MINTS – A General Framework and Tool for Supporting Test-suite Minimization

Hwa-You Hsu and Alessandro Orso

School of CS – College of Computing
Georgia Institute of Technology
http://www.cc.gatech.edu/~orso

Partially supported by: NSF, US Air Force, and IBM
Regression Testing

Test suite T

Program P0
Regression Testing
Regression Testing
Regression Testing

Test suite $T$ → Regression test selection → Test suite $T'$

Program $P_0$ → Program $P_1$
Regression Testing

Regression test selection → Test suite $T'$ → Test-suite augmentation → Test suite $T_{aug}$

Program $P_0$ → Program $P_1$
Regression Testing

Regression test selection

Test suite T' 

Test suite Taug
Test Suite Minimization

Test suite Taug
Test Suite Minimization

Test suite Taug → Test-suite minimization → Minimized test suite

Redundant test cases
Test Suite Minimization

Criteria:
- coverage
- fault-detection ability
- time
- cost
- ...

Test suite Taug

Test-suite minimization

Minimized test suite

Redundant test cases
## A Simple Example

**Test suite Taug**

<table>
<thead>
<tr>
<th></th>
<th>t1</th>
<th>t2</th>
<th>t3</th>
<th>t4</th>
</tr>
</thead>
<tbody>
<tr>
<td>stmt1</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>stmt2</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>stmt3</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
A Simple Example

Test suite Taug

Minimize test suite while maintaining the same level of coverage
A Simple Example

Minimize test suite while maintaining the same level of coverage
A More Realistic Example
A More Realistic Example

Relevant parameters:
1. Test suite to minimize: \( T = \{t_1, t_2, t_3, t_4\} \)
2. Requirements to cover: \( R = \{\text{stmt1, stmt2, stmt3}\} \)

<table>
<thead>
<tr>
<th></th>
<th>t1</th>
<th>t2</th>
<th>t3</th>
<th>t4</th>
</tr>
</thead>
<tbody>
<tr>
<td>stmt1</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>stmt2</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>stmt3</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Criteria of interest:
C1 – maintain coverage
A More Realistic Example

Relevant parameters:
1. Test suite to minimize: \( T = \{t_1, t_2, t_3, t_4\} \)
2. Requirements to cover: \( R = \{\text{stmt1, stmt2, stmt3}\} \)
3. Test-related data: cost and fault-detection data

<table>
<thead>
<tr>
<th></th>
<th>t1</th>
<th>t2</th>
<th>t3</th>
<th>t4</th>
</tr>
</thead>
<tbody>
<tr>
<td>stmt1</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>stmt2</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>stmt3</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Criteria of interest:
C1 – maintain coverage
A More Realistic Example

Relevant parameters:
1. Test suite to minimize: $T = \{t_1, t_2, t_3, t_4\}$
2. Requirements to cover: $R = \{\text{stmt1, stmt2, stmt3}\}$
3. Test-related data: cost and fault-detection data

<table>
<thead>
<tr>
<th>stmt</th>
<th>t1</th>
<th>t2</th>
<th>t3</th>
<th>t4</th>
</tr>
</thead>
<tbody>
<tr>
<td>stmt1</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>stmt2</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>stmt3</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Criteria of interest:
C1 – maintain coverage

| Time to run | 22 | 4 | 16 | 2 |
| Setup effort| 3  | 0 | 11 | 9 |
A More Realistic Example

Relevant parameters:
1. Test suite to minimize: \( T = \{t_1, t_2, t_3, t_4\} \)
2. Requirements to cover: \( R = \{\text{stmt1, stmt2, stmt3}\} \)
3. Test-related data: cost and fault-detection data

<table>
<thead>
<tr>
<th></th>
<th>t1</th>
<th>t2</th>
<th>t3</th>
<th>t4</th>
</tr>
</thead>
<tbody>
<tr>
<td>stmt1</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>stmt2</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>stmt3</td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Time to run</td>
<td>22</td>
<td>4</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>Setup effort</td>
<td>3</td>
<td>0</td>
<td>11</td>
<td>9</td>
</tr>
<tr>
<td>Fault detection ability</td>
<td>8</td>
<td>4</td>
<td>10</td>
<td>2</td>
</tr>
</tbody>
</table>

Criteria of interest:
C1 – maintain coverage
A More Realistic Example

**Relevant parameters:**
1. Test suite to minimize: $T = \{t_1, t_2, t_3, t_4\}$
2. Requirements to cover: $R = \{\text{stmt1}, \text{stmt2}, \text{stmt3}\}$
3. Test-related data: cost and fault-detection data

