ICIC Express Letters
Volume **10**, Number **12**, December **2016**

ICIC International ©2016 ISSN 1881-803X
pp. **2935–2942**

FAILED AND SUCCESSFUL EXECUTION SPECTRA-BASED SUSPICIOUSNESS METRICS FOR FAULT LOCALIZATION IN SOFTWARE SYSTEM

Wanchang Jiang[1,2], Jiadong Ren[1] and Yuan Huang[1]

[1]College of Information Science and Engineering
Yanshan University
No. 438, Hebei Avenue, Qinhuangdao 066004, P. R. China
jwchang84@163.com; jdren@ysu.edu.cn; 757918272@qq.com

[2]School of Information Engineering
Northeast Dianli University
No. 169, Changchun Road, Chuanying Dist., Jilin 132012, P. R. China

Received May 2016; accepted August 2016

Abstract. *Fault localization techniques are proposed based on the knowledge of software system to improve the reliability of software system. In this paper, based on the decisive factor of failed execution spectrum and the secondary factor of successful execution spectrum, two suspiciousness computation metrics FP3 and FP4 are proposed to compute the suspiciousness of each statement to be the fault. Metric FP3 with different weights for each part is also discussed. On the basis of our proposed suspiciousness metrics, a software fault localization algorithm is designed to apply the above proposed metrics to obtain statement ranking to assist effectively locating fault in software. Experiments are conducted on the program with test suites of different types and sizes in the Software-artifact Infrastructure Repository. The result verifies that our metrics are feasible and effective for fault localization, that our metrics improve the examination rate 12.8% on average over other methods, and that fewer statements need to be examined for fault localization. And the efficiency of locating fault of statement granularity is improved.*

Keywords: Suspiciousness metric, Failed execution spectrum, Successful execution spectrum, Metric-based fault localization

1. Introduction.

Software testing is the most important way of ensuring the reliability of software system, an artificial product, especially for software system in telecommunication, banking, and electricity industry. During the software testing, some information of testing such as execution trace for each test case is collected, and then fault can be diagnosed by using the information. However, the expenditure of running all test cases is high and the resource is limited in reality. Test case selection technique is designed to reduce the size of test suite for the software system, especially for the large ones [1]. In addition, test case prioritization technique is designed to improve the effectiveness of software testing [2].

Therefore, based on the knowledge of software, fault localization techniques of different granularity are proposed to improve the software reliability in the different phases of the software test lifecycle, such as integration testing and module testing. Methods are designed to measure the importance of functions in software network to improve software stability [3, 4]. To further improve the efficiency of locating the fault of statement granularity, program spectra-based suspiciousness metrics are designed to compute suspiciousness value of each statement to be the fault. For example, failed execution spectrum-based suspiciousness metrics of WONG1 [5], Ochiai [6] and Zoltar [7] are designed. Since only few test cases are failed ones, most statements are executed in successful executions but not in failed ones. And then the metric cannot work when this only decisive spectrum decreases to zero.

So, several types of suspiciousness metrics are designed by using failed execution spectrum and using some other spectra as the decisive factors in suspiciousness computation, such as metrics of WONG2 and WONG3 [5], and those of Sokal and HAN [8]. However, it is unreasonable to assume that decisive factors have the same effect on the suspiciousness. Therefore, the fault may have the low suspiciousness ranking.

Therefore, to improve the effectiveness and efficiency of fault localization, two new suspiciousness metrics FP3 and FP4 are proposed by using failed execution spectrum and successful execution spectrum as the decisive factor and the secondary factor respectively. Metric FP3 with different weights for each part is also discussed. Then, a suspiciousness metric-based fault localization algorithm is designed to use our proposed metrics FP3 and FP4 to compute suspiciousness of each statement to be the fault. Statements in the software are ranked based on the suspiciousness, and then the statement ranking is used to help the programmer locate fault. The fault localization algorithm with our metrics FP3 and FP4 can decrease statements that need to be examined until the fault is located for fault localization.

The organization of this paper is as follows. Section 2 discusses the preliminaries. Section 3 proposes two suspiciousness computation metrics. A suspiciousness metric-based fault localization algorithm is described in Section 4. Section 5 describes the experiments on our proposed suspiciousness metrics for fault localization. Finally, we conclude our work and give future work in Section 6.

