

Chapter 44

Evolutionary Approaches to Test Data Generation for Object–Oriented Software: Overview of Techniques and Tools

Ana Filipa Nogueira

Polytechnic Institute of Leiria, Portugal & University of Coimbra, Portugal

José Carlos Bregieiro Ribeiro

Polytechnic Institute of Leiria, Portugal

Francisco Fernández de Vega

University of Extremadura, Spain

Mário Alberto Zenha-Rela

University of Coimbra, Portugal

ABSTRACT

In object-oriented evolutionary testing, metaheuristics are employed to select or generate test data for object-oriented software. Techniques that analyse program structures are predominant among the panoply of studies available in current literature. For object-oriented evolutionary testing, the common objective is to reach some coverage criteria, usually in the form of statement or branch coverage. This chapter explores, reviews, and contextualizes relevant literature, tools, and techniques in this area, while identifying open problems and setting ground for future work.

INTRODUCTION

Search-Based Software Engineering (SBSE) seeks to reformulate Software Engineering (SE) problems as search-based optimisation problems. It has been applied to a wide variety of SE areas, including requirements engineering, project planning and cost estimation, automated maintenance, service-oriented software engineering, compiler optimisation and quality assessment (Harman, 2007). Most of the overall literature (an estimated 59%) in the SBSE area is, however, concerned with Software Testing (ST) related applications, with structural test data generation being the most studied sub-topic (Harman, Mansouri, & Zhang, 2009).

The application of Evolutionary Algorithms (EAs) to test data generation or selection is often referred to as *Evolutionary Testing (ET)* (Tonella, 2004b; Wappler & Wegener, 2006b) or *Search-Based Test Data Generation (SBTDG)* (McMinn, 2004). ET consists of exploring the space of test programs by using metaheuristic techniques that direct the search towards the potentially most promising areas of the input space (Bertolino, 2007); its foremost objective is usually that of searching for a set of test programs that satisfies a predefined test criterion.

EAs have already been applied, with significant success, to the search for test data; the first application of heuristic optimisation techniques to test data generation was presented in 1992 (Xanthakis, Ellis, Skourlas, Gall, & K. Karapoulios, 1992). However, research has been mainly geared towards generating test data for procedural software, and traditional methods – despite their effectiveness and efficiency – cannot be applied without adaptation to Object-Oriented (OO) systems.

The application of search-based strategies to unit testing of OO programs is, in fact, fairly recent – the first approach was presented in 2004 (Tonella, 2004b) – and is yet to be investigated comprehensively (Harman, Hassoun, Lakhotia, McMinn, & Wegener, 2007). Interesting review articles on the topic of SBSE, and Search-Based Software Testing (SBST) in particular, include: (McMinn, 2004; Mantere & Alander, 2005; Xiao, El-Attar, Reformat & Miller, 2007; Afzal, Torkar, & Feldt, 2009; Ali, Briand, Hemmati, & Panesar-Walawege, 2009; Harman et al., 2009; Harman & McMinn, 2010; Maragathavalli, 2011; McMinn, 2011; Harman, Mansouri, & Zhang, 2012; Varshney & Mehrotra, 2013; Anand *et al.*, 2013). McMinn surveys the use of metaheuristic search techniques for the automatic generation of test data (McMinn, 2004); because the work on SBST had, thus far (2004), been largely restricted to programs of a procedural nature, these are the main subject of this review. In (Mantere & Alander, 2005), a review of the application of Genetic Algorithm (GA)-based optimisation methods to ST is presented; the authors stress out that all the researchers in this area report good (or, at least, encouraging) results regarding their use. Xiao *et al.* reported the experimental results regarding the effectiveness of five different optimisation techniques over five different C/C++ programs (Xiao *et al.*, 2007). The results show that the GA-based approach outperformed the remaining techniques – e.g., Simulated Annealing (SA), Genetic Simulated Annealing (GSA), Simulated Annealing with Advanced Adaptive Neighborhood (SA/ANN) and Random Testing – achieving the best overall performance.

The work proposed in (Afzal *et al.*, 2009) builds on on McMinn's research (McMinn, 2004) and presented a review on how search-based techniques are used to test non-functional properties of the software, focusing on: the properties studied, the fitness functions implemented and the constraints and limitations found when testing each property. The set of non-functional properties identified include (listed in descending order of the number of papers that investigate it): execution time, security, usability, safety and quality of service. A systematic review on the way SBST techniques have been empirically assessed is presented in (Ali *et al.*, 2009). In (Harman et al., 2009, 2012) a thorough index and classifica-

24 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/evolutionary-approaches-to-test-data-generation-for-object-oriented-software/294500

Related Content

Addressing Highly Dynamic Changes in Service-Oriented Systems: Towards Agile Evolution and Adaptation

Andreas Metzger and Elisabetta Di Nitto (2014). *Software Design and Development: Concepts, Methodologies, Tools, and Applications* (pp. 164-177).

www.irma-international.org/chapter/addressing-highly-dynamic-changes-service/77704

Issues about the Adoption of Formal Methods for Dependable Composition of Web Services

Manuel Mazzara and Michele Ciavotta (2014). *International Journal of Systems and Service-Oriented Engineering* (pp. 35-50).

www.irma-international.org/article/issues-about-the-adoption-of-formal-methods-for-dependable-composition-of-web-services/119658

Information Feedback Based Architecture for Handling the Scalability Issues in the Reusable Cloud Components

Manjunath Ramachandra and Pandit Pattabhirama (2012). *Software Reuse in the Emerging Cloud Computing Era* (pp. 186-202).

www.irma-international.org/chapter/information-feedback-based-architecture-handling/65172

Developing Semantically-Enabled Families of Method-Oriented Architectures

Mohsen Asadi, Bardia Mohabbati, Dragan Gašević, Ebrahim Bagheri and Marek Hatala (2012). *International Journal of Information System Modeling and Design* (pp. 1-26).

www.irma-international.org/article/developing-semantically-enabled-families-method/70923

Assessing and Benchmarking Sustainability in Organisations: An Integrated Conceptual Model

Arunasalam Sambhanthan (2017). *International Journal of Systems and Service-Oriented Engineering* (pp. 22-43).

www.irma-international.org/article/assessing-and-benchmarking-sustainability-in-organisations/201206