

Search-based Software Engineering

Jens Kosiol

jens.kosiol@uni-kassel.de

Seminar SoSe 2023

Ausgangsbeobachtung

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

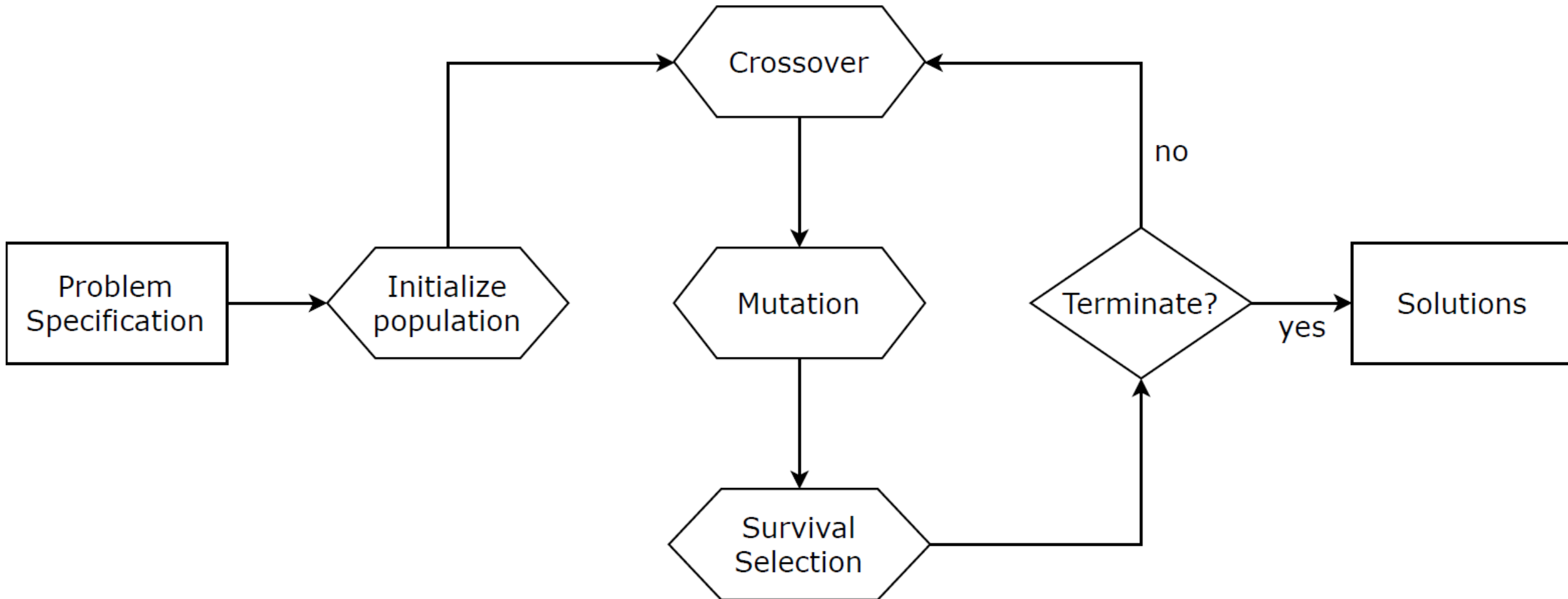
- Es gibt widersprüchliche Anforderungen
- Man muss mit Inkonsistenzen umgehen können
- Es gibt eine Vielzahl potentieller Lösungen
- Typischerweise gibt es keine perfekte Lösungen, aber es ist möglich, gute Lösungen zu erkennen
- Es gibt keinen klaren Algorithmus, um die beste/eine gute Lösung zu berechnen

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
- Meist nicht-deterministische Näherungsverfahren
- Problemunabhängig
- Beispiele: Evolutionäre/Genetische Algorithmen, Simulated Annealing, Ameisenalgorithmen, Partikelschwarmoptimierung, ...

