2017 IEEE International Conference on Software Quality, Reliability and Security (Companion Volume)

An In-Depth Study of the Efficiency of Risk Evaluation Formulas for Multi-Fault Localization

Xiaolin Ju *† ‡ Xiang Chen *‡ Yibiao Yang † Shujuan Jiang ‡§ Junyan Qian ‡ Baowen Xu †
*School of Computer Science and Technology, Nantong University, China
†School of Computer Science and Technology, Nanjin University, China
‡Guanxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, China
§School of Computer Science and Technology, China University of Mining and Technology, China
{ju.xl, xchencs}@ntu.edu.cn, {yangyibiao, bwxu}@nju.edu.cn, shjjiang@cumt.edu.cn, junyanq@gmail.com

***Abstract*—A large amount of risk evaluation formulas have been proposed for spectrum-based fault localization (SBFL) in prior studies. A recent study by Xie et al. developed an innovative framework to theoretically analyze the effectiveness of those risk evaluation formulas in SBFL. Xie et al.'s study was based on the assumption that program has only one fault. In other words, they investigated SBFL in the context of single-fault SBFL. However, in practice, programs might have more than one faults. In this paper, we first propose a novel theoretical analysis framework for the risk evaluation formulas in the context of multi-faults SBFL. Our framework is based on a new effort cost evaluation $L\text{–}Score$. By applying our framework, we then conduct case studies to investigate the efficiency of existing formulas in the context of multi-faults SBFL.**

***Keywords*—multi-fault localization; risk evaluation formulas; efficiency analysis; testing; debugging**

I. INTRODUCTION

Spectrum based fault localization(SBFL) has received extensive attention among so many researchers for its simplicity and high efficiency [1]. This kind of technologies evaluated the risk of each program entity (e.g., statement, basic block, branch, predicate and function) with the coverage spectrum of program executions and ranked these entities by the risk values with a descending order. Then the developers inspected each entity to identify software errors during debugging [2]. Typical SBFL techniques include Tarantula [2], Ochiai [3], Naish [4], SAFL [5], and so on (e.g., [6]). Generally, examining score, the ratio of faults detected with examined codes, is used to evaluate the efficiency of a fault localization technology.

Commonly, researchers evaluated the efficiency of SBFL techniques with a series of benchmarks or some industry programs by comparing their examining scores. To carry out a fair comparison, it is of great concern to construct the same experimental conditions, such as the same subjects, the same seeded or real faults, the same test suites, etc. However, these fair design efforts cannot avoid a critical defect: the experimental results depend on the environment settings, such as fault types, the composition of test suite, even the composition of the program spectra, etc. A Recent study by Pearson et al. claimed that the formula has relatively little effect on how well a SBFL technique performs [7]. This weakness has prevented the generalization and application of these fault locating technologies in the industry.

Furthermore, costs of fault localization between two techniques differed minor in many empirical studies. For example, the experimental evaluation indicates that the costs of `Ochiai` and `Tarantula` difference are less than 2% of codes [3], but the scale of these subjects in their experiment is very small. Usually, 2% of codes means only several statements, which makes no actual effects in real program debugging.

All this raises the question that why the efficiency of different risk formulas always makes no significant difference in actual program debugging activities? Some researchers began to conduct theoretical analysis to pursue the answer. An early work by Lee et al. shows the equation between Tarantula and q_e[8]. They further inspected more risk evaluation formulas and proved some of them are equivalent. Later, Xie et al. proposed a simple analytical framework, with which, they analyzed 30 risk formulas, and found two groups (5 formulas) maximal formulas [1]. Using the theoretical analysis framework, they conducted a further study on 30 risk formulas, which were generated through Genetic Programming (GP) by Yoo et al., and found 4 maximal formulas out of all these compared ones [9], [10]. Recently, Naish et al. [11] proposed an improved SBFL approach which categorizes program statements into different classes by static program analysis and assigns them different weights based on the likelihood of being faulty statement, then evaluated the suspiciousness of each statements according to these weights for fault locating. Furthermore, W.K. Chan et al. extended Xie's framework with empirical results. They found that ER formulas are same on different code level of abstraction, and non-ER formulas can be more effective than ER formulas [12].

Illuminate by Xie's framework, we introduced a method for faults localization effort evaluation and proposed a new framework that can analyze the risk evaluation formula theoretically in the multi-fault locating scenario.

The main contributions of this paper can be summarized as:

- a new effort cost evaluation ($L\text{–}Score$) for the multi-fault locating scenario.
- a framework that can theoretical analysis the efficiency of risk evaluation formulas for multi-fault locating.
- case studies for the efficiency of existing formulas by applying our framework.

978-1-5386-2072-4/17 $31.00 © 2017 IEEE
DOI 10.1109/QRS-C.2017.58

II. Background

A. Spectrum based Fault Localization

In 2002, Jones et al. proposed a spectrum based fault localization approach (named *Tarantula*) based on the assumption that program executing spectrum could approximate fault causality. Since then, researchers have proposed many spectrum based fault localization approaches with different forms of formulas to evaluate the suspiciousness of each entities using its coverage of program executions [13], [6].

The information, collected for evaluating the suspiciousness of each entities, is usually shown as follows:

$A_i =< N^i_{CF}, N^i_{CS}, N^i_{UF}, N^i_{US} >$

where

- N^i_{CF} : The number of failed test cases that can cover the program entity e_i;
- N^i_{CS} : The number of passed test cases that can cover the program entity e_i;
- N^i_{UF} : The number of failed test cases that cannot cover the program entity e_i;
- N^i_{US} : The number of passed test cases that cannot cover the program entity e_i.

