Guest Editorial for the Special Section from the 9th International Symposium on Search Based Software Engineering

More than any time in the past, software really matters. We are no longer in the era of “internet of things” but the “internet of things that really matter”. Software is essential to international, financial and transport systems; our energy generation and distribution systems; and even the pacemakers that control the beat of our hearts. Looking forward, to the forthcoming age of autonomous cars and flying drones, it is clear that software (written in traditional programming languages or in some next-generation interpretation) will be the key that determines what we can do, when, where, and how.

Search Based Software Engineering (SBSE) is an approach to software engineering in which search based optimisation algorithms are used to identify optimal or near-optimal solutions and give insight into various software engineering tasks, ranging from model-based reasoning, project management to software testing and maintenance and evolution.

To better explore this problem domain, researchers in this area meet at SSBSE - the International Symposium on Search-based Software Engineering. Now a dozen years old, SSBSE meets annually to discuss the latest results as well as guess what might be the best way forward.

In 2017, SSBSE met in Paderborn, Germany (co-located with ESEC/FSE’17)[1]. Photos from that event (http://ssbse17.github.io/photos/) document the energy of that meeting. The event itself was organized by Prof. Lars Grunske (Humboldt University of Berlin, Germany) while we served as Co-chairs of the program review committee.

The aim of this special section is to present revised and extended versions of some of the best papers from that meeting. The SSBSE’17 program committee made some “distinguished papers” awards and these were invited to extend their conference work for this journal. These extended papers were then rigorously assessed (single-blind review). Based on that review, and the subsequent refinements and clarifications, three papers were selected for this special section. One concerned with a novel test suite generation algorithm that considers a large number of objectives and limited budget, and outperforms state-of-the-art under those conditions, while the other two address the difficult problem of selecting which search-based algorithm should be applied to which kind of test generation problem.

In “Test Suite Generation with the Many Independent Objective (MIO) Algorithm”, Andrea Arcuri

[1] 11th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering

evaluates a new algorithm that is designed to be efficient for test suite generation with a large number of objectives and limited search budgets. It is tailored based on the specific properties of test suite generation. A thorough empirical study shows that the proposed MIO algorithm can achieve higher coverage than other state-of-the-art search-based test generation algorithms.

In “An Empirical Evaluation of Evolutionary Algorithms for Unit Test Suite Generation”, José Campos, Yan Ge, Gordon Fraser, Marcelo Eler and Andrea Arcuri discuss how to optimize the optimization of test suites. This article empirically evaluates over a dozen of evolutionary algorithms (and two random approaches). Their results show that the choice of algorithm can have a substantial influence on the performance of whole test suite optimisation. Also, evolutionary algorithms are not as good as evolutionary methods (for this task) and the DynaMOSA many-objective search algorithm was the most effective seen in that sample. That said, no algorithm was clearly superior in all cases, which motivated future work on improved search algorithms for unit test generation.

In “A Large Scale Empirical Comparison of State-of-the-art Search-based Test Case Generators”, Annibale Panichella, Fitsum Meshesha Kifetew and Paolo Tonella test six test case generation methods on 180 non-trivial Java classes. In that sample, they found that single target approaches were generally superior to multi-target approaches. Also, in a result that endorses the other paper in this collection, within the multi-target approaches, DynaMOSA/MOSA performed best.

Acknowledgments: We would like to thank all authors and all reviewers for the effort they made for this special section. Special thanks go to the staff of the Elsevier Information and Software Technology Journal and especially to Prof. Claes Wohlin for the support throughout the reviewing process.

And our final thanks go to SSBSE’17’s general chair Prof. Lars Grunske (pictured, centred) who taught us how conference organizers can better serve the research community. Research is a very serious business-- but it can also be fun.

Guest Editor
Justyna Petke
Principal Research Fellow
and Proleptic Senior Lecturer (Associate Prof.)
at University College London

Guest Editor
Tim Menzies
Professor at North Carolina State University