Search Based Big Data Android App 
Energy Genetic Improvement in the Cloud 

Karl Spearman, Charles Pearson & Hermann Mark 
UCL, CRUST Center, London, Great Britain.
Search Based Big Data Android App Energy Genetic Improvement in the Cloud

Karl Spearman, Charles Pearson & Hermann Mark
UCL, CRUST Center, Londen, Grate Britton.
Results

• Comparing Algorithms Advocated and Base-of-the-art.
• Lower is better (in our study).
## Results

- Comparing Algorithms Advocated and Base-of-the-art.
- Lower is better (in our study).

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>6.02</td>
<td>0.0</td>
<td>0</td>
<td>0.02</td>
<td>0.04</td>
<td>0.05</td>
<td>0.01</td>
<td>0.03</td>
<td>15</td>
<td>9.04</td>
</tr>
<tr>
<td>B</td>
<td>0.05</td>
<td>0.0</td>
<td>2</td>
<td>0.09</td>
<td>0.08</td>
<td>0.06</td>
<td>0.05</td>
<td>0.09</td>
<td>0.09</td>
<td>0.08</td>
</tr>
</tbody>
</table>
Results

- Comparing Algorithms Advocated and Base-of-the-art.
- Lower is better (in our study).

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>6.02</td>
<td>0.0</td>
<td>0</td>
<td>0.02</td>
<td>0.04</td>
<td>0.05</td>
<td>0.01</td>
<td>0.03</td>
<td>15</td>
<td>9.04</td>
</tr>
<tr>
<td>B</td>
<td>0.05</td>
<td>0.0</td>
<td>0</td>
<td>0.09</td>
<td>0.08</td>
<td>0.06</td>
<td>0.05</td>
<td>0.09</td>
<td>0.09</td>
<td>0.08</td>
</tr>
</tbody>
</table>

- Effect size is 0.315. **Success!**
- **AND SO.** Our Algorithm A performs better!
Results

- Comparing Algorithms Advocated and Base-of-the-art.
- Lower is better (in our study).

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>6.02</td>
<td>0.0</td>
<td>0</td>
<td>0.02</td>
<td>0.04</td>
<td>0.05</td>
<td>0.01</td>
<td>0.03</td>
<td>15</td>
<td>9.04</td>
</tr>
<tr>
<td>B</td>
<td>0.05</td>
<td>0.0</td>
<td>2</td>
<td>0.09</td>
<td>0.08</td>
<td>0.06</td>
<td>0.05</td>
<td>0.09</td>
<td>0.09</td>
<td>0.08</td>
</tr>
</tbody>
</table>

- Effect size is 0.315. *Success!*
- *AND SO.* Our Algorithm A performs better!
This work introduces

- Guidelines for ensuring that the Vargha Delaney effect size test tells us whether results are usefully better, not just whether they are better.

- Specific to a Search Based Software Engineering context.
Effect Size Testing

- For comparing randomized sets of results (common in SBSE).
- Hypothesis testing indicates whether a difference is significant.
- **Effect size testing indicates how big the difference is.**
Vargha Delaney A test for effect size

Calculates $A_{12}$, the probability that a randomly chosen value from group 1 is higher than one from group 2:

$$A_{12} = \text{Prob}(X_1 > X_2) + 0.5\text{Prob}(X_1 = X_2)$$
The Problem

• A difference that is small enough to be irrelevant is counted the same as a large difference.

• So if solution A wins by an insignificant margin in 70% of cases and solution B wins by a significant margin in 30% solution A wins, even if its benefits are of no practical use.
The solution is to ensure that only meaningful differences are considered.
Transform data to be **meaningful**
Well-known statistical approach
... yet not often done in SBSE
Two Approaches

- **Pre Transformed Data (PTD)**

  The simplest to implement - just modify the data beforehand.

- **Modified Comparison Function (MCF)**

  Some things can't be done through PTD and so instead modify the comparison function.

  \[ A_{12} = P(X_1 > X_2) + 0.5P(X_1 = X_2) \]
How this may be implemented: SBSE examples.

- Implementation differences: Only a speed up greater than could be achieved by parallelization or different hardware is counted.
- Moore's law: An improvement needs to be at least double its competition.
- Delays: Delays of less than 10 seconds are ignored. Overnight delays are all counted as identical.
Uses for MCF

• With decomposable fitness functions solutions can be compared using every part of the fitness function:
  • Pareto dominance.
  • Has to improve in $n$ instances.
  • Comparing growth function across instances.
  • And much more...
The Results

The same example as before BUT

All results under 0.1 are now discounted as "meaningless"

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>6.02</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>15</td>
<td>9.04</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The effect size is now 0.615

Algorithm B now wins
Final word

This can bridge the gap between saying results are "better" and saying they are "meaningfully better".

BUT:

• Caution advised.

• Standards across the research community or agreed with clients should be reached.
References


Any questions?