Many evaluation formulas were proposed for creating the rank list of all entities. The basic intuition behind these formulas lies in the following observations: The more passed test cases that executed a program entity, the less likely it is for the entity to be faulty; The more failed test cases that executed a program entity, the more likely it is for the entity to be faulty;

B. Metrics for the Evaluation of Fault Localization

Most fault localization approaches do not stop examining the code until one bug was identified. Intuitively, the percentage of code that needs to be examined can be represented in the efficiency of the fault localization technology.

The T–$Score$ evaluated the percentage of entities that need not be examined. On the contrary, the $EXAM$–$Score$, also named $expense$ score, evaluated the percentage of entities that need to be examined until the first bug is located[14].

$$EXAM = \frac{\textit{Number of entities examined}}{\textit{Total number of entities}} \times 100\% \qquad (1)$$

C. Theoretical Analysis Framework for Locating Single Fault

In [1], Xie et al. proposed a theoretical analysis framework to compare the effectiveness of a group of risk evaluation formulas. Next, we gave a brief discussion of their pioneering work.

DEFINITION 1 (Mutually exclusive subsets): Given a n-statements program $P =< s_1, s_2, \cdots, s_n >$, and a test suite $T = \{t_1, t_2, \cdots, t_m\}$, and a risk evaluation formula R. Suppose that statement S_f is the real fault in P, run P with test suite T and with four statistics $A_i =< N^i_{CF}, N^i_{CS}, N^i_{UF}, N^i_{US} >$ of each statement collected, then formula R can decompose the statements of P into three mutually exclusive subsets as follow[1]:

1) $S^R_B = \{s_i \in S \mid R(s_i) > R(s_f),\ 1 \le i \le n\}$
2) $S^R_F = \{s_i \in S \mid R(s_i) = R(s_f),\ 1 \le i \le n\}$
3) $S^R_A = \{s_i \in S \mid R(s_i) < R(s_f),\ 1 \le i \le n\}$

The meaning of three exclusive subsets can be explained as follows: (1) S^R_B consists of the statements whose suspiciousness are higher than that of the real fault s_f; (2) S^R_F consists of the statements whose suspiciousness are equal to of the real fault s_f; (3) S^R_A consists of the statements whose suspiciousness are lower than that of the real fault s_f.

Obviously, the size of sets S^R_B and S^R_F greatly affects the efficiency of the fault locating technology. The smaller of sets S^R_B and S^R_F are, the more effective the technique is.

Based on Definition 1, the efficiency of two suspiciousness formulas R_1 and R_2 for single fault locating scenarios can be theoretically analyzed by comparing the size of set S^{R1}_A, set S^{R2}_A, and the size of set S^{R1}_F, set S^{R2}_F. So we can compare the efficiency of risk evaluation formulas for the single-fault locating scenario.

III. An Enhanced Theoretical Analysis Framework for Multi-Fault Locating

In this section, we first introduce the terminologies used in the enhanced framework. Then we discuss an enhanced theoretical analysis framework for multi-fault localization. Furthermore, we also ignored the omission faults in the new framework, since SBFL cannot assign suspiciousness to those missing statements.

A. Preliminaries

DEFINITION 2 (Number of located faults - $\lceil L(\alpha) \rfloor$): For a rank list generated by one fault localization technique with n statements composed. If the developer examined the top l statements of the rank list and identified k faults, then $\alpha = l/n$ denote the percentage of examined code, and $L(\alpha) = k$ denote the number of located faults under the examining level α.

DEFINITION 3 (Better): For two risk evaluation formulas R_1 and R_2, we denote the number of located faults as $L_1(\alpha)$ and $L_2(\alpha)$ under the same code examine level α. If $L_1(\alpha) \ge L_2(\alpha)$, then R_1 is *Better* than R_2, denoted as $R_1(\alpha) \succeq R_2(\alpha)$.

DEFINITION 4 (Congenial): For two risk evaluation formulas R_1 and R_2, we denote the number of located faults as $L_1(\alpha)$ and $L_2(\alpha)$ under the same code examine level α. If $L_1(\alpha) = L_2(\alpha)$, then R_1 is *Congenial* with R_2, denoted as $R_1(\alpha) \cong R_2(\alpha)$.

Furthermore, we can define a new equation L-$score$ to evaluate the efficiency of fault locating when examining the rank list as shown in equation 2. Intuitively, the higher L-$Score$ is, the faster and more faults are located in multi-fault locating.

$$L\text{–}Score = \frac{L(\alpha)}{\alpha} \times 100\% \qquad (2)$$

B. An Enhanced Theoretical Analysis Framework

For a n-statements program $P =< s_1, s_2, \cdots, s_n >$, a test suite $T = \{t_1, t_2, \cdots, t_m\}$, and a fault locating technology with risk evaluation formula R. First, run program P and trace the executions, then decomposed the test suite T and program P into two subsets respectively: T_p (passed test cases) and T_f (failed test cases), S_p (statements covered by passed tests) and S_f (statements covered by failed tests), then applied R with the collected information $A_i =< N^i_{CF}, N^i_{CS}, N^i_{UF}, N^i_{US} >$, the likelihood of fault of each statement were calculated to form a rank list L. Multi-fault locating is to examine the top l statements in the rank list L.

DEFINITION 5 (Mutually exclusive subsets under the dynamic examining level α): For a n-statements program $P =< s_1, s_2, \cdots, s_n >$, applying suspiciousness formula R to generate the rank list L. Under the examining level α ($\alpha = l/n$), the l (Top l) examined statements in list L can be decomposed into three mutually exclusive subsets as follow:

1) $S^R_X(\alpha) = \{s_i \in S_\alpha \mid s_i \in S_f \wedge s_i \notin S_p \}$
2) $S^R_Y(\alpha) = \{s_i \in S_\alpha \mid s_i \in S_f \wedge s_i \in S_p \}$
3) $S^R_Z(\alpha) = \{s_i \in S_\alpha \mid s_i \notin S_f \wedge s_i \in S_p \}$

where $S^R_X(\alpha)$ consists of the statements that only covered by failed test cases. $S^R_Y(\alpha)$ consists of the statements that covered by both failed test cases and passed test cases. $S^R_Z(\alpha)$ consists of the statements that only covered by passed test cases. If $\alpha = 100\%$, the exclusive subsets in definition 5 can be simply denoted as S_X , S_Y and S_Z. The value of four statistics $N^i_{CF}, N^i_{CS}, N^i_{UF}, N^i_{US}$ and the suspiciousness can be derived easily as shown in Table I.