Beispiel: Evolutionäre Algorithmen



„Kochrezept“ SBSE

- 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?)
- Wähle passendes metaheuristisches Suchverfahren und wende es an
- Vergleiche erzielte Qualität der Lösungen mit der Qualität, die andere Verfahren liefern

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

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 (preprint: https://www.carolemieux.com/codamosa_icse23.pdf)
- 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

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

Improving microservices extraction using evolutionary search

- Khaled Sellami, Ali Ouni, Mohamed Aymen Saied, Salah Bouktif, Mohamed Wiem Mkaouer: Improving microservices extraction using evolutionary search. Inf. Softw. Technol. 151: 106996 (2022)
- Benutzen einen evolutionären Algorithmus (IBEA), um Microservices aus dem Quellcode monolithischer Anwendungen zu extrahieren
- Multi-objective: lose Kopplung, hohe Kohesion, hohe Anzahl von Microservices extrahieren

Dynamic Software Containers Workload Balancing via Many-Objective Search

- Anwar Ghammam, Thiago Ferreira, Wajdi Aljedaani, Marouane Kessentini, and Ali Husain: Dynamic Software Containers Workload Balancing via Many-Objective Search. To appear in: IEEE Transactions on Services Computing (online first, 2023)
- Benutzen evolutionäre Algorithmen (z.B. NSGA-III), um Docker-Container auf Netzwerkknoten zu verteilen
- Many-objective: möglichst wenig Knoten verwenden; möglichst wenig Container pro Knoten; Minimierung der Kopplung zwischen den Knoten; Minimierung der Änderungen vom Ist-Zustand

Variability testing of software product line

- Thiago do Nascimento Ferreira, Silvia Regina Vergilio, Marouane Kessentini: Variability testing of software product line: A preference-based dimensionality reduction approach. *Inf. Softw. Technol.* 152: 107031 (2022)
- Benutzen evolutionäre Suche, um Testkonfigurationen für Softwareproduktlinien zu erstellen
- Arbeiten Nutzerfeedback ein, um während der Suche die Dimension des Suchraums reduzieren zu können

Better Automatic Program Repair by Using Bug Reports and Tests Together

- Manish Motwani, Yuriy Brun: Better Automatic Program Repair by Using Bug Reports and Tests Together. ICSE 2023 (preprint: <https://people.cs.umass.edu/~brun/pubs/pubs/Motwani23icse.pdf>)
- Kombinieren zwei Methoden der Fehlerlokalisierung, um Verfahren für die automatisierte Reparatur von Programmen zu verbessern
- Das Zusammenführen der Vorschläge für die Orte der Fehler funktioniert über ein Suchverfahren (cross-entropy Monte Carlo rank aggregation)

Requirements traceability recovery for the purpose of software reuse

- Mohamed Salah Hamdi, Adnane Ghannem, Marouane Kessentini: Requirements traceability recovery for the purpose of software reuse: an interactive genetic algorithm approach. *Innov. Syst. Softw. Eng.* 18(1): 193–213 (2022)
- Erzeugen automatisiert Verbindungen zwischen Stellen im Quellcode und Anforderungen; auch anwendbar auf „neue“ Anforderungen und „alten“ Quellcode
- Nutzen einen genetischen Algorithmus, berücksichtigen dabei, wie Quellcode bereits wiederverwendet wurde, und ermöglichen User-Feedback

Multi-objective software performance optimisation at the architecture level

- Youcong Ni, Xin Du, Peng Ye, Leandro L. Minku, Xin Yao, Mark Harman, Ruliang Xiao: Multi-objective software performance optimisation at the architecture level using randomised search rules. *Inf. Softw. Technol.* 135: 106565 (2021)
- Benutzen einen eigens entworfenen evolutionären Algorithmus, um Softwarearchitekturen auf Performance hin zu optimieren
- Entwerfen eigene Suchregeln für den Suchprozess, um erklärbare Ergebnisse zu erhalten

