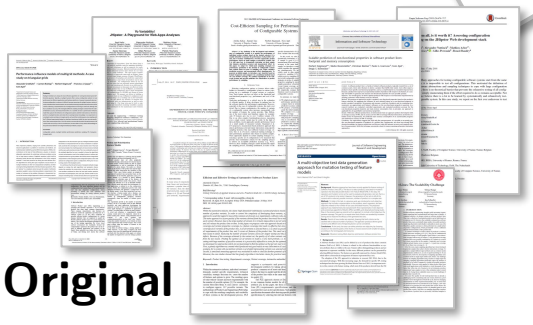
Technical Exclusion

- Conference/Journal
- English
- Available "for free" online

Content Inclusion

- Introduces, Compares, or Evaluates Sampling Approach(es)
- Topic: Product Sampling or Interaction Testing
- Feature Model, Boolean Algebra, or Decision Model

Papers from Original Survey



Classification

Golden Set

Forward Snowballing

Set of Papers

Inclusion & Exclusion Criteria

Technical Exclusion

- Conference/Journal
- English
- Available "for free" online

Content Inclusion

- Introduces, Compares, or Evaluates Sampling Approach(es)
- Topic: Product Sampling or Interaction Testing
- Feature Model, Boolean Algebra, or Decision Model

Papers from Original Survey

Forward and Backward Snowballing

Classification

Golden Set

Forward Snowballing

Set of Papers

Inclusion & Exclusion Criteria

Technical Exclusion

- Conference/Journal
- English
- Available "for free" online

Content Inclusion

- Introduces, Compares, or Evaluates Sampling Approach(es)
- Topic: Product Sampling or Interaction Testing
- Feature Model, Boolean Algebra, or Decision Model