TABLE I
VALUE OF STATISTICS AND SUSPICIOUSNESS OF EACH STATEMENT

s_i	N_{CF}	N_{CS}	N_{UF}	N_{US}	suspiciousness
$s_i \in S_X$	N_F	0	0	N_S	$R(A_i) = R_X$
$s_i \in S_Y$	$[1, N_F]$	$[1, N_S]$	$[0, N_F)$	$[0, N_S)$	$R(A_i) = Var$
$s_i \in S_Z$	0	N_S	N_F	0	$R(A_i) = R_Z$

THEOREM 1: For any risk evaluation formula $R = f(A_i)$, the suspiciousness of all statements in subsets S_X and S_Z are constants R_X and R_Z respectively.

PROOF. First, the suspiciousness of each statement is only decided by the four statistics $N^i_{CF}, N^i_{CS}, N^i_{UF}$ and N^i_{US}. Second, as shown in Table I, the four statistics of sets S_X and S_Z are changeless respectively. Therefore, for a given test suite, the suspiciousness of statement in subsets S_X and S_Z are two constants calculated by the evaluation formula R.

With theorem 1, we can make the proposition as follow:

PROPOSITION 1: There is no difference of the efficiency of different fault localization techniques, where the examining level $\alpha < \dfrac{||S_X||}{n}$ or $\dfrac{||S_X \cup S_Y||}{n} \leq \alpha \leq \dfrac{||S_X \cup S_Y \cup S_Z||}{n}$.

PROOF. As shown in table I, the suspiciousness of statements in S_X and S_Z are two invariants R_X and R_Y, respectively. That is to say, the sequence of statements in both set S_X and set S_Z have nothing to do with the risk evaluation formula. Therefore, different fault evaluation formulas make no difference of the efficiency of fault locating when the examining level $\alpha < \dfrac{||S_X||}{n}$ or $\dfrac{||S_X \cup S_Y||}{n} \leq \alpha \leq \dfrac{||S_X \cup S_Y \cup S_Z||}{n}$.

The risk evaluation formula R can affect the sort of the statements in the subset S_Y in the rank list L when the examining level $\dfrac{||S_X||}{n} \leq \alpha \leq \dfrac{||S_X \cup S_Y||}{n}$.

Intuitively speaking, a statement is more likely to be faulty if it is only executed by failed test cases. On the contrary, a statement is more likely to be correct if it is only executed by passed test cases. Based on definition 5 and theorem 1, two similar cases are discussed in detail in the next two subsections.

C. WellRanked List Analysis

DEFINITION 6 (WellRanked formula): For a risk evaluation formula R, We called R as *WellRanked formula* (denote as $\widehat{R}$), if R satisfies with the condition: $R_X > R(y_i) > R_Z$, where $R(y_i)$ is the suspiciousness of statement y_i ($y_i \in S_Y$), the constants R_X and R_Z are the suspiciousness of statements in S_X and S_Z, respectively.

For a *WellRanked formula* R, We called the rank list generated by R as *WellRanked list*. There are three possible cases of examination for faults locating as shown in Figure 1, when we examined the rank list (L) generated by the *WellRanked formula* formula R.

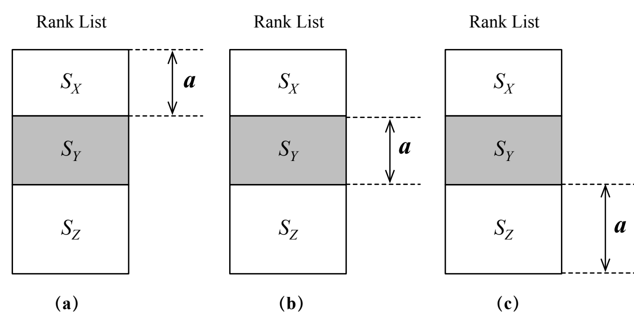

Fig. 1. Possible examinations on *WellRanked* list.

Considering the subgraph (a) and (c) in Figure 1 with Table I, we can now establish two lemmas as follows.

LEMMA 1: For two *WellRanked* formula R_1 and R_2, and the examining level α ($\alpha \leq \frac{||S_X||}{n}$), the efficiency of R_1 and R_2 are congenial, denote as $R_1(\alpha) \cong R_2(\alpha)$.

PROOF. Due to the examining level satisfying $\alpha \leq \frac{||S_X||}{n}$, the developer only needs to examine the statements in the subset S_X. With theorem 1, we know that the suspiciousness of statements in S_X is constants. That is to say, the order of statements in S_X of the rank lists generated by R_1 and R_2 are same with each other, then the number of faults located are same with each other, too. Then we have $L_1(\alpha) = L_2(\alpha)$. Therefore, $R_1(\alpha) \cong R_2(\alpha)$ is hence established.

LEMMA 2: For two *WellRanked* formula R_1 and R_2, and the examine level α ($\frac{||S_X \cup S_Y||}{n} < \alpha \leq 1$), the efficiency of R_1 and R_2 are congenial, denote as $R_1(\alpha) \cong R_2(\alpha)$.

