

Search-based Software Engineering

Jens Kosiol

jens.kosiol@uni-kassel.de

Seminar SoSe 2024

Ausgangsbeobachtung

Viele Probleme aus dem Bereich der Softwaretechnik haben die folgenden Eigenschaften:¹

- Es gibt widersprüchliche Anforderungen.
- Man muss mit Inkonsistenzen umgehen können.
- Es gibt eine Vielzahl potentieller Lösungen.
- Typischerweise gibt es keine perfekte Lösungen, aber es ist möglich, gute Lösungen zu erkennen.
- Es gibt keinen klaren Algorithmus, um die beste/eine gute Lösung zu berechnen.

¹ Nach: Mark Harman, Bryan F. Jones: Search-based software engineering. Inf. Softw. Technol. 43(14): 833–839 (2001)

Metaheuristische Suchverfahren

- Lösungsverfahren für (kombinatorische) Optimierungsprobleme
- Durchsuchen einen Suchraum nach Lösungen von hoher Qualität
- Sind oft nicht-deterministische Näherungsverfahren
- Sind problemunabhängig
- Beispiele:
 - Evolutionäre/Genetische Algorithmen
 - Simulated Annealing
 - Ameisenalgorithmen
 - Partikelschwarmoptimierung

„Kochrezept“ Search-based Software Engineering

1. Übersetze ein Problem aus der Softwaretechnik in ein Optimierungsproblem
 - Suchraum definieren (Wie stelle ich potentielle Lösungen für mein Problem so dar, dass metaheuristische Suchverfahren damit arbeiten können?)
 - Qualität messbar machen durch Fitnessfunktionen (bezüglich dieser wird optimiert; ggf. Verwendung von Proxies)
 - Ggf. Constraints formulieren (Gibt es Arten von potentiellen Lösungen, die auf gar keinen Fall als gute Lösungen in Frage kommen und ausgeschlossen werden sollten?)
2. Wähle passendes sogenanntes **metaheuristisches Suchverfahren** (evolutionäre Algorithmen, Simulated Annealing, Schwarmalgorithmen, ...) und wende es an, um das Ausgangsproblem zu lösen.
3. (In der Wissenschaft zusätzlich: Vergleiche erzielte Qualität der Lösungen mit der Qualität, die andere Verfahren liefern)

Mögliche Seminarthemen (Auswahl)

Deploying Search Based Software Engineering with Sapienz at Facebook

- Nadia Alshahwan, Xinbo Gao, Mark Harman, Yue Jia, Ke Mao, Alexander Mols, Taijin Tei, Ilya Zorin: Deploying Search Based Software Engineering with Sapienz at Facebook. SSBSE 2018: 3–45
- Das Paper bietet einen Überblick darüber, wie Facebook SBSE praktisch einsetzt.

Approximate Oracles and Synergy in Software Energy Search Spaces

- Bobby R. Bruce, Justyna Petke, Mark Harman, Earl T. Barr:
Approximate Oracles and Synergy in Software Energy Search Spaces.
IEEE Trans. Software Eng. 45(11): 1150–1169 (2019)
- Benutzen *Genetic Improvement*, um den Energieverbrauch von Software zu optimieren
- Erlauben, dass die modifizierte Software das Verhalten der Originalsoftware lediglich approximiert

Testing with Fewer Resources

- Giovanni Grano, Christoph Laaber, Annibale Panichella, Sebastiano Panichella: Testing with Fewer Resources: An Adaptive Approach to Performance-Aware Test Case Generation. IEEE Trans. Software Eng. 47(11): 2332–2347 (2021)
- Benutzen einen genetischen Algorithmus (multi-objective), um Tests zu generieren
- Die erzeugte Testsuite soll möglichst performant sein, ohne Coverage zu opfern

Inferring test models from user bug reports

- Giovanni Guizzo, Francesco Califano, Federica Sarro, Filomena Ferrucci, Mark Harman: Inferring test models from user bug reports using multi-objective search. *Empir. Softw. Eng.* 28(4): 95 (2023)
- Leiten aus Bug Reports Modelle (Automaten) ab, die es einem erlauben, den Bug zu reproduzieren.

