Reducing the Cost of Regression Testing by Identifying Irreplaceable Test Cases

Chu-Ti Lin, Dept. of Computer Sci. and Info. Eng., National Chiayi University, Taiwan
Kai-Wei Tang, Cloud System Software Institute, Institute for Information Industry, Taiwan
Cheng-Ding Chen, Dept. of Computer Sci. and Info. Eng., National Chiayi University, Taiwan
Gregory M. Kapfhammer, Dept. of Computer Science, Allegheny College, Meadville, PA

August 28, 2012
The Sixth International Conference on Genetic and Evolutionary Computing
Outline

- Introduction
- Related work
- Reducing the execution cost of a test suite
- Experimental analysis
- Conclusion
Introduction: Software Testing

- **Software testing**
  - To detect and isolate defects while implementing software systems.

- **Test case**
  - A set of input data and expected output results which are designed to exercise a specific software function or test requirement.

<table>
<thead>
<tr>
<th>Test case</th>
<th>Test requirement</th>
<th>$r_1$</th>
<th>$r_2$</th>
<th>$r_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td></td>
<td>●</td>
<td>●</td>
<td></td>
</tr>
<tr>
<td>$t_2$</td>
<td></td>
<td></td>
<td></td>
<td>●</td>
</tr>
<tr>
<td>$t_3$</td>
<td></td>
<td></td>
<td>●</td>
<td></td>
</tr>
<tr>
<td>$t_4$</td>
<td></td>
<td>●</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Introduction: Test Suite

- It is difficult for a single test case to satisfy all of the specified test requirements.
- A considerable number of test cases are usually generated and collected in a test suite.

<table>
<thead>
<tr>
<th>Test case</th>
<th>Test requirement</th>
<th>$r_1$</th>
<th>...</th>
<th>$r_{3000}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td></td>
<td>●</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$t_2$</td>
<td></td>
<td></td>
<td></td>
<td>●</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$t_?$</td>
<td></td>
<td>●</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Introduction: Regression Testing

• In an attempt to ensure both the correctness of new code and its proper integration into the system, all test case in test suite $T$ should be executed.
Introduction: Test Suite Reduction

- To remove the redundant test cases while still ensuring that all test requirements are satisfied.

<table>
<thead>
<tr>
<th>Test</th>
<th>$r_1$</th>
<th>$r_2$</th>
<th>$r_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>●</td>
<td>●</td>
<td></td>
</tr>
<tr>
<td>$t_2$</td>
<td></td>
<td></td>
<td>●</td>
</tr>
<tr>
<td>$t_3$</td>
<td></td>
<td>●</td>
<td></td>
</tr>
<tr>
<td>$t_4$</td>
<td>●</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Greedy Algorithm

• A commonly-used method for finding the near-optimal solution to the test suite reduction problem.
• It repeatedly removes the test $t$ that has the maximum $\text{Coverage}(t)$ from $T$ to $RS$ until all of the requirements are covered.
  • $\text{Coverage}(t)$ is the number of uncovered test requirements satisfied by test case $t$. 
Greedy-based Algorithms

• Many test suite reduction algorithms are developed based on Coverage metric.
  • HGS algorithm proposed by Harrold et al. [4]
  • GE and GRE proposed by Chen and Lau [10]
Reduction Using Greedy Algorithm

<table>
<thead>
<tr>
<th>Test</th>
<th>Cost</th>
<th>$r_1$</th>
<th>$r_2$</th>
<th>$r_3$</th>
<th>Coverage($t$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>6</td>
<td>●</td>
<td>●</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>$t_2$</td>
<td>2</td>
<td></td>
<td></td>
<td>●</td>
<td>1</td>
</tr>
<tr>
<td>$t_3$</td>
<td>1</td>
<td></td>
<td>●</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>$t_4$</td>
<td>3</td>
<td></td>
<td></td>
<td>●</td>
<td>1</td>
</tr>
</tbody>
</table>

Greedy: $RS=\{t_1, t_2\}$, total cost = 8

Optimal solution: $RS=\{t_2, t_3, t_4\}$, total cost = 6
Reduction with Ratio

- Ma et al. [11] and Smith and Kapfhammer [12] evaluated the test cases using

\[
\text{Ratio}(t) = \frac{\text{Coverage}(t)}{\text{Cost}(t)}
\]

where \( \text{Cost}(t) \) represents the execution cost of the test case \( t \).