PROOF. Due to the examining level α satisfying $\frac{||S_X \cup S_Y||}{n} < \alpha \leq 1$, the developer needs to examine the whole subset S_X, S_Y, and part of the statements in subset S_Z. With theorem 1, we know that the suspiciousness of statements in S_Z are constants. That is to say, the order of statements in S_Z of the rank lists generated by R_1 and R_2 are same, then the number of faults located are same with each other, $L_1(\alpha) = L_2(\alpha)$. Therefore, $R_1(\alpha) \cong R_2(\alpha)$ is hence established.

Considering the subgraph (b) in Figure 1, the suspiciousness of statements of S_Y is functioning normally to depend on $N^i_{CF}, N^i_{CS}, N^i_{UF}$ and N^i_{US}. For any determined test suite, the following equations $N^i_{CF} + N^i_{UF} = N_F$, $N^i_{CS} + N^i_{US} = N_S$, and $N^i_S + N^i_F = N$ are established. So we can consider the two independent variables N^i_{CF} and N^i_{CS} and classified the formulas into 8 different types depends on the monotonicity of the two variables combinations as table II shown.

TABLE II
TYPES OF RISK EVALUATION FORMULAS

Type	N_{CF}	N_{CS}
M1	increasing	increasing
M2	decreasing	decreasing
M3	increasing	decreasing
M4	decreasing	increasing
M5	indeterminacy	increasing
M6	indeterminacy	decreasing
M7	increasing	indeterminacy
M8	decreasing	indeterminacy

LEMMA 3: For any two *WellRanked* formula R_1 and R_2, and the examine level α ($\frac{||S_X||}{n} < \alpha \leq \frac{||S_X \cup S_Y||}{n}$), the efficiency of R_1 and R_2 can be discussed of the followed two groups.

1) If there are some faults in set S_Y, and R_1 and R_2 are monotonic functions of N_{CS} and N_{CF}, that is to say, they are belonged to the same type of M_1, M_2, M_3 or M_4 listed in table II, then the efficiency of R_1 and R_2 are congenial, denote as $R_1(\alpha) \cong R_2(\alpha)$; otherwise, the the efficiency of R_1 and R_2 cannot be compared.
2) If there is not even a fault among the set S_Y, then the efficiency of R_1 and R_2 are congenial, denote as $R_1(\alpha) \cong R_2(\alpha)$

PROOF. For the first case of lemma 3, the faults in S_Y are called coincidental correctness faults[15]. Since R_1 and R_2 are monotonic functions of N_{CS} and N_{CF}, the statements of set S_Y in ranking list generated by two formulas are with the same order. Accordingly the efficiency of R_1 is same with that of R_2, denoted as $R_1(\alpha) \cong R_2(\alpha)$. On the contrary, it may be difficult to decide which formula is better. For the second case lemma 3, there is no contribution to fault localization among S_Y examining process since no fault of S_Y. Therefore, $R_1(\alpha) \cong R_2(\alpha)$ is hence established.

With lemma 1, 2, 3 , the following theorem 2 are established.

THEOREM 2: For two *WellRanked* evaluation formulas R_1 and R_2, the following conclusion is established.

1) $R_1(\alpha) \cong R_2(\alpha)$ ($\alpha \in [0,1]$) is established without regard to coincidental correctness faults of set S_Y.
2) $R_1(\alpha) \cong R_2(\alpha)$ ($\alpha \in [0,1]$) is established, if R_1 and R_2 are monotonic functions and same with each other in table II, with regard to coincidental correctness faults of set S_Y.

PROOF. To prove the first item. Since there is no fault in set S_Y, after the Lemma 1, Lemma 2, and Lemma 3, we can easily make the conclusion that $R_1(\alpha) \cong R_2(\alpha)$ ($\alpha \in [0,1]$).

To prove the second item. Since there are some coincidental correctness faults of set S_Y, the order in the *WellRanked list* are determined by the results of $R1$ and $R2$. If the monotonies of R_1 and R_2 are same with each other, then we can make a conclusion that $R_1(\alpha) \cong R_2(\alpha)$ ($\alpha \in [0,1]$).

In summary, we can make the conclusion that the *WellRanked formulas* are of the same efficiency in fault localization.

In this section, we have discussed properties of *WellRanked* formulas. The risk value of statements of S_X is strictly greater than that of S_Y, and the risk values of statements of S_Y are strictly greater than that of S_Z. Unfortunately, these strong conditions have not always been established. In the next section, we will discuss the rest cases which do not satisfy with these situations.

D. MayRanked List Analysis

DEFINITION 7 (MayRanked Formula): For a risk evaluation formula R, We called R as *MayRanked* formula (denote as $\widetilde{R}$), if R satisfies the condition: $R_X \geq R(y_i) \geq R_Z$, $R(y_i)$ is the suspiciousness of statement y_i ($y_i \in S_Y$), constants R_X and R_Z are the suspiciousness of statements in S_X and S_Z, respectively.

For a *MayRanked* formula R, we called the rank list generated by R as *MayRanked list*. In order to ascertain the subject for further elaboration, we shall use the following remarks.

$S_{X'} = \{y_i \in S_Y | R(y_i) = R_X\}$
$S_{Z'} = \{y_j \in S_Y | R(y_j) = R_Z\}$

The suspiciousness of statements of sets $S_{X'}$ and $S_{Z'}$ are equal to that of R_X and R_Z, respectively. Therefore, the statements of set $S_{X'}$ can be joined into set S_X, and the statements of set $S_{Z'}$ can be joined into set S_Z.

Similarly, there are three possible cases of examination for locating faults as shown in Fig 2, when we examined the rank list (L) generated by a *MayRanked* formula R.