<table>
<thead>
<tr>
<th></th>
<th>t1</th>
<th>t2</th>
<th>t3</th>
<th>t4</th>
</tr>
</thead>
<tbody>
<tr>
<td>stmt1</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>stmt2</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>stmt3</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Time to run</td>
<td>22</td>
<td>4</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>Setup effort</td>
<td>3</td>
<td>0</td>
<td>11</td>
<td>9</td>
</tr>
<tr>
<td>Fault detection ability</td>
<td>8</td>
<td>4</td>
<td>10</td>
<td>2</td>
</tr>
</tbody>
</table>

**Criteria of interest:**
C1 – maintain coverage
C2 – minimize time to run
### Relevant parameters:
1. Test suite to minimize: $T = \{t_1, t_2, t_3, t_4\}$
2. Requirements to cover: $R = \{\text{stmt1}, \text{stmt2}, \text{stmt3}\}$
3. Test-related data: cost and fault-detection data

<table>
<thead>
<tr>
<th>stmt1</th>
<th>stmt2</th>
<th>stmt3</th>
<th>stmt4</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>t2</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>t3</td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>t4</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Criteria of interest:
- C1 – maintain coverage
- C2 – minimize time to run
- C3 – minimize setup effort

<table>
<thead>
<tr>
<th>Time to run</th>
<th>Setup effort</th>
<th>Fault detection ability</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>16</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>2</td>
</tr>
</tbody>
</table>
A More Realistic Example

**Relevant parameters:**
1. Test suite to minimize: $T = \{t_1, t_2, t_3, t_4\}$
2. Requirements to cover: $R = \{\text{stmt1}, \text{stmt2}, \text{stmt3}\}$
3. Test-related data: cost and fault-detection data

<table>
<thead>
<tr>
<th></th>
<th>t1</th>
<th>t2</th>
<th>t3</th>
<th>t4</th>
</tr>
</thead>
<tbody>
<tr>
<td>stmt1</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>stmt2</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>stmt3</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Time to run</td>
<td>22</td>
<td>4</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>Setup effort</td>
<td>3</td>
<td>0</td>
<td>11</td>
<td>9</td>
</tr>
<tr>
<td>Fault detection ability</td>
<td>8</td>
<td>4</td>
<td>10</td>
<td>2</td>
</tr>
</tbody>
</table>

**Criteria of interest:**
C1 – maintain coverage
C2 – minimize time to run
C3 – minimize setup effort
C4 – maximize fault detection
State of the Art

Several approaches in the literature (e.g., [HGS93], [H99], [MB03], [BMK04], [TG05])

Two main limitations:

- **Single criterion**
  (typically, coverage)

- **Approximated**
  (problem is NP-complete)

Only exception is [BMK04]: two criteria, but still limited in terms of expressiveness
Our Contribution

MINTS – novel technique (and freely-available tool) for test-suite minimization that:

- Lets testers specify a wide range of multi-criteria test-suite minimization problems
- Automatically encodes problems in binary ILP form
- Leverages different ILP solvers to find optimal solutions in a “reasonable” time
Outline

- Introduction
- Our technique
- Empirical evaluation
- Conclusion and future work
Outline

- Introduction
- Our technique
- Empirical evaluation
- Conclusion and future work
Outline

- Introduction
- Our technique
- Empirical evaluation
- Conclusion and future work
Overview of MINTS

Test suite

Coverage data
Cost data
Fault detection data

Minimization criteria
- Criterion #1
- Criterion #2
- Criterion #n
- Minimization policy

Minimized Test suite

Solver 1

Solution (or timeout)

Minimization problem (suitably encoded)

MINTS tool
Overview of MINTS
Overview of MINTS

Test suite

Coverage data
Cost data
Fault detection data

Test-related data

Minimization criteria
Criterion #1
Criterion #2
Criterion #n
Minimization policy

MINTS tool

Minimized Test suite

Solver 1
... Solver n

Minimization problem (suitably encoded)
Solution (or timeout)

Testing team
Overview of MINTS

Test-related data
- Coverage data
- Cost data
- Fault detection data

Minimization criteria
- Criterion #1
- Criterion #2
- Criterion #n

Minimization policy

Testing team

Solver 1

Minimized Test suite

Solver n
Minimization Criteria

Absolute criteria

- Introduce a constraint
- Example: C1 – Maintain statement coverage

Relative criteria

- Introduce an objective
- Example: C2 – Minimize time to run

Note: the same set of data can be used for either type of criteria
Minimization Policy

- Defines how to combine different objectives
  - Weighted
  - Prioritized
  - Hybrid
Minimization Policy

- Defines how to combine different objectives

**Weighted**

- Testers associate a weight to each objective
- Weights indicate relative importance

Example: very limited man power:
- C2 – minimize time to run ➡ 0.1
- C3 – minimize setup effort ➡ 0.8
- C4 – maximize fault detection ➡ 0.1

