REPRESENTATIONS AND OPERATORS FOR IMPROVING EVOLUTIONARY SOFTWARE REPAIR

Claire Le Goues
Westley Weimer
Stephanie Forrest

http://genprog.cs.virginia.edu
“Everyday, almost 300 bugs appear […] far too many for only the Mozilla programmers to handle.”

— Mozilla Developer, 2005

PROBLEM: BUGGY SOFTWARE

Annual cost of software errors in the US: $59.5 billion (0.6% of GDP).

Average time to fix a security-critical error: 28 days.

10%: Everything Else

90%: Maintenance
APPROACH: EVOLUTIONARY COMPUTATION
Input: source code, specification

Genetic Programming Search

Output: repaired version of the program

http://genprog.cs.virginia.edu
SEARCH SPACE
OTHER GP PROBLEMS

GECCO GP-TRACK
BEST PAPERS

Learning: expression trees or lists
Population: 64 – 2500
Iterations: 50 – 10000
Max variant size: 16 operations, 48 operations, 17 levels, 11 levels

PROGRAM REPAIR

Learning: patches or repaired programs
Population: 40
Iterations: 10
Max variant size: unbounded
Largest benchmark program: 2.8 million lines of C code.
EC-based repair starts with a large genome.

SEARCH SPACE

The starting individual is mostly correct.
Input: source code, specification

Genetic Programming Search

Output: repaired version of the program

http://genprog.cs.virginia.edu
OUR GOAL

IN-DEPTH STUDY OF REPRESENTATION AND OPERATORS FOR EVOLUTIONARY PROGRAM REPAIR.
OUR GOAL

IN-DEPTH STUDY OF REPRESENTATION AND OPERATORS FOR EVOLUTIONARY PROGRAM REPAIR.
INPUT

EVALUATE FITNESS

DISCARD

ACCEPT

CROSSOVER, MUTATE, SELECT

OUTPUT
INPUT

EVALUATE FITNESS

CROSSOVER, MUTATE, SELECT

DISCARD

ACCEPT

OUTPUT
**REPRESENTATION**

**Input:**

- Diagram 1
- Diagram 2
- Diagram 3

**AST/WP:**

1. 2
2. 4
3. 5

**Patch:**

- Delete(3)
- Replace(5,1)
- Insert(5,4)
- Insert(3,3)
- Delete(4)
- Replace(3,5)
Mutation operators

- Manipulate only existing genetic material.
- Semantic checking improves probability that mutation is viable.

Crossover:

- One-point: on the weighted path or edit list.
- Patch-subset: uniform, on the edit list.
Mutation operators

- Manipulate only existing genetic material.
- Semantic checking improves probability that mutation is viable.

Crossover:

- One-point: on the weighted path or edit list.
- Patch-subset: uniform, on the edit list.

Aside: mutation operator selection matters!
Input: [✓✓✓✗]

Legend:
- Likely faulty.
- Maybe faulty.
- Not faulty.

http://genprog.cs.virginia.edu
Input: [✓✓✓✓✓]

Legend:
- Red: High change probability.
- Yellow: Low change probability.
- Green: Not changed.

http://genprog.cs.virginia.edu
Input: [✓ ✔ ✔ ×]

Default: 10 : 1 ratio

Legend:
- High change probability.
- Low change probability.
- Not changed.

http://genprog.cs.virginia.edu
OUR GOAL

IN-DEPTH STUDY OF REPRESENTATION AND OPERATORS FOR EVOLUTIONARY PROGRAM REPAIR.
Benchmarks: 105 bugs in 8 real-world programs.\(^1\)

- 5 million lines of C code, 10,000 test cases.
- Bugs correspond to human-written repairs for regression test failures.

Default parameters, for comparison:

- Patch representation.
- Mutation operators selected with equal random probability. 1 mutation, 1 crossover/individual/iteration.
- Population size: 40. 10 iterations or 12 wall-clock hours, whichever comes first. Tournament size: 2.