The size of set $S_{X'}$ and $S_{Z'}$ in figure 2 makes the difference remarkably between figure 2 and figure 1. For instance, (1) if $S_{X'} = \emptyset$, the subgraph (a) in figure 2 is equivalent to that of figure 1. That is to say, the *MayRanked* formula is equivalent to *MayRanked* formula when examining level $\alpha \leq \frac{||S_X||}{n}$. (2) if $S_{Z'} = \emptyset$, the subgraph (c) in figure 2 is equivalent to that of figure 1. That is to say, the *MayRanked formula* is equivalent

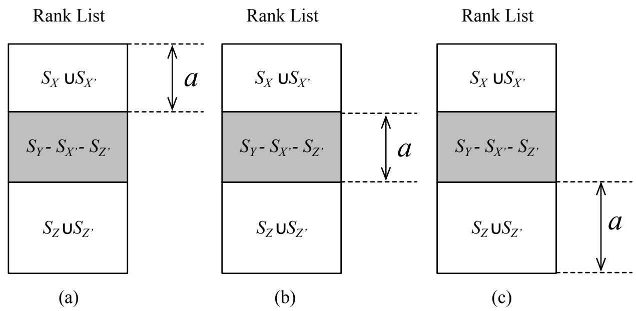

Fig. 2. Possible examinations on *MayRanked* list.

to *MayRanked formula* when examining level $\alpha \geq \frac{||S_X \cup S_Y||}{n}$. (3) if $S_{X'} \neq \emptyset \wedge S_{Z'} \neq \emptyset$, then it can be comprehended that the statements in set $S_{X'}$ or $S_{Z'}$ are joined to the other two sets (S_X and S_Z). Meanwhile, some faults in set $S_{X'}$ or $S_{Z'}$ may spread to the other two sets S_X and S_Z, respectively. In this case, it is unclear whether or not the formula efficiency is boosted for locating faults. Therefore, we proceeded to a more detailed analysis as follows.

For a *MayRanked* formula R which is satisfied with condition $S_{X'} \cup S_{Z'} \neq \emptyset$, and given a *WellRanked* formula R' for comparing, the $\widehat{R'}$ can generate a rank list composed by three sets S_X, S_Y and S_Z, while $\widetilde{R}$ can generate a rank list composed by three sets $S_X \cup S_{X'}$, $S_Y \cup S_{Y'}$ and $S_Z \cup S_{Z'}$.

DEFINITION 8 (I-MayRanked formula): For a *MayRanked* formula R, R is I-*MayRanked formula*, where $S_{X'} = \emptyset \wedge S_{Z'} \neq \emptyset$.

DEFINITION 9 (II-MayRanked formula): For a *MayRanked* formula R, R is II-*MayRanked formula*, where $S_{X'} \neq \emptyset \wedge S_{Z'} = \emptyset$.

DEFINITION 10 (III-MayRanked formula): For a *MayRanked* formula R, R is III-*MayRanked formula*, where $S_{X'} \neq \emptyset \wedge S_{Z'} \neq \emptyset$.

THEOREM 3: For two sets S_X and S_Z with p and q faults, respectively. Two sets $S_{X'}$ and $S_{Z'}$ with p' and q' faults. Let's suppose these $p + p'$ faults spread in $S_X \cup S_{X'}$, and these $q + q'$ faults spread in $S_Z \cup S_{Z'}$. The following conclusions are established:

1) If $\frac{p}{||S_X||} < \frac{p'}{||S_{X'}||}$, that is to say, the fault density of set $S_{X'}$ is greater than that of S_X, the faults in $S_{X'}$ will "spread" to the area of S_X easily. Then, the efficiency of *MayRanked* formula R will be raised comparing with that *WellRanked* formula R'. Otherwise, the *MayRanked* formula R will be less efficient.
2) If $\frac{q}{||S_Z||} > \frac{q'}{||S_{Z'}||}$, that is to say, the fault density of set $S_{Z'}$ is less than that of S_Z, the set $S_{Z'}$ will "absorb" some faults from set S_Z. Then, the efficiency of *MayRanked* formula R will be raised comparing with that *WellRanked* formula R'. Otherwise, the *MayRanked* formula R will be less efficient comparing with R'.

When both S'_X and S'_Z is not null, the efficiency of *MayRanked* formula Rwhich may rise or fallis determined by comparing faulty density. With theorem 3, we deduced the following three propositions.

PROPOSITION 2: For a "I-*MayRanked* formula" R, the two conclusions are established under examine level α.

1) when $\alpha < \frac{||S_X \cup S_Y - S_{Z'}||}{n}$, the efficiency of R is equivalent to *WellRanked* formula.
2) when $\alpha \geq \frac{||S_X \cup S_Y - S_{Z'}||}{n}$, if the fault density of set S'_Z is lower than that of set S_Z, then the efficiency of R is raised comparing with *WellRanked* formulas. Otherwise, the efficiency of R is decreased comparing with *WellRanked* formulas.

PROPOSITION 3: For a "II-*MayRanked* formula" R, if the fault density of set S'_X is higher than that of set S_X, then the efficiency of R, with arbitrarily examine level, is raised comparing with *WellRanked* formulas. Otherwise, the efficiency of R is decreased comparing with *WellRanked* formulas.

PROPOSITION 4: For a "III-*MayRanked* formula" R, the two conclusions are established under examine level α.

1) when $\alpha < \frac{||S_X \cup S_Y - S_{Z'}||}{n}$, if the fault density of set S'_X is higher than that of set S_X, then the efficiency of R is raised comparing with *WellRanked* formulas. Otherwise, the efficiency of R is decreased.
2) when $\alpha \geq \frac{||S_X \cup S_Y - S_{Z'}||}{n}$, if the fault density of set S'_Z is lower than that of set S'_Z, then the efficiency of R is raised comparing with *WellRanked* formulas. Otherwise, the efficiency of R is decreased.