2. **Preliminaries.**

In this section, concepts of fault program, test suite, execution trace spectra and program spectra are given.

Definition 2.1. *Let $\{S_1, \cdots, S_i, \cdots, S_N\}$ denote a fault program P_f which contains one fault or many faults that can be an error or bug, wherein the ith statement S_i can be a line of code or a block of code.*

Definition 2.2. *Let $\{T_1, \cdots, T_j, \cdots, T_M\}$ denote a test suite of test cases to be executed to test a fault program, where T_j is a test case. If the actual result of execution with T_j is different from the expected result, a failed execution occurs with a failed test case T_j, and the result r_j denotes as 0. Otherwise, a successful execution happens with a passed test case T_j. And the number of failed and passed test cases is denoted as N_f and N_p respectively.*

Definition 2.3. *Execution trace spectra are extracted from execution traces of program running with test cases, which can be denoted as a two-dimensional structure $\{e_{ij} | 1 \leq i \leq N, 1 \leq j \leq M\}$. The element e_{ij} indicates whether S_i is executed or not in the execution with T_j, which denotes 1 or 0.*

Definition 2.4. *Program spectra of $a_{np}(S_i)$, $a_{nf}(S_i)$, $a_{ep}(S_i)$ and $a_{ef}(S_i)$ are defined with $\{e_{ij}\}$ to collect statistical information about program running of S_i. The first subscript 'e' or 'n' indicates whether the statement is covered or not by one execution, and the second subscript 'p' or 'f' indicates whether the corresponding test case passed or failed. Failed execution spectrum $a_{ef}(S_i)$ and failed non-execution spectrum $a_{nf}(S_i)$ are respectively the number of failed executions in which S_i is executed or not. And successful execution spectrum $a_{ep}(S_i)$ and successful non-execution spectrum $a_{np}(S_i)$ are respectively the number of successful executions in which S_i is executed or not. To simplify description, program spectra are denoted as a_{ef}, a_{ep}, a_{nf} and a_{np} for short.*

3. Suspiciousness Metrics Based on a_{ef} and a_{ep}. With failed execution spectrum a_{ef} as the decisive factor and successful execution spectrum a_{ep} as the secondary factor, two new suspiciousness computation metrics FP3 and FP4 are proposed to obtain suspiciousness of each statement to be the fault.

A main structure in a certain form is important to determine the performance of suspiciousness metric for fault localization. It is proved that the fault statement would be covered by more failed executions and less successful ones in comparison to other statements. Thus, a_{ef} is considered as the decisive factor and a_{ep} as the secondary factor. If one statement is covered only by failed executions without any successful one, this statement is considered the most likely to contain the fault. Therefore, a_{ef} is positively correlated and a_{ep} is inversely correlated with the suspiciousness. Based on a_{ef} and a_{ep}, the construction of a_{ep}-based fractional expression is emphasized with some spectra a_{ef}, a_{ep}, a_{nf} and a_{np} to balance the influence between a_{ef} and a_{ep} on the suspiciousness.

Therefore, a new suspiciousness computation metric FP3 is proposed based on the decisive factor a_{ef} and the secondary factor a_{ep}. As the decisive factor, a_{ef} is included in the formula directly, which is positively correlated with the suspiciousness. The secondary factor a_{ep} is expressed as a fractional expression. The numerator of this a_{ep}-based expression is a_{ep}. The sum of a_{ef} and a_{ep}, inversely correlated parameters of a_{ep}, is included into denominator to reduce the influence of a_{ep} on suspiciousness. In addition, the numerator a_{ep} is included in the denominator to further reduce the influence of a_{ep} on suspiciousness. And subtraction operation is used to express that this a_{ep}-based expression is inversely correlated with the suspiciousness. The suspiciousness of statement S_i to be the fault can be computed by the metric FP3, which denotes $SUS_{FP3}(S_i)$ shown in the following formula.

$$SUS_{FP3}(S_i) = a_{ef}(S_i) - \frac{a_{ep}(S_i)}{a_{ef}(S_i) + a_{np}(S_i) + a_{ep}(S_i)} \tag{1}$$

where 'F' denotes a_{ef} itself, 'P' denotes the a_{ep}-based expression, and '3' is the number of spectra in the denominator of the a_{ep}-based expression.