**Prioritized**

**Hybrid**
Minimization Policy

- Defines how to combine different objectives
  - Weighted
  - Prioritized

  Testers specify a priority order for each objective
  - Priorities indicate order of processing

  Example: C3 ➔ 1, C2 ➔ 2, C4 ➔ 3:
  - S1 ⊆ 2T = min setup effort
  - S2 ⊆ S1 = min testing time
  - S3 ⊆ S2 = max fault detection

- Hybrid

<table>
<thead>
<tr>
<th>Objective</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C2</td>
<td>minimize time to run</td>
</tr>
<tr>
<td>C3</td>
<td>minimize setup effort</td>
</tr>
<tr>
<td>C4</td>
<td>maximize fault detection</td>
</tr>
</tbody>
</table>
Minimization Policy

- Defines how to combine different objectives
  - Weighted
  - Prioritized
  - Testers specify a priority order for each objective
  - Priorities indicate order of processing
  - Example: C3 ➞ 1, C2 ➞ 2, C4 ➞ 3:
    S1 ⊆ 2^T = min setup effort
    S2 ⊆ S1 = min testing time
    S3 ⊆ S2 = max fault detection

- Hybrid
  - C2 – minimize time to run
  - C3 – minimize setup effort
  - C4 – maximize fault detection
Minimization Policy

- Defines how to combine different objectives
- **Weighted**
- **Prioritized**
  - Testers specify a priority order for each objective
  - Priorities indicate order of processing
  - Example: C3 ➞ 1, C2 ➞ 2, C4 ➞ 3:
    - $S_1 \subseteq 2^T = \text{min setup effort}$
    - $S_2 \subseteq S_1 = \text{min testing time}$
    - $S_3 \subseteq S_2 = \text{max fault detection}$
- **Hybrid**

C2 - minimize time to run
C3 - minimize setup effort
C4 - maximize fault detection
Minimization Policy

- Defines how to combine different objectives
- **Weighted**
- **Prioritized**
  - Testers specify a priority order for each objective
  - Priorities indicate order of processing
  - Example: C3 ➔ 1, C2 ➔ 2, C4 ➔ 3:
    - S1 ⊆ 2^T = min setup effort
    - S2 ⊆ S1 = min testing time
    - S3 ⊆ S2 = max fault detection
- **Hybrid**

C2 – minimize time to run
C3 – minimize setup effort
C4 – maximize fault detection
Minimization Policy

- Defines how to combine different objectives
  - Weighted
  - Prioritized
  - Testers specify a priority order for each objective
  - Priorities indicate order of processing
  - Example: C3 ➔ 1, C2 ➔ 2, C4 ➔ 3:
    - S1 ≤ 2^T = min setup effort
    - S2 ≤ S1 = min testing time
    - S3 ≤ S2 = max fault detection
  - Hybrid

C2 – minimize time to run
C3 – minimize setup effort
C4 – maximize fault detection
Minimization Policy

- Defines how to combine different objectives
  - Weighted
  - Prioritized
  - Hybrid

  Testers cluster objectives into groups and
  - assign weights to objects within group
  - assign priorities to groups
Overview of MINTS

Minimization criteria
- Criterion #1
- Criterion #2
- Criterion #n

Minimization policy

Test suite

Test-related data
- Coverage data
- Cost data
- Fault detection data

MINTS tool

Minimized Test suite

Minimization problem (suitably encoded)
Solution (or timeout)

Solver 1

Solver n
Overview of MINTS

Test suite

Coverage data
Cost data
Fault detection data

Minimization criteria
Criterion #1
Criterion #2
Criterion #n
Minimization policy

Solver 1
...
Solver n

Minimized Test suite
Multi-criteria minimization as a binary ILP problem: Encoding

<table>
<thead>
<tr>
<th></th>
<th>t1</th>
<th>t2</th>
<th>t3</th>
<th>t4</th>
</tr>
</thead>
<tbody>
<tr>
<td>stmt1</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>stmt2</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>stmt3</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Time to run</td>
<td>22</td>
<td>4</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>Setup effort</td>
<td>3</td>
<td>0</td>
<td>11</td>
<td>9</td>
</tr>
<tr>
<td>F. detection</td>
<td>8</td>
<td>4</td>
<td>10</td>
<td>2</td>
</tr>
</tbody>
</table>
Multi-criteria minimization as a binary ILP problem: Encoding

Minimized test suite \( MT=\{o_i\}, \ 1 \leq i \leq |T|, \ o_i=1 \iff t_i \in MT \)
Multi-criteria minimization as a binary ILP problem: Encoding

Minimized test suite $MT=\{o_i\}, \ 1 \leq i \leq |T|, \ o_i=1 \ \text{iff} \ t_i \in MT$

Test-related data (types 1..n) $d_{all}=\{d_{i,j}\}, \ 1 \leq i \leq |n|, 1 \leq j \leq |T|$

