

**Addis Ababa Institute of Technology**

**Center of Information Technology and Scientific Computing**

**Department of IT/SW Eng.**

**Article Review on Model-based test prioritization heuristic**

**Methods and their evaluation**

Submitted to: Instructor Natnael

Team Members

1. Elbetel Gezahegn ATR/3445/08
2. Kimiya Mohammed ATR/2765/08
3. Lulit Mulugeta ATR/3806/08
4. Rahel Getachew ATR/0621/08

**DATE: 27/04/2018**

**ABSTRACT**

Software maintenance is an important and costly activity of the software development lifecycle. To ensure proper maintenance the software undergoes regression testing. It is very inefficient to re execute every test case in regression testing for small changes. Hence test case prioritization is a technique to schedule the test case in an order that maximizes some objective function. A variety of objective functions are applicable, one such function involves rate of fault detection – a measure of how quickly faults are detected within the testing process. This paper focuses on the regression testing for the test case prioritization. An attempt had been put up over here to explore the investigations carried out, the critics viewed and conclusions obtained from this document review.

**Keywords:**

Test prioritization, regression testing, model-based test prioritization, Heuristics.

1. **INTRODUCTION**

Nowadays software development is quality oriented development. Quality can be ensured by very good testing techniques. Regression testing is the process of testing a modified system using the old test suites. Developers need to make sure that modifications are correct and do not adversely affect the unchanged portion of the system. During regression testing the modified parts of the system are first tested. Then the whole system needs to be retested using the old test suite to have confidence that the modifications did not introduce new faults into the system.

This document explains about the research done on Model Based Test Prioritization heuristic methods and their evaluation. The first section deals with literature review. This explains about the overall summarized version of the research paper. The second section views on other related research review paper. This gives summarized description about the related research review paper. The third section is about comparison or critique on the paper we have dealt with. This section states the strong sides of the paper, different perspectives taken into the document format, problem identification, and quality of cited papers, proposed solution and evaluation. Last but not least section is about conclusion. This states what was done on the research paper and the outcome found on it put in a summarized and simple terms.

1. **LITERATURE REVIEW**

**Section 1**

In this paper regression test has been used to confirm that a recent program or code change has not adversely affected existing features. System retesting can be used to check whether the modifications have brought any new faults to the system. But this is very expensive as it requires huge time and resources. As a result test prioritization has been introduced. Test prioritization is a technique used to decide the priority of test cases based on some criterion and increases the chance of the early detection of faults. Out of several test prioritization methods this paper discussed model-based test prioritization method. It is a method which uses system models to prioritize tests.

**Section 2**

In this section of the paper, basic understanding about test prioritization has been given. The paper illustrates test prioritization with respect to early fault detection. A formula for computing rps (d); relative position of the first failed test in a test suite has been given. Rps (d) represents test suite fraction. Rate of fault detection is measured as a function of the percentage of faults detected in terms of the test suite fraction. The main goal of test prioritization is to improve the result of this function.

The simplest test prioritization method is random test prioritization which gives N number of possible test sequences for N size of test suite with randomly ordered test cases.

**Section 3**

In this section model-based test prioritization methods have been discussed in detail. The system models are composed of states and transitions between the states which are developed using EFSM. These models are executed for the whole test suites to collect information which are going to be used for prioritizing test cases. By identifying difference between the original and modified model model-based prioritization method detect faults in the modified system. The difference between these two models is a set of elementary model modifications, which is either transition addition or transition deletion. Depending on the information collected different prioritization methods can be used. In this paper two model-based test prioritization models were presented; selective test prioritization method and model dependence-based test prioritization method.

* **Selective Test Prioritization**: - assigns high priority to tests which execute modified transitions in the modified models. It assigns low prioritization to tests that don’t execute modified models. Tests with high priority executed before tests with low priority when the system retested. Both the high prioritized and low prioritized tests ordered randomly.
* **Model Dependence-Based Test Prioritization**: - using dependency analysis collect information to identify interactions between the added or deleted transition and the remaining model transitions to prioritize high priority tests. The dependencies can be either data dependency or control dependency. Executing the modified model by tests in the test suite identify unique interaction pattern between model transitions and added or deleted transitions. Interaction patterns are represented as model dependence sub-graphs. There are types of interaction patterns related to a modification; three of them were discussed in this paper:-
  + **Affecting Interaction pattern: -** identifymodel transitions that affect the added/deleted transition when the modified model executes on tests in the test suite. By starting from the added/deleted transition, traversing backward on the dependence sub-graph will result this pattern (we should remove edges which hadn’t traversed through).
  + **Affected interaction pattern: -** identifymodel transitions that are affected by the added/deleted transition when the modified model executes on tests in the test suite. By starting from the added/deleted transition, traversing forward on the dependence sub-graph will result this pattern (we should remove edges which hadn’t traversed through).
  + **Side-effect Interaction pattern: -** as a result of addition or removal of transitions, new dependency or existing dependency might be created or destroyed, respectively. These dependences are called side-effect interaction pattern.