In addition, metric FP3 with different weights for each part is respectively discussed. Parameter α is introduced to increase the weight of a_{ef} in the metric, wherein α>1. Then suspiciousness $SUS_{FP_\alpha 3}(S_i)$ of statement S_i can be computed.

$$SUS_{FP_\alpha 3}(S_i) = \alpha \cdot a_{ef}(S_i) - \frac{a_{ep}(S_i)}{a_{ef}(S_i) + a_{np}(S_i) + a_{ep}(S_i)} \tag{2}$$

Similarly, to decrease the influence of the a_{ep}-based fraction on suspiciousness, parameter β (β<1) is used. Then suspiciousness of S_i can be computed as $SUS_{FP_\beta 3}(S_i)$.

$$SUS_{FP_\beta 3}(S_i) = a_{ef}(S_i) - \frac{\beta \cdot a_{ep}(S_i)}{a_{ef}(S_i) + a_{np}(S_i) + a_{ep}(S_i)} \tag{3}$$

As for the values obtained by above three formulas, the suspiciousness is monotonically increasing with the decisive factor a_{ef} and decreasing with secondary factor a_{ep}. For the sum of a_{ep} and a_{np} is N_p, a_{np} decreases with the increase of a_{ep}. The suspiciousness value is monotonically decreasing with a_{np}. Since the numerator a_{ep} is included in the denominator of the a_{ep}-based expression, the upper limit of the expression is 1. The a_{ep}-based expression equals the limit when the sum of a_{ef} and a_{np} is 0, namely both a_{np} and a_{ef} equal 0. When a_{ef} is nonzero, the suspiciousness mainly depends on a_{ef}. When a_{ef} equals N_f and a_{ep} equals 0, the metric has the maximal value. Otherwise, when a_{ef} is 0, only a_{ep} plays the decisive role in computing suspiciousness, and the metric has the minimal value when a_{ep} equals N_p.

Based on the analysis above, even with α or β, the function monotony of each parameter of a_{ep}, a_{ef} and a_{np} and the suspiciousness value is not changed. And metrics with α or β

have the same parameter value when having the same minimal or maximal values. As a result, all above three formulas have the same performance.

When statement is not executed in failed execution with failed tests, the possibility of statement to be the fault is increased in this situation. Therefore, a new suspiciousness metric FP4 is proposed based on a_{ef} and a_{ep}, whose a_{ep}-based fraction is different from that of FP3. Failed non-execution spectrum a_{nf} is included in the denominator of a_{ep}-based fraction, which gives much less weight of a_{ep} to the result of suspiciousness. Using the metric FP4, the suspiciousness $SUS_{FP4}(S_i)$ of S_i can be computed as follows.

$$SUS_{FP4}(S_i) = a_{ef}(S_i) - \frac{a_{ep}(S_i)}{a_{ef}(S_i) + a_{np}(S_i) + a_{nf}(S_i) + a_{ep}(S_i)} \tag{4}$$

FP4 has the same function monotony of a_{ef}, a_{np} and a_{ep} as EP3, which have the same relationship with the suspiciousness value. When a_{ef} equals N_f and a_{ep} equals 0, the metric has the same maximal value. When a_{ef} is 0 and a_{ep} equals N_p, the metric has the same minimal value.

4. Suspiciousness Metric-Based Fault Localization Algorithm.

A suspiciousness metric-based fault localization algorithm is designed to illustrate the application of suspiciousness metrics FP3 and FP4 to assist locating the fault in a program.

