Parameter Tuning for Search-Based Test-Data Generation Revisited
Support for Previous Results

Anton Kotelyanskii
Gregory M. Kapfhammer

creative commons licensed ( BY-NC-ND ) flickr photo shared by sunface13
Software Testing
Software Testing

Test Suites
Software Testing

Test Suites
Automatic Generation
Software Testing

Test Suites
Automatic Generation
Confronting Challenges
Software Testing

Test Suites
Automatic Generation
Confronting Challenges
Evaluation Strategies
Empirical Studies
Empirical Studies

Challenges
Empirical Studies

Challenges
Importance
Empirical Studies

Challenges
Importance
Replication
Empirical Studies

Challenges
Importance
Replication
Rarity
EvoSuite

Amazing test suite generator

creative commons licensed (BY-SA) flickr photo shared by mcclanahoochie
EvoSuite

Amazing test suite generator
Uses a genetic algorithm

creative commons licensed ( BY-SA ) flickr photo shared by mcclanahoochie
EvoSuite

Amazing test suite generator
Uses a genetic algorithm
*Input*: A Java class

creative commons licensed (BY-SA) flickr photo shared by mcclanahoochie
EvoSuite

Amazing test suite generator
Uses a genetic algorithm

*Input:* A Java class
*Output:* A JUnit test suite

creative commons licensed (BY-SA) flickr photo shared by mcclanahoochie
EvoSuite

Amazing test suite generator
Uses a genetic algorithm
Input: A Java class
Output: A JUnit test suite
http://www.evosuite.org/

creative commons licensed ( BY-SA ) flickr photo shared by mcclanahoochie
Parameter Tuning
Parameter Tuning

RSM: Response surface methodology
Parameter Tuning

*RSM*: Response surface methodology

*SPOT*: Sequential parameter optimization toolbox
Parameter Tuning

*RSM*: Response surface methodology

*SPOT*: Sequential parameter optimization toolbox

Successfully applied to many diverse problems!
Defaults or Tuned Values?
Experiment Design

Eight EvoSuite parameters

creative commons licensed ( BY-NC ) flickr photo shared by Michael Kappel
Experiment Design

Eight EvoSuite parameters
Ten projects from SF100

creative commons licensed (BY-NC) flickr photo shared by Michael Kappel
Experiment Design

Eight EvoSuite parameters
Ten projects from SF100
475 Java classes for subjects

creative commons licensed (BY-NC) flickr photo shared by Michael Kappel
Experiment Design

Eight EvoSuite parameters
Ten projects from SF100
475 Java classes for subjects
100 trials after parameter tuning

creative commons licensed (BY-NC) flickr photo shared by Michael Kappel
Experiment Design

Eight EvoSuite parameters
Ten projects from SF100
475 Java classes for subjects
100 trials after parameter tuning
Aiming to improve statement coverage

creative commons licensed (BY-NC) flickr photo shared by Michael Kappel
<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population Size</td>
<td>5</td>
<td>99</td>
</tr>
<tr>
<td>Chromosome Length</td>
<td>5</td>
<td>99</td>
</tr>
<tr>
<td>Rank Bias</td>
<td>1.01</td>
<td>1.99</td>
</tr>
<tr>
<td>Number of Mutations</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>Max Initial Test Count</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>Crossover Rate</td>
<td>0.01</td>
<td>0.99</td>
</tr>
<tr>
<td>Constant Pool Use Probability</td>
<td>0.01</td>
<td>0.99</td>
</tr>
<tr>
<td>Test Insertion Probability</td>
<td>0.01</td>
<td>0.99</td>
</tr>
</tbody>
</table>
Experiments
Experiments

184 days of computation time estimated
Experiments

184 days of computation time estimated
Cluster of 70 computers running for weeks
Experiments

184 days of computation time estimated
Cluster of 70 computers running for weeks
Identified 139 "easy" and 21 "hard" classes
Experiments

184 days of computation time estimated
Cluster of 70 computers running for weeks
Identified 139 "easy" and 21 "hard" classes
Mann–Whitney U–test and
Experiments

