Abstract

Test case prioritization (TCP) techniques rank test cases by optimizing specific criteria, such as rate of bug detection or the rate of code coverage. Usually, a TCP technique assigns a priority score to each test case based on the number of previously undetected faults it detects or the extent of incremental code coverage it achieves. A tester can select the most important test cases from a prioritized list based on time and cost constraints. A large number of TCP techniques have been proposed by researchers based on various criteria, such as code coverage, requirements coverage, model coverage, etc. However, there is scope for further improvements.

In this context, we first propose a TCP technique based on the ability of the test cases to detect "hard to detect" bugs (HDFDC). In our method, we first seed a large number of bugs into a program to generate mutants. Each mutant is executed with the test suite, and the execution results are recorded in a fault matrix. Using the fault matrix, we first prioritize the test cases based on their 'hard to detect' values. Our technique assigns higher priority to those test cases which reveal bugs in a program-component that are hard to expose. The remaining test cases are prioritized using our proposed metric, Fault Detection Capability (FDC), which measures the bug detection capability of test cases. Our experimental results showed that our proposed TCP technique outperformed existing TCP techniques by 43.82% on an average in terms of the average percentage of faults detected (APFD) metric.

Subsequently, we propose three novel fault-based regression test case prioritization techniques (TCP) for object-oriented (OO) programs. All these three prioritization techniques are mutation-based. These three prioritization techniques are: optimized version of bug detection based prioritization technique (optimized BD-TCP), expected value of bug detection-based prioritization (Exp-TCP), and Bayesian probabilistic based prioritization (Bayesian-TCP). Finally, we ensemble the results of these three base prioritizers (optimized BD-TCP, Exp-TCP, Bayseian) and our previously proposed HDFDC technique using three ensemble methods: Averaging, Kendall Tau ranking, and Schulze's Algorithm of ranking. The application of Schulze's Algorithm and the Kendall Tau metric to ensemble results of TCP techniques has not previously been reported in the literature. Our experimental results showed that Averaging, Kendall Tau, and Schulze ensemble methods achieved 27.87%, 31.71%, and 31.58% average APFD score improvements respectively over existing techniques.

Sometimes the different modules of a system have widely different fault proneness, due to factors such as whether the functionalities are complex, the competency of the team that developed it, and the time at which it was developed. In this context, we propose a novel regression test case prioritization (RTCP) technique based on the fault proneness of a software module. First, we build a neural network model to predict the fault proneness of a module by considering different parameters such as bug history, program complexity, etc. Next, the test cases were prioritized based on their bug-detection capability in highly fault-prone modules. A test case which detects more bugs in highly fault-prone modules compared to other test cases is assigned a higher priority. Our experimental studies show that our fault proneness-based test case prioritization approach outperformed related TCP techniques in terms of APFD score.

We have experimentally studied the performance of all our proposed prioritization techniques and observed that the Kendall Tau approach outperformed other techniques in terms of early fault detection, whereas the Exp-TCP technique underperformed other techniques for all the considered projects.

Keywords: Regression testing, Regression test case prioritization, Mutation testing, Object-oriented programs, Ensemble learning, Fault proneness, Artificial Neural Network