- It aims to reduce the cost of running a test suite.
Reduction with Ratio

<table>
<thead>
<tr>
<th>Test</th>
<th>Cost</th>
<th>$r_1$</th>
<th>$r_2$</th>
<th>$r_3$</th>
<th>$Ratio(t)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>6</td>
<td>●</td>
<td>●</td>
<td></td>
<td>0.67</td>
</tr>
<tr>
<td>$t_2$</td>
<td>2</td>
<td></td>
<td></td>
<td>●</td>
<td>0.5</td>
</tr>
<tr>
<td>$t_3$</td>
<td>1</td>
<td></td>
<td></td>
<td>●</td>
<td>1</td>
</tr>
<tr>
<td>$t_4$</td>
<td>3</td>
<td>●</td>
<td></td>
<td></td>
<td>0.33</td>
</tr>
</tbody>
</table>
# Reduction with Ratio

<table>
<thead>
<tr>
<th>Test</th>
<th>Cost</th>
<th>$r_1$</th>
<th>$r_2$</th>
<th>$r_3$</th>
<th>$Ratio(t)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>6</td>
<td>●</td>
<td>-</td>
<td></td>
<td>0.17</td>
</tr>
<tr>
<td>$t_2$</td>
<td>2</td>
<td>-</td>
<td></td>
<td>●</td>
<td>0.5</td>
</tr>
<tr>
<td>$t_3$</td>
<td>1</td>
<td>-</td>
<td></td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>$t_4$</td>
<td>3</td>
<td>●</td>
<td>-</td>
<td></td>
<td>0.33</td>
</tr>
</tbody>
</table>
### Reduction with Ratio

<table>
<thead>
<tr>
<th>Test</th>
<th>Cost</th>
<th>$r_1$</th>
<th>$r_2$</th>
<th>$r_3$</th>
<th>Ratio($t$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>6</td>
<td>●</td>
<td>-</td>
<td>-</td>
<td>0.17</td>
</tr>
<tr>
<td>$t_2$</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$t_3$</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$t_4$</td>
<td>3</td>
<td>●</td>
<td>-</td>
<td>-</td>
<td>0.33</td>
</tr>
</tbody>
</table>
Reduction with Ratio

<table>
<thead>
<tr>
<th>Test</th>
<th>Cost</th>
<th>$r_1$</th>
<th>$r_2$</th>
<th>$r_3$</th>
<th>$\text{Ratio}(t)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>6</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>$t_2$</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$t_3$</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$t_4$</td>
<td>3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

$\text{Greedy}_{\text{WithRatio}} : RS = \{t_2, t_3, t_4\}, \text{ total cost} = 6$
## ReduceWithRatio Problems

<table>
<thead>
<tr>
<th>Test</th>
<th>Cost</th>
<th>$r_1$</th>
<th>$r_2$</th>
<th>$r_3$</th>
<th>$r_4$</th>
<th>$r_5$</th>
<th>$r_6$</th>
<th>$\text{Ratio}(t)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>4</td>
<td>⬤</td>
<td>⬤</td>
<td>⬤</td>
<td></td>
<td></td>
<td></td>
<td>0.75</td>
</tr>
<tr>
<td>$t_2$</td>
<td>7</td>
<td></td>
<td>⬤</td>
<td>⬤</td>
<td>⬤</td>
<td>⬤</td>
<td>⬤</td>
<td>0.57</td>
</tr>
<tr>
<td>$t_3$</td>
<td>3</td>
<td>⬤</td>
<td></td>
<td></td>
<td></td>
<td>⬤</td>
<td>⬤</td>
<td>0.67</td>
</tr>
<tr>
<td>$t_4$</td>
<td>4</td>
<td></td>
<td>⬤</td>
<td></td>
<td></td>
<td>⬤</td>
<td>⬤</td>
<td>0.50</td>
</tr>
</tbody>
</table>
## Problem of ReduceWithRatio

<table>
<thead>
<tr>
<th>Test</th>
<th>Cost</th>
<th>$r_1$</th>
<th>$r_2$</th>
<th>$r_3$</th>
<th>$r_4$</th>
<th>$r_5$</th>
<th>$r_6$</th>
<th>Ratio($t$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$t_2$</td>
<td>7</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>•</td>
<td>•</td>
<td></td>
<td>0.29</td>
</tr>
<tr>
<td>$t_3$</td>
<td>3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td>•</td>
<td>0.33</td>
</tr>
<tr>
<td>$t_4$</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td>•</td>
<td>0.25</td>
</tr>
</tbody>
</table>
## Problem of ReduceWithRatio

<table>
<thead>
<tr>
<th>Test</th>
<th>Cost</th>
<th>$r_1$</th>
<th>$r_2$</th>
<th>$r_3$</th>
<th>$r_4$</th>
<th>$r_5$</th>
<th>$r_6$</th>
<th>Ratio($t$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$t_2$</td>
<td>7</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>$\bullet$</td>
<td>$\bullet$</td>
<td>-</td>
<td>0.29</td>
</tr>
<tr>
<td>$t_3$</td>
<td>3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$t_4$</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
</tr>
</tbody>
</table>
Problem of ReduceWithRatio

