A Revisit of Fault-Detecting Probability of Combinatorial Testing

Min Yu Feiyan She Yuanchao Qi Ziyuan Wang∗ Weifeng Zhang
School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, China
∗Corresponding: wangziyuan@njupt.edu.cn

Abstract—The lower bound of fault-detecting probability of \( \tau \)-way combinatorial test suite for Boolean-specification testing has been proposed [1]. However, the formula neglected the situation that for different minimal failure-causing schemas, the coverage of test suites may be non-independent events. Hence, multiplying directly the probabilities of non-independent events is incorrect, which causes that the result calculated by previous formula may be greater. In this paper, we give counterexamples to demonstrate the mistakes in the formula derivation. Furthermore, two experiments are designed to illustrate that the actual fault-detecting probabilities and ratios are usually less than the theoretical fault-detecting probabilities calculated from the previous formula.

Index Terms—Software testing, combinatorial testing, fault-detecting probability, minimal failure-causing schema.

I. INTRODUCTION

Combinatorial testing is a well-accepted testing method which has been widely studied and applied [2]. There are growing concerns about the fault-detecting ability of combinatorial testing, due to a controversy about combinatorial test suites effectiveness of detecting faults.

The fault-detecting probability was proposed to evaluate the ease of a combinatorial test suite detecting a fault. Previous works have put forward formulas for fault-detecting probability of combinatorial testing [1][3]. The lower bound of the fault-detecting probability of fixed-strength combinatorial test suite for 2-level system has been raised in [1]. On this basis, the article [3] further proposed the lower bound of the probability that fixed-strength combinatorial test suite detects faults for mixed-level system. However, we find that the events that combinatorial test suites detect faults under different minimal failure-causing schemas may be non-independent sometimes. So it is incorrect to multiply directly the probabilities of non-independent events in the formula. In terms of this mistake, we carry out some relevant researches and experiments.

In this paper, we take Boolean expressions from TCAS system as study object to find counterexamples of the formula and generate combinatorial test suites with different strengths by some classic combinatorial test generation algorithms. By analyzing the test suites of boolean expressions, the fault-detecting probabilities and ratios of the combinatorial test suites are calculated. A contrast of experimental results and theoretical values calculated by the formula is made by us.

The rest of this paper is organized as follows. The 2nd section proposes the formula of fault-detecting probability and puts forward the counterexample of the formula. The 3rd section is the description of experiments and gives experimental results to illustrate the differences with theoretical values. There is a conclusion finally.

II. FAULT-DETECTING PROBABILITY

A. Existing formula of fault-detecting probability

Suppose there are \( m \) minimal failure-causing schemas, with strength \( k_1, k_2, \ldots, k_m \) respectively, for a fault. For an arbitrary 2-level \( \tau \)-way combinatorial test suite \( T \), the probability of event that \( T \) detects this fault should be [1]:

\[
p(\tau) = \begin{cases} 1 \quad & \tau \geq \min_{i=1}^{m}\{k_i\} \\ 1 - \prod_{i=1}^{m} (1 - \frac{1}{2^{k_i - \tau}})^{C_{k_i}} \quad & \tau < \min_{i=1}^{m}\{k_i\} \end{cases}
\]

B. Counter-example

There is Lemma 6 in [1]: “Suppose there are totally \( m \) schemas, and their strengths are \( k_1, k_2, \ldots, k_m \) respectively. Let \( T \) be a 2-level suite, in which all possible \( \tau \)-value sub-schemas of all \( m \) schemas are covered by at least test case \( \tau < \min_{i=1}^{m}\{k_i\} \). The probability of event that none of these \( m \) schemas are covered by \( T \) is equal to or less than

\[
\prod_{i=1}^{m} (1 - \frac{1}{2^{k_i - \tau}})^{C_{k_i}}.
\]

If there are multiple minimal failure-causing schemas, the lemma 6 calculates probability for each and multiply them. However, only when events are mutually independent, they can be multiplied in probability theory. If a test case covers current minimal failure-causing schema, it may cover others. That is, many overlapping test cases are repeatedly calculated when taking each minimal failure-causing schema as an independent event. The actual probability should discard overlaps.

For instance, there are 2 minimal failure-causing schemas \( S_1=(1\ 1\ 1\ -\ -\ -) \), \( S_2=(\ -\ -\ -\ 1\ 1\ 1\ -) \), \( k=3 \). Their \( \tau=2 \)-value sub-schemas are \( (1\ 1\ -\ -\ -) \), \( (\ -\ -\ -\ 1\ 1\ 1\ -) \), \( (1\ 1\ -\ -\ -) \) and \( (\ -\ -\ -\ 1\ 1\ 1\ -) \). Let \( A \) be the event that \( S_1 \) is not covered by a 2-way combinatorial test suite, and \( B \) the same event for \( S_2 \). According to Lemma 5 in [1], \( P(A) = P(B) \leq (1 - \frac{1}{2})^3 = \frac{1}{8} \). According to Lemma 6 in [1], \( P(AB) = P(A) \times P(B) \leq \frac{1}{32} \) and \( P_{\tau=2} = 1 - P(AB) \geq \frac{63}{64} \).