For a given fault program, execution traces and results are recorded for test case running. Then, program spectra are extracted from the execution traces. Finally, the suspiciousness of statements can be computed by using the suspiciousness metrics. All statements are ranked according to the suspiciousness from high to low. The fault localization algorithm is presented as follows.

```
Algorithm: Suspiciousness metric-based fault localization algorithm
Input: fault program $P_f$, test suite $\{T_j\}$
Output: sequence $\{S_{i_1}, S_{i_2} \cdots S_{i_N}\}$ for each metric
For each test case $T_j$ in $\{T_j\}$
    Execute $P_f$ with test case $T_j$
    Gather execution trace
    Compare actual result with expected result, and output $r_j$
End For
Output statements $\{S_1, \cdots, S_i, \cdots, S_N\}$
Collect $N_f$ and $N_p$ of test cases with $r_j$
Extract execution trace spectra $\{e_{ij}\}$ from execution traces
For each statement $S_i$
    Compute program spectra $a_{np}(S_i)$, $a_{nf}(S_i)$, $a_{ep}(S_i)$ and $a_{ef}(S_i)$ by using $\{e_{ij}\}$,
    $N_f$ and $N_p$
    Compute suspiciousness $SUS_{FP3}(S_i)$ of statement $S_i$ by metric FP3
    Compute suspiciousness $SUS_{FP4}(S_i)$ of statement $S_i$ by metric FP4
End For
Rank statements based on $SUS_{FP3}(S_i)$
Output sequence $\{S_{i_1}, S_{i_2} \cdots S_{i_N}\}$ for metric of FP3
Rank statements based on $SUS_{FP4}(S_i)$
Output sequence $\{S_{i_1}, S_{i_2} \cdots S_{i_N}\}$ for metric of FP4
```

As a result, a sequence of statements $\{S_{i_1}, S_{i_2} \cdots S_{i_N}\}$ is obtained by the metric-based algorithm, where i_k is the statement number of S_{i_k} with the kth suspiciousness value and N is the number of statements to be inspected. However, several statements may have the same suspiciousness. In this case, with Formula (5), the medium number $Rank_{FP3}(S_i)$

will be computed as the ranking of these statements. And the bottom integral function is used to compute the ranking when non-integer result is obtained.

$$Rank_{FP3}(S_i) = \left\lfloor \frac{\{j|SUS_{FP3}(S_j)<SUS_{FP3}(S_i)\} + (N - \{k|SUS_{FP3}(S_k)>SUS_{FP3}(S_i)\} + 1)}{2} \right\rfloor \quad (5)$$

The programmer examines statements according to the sequence starting from top-rank statements one by one until the fault is determined. To evaluate the performance of metric for fault localization, the ratio of inspected statements is computed as examination rate. And the examination rate $E\_Rate_{FP3}(P_f)$ of the fault program P_f with metric FP3 is given as follows, where S_{fault} is the fault statement.

$$E\_Rate_{FP3}(P_f) = \frac{SUS_{FP3}(S_{fault})}{N} \quad (6)$$

5. **Experiment.** Using the Software-artifact Infrastructure Repository (SIR) [6], experiments are conducted under Fedora Core System. The suspiciousness metric-based fault localization algorithm is realized by Java programming language, and the performance of our proposed metrics for fault localization is compared with that of five previous metrics, that is, a_{ef}-based metrics of Ochiai (OC) and Zoltar (ZOL), and spectra-based metrics of Sokal (SOK), WONG2 and WONG3.

5.1. **Experiment setup.** Program "tcas" in SIR is provided with seeded faults, which has 41 fault versions of different kinds. 35 versions are used to investigate how well our metrics perform, which include the version containing one fault and the version with executable fault statement. In addition, for the fault of the macro definition or array definition, the version can be used when the suspiciousness of statement firstly using the macro can be computed. With thousands of test cases, test suite of "Universe" type is not suitable for application in reality. To verify the effectiveness and stability of suspiciousness metric-based fault localization, test suites of four types "bigrand", "bigcov", "cov" and "cov-extended" in SIR are utilized. Test suite of "bigcov" type is generated to achieve branch coverage. Test suite of "cov" is generated to achieve branch coverage in the minimal fashion, and the size is about 10% of that of "bigcov". Test suite of "cov-extended" is about half that of "bigcov". With the same size of "bigcov", test suite of "bigrand" is generated randomly. Then, four groups of experiments are conducted by using four test suites of each type, with only one type used in each group.

