ISSN:2321-1156
International Journal of Innovative Research in Technology & Science(IJIRTS)

TEST CASE PRIORITIZATION TECHNIQUES ON REGRESSION TESTING

M. Thillaikarasi , Assistant Professor; K. Seetharaman, Associate Professor, Annamalai University

Abstract

Test case prioritization techniques involve scheduling over test cases in an order that improves the performance of regression testing. It is inefficient to re executing every test case for every plan function if once change occurs. Test case prioritization techniques to organize the test instances in a test suite by ordering such that the most beneficial are executed first thus allowing for an increase in the effectiveness of testing. One of the performance goals i.e. The fault detection rate, is a criterion of how quickly faults are detected during the testing procedure. In this paper I introduce a new test case prioritization algorithm, which calculates average faults found per minute. I show the results illustrating the strength of algorithm with the help of APFD metric. The primary purpose of my paper is to find out the effectiveness of prioritized and non-prioritized case with the help of APFD.

Introduction

Regression testing is the re-execution of some subset of test that has already been taken. In regression testing as integration testing proceeds, number of regression tests increases and it is impractical and inefficient to re execute every test for every program function if one time change occurs. It is an expensive testing process used to detect regression faults. Regression test suites are a great deal simply try out that software engineers have previously acquired, and that have been preserved so that they can be used afterward to perform regression testing [1,2, 3]. So regression testing can be determined as follows:
Let P be a program and P' be a modified version of P and T be a test suite developed for P. Regression testing is concerned with validating P'. Regression test selection techniques attempt to cut the price of regression testing by selecting and carrying just a subset of the test instances in an existing test suite. In the previous work an Average Percentage of Faults Detected (APFD) metric [1] was used to
Determine the strength of the new test case orderings, but it's considered faults and test examples cost to be uniform.. .

Problem Definition

Rothermel at el. [2, 7] defines the test case prioritization problem as follows:

Given: T, a test suite; PT, the set of permutations of T; f, a function from PT to the real numbers.
Problem: Find T' belongs to PT such that (for all T") (T" goes to PT) (T" ≠ T') [f (T') ≥ f (T")].
Here, PT represents the set of all possible prioritizations (orderings) of T and f is a function that, applied to any such ordering, yields an award value for that ordering [2,7]. The aim of this research is to produce a test case prioritization technique that prioritizes test cases on the basis of detection of fault rate.

Methodology

This part provides the methodologies that are linked to regression testing. At that place are four methodologies that are available for regression testing. These methods are [2,5, 8]

a) Retest –all.
In this technique the test cases that no longer apply to modified version of program are discarded and all the remaining lot of trial examples is applied to test the modified plan..

b) Regression test selection.
Retest all technique takes time and endeavor as all trial events are applied to prove the program again, so many beside being quite expensive. This technique much better as it uses information about the program, modified course of study, test cases to select a subset of trial examples for testing.

c)Test suite Reduction.
This technique uses information about the plan and test suite to transfer the test examples, which have become redundant with time, as new functionality is added. It is different from Regression test e extract as the former does not permanently remove test cases, but selects those that are needed. The vantage of this technique is that it brings down cost of validating, executing, managing test suites over future releases of the software, but the downside of this is that it might melt off the error detection Capacity with the reduction of test suite size.

d) Test Case Prioritization.
In this technique each test cases are assigned precedence. The precedence is set according to some standard and test cases with highest priority are scheduled first For example criterion may be that the test case which has faster code cov-

erage gets the highest precedence. Advantage to previous techniques is that it doesn't throw away or permanently take out the trial cases from test suite. Another criterion was money may be rate at which fault is detected.

Determining test suite effectiveness

The operation of the prioritization technique used in this report, it is necessary to appraise strength of the grading of the test rooms. Strength will be appraised by the rate of defects found. The following metric is used to calculate the level of effectiveness.