For both methods the paper present algorithms to show what the implementation should look like.

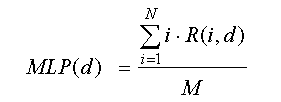
**Section 4**

Because of the methods discussed on the above section, especially the model dependence-based prioritization method, complexity and storing whole model trace to build interaction pattern makes it less effective. Therefore, in this section of the paper, additional model-based test prioritization methods have been introduce for more simple and efficient test prioritization.

* Heuristic #1:- higher priority is given to tests that test large amount of modified transitions than tests which test small amount of modified transitions. Since the higher prioritized tests execute many modified transitions the probability of revealing many faults is higher than the lower prioritized tests. All tests that have highest priority executed before others. And tests with no modified transition will execute at the end in a random sequence.
* Heuristic #2:- is modified version of Heuristic #1. It illustrates failed tests can be detected at the lower prioritized buckets too. So giving more chance to lower prioritized tests improves early fault detection. Starting from the highest prioritized tests to the lowest prioritized tests, it takes only one test at a time for execution and repeats this to the end. And tests with no modified transition will execute at the end in a random sequence.
* Heuristic #3:- higher priority is given to tests that have higher frequency of execution of modified transitions than tests that have lower frequency of execution of modified transitions. Since the higher prioritized tests execute more frequently the probability of revealing many faults is higher than the lower prioritized tests. All tests that have highest priority executed before others. And tests with no modified transition will execute at the end in a random sequence.
* Heuristic #4:- is modified version of Heuristic #3. It illustrates failed tests can be detected at the lower prioritized buckets too. So giving more chance to lower prioritized tests improves early fault detection. Starting from the highest prioritized tests to the lowest prioritized tests, it takes only one test at a time for execution and repeats this to the end. And tests with no modified transition will execute at the end in a random sequence.
* Heuristic #5:- higher priority is given to tests that execute transitions that have been executed the least number of times. So giving balanced execution for the tests is the main goal. And tests with no modified transition will execute at the end in a random sequence.

**Section 5**

In this section of the paper, a metric called most likely relative position (MLP) has been used to measure the effectiveness of early fault detection. For a given test suite using different test prioritization methods produce different test prioritization sequences and rate of fault detection. The MLP (d) of the first failed test that detects the fault d over all possible prioritizes test sequences is computed using this formula: -



* N: - size of test suite.
* i: - position of the first failed test.
* d: - detected fault.
* M: - number of all possible prioritized tests that may be generated by a given test prioritized method for a given test suite.
* R(i, d): - is number of prioritized test sequences that may be generated by a given test prioritization method for which the first failed test caused by fault is in the I position.

Some test prioritization methods use the analytical approach to determine the relative position of the first failed test by simply driving the above formula. For example random prioritization and selective test prioritization use this approach. But many test prioritization methods which determine the relative position of the first failed test experimentally; using the above formula is very expensive and difficult, as M might be very large. To solve these problems randomized approach estimation is used for the remaining test prioritization methods. The estimation used to generate prioritized test sequences according to the test prioritization method used. After a large number of test sequences are generated, the above formula will be used to determine the relative position of the first failed test.

**Section 6**

In this section of the paper, an experimental study which measures the effectiveness of early fault detection of test prioritization methods has been discussed. In the study five models with different size of transitions, states and lines of source codes were used.

The result from the experimental study indicates model-based prioritization and Heuristic #5 prioritization have exhibited the best effectiveness in this experimental study.

1. **Review of Research Papers Specific to the Domain**

**A SURVEY ON MODEL BASED TEST CASE**  
**PRIORITIZATION**

**Section I**

This related paper starts off by mentioning the need for test case prioritization and the different types of prioritization methods that exist. In relation with our paper, which is about model-based test prioritization, it describes how it is the best one compared to code-based test prioritization. The general purpose of this research is to find and examine the current techniques and issues in model-based test case prioritization.

**Section II**

In this section, the paper then continues describing the different prioritization methodologies, namely code-based test case prioritization, model-based test case prioritization and requirement-based test case prioritization. In model-based test case prioritization, study has shown that model-based test prioritization may improve the early fault detection as compared to code-based test prioritization because the execution of the model is very fast as compared to the execution of the actual system.