5.2. **Experiment results and analysis.** On the basis of statement ranking results with four test suites of "bigrand", the average examination rate of each fault version with each metric is obtained, which is shown in Figure 1.

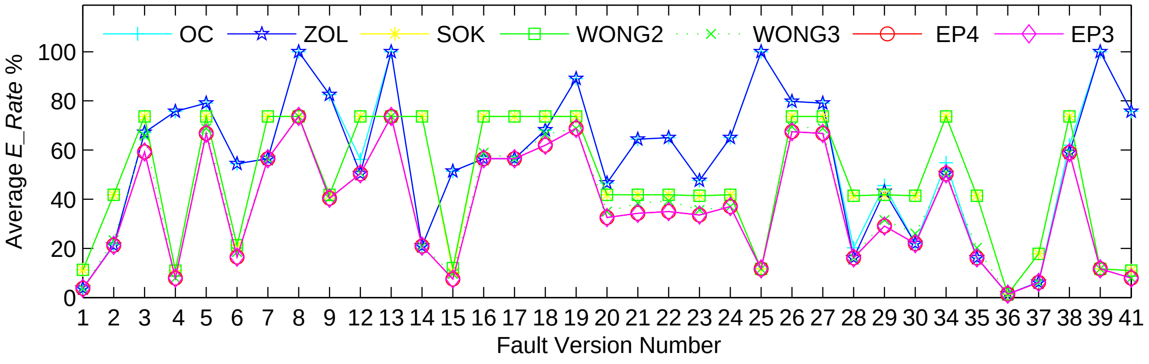

FIGURE 1. The average examination rate of "tcas" with "bigrand" suites

A small examination rate of the fault program with some metric is expected, because fewer statements need to be inspected to locate fault. The experiments show that the performance of FP3 with α>1 or β <1 is the same as that with $\alpha = 1$ and $\beta = 1$, which verifies the conclusion in Section 3. As a result, in this section, we only discuss FP3 with $\alpha = 1$ and $\beta = 1$. Our metrics outperform other metrics of OC, ZOL, SOK, WONG2 and WONG3 with "bigrand" suites, and gain an average decrease of 20.3%, 19.7%, 11.1%, 11.2% and 1.4% respectively. The performance of FP4 is the same as that of FP3, which is coincident with the analysis in Section 3. Metrics of OC and ZOL are completely ineffective for versions 8, 13, 25, 39 when a_{ef} is zero. In contrast, our metrics can even work well in this case. For example, with metric of EP3, the examination rate of version 25 is 11.6%.

With statement ranking result obtained by each test suite of "bigcov" type, the average examination rate of each version with each metric is shown in Figure 2. Our metrics decrease the average examination rate about 9.1% on average over the other metrics, and up to 13.9% in specific case. OC and ZOL are ineffective for some versions when failed execution spectrum is zero. Taking version 4 as an example, in comparison with OC, ZOL, SOK and WONG2, our metrics gain an average decrease of 11.7%, 11.7%, 10.3% and 4.5% respectively.