<table>
<thead>
<tr>
<th>Test</th>
<th>Cost</th>
<th>$r_1$</th>
<th>$r_2$</th>
<th>$r_3$</th>
<th>$r_4$</th>
<th>$r_5$</th>
<th>$r_6$</th>
<th>Ratio(t)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$t_2$</td>
<td>7</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$t_3$</td>
<td>3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$t_4$</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
</tr>
</tbody>
</table>

GreedyWithRatio : $RS = \{t_1, t_2, t_3\}$, total cost = 14
Problem of ReduceWithRatio

$t_1$ is replaceable

<table>
<thead>
<tr>
<th>Test</th>
<th>Cost</th>
<th>$r_1$</th>
<th>$r_2$</th>
<th>$r_3$</th>
<th>$r_4$</th>
<th>$r_5$</th>
<th>$r_6$</th>
<th>Ratio(t)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>4</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
<td></td>
<td></td>
<td>0.75</td>
</tr>
<tr>
<td>$t_2$</td>
<td>7</td>
<td></td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
<td>0.57</td>
</tr>
<tr>
<td>$t_3$</td>
<td>3</td>
<td>•</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>•</td>
<td>0.67</td>
</tr>
<tr>
<td>$t_4$</td>
<td>4</td>
<td></td>
<td></td>
<td>•</td>
<td></td>
<td></td>
<td>•</td>
<td>0.50</td>
</tr>
</tbody>
</table>

GreedyWithRatio : $RS=\{ t_1, t_2, t_3 \}$, total cost = 14

Optimal solution : $RS=\{ t_2, t_3 \}$, total cost = 10
Reduction Using Irreplaceability

• Concept:
  • Evaluating a test case by identifying if it is replaceable.
  • We posit that $t$ has a higher replaceability with respect to $r$ in this case
    • That is, $t$ has a lower irrereplaceability with respect to $r$. 
Evaluating the Irreplaceability

- The irreplaceability of \( t \) with respect to the requirement \( R = \{ r_1, r_2, r_3, \ldots, r_m \} \) can be defined as

\[
\text{Irreplaceability}(t) = \frac{\sum_{i=1}^{m} \text{Contribution}(t, r_i)}{\text{Cost}(t)}
\]

where

\[
\text{Contribution}(t, r_i) = \begin{cases} 
0, & \text{if } t \text{ cannot satisfy } r_i \\
\frac{1}{\text{the number of test cases that satisfy } r_i}, & \text{if } t \text{ satisfies } r_i
\end{cases}
\]
Reduction with Irreplaceability

<table>
<thead>
<tr>
<th>Test</th>
<th>Cost</th>
<th>$r_1$</th>
<th>$r_2$</th>
<th>$r_3$</th>
<th>$r_4$</th>
<th>$r_5$</th>
<th>$r_6$</th>
<th>Irreplaceability($t$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>4</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
<td></td>
<td></td>
<td>0.33</td>
</tr>
<tr>
<td>$t_2$</td>
<td>7</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
<td>0.40</td>
</tr>
<tr>
<td>$t_3$</td>
<td>3</td>
<td>•</td>
<td></td>
<td></td>
<td>•</td>
<td>•</td>
<td></td>
<td>0.33</td>
</tr>
<tr>
<td>$t_4$</td>
<td>4</td>
<td>•</td>
<td>•</td>
<td></td>
<td>•</td>
<td>•</td>
<td></td>
<td>0.21</td>
</tr>
</tbody>
</table>
## Reduction with Irreplaceability

<table>
<thead>
<tr>
<th>Test</th>
<th>Cost</th>
<th>( r_1 )</th>
<th>( r_2 )</th>
<th>( r_3 )</th>
<th>( r_4 )</th>
<th>( r_5 )</th>
<th>( r_6 )</th>
<th>Irreplaceability((t))</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_1 )</td>
<td>4</td>
<td>( \bullet )</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.13</td>
</tr>
<tr>
<td>( t_2 )</td>
<td>7</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>( t_3 )</td>
<td>3</td>
<td>( \bullet )</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.33</td>
</tr>
<tr>
<td>( t_4 )</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>( \bullet )</td>
<td>0.13</td>
</tr>
</tbody>
</table>
Reduction with Irreplaceability

<table>
<thead>
<tr>
<th>Test</th>
<th>Cost</th>
<th>$r_1$</th>
<th>$r_2$</th>
<th>$r_3$</th>
<th>$r_4$</th>
<th>$r_5$</th>
<th>$r_6$</th>
<th>Irreplaceability($t$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>$t_2$</td>
<td>7</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$t_3$</td>
<td>3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$t_4$</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
</tr>
</tbody>
</table>

**GreedyWithIrreplaceability**: $RS= \{ t_2, t_3 \}$, total cost = 10

**Optimal solution**: $RS=\{ t_2, t_3 \}$, total cost = 10

**GreedyWithRatio**: $RS=\{t_1, t_2, t_3\}$, total cost = 14
Experimental Data Set

- The Siemens suite of programs from the SIR are frequently chosen benchmarks for evaluating test suite reduction methods [15].