In summary, applying the analytical framework for multi-fault localization scenario presented in this section, we can analyze and compare the efficiency of risk evaluation formulas according to the following steps: first, the risk evaluation formulas are determined whether *WellRanked* type or *MayRanked* type; second, the efficiency of these formulas are compared applying theorem 2 or theorem 3 according to the type of the formulas.

IV. Case Studies

In this section, we will give a case study of analyzing these formulas [1] based on our framework in Section III.

A. Analyzing ***WellRanked*** Formulas

First, we identified whether these formulas are *WellRanked* based on the definition 6. Second, we analyzed the monotonicity of these *WellRanked* formulas by considering the two independent variables N^i_{CF} and N^i_{CS}. Finally, we obtained all *WellRanked* formulas as shown in Table III, and these formulas all belong to "M3" type in monotonicity as stated in Table II.

Taking the proof of formula *Naish2* and *HSS* for example, the corresponding proof processes are given as following. The rest formulas can be proved in a similar way, and hence their proofs are omitted.

PROPOSITION 5: Naish2 is a *M3-WellRanked* formula.

PROOF. (*NA2* is an abbreviation for *Naish2*)

First, follow from the definition of *Naish2* shown in Table III and the metrics stated in Table I, for *Naish2 (abbr. NA2)*,

TABLE III
TYPES OF M3-WELLRANKED FORMULAS

Name	Formula
Naish 2	$N_{CF} - \frac{N_{CS}}{N_{CS}+N_{US}+1}$
Wong3	$\begin{cases} N_{CF} - N_{CS} & , N_{CS} \le 2 \\ N_{CF} - 1.80 - 0.1N_{CS} & , N_{CS} \in (2, 10\ ] \\ N_{CF} - 2.79 - 0.001N_{CS} & , N_{CS} \in (10, \infty) \end{cases}$
Jaccard	$\frac{N_{CF}}{N_{CF}+N_{UF}+N_{CS}}$
Anderberg	$\frac{N_{CF}}{N_{CF}+2(N_{UF}+N_{CS})}$
Sϕrensen-Dice	$\frac{2N_{CF}}{2N_{CF}+N_{UF}+N_{CS}}$
Dice	$\frac{2N_{CF}}{N_{CF}+N_{UF}+N_{CS}}$
Goodman	$\frac{2N_{CF}-N_{UF}-N_{CS}}{2N_{CF}+N_{UF}+N_{CS}}$
Tarantula	$\frac{N_{CF}(s)}{N_F} / \frac{N_{CS}(s)}{N_S} + \frac{N_{CF}(s)}{N_F}$
q_e	$\frac{N_{CF}}{N_{CF}+N_{CS}}$
CBI Inc.	$\frac{N_{CF}}{N_{CF}+N_{CS}}$- $\frac{N_{CF}+N_{UF}}{N_{CF}+N_{CS}+N_{UF}+N_{US}}$
Wong2	$N_{CF} - N_{CS}$
Hamann	$\frac{N_{CF}+N_{US}-N_{UF}-N_{CS}}{N_{CF}+N_{CS}+N_{UF}+N_{US}}$
Simple Matching	$\frac{N_{CF}+N_{US}}{N_{CF}+N_{CS}+N_{UF}+N_{US}}$
Sokal	$\frac{2(N_{CF}+N_{US})}{2(N_{CF}+N_{US})+N_{CS}+N_{UF}}$
Rogers & Tanimoto	$\frac{N_{CF}+N_{US}}{N_{CF}+N_{US}+2(N_{CS}+N_{UF})}$
Hamming etc.	$N_{CF} + N_{US}$
Euclid	$\sqrt{N_{CF} + N_{US}}$
Ochiai	$\frac{N_{CF}}{\sqrt{(N_{CF}+N_{UF})(N_{CF}+N_{CS})}}$
M2	$\frac{N_{CF}}{N_{CF}+N_{US}+2(N_{UF}+N_{CS})}$
AMPLE2	$\frac{N_{CF}}{N_{CF}+N_{UF}} - \frac{N_{CS}}{N_{CS}+N_{US}}$
Arithmetic Mean	$\frac{2N_{CF}N_{US}-2N_{UF}N_{CS}}{(N_{CF}+N_{CS})(N_{UF}+N_{US})+(N_{CF}+N_{UF})(N_{CS}+N_{US})}$
Cohen	$\frac{2N_{CF}N_{US}-2N_{UF}N_{CS}}{(N_{CF}+N_{CS})(N_{CS}+N_{US})+(N_{CF}+N_{UF})(N_{UF}+N_{US})}$
Scott	$\frac{4N_{CF}N_{US}-4N_{UF}N_{CS}-(N_{UF}-N_{CS})^2}{(2N_{CF}+N_{UF}+N_{CS})(2N_{US}+N_{UF}+N_{CS})}$

we have

$$\begin{cases} R_X^{NA2} = N_F \\ R_Z^{NA2} = -N_S/(N_S+1) \\ R_Y^{NA2} = N_{CF} - N_{CS}/(N_{CS}+N_{US}+1)\ . \end{cases}$$

Since $N_F = N_{CF} + N_{UF}$ and the metrics N_{CF}, N_{UF}, N_{CS} and N_{US} are all greater than 0. Then we have $N_F > N_{CF}$, $N_{CS}/(N_{CS}+N_{US}+1) > 0$, $N_{CF} > N_{CF} - N_{CS}/(N_{CS} + N_{US}+1)$.

Thus, $N_F > N_{CF} - N_{CS}/(N_{CS}+N_{US}+1)$. Immediately, we have proved $R_X^{NA2} > R_Y^{NA2}$.

Similarly, we have $-N_S/(N_S+1) < -1$, while $N_{CF} - N_{CS}/(N_{CS}+N_{US}+1) > 0$. Immediately, we also have proved $R_Y^{NA2} > R_Z^{NA2}$.