Multi-objective code reviewer recommendations

- Soumaya Rebai, Abderrahmen Amich, Somayeh Molaei, Marouane Kessentini, Rick Kazman: Multi-objective code reviewer recommendations: balancing expertise, availability and collaborations. *Autom. Softw. Eng.* 27(3): 301–328 (2020)
- Automatisieren das Finden von passenden Reviewern für Code (Algorithmus: NSGA-II)
- Multi-objective: Die Reviewer sollten geeignet sein, Zeit haben und möglichst wenig mit der Person zusammengearbeitet haben, deren Code gereviewed wird

Moderne Forschungsthemen SBSE

- Kombination von Methoden
- Inkorporation von Domänenwissen
- Erklärbarkeit der Ergebnisse

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

Paperliste

- Chaima Abid, Dhia Elhaq Rzig, Thiago do Nascimento Ferreira, Marouane Kessentini, Tushar Sharma: X-SBR: On the Use of the History of Refactorings for Explainable Search-Based Refactoring and Intelligent Change Operators. *IEEE Trans. Software Eng.* 48(10): 3753–3770 (2022)
- Vahid Alizadeh, Marouane Kessentini, Mohamed Wiem Mkaouer, Mel Ocinneide, Ali Ouni, Yuanfang Cai: An Interactive and Dynamic Search-Based Approach to Software Refactoring Recommendations. *IEEE Trans. Software Eng.* 46(9): 932–961 (2020)
- 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)
- Thiago do Nascimento Ferreira, Silvia Regina Vergilio, Marouane Kessentini: Variability testing of software product line: A preference-based dimensionality reduction approach. *Inf. Softw. Technol.* 152: 107031 (2022)
- Anwar Ghammam, Thiago Ferreira, Wajdi Aljedaani, Marouane Kessentini, Ali Husain: Dynamic Software Containers Workload Balancing via Many-Objective Search. To appear in: *IEEE Transactions on Services Computing* (online first, 2023)

Paperliste (Fortsetzung)

- 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)
- Giovani Guizzo, Justyna Petke, Federica Sarro, Mark Harman: Enhancing Genetic Improvement of Software with Regression Test Selection. ICSE 2021: 1323–1333
- Mohamed Salah Hamdi, Adnane Ghannem, Marouane Kessentini: Requirements traceability recovery for the purpose of software reuse: an interactive genetic algorithm approach. Innov. Syst. Softw. Eng. 18(1): 193– 213 (2022)
- 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 (preprint: https://www.carolemieux.com/codamosa_icse23.pdf)
- Thainá Mariani, Marouane Kessentini, Silvia Regina Vergilio: Generation of refactoring algorithms by grammatical evolution. Empir. Softw. Eng. 27(5): 110 (2022)
- Manish Motwani, Yuriy Brun: Better Automatic Program Repair by Using Bug Reports and Tests Together. ICSE 2023 (preprint: <https://people.cs.umass.edu/~brun/pubs/pubs/Motwani23icse.pdf>)

Paperliste (Fortsetzung)

- Youcong Ni, Xin Du, Peng Ye, Leandro L. Minku, Xin Yao, Mark Harman, Ruliang Xiao: Multi-objective software performance optimisation at the architecture level using randomised search rules. *Inf. Softw. Technol.* 135: 106565 (2021)
- Soumaya Rebai, Abderrahmen Amich, Somayeh Molaei, Marouane Kessentini, Rick Kazman: Multi-objective code reviewer recommendations: balancing expertise, availability and collaborations. *Autom. Softw. Eng.* 27(3): 301–328 (2020)
- 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)
- Khaled Sellami, Ali Ouni, Mohamed Aymen Saied, Salah Bouktif, Mohamed Wiem Mkaouer: Improving microservices extraction using evolutionary search. *Inf. Softw. Technol.* 151: 106996 (2022)
- Makram Soui, Mabrouka Chouchane, Mohamed Wiem Mkaouer, Marouane Kessentini, Khaled Ghédira: Assessing the quality of mobile graphical user interfaces using multi-objective optimization. *Soft Comput.* 24(10): 7685–714 (2020)