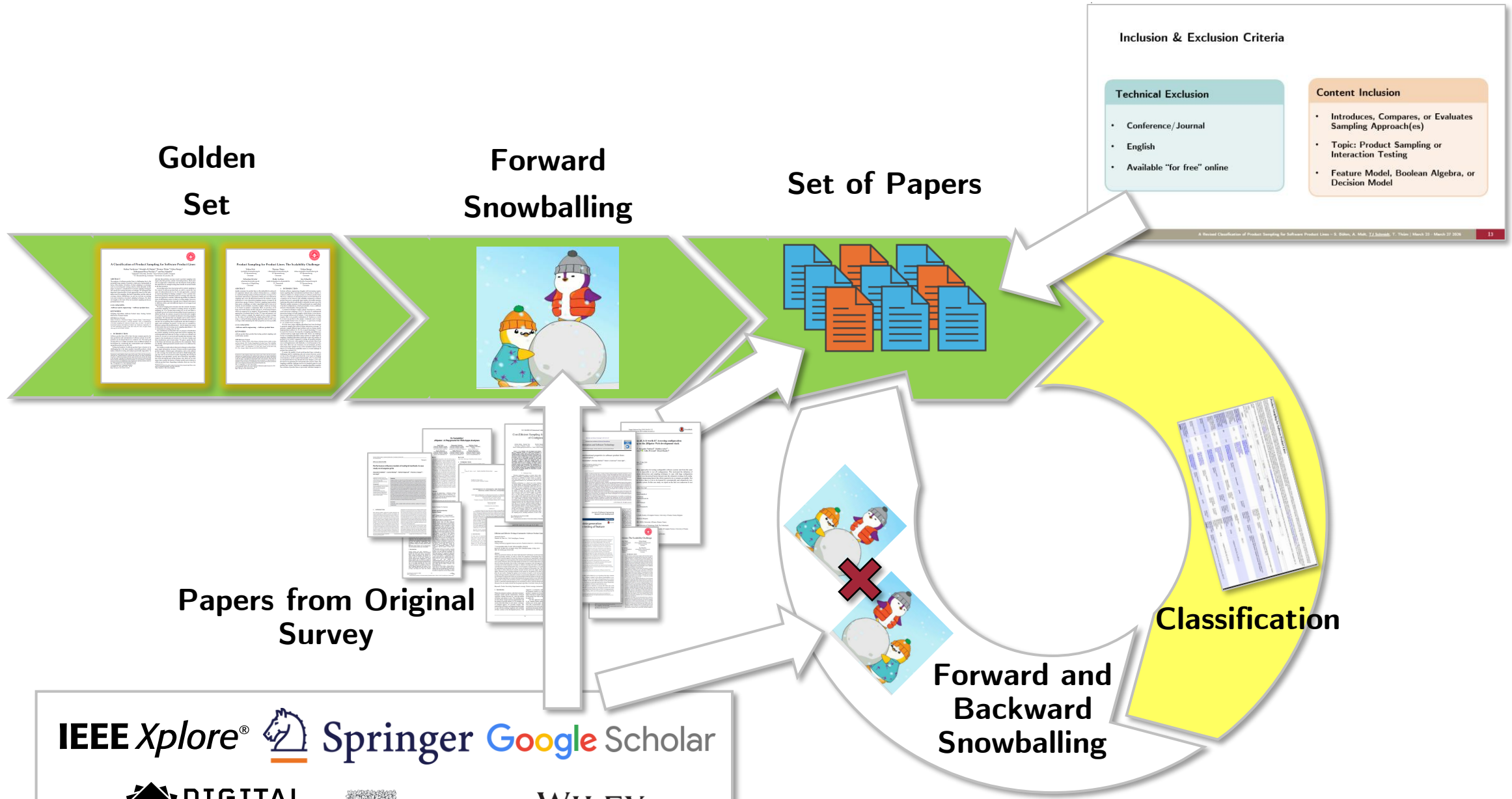
Papers from Original Survey

Forward and Backward Snowballing

Classification

IEEE Xplore® Springer Google Scholar

ACM DIGITAL LIBRARY Scopus WILEY Online Library


Inclusion & Exclusion Criteria
Technical Exclusion
Conference/Journal
English
Available "for free" online
Content Inclusion
Introduces, Compares, or Evaluates Sampling Approach(es)
Topic: Product Sampling or Interaction Testing
Feature Model, Boolean Algebra, or Decision Model
Golden Set
Forward Snowballing
Set of Papers
Papers from Original Survey
Forward and Backward Snowballing
Classification
IEEE Xplore
Springer
Google Scholar
ACM DIGITAL LIBRARY
Scopus
WILEY Online Library

Golden Set

Forward Snowballing

Set of Papers

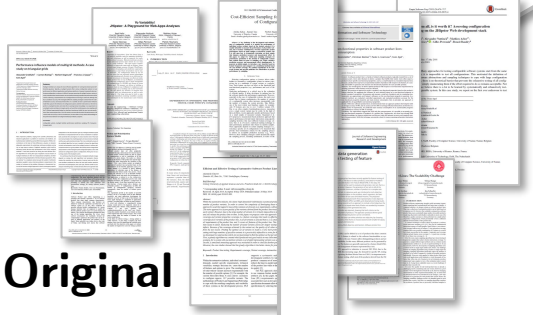
Inclusion & Exclusion Criteria

Technical Exclusion

- Conference/Journal
- English
- Available "for free" online

Content Inclusion

- Introduces, Compares, or Evaluates Sampling Approach(es)
- Topic: Product Sampling or Interaction Testing
- Feature Model, Boolean Algebra, or Decision Model