Thus, *Naish2* is satisfied with conditions $R_X^{NA2} > R_Y^{NA2} > R_Z^{NA2}$. After Definition 6, *Naish2* is a *WellRanked* formula.

Second, after the formula *Naish2*, the suspiciousness increases monotonically with increasing N_{CF} and decreasing N_{CS}, then *Naish2* belongs to "M3" type as stated in Table II.

In summary, *Naish2* is a *M3-WellRanked* formula.

PROPOSITION 6: HSS is a *M3-WellRanked* formula.

PROOF. (*H* is an abbreviation for *HSS*)

First, follow from the definition of *HSS* shown in Table III and the metrics stated in Table I, for *HSS, (abbr. H)*, we have

$$\begin{cases} R_X^H = \frac{N_F^2}{N} \\ R_Z^H = 0 \\ R_Y^H = \frac{N_{CF}^2}{N} - \frac{N_{CF}\ N_{CS}}{N^2}\ . \end{cases}$$

(A). To prove that $R_X^H > R_Y^H$.

Since $N_F \ge N_{CF}$, we have $\frac{N_F^2}{N} \ge \frac{N_{CF}^2}{N}$.

Furthermore, after the metrics of line 3 stated in Table I, we have $\frac{N_{CF}N_{CS}}{N^2} > 0$

Then, $\frac{N_F^2}{N} \ge \frac{N_{CF}^2}{N} > \frac{N_{CF}^2}{N} - \frac{N_{CF}N_{CS}}{N^2}$.

Therefore, $R_X^H > R_Y^H$.

(B). To prove that $R_Y^H > R_Z^H$.

From line 3 in Table I, we have $1 \le N_{CF} \le N_F < N$, and $1 \le N_{CS} \le N_F < N$, $N_{CF} > 1 > \frac{N_{CS}}{N}$.

Then $N_{CF} > \frac{N_{CS}}{N}$ established, multiplying each side by $\frac{N_{CF}}{N}$ and rearranging the terms, we have $\frac{N_{CF}^2}{N} > \frac{N_{CF}N_{CS}}{N^2}$.Then we have $\frac{N_{CF}^2}{N} - \frac{N_{CF}N_{CS}}{N^2} > 0$.

Therefore, $R_Y^H > R_Z^H$.

Considering proof (A) and (B) that we know *HSS* is satisfied with conditions $R_X^H > R_Y^H > R_Z^H$. After Definition 6, *HSS* is a *WellRanked* formula.

Second, We will analyze the Monotonicity of formula *HSS*. For simplify, we use x, y to denote N_{CF}, N_{CS}, respectively. and *HSS* can be represent as $z(x, y) = \frac{x^2}{N} - \frac{xy}{N^2}$, N is the total number of test cases which is a constant in fault locating.

(C). To prove that *HSS* increases monotonically with increasing N_{CF}, ($\frac{\partial z}{\partial x} > 0$).

Since $\frac{\partial z}{\partial x} = \frac{2x}{N} - \frac{y}{N^2}$. And after the metrics stated in Table I , we have $x \ge 1, 0 < y < N$, then $2x > 1 > \frac{y}{N}$.

Thus, $\frac{2x}{N} > \frac{y}{N^2}$.

Therefore, $\frac{\partial z}{\partial x} > 0$.

(D). To prove that *HSS* increases monotonically with decreasing N_{CS}, ($\frac{\partial z}{\partial y} < 0$).

Since $\frac{\partial z}{\partial y} = -\frac{x}{N^2}$. And after the metrics stated in Table I , we have $x > 0, N > 0$, then $-\frac{x}{N^2} < 0$.

Therefore, $\frac{\partial z}{\partial y} < 0$.

Considering proof (C) and (D) that we know *HSS* increases monotonically with increasing N_{CF} and decreasing N_{CS}, then *HSS* belongs to "M3" type as stated in Table II.

In summary, *HSS* is a *M3-WellRanked* formula.

B. Analyzing ***MayRanked*** *Formulas*

It follows from Definition 7, Definition 8, Definition 9 and Definition 10 that the suspiciousness formulas can be classified as shown in Table IV.

Next, we give the detailed proofs of the formula *Wong1* for example. The other two formulas can be proved in a similar way as that of *Wong1*, and hence their proofs are omitted.

PROPOSITION 7: Wong1 is a *II-MayRanked* formula.

PROOF. (*W1* is an abbreviation for *Wong1*)

First, follow from the definition of *Wong1* shown in Table IV and the metrics stated in Table I, for *Wong1 (abbr. W1)*,

TABLE IV
TYPES OF RISK EVALUATION FORMULAS

Type	Name	Formula
II	Wong 1	N_{CF}
II	Russel&Rao	$\frac{N_{CF}}{N_{CF}+N_{UF}+N_{CS}+N_{US}}$
II	Kulczyski2	$\frac{1}{2}\Big(\frac{N_{CF}}{N_{CF}+N_{UF}}+\frac{N_{CF}}{N_{CF}+N_{CS}}\Big)$
III	Naish 1	$\begin{cases} -1 & \text{, if } N_{CF} < N_F \\ N_S - N_{CS} & \text{, if } N_{CF} = N_F \end{cases}$
III	Binary	$\begin{cases} 0 & \text{, if } N_{CF} < N_F \\ 1 & \text{, if } N_{CF} = N_F \end{cases}$

we have

$$\begin{cases} R_X^{W1} = N_F \\ R_Z^{W1} = 0 \\ R_Y^{W1} = N_{CF} \ . \end{cases}$$

It is satisfied with Definition 7. Thus, the formula *Wong1* is a *MayRanked* formula.

