Chapter 34
Building Ant System for Multi-Faceted Test Case Prioritization: An Empirical Study

Manoj Kumar Pachariya
MCNUJC, Bhopal, Madhya Pradesh, India

ABSTRACT

This article presents the empirical study of multi-criteria test case prioritization. In this article, a test case prioritization problem with time constraints is being solved by using the ant colony optimization (ACO) approach. The ACO is a meta-heuristic and nature-inspired approach that has been applied for the statement of a coverage-based test case prioritization problem. The proposed approach ranks test cases using statement coverage as a fitness criteria and the execution time as a constraint. The proposed approach is implemented in MatLab and validated on widely used benchmark dataset, freely available on the Software Infrastructure Repository (SIR). The results of experimental study show that the proposed ACO based approach provides near optimal solution to test case prioritization problem.

1. INTRODUCTION

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing demonstrates the business view and helps in identifying the risks generated during software development (Kumar et al., 2013; Kumar et al., 2014). The prominent objectives of software testing are to meet the requirements that guided its design and development, work as expected by client, and satisfy the needs of stakeholders. Testing is done to ensure code and data flow coverageability of the program. The code coverageability includes the statement, branch, loop, and path coverageability. Testing team has to ensure that all program elements are executed at least once (Kumar et al., 2014; Epitropakis et al., 2015; Li et al., 2007).

In regression testing, execution of the selected test cases on SUT (Software Under Test) enhances the confidence, reliability and quality of software product (Li et al., 2007). The regression testing process is

DOI: 10.4018/978-1-6684-3702-5.ch034

concerned to maintenance phase. In regression testing, there is a need of rerunning the already executed test suite on software under test (SUT) after modification carried out in original software. In regression testing, test case prioritization is required to expose the faults in SUT at the early hours. Re-executing the complete test suite is not practical and not cost effective. Software industry requires cost effective approach for testing the software product adequately due to lacking time and resource constraint. The test case prioritization approaches will meet the requirement of software industry (Islam et al., 2012; Sun et al., 2013).

For testing a program, a test case is an input value for determining the failure and pass of program. It is used to ensure the validation of product and verification of process. The test case optimization is process of identifying the minimal cardinality subset of test cases from the large pool of test cases. The test case optimization is commonly concern with test case minimization, selection, prioritization, filtration and classification. The test case optimization reduces the efforts; duration and cost of testing as it provides optimal subset of test cases for audit (Kumar et al., 2011a; Kumar et al., 2011b; Kumar et al., 2011c).

Test case minimization is the process of identifying and removing the redundant test cases from large pool according to the objectives of testing. This subset of test cases is used to audit the program. This subset attains the same value of objective as the entire pool does. The identification of minimal cardinality subset of test cases from the large pool of test case is called the test case selection. The test case minimization reduces the set permanently by removing redundant and obsolete test case in the set while test case selection temporarily chooses optimal or the best fit test cases from the large pool according the test fitness criterion (Kumar et al., 2012; Kumar 2015).

For attaining the optimal value of testing objectives as early as possible, the ranking/ ordering/ scheduling of test cases is known as test case prioritization. Test case filtration is to chunk out subset of closely related test cases to optimize the objectives of testing (Tyagi & Malhotra, 2014). Test case prioritization is a critical problem of software testing. Since several factors may be considered in order to find the best order for test cases, several search-based techniques have been applied to find solutions for test case prioritization problem (Li et al., 2007; He & Bai, 2015).

Genetic algorithms are used to reorder test cases in a test suite using execution time as a constraint had shown that prioritization technique is appropriate for manual regression testing environment and explains how the baseline approach can be extended to operate in additional time constrained testing circumstances. Most of the researchers had explored genetic algorithm, ant colony optimization, linear programming etc. based approaches to find out the subset of test cases from a large pool of test cases but multi-faceted test case prioritization has not been explored and evaluated thoroughly (Sabharwal et al., 2011).

Literature study is the evidence that most of the researchers have explored and applied the greedy algorithms for test case prioritization. These algorithms provide the suboptimal solutions by identifying the local optimal solution in search space. For identification of the global optimal solution of test case optimization problem, nature inspired, and evolutionary algorithms are most suitable and helpful (Li et al., 2007; Kumar et al., 2011; Walcott et al., 2006). Some of these works apply ant colony-based algorithm, but the Statement Coverage along with time constraint was not considered. On the basis of fault detecting capability many interesting results have been received but the test case prioritization based on statement coverage with time constraint using Ant Colony Optimization technique has not been explored. So, there is still space for the researchers to experiment and validate the ant colony optimization-based approach to find out the order of test cases on the basis of statement coverageability (Singh et al., 2010;