**Contemporary Engineering Sciences, Vol. 7, 2014, no. 14, 677 - 682**
**HIKARI Ltd, www.m-hikari.com**
**http://dx.doi.org/10.12988/ces.2014.4674**

# Improving Spectrum-based Fault-localization through Spectra Cloning for Fail Test Cases Beyond Balanced Test Suite

**Patrick Daniel, Kwan Yong Sim**

[1,2]Faculty of Engineering, Computing and Science
Swinburne University of Technology, Sarawak Campus
Kuching, Sarawak, Malaysia

**Soonuk Seol**

[3]School of Electrical, Electronics and Communication Engineering
Korea University of Technology and Education (KOREATECH)
Cheonan, Chungnam, Rep. of Korea



## Abstract

Spectrum-based Fault Localization (SBFL) has been widely studied as a debugging technique to reduce time and effort in locating faulty code in software. In SBFL, execution profiles (spectra) of pass and fail test cases are analyzed with SBFL metric to rank software code according to their likeliness to be faulty. However, there are significantly more pass test cases than fail test cases in typical test suites for faulty programs. The domination of pass test cases creates imbalance test suites that have a negative impact on the performance of SBFL. This is attributed to the fact that the execution profiles of fail test cases provide the essential information on the location of faulty code. In this paper, we propose to clone the spectra of fail test cases beyond balanced test suite to improve the performance of SBFL. Our empirical study shows that the proposed cloning method significantly improves the performance of commonly used SBFL metrics. Furthermore, we attempt to identify the amount of cloning required to achieve the optimal performance for the proposed cloning method.

**Keywords**: Software Engineering, Software Testing, Software Debugging

## 1 Introduction

In recent years, Spectrum-based Fault Localization (SBFL) has emerged as a promising technique for fault localization. SBFL utilizes the code execution profile of test case, which is commonly known as “Spectrum”. The spectrum is a record of the lines of code in a program which have been executed (or not executed) by a test case. A collection of spectrum, or “Spectra” (plural form of spectrum), can be sourced from the testing process. These spectra are analyzed in the spectrum-based fault localization process based on the intuition that lines of code that have been executed by more fail test cases and less pass test cases are more likely to be faulty, and vice-versa.

In SBFL, four common coefficients are computed for each line of code in a program [1]. These coefficients are aef, anf, aep, and anp. Firstly, aef represents the number of fail test cases that have executed a particular line of code. Secondly, anf represents the number of fail test cases that have not executed a particular line of code. On the other hand, the third coefficient, aep, represents the number of pass test cases that have executed a particular line of code, whereas the forth coefficient, anp, represents the number of pass test cases that have not executed a particular line of code. Intuitively, a faulty line of code will have high values for aef and anp and low values for anf and aep.

Many ranking metrics [2][3][4] have been proposed for SBFL based on one or more of these four coefficients to calculate a score that indicate the likeliness of a particular line of code to be faulty. The higher the score, the more likely the line of code is faulty, and vice versa. These scores will be used to rank the lines of codes in the software according to their likeliness to be faulty. During the faulty localization process, software developer will inspect the highest ranked lines of codes first with the aim save the time and efforts required to locate the faulty line of code.

In a recent study on the effect of imbalance test suites on SBFL performance [5], it was observed that the test suites for many faulty programs have significantly more pass test cases than fail test cases. These include faulty programs in Siemen Test Suites which are commonly used in SBFL studies. However, the same study also found that the domination of pass test cases have a negative impact on the performance of SBFL. This is because the execution profiles of fail test cases provide the essential information on the location of faulty code. This finding was further supported in follow up studies [6] and [7]. In response to this, a strategy to clone the spectra of fail test cases in order to construct balanced test suites (test suite with approximately equal number of pass and fail test cases) has been proposed in [7].

In an unrelated study on debugging with SBFL in extreme scenarios [8], it

has been found that certain SBFL metrics could perform better in an extreme scenario where there are many failed test cases and only one or no pass test case. Inspired by this finding, we propose cloning of spectra for fail test cases beyond the balanced test suites proposed in [7] to further improve the performance of SBFL. In this paper, we also attempt to answer the following research questions:

1. Can cloning the spectra of fail test cases beyond balanced test suite further improve the performance of SBFL metrics beyond the performance for a balanced test suite?
2. If the answer is yes, what is the amount of spectra cloning for fail test cases required to achieve the optimal performance?

The remaining of the paper is structured as below: Section 2 presents the background of SBFL metrics as well as the software artifacts used as faulty programs for empirical studies in this paper. Section 3 outlines the experiment procedure while the results of the experiments are presented in Section 4. Section 5 discusses the findings and Section 6 concludes the paper and discusses the future work planned.