Papers from Original Survey

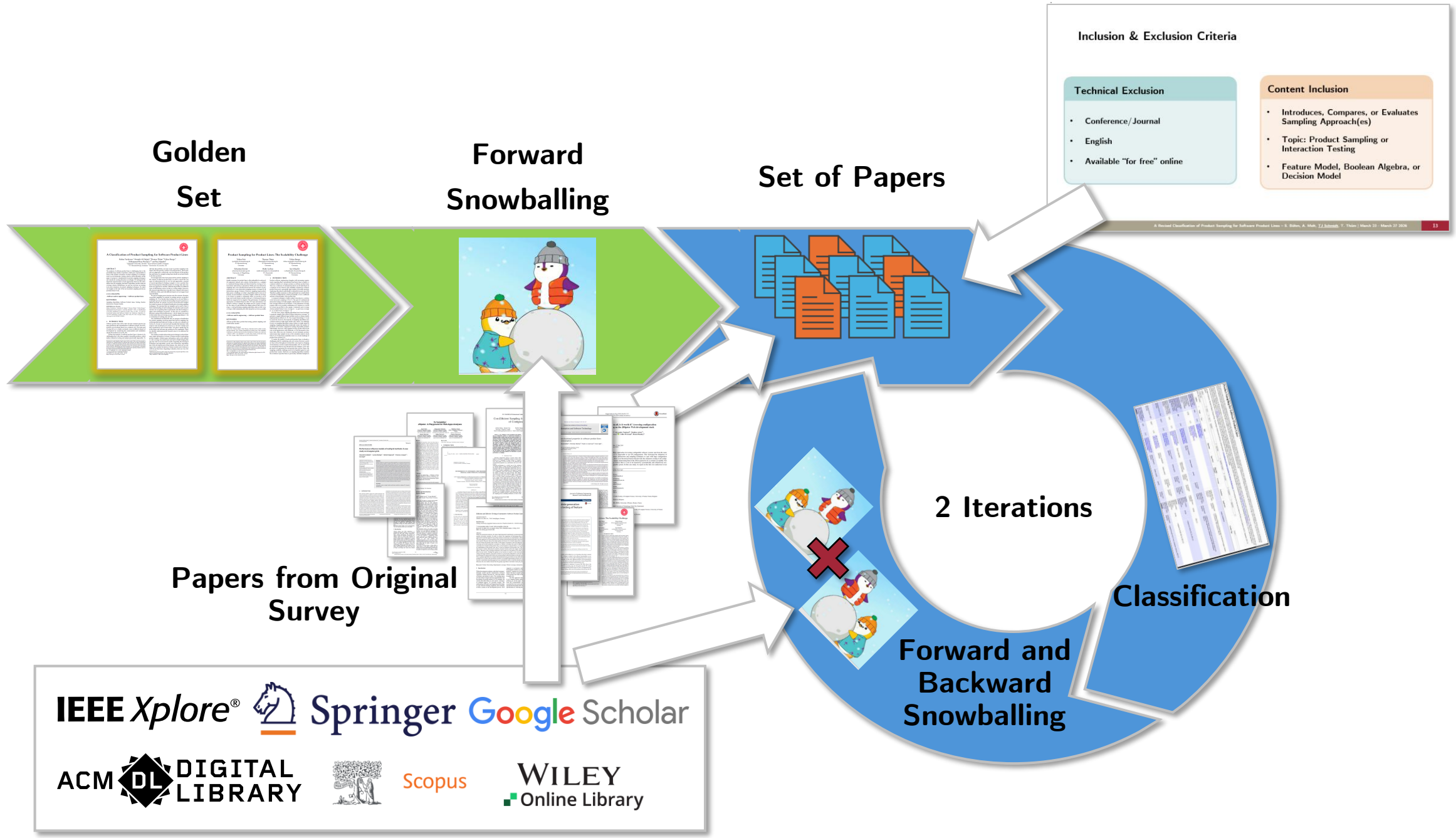
1 Iteration

Classification

Forward and Backward Snowballing

IEEE Xplore® Springer Google Scholar

ACM DIGITAL LIBRARY Scopus WILEY Online Library


Golden Set
Forward Snowballing
Set of Papers
Inclusion & Exclusion Criteria
Technical Exclusion
Conference/Journal
English
Available "for free" online
Content Inclusion
Introduces, Compares, or Evaluates Sampling Approach(es)
Topic: Product Sampling or Interaction Testing
Feature Model, Boolean Algebra, or Decision Model
Papers from Original Survey
2 Iterations
Classification
Forward and Backward Snowballing
IEEE Xplore
Springer
Google Scholar
ACM DIGITAL LIBRARY
Scopus
WILEY Online Library

Golden Set

Forward Snowballing

Set of Papers

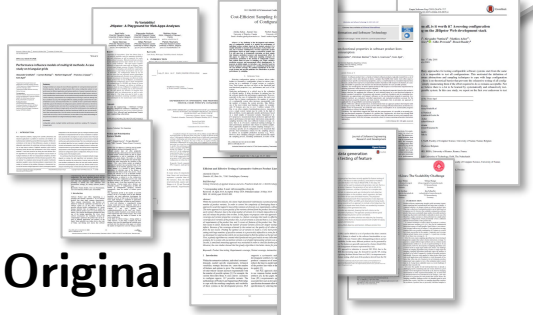
Inclusion & Exclusion Criteria

Technical Exclusion

- Conference/Journal
- English
- Available "for free" online

Content Inclusion

- Introduces, Compares, or Evaluates Sampling Approach(es)
- Topic: Product Sampling or Interaction Testing
- Feature Model, Boolean Algebra, or Decision Model



Papers from Original Survey

x Iterations

Classification

Forward and Backward Snowballing

IEEE Xplore® Springer Google Scholar

ACM DIGITAL LIBRARY Scopus WILEY Online Library

Golden Set

Forward Snowballing

Set of Papers

Inclusion & Exclusion Criteria

Technical Exclusion

- Conference/Journal
- English
- Available "for free" online

Content Inclusion

- Introduces, Compares, or Evaluates Sampling Approach(es)
- Topic: Product Sampling or Interaction Testing
- Feature Model, Boolean Algebra, or Decision Model

Results

Papers from Original Survey

x Iterations

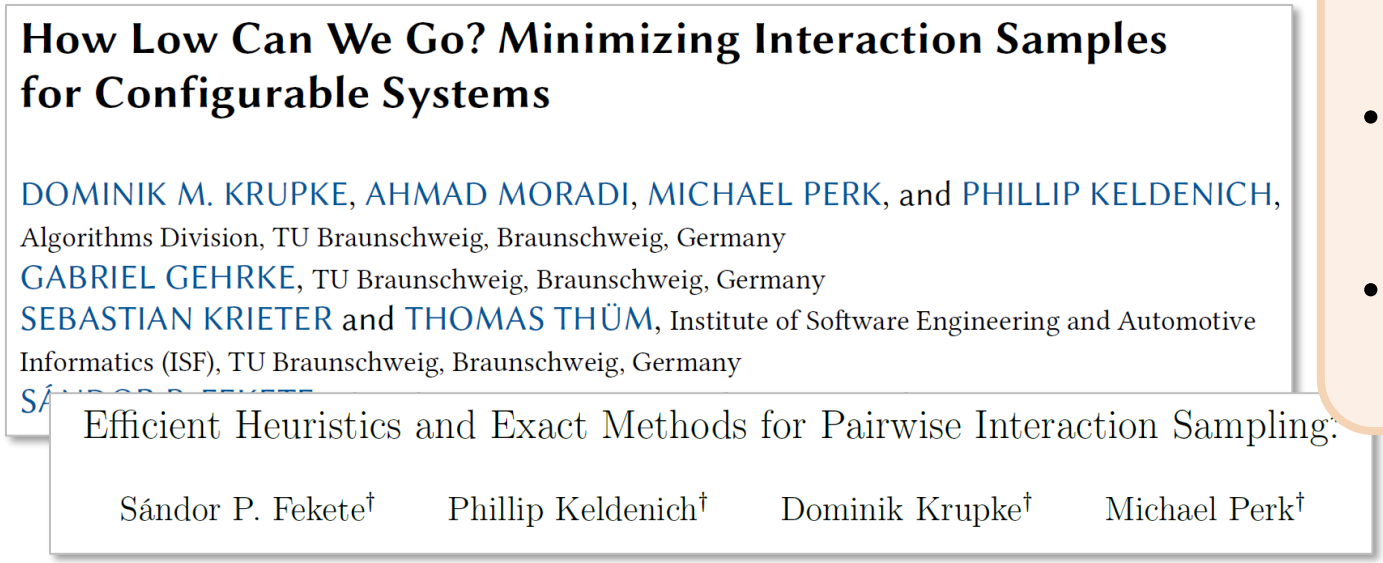
Forward and Backward Snowballing

Classification

IEEE Xplore® Springer Google Scholar

ACM DIGITAL LIBRARY Scopus WILEY Online Library

Exact Procedures for t-wise Sampling



How Low Can We Go? Minimizing Interaction Samples for Configurable Systems

DOMINIK M. KRUPKE, AHMAD MORADI, MICHAEL PERK, and PHILLIP KELDENICH,
Algorithms Division, TU Braunschweig, Braunschweig, Germany
GABRIEL GEHRKE, TU Braunschweig, Braunschweig, Germany
SEBASTIAN KRIETER and THOMAS THÜM, Institute of Software Engineering and Automotive Informatics (ISF), TU Braunschweig, Braunschweig, Germany

