Handling constraints in combinatorial interaction testing in the presence of multi objective particle swarm and multithreading

Bestoun S. Ahmed a,b,*, Luca M. Gambardella c, Wasif Afzal c, Kamal Z. Zamli d

a Istituto Dalle Mole di Studi sullIntelligenza Artificiale (IDSA), CH-6928 Manno-Lugano, Switzerland
b Department of Computer Science, Faculty of Electrical Engineering, Czech Technical University, Karlovo nam. 13, 121 35 Praha 2, Czech Republic
c School of Innovation, Design and Engineering, Mirdalven University, Sweden
d Faculty of Computer Systems and Software Engineering, University Malaysia Pahang, Gambang, Malaysia

A R T I C L E   I N F O

Article history:
Received 13 February 2016
Revised 14 February 2017
Accepted 14 February 2017
Available online 20 February 2017

Keywords:
Constrained combinatorial interaction
Multi-objective particle swarm optimisation
Test generation tools
Search-based software engineering
Test case design techniques

A B S T R A C T

Context: Combinatorial testing strategies have lately received a lot of attention as a result of their diverse applications. In its simple form, a combinatorial strategy can reduce several input parameters (configurations) of a system into a small set based on their interaction (or combination). In practice, the input configurations of software systems are subjected to constraints, especially in case of highly configurable systems. To implement this feature within a strategy, many difficulties arise for construction. While there are many combinatorial interaction testing strategies nowadays, few of them support constraints.

Objective: This paper presents a new strategy, to construct combinatorial interaction test suites in the presence of constraints.

Method: The design and algorithms are provided in detail. To overcome the multi-judgement criteria for an optimal solution, the multi-objective particle swarm optimisation and multithreading are used. The strategy and its associated algorithms are evaluated extensively using different benchmarks and comparisons.

Results: Our results are promising as the evaluation results showed the efficiency and performance of each algorithm in the strategy. The benchmarking results also showed that the strategy can generate constrained test suites efficiently as compared to state-of-the-art strategies.

Conclusion: The proposed strategy can form a new way for constructing of constrained combinatorial interaction test suites. The strategy can form a new and effective base for future implementations.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

In the last decade, various studies on combinatorial interaction approaches have gained a lot of awareness and several test generation approaches were developed. In software engineering, combinatorial interaction testing (CIT) aims to generate minimised test suites that manipulate the variables of input parameters based on their combination. Each combination could form a specific configuration of the software-under-test (SUT). The goal is to cover all possible r-combinations (sometimes called r-tuples) by an optimised set (where r is the interaction strength) [1]. This could be a difficult task in case of highly configurable systems, which leads to combinatorial explosion and non-deterministic polynomial-time hard (NP-hard) problems [2]. In addition, the problem of constrained interactions has recently appeared [3].

Nowadays, software development is shifted from building an isolated, product-by-product approach to software product lines (SPL) [4]. In addition to the many features this approach provides like minimising the cost and market reachability time, it facilitates the idea of customisable software products. With this approach, there are many customisable features that could be added to or extracted from the functionality of a specific software based on the needs of the developers or customers. Eclipse represents a well-known example of SPL in which many functions and plug-ins (i.e., having different features) can be added or extracted from its main framework [5]. Evidence showed that most of the faults may result due to the interaction among these features [16]. Hence, all the interactions must be tested carefully. However, in reality, there are many constraints among the features. Some features must or must not appear with others and they may be included or excluded from the test suite during the testing process. CIT strategies can tackle these interactions efficiently, however, with ordinary

* Corresponding author. Tel: +420 224 355 752.
E-mail addresses: albebes@fel.cvut.cz, bestoon82@gmail.com
(B.S. Ahmed), luca@idsaia.ch (L.M. Gambardella), wasif.afzal@mdh.se (W. Afzal), kamalz@ump.edu.my (K.Z. Zamli).

http://dx.doi.org/10.1016/j.infsof.2017.02.004
0950-5849/© 2017 Elsevier B.V. All rights reserved.