
Master Thesis Topic

An Evaluation of Metaheuristic Search Strategies For Automatic Software Repair

Motivation and Background

Repairing software defects is a very expensive task. It consumes up to 90% of the total lifecycle cost of a given piece of software[1]. The necessity of effective and automatic tools to find and fix software defects is high. Different approaches can be found in the research community, from bug localization based on contracts, invariants, or machine learning, until those that use Search-Based Software Engineering (SBSE) Metaheuristics like GenProg[2] to locate and fix software bugs. The later ones have gained special attention in the last few years, given that it has been demonstrated that such techniques can solve many Software Engineering problems faster and cheaper compared with traditional approaches. Nevertheless, other metaheuristics like Ant Colony Optimization[3], Tabu Search[4] and Simulated Annealing[5] have been not evaluated to fix bugs. These are presented as promising approaches to solve combinatorial optimization problems. Modeling the automatic software repair problem as one of them can bring interesting results. Additionally, the analysis of the Search Space and the adaptation of the SBSE algorithm to it can bring better results. This has not been done so far and represents a big challenge towards the complete automation of software fixing.

Goals

- Evaluate (based on simulations and realistic examples) SBSE Metaheuristics (Ant Colony Optimization, Tabu Search and Simulated Annealing) to automatically repair software defects.
- Adapt this algorithms with the results of Search Space Classification and demonstrate how this improves the automatic software repair.
- Analyze the results and determine why these algorithms outperform the existing ones by automatic software repair.

Description of the Task

The project aims to apply Ant Colony Optimization, Tabu Search and Simulated Annealing to automatically repair software defects:

- Understand Ant Colony Optimization, Tabu Search and Simulated Annealing.
- Implement these algorithms.
- Perform an experimental evaluation. Compare with current benchmarks
- Analyze the results. Determine why these algorithms, and under which circumstances, they outperform current approaches.

Research Type

Theoretical Aspects:

**** *

Industrial Relevance:

**** *

Implementation

**** **

Prerequisite

The student should be enrolled in the bachelor/master of software engineering/informatics program, and has completed the required course modules to start a bachelor/master thesis.

Skills required

Programming skills in Java or C++, Understanding of, or willingness to learn, the software engineering and statistical foundations needed for the project.

References

- [1] Le Goues, Claire. Automatic program repair using genetic programming. Diss. University of Virginia, 2013.
- [2] C. Le Goues, T. Nguyen, S. Forrest, and W. Weimer. Genprog:generic method for automatic software repair. *Software Engineering, IEEE Transactions*38(1):54–72, 2012b..
- [3] Dorigo, Marco, and Mauro Birattari. "Ant colony optimization." *Encyclopedia of Machine Learning*. Springer US, 2010. 36-39.
- [4] Glover, Fred, and Manuel Laguna. *Tabu search*. Springer US, 1999.
- [5] Van Laarhoven, Peter JM, and Emile HL Aarts. *Simulated annealing*. Springer Netherlands, 1987.

Contacts

Lars Grunske, Humboldt-Universität zu Berlin, Institut für Informatik, Lehrstuhl Software Engineering, Unter den Linden 6, 10099 Berlin, Germany

Application

Please contact me during my office hours or send me an email with the title: “[ThesisProject]-SBSE4AGA” to se-career@informatik.hu-berlin.de