55/105 bugs repaired using default parameters.

Some bugs are more difficult to repair than others!

- Easy: 100% success rate on default parameters.
- Medium: 50 – 100% success rate on default parameters
- Hard: 1 – 50% success rate on default parameters
- Unfixed: 0% success

**Metrics:**

- # fitness evaluations to a repair
- GP success rate.
RESEARCH QUESTIONS

Representation:
• Which representation choice gives better results?
• Which representation features contribute most to success?

Crossover: Which crossover operator is best?

Operators:
• Which operators contribute the most to success?
• How should they be selected?

Search space: How should the representation weight program statements to best define the search space?
RESEARCH QUESTIONS

Representation:
  • Which representation choice gives better results?
  • Which representation features contribute most to success?

Crossover: Which crossover operator is best?

Operators:
  • Which operators contribute the most to success?
  • How should they be selected?

Search space: How should the representation weight program statements to best define the search space?
RESEARCH QUESTIONS

Representation:

• Which representation choice gives better results?
• Which representation features contribute most to success?

Crossover: Which crossover operator is best?

Operators:

• Which operators contribute the most to success?
• How should they be selected?

Search space: How should the representation weight program statements to best define the search space?
Procedure:

Compare AST/WP to PATCH on original benchmarks with default parameters.
For both representations, test effectiveness of:

1. Crossover.
2. Semantic check.

Results:

1. Patch outperforms AST/WP (14 – 30%).
2. Semantic check strongly influences success rate of both representations.
3. Crossover also improves results.
RESEARCH QUESTIONS

Representation:

• Which representation choice gives better results?
• Which representation features contribute most to success?

Crossover: Which crossover operator is best?

Operators:

• Which operators contribute the most to success?
• How should they be selected?

Search space: How should the representation weight program statements to best define the search space?
## Crossover: Results

<table>
<thead>
<tr>
<th>Crossover Operator</th>
<th>Success Rate</th>
<th>Fitness evaluations to repair</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>54.4%</td>
<td>82.43</td>
</tr>
<tr>
<td>Default/“Uniform”</td>
<td>61.1%</td>
<td>163.05</td>
</tr>
<tr>
<td>One-Point/AST-WP</td>
<td>63.7%</td>
<td>114.12</td>
</tr>
<tr>
<td>One-Point/Patch</td>
<td>65.2%</td>
<td>118.20</td>
</tr>
</tbody>
</table>
RESEARCH QUESTIONS

Representation:

• Which representation choice gives better results?
• Which representation features contribute most to success?

Crossover: Which crossover operator is best?

Operators:

• Which operators contribute the most to success?
• How should they be selected?

Search space: How should the representation weight program statements to best define the search space?
SEARCH SPACE: SETUP

Hypothesis: statements executed only by the failing test case(s) should be mutated more often than those also executed by the passing test cases.

Procedure: examine that ratio in actual repairs.

Result:

Expected: $10 : 1$

vs.

Actual: $1 : 1.85$
SEARCH SPACE: REPAIR TIME

# fitness evaluations to repair

Search difficulty

- Easy
- Medium
- Hard
- All

<table>
<thead>
<tr>
<th>Difficulty</th>
<th>Default</th>
<th>Realistic</th>
<th>Equal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy</td>
<td>34.3</td>
<td>49.1</td>
<td>36.3</td>
</tr>
<tr>
<td>Medium</td>
<td>93.6</td>
<td>75.7</td>
<td>67.0</td>
</tr>
<tr>
<td>Hard</td>
<td>103.1</td>
<td>27.1</td>
<td>57.7</td>
</tr>
<tr>
<td>All</td>
<td>66.0</td>
<td>59.5</td>
<td>58.6</td>
</tr>
</tbody>
</table>

http://genprog.cs.virginia.edu
SEARCH SPACE: SUCCESS RATE

GP Success Rate

Search difficulty

- Easy
- Medium
- Hard
- All

Default
Realistic
Equal

http://genprog.cs.virginia.edu
This EC problem is atypical; atypical problems warrant study.

