Guest Editorial

He Jiang, Dalian University of Technology, China
Ke Tang, University of Science and Technology of China, China
Justyna Petke, University College London, UK
Mark Harman, University College London, UK

Search Based Software Engineering

Search Based Software Engineering (SBSE) consists of the application of Computational Intelligence (CI) algorithms to hard optimization problems in Software Engineering (SE). It has become an important application field for CI. The term SBSE was coined by Harman and Jones in 2001, although there was work on the application of CI algorithms to SE before this date.

After more than fifteen years development, CI algorithms have been used to solve SE tasks in almost all the stages of an SE lifecycle, including requirements, designing, coding, testing and maintenance. Using SBSE, an SE task can be solved by three steps. First, define a suitable candidate solution representation or search space for the software problem. Second, define the suitable objective function for the search space. Third, employ a variety of search algorithms to seek solutions within its search space.

From the perspective of CI algorithms, a wide range of CI algorithms are successfully applied in SE, including genetic algorithms, genetic programming, random walk, differential evolution, multi-objective evolutionary algorithms, etc. From the perspective of SE, almost all the stages of SE are practical applications for CI algorithms. Among these stages, software testing is the most flourishing one. Many research achievements and CI based open-source tools are published to conduct test case generation, test case priorities, etc.

Due to the flourishing applications of CI algorithms for SE, we organized this special issue to solicit research results and novel applications of SBSE. As a result, the special issue received 14 submissions. Each submission underwent a rigorous review process, at the end of which we were able to select three papers to be included in the special issue. Many of the other submissions were of high-quality and we expect to see them published in due course elsewhere.

The three papers finally selected for this special issue focus primarily on software testing. Multiple CI algorithms are leveraged to solve diverse problems in this stage, including test suite reduction, software production line testing, and test case generation.

The *first paper* is “Multi-Level Random Walk for Software Test Suite Reduction” authored by Zongzheng Chi, Jifeng Xuan, Zhilei Ren, Xiaoyuan Xie, and He Guo. The authors address the problem of software test suite reduction to support continuous development and testing. The problem is defined as providing a subset of test suite without losing pre-defined test requirements. In the paper, a novel multi-level random walk algorithm is proposed.

The proposed algorithm converts the problem of finding a minimized test suite into a search-based problem, namely finding “backbones” of test case instances and refining the final solution. The “backbones” is defined as the common part (i.e., the intersection) of local optima. In the process of finding the backbones, the authors leverage the common part of local optima by random walk search to simplify the original problem instance. They iteratively reduce the problem scale by fixing the backbone and discarding shielded test cases with no contribution to test requirements. They repeat extracting the backbone and reducing the scale for multiple times until a small problem is achieved. In the process of refining the final solution, a solution to the simplified problem is combined with the backbones to refine the final solution to the original problem.

Their experiments explore the algorithms’ ability of test suite reduction in three directions, i.e., the similarity between local and global optima, the change of backbone scales with the number of local optima, and the ability of problem scale downgrading with levels. Extensive experiments confirm that the proposed algorithm is superior to five state-of-the-art heuristics for several open source Java projects within less time.

The *second paper* is “Hyper-Heuristic Based Product Selection for Software Product Line Testing”. In the paper, Thiago N. Ferreira, Jackson A. Prado Lima, Andrei Strickler, Josiel N. Kuk, Silvia R. Vergilio, and Aurora Pozo introduce a Hyper-Heuristic (HH) approach to dynamically select the best operator for feature model selection in software product line testing.

Feature model selection is a problem to select the features of the most interesting ones in a software product line for testing. In their paper, they propose a HH approach to select the best Multi-Objective Evolutionary Algorithms (MOEAs) for feature model selection. More specifically, the authors employ three objectives for consideration, namely the number of products, dead mutants, and covered pairs. The HH algorithm dynamically selects the best operators at a given moment in the evolution process. In their work, they implement and evaluate four MOEAs, including NSGA-II, SPEA2, IBEA, and MOEA/D-DRA, and two selection methods, namely random and UCB (Upper Confidence Bound) based methods.

Evaluation results show that the HH-based NSGA-II algorithm generates the best results. Moreover, the UCB method outperforms the random selection in large

instances. Their investigation shows new insights for using HH approaches in software product line testing.

The *third paper* is “Differential Evolution Based on Self-Adaption Fitness Function for Automated Test Case Generation”. Han Huang, Fangqing Liu, Xiaoyan Zhuo, and Zhifeng Hao propose a Differential Evolution (DE) algorithm for automatic test case generation in white-box testing.

In their work, the authors first define a self-adaption fitness function to prompt the meta-heuristic algorithm to use more information from the found paths. The proposed fitness function aims to lead the meta-heuristic algorithm to produce more uncovered paths based on the found paths. The fitness value of the test case is in direct proportion to the number of uncovered branches of the found paths. With the proposed fitness function, the authors choose the DE algorithm to make the most use of the found solutions to produce new solutions. They modify the DE algorithm by adding “DE/rand/1” mutation.

There are several findings in their work. First, the proposed self-adaption fitness function enhances the capacity of the modified DE for 100% path coverage with the minimum number of test cases. Second, the higher performance of the proposed DE can be attributed to the property that the modified DE together with the self-adaption fitness function makes effective use of the found paths.

We hope that this special issue provides insights for SBSE. We thank the authors who submitted their interesting work to this special issue. We are in debt to the reviewers who gave timely and high-quality reviews on the submissions to help the authors to improve their work. We would also like to thank the Editor in Chief, Dr. Hisao Ishibuchi for the support throughout the preparation of this special issue. We would like to thank the IEEE Computational Intelligence Magazine for hosting this special issue.