

# Supplementary material for:

## Mutant reduction evaluation: what is there and what is missing?

Peng Zhang, Yang Wang, Xutong Liu, Yanhui Li,  
Yibiao Yang, Ziyuan Wang, Xiaoyu Zhou, Lin Chen, Yuming Zhou

----- ◆ -----

### 1 AUTOMATICALLY GENERATED TEST SUITES FOR EVALUATION

In the manuscript, we used the manually written tests as the evaluation objects of mutation testing. The following question may be raised: can the conclusion on manually written tests be extended to the automatically generated tests? To investigate this question, we used *EvoSuite* to generate tests. We set the criterion as “MUTATION” and used other default configuration. However, the mutation score is relatively low. As a result, it may be impractical to use automated test case generation tools to obtain test suites with a high strength in real-world projects. To this concern, we delete the surviving mutants and use the remaining mutants for evaluation. This demo experiment uses one project *pool*. The other settings are the same with RQ1. Table 1 reports for 5 strategies the average *OP*, *VMS*, and *GMS*. We can find that the conclusion is consistent to the manuscript.

Table 1. The discrimination ability comparison between *OP*, *VMS*, and *GMS* under an automatically generated test suite

Project	OP					Variation of MS					GMS				
	RMS	Sentinel	COS	SMS	CMS	RMS	Sentinel	COS	SMS	CMS	RMS	Sentinel	COS	SMS	CMS
P2	0.923	0.926	0.912	0.965	0.959	0	0	0	0	0	0.755	0.762	0.791	0.934	0.831

### 2 IF THE DUPLICATED MUTANTS AFFECT THE ORIGINAL MUTATION SCORE, CAN IT STILL BE USED AS THE GROUND TRUTH?

It is widely accepted that duplicated mutants are redundant. To this concern, we need to discuss whether these mutants affect our conclusions. Actually, it is easy to produce a different ordering by duplicated mutants. Assume that  $m_1$  is killed by  $t_1$  and  $m_2$  is killed by  $t_2$ . Then, we have:

$$MS(\{m_1, m_2\}, \{t_1\}) = MS(\{m_1, m_2\}, \{t_2\}) = 0.5$$

$$MS(\{m_1, m_1, m_1, m_2\}, \{t_1\}) = 0.75 > MS(\{m_1, m_1, m_1, m_2\}, \{t_2\}) = 0.25$$

It can be seen that the order between  $\{t_1\}$  and  $\{t_2\}$  is twisted by adding the duplicated  $m_1$ . As a result, it is natural to ask the following question: if the redundant mutants affect the original mutation score, can it still be used as the ground truth?

Let us first clarify our conclusion: for those duplicated mutants, whether they are removed or not will not affect our conclusion in the manuscript. In other words, if we add the “subset” restriction, we can avoid this problem. Given a mutant set  $M$ , for two test suites  $T_1 \subset T_2$

(1) if the following constraint is held:

$$MS(M, T_1) = \frac{|KM(M, T_1)|}{|M|} < MS(M, T_2) = \frac{|KM(M, T_2)|}{|M|}$$

we next analyze whether the deletion of duplicated mutants will affect the order:

- $m_1$  is alive under  $T_2$  and  $T_1$  and there are  $k$  duplicated  $m_1$  in  $M$ . After deleting  $k$   $m_1$ , we obtain the reduced mutant set  $M'$ :

$$MS(M', T_1) = \frac{|KM(M', T_1)|}{|M'|} = \frac{|KM(M, T_1)|}{|M'|} < MS(M', T_2) = \frac{|KM(M', T_2)|}{|M'|} = \frac{|KM(M, T_2)|}{|M'|}$$

- $m_1$  is killed under  $T_2$  and  $T_1$  and there are  $k$  duplicated  $m_1$  in  $M$ . After deleting  $k$   $m_1$ , we obtain the reduced mutant set  $M'$ :

$$\begin{aligned} MS(M', T_1) &= \frac{|KM(M', T_1)|}{|M'|} = \frac{|KM(M, T_1)| - k}{|M'|} \\ &< MS(M', T_2) = \frac{|KM(M', T_2)|}{|M'|} = \frac{|KM(M, T_2)| - k}{|M'|} \end{aligned}$$

- $m_1$  is killed under  $T_2$  and alive under  $T_1$  while there are  $k$  duplicated  $m_1$  in  $M$ . Then we have:

$$|KM(M, T_1)| \leq |KM(M, T_2)| - (k + 1)$$

After deleting  $k$   $m_1$ , we obtain the reduced mutant set  $M'$ :

$$\begin{aligned} MS(M', T_1) &= \frac{|KM(M', T_1)|}{|M'|} = \frac{|KM(M, T_1)|}{|M'|} \\ &< MS(M', T_2) = \frac{|KM(M', T_2)|}{|M'|} = \frac{|KM(M, T_2)| - k}{|M'|} \end{aligned}$$

Note that  $m_1$  can not be killed under  $T_1$  while it is alive under  $T_2$ .