<table>
<thead>
<tr>
<th>stmt1</th>
<th>t1</th>
<th>t2</th>
<th>t3</th>
<th>t4</th>
</tr>
</thead>
<tbody>
<tr>
<td>stmt1</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>stmt2</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>stmt3</td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

| Time to run | 22 | 16 | 4  | 2  |
| Setup effort | 3  | 11 | 0  | 9  |
| F. detection | 10 | 8  | 4  | 2  |
Multi-criteria minimization as a binary ILP problem: Encoding

Minimized test suite \( MT = \{ o_i \}, \quad 1 \leq i \leq |T|, \quad o_i = 1 \text{ iff } t_i \in MT \)

Test-related data (types 1..n) \( d_{all} = \{ d_{i,j} \}, \quad 1 \leq i \leq |n|, 1 \leq j \leq |T| \)

Test-related data (type x) \( d_x = \{ d_{x,j} \}, \quad 1 \leq j \leq |T| \)

<table>
<thead>
<tr>
<th></th>
<th>t1</th>
<th>t2</th>
<th>t3</th>
<th>t4</th>
</tr>
</thead>
<tbody>
<tr>
<td>stmt1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>stmt2</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>stmt3</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time to run</td>
<td>22</td>
<td>4</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>Setup effort</td>
<td>3</td>
<td>0</td>
<td>11</td>
<td>9</td>
</tr>
<tr>
<td>F. detection</td>
<td>8</td>
<td>4</td>
<td>10</td>
<td>2</td>
</tr>
</tbody>
</table>
Multi-criteria minimization as a binary ILP problem: Encoding

Minimized test suite \( MT=\{o_i\}, 1 \leq i \leq |T|, \ o_i=1 \text{ iff } t_i \in MT \)

Test-related data (types 1..n) \( d_{all}=\{d_{i,j}\}, 1 \leq i \leq \mid n \mid, 1 \leq j \leq |T| \)

Test-related data (type x) \( d_x=\{d_{x,j}\}, 1 \leq j \leq |T| \)

Absolute criteria (type x): \( \sum_{j=1..|T|} d_{x,j} o_j \oplus \text{ const} \)

\( \oplus = \lt, \leq, =, \geq, \text{ or } > \)

<table>
<thead>
<tr>
<th></th>
<th>t1</th>
<th>t2</th>
<th>t3</th>
<th>t4</th>
</tr>
</thead>
<tbody>
<tr>
<td>stmt1</td>
<td>1</td>
<td>1</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>stmt2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>stmt3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Time to run</td>
<td>22</td>
<td>4</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>Setup effort</td>
<td>3</td>
<td>0</td>
<td>11</td>
<td>9</td>
</tr>
<tr>
<td>F. detection</td>
<td>8</td>
<td>4</td>
<td>10</td>
<td>2</td>
</tr>
</tbody>
</table>

Friday, May 22, 2009
Multi-criteria minimization as a binary ILP problem: Encoding

Minimized test suite \( MT = \{ o_i \}, \ 1 \leq i \leq |T|, \ o_i = 1 \ \text{iff} \ t_i \in MT \)

Test-related data (types 1..n) \( d_{all} = \{ d_{i,j} \}, \ 1 \leq i \leq |n|, 1 \leq j \leq |T| \)

Test-related data (type x) \( d_x = \{ d_{x,j} \}, \ 1 \leq j \leq |T| \)

Absolute criteria (type x): \( \sum_{j=1..|T|} d_{x,j} o_j \oplus \text{const} \)

For example:

Criterion #1: \( \sum_{j=1..4} d_{1,j} o_j = o_1 + o_3 \geq 1 \) (maintain coverage) \( \sum_{j=1..4} d_{2,j} o_j = o_1 + o_2 \geq 1 \)

\( \oplus = <, \leq, =, \geq, \text{or} \geq > \)

<table>
<thead>
<tr>
<th></th>
<th>t1</th>
<th>t2</th>
<th>t3</th>
<th>t4</th>
</tr>
</thead>
<tbody>
<tr>
<td>stmt1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>stmt2</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>stmt3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Time to run</td>
<td>22</td>
<td>4</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>Setup effort</td>
<td>3</td>
<td>0</td>
<td>11</td>
<td>9</td>
</tr>
<tr>
<td>F. detection</td>
<td>8</td>
<td>4</td>
<td>10</td>
<td>2</td>
</tr>
</tbody>
</table>
Multi-criteria minimization as a binary ILP problem: Encoding

Minimized test suite \(MT=\{o_i\}, 1 \leq i \leq |T|, o_i=1 \text{ iff } t_i \in MT\)

Test-related data (types 1..n) \(d_{all}=\{d_{i,j}\}, 1 \leq i \leq |n|, 1 \leq j \leq |T|\)