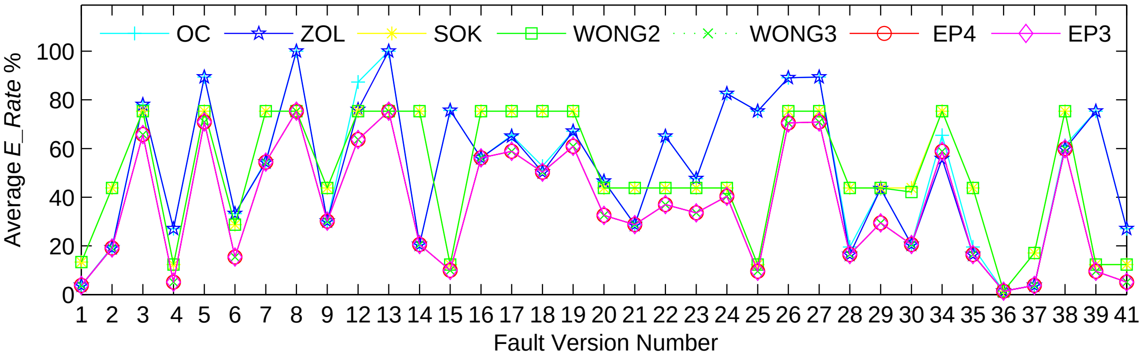

FIGURE 2. The average examination rate of "tcas" with "bigcov" suites

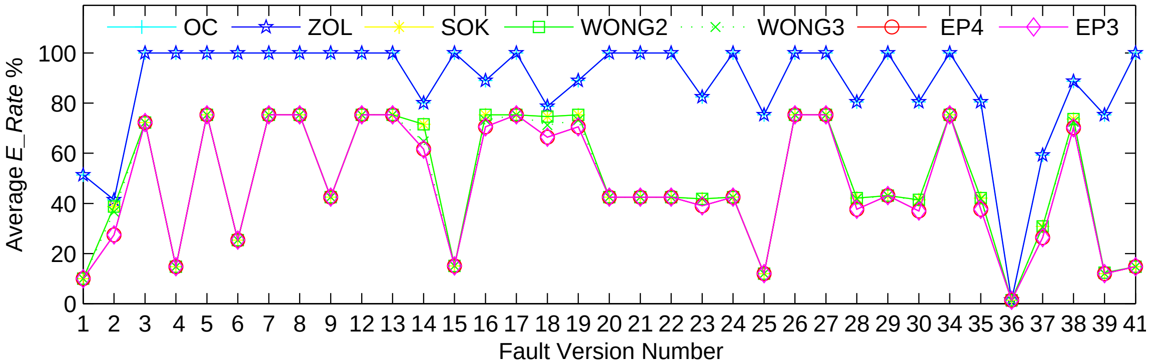

FIGURE 3. The average examination rate of "tcas" with "cov" suites

Based on suspiciousness ranking result obtained by four test suites of "cov", the average examination rate of each version with each metric is obtained, as shown in Figure 3.

With suites of "cov" type, metrics of OC and ZOL have the worst performance, which are ineffective for most versions, such as versions 3, 4, 5, 6 and so on. However, our metrics can even work well for these fault versions. Our metrics make the examination rate smaller than that of other metrics OC, ZOL, SOK, WONG2 and WONG3. Our metrics gain an average decrease of 39.9%, 39.9%, 1.8%, 1.8% and 1.3% respectively.

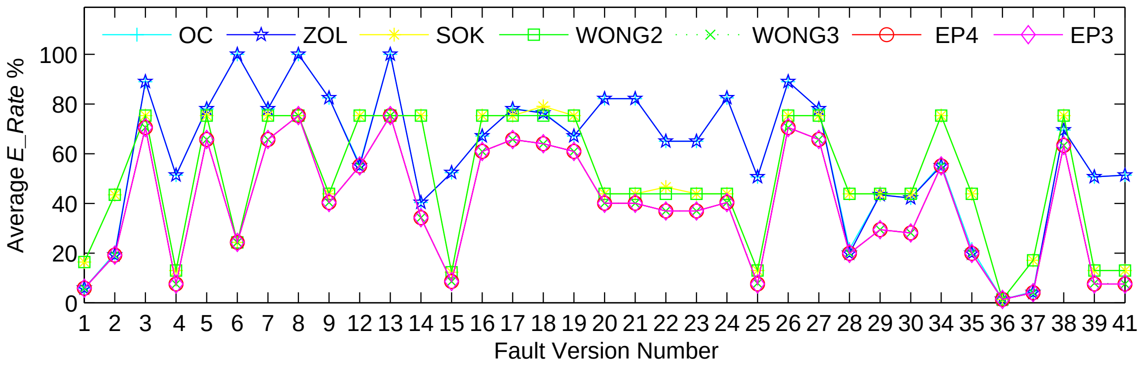

Figure 4. The average examination rate of “tcas” with “cov-extended” suites

Table 1. Standard deviation of examination rate of each metric

	EP3	EP4	OC	ZOL	SOK	WONG2	WONG3
bigrand	0.238	0.238	0.360	0.363	0.255	0.254	0.240
bigcov	0.252	0.252	0.371	0.372	0.253	0.253	0.252
cov	0.253	0.253	0.292	0.300	0.250	0.250	0.248
cov-extended	0.254	0.254	0.384	0.386	0.255	0.252	0.254

Finally, statement suspiciousness ranking results are computed with four test suites of “cov-extended”. Then, for each metric, the average examination rate of each version is obtained as shown in Figure 4.