A) Average percentage of faults detected (APFD) metric

To quantify the goal of increasing a subset of the test suite's rate of fault detection, i use a metric called APFD developed by Elbaum et al. [1,2,4] that measures the average rate of fault detection per percentage of test suite execution. The APFD is calculated by taking the weighted average of the number of faults detected during the run of the test suite. APFD can be calculated using a notation:
Let T -> The test suite under evaluation
m -> the number of faults contained in the program under test P
n -> The total number of test cases and
TFi -> The position of the first test in T that exposes fault i.

$$APFD = 1 - \frac{TF_1 + TF_2 + \ldots\ldots + TF_m}{nm} + \frac{1}{2n}$$

So as the formula for APFD shows that calculating APFD is only possible when prior knowledge of faults is available. APFD calculations therefore are only used for evaluation.

Proposed work

A new prioritization technique

A) Introduction

Earlier work [1,2,4] may take long time (may be month or year) depending on the size of the test suite and how long each test case takes be run. However, through the use of an effective prioritization technique, testers can re order the test cases to obtain an increased rate of fault detection. The technique presented in this paper implemented a new regression test suite prioritization algorithm that prioritizes the test cases with the goal of maximizing the number of faults that are likely to be found during the constrained execution.

B) The algorithm

Input: Test suite T, number of faults detected by a test case f, and cost to run each test case Tcost.
Output: Prioritized Test suite T'.

1: **begin**
2: set T' empty
3: **for each** test case t ε T **do**
4: calculate average faults found per minute as f/Tcost
5: **end for**
6: sort T in descending order based on the on the value of each test case
7: let T' be T
8: **end**

With the assumption that the desired execution time to run the test cases is known in advance, one can trace the number of faults each test case find and in how much time it takes to find the faults. So using this information as input the algorithm prioritizes the test cases of particular test suite. The algorithm calculates the average number of faults found per minute by a test case and using this value sorts the tests cases in decreasing order of test suite.

C) Results

Below table shows the number of faults detected by a test case in the test suite and total time taken by each test case.

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
F1	*					*				
F2				*			*	*	*	
F3		*			*	*				*
F4							*			
F5		*						*	*	
F6				*						
F7				*	*					
F8		*	*							
F9						*				
F10	*									*
No. of fault	2	3	1	3	2	3	2	2	2	2
Time	5	7	11	4	10	12	6	15	8	9

APFD result from proposed algorithm:

VTi=fault/time(rate of fault detection)

The calculations are:

VT1=2/5=0.4 VT2=3/7=0.42,
VT3=1/11=0.09 VT4=3/4=0.75
VT5=2/10=0.2 VT6=3/12=0.25
VT7=2/6=0.33 VT8=2/15=0.133
VT9=2/8=0.25 VT10=2/9=0.22
Priority set according to decreasing order of value of VTi, since more the rate of fault detection more will be the priority.Hence the prioritized order is:

T4, T2, T1, T7, T6, T9, T10, T5, T8, T3

In the above table

m=no. of faults = 10
n=no. of test cases = 10
So putting the values of m , n ,TFi(The position of the first test in T that exposes fault i) in the equation:

$$APFD = 1 - \frac{TF_1 + TF_2 + \ldots\ldots + TF_m}{nm} + \frac{1}{2n}$$

Putting values:

$$APFD = 1 - \frac{3+1+2+4+2+1+1+2+5+3}{10*10} + \frac{1}{20}$$

$$= \mathbf{0.81}$$

$$APFD = 1 - \frac{1+4+2+7+2+4+5+3+6+1}{10*10} + \frac{1}{20}$$

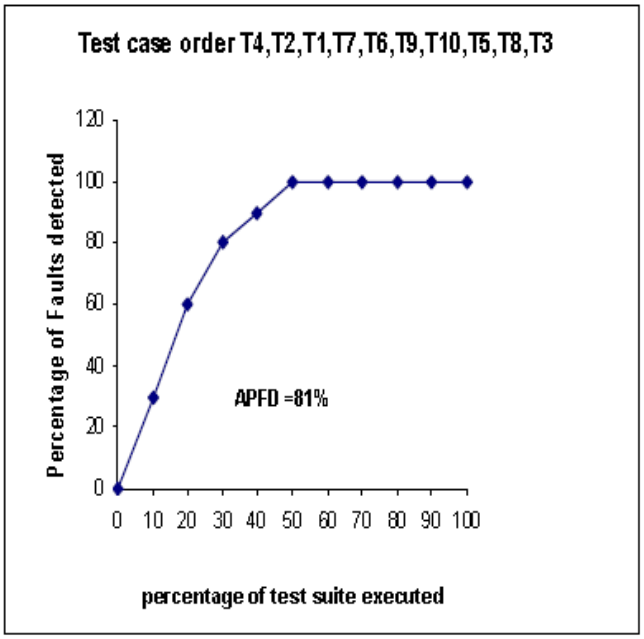
$$= \mathbf{0.70}$$

Analysis of APFD

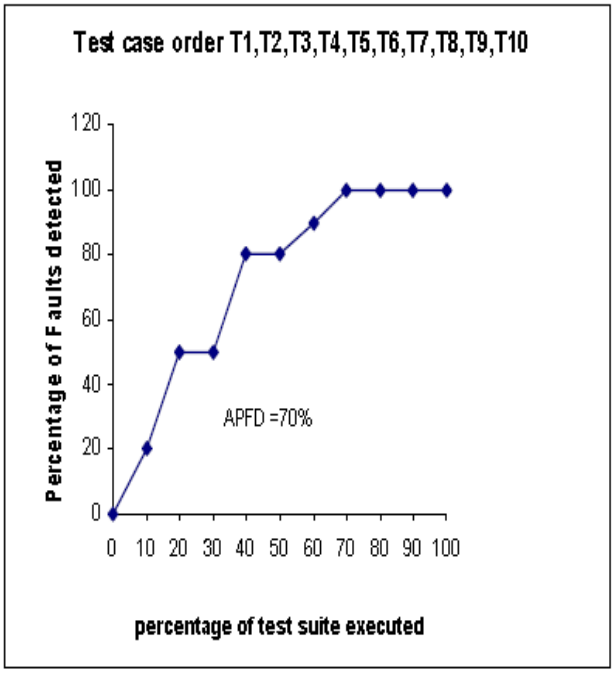
The comparison is drawn between prioritized and non-prioritized case, which shows that value obtained for prioritized case (new approach) is more than previous method, hence more effective of prioritized case

Below two graphs showing the Results forprioritized and non-prioritized ca

APFD graph for prioritized test suite



APFD graph for non-prioritized test suite



Conclusion

This paper proposed an algorithm for test case prioritization in order to improve regression testing. Analysis is done for prioritized and nonprioritized cases with the help of APFD (average percentage fault detection) metric. Graphs prove that prioritized case is more effective. In future we will try on test case prioritization over requirement analysis using APFD and risk metrics.

Refrences

[1] Alexey G. Malishevsky, Joseph R. Ruthruff, Gregg Rothermel, Sebastian Elbaum, Costcognizant Test Case Prioritization, 2006

[2] S. Elbaum, A. Malishevsky, and G.Rothermel Test case prioritization: A family of empirical studies. IEEE Transactions on Software Engineering, February 2002.

[3] Roger S. Pressman, Software engineering a practitioner's approach 6/e, 2005

[4] Sebastian Elbaum, Gregg Rothermel, Satya Kanduri, Alexey G. Malishevsky, Selecting a Cost-Effective Test Case Prioritization Technique, April 20, 2004

[5] Aditya P.Mathur, Foundation of software testing, Pearson Education 1st edition.

[6] Maruan Khoury, Cost-Effective Regression Testing, 2006

[7] G. Rothermel, R.H. Untch, C. Chu, and M.J. Harrold, "Prioritizing Test Cases for Regression Testing," IEEE Trans. Software Eng., vol. 27, no. 10, pp. 929-948, Oct. 2001.

[8] Maruan Khoury, Cost-Effective Regression Testing, 2006