Test-related data (type x) \(d_x=\{d_{x,j}\}, 1 \leq j \leq |T|\)

Absolute criteria (type x): \(\sum_{j=1..|T|} d_{x,j} o_j \oplus \text{const}\)

Relative criteria (type x): \(\min/\max \sum_{j=1..|T|} \text{norm}(d_{x,j}) o_j \quad \left(\sum_{j=1..|T|} \text{norm}(d_{j}) = 1\right)\)

\(\oplus = <, \leq, =, \geq, \text{ or } >\)

<table>
<thead>
<tr>
<th></th>
<th>t1</th>
<th>t2</th>
<th>t3</th>
<th>t4</th>
</tr>
</thead>
<tbody>
<tr>
<td>stmt1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>stmt2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>stmt3</td>
<td>1</td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Time to run</td>
<td>22</td>
<td>4</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>Setup effort</td>
<td>3</td>
<td>0</td>
<td>11</td>
<td>9</td>
</tr>
<tr>
<td>F. detection</td>
<td>8</td>
<td>4</td>
<td>10</td>
<td>2</td>
</tr>
</tbody>
</table>

Friday, May 22, 2009
Multi-criteria minimization as a binary ILP problem: Encoding

Minimized test suite \( MT = \{ o_i \} \), \( 1 \leq i \leq |T| \), \( o_i = 1 \) iff \( t_i \in MT \)

Test-related data (types 1..n) \( d_{all} = \{ d_{i,j} \} \), \( 1 \leq i \leq |n| \), \( 1 \leq j \leq |T| \)

Test-related data (type x) \( d_x = \{ d_{x,j} \} \), \( 1 \leq j \leq |T| \)

Absolute criteria (type x): \( \sum_{j=1}^{\mid T\mid} d_{x,j} o_j \oplus \text{const} \)

Relative criteria (type x): \( \min/\max \sum_{j=1}^{\mid T\mid} \|d_{x,j}\| o_j \)  
\( (\sum_{j=1}^{\mid T\mid} \|d_{j}\| = 1) \)

For example:

Criterion #2 (minimize time to run):  
\[
\min \sum_{j=1..4} \|d_{3,j}\| o_j = 0.5o_1 + 0.1o_2 + 0.36o_3 + 0.04o_4
\]

\( \oplus = <, \leq, =, \geq, \text{or } > \)
Multi-criteria minimization as a binary ILP problem:

Encoding

Minimized test suite $MT=\{o_i\}, \ 1\leq i \leq |T|, \ o_i=1 \text{ iff } t_i \in MT$

Test-related data (types 1..n) $d_{all}=\{d_{i,j}\}, \ 1\leq i \leq |n|, 1\leq j \leq |T|$

Test-related data (type x) $d_x=\{d_{x,j}\}, \ 1\leq j \leq |T|$

Absolute criteria (type x): $\sum_{j=1..|T|} d_{x,j} o_j \oplus \text{ const}$

Relative criteria (type x): min/max $\sum_{j=1..|T|} \text{norm}(d_{x,j}) o_j$

Minimization policies

<table>
<thead>
<tr>
<th></th>
<th>t1</th>
<th>t2</th>
<th>t3</th>
<th>t4</th>
</tr>
</thead>
<tbody>
<tr>
<td>stmt1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>stmt2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>stmt3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Time to run</td>
<td>22</td>
<td>4</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>Setup effort</td>
<td>3</td>
<td>0</td>
<td>11</td>
<td>9</td>
</tr>
<tr>
<td>F. detection</td>
<td>8</td>
<td>4</td>
<td>10</td>
<td>2</td>
</tr>
</tbody>
</table>
Multi-criteria minimization as a binary ILP problem: Encoding

Minimized test suite $MT=\{o_i\}, \ 1 \leq i \leq |T|$, $o_i = 1$ iff $t_i \in MT$

Test-related data (types 1..n) $d_{all}=\{d_{i,j}\}, \ 1 \leq i \leq |n|, 1 \leq j \leq |T|$

Test-related data (type x) $d_{x}=\{d_{x,j}\}, \ 1 \leq j \leq |T|$

Absolute criteria (type x): $\sum_{j=1..|T|} d_{x,j} o_j \oplus \text{const}$

Relative criteria (type x): $\min/\max \sum_{j=1..|T|} \text{norm}(d_{x,j}) o_j \quad (\sum_{j=1..|T|} \text{norm}(d_{j}) = 1)$

Minimization policies

Weighted: $\{\alpha_j\}, \ 1 \leq j \leq \#\text{relative criteria}$

<table>
<thead>
<tr>
<th>stmt1</th>
<th>stmt2</th>
<th>stmt3</th>
<th>stmt4</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>t2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>t3</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>t4</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