184 days of computation time estimated
Cluster of 70 computers running for weeks
Identified 139 "easy" and 21 "hard" classes
Mann-Whitney U-test and
Vargha-Delaney effect size
<table>
<thead>
<tr>
<th>Category</th>
<th>Effect Size</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Results Across Trials and Classes</td>
<td>0.5029</td>
<td>0.1045</td>
</tr>
<tr>
<td>No &quot;Easy&quot; and &quot;Hard&quot; Classes</td>
<td>0.5048</td>
<td>0.0314</td>
</tr>
</tbody>
</table>
## Results

Using *lower-is-better* inverse statement coverage

<table>
<thead>
<tr>
<th>Category</th>
<th>Effect Size</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Results Across Trials and Classes</td>
<td>0.5029</td>
<td>0.1045</td>
</tr>
<tr>
<td>No &quot;Easy&quot; and &quot;Hard&quot; Classes</td>
<td>0.5048</td>
<td>0.0314</td>
</tr>
</tbody>
</table>
**Results**

<table>
<thead>
<tr>
<th>Category</th>
<th>Effect Size</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Results Across Trials and Classes</td>
<td>0.5029</td>
<td>0.1045</td>
</tr>
<tr>
<td>No &quot;Easy&quot; and &quot;Hard&quot; Classes</td>
<td>0.5048</td>
<td>0.0314</td>
</tr>
</tbody>
</table>

Using *lower-is-better* inverse statement coverage
Effect size *greater* than 0.5 means that tuning is *worse*
## Results

<table>
<thead>
<tr>
<th>Category</th>
<th>Effect Size</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Results Across Trials and Classes</td>
<td>0.5029</td>
<td>0.1045</td>
</tr>
<tr>
<td>No &quot;Easy&quot; and &quot;Hard&quot; Classes</td>
<td>0.5048</td>
<td>0.0314</td>
</tr>
</tbody>
</table>

Using *lower-is-better* inverse statement coverage. Effect size *greater* than 0.5 means that tuning is *worse*. Testing shows we do not *always* reject the null hypothesis.
Results

<table>
<thead>
<tr>
<th>Category</th>
<th>Effect Size</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Results Across Trials and Classes</td>
<td>0.5029</td>
<td>0.1045</td>
</tr>
<tr>
<td>No &quot;Easy&quot; and &quot;Hard&quot; Classes</td>
<td>0.5048</td>
<td>0.0314</td>
</tr>
</tbody>
</table>

Using *lower-is-better* inverse statement coverage. Effect size *greater* than 0.5 means that tuning is *worse*. Testing shows we do not *always* reject the null hypothesis. Additional empirical results in the QSIC 2014 paper!
Discussion
Discussion

Tuning improved scores for 11 classes

creative commons licensed (BY) photo shared by Startup Stock Photos
Discussion

Tuning improved scores for 11 classes
Otherwise, same as or worse than defaults
Discussion

Tuning improved scores for 11 classes
Otherwise, same as or worse than defaults
A "soft floor" may exist for parameter tuning

Creative commons licensed (BY) photo shared by Startup Stock Photos
Discussion

Tuning improved scores for 11 classes
Otherwise, same as or worse than defaults
A "soft floor" may exist for parameter tuning
Additional details in the QSIC 2014 paper!
Practical Implications
Practical Implications

Fundamental Challenges
Practical Implications

Fundamental Challenges
Tremendous Confidence
Practical Implications

Fundamental Challenges
Tremendous Confidence
Great Opportunities
Important Contributions
Important Contributions

Comprehensive Experiments

creative commons licensed ( BY-NC-ND ) flickr photo shared by sunface13
Important Contributions

Comprehensive Experiments
Conclusive Confirmation

creative commons licensed ( BY-NC-ND ) flickr photo shared by sunface13
Important Contributions

Comprehensive Experiments
Conclusive Confirmation
For EvoSuite, *Defaults = Tuned*

creative commons licensed (BY-NC-ND) flickr photo shared by sunface13