Using Mutation Analysis for Assessing and Comparing Test Coverage Criteria

IEEE Transactions on Software Engineering (TSE 2006)

James H. Andrews, Lionel C. Briand, Yvan Labiche, and Akbar Siami Namin

2013-01-10
Dongwon Seo

© KAIST SE LAB 2012
Contents

- Introduction
- Related Work
- Experimental Description
- Analysis Results
- Conclusion
- Discussion
Primary purpose of software testing is to detect software faults as much as possible.

In order to detect many faults, many good test cases are needed.

* What are the criteria of good test cases to detect many faults?
Code coverage

- It is one of the testing techniques to distinguish whether test cases is good.
- It describes the degree (coverage level) to which the source code of a program has been tested by test cases.
- It assumes that if coverage level is high, fault detection probability is also high.
Control flow code coverage
- Block coverage is composed of statement.
- Decision coverage is composed of branch statement.

Data flow code coverage
- C-use coverage is composed of variable defined and computational expression.
- P-use coverage is composed of variable defined and conditional expression.

Subsumption relationship*

Real programs of appropriate size with real faults are hard to find and hard to prepare.

We have to make faulty version.

Problem to make faulty version

- **Manually**
  - Realistic
  - But subjective
  - And too small

- **Automatically**
  - Systematic
  - Repeatable
  - But unrealistic
Mutation Analysis

- Well defined fault-seeding process.
- Imitate programmer’s mistakes (actual faults)
- Generate automatically and systematically variant (mutant) as the result of applying an operator to the original code.

Original code

```java
if (a && b) {
    c = 1;
} else {
    c = 0;
}
```

Mutant

```java
if (a || b) {
    c = 1;
} else {
    c = 0;
}
```
Motivation

- Hand seeded and real faulty version is not enough to compare coverage criteria.
- Generated mutants can be used as faulty versions.
- Our analysis process can be used to compare testing coverage criteria more systematically than previous studies.

Goal

- Providing more systematic analysis to compare testing techniques
- Providing analysis results comparing to coverage criteria
## Related work

<table>
<thead>
<tr>
<th>Goal</th>
<th>Hutchins (ICSE 1994)</th>
<th>Frankl and Lakounenko (ACM SIGSOFT 1998)</th>
<th>This work (TSE 2006)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Comparing to coverage criteria</td>
<td>Comparing to coverage criteria</td>
<td>Comparing to coverage criteria</td>
</tr>
<tr>
<td><strong>Subject program</strong></td>
<td>33 to 66 LOCs</td>
<td>141 to 512 LOCs</td>
<td>6218 LOCs</td>
</tr>
<tr>
<td></td>
<td>6218 LOCs</td>
<td>6218 LOCs</td>
<td></td>
</tr>
<tr>
<td><strong>Faults</strong></td>
<td>7 actual faults</td>
<td>130 hand seeded Faults</td>
<td>33 actual faults</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>34 actual faults</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>736 mutants</td>
</tr>
<tr>
<td><strong>Analysis</strong></td>
<td>Coverage, Test suite size, Fault detection effectiveness</td>
<td>Coverage, Test suite size, Fault detection effectiveness</td>
<td>Coverage, Test suite size, Fault detection effectiveness</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Experiments design

- We investigate the relative cost and effectiveness of the coverage criteria.

Q1. Are mutation scores good predictors of actual fault detection rates?
Q2. What is the cost of achieving given levels of the coverage?

Experiments design

- We investigate the relative cost and effectiveness of the coverage criteria.

Coverage criteria (= coverage level)

Cost (= test suite size)

Effectiveness (= fault detection ratio)

By mutants

By actual faults

Q1

Q2
Experiments design

- We investigate the relative cost and effectiveness of the coverage criteria.

Q3. Can we determine what levels of coverage, for each criteria, should be achieved to obtain reasonable levels of fault detection effectiveness?
Experiments design

- We investigate the relative cost and effectiveness of the coverage criteria.

