Search for Prioritized Test Cases during Web Application Testing

Munish Khanna, YMCA University of Science & Technology, Faridabad, India
Naresh Chauhan, YMCA University, Faridabad, India
Dilip Kumar Sharma, GLA University, Mathura, India

ABSTRACT

Regression testing of evolving software is a critical constituent of the software development process. Due to resources constraints, test case prioritization is one of the strategies followed in regression testing during which a test case that satisfies predefined objectives the most, as the tester perceives, would be executed the earliest. In this study, all the experiments were performed on three web applications consisting of 65 to 100 pages with lines of code ranging from 5000 to 7000. Various state-of-the-art approaches such as, heuristic approaches, Greedy approaches, and meta heuristic approaches were applied so as to identify the prioritized test sequence which maximizes the value of average percentage of fault detection. Performance of these algorithms was compared using different parameters and it was concluded that the Artificial Bee Colony algorithm performs better than all. Two novel greedy algorithms are also proposed in the study, of which the goal is to smartly manage the state of a tie, where a tie exhibits the condition that all the test cases participating in the tie are of equal significance in achieving the objective. It has also been validated that the performance of these novel proposed algorithm(s) is better than that of traditionally followed greedy approach, most of the time.

KEYWORDS
2-opt Algorithm, Artificial Bee Colony Algorithm (ABC), Genetic Algorithm (GA), Greedy Algorithm, Immune Genetic Algorithm (IGA), Search Algorithms, Test Case Prioritization, Web Application testing

1. INTRODUCTION

In the present E-Commerce-oriented state of business, web applications play very vital role. These are buildup of different web technologies and are hosted over web servers so that users can access them using intranet or internet through a web browser. Being volatile in nature, effective and consistent testing of web applications becomes necessary. In order to satisfy the never-ending expectations of the customers and to sustain in a highly competitive era, frequent updates and up gradations in web applications (dynamic websites) on hourly or daily basis are required, without suspending the provided services. Regression testing is required to validate these updates and up gradations because any fault in a specific page can interfere with one or more functionalities of the website. Moreover, structure of these applications is complex; any dynamic website consists of a number of pages that are, in turn, connected to several other pages hence fault on any one page can hamper performance. The efficacy of proper testing is to be measured in terms of effective and complete testing within a limited time.

Regression testing is one of the inevitable processes executed during the maintenance phase of SDLC (Chauhan, 2010; Mathur, 2012; Singh, 2012).

DOI: 10.4018/IJAMC.2019040101

Copyright © 2019, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
The formal definition of regression testing as specified by IEEE is,

“Selective retesting of a system or component to verify that modifications have not caused unintended effects and that the system or component still complies with its specified requirement…” (IEEE Standard Computer Dictionary). The main objective behind its execution is to ensure that changes introduced in the delivered software do not initiate any new fault in the modified as well as the previously flawlessly working and tested code. It is a complex process to certify that recently introduced changes/features do not hamper the performance of the existing, unchanged part of the software.

The formal definition (Rothermal, Untch, & Harrold, 2001) of Test Case Prioritization (TCP), one of the strategies followed during regression testing, is given below.

Given: Let a test suite T be given along with the set of all possible permutations (ordering) of T, PT, of T and a function f from PT to the real numbers.

Problem: Find a new permutation T’ which belongs to the set of permutations PT such that its value for f(T’) is greater than or equal to any other permutation T” which also belongs to PT where PT represents the set of all possible permutations of T, and f is a function that, when applied to any such ordering, yields an award value for that ordering. It is assumed that higher award values are preferable to lower ones. In other words, f is a quantitative measure that satisfies the goal of escalating the likelihood of satisfying predefined objectives.

TCP is a discrete combinatorial problem which falls under the class of NP-Hard problems (Cormen, Leiserson, Rivest & Stein, 2001; Horowitz, Sahni & Rajasekaran, 2007; Jingui, Min & Immune, 2010) and its time complexity is exponential in nature. Finding optimal sequence of a large size test suite is a challenging task for which various attempts have been made by the researcher’s fraternity; however, finding a near optimal solution could be an alternative. Several software engineering problems are formulated as single objective optimization problems and TCP is one example. During TCP there could be various objective(s) to be satisfied. However, in this study finding the highest rate of fault coverage is the objective. During literature study, it has been noted that diverse optimization problems have been effectively solved using nature inspired approaches and the achieved results are optimal or nearby optimal in nature. Various nature inspired techniques such as Hill Climbing, Genetic Algorithm and Tabu search have already been applied to solve test cases prioritization problem (Li, Harman, & Hierons, 2007).

In the published literature, authors (Li, Harman, & Hierons, 2007) stated that it is not possible to find which fault would be exposed by a test case in advance i.e., before actual completion of the testing process of the software. Hence their work focuses on coverage of various parameters (block, decision and statement) of the program. However, this proposed work emphasizes on coverage of faults only, using random fault seeding concept, with the support of test cases versus fault matrix, fault severity matrix and test case cost matrix. Faults belonging to various categories were manually seeded in the software (dynamic website) at random locations, so as to achieve the essence of real faulty software, and the corresponding test cases which would expose these faults were generated using Selenium testing tool.

APFD is a measure of the rate of fault detection (Rothermal, Untch, Chu, & Harrold, 1999), and a measurement of efficiency during prioritization of test cases when fault coverage is the criteria. The best approach for achieving maximum APFD value lies in the fact that maximum faults should be exposed, at faster rate, while executing minimum number of test cases. In other words, test cases having maximum fault coverage should be executed earliest.

The APFD for the test cases sequence T is given by

\[
APFD = 1 - \frac{TC_1 + TC_2 + \ldots + TC_m}{nm} + \frac{1}{2n}
\]

(1)
Related Content

**Pseudo-Cut Strategies for Global Optimization**
[www.igi-global.com/article/pseudo-cut-strategies-global-optimization/62302?camid=4v1a](www.igi-global.com/article/pseudo-cut-strategies-global-optimization/62302?camid=4v1a)

**Predicting Uncertain Behavior and Performance Analysis of the Pulping System in a Paper Industry using PSO and Fuzzy Methodology**
The Information Paradox: Researching Health Service Information Systems Development
www.igi-global.com/article/the-information-paradox/152591?camid=4v1a

A Gravitational Search Algorithm Approach for Optimizing Closed-Loop Logistics Network
www.igi-global.com/chapter/gravitational-search-algorithm-approach-optimizing/69899?camid=4v1a