

The Project Proposal

CS569

Chao Zhang

Prof Groce Alex

1. Introduction

The software testing is become more and more important right now. It is one of the most important technology to test the software and assure the quality of the software. Because of this, we already have many different ways to finish the test. Those testing approaches can generate the test case, by this way, they can detect the system failures. With those methods, random testing is a basic and wide used one because it is just pick the objects from the set of almost all possible inputs randomly, then the generate test cases and executing the software by using those test cases. But there are still people call it least effective method. So I think there are two problems with it, the first one is how to make it more accurate and efficient. The other one is how to generate a good random test. As we know that the random in the computer is not real random. If we give a huge amount of inputs which may include many meaningless test, this way can cover more possibilities but will lower the efficiency for sure. The most important thing for the testers is to find out some algorithms to generate the test case which can cover more branches and gives more effective set for the random test to minimize the test cases.

2. Proposal

For this project, I would like to try to implement the adaptive random test generation, the ART may can use fewer test cases to archive the test goal. It can make the random test more effective. I got this idea from the article *Adaptive Random Testing*, the author said that the adaptive random testing is based on the intuition. It can distribute the test cases evenly for the SUT (software under test). So the Adaptive Random Testing is a smarter algorithm which can choose better test cases internally to reduce meaningless testing input like the similar input to test the same branches or statement in different order. So this ART may be a good solution to solve the problems with the random test. In this paper, authors mainly considered to use Euclidean distance between the elements in executed set and candidate set. There also have another paper *Code Coverage of Adaptive Random Testing* said the ART can have higher coverage than the random testing in the same numbers of the test cases. Also, this paper shown that the ART gives higher effectiveness and confidence on the reliability for the SUT even without failure founded. So the TSTL can be better with the ART algorithm implemented. That is why I choose the ART for the project.

3. Plan

To implement the ART algorithm with the TSTL this term, I will reread those papers to make sure I understand what is the ART algorithm really is. Also I need a better understand of the random test with the TSTL. After that, I will use the avl.tstl and linklist.tstl as the tester to test my ART. If it worked, then, I will start working at the coverage. The final step is to compare the ART and the random test to see if the ART have a lower F-measure.

4. Reference

- [1] H. L. Tsong Yueh Chen, Fei-Ching Kuo. Code coverage of adaptive random testing, 2013.
- [2] I. M. T.Y. Chen, H. Leung. Adaptive random testing, 2004.