However, since \( A \) and \( B \) are not independent, \( P(AB) = P(A) \times P(B) \) can’t be used to calculate the probability that \( A \) and \( B \) occur at the same time. In fact, if \( A \) occurs, there should be 3 test cases \( (1\ 1\ 0\ x\ x) \), \( (1\ 0\ 1\ x\ x) \), \( (0\ 1\ 1\ x\ x) \) in combinatorial test suite, in which only \( (0\ 1\ 1\ x\ x) \) has chance to cover schemas \( (\ -\ -\ -\ 1\ 1\ 1\ -) \). \( P(AB) = P(A) \times P(B|A) \leq
Experimental results

By the running of large number of combinatorial test suites, $\tau$ different each combinatorial test generation algorithm will generate for each original Boolean expression from TCAS system, calculated according to the formula in [1]. In the second step, detecting probability of combinatorial test suite will be calculated in order to output the experiment. When running an algorithm, a random generated variable strength combinatorial testing, are also utilized in our experiment. There have been formula about the fault-detecting probability of combinatorial in previous researches. Due to overlooking the independence of probability event, the formula is inaccurate. In this work, we take 20 boolean expressions as experimental subject to research fault-detecting frequencies and ratios of combinatorial test suites generated by five classic combinatorial test generation algorithms. From the results we come to a conclusion there are obvious deviations between theoretical and real value of fault detection. Because of higher theoretical values than experimental results, the formula in [1] is mistaken.

In future, we will correct the mistakes in the formula and figure out the accurate theoretical fault-detecting probability.

C. Experimental results

1) Results for RQ1: In Fig.1, there are 20 groups of box-graphs in each figure, where each group stands for an original Boolean expression from TCAS. Besides the theoretical faulting-detecting probabilities in the first box-graph of each group, the last 5 box-graphs in each group illustrate the ratios of killed mutants, which are mutated from current original Boolean expression, by each $\tau$-way combinatorial test suite that generated by Greedy algorithm, DDA algorithm, IPO algorithm, DensityRO algorithm, and ReqMerge algorithm respectively, where $\tau=2, 3, 4$.

2) Results for RQ2: In Fig. 2, there are 20 groups of box-graphs in each figure, where each group stands for an original Boolean expression from TCAS. Besides the theoretical faulting-detecting probabilities in the first box-graph of each group, the last 5 box-graphs in each group illustrate the ratios of killed mutants, which are mutated from current original Boolean expression, by each $\tau$-way combinatorial test suite that generated by Greedy algorithm, DDA algorithm, IPO algorithm, DensityRO algorithm, and ReqMerge algorithm respectively, where $\tau=2, 3, 4$.

These results indicate that, the mutants’ actual fault-detecting frequencies and the mean values of the ratios of detected faults are usually less than the mean values of mutants’ theoretical fault-detecting probabilities calculated by formula in [1].

B. Experiment setup

We take 20 Boolean expressions that extracted from the TCAS system as the experimental subjects. For each expression, we create mutants by 10 fault types and get 19131 non-equivalent mutants.

Combinatorial test suites with different strengths are generated by some classic combinatorial test generation algorithms including Greedy algorithm [4], DDA algorithm [5], and IPO algorithm [6]. And besides, two algorithms named ReqMerge [7] and DensityRO [8], which were mainly designed for variable strength combinatorial testing, are also utilized in our experiment. When running an algorithm, a random generated seeding test case will be assigned in order to output the rich diversity combinatorial test suites in the large number of runnings of combinatorial testing.

In the first step, for each mutant, the theoretical fault-detecting probability of combinatorial test suite will be calculated according to the formula in [1]. In the second step, for each original Boolean expression from TCAS system, each combinatorial test generation algorithm will generate 100 different $\tau$-way combinatorial test suites for $\tau=2, 3, 4$. By the running of large number of combinatorial test suites, experimental results could be obtained:

- To answer the 1st research question, for each mutant, the fault-detecting frequency in the large number of the running of combinatorial testing should be collected.
- To answer the 2nd research question, for each combinatorial test suite, the ratio of the number of killed mutants to the total number of mutants should be collected.

C. Experimental results

1) Results for RQ1: In Fig.1, there are 20 groups of box-graphs in each figure, where each group stands for an original Boolean expression from TCAS. Besides the theoretical faulting-detecting probabilities in the first box-graph of each group, the last 5 box-graphs in each group illustrate fault-detecting frequencies of mutants, which are mutated from current original Boolean expression, in the running of $\tau$-way combinatorial test suites that generated by Greedy algorithm, DDA algorithm, IPO algorithm, DensityRO algorithm, and ReqMerge algorithm respectively, where $\tau=2, 3, 4$.

IV. Conclusion

There have been formula about the fault-detecting probability of combinatorial in previous researches. Due to overlooking the independence of probability event, the formula is inaccurate. In this work, we take 20 boolean expressions as experimental subject to research fault-detecting frequencies and ratios of combinatorial test suites generated by five classic combinatorial test generation algorithms. From the results we come to a conclusion there are obvious deviations between theoretical and real value of fault detection. Because of higher theoretical values than experimental results, the formula in [1] is mistaken.

In future, we will correct the mistakes in the formula and figure out the accurate theoretical fault-detecting probability.

ACKNOWLEDGMENT

This work is supported by the National Nature Science Foundation of China (61772259).

REFERENCES


Fig. 1 Compare Fault-Detecting Probabilities and Frequencies of $\tau$-way Combinatorial Test Suite for 20 Boolean Expressions ($\tau = 2, 3, 4$)

Fig. 2 Compare Fault-Detecting Probabilities and Ratios of Detected Faults of $\tau$-way Combinatorial Test Suite for 20 Boolean Expressions ($\tau = 2, 3, 4$)