We studied representation and operators for EC-based bug repair. These choices matter, especially for difficult bugs.

- Incorporating all recommendations, GenProg repairs 5 new bugs; repair time decreases by 17–43% on difficult bugs.

We don’t know why some of these things are true, but:

- We now have lots of interesting data to dig into!
- We are currently (as we sit here) doing more and bigger runs with the new parameters on the as-yet unfixed bugs.
PLEASE ASK QUESTIONS
## REPRESENTATION BENCHMARKS

<table>
<thead>
<tr>
<th>Program</th>
<th>LOC</th>
<th>Tests</th>
<th>Bug</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gcd</td>
<td>22</td>
<td>6</td>
<td>infinite loop</td>
<td>Example</td>
</tr>
<tr>
<td>uniq-utx</td>
<td>1146</td>
<td>6</td>
<td>segfault</td>
<td>Text processing</td>
</tr>
<tr>
<td>look-utx</td>
<td>1169</td>
<td>6</td>
<td>segfault</td>
<td>Dictionary lookup</td>
</tr>
<tr>
<td>look-svr</td>
<td>1363</td>
<td>6</td>
<td>infinite loop</td>
<td>Dictionary lookup</td>
</tr>
<tr>
<td>units-svr</td>
<td>1504</td>
<td>6</td>
<td>segfault</td>
<td>Metric conversion</td>
</tr>
<tr>
<td>deroff-utx</td>
<td>2236</td>
<td>6</td>
<td>segfault</td>
<td>Document processing</td>
</tr>
<tr>
<td>nullhttpd</td>
<td>5575</td>
<td>7</td>
<td>buffer exploit</td>
<td>Webserver</td>
</tr>
<tr>
<td>indent</td>
<td>9906</td>
<td>6</td>
<td>infinite loop</td>
<td>Source code processing</td>
</tr>
<tr>
<td>flex</td>
<td>18775</td>
<td>6</td>
<td>segfault</td>
<td>Lexical analyzer generator</td>
</tr>
<tr>
<td>atris</td>
<td>21553</td>
<td>3</td>
<td>buffer exploit</td>
<td>Graphical tetris game</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>63249</strong></td>
<td></td>
<td></td>
<td><strong><a href="http://genprog.cs.virginia.edu">http://genprog.cs.virginia.edu</a></strong></td>
</tr>
</tbody>
</table>
## SEARCH SPACE BENCHMARKS

<table>
<thead>
<tr>
<th>Program</th>
<th>LOC</th>
<th>Tests</th>
<th>Bugs</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fbc</td>
<td>97,000</td>
<td>773</td>
<td>3</td>
<td>Language (legacy)</td>
</tr>
<tr>
<td>gmp</td>
<td>145,000</td>
<td>146</td>
<td>2</td>
<td>Multiple precision math</td>
</tr>
<tr>
<td>gzip</td>
<td>491,000</td>
<td>12</td>
<td>5</td>
<td>Data compression</td>
</tr>
<tr>
<td>libtiff</td>
<td>77,000</td>
<td>78</td>
<td>24</td>
<td>Image manipulation</td>
</tr>
<tr>
<td>lighttpd</td>
<td>62,000</td>
<td>295</td>
<td>9</td>
<td>Web server</td>
</tr>
<tr>
<td>php</td>
<td>1,046,000</td>
<td>8,471</td>
<td>44</td>
<td>Language (web)</td>
</tr>
<tr>
<td>python</td>
<td>407,000</td>
<td>355</td>
<td>11</td>
<td>Language (general)</td>
</tr>
<tr>
<td>wireshark</td>
<td>2,814,000</td>
<td>63</td>
<td>7</td>
<td>Network packet analyzer</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>5,139,000</td>
<td>10,193</td>
<td>105</td>
<td></td>
</tr>
</tbody>
</table>

55/105 bugs repaired using default parameters.

http://genprog.cs.virginia.edu