As shown in Figure 4, our metrics perform well with “cov-extended” suites. In contrast, the performance of OC and ZOL is not stable. Our metrics outperform other metrics of OC, ZOL, SOK and WONG2, and gain an average decrease of 20.7%, 20.5%, 10.5% and 10.3% respectively. In addition, our metrics even have better performance than WONG3 for versions 25 and 39.

To compare the stability of each metric for fault localization with suites of different types, the standard deviation of examination rate of each metric on all fault versions is computed as shown in Table 1. The standard deviation of our metrics is smaller than that of other metrics with suites of different types, except for metrics of WONG with suites of “cov” and “cov-extended”. And it is proved that our metrics have the stable performance with suites of different types.

6. **Conclusions.** We propose two new suspiciousness metrics on the basic of a_{ef} and a_{ep}. a_{ep}-based fractional expression is designed to reflect the influence of the spectrum on the likelihood of each statement to be the fault. Then a suspiciousness metric-based fault localization algorithm is designed to apply our proposed metrics to obtain statement ranking for fault localization. Experiments show that the ineffectiveness of a_{ef}-based metrics is solved. Fault mostly has the smaller examination rate with test suites of different types by our metrics instead of other metrics, especially the type of “bigrand”. Our metrics are insensitive to the type of suite, and fewer statements need to be examined until the fault is located. It is possible to apply FP3 and FP4 to improving the effectiveness and efficiency of fault localization.

In the future work, more emphasis should be put on the localization of multiple types of faults, such as a fault in an assignment statement, a fault of missing partial code and a fault in a condition statement. It should be considered how to use the data dependence and control dependence information to improve suspiciousness metric-based fault localization method of localizing these types of faults.

Acknowledgment. This work was supported by the National Natural Science Foundation of China (No. F020512, F020204), the Natural Science Foundation of Hebei Province (No. F2014203152) and the Education Department of Jilin Province [2016]96.

REFERENCES

[1] Y. Wang, Z. Chen, Y. Feng, B. Luo and Y. Yang, Using weighted attributes to improve cluster test selection, *Proc. of the 6th IEEE Int'l Conf. on Software Security and Reliability*, Washington, D.C., USA, pp.44-58, 2012.

[2] C. Fang, Z. Chen, K. Wu and Z. Zhao, Similarity-based test case prioritization using ordered sequences of program entities, *Software Quality Journal*, vol.22, no.2, pp.335-361, 2014.

[3] H. He, J. Wang and J. Ren, Measuring the importance of functions in software execution network based on complex network, *International Journal of Innovative Computing, Information and Control*, vol.11, no.2, pp.719-731, 2015.

[4] G. Huang, X. Chen, H. Wu, P. Zhang and J. Ren, A novel approach to identify influential functions in complex software network based on complex network, *ICIC Express Letters*, vol.10, no.2, pp.485-492, 2016.

[5] W. Wong, Y. Qi, L. Zhao and K. Cai, Effective fault localization using code coverage, *Proc. of the 31st Annual International Computer Software and Applications Conference*, Beijing, China, pp.449-456, 2007.

[6] P. Daniel, K. Y. Sim and S. Seol, Improving spectrum-based fault-localization through spectra cloning for fail test cases, *Contemporary Engineering Sciences*, vol.7, no.14, pp.677-682, 2014.

[7] T. Janssen, R. Abreu and A. J. C. Van Gemund, Zoltar: A spectra-based fault localization tool, *Proc. of the 2009 ESEC/FSE Workshop on Software Integration and Evolution Runtime*, Amsterdam, The Netherlands, pp.23-29, 2009.

[8] L. Naish, H. J. Lee and K. Ramamohanarao, A model for spectra-based software diagnosis, *ACM Trans. Software Engineering and Methodology*, vol.20, no.3, pp.1-32, 2011.