Furthermore, since $N_F \geq N_{CF} > 0$, then we have $S_{X'} = S_Y$ and $S_{Z'} = \emptyset$, that is to say $R_X^{W1} \geq R_Y^{W1} > R_Z^{W1}$. Therefore *Wong1* is *II* type.

In summary, *Wong1* is a *II-MayRanked* formula.

Although the formulas include $Wong1$, $Russel\&Rao$, $Naish1$ and $Binary$ in Table IV have been proved to be maximal by Xie's framework[1], they are classified to different type applied by our framework. Therefore, the maximal formulas in single-fault locating scenario does not outstanding necessarily in multi-fault locating scenario.

V. CONCLUSIONS AND FUTURE WORK

The motivation of this study was to better understand the efficiency of SBFL formulas in real program debugging. To this end, we first proposed a practical metric (L-$Score$) for measurement for multi-fault locating efforts. Then, illuminated by Xie's work [1], we present a framework for multi-fault locating based on which we have conducted a theoretical analysis of a set of existing formulas. The result indicated that the formula has relatively little effect on how well a fault localization technique performs.

Our work is preliminary but the result is encouraging and interesting. In the future, there will be some aspects to improve and study. (1) more features should be introduced into the designing of evaluation formula. (2) besides formula designing, we also need to construct multi-cost model. (3) an interesting work is to investigate the characteristics of real faults. (4) we still should take into account the effort of patch generation. That is to say, the bug fixing effort should be considered when building fault locating models.

ACKNOWLEDGMENT

This work was supported in part by awards from the National Natural Science Foundation of China under Grant Nos. 61502497, 61602267 and 61673384, the Guangxi Key Laboratory of Trusted Software Research Plan under Grant Nos. KX201530 and KX201532, and the National Natural Science Foundation of Guanxi under Grant No 2015GXNSFDA139038.

REFERENCES

[1] X. Xie, T. Y. Chen, F.-C. Kuo, and B. Xu, "A theoretical analysis of the risk evaluation formulas for spectrum-based fault localization," *ACM Transactions on Software Engineering and Methodology*, vol. 22, no. 4, pp. 1–40, 2013.

[2] J. A. Jones and M. J. Harrold, "Empirical evaluation of the tarantula automatic fault-localization technique," in *Proceedings of the 20th IEEE/ACM International Conference on Automated Software Engineering (ASE 2005)*. New York, USA: ACM, 2005, pp. 273–282.

[3] R. Abreu, P. Zoeteweij, and A. J. C. van Gemund, "An evaluation of similarity coefficients for software fault localization," in *Proceedings of the 12th Pacific Rim International Symposium on Dependable Computing (PRDC 2006)*, 2006, pp. 39–46.

[4] L. Naish, H. J. Lee, and K. Ramamohanarao, "A model for spectra-based software diagnosis," *ACM Transactions on Software Engineering and Methodology*, vol. 20, no. 3, pp. 11–43, 2011.

[5] D. Hao, L. Zhang, Y. Pan, H. Mei, and J. Sun, "On similarity-awareness in testing-based fault localization," in *Proceedings of the 23rd IEEE/ACM International Conference on Automated Software Engineering (ASE 2008)*, vol. 15. New York, USA: ACM, 2008, pp. 207–249.

[6] W. E. Wong, R. Gao, Y. Li, R. Abreu, and F. Wotawa, "A survey on software fault localization," *IEEE Transactions on Software Engineering*, vol. 42, no. 8, pp. 1–41, 2016.

[7] S. Pearson, J. Campos, R. Just, G. Fraser, R. Abreu, M. D. Ernst, D. Pang, and B. Keller, "Evaluating and improving fault localization techniques," in *ICSE'17, Proceedings of the 39th International Conference on Software Engineering*, Conference Proceedings.

[8] H. J. Lee, L. Naish, and K. Ramamohanarao, "Study of the relationship of bug consistency with respect to performance of spectra metrics," in *International Conference on Computer Science and Information Technology*, 2009.

[9] S. Yoo, "Evolving human competitive spectra-based fault localisation techniques," in *Search Based Software Engineering*. Berlin Heidelberg: Springer, 2012, pp. 244–258.

[10] X. Xie, F.-C. Kuo, T. Y. Chen, S. Yoo, and M. Harman, "Provably optimal and human-competitive results in sbse for spectrum based fault localisation," in *Search Based Software Engineering*. Berlin Heidelberg: Springer, 2013, pp. 224–238.

[11] N. Neelofar, L. Naish, J. Lee, and K. Ramamohanarao, "Improving spectral-based fault localization using static analysis," *Software: Practice and Experience*, pp. n/a–n/a, 2017, spe.2490. [Online]. Available: http://dx.doi.org/10.1002/spe.2490

[12] C. M. Tang, W. Chan, and Y. Yu, "Extending the theoretical fault localizaton effectiveness hierarchy with empirical results at different code abstraction levels," in *Proceedings of the 38th Annual International Computer Software and Applications Conference (COMPSAC 2014)*. IEEE, 2014, pp. 161–170.

[13] R. Santelices, J. A. Jones, Y. Yu, and M. J. Harrold, "Lightweight fault-localization using multiple coverage types," in *Proceedings of the 31st International Conference on Software Engineering (ICSE 2009)*. Los Alamitos, CA: IEEE computer society, 2009, pp. 56–66.

[14] W. E. Wong, V. Debroy, and B. Choi, "A family of code coverage-based heuristics for effective fault localization," *Journal of Systems and Software*, vol. 83, no. 2, pp. 188–208, 2010.

[15] W. Masri and R. A. Assi, "Prevalence of coincidental correctness and mitigation of its impact on fault localization," *ACM Transactions on Software Engineering and Methodology*, vol. 23, no. 1, pp. 1–28, 2014.