International Journal of Open Source Software and Processes
Volume 12 • Issue 2 • April-June 2021

Search-Based Regression Testing Optimization

Nagwa R. Fisal, Suez Canal University, Egypt
Abeer Hamdy, British University in Egypt, Egypt
Essam A. Rashed, Suez Canal University, Egypt

ABSTRACT

Regression testing is one of the essential activities during the maintenance phase of software projects. It is executed to ensure the validity of an altered software. However, as the software evolves, regression testing becomes prohibitively expensive. In order to reduce the cost of regression testing, it is mandatory to reduce the size of the test suite by selecting the most representative test cases that do not compromise the effectiveness of the regression testing in terms of fault-detection capability. This problem is known as test suite reduction (TSR) problem, and it is known to be an NP-complete. The paper proposes a multi-objective adapted binary bat algorithm (ABBA) to solve the TSR problem. The original binary bat (OBBA) algorithm was adapted to enhance its exploration capabilities during the search for a Pareto-optimal surface. The effectiveness of the ABBA was evaluated using six Java programs with different sizes. Experimental results showed that for the same fault discovery rate, the ABBA is capable of reducing the test suite size more than the OBBA and the BPSO.

KEYWORDS

Binary Bat Algorithm, Multi-Objective Optimization, Mutation Testing, Regression Testing, Search-Based Software Engineering, Software Testing, Test Suite Reduction

INTRODUCTION

As software testing is known to be an expensive process, open-source software is usually released with several bugs; e.g., at the early releases of Mozilla and Eclipse, about 170 and 120 bugs respectively were reported daily (Abeer Hamdy & El-Laithy, 2020; Abeer Hamdy & Ellaithy, 2020; Abeer Hamdy & Ezzat, 2020). It is essential to design a cost-effective test plan that detects as many defects as possible before the release of the open-source software to ensure the quality of the delivered software. Especially, during the maintenance phase, enhancements and modifications are made to the software, which necessitates the development and execution of new test cases to test the modifications; in addition to the re-execution of the earlier test cases, to test the software stability after enhancements (Catal & Mishra, 2013). Testing the behavior of the whole system under test (SUT) before release and after each modification is called regression testing (Leung & White, 1989; Rosero, Gómez, & Rodríguez, 2016). The cost of regression testing increases over time due to the increase in the test suite size. So, it is important to find the smallest representative subset of the test suite without compromising the fault-detection capability of the original test suite (Gotlieb & Marijan, 2014; Nadeem & Awais, 2006). This problem is known as test suite reduction problem (TSR). One way to assess the capabilities of the reduced test suite, in discovering bugs, is through the utilization of a fault-based testing technique called mutation testing. Mutation testing calculates a score for the

DOI: 10.4018/IJOSSP.2021040101

test suite which indicates its capabilities on discovering bugs in the SUT (Jia & Harman, 2010). The TSR problem is known to be a combinational optimization problem that can be described as a set covering problem which is known to be NP-complete (Gary & Johnson, 1979). In practice, there is no efficient solution for NP-complete problems. However, suboptimal solutions could be found using search-based optimization (SBO) algorithms (Chen & Lau, 1998). Bat algorithm (BA) is a recent and efficient SBO algorithm, which mimics the echolocation behavior of bats to find a global optimal solution (Yang, 2010). The performance of the BA was reported in the literature to be superior to other SBO algorithms such as the particle swarm optimization (PSO) (Eberhart & Kennedy, 1995; A Hamdy & Mohamed, 2019) and Genetic algorithms (GA) (Abeer Hamdy, 2014), over the majority of benchmark functions and real applications.

Aims and Contributions

The aim of this paper is to reduce the cost of the regression testing, through reducing the test suite size using the BA. Our contributions to accomplish this aim are summarized as follows:

- Proposing modifications to the Original Binary Bat Algorithm (OBBA) (Mirjalili, Mirjalili, & Yang, 2014) to enhance its exploration and exploitation capabilities; so to reduce its occasionally failure to converge to global optimum solutions.
- Formulating the TSR problem in terms of two objectives which are: the cost of the reduced test suite and the mutation score. Then, applying the variable weighted sum method (Yang, 2011) to guide the Adapted binary BA (ABBA) search for the non-dominated solutions that form a Pareto-optimal surface.
- Evaluating the performance of the ABBA against each of the OBBA and the Binary Particle Swarm Optimization BPSO (Bansal et al., 2011) in solving the multi-objectives TSR problem over six Java programs of different test suite sizes and different number of mutants.

The rest of the paper is organized as follows: Section 2 introduces some important preliminaries for this work. Section 3 discusses the previous studies that tackled the TSR problem. Section 4 presents the multi-objective adapted binary bat algorithm for solving the TSR problem. Section 5 discusses the experiments and results. Finally, Section 6 concludes the paper and introduces possible extensions to this work.

BACKGROUND

Test Suite Reduction Problem

Given: A test suite TS which includes d test cases, and a set of n mutants $\left\{mu_1,\ldots,mu_n\right\}$, that should be killed to provide an adequate testing of the SUT. Each test case tc_j can kill one or more mutants mu_i.

Problem: Find an adequate subset $TS' \subseteq TS$ that can kill as many as possible number of mutants and includes as few as possible number of test cases. These two objectives are contradictory; this is the reason we formulated the TSR problem as a multi-objective optimization problem.

Pareto Optimal Concepts

In multi-objective optimization problems, there is no single solution but a set of multiple trade-off solutions (Ngatchou, Zarei, & El-Sharkawi, 2005). The vector of decision variables that optimizes the considered objective functions and satisfy the problem constrains is called a Pareto front. Thus,