$\oplus = <, \leq, =, \geq, \text{ or } >$

- Time to run
  - 22
  - 4
  - 16
  - 2

- Setup effort
  - 3
  - 0
  - 11
  - 9

- F. detection
  - 8
  - 4
  - 10
  - 2

Friday, May 22, 2009
Multi-criteria minimization as a binary ILP problem: Encoding

Minimized test suite \( MT=\{o_i\}, \ 1 \leq i \leq |T| \), \( o_i = 1 \) iff \( t_i \in MT \)

Test-related data (types 1..n) \( d_{all}=\{d_{i,j}\}, \ 1 \leq i \leq |n|, 1 \leq j \leq |T| \)

Test-related data (type x) \( d_x=\{d_{x,j}\}, \ 1 \leq j \leq |T| \)

Absolute criteria (type x): \( \sum_{j=1..|T|} d_{x,j} o_j \oplus \text{const} \)

Relative criteria (type x): \( \min/\max \sum_{j=1..|T|} \text{norm}(d_{x,j}) o_j \)

(\( \sum_{j=1..|T|} \text{norm}(d_{,j}) = 1 \))

Minimization policies

- Weighted: \( \{\alpha_j\}, \ 1 \leq j \leq \#\text{relative criteria} \)
- Prioritized: criterion \( \Rightarrow \) integer

### Encoding

\( \oplus = <, <=, =, >=, \text{or} > \)

<table>
<thead>
<tr>
<th></th>
<th>t1</th>
<th>t2</th>
<th>t3</th>
<th>t4</th>
</tr>
</thead>
<tbody>
<tr>
<td>stmt1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>stmt2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>stmt3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time to run</td>
<td>22</td>
<td>4</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>Setup effort</td>
<td>3</td>
<td>0</td>
<td>11</td>
<td>9</td>
</tr>
<tr>
<td>F. detection</td>
<td>8</td>
<td>4</td>
<td>10</td>
<td>2</td>
</tr>
</tbody>
</table>

Friday, May 22, 2009
Multi-criteria minimization as a binary ILP problem: Weighted policy

Given

- $n$ relative criteria involving test data $d_{x1}, ..., d_{xn}$
- $m$ absolute criteria involving test data $d_{y1}, ..., d_{ym}$
- A weighted policy with weights $\alpha_1, ..., \alpha_n$
Multi-criteria minimization as a binary ILP problem: Weighted policy

Given

- n relative criteria involving test data $d_{x1}, ..., d_{xn}$
- m absolute criteria involving test data $d_{y1}, ..., d_{ym}$
- A weighted policy with weights $\alpha_1, ..., \alpha_n$

MINTS encode the minimization problem as

\[
\text{minimize} \\
\sum_{i=1}^{n} \alpha_i \sum_{j=1}^{\mid T \mid} \text{norm}(d_{xi,j}) o_j
\]

\text{under the constraints}

\[
\sum_{j=1}^{\mid T \mid} d_{y1,j} o_j \oplus \text{const}_1 \\
... \\
\sum_{j=1}^{\mid T \mid} d_{y1,j} o_j \oplus \text{const}_1
\]
Multi-criteria minimization as a binary ILP problem: Weighted policy

Given

- $n$ relative criteria involving test data $d_{x1}, \ldots, d_{xn}$
- $m$ absolute criteria involving test data $d_{y1}, \ldots, d_{ym}$
- A weighted policy with weights $\alpha_1, \ldots, \alpha_n$

MINTS encode the minimization problem as

$$\text{minimize}$$
$$\sum_{i=1}^{n} \alpha_i \sum_{j=1}^{|T|} \text{norm}(d_{xi,j}) o_j$$

under the constraints
$$\sum_{j=1}^{|T|} d_{y1,j} o_j \oplus \text{const}_1$$
$$\vdots$$
$$\sum_{j=1}^{|T|} d_{ym,j} o_j \oplus \text{const}_1$$
Multi-criteria minimization as a binary ILP problem: Weighted policy

Given

- $n$ relative criteria involving test data $d_{x1}, \ldots, d_{xn}$
- $m$ absolute criteria involving test data $d_{y1}, \ldots, d_{ym}$
- A weighted policy with weights $\alpha_1, \ldots, \alpha_n$

MINTS encode the minimization problem as

\[
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{n} \alpha_i \sum_{j=1}^{|T|} \text{norm}(d_{xi,j})o_j \\
\text{under the constraints} & \quad \sum_{j=1}^{|T|} d_{y1,j} o_j \oplus \text{const}_1 \\
& \quad \ldots \\
& \quad \sum_{j=1}^{|T|} d_{y1,j} o_j \oplus \text{const}_1
\end{align*}
\]
Multi-criteria minimization as a binary ILP problem: Weighted policy