Efficient Heuristics and Exact Methods for Pairwise Interaction Sampling

Sándor P. Fekete† Phillip Keldenich† Dominik Krupke† Michael Perk†

Measure quality via lower bounds

- Identify the exact optimal sample when possible + proof
- If exact optimum is unknown, provide upper and lower bounds
- Estimate distance from the optimum using these bounds

Algorithms for Specific Hardware

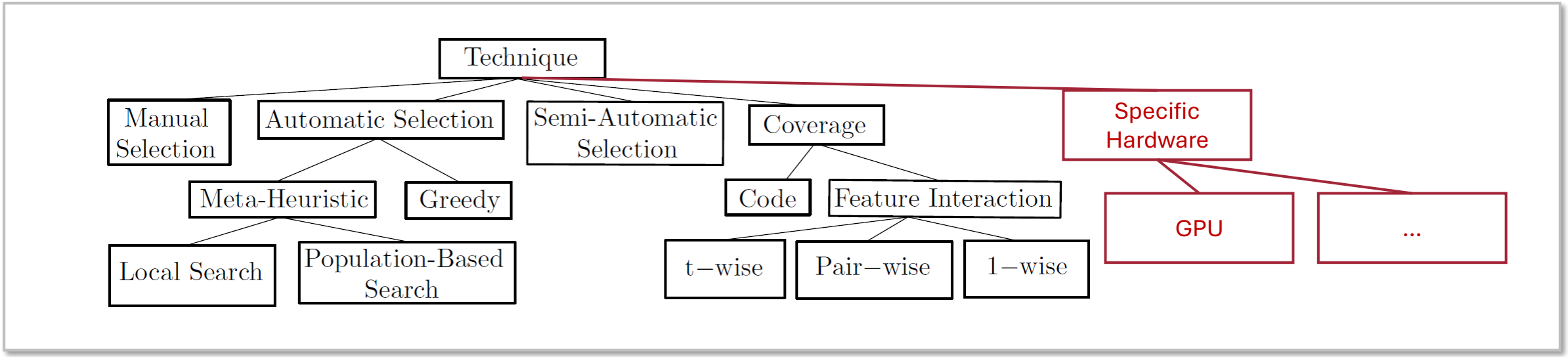
Speed Things Up: Leveraging GPU Processing Capabilities for Faster Sampling of Software Product Lines

Lennard Hettich
University of Stuttgart
Stuttgart, Germany
lennard.hettich@gmx.de

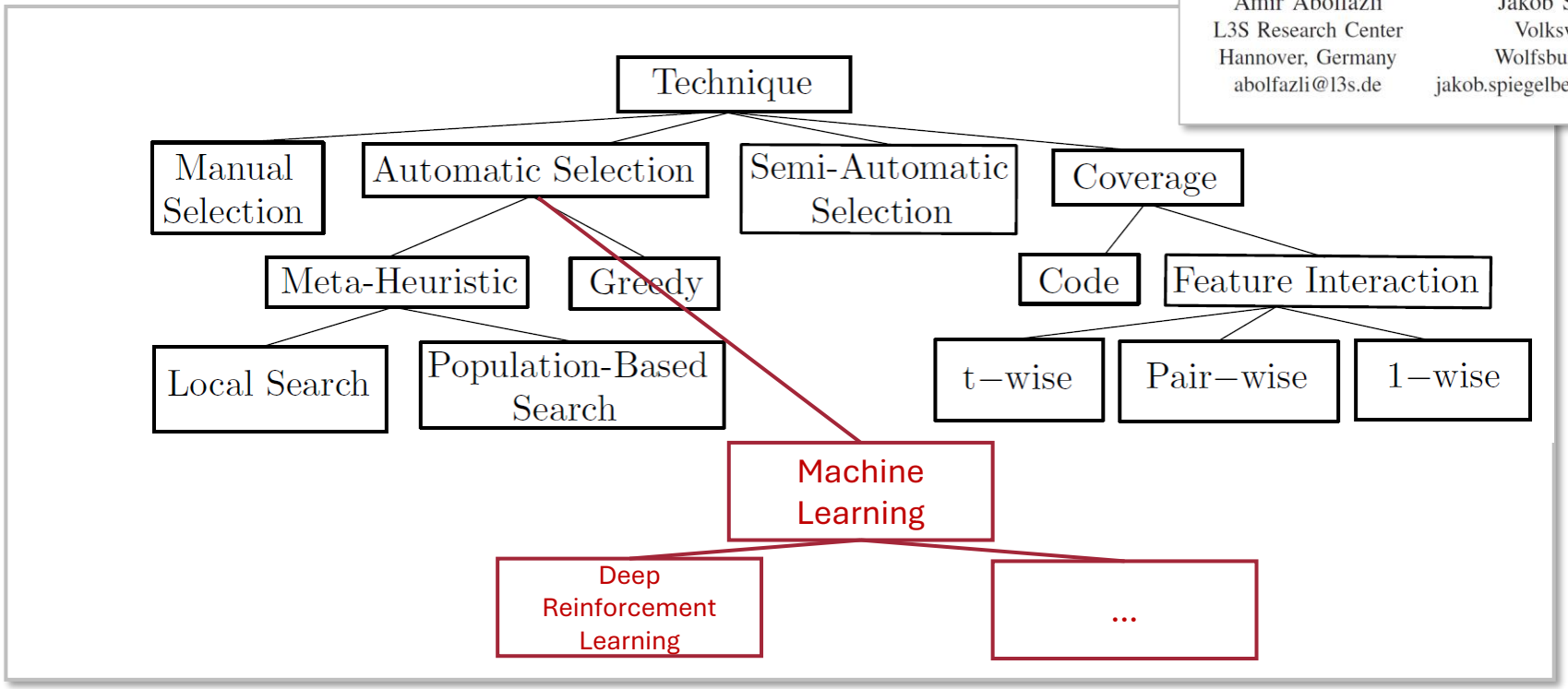
Johannes Stümpfle
University of Stuttgart
Stuttgart, Germany
johannes.stuempfle@ias.uni-stuttgart.de