## 2 Empirical Studies

In this paper, empirical studies to evaluate the performance of three commonly used SBFL metrics listed in Table 1. Experiments are conducted on faulty versions of seven published programs from the well-known Siemens Test Suite. Siemens Test Suite is commonly used as the source of faulty program artifacts in fault localization literatures [1][5][6][7][8][9]. These programs are *print_tokens*, *print_token2*, *replace*, *schedule*, *schedule2*, *tcas* and *tot_info*. Excluding the unusable versions, a total of 62 faulty versions of these programs have been used in the experiments

**Table 1. SBFL metrics.**

| | | |
|---|---|---|
| Jaccard [2] = $\frac{aef}{aef+anf+aep}$ | Euclid [3] = $\sqrt{aef+anp}$ | Ochiai [4] = $\frac{aef}{\sqrt{(aef+anf)(aef+aep)}}$ |

To evaluate the performance of each SBFL metric when the spectra of fail test cases are being cloned, experiments are conducted using the following procedure:

1. Executed all test cases on each faulty version of the programs in Siemen Test Suite to obtain the spectra from all pass and fail test cases.
2. Clone (duplicate) the whole set of spectra for fail test cases for a set number of times.
3. Calculate the SBFL metric scores for each line of code to obtain the percentage of code inspected (*pci*, %) to locate the faulty line of code and record the percentage of *pci* as the performance of the SBFL metric. Repeat this step for each SBFL metric to obtain the *pci* of each SBFL metric.

4. Repeat the Step 1 to 3 on all faulty versions to obtain the average *pci* for each SBFL metric.

The above experiment procedures are repeated by changing the number of times the spectra of fail test cases are cloned in Step 2 over a range of 2X to 2000X, which is far beyond the balanced test suite.

## 3 Results

The number of pass and fail test cases is different from one faulty program to another. Table 2 presents the average ratios of pass test cases to fail test cases for faulty versions of each program in Siemen Test Suite. It can be observed from the first raw, last column of the Table 2 that the overall average ratio of pass test cases to fail test cases is approximately 1:0.04 originally without cloning. Therefore, in order to achieve a balanced test suite, the spectra of fail test cases need to be cloned or duplicated for approximately 25 times on average.

The impact of spectra cloning for fail test cases on the performance of SBFL metrics studied are plotted in Figure 1 and summarized in Table 3. The performance of SBFL metric is recorded as the percentage of code inspected (*pci*) to locate the faulty line of code. The lower the *pci*, the better is the performance of an SBFL metric. From the results in Table 3, it can be observed that spectra cloning of fail test cases improve the performance of the three SBFL metrics studied. For SBFL metrics Jaccard, Euclid and Ochiai, their performances (*pci*) progressively improve to a lowest *pci* before deteriorating slightly (indicated in red in the last row of Table 3). This trend is also observable from the graphs in Figure 1 which plot the *pci* for the range of duplications or cloning for the spectra of fail test cases.

**Table 2. Average ratio of pass test cases to fail test cases.**

| Spectra Cloning for Fail Test Cases | Ratio of Pass Test Cases : Fail Test Cases | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | *printtokens* | *printtokens2* | *replace* | *schedule* | *schedule2* | *tcas* | *tot_info* | *OVERALL* |
| no cloning | 1 : 0.04 | 1 : 0.06 | 1 : 0.02 | 1 : 0.07 | 1 : 0.01 | 1 : 0.03 | 1 : 0.08 | 1 : 0.04 |
| 10X | 1 : 0.38 | 1 : 0.60 | 1 : 0.21 | 1 : 0.66 | 1 : 0.14 | 1 : 0.27 | 1 : 0.79 | 1 : 0.43 |
| 50X | 1 : 1.88 | 1 : 3.02 | 1 : 1.03 | 1 : 3.28 | 1 : 0.68 | 1 : 1.33 | 1 : 3.96 | 1 : 2.17 |
| 100X | 1 : 3.77 | 1 : 6.04 | 1 : 2.07 | 1 : 6.57 | 1 : 1.37 | 1 : 2.67 | 1 : 7.92 | 1 : 4.34 |

**Table 3. The performances of SBFL metrics for spectra cloning of fail test cases.**

| Spectra Cloning for Fail Test Cases | Performance, *pci* (%) | | |
|---|---|---|---|
| | Jaccard | Euclid | Ochiai |
| no cloning | 8.87 | 17.39 | 6.91 |
| 5X | 8.07 | 15.48 | 6.46 |
| 10X | 7.15 | 13.82 | 6.27 |
| 50X | 6.08 | 9.09 | 5.79 |
| 100X | 5.69 | 7.20 | 5.51 |
| 500X | 4.98 | 5.04 | 4.98 |
| 1000X | 4.98 | 4.98 | 4.98 |
| 2000X | 5.08 | 5.05 | 5.08 |

