

Introduction to the special issue on search based software engineering

Massimiliano Di Penta · Simon Poulding

Published online: 20 January 2011
© Springer Science+Business Media, LLC 2011

This special issue of Empirical Software Engineering consists of revised and extended versions of five selected papers originally presented at the 1st International Symposium on Search Based Software Engineering (SSBSE 2009). The symposium was held at Cumberland Lodge, Windsor, UK, in May 2009, and brought together the academic researchers and industrial practitioners that make up the international Search Based Software Engineering (SBSE) community for the first conference dedicated to the subject.

Search Based Software Engineering is the application of search-based optimization techniques—for example, genetic algorithms, hill climbing, simulated annealing, taboo search, and ant colony optimization—to the solution of software engineering problems. Many such problems are often difficult or impossible to solve manually, and moreover, as modern software becomes more complex and adopts distributed and highly parallel architectures, are proving to be increasingly intractable using traditional automated techniques. In contrast, SBSE approaches can leverage sophisticated optimisation algorithms running on affordable computing infrastructure to solve these problems. SBSE approaches also demonstrate other advantages: they can be used to find acceptably good solutions when time, or resources, are limited, and—by utilizing multi-objective search algorithms—can derive a range of solutions that illustrate potential trade-offs between competing goals. For these reasons, both academic and industrial interest in SBSE is increasing rapidly. A recent survey,¹ undertaken by the UK-based SEBASE project, indicated an exponential growth of

¹<http://www.sebase.org/sbse/publications/>

M. Di Penta (✉)
Department of Engineering, University of Sannio, Benevento, Italy
e-mail: dipenta@unisannio.it

S. Poulding
Department of Computer Science, University of York, York, UK
e-mail: smp@cs.york.ac.uk

the number of SBSE publications in the last ten years, and in 2009 (the last full year for which figures are available) over 120 such papers were published worldwide.

Early applications of search-based techniques to software engineering tended to focus on approaches to generating test data. (A survey of these approaches has been published by McMinn (2004) and recently by Ali et al. (2010)). However, over the last decade, the scope of SBSE has expanded to cover domains throughout the engineering lifecycle, such as: software project management, cost and effort estimation, requirements prioritization and selection, architectural design, software refactoring and modularization, concurrent software verification, fault prediction, system tuning, and automated bug fixing.

A common characteristic of SBSE research is the importance of convincing and principled empirical work. This is difficult enough within the software engineering domain itself, but the utilization of stochastic search algorithms as solution techniques presents a further challenge. It is often far from obvious that a proposed SBSE technique outperforms simpler approaches (such as manual solution, or naive automated approaches involving random or exhaustive search); that the approach has general utility across a range of problem instances; and that the approach can be applied to software of realistic scale. Empirical work is vital in supplying such evidence, as has been highlighted in the recent paper (Ali et al. 2010), which surveys and provides guidance as to empirical work on search-based software testing.

The five papers in this special issue show the wide range of problem domains and solution techniques that are actively being researched by the SBSE community, as well as that effective empirical work is at the core of the most important SBSE research.

In the paper “WCET Analysis of Modern Processors Using Multi-Criteria Optimisation”, Bate and Khan empirically investigate the effectiveness of search-based techniques to find the Worst Case Execution Time (WCET) of a program on complex processors with varying levels of cache, branch prediction and out-of-order execution. The paper shows that there is no single criteria that guides the search in all circumstances, but that different criteria should be adopted in different situations.

In the paper “A Study of the Bi-Objective Next Release Problem”, Durillo, Zhang, Harman, and Nebro investigated the effectiveness of three state-of-the-art multi-objective metaheuristics (NSGA-II, MOCell, and PAES) in solving the next release problem, i.e. determining the (near) optimal set of software features to be included in the next release of the product that accommodates the conflicting objectives of cost and customers’ satisfaction. Results from a study on six academic problems and data from an industrial project show that (i) NSGA-II provides the highest number of optimal solutions, (ii) MOCell provides the widest range of different solutions, and (iii) PAES is the fastest technique.