**Section III**

Here, the paper talks about advantages and disadvantages of different prioritization methods. For the model-based test prioritization, the experimental study presented in the paper was limited to two test prioritization heuristics only; and this is mentioned as a disadvantage. As an advantage, finding out the effectiveness of different techniques could be mentioned. To compare different test prioritization methods, the concept of the *most likely relative position*, *RP*(*d*), is used. The results from the experimental study indicate that some model-based test prioritization methods may improve on average the effectiveness of early fault detection as compared to random prioritization. The best performance is shown by the model-based test prioritization (IP) and Heuristic #3.

**Section IV**

In this section, the paper concludes its research by stating the metrics used for determining the effectiveness of prioritization techniques. It then talks about the future plan of including more numbers of criteria and applying new techniques to explore more effective prioritized test suite. It also presents the proposed frame work for the prioritization of test cases in a diagram.

On this paper titled "A SURVEY ON MODEL BASED TEST CASE PRIORITIZATION" could have been explained in a simpler, more detailed and understandable way. It could have used visual representation so that it could be easily understood.

1. **CRITQICS AND COMPARISON**

The paper mainly considers test prioritization with respect to early fault detection.

**Document format**

This paper is organized in a clear and understandable way.

It starts with a basic introduction which explains about regression testing, and also talks a little about test prioritization. In the next section .It explains about model-based testing including graphs as an example to visually explain things. Under it, it includes the topics Selective test prioritization and Model-Dependence Based Test Prioritization, Each topic is explained in precise way supported by algorithms.

It also mentions additional model-based test prioritization methods in the next section. In this section five Heuristics are mentioned. Each heuristic has its own characteristics, each explained with their own algorithms. The heuristics are explained briefly with flow and their relationships are explained and compared with examples.

The paper also finishes with a conclusion that mentions that the experiment was promising.

**Problem identification**

The basic problem identified in this paper is the idea of test case prioritization with respect to early fault detection.

**Proposed solution**

The problem identified in this paper is the problem of test case prioritization. Different ideas are mentioned as solutions to this problem throughout the paper. These ideas are explained in a brief way. In the 'Additional Model-Based Test Prioritization methods' section, each method has its own way of prioritizing test cases. Each method modifies the previous one or introduces new approaches. Each of these methods is supported by algorithms and graphs, which makes things more clear and understandable.

**Evaluation**

As mentioned in section 2, the goal of this paper is to increase the likelihood of revealing faults earlier during execution of prioritized test case. Each section in this paper revolves around this goal and tries to achieve it in its own ways. The result of this paper in general was promising.

Although the outcome of the experimental study was mostly a success, it also indicates that some types of information about models may not improve effectiveness of early fault detection. The experimental study in this paper was relatively small. If it had focused on larger models and systems, it would have given a better understanding of advantages and limitations of model-based test prioritization.

**Comparison**

The two papers titled "A SURVEY ON MODEL BASED TEST CASE PRIORITIZATION" and "MODEL BASED TEST PRIORITIZATION HEURISTIC METHODS AND THEIR EVALUATION" shows an experimental study on model-based testing.

Both papers start off by mentioning the need for test case prioritization and the different types of prioritization methods that exist, specifically model-based test prioritization.

Both papers mention the different models and explain them in their own ways.

Both talk about the future plan of including more numbers of criteria and applying new techniques or expanding the research.

The general purpose of the paper “A SURVEY ON MODEL BASED TEST CASE PRIORITIZATION" is to find and examine the current techniques and issues in model-based test case prioritization.

The paper "A SURVEY ON MODEL BASED TEST CASE PRIORITIZATION" compares model-based test prioritization with code-based test prioritization and other methods.

In the paper titled "A SURVEY ON MODEL BASED TEST CASE PRIORITIZATION" experimental study presented was limited to two test prioritization heuristics only. While the second paper talks about model-based testing in a broad and detailed way, mentioning a number of heuristics each supported with their own algorithms.

The paper titled "MODEL BASED TEST PRIORITIZATION HEURISTIC METHODS AND THEIR EVALUATION" has detail explanation and includes visual representation as well as algorithms compared to the paper titled "A SURVEY ON MODEL BASED TEST CASE PRIORITIZATION”.

1. **CONCLUSION AND RECOMMENDATION**

We have reviewed the paper “Model-based test prioritization heuristic methods and their evaluation”. We have carefully read the paper and reviewed each section and included the outcomes of our reviews. We also included a related paper and summarized it in a simple and understandable way. In this paper we have criticized both the actual and related paper. We also put comparison between them.

1. **REFERENCES**

[1] A Systematic Literature Review of the Test Case Prioritization Technique for Sequence of Events by Johanna Ahmad and Salmi Baharom

[2] A SURVEY ON MODEL BASED TEST CASE PRIORITIZATION by Sanjukta Mohanty, Arup Abhinna Acharya and, Durga Prasad Mohapatra

[3] Literature Review of Model Based Test case Prioritization by Shweta A. Joshi 1, Prof. B.S. Tiple.