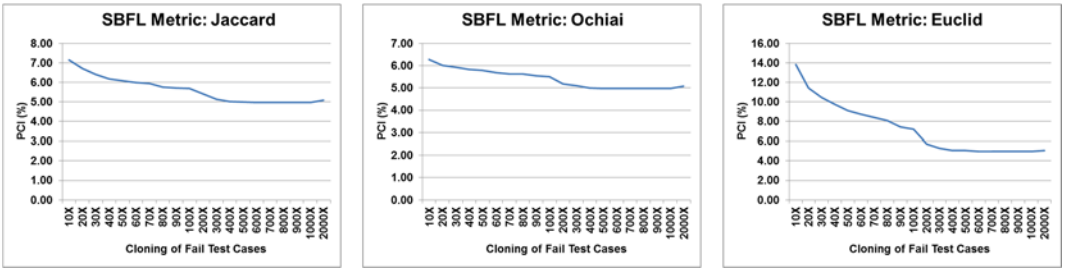


**Figure 1. The performances of SBFL metrics for cloning of the spectra of fail test cases.**

## 4 Discussion

Based on the empirical evidence in the previous section, we attempt to answer the research questions posted in Section 1.

*Research Question 1: Can cloning the spectra of fail test cases beyond balanced test suite further improve the performance of SBFL metrics beyond the performance of a balanced test suite?* From ratios of pass to fail test cases in Table 2, it could be observed that balance test suites (one pass test case to one fail test case) can be achieved by most programs by cloning the spectra failed test cases between 10X to 50X (except for *schedule2*, which require cloning between 50X to 100X). Therefore, we can safely consider cloning beyond 100X as beyond balanced test suite. From the experiment results in Table 3 and Figure 1, performance improvements (*pci* reduction) beyond 100X of cloning can be observed for Jaccard, Euclid and Ochiai. Therefore, we could conclude that cloning the spectra of fail test cases beyond balanced test suite can further improve the performance of these three SBFL metrics.

*Research Question 2: If the answer to Research Question 1 is yes, what is the amount of cloning required to achieve the optimal performance?* From Figure 1, the minimum amount of cloning required to achieve the optimal performance (lowest *pci*) is approximately 500X to 600X. This optimal performance remains when further cloning is done until 1000X. Beyond 1000X of cloning, the performances of these SBFL metrics start to deteriorate.

## 5 Conclusions and Future Work

In this paper, we have proposed spectra cloning for fail test cases beyond balanced test suite to further improve the performance of SBFL metrics. Empirical studies have been conducted on faulty programs in Siemen Test Suites to evaluate the performance improvements for three commonly studied SBFL metrics. Our experiment results have shown that spectra cloning of fail test cases beyond balanced test suites can further improve the performance of these SBFL metrics. We have further identified the amount of cloning required to achieve the optimal performance for the SBFL metrics.

For future work, we are currently extending the experiments in this paper to conduct a comprehensive study for over 30 SBFL metrics. On the other hand, we also plan to study the effect of cascading spectra cloning on noise reduction schemes for SBFL proposed in [9]. The combination of these two methods can potentially bring further improvements to the performance of SBFL metrics.

## References

[1] R. Abreu, P. Zoeteweij, and A. van Gemund, On the Accuracy of Spectrum-based Fault Localization. *Testing: Academic and Industrial Conference – Practice and Research Techniques* (2007), 89 – 98.

[2] P. Jaccard, Etude Comparative de la Distribution Florale Dans une Portion des Alpes et des Jura. Bull. Soc, Vaudoise Sci. Nat (1901), 547 – 579.

[3] E. Krause, Taxicab Geometry. Mathematics Teacher, (1973), 695 – 706.

[4] A. Ochiai, Zoogeographic Studies on the Soleoid Fishes found in Japan and its Neighbouring Regions. Bull. Jpn. Soc. Sci. Fish (1957), 526 – 530.

[5] C. Gong, Z. Zheng, W. Li and P. Hao, Effects of class imbalance in test suites: an empirical study of spectrum-based fault localization, *2012 IEEE 36th Annual Computer Software and Applications Conference Workshops* (2012), 470-475, IEEE.

[6] P. Rao, Z. Zheng, T.Y. Chen, N. Wang, K. Cai, Impacts of Test Suite's Class Imbalance on Spectrum-Based Fault Localization Techniques. *2013 13th International Conference on Quality Software* (QSIC) (2013), 260-267.

[7] Y. Gao, Z. Zhang, L. Zhang, C. Gong, and Z. Zheng, A Theoretical Study: The Impact of Cloning Failed Test Cases on the Effectiveness of Fault Localization. *2013 13th International Conference on Quality Software* (QSIC) (2013), 288-291.

[8] P. Daniel and K.Y. Sim, Debugging in the Extreme: Spectrum-based Fault Localization with Limited Test Cases, *International Journal of Software Engineering and Its Applications* (IJSEIA), **7(5)** (2013), 403–412.

[9] P. Daniel and K.Y. Sim, Noise Reduction for Spectum-based Fault Localization, *International Journal of Control and Automation* (IJCA), **6(5)** (2013), 117-126.

**Received: May 1, 2014**