Empirical Evaluation of Test Suite Reduction

Adrian Regenfuß

December 7, 2020

Abstract

As a response to ever-growing test suites with long runtimes, several different approaches have been developed to reduce the time for test suite execution to give useful results. One of these approaches is test suite reduction: selecting a sample of tests that maximizes coverage on the tested source code. This paper attempts to replicate the findings in **?**, which borrows techniques from big data to handle very large test suites. We use independently generated testing data from open source projects, and find that TODO.

1 Introduction

2 Related Work

3 Handling Large Test Suites

3.1 Methods

3.1.1 Test Case Selection

3.1.2 Test Case Priorization

3.1.3 Test Suite Reduction

3.2 Algorithms for Reduction

3.2.1 Greedy Selection

3.2.2 Clustering

3.2.3 Searching

3.2.4 Hybrids

3.3 Classifying the FAST algorithms

4 Preparation

4.1 Gathering Data

4.1.1 Selecting Projects

4.1.2 Generating Coverage Information

4.1.3 Collecting Fault Coverage Information

4.1.4 Combining Tests Suites

5 Method

6 Results

6.1 Test Suite Reduction

6.2 Fault Detection Loss

6.3 Running Time

6.4 Comparison to "Scalable Approaches to Test Suite Reduction"

7 Limitations

8 Future Work

9 Summary