In the paper “Evaluating Improvements to a Meta-Heuristic Search for Constrained Interaction Testing”, Garvin, Cohen, and Dwyer reformulate Combinatorial Interaction Testing (CIT), a sampling technique for finding interaction faults in highly-configurable systems, using simulated annealing and with an approach to efficiently incorporate constraints. The application of the technique to a set of 35 realistic constrained problems and on some unconstrained problems show that the proposed approach reduces the run time by a factor of 90 while achieving the same coverage objectives as the non-optimized technique.

In the paper “Using Search-Based Algorithms for Ajax Event Sequence Generation during Testing”, Marchetto and Tonella deal with state-based testing of Web

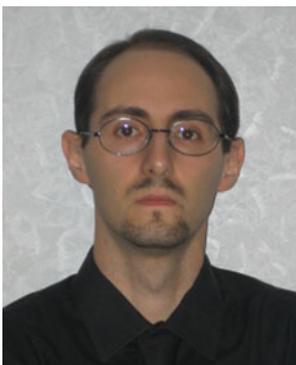
applications making use of Ajax (i.e., of asynchronous communication between client pages and Web server). In such a context, the interaction between a page and a Web server is abstracted using a finite state machine and sequence of interacting events. The authors apply search-based optimization techniques, namely hill-climbing and simulated annealing, to handle the search space of interacting event sequences which can grow exponentially in size with sequence length. The approach has been evaluated on two open-source Ajax applications, and results show that the heuristics help to keep the number of sequences to be explored manageably small while preserving their ability to reveal faults.

In the paper “Design Evolution Metrics for Defect Prediction in Object Oriented Systems”, Kpodjedo, Ricca, Galinier, Guéhéneuc and Antoniol propose an approach aimed at predicting defects in a software system based on a set of metrics describing how the design evolves, i.e., how the number of attributes, methods and relations change in a class over the time. The metrics are calculated by an algorithm that utilizes taboo search. The authors compare the proposed approach to existing known metrics applied to several releases of three open-source systems. Results indicate that the proposed metrics help to achieve a better prediction of defects when used in conjunction with the existing metrics suite, compared to the use of the existing suite alone, and that they make significantly better predictions of defect density than other metrics.

If you are new to the field of Search Based Software Engineering, these papers serve as a very informative introduction to the subject. If you are an existing SBSE researcher or practitioner, you will hopefully find useful examples of problems, techniques, and empirical methods that are new to you. In either case, we hope you enjoy the papers in this special issue.

References

- Ali S, Briand LC, Hemmati H, Panesar-Walawege RK (2010) A systematic review of the application and empirical investigation of search-based test case generation. *IEEE Trans Softw Eng* 36:742–762
- McMinn P (2004) Search-based software test data generation: a survey. *Softw Test Verif Reliab* 14(2):105–156



Massimiliano Di Penta is assistant professor at the University of Sannio, Department of Engineering, Italy. He received his laurea degree in Computer Engineering in 1999 and his PhD in Computer

Engineering in 2003. His research interests include software maintenance and evolution, reverse engineering, empirical software engineering, search-based software engineering service-centric software engineering. He is author of over 130 papers appeared on journals, conferences and workshops. He serves and has served in the organizing and program committees of several conferences such as ICSE, ASE, ICSM, ICPC, CSMR, GECCO, MSR, SCAM, WCRE, and many others. He has been general chair of SCAM 2010, SSBSE 2010, WSE 2008, general co-chair of WCRE 2008, and program co-chair of SSBSE 2009, WCRE 2006 and 2007, IWPSE 2007, WSE 2007, SCAM 2006, STEP 2005, and of other workshops. He is steering committee member of ICPC, SCAM, CSMR, WCRE, IWPSE, and SSBSE. He is in the editorial board of the Empirical Software Engineering Journal edited by Springer, and of the Journal of Software Maintenance and Evolution: Research and Practice, edited by Wiley.



Simon Poulding received a BSc in Mathematics and an MSc in Mathematics with Modern Applications from the University of York in 1989 and 2006, respectively. From 1989 to 2005 he was a software engineer in the financial and healthcare industries, specializing in enterprise management systems and database applications. He is currently a lecturer in the Department of Computer Science at the University of York. His research interests are search-based statistical testing, the scalability and robustness of algorithms used by search-based software engineering techniques, and the use of reliable and efficient empirical methods in computer science.