<table>
<thead>
<tr>
<th>Program</th>
<th>Test pool</th>
<th>Test requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>printtokens</td>
<td>4,130</td>
<td>140</td>
</tr>
<tr>
<td>printtokens2</td>
<td>4,115</td>
<td>138</td>
</tr>
<tr>
<td>replace</td>
<td>5,542</td>
<td>126</td>
</tr>
<tr>
<td>schedule</td>
<td>2,650</td>
<td>46</td>
</tr>
<tr>
<td>schedule2</td>
<td>2,710</td>
<td>72</td>
</tr>
<tr>
<td>tcas</td>
<td>1,608</td>
<td>16</td>
</tr>
<tr>
<td>totinfo</td>
<td>1,052</td>
<td>44</td>
</tr>
</tbody>
</table>
Randomly pick \( n \) test cases (\( 1 \leq n \leq 0.5 \times loc \)) from the test pool, and include them in \( T \).

Can \( T \) satisfy all test requirements?

- Yes: Return \( T \).
- No: Randomly choose one more test case \( t \).

Can \( t \) satisfy any unsatisfied requirements?

- Yes: Include the test case \( t \) in \( T \).
- No: Repeat the process.
Evaluating the Reduction Capability

- **Criterion**

  \[
  SCR(T, RS) = \frac{Cost(T) - Cost(RS)}{Cost(T)} \times 100\%
  \]

  where

  - \(Cost(T)\): the cost required to execute the original test suite \(T\);
  - \(Cost(RS)\): the cost associated with running the representative set \(RS\).
## Experiment Result

<table>
<thead>
<tr>
<th>Test Suite Program</th>
<th>Original</th>
<th>( \text{RS}_{\text{Greedy}} )</th>
<th>( \text{RS}_{\text{WithRatio}} )</th>
<th>( \text{RS}_{\text{WithIrrereplaceability}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cost*</td>
<td>Cost*</td>
<td>Cost*</td>
<td>Cost*</td>
</tr>
<tr>
<td>Printtokens</td>
<td>914.67</td>
<td>117.32</td>
<td>115.04</td>
<td>81.73</td>
</tr>
<tr>
<td>printtokens2</td>
<td>717.84</td>
<td>58.29</td>
<td>56.19</td>
<td>48.53</td>
</tr>
<tr>
<td>Replace</td>
<td>1068.90</td>
<td>88.28</td>
<td>81.06</td>
<td>76.06</td>
</tr>
<tr>
<td>Schedule</td>
<td>493.77</td>
<td>18.71</td>
<td>16.35</td>
<td>15.32</td>
</tr>
<tr>
<td>schedule2</td>
<td>651.82</td>
<td>40.14</td>
<td>28.60</td>
<td>26.80</td>
</tr>
<tr>
<td>Tcas</td>
<td>219.39</td>
<td>23.74</td>
<td>21.53</td>
<td>20.74</td>
</tr>
<tr>
<td>Totinfo</td>
<td>690.97</td>
<td>52.15</td>
<td>26.43</td>
<td>26.14</td>
</tr>
</tbody>
</table>

*The cost is measured in millisecond (ms).

- Both \( \text{ReduceWithIrrereplaceability} \) and \( \text{ReduceWithRatio} \) exhibit excellent cost reduction capabilities.
- The SCR scores of \( \text{ReduceWithRatio} \) are not as good as those of \( \text{ReduceWithIrrereplaceability} \).
Summary of Contribution

• Key motivators
  • Most existing test suite reduction algorithms attempt to minimize the size of a regression test suite.
  • Reduction using Ratio metric does not always perform in a satisfactory manner.

• Method
  • Evaluating a test case by identifying if it is replaceable.
  • It repeatedly picks the test \( t \) that has the maximum Irreplaceability (\( t \)).
Summary of Contribution

• Empirical studies reveals that
  • Reduction using Irreplaceability is the best method for decreasing the cost of test suite execution.
Future Work

- Improving Greedy by Irreplaceability
- Improving HGS and GRE by Irreplaceability
- Empirical studies on Siemens programs
- Empirical studies on larger programs