Nasser Jazdi
University of Stuutgart
Stuttgart, Germany
Nasser.Jazdi@ias.uni-stuttgart.de

Michael Weyrich
University of Stuttgart
Stuttgart, Germany
michael.weyrich@ias.uni-stuttgart.de



Deep Reinforcement Learning Approach to Sampling



A Deep Reinforcement Learning Approach to Configuration Sampling Problem

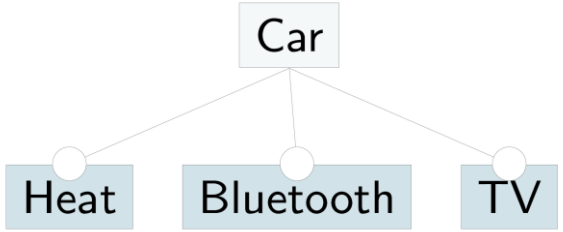
Amir Abolfazli
L3S Research Center
Hannover, Germany
abolfazli@l3s.de

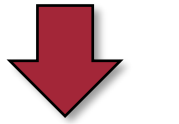
Jakob Spiegelberg
Volkswagen AG
Wolfsburg, Germany
jakob.spiegelberg@volkswagen.de

Gregory Palmer
L3S Research Center
Hannover, Germany
gpalmer@l3s.de

Avishek Anand
Delft University of Technology
Delft, Netherlands
avishek.anand@tudelft.nl

Sampling of Deployed Configurations





Covering T-Wise Interactions of Deployed Configurations

Rahel Sundermann
University of Ulm
Ulm, Germany
rahel.sundermann@uni-ulm.de

Sabrina Böhm
University of Ulm
Ulm, Germany
sabrina.boehm@uni-ulm.de

Sebastian Krieter
TU Braunschweig, Germany
Braunschweig, Germany
sebastian.krieter@tu-braunschweig.de

Malte Lochau
University of Siegen
Siegen, Germany
malte.lochau@uni-siegen.de

Thomas Thüm
TU Braunschweig, Germany
Braunschweig, Germany
thomas.thuem@tu-braunschweig.de