Q4. What is the relative cost effectiveness of the investigated coverage criteria?
Experiments design

- We investigate the relative cost and effectiveness of the coverage criteria.

Q5. What is the gain of using coverage criteria compared to random test suites?
Experiments design

- We investigate the relative cost and effectiveness of the coverage criteria.

Q6. Do we find a statistically significant relationship between coverage level and fault detection effectiveness when we account for test suite size?
Experiments design

- We investigate the relative **cost** and **effectiveness** of the coverage criteria.

Q7. How is the cost-benefit analysis of coverage criteria affected by variations in fault detection difficulty?
Faulty versions from mutants will be used to compare coverage criteria.
Mutant generation

These four classes of mutation used.
- Replace an integer constant C by 0, 1, -1, ((C)+1, or ((C)-1).
- Replace an arithmetic, relational, logical, bitwise logical, increment/decrement, or arithmetic-assignment operator by another operator from the same class.
- Negate the decision in an if or while statement.
- Delete a statement.

So many mutants (11, 379) generated.
We only use 10th mutant generated. (11,379 -> 1138)
We removed equivalent mutants (1138 -> 736)
- The mutants that were not killed by any test case is referred as equivalent mutants
Experimental Description (4/5)

Test pool

Select test cases by coverage level

13,585 Test cases

Select test cases Randomly

from Previous work data

Coverage test suites for each criterion (Q1~Q7)

Coverage Test suites

5 Test suites that achieved 50.00 ~ 50.99 coverage level

Random Test suites

5 Test suites that achieved 95.00 ~ 95.99 coverage level

Random test suite (Q5)

1700 Test suites with each size from one to 150

Test pool

Select test cases by coverage level

13,585 Test cases

Select test cases Randomly

from Previous work data

Coverage test suites for each criterion (Q1~Q7)

Coverage Test suites

5 Test suites that achieved 50.00 ~ 50.99 coverage level

Random Test suites

5 Test suites that achieved 95.00 ~ 95.99 coverage level

Random test suite (Q5)

1700 Test suites with each size from one to 150

© KAIST SE LAB 2012
Overall experiments approach

Test suites
- Random Test suites
- Coverage Test suites

Test programs
- 736 mutants
- Faulty version 1
- Faulty version...
- Faulty version 736

Testing to detect faults

Cost
(= test suite size)

Coverage criteria
(= coverage level)

Effectiveness
(= fault detection ratio)

Coverage criteria

* For each coverage criteria, (Block, Decision, P-use, D-use) These analyses will be applied.
Q1. Is Am good predictor of Af?

- Am is mutant detection ratio (=mutant score).
- Af is actual fault detection ratio.

Af and AM is both proportional to coverage level.
Q1. Is Am good predictor of Af?

- MRE (Magnitude of Relative Error) is measure for evaluating the accuracy of predictive system.
  - $MRE = \frac{|Af - Am|}{Af}$

- The higher %coverage will make the prediction of Af based on Am more accurate.

- MRE decreases as %coverage increases.
Q1. Is Am good predictor of Af?

Modeling a linear regression between $R^2$ of the

Based on such a model, Am is a unbiased predictor of Af.

Based on such a model, Am is a unbiased predictor of Af.

$R^2 = 0.908$
Q2. What is the cost of achieving given % coverage criteria?

- Modeling exponential regression between Size and %Coverage

Cost of achieving given level of coverage criteria: Block < C-Use < Decision < P-use

$R^2 = 0.943 \sim 0.977$
Q3. Can we determine what % coverage criteria should be achieved to obtain reasonable Am?

- Modeling exponential regression between %coverage and AF

![Graph showing the relationship between Am and %Coverage with R² range from 0.943 to 0.977.]

Am of achieving given level of coverage criteria: Block < C-Use < Decision < P-Use
Q4. What is the relative cost effectiveness of the investigated control and data flow coverage criteria?
- Modeling exponential regression between size and Am

None of the four criteria is more cost-effective than the others.

\[ R^2 = 0.976 \]
Q5. What is the gain of using coverage criteria compared to random test suites (=null criterion)?