(2) if the following constraint is held:

$$MS(M, T_1) = \frac{|KM(M, T_1)|}{|M|} = MS(M, T_2) = \frac{|KM(M, T_2)|}{|M|}$$

we next analyze whether the deletion of duplicated mutants will affect the order:

- $m_1$  is alive under  $T_2$  and  $T_1$  and there are  $k$  duplicated  $m_1$  in  $M$ . After deleting  $k$   $m_1$ , we obtain the reduced mutant set  $M'$ :

$$MS(M', T_1) = \frac{|KM(M', T_1)|}{|M'|} = \frac{|KM(M, T_1)|}{|M'|} = MS(M', T_2) = \frac{|KM(M', T_2)|}{|M'|} = \frac{|KM(M, T_2)|}{|M'|}$$

- $m_1$  is killed under  $T_2$  and  $T_1$  and there are  $k$  duplicated  $m_1$  in  $M$ . After deleting  $k$   $m_1$ , we obtain the reduced mutant set  $M'$ :

$$\begin{aligned} MS(M', T_1) &= \frac{|KM(M', T_1)|}{|M'|} = \frac{|KM(M, T_1)| - k}{|M'|} \\ &= MS(M', T_2) = \frac{|KM(M', T_2)|}{|M'|} = \frac{|KM(M, T_2)| - k}{|M'|} \end{aligned}$$

Note that  $m_1$  must keep the same label under  $T_1$  and  $T_2$ .

Based on the above analysis, we can see that if we only compare the order among a suite with its subsets, the order never will be twisted by duplicated mutants.

### 3 RMS CAN MAINTAIN THE MUTATION SCORE IN MATHEMATICAL EXPECTATION

Assume that  $T$  is the test suite on the SUT with the original mutant set  $M$ . For RMS, let  $M'$  be the set of the mutants reduced from  $M$ . The random reduction strategy RMS always maintains the following property:

$$E[MS(M', T)] = MS(M, T)$$

where  $E$  is a mathematical expectation. In the following, we give a brief proof. Assume that  $|M| = n$ ,  $|M'| = m$ , and  $|KM(M, T)| = k$ . Then, we have:

$$\begin{aligned}
E[MS(M', T)] &= E\left[\frac{|KM(M', T)|}{|M'|}\right] \\
&= \frac{E[|KM(M', T)|]}{|M'|} \\
&= \frac{1}{m} E[|KM(M', T)|] \\
&= \frac{1}{m} \frac{1}{C_n^m} \sum_{i=0}^k i C_k^i C_{n-k}^{m-i} \\
&= \frac{1}{m} \frac{1}{C_n^m} \sum_{i=1}^k i C_k^i C_{n-k}^{m-i} \\
&= \frac{1}{m} \frac{1}{C_n^m} \sum_{i=1}^k i \frac{k!}{(k-i)! i!} C_{n-k}^{m-i} \\
&= \frac{1}{m} \frac{1}{C_n^m} \sum_{i=1}^k k \frac{(k-1)!}{(i-1)! ((k-1)-(i-1))!} C_{n-k}^{m-i} \\
&= \frac{1}{m} \frac{1}{C_n^m} k \sum_{i=1}^k C_{k-1}^{i-1} C_{n-k}^{m-i} \\
&= \frac{1}{m} \frac{C_{n-1}^{m-1}}{C_n^m} k \\
&= \frac{1}{m} \frac{m}{n} k \\
&= \frac{k}{n} \\
&= MS(M, T)
\end{aligned}$$

This means that RMS always keeps the mutation score unchanged in expectation. Under VMS, it is natural that all pretty strategies are similar to a random strategy. However, from the viewpoint of RMS, all mutants have the same weight. That is to say, RMS focuses on the number of reductions more than the effect of reductions. If an indicator on the effect of reductions gives a high evaluation for the strategy that does not pay much attention to the effect of reductions, the rationality of this indicator maybe questionable.

#### 4 IS THE CONCLUSION OF OP GREATLY AFFECTED BY THE MUTATION TESTING TOOL?

In the manuscript, we used *PIT* as the tool of mutation testing. Then the question is that can the conclusion be extended to other tools? To investigate this question, we use *Major* to execute mutation testing. However, *Major* is executed by *ant* and most of the used projects are not built by *ant*. To this concern, we use *defects4j*, which is a widely used dataset for software testing. *defects4j* provides the option for mutation analysis based on *Major*. It saves us the work of building projects one by one. Meanwhile, most of the programs/projects used in our manuscript are used in *defects4j*. However, we use the different versions compared to *defects4j*.