Iterative Sampling with Additional Analysis

Incremental Identification of T-Wise Feature Interactions

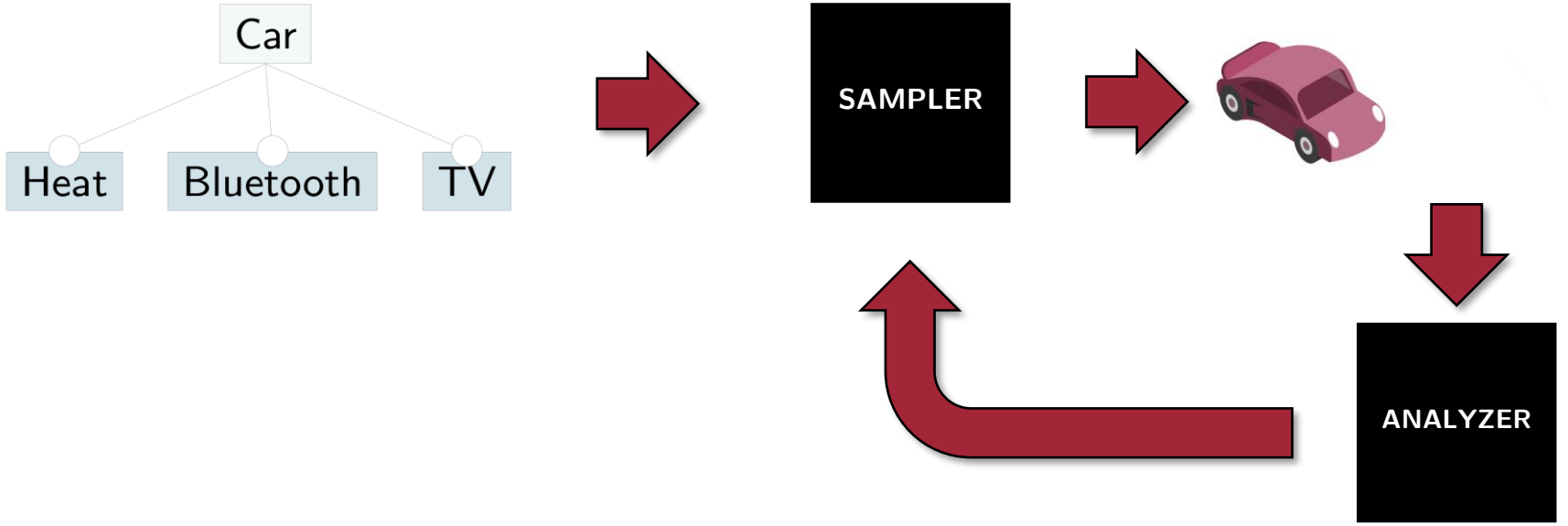
Sabrina Böhm
University of Ulm
Germany

Sebastian Krieter
University of Ulm
Germany

Tobias Heß
University of Ulm
Germany

Thomas Thüm
University of Ulm
Germany

Malte Lochau
University of Siegen
Germany



Iterative Sampling with Additional Analysis

Incremental Identification of T-Wise Feature Interactions

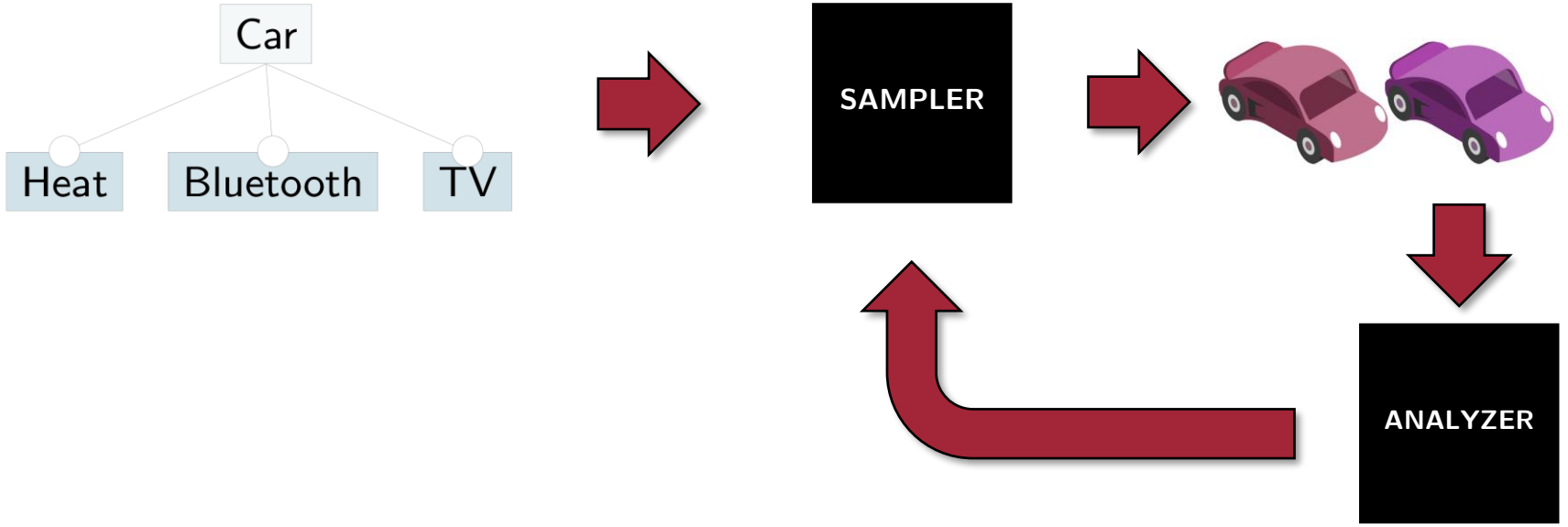
Sabrina Böhm
University of Ulm
Germany

Sebastian Krieter
University of Ulm
Germany

Tobias Heß
University of Ulm
Germany

Thomas Thüm
University of Ulm
Germany

Malte Lochau
University of Siegen
Germany



Iterative Sampling with Additional Analysis

Incremental Identification of T-Wise Feature Interactions

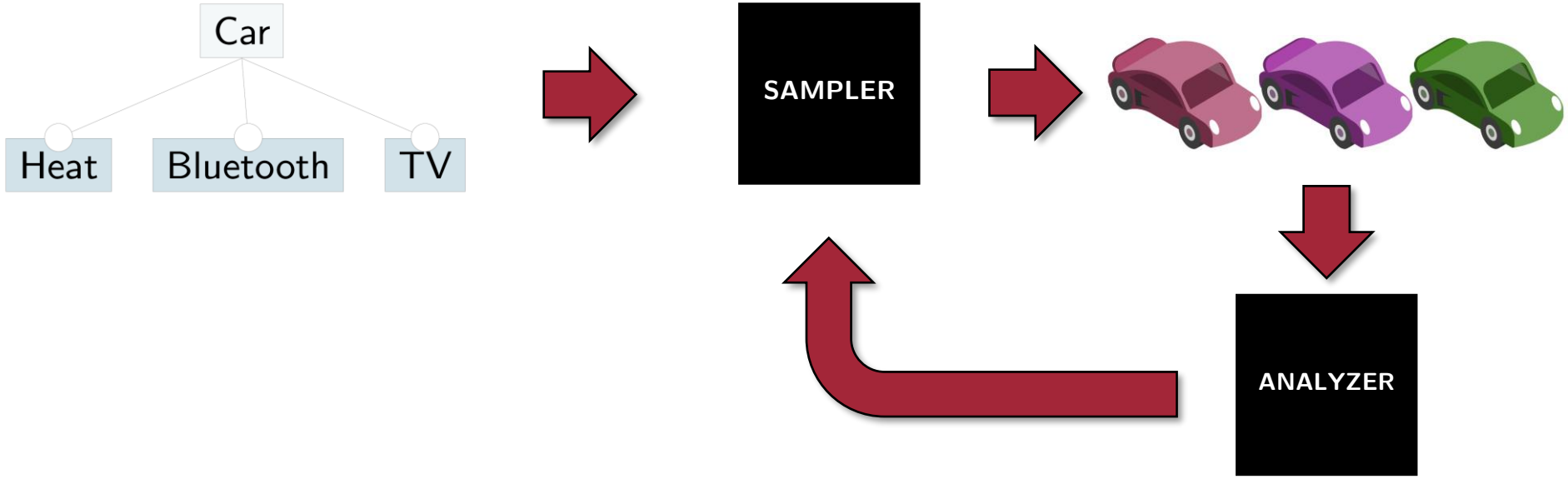
Sabrina Böhm
University of Ulm
Germany

Sebastian Krieter
University of Ulm
Germany

Tobias Heß
University of Ulm
Germany

Thomas Thüm
University of Ulm
Germany

Malte Lochau
University of Siegen
Germany



Maximum Coverage for k Configurations