- Random test suites are used instead of coverage suites.
- Random test suite means null criterion.

Coverage criteria are more cost-effective than null criterion.

None of the four criteria is more cost-effective than the others.

\[ R^2 = 0.96 \text{ (random)} \]
Q6. Do we still find a statistically significant relationship between coverage level and fault detection effectiveness when we account for test suite size?

- Modeling multiple regression between Am and two covariates (%coverage, size)
Q6. Do we still find a statistically significant relationship between coverage level and fault detection effectiveness when we account for test suite size?

PreAm is Am predictor obtained from between size and %coverage.

The both size and coverage play a complementary role in explaining fault detection.

$R^2 = 0.99$
Q7. How is the cost-benefit analysis of coverage criteria affected by variations in fault detection difficulty?

- we focus on two subset of mutants.

- Mutants with 5% detection probability (hard)
- Mutants with 1.5% detection probability (very hard)

736 Mutants
Q7. How is the cost-benefit analysis of coverage criteria affected by variations in fault detection difficulty?

- Nearly linear relationship in “Hard” mutants
- Previous study reported relationship between Am and size as linear relationship
- But it is because they remove fault to be detect easily through hand seeding
Q7. How is the cost-benefit analysis of coverage criteria affected by variations in fault detection difficulty?

- In Last 10~20%, All and others have some different exponential relationship.
- Previous study reported relationship between Am and coverage as extreme exponential relationship.
- But it is because they remove fault to be detect easily through hand seeding.
Conclusion

Contribution

- Introducing the feasibility of using mutation analysis

- Applying mutation analysis to fundamental questions regarding the relationships between fault detection, test suite size, and control/data flow coverage.

- Showing a way to tune the mutation analysis process to possible differences in fault detection probabilities in a specific environment.
Discussion

❖ Pros
   ▪ It provides detail analysis from many experiments.
   ▪ It provides results of previous studies in order to justify their experiments.

❖ Cons
   ▪ The number of actual faults is small.
   ▪ It uses only test suite size to assess cost.
Thank You.
Appendix

Control flow code coverage

```c
if(c==' ' || c == '\n' || c == '\t')
state = OUT;
else if (state == OUT) {
state = IN;
++nw;
}
*p_n1 = n1;
*p_nw = nw;
*p_nc = nc;

Block coverage

state = OUT;
n1 = nw = nc = 0;
while(EOF != (c = getc(file))){
    ++nc;
    if( c=='\n')
        ++nl;
    TRUE    if(c==' ' || c == '\n' || c == '\t')
    state = OUT;
    else if (state == OUT) {
        state = IN;
        ++nw;
    }
    *p_n1 = n1;
}
state = OUT;
n1 = nw = nc = 0;
while(EOF != (c = getc(file))){
    ++nc;
    if( c=='
')
        ++nl;
    FALSE   if(c==' ' || c == '\n' || c == '\t')
    state = OUT;
    else if (state == OUT) {
        state = IN;
        ++nw;
    }
```

Decision coverage
Appendix

Data flow code coverage

```c
if(c==' ' || c == '\n' || c == '\t')
state = OUT;
else if (state == OUT) {
    state = IN;
    ++nw;
}
*p_nl = nl;
*p_nw = nw;
*p_nc = nc;
```

C-Use of variable “nw”

```c
Def state = OUT;
nl = nw = nc = 0;
while(EOF != (c = getc(file))){
    ++nc;
    if( c== '\n')
        ++nl;
    if(c==' ' || c == '\n' || c == '\t')
        state = OUT;
TRUE else if (state == OUT) {
    state = IN;
    ++nw;
}
```

P-Use of variable “state”

```c
Def state = OUT;
nl = nw = nc = 0;
while(EOF != (c = getc(file))){
    ++nc;
    if( c== '\n')
        ++nl;
    if(c==' ' || c == '\n' || c == '\t')
        state = OUT;
FALSE else if (state == OUT) {
    state = IN;
    ++nw;
}
```