We designed the experiment as follows: for a program we used in the manuscript, if it exists in the last fixed (i.e. bug-free) version for the corresponding project in the *defects4j* dataset, we use *Major* to analyze the mutants. For example, the program “*Option*” is in project “*Cli*”. Then we use the command: “*defects4j checkout -p Cli -v 40f*” to obtain the last fixed version of *Cli*. After that, if “*Option.java*” is in it, we run the test cases one by one to obtain the kill matrix. Other experimental settings are the same as RQ1 in the manuscript. Note that code of “*Option*” used here may be different to the “*Option*” in the manuscript. Then the result is showed in Table 2. From Table 2, we can find that the overall conclusion is consistent to the

manuscript: SMS and CMS are better than other 3 strategies. To conclude, when using OP to compare several strategies, the overall conclusion is not affected by the mutation testing tools.

Table 2. The OP values computed against Major

Program	OP				
	RMS	Sentinel	COS	SMS	CMS
Crypt	0.863	0.891	0.869	0.970	0.863
StringUtils	0.797	0.798	0.790	0.902	0.867
Md5Crypt	0.754	0.740	0.753	0.782	0.868
DefaultParser	0.892	0.864	-	0.906	0.929
Option	0.549	0.376	0.450	0.557	0.637
ResizableDoubleArray	0.629	0.599	0.609	0.755	0.677
UnixCrypt	0.709	0.702	0.721	0.758	0.717
HoplFormatter	0.695	0.655	0.691	0.872	0.770
avg.	0.736	0.703	0.698	0.813	0.791

## 5 THE PROBLEM OF APPLYING THE ORDERED RELATIONSHIP WRT TO SUBSUMING MUTATION SCORE

In the current literature, it is believed that the subsumed mutants inflate the mutation score and hence may lead to inaccurate measurement of test suite effectiveness in defect detection. In the following, we use an example to show that this is not the case. Considering the following kill matrix, we know that:  $M = \{m1, m2, m3, m4\}$ ,  $M_s = \{m1, m4\}$ .

	t1	t2	t3
m1	1	0	0
m2	1	1	0
m3	1	1	1
m4	0	0	1

Now, given two test sets  $\{t3\}$  and  $\{t2, t3\}$ , which one has a higher ability to detect defects? According to  $MS(M, .)$ , we have:

$$MS(M, \{t3\}) = 2/4 = 0.5 \quad \text{and} \quad MS(M, \{t2, t3\}) = 3/4 = 0.75$$

Therefore, we can conclude that  $\{t2, t3\}$  is more effective than  $\{t3\}$  in detecting defects. This is consistent with the fact:  $\{t2, t3\}$  can detect three mutants (i.e., m2, m3, and m4) while  $\{t3\}$  can only detect two mutants (i.e., m3 and m4). However, according to  $MS(M_s, .)$ , we have:

$$MS(M_s, \{t3\}) = 1/2 = 0.5 \quad \text{and} \quad MS(M_s, \{t2, t3\}) = 1/2 = 0.5$$

This means that  $\{t2, t3\}$  and  $\{t3\}$  have the same effectiveness in detecting defects. Clearly,  $MS(M_s, .)$  twists the test suite effectiveness order between  $\{t3\}$  and  $\{t2, t3\}$ .

Furthermore, if we consider only the test set  $\{t2\}$ , we have:  $MS(M, \{t2\}) = 2/4 = 0.5$  and  $MS(M_s, \{t2\}) = 0$ . According to the subsuming score,  $\{t2\}$  is unable to detect any defect. However, this is not true, as  $\{t2\}$  can detect m2 and m3. Beside this, we use the following code to show the fact:

		t1(a=-3,b=-2)	t2(a=3,b=2)	t3(a=1,b=0)
--	--	---------------	-------------	-------------

original code	<pre>if b!=0:     return a/b else:     return a</pre>	return 1.5	return 1.5	return 1
m1	<pre>if b0:     return a/b else:     return a</pre>	return -3	return 1.5	return 1
m2	<pre>if b!=0:     return a/b else:     return a</pre>	return 6	return 6	return 1
m3	<pre>if b=0:     return a/b else:     return a</pre>	return -3	return 3	error
m4	<pre>if b!=:     return a/b else:     return a</pre>	return 1.5	return 1.5	error

In this example, t3 checks that 0 cannot be used as a divisor, and the other two test cases can be considered to check the function. In this sense, {t2, t3} is more effective than {t3}.

As a result, *the order reflected by the subsumed mutants may be twisted by only subsuming mutants*. Based on the above analysis, we can see that, it is questionable to claim that a more correct order will be obtained after excluding the subsumed mutants. Note that we have been focusing on “order”. As in the above example, m2 and m3 are easy to kill, which means it is easy to inflate the mutation score by them. However, it cannot be denied that they may reflect some unique relationships between test suites.

To this concern, when only comparing the order among a test suite with its subsets, a conservative and plausible way is to use the test effectiveness order produced by the original mutation score as the ground truth when computing OP. Even if we admit that  $MS(M_s, .)$  has a higher ability to predict test completeness compared with  $MS(M, .)$ , this does not mean that  $MS(M_s, .)$  has a higher ability to measure test suite effectiveness. From the measurement theory,  $MS(M_s, .)$  is not a rescaling of  $MS(M, .)$ , as the representation condition is broken. In this sense, removing the subsumed mutants is harmful for computing OP.