Solving the *t*-Wise Coverage Maximum Problem via Effective and Efficient Local Search-Based Sampling

CHUAN LUO and JIANPING SONG, School of Software, Beihang University, Beijing, China
QIYUAN ZHAO, School of Computing, National University of Singapore, Singapore, Singapore
BINQI SUN, Chair of Cyber-Physical Systems in Production Engineering, Technical University of Munich, Munich, Germany
JUNJIE CHEN, College of Intelligence and Computing, Tianjin University, Tianjin, China
HONGYU ZHANG, School of Big Data and Software Engineering, Chongqing University, Chongqing, China
JINKUN LIN, SeedMath Technology Limited, Beijing, China
CHUNMING HU, School of Software, Beihang University, Beijing, China

Idea

Get the highest coverage for a given sample size *k*

Note

Not seen so far: "for *p*% coverage, how many configurations?"

t-wise ≠ t-wise

Coverage Metrics for T-Wise Feature Interactions

Sabrina Böhm
University of Ulm, Germany
0009-0003-4605-8574

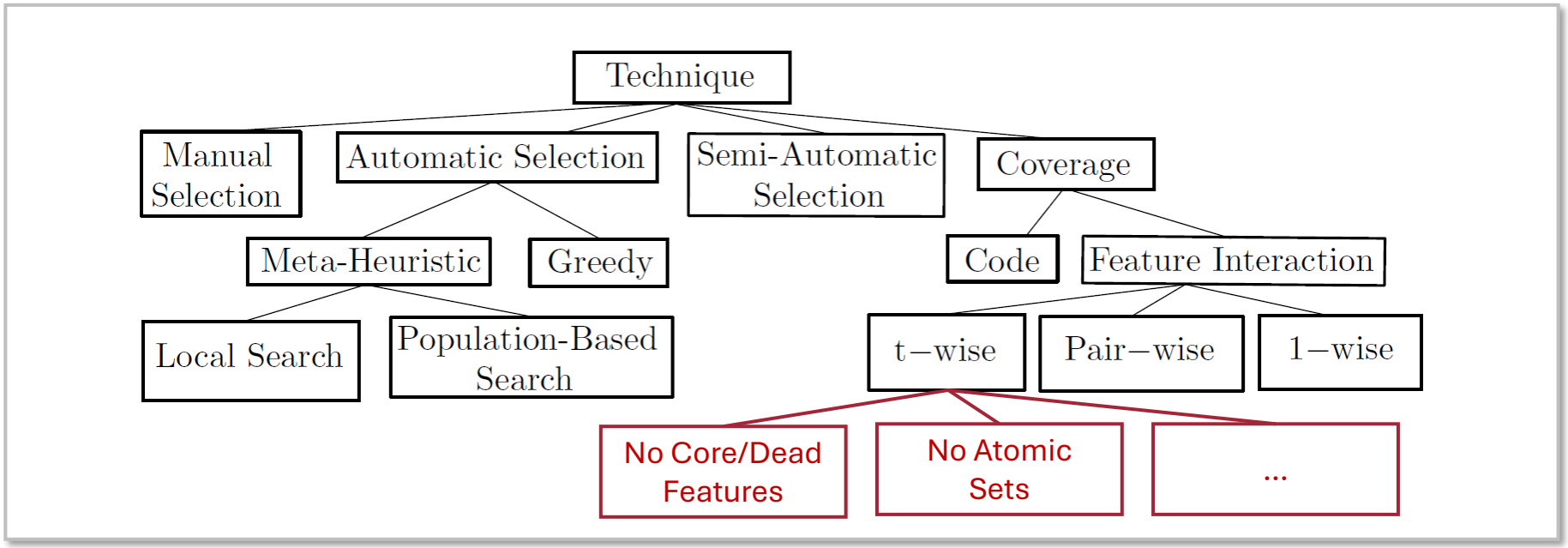
Tim Jannik Schmidt
University of Ulm, Paderborn University, Germany
0009-0005-1650-9500