CODAMOSA: Escaping Coverage Plateaus in Test Generation with Pre-trained Large Language Models

- Caroline Lemieux, Jeevana Priya Inala, Shuvendu K. Lahiri, Siddhartha Sen: CodaMosa: Escaping Coverage Plateaus in Test Generation with Pre-trained Large Language Models. ICSE 2023: 919–931
- Automatisierte Generierung von Tests für Python-Code
- Kombinieren Search-Based Software Testing mit LLMs (Codex), um bessere Code-Abdeckung zu erreichen

Enabling Decision and Objective Space Exploration for Refactoring

- Soumaya Rebai, Vahid Alizadeh, Marouane Kessentini, Houcem Fehri, Rick Kazman: Enabling Decision and Objective Space Exploration for Interactive Multi-Objective Refactoring. *IEEE Trans. Software Eng.* 48(5): 1560–1578 (2022)
- Kombinieren einen genetischen Algorithmus (NSGA-III) mit Clusteringverfahren, um Nutzerinteraktion bei der Wahl von vorgeschlagenen Refactorings zu ermöglichen

Vorträge/Ausarbeitungen

- Vorstellung des Problems aus der Softwaretechnik, das gelöst wird
- Vorstellung der (Such)Methode(n) mit der/denen das Problem gelöst wird
- Umsetzung
 - Wie wird das metaheuristische Suchverfahren auf das Problem angewendet (Problemrepräsentation)?
 - Wie wird die Kombination von Methoden ermöglicht?
- Einschätzung
 - Vor- und Nachteile
 - Vergleich mit existierenden Techniken
 - Eigenes Ausprobieren

Paperliste

- Nadia Alshahwan, Xinbo Gao, Mark Harman, Yue Jia, Ke Mao, Alexander Mols, Taijin Tei, Ilya Zorin: Deploying Search Based Software Engineering with Sapienz at Facebook. SSBSE 2018: 3–45; Link: https://link.springer.com/chapter/10.1007/978-3-319-99241-9_1
- Bobby R. Bruce, Justyna Petke, Mark Harman, Earl T. Barr: Approximate Oracles and Synergy in Software Energy Search Spaces. IEEE Trans. Software Eng. 45(11): 1150–1169 (2019); Link: <https://ieeexplore.ieee.org/document/8338061>
- Rodrigo Casamayor, Carlos Cetina, Oscar Pastor, Francisca Pérez: Studying the Influence and Distribution of the Human Effort in a Hybrid Fitness Function for Search-Based Model-Driven Engineering. IEEE Trans. Software Eng. 49(12): 5189–5202 (2023); Link: <https://ieeexplore.ieee.org/document/10314766>
- Giovanni Grano, Christoph Laaber, Annibale Panichella, Sebastiano Panichella: Testing with Fewer Resources: An Adaptive Approach to Performance-Aware Test Case Generation. IEEE Trans. Software Eng. 47(11): 2332–2347 (2021); Link: <https://ieeexplore.ieee.org/document/8865437>

Paperliste (Fortsetzung)

- Giovanni Guizzo, Francesco Califano, Federica Sarro, Filomena Ferrucci, Mark Harman: Inferring test models from user bug reports using multi-objective search. *Empir. Softw. Eng.* 28(4): 95 (2023); Link: <https://link.springer.com/article/10.1007/s10664-023-10333-8>
- Giovanni Guizzo, Justyna Petke, Federica Sarro, Mark Harman: Enhancing Genetic Improvement of Software with Regression Test Selection. *ICSE 2021*: 1323–1333; Link: <https://ieeexplore.ieee.org/document/9401972>
- Caroline Lemieux, Jeevana Priya Inala, Shuvendu K. Lahiri, Siddhartha Sen: CodaMosa: Escaping Coverage Plateaus in Test Generation with Pre-trained Large Language Models. *ICSE 2023*: 919–931; Link: <https://ieeexplore.ieee.org/document/10172800>
- Soumaya Rebai, Vahid Alizadeh, Marouane Kessentini, Houcem Fehri, Rick Kazman: Enabling Decision and Objective Space Exploration for Interactive Multi-Objective Refactoring. *IEEE Trans. Software Eng.* 48(5): 1560–1578 (2022); Link: <https://ieeexplore.ieee.org/document/9200705>

Paperliste (Fortsetzung)

- Bernardino Romera-Paredes, Mohammadamin Barekatin, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, Alhussein Fawzi: Mathematical discoveries from program search with large language models. Nat. 625(7995): 468–475 (2024); Link: <https://www.nature.com/articles/s41586-023-06924-6>