Sebastian Krieter
TU Braunschweig, Germany
0000-0001-8069-9584

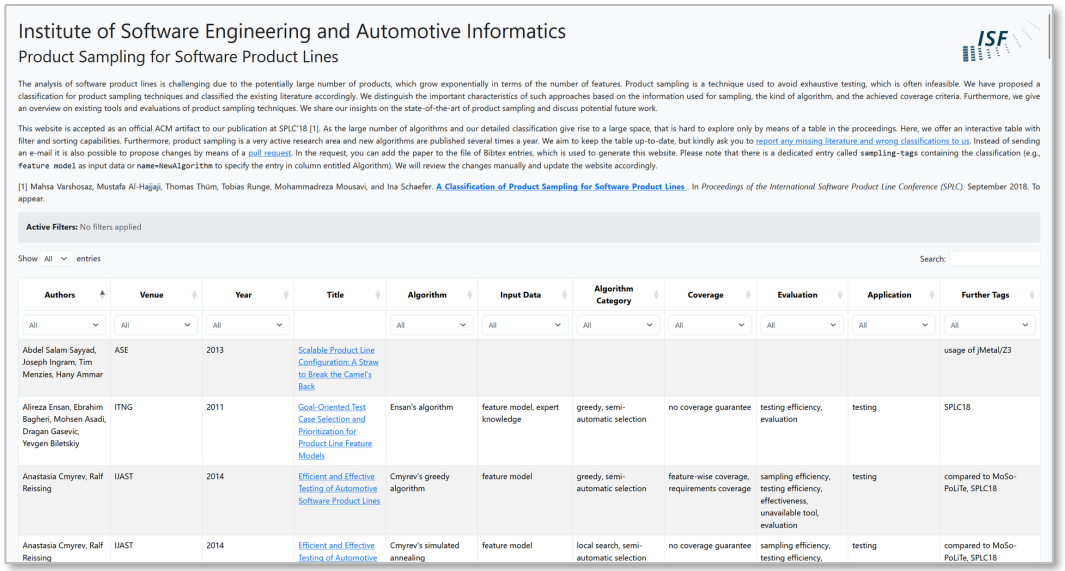
Tobias Pett
Karlsruhe Institute for Technology, Germany
0000-0001-7652-6525

Thomas Thüm
TU Braunschweig, Germany
0000-0001-8069-9584

Malte Lochau
University of Siegen, Germany
0000-0002-8404-753X



A Revised Classification of Product Sampling for Software Product Lines



Institute of Software Engineering and Automotive Informatics

Product Sampling for Software Product Lines

The analysis of software product lines is challenging due to the potentially large number of products, which grow exponentially in terms of the number of features. Product sampling is a technique used to avoid exhaustive testing, which is often infeasible. We have proposed a classification for product sampling techniques and classified the existing literature accordingly. We distinguish the important characteristics of such approaches based on the information used for sampling, the kind of algorithm, and the achieved coverage criteria. Furthermore, we give an overview on existing tools and evaluations of product sampling techniques. We share our insights on the state-of-the-art of product sampling and discuss potential future work.

This website is accepted as an official ACM artifact to our publication at SPLC'18 [1]. As the large number of algorithms and our detailed classification give rise to a large space, that is hard to explore only by means of a table in the proceedings. Here, we offer an interactive table with filter and sorting capabilities. Furthermore, product sampling is a very active research area and new algorithms are published several times a year. We aim to keep the table up-to-date, but kindly ask you to report any missing literature and wrong classifications to us. Instead of sending an e-mail it is also possible to propose changes by means of a pull request. In the request, you can add the paper to the file of Bibtex entries, which is used to generate this website. Please note that there is a dedicated entry called `sampling-tags` containing the classification (e.g., `feature model` as input data or `name=NewAlgorithm` to specify the entry in column entitled Algorithm). We will review the changes manually and update the website accordingly.

[1] Mahsa Varshosaz, Mustafa Al-Hajjaji, Thomas Thüm, Tobias Runge, Mohammadreza Mousavi, and Ina Schaefer. A Classification of Product Sampling for Software Product Lines. In *Proceedings of the International Software Product Line Conference (SPLC)*. September 2018. To appear.

Active Filters: No filters applied

Show All entries Search:

Authors	Venue	Year	Title	Algorithm	Input Data	Algorithm Category	Coverage	Evaluation	Application	Further Tags
Abdel Salam Sayyad, Joseph Ingram, Tim Menzies, Hany Ammar	ASE	2013	Scalable Product Line Configuration: A Straw to Break the Camel's Back							usage of jMetal/Z3
Alireza Ensan, Ebrahim Bagheri, Mohsen Asadi, Dragan Gasevic, Yevgen Biletskiy	ITNG	2011	Goal-Oriented Test Case Selection and Prioritization for Product Line Feature Models	Ensan's algorithm	feature model, expert knowledge	greedy, semi-automatic selection	no coverage guarantee	testing efficiency, evaluation	testing	SPLC18
Anastasia Cmyrev, Ralf Reissing	IJAST	2014	Efficient and Effective Testing of Automotive Software Product Lines	Cmyrev's greedy algorithm	feature model	greedy, semi-automatic selection	feature-wise coverage, requirements coverage	sampling efficiency, testing efficiency, effectiveness, unavailable tool, evaluation	testing	compared to MoSo-PoLiTe, SPLC18
Anastasia Cmyrev, Ralf Reissing	IJAST	2014	Efficient and Effective Testing of Automotive	Cmyrev's simulated annealing	feature model	local search, semi-automatic selection	no coverage guarantee	sampling efficiency, testing efficiency,	testing	compared to MoSo-PoLiTe, SPLC18

https://tubs-isf.github.io/

Question for you:

Are we missing recent developments or further categories?

Discoveries

- Exact Procedures for t-wise Sampling
- Algorithms for Specific Hardware
- Deep Reinforcement Learning Approach to Sampling
- Sampling of Deployed Configurations
- Iterative Sampling with Additional Analysis
- Maximum Coverage for k Configurations
- t-wise $\neq$ t-wise