Given

- n relative criteria involving test data $d_{x1}, ..., d_{xn}$
- m absolute criteria involving test data $d_{y1}, ..., d_{ym}$
- A weighted policy with weights $\alpha_1, ..., \alpha_n$

MINTS encode the minimization problem as

Minimize

$0.1(0.5o_1 + 0.2o_2 + 0.36o_3 + 0.04o_4) + 0.8(0.13o_1 + 0.48o_3 + 0.39o_4) - 0.1(0.3o_1 + 0.17o_2 + 0.42o_3 + 0.08o_4)$

Under the constraints

$o_1 + o_3 \geq 1, o_1 + o_2 \geq 1, o_3 + o_4 \geq 1$

$\Rightarrow \text{MT} = \{0,1,1,0\}$
Empirical Evaluation

Goal: assess usefulness and practicality of the approach

RQ1: How often can MINTS find an optimal solution “quickly”?

RQ2: How does MINTS compare with a heuristic approach?

RQ3: How does the use of a specific solver affect MINTS’s performance?
Empirical Evaluation

**Goal**: assess usefulness and practicality of the approach

**RQ1**: How often can MINTS find an optimal solution “quickly”?

**RQ2**: How does MINTS compare with a heuristic approach?

**RQ3**: How does the use of a specific solver affect MINTS’s performance?
## Experimental Subjects and Solvers Considered

<table>
<thead>
<tr>
<th>Subject</th>
<th>LOC</th>
<th>COV</th>
<th>#Test Cases</th>
<th>#Versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcas</td>
<td>173</td>
<td>72</td>
<td>1608</td>
<td>5</td>
</tr>
<tr>
<td>schedule2</td>
<td>307</td>
<td>146</td>
<td>2700</td>
<td>5</td>
</tr>
<tr>
<td>tot_info</td>
<td>406</td>
<td>136</td>
<td>1052</td>
<td>5</td>
</tr>
<tr>
<td>schedule</td>
<td>412</td>
<td>166</td>
<td>2650</td>
<td>5</td>
</tr>
<tr>
<td>replace</td>
<td>562</td>
<td>263</td>
<td>5542</td>
<td>5</td>
</tr>
<tr>
<td>print_tokens</td>
<td>563</td>
<td>194</td>
<td>4130</td>
<td>5</td>
</tr>
<tr>
<td>print_tokens2</td>
<td>570</td>
<td>197</td>
<td>4115</td>
<td>5</td>
</tr>
<tr>
<td>flex</td>
<td>12,421</td>
<td>567</td>
<td>548</td>
<td>5</td>
</tr>
<tr>
<td>LogicBlox</td>
<td>570,595</td>
<td>29204</td>
<td>393</td>
<td>5</td>
</tr>
<tr>
<td>Eclipse</td>
<td>1,892,226</td>
<td>35903</td>
<td>3621</td>
<td>5</td>
</tr>
</tbody>
</table>
### Experimental Subjects and Solvers Considered

<table>
<thead>
<tr>
<th>Subject</th>
<th>LOC</th>
<th>COV</th>
<th>#Test Cases</th>
<th>#Versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcas</td>
<td>173</td>
<td>72</td>
<td>1608</td>
<td>5</td>
</tr>
<tr>
<td>schedule2</td>
<td>307</td>
<td>146</td>
<td>2700</td>
<td>5</td>
</tr>
<tr>
<td>tot_info</td>
<td>406</td>
<td>136</td>
<td>1052</td>
<td>5</td>
</tr>
<tr>
<td>schedule</td>
<td>412</td>
<td>166</td>
<td>2650</td>
<td>5</td>
</tr>
<tr>
<td>replace</td>
<td>562</td>
<td>263</td>
<td>5542</td>
<td>5</td>
</tr>
<tr>
<td>print_tokens</td>
<td>563</td>
<td>194</td>
<td>4130</td>
<td>5</td>
</tr>
<tr>
<td>print_tokens2</td>
<td>570</td>
<td>197</td>
<td>4115</td>
<td>5</td>
</tr>
<tr>
<td>flex</td>
<td>12,421</td>
<td>567</td>
<td>548</td>
<td>5</td>
</tr>
<tr>
<td>LogicBlox</td>
<td>570,595</td>
<td>29204</td>
<td>393</td>
<td>5</td>
</tr>
<tr>
<td>Eclipse</td>
<td>1,892,226</td>
<td>35903</td>
<td>3621</td>
<td>5</td>
</tr>
</tbody>
</table>
Experimental Subjects and Solvers Considered

Subjects:

<table>
<thead>
<tr>
<th>Subject</th>
<th>LOC</th>
<th>COV</th>
<th>#Test Cases</th>
<th>#Versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcas</td>
<td>173</td>
<td>72</td>
<td>1608</td>
<td>5</td>
</tr>
<tr>
<td>schedule2</td>
<td>307</td>
<td>146</td>
<td>2700</td>
<td>5</td>
</tr>
<tr>
<td>tot_info</td>
<td>406</td>
<td>136</td>
<td>1052</td>
<td>5</td>
</tr>
<tr>
<td>schedule</td>
<td>412</td>
<td>166</td>
<td>2650</td>
<td>5</td>
</tr>
<tr>
<td>replace</td>
<td>562</td>
<td>263</td>
<td>5542</td>
<td>5</td>
</tr>
<tr>
<td>print_tokens</td>
<td>563</td>
<td>194</td>
<td>4130</td>
<td>5</td>
</tr>
<tr>
<td>print_tokens2</td>
<td>570</td>
<td>197</td>
<td>4115</td>
<td>5</td>
</tr>
<tr>
<td>flex</td>
<td>12,421</td>
<td>567</td>
<td>548</td>
<td>5</td>
</tr>
<tr>
<td>LogicBlox</td>
<td>570,595</td>
<td>29204</td>
<td>393</td>
<td>5</td>
</tr>
<tr>
<td>Eclipse</td>
<td>1,892,226</td>
<td>35903</td>
<td>3621</td>
<td>5</td>
</tr>
</tbody>
</table>
# Experimental Subjects and Solvers Considered

**Subjects:**

<table>
<thead>
<tr>
<th>Subject</th>
<th>LOC</th>
<th>COV</th>
<th>#Test Cases</th>
<th>#Versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcas</td>
<td>173</td>
<td>72</td>
<td>1608</td>
<td>5</td>
</tr>
<tr>
<td>schedule2</td>
<td>307</td>
<td>146</td>
<td>2700</td>
<td>5</td>
</tr>
<tr>
<td>tot_info</td>
<td>406</td>
<td>136</td>
<td>1052</td>
<td>5</td>
</tr>
<tr>
<td>schedule</td>
<td>412</td>
<td>166</td>
<td>2650</td>
<td>5</td>
</tr>
<tr>
<td>replace</td>
<td>562</td>
<td>263</td>
<td>5542</td>
<td>5</td>
</tr>
<tr>
<td>print_tokens</td>
<td>563</td>
<td>194</td>
<td>4130</td>
<td>5</td>
</tr>
<tr>
<td>print_tokens2</td>
<td>570</td>
<td>197</td>
<td>4115</td>
<td>5</td>
</tr>
<tr>
<td>flex</td>
<td>12,421</td>
<td>567</td>
<td>548</td>
<td>5</td>
</tr>
<tr>
<td>LogicBlox</td>
<td>570,595</td>
<td>29204</td>
<td>393</td>
<td>5</td>
</tr>
<tr>
<td>Eclipse</td>
<td>1,892,226</td>
<td>35903</td>
<td>3621</td>
<td>5</td>
</tr>
</tbody>
</table>
Experimental Subjects and Solvers Considered

Subject: Experimental Subjects and Solvers Considered

<table>
<thead>
<tr>
<th>Subject</th>
<th>LOC</th>
<th>COV</th>
<th>#Test Cases</th>
<th>#Versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcas</td>
<td>173</td>
<td>72</td>
<td>1608</td>
<td>5</td>
</tr>
<tr>
<td>schedule2</td>
<td>307</td>
<td>146</td>
<td>2700</td>
<td>5</td>
</tr>
<tr>
<td>tot_info</td>
<td>406</td>
<td>136</td>
<td>1052</td>
<td>5</td>
</tr>
<tr>
<td>schedule</td>
<td>412</td>
<td>166</td>
<td>2650</td>
<td>5</td>
</tr>
<tr>
<td>replace</td>
<td>562</td>
<td>263</td>
<td>5542</td>
<td>5</td>
</tr>
<tr>
<td>print_tokens</td>
<td>563</td>
<td>194</td>
<td>4130</td>
<td>5</td>
</tr>
<tr>
<td>print_tokens2</td>
<td>570</td>
<td>197</td>
<td>4115</td>
<td>5</td>
</tr>
<tr>
<td>flex</td>
<td>12,421</td>
<td>567</td>
<td>548</td>
<td>5</td>
</tr>
<tr>
<td>LogicBlox</td>
<td>570,595</td>
<td>29204</td>
<td>393</td>
<td>5</td>
</tr>
<tr>
<td>Eclipse</td>
<td>1,892,226</td>
<td>35903</td>
<td>3621</td>
<td>5</td>
</tr>
</tbody>
</table>
## Experimental Subjects and Solvers Considered

### Subjects:

<table>
<thead>
<tr>
<th>Subject</th>
<th>LOC</th>
<th>COV</th>
<th>#Test Cases</th>
<th>#Versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcas</td>
<td>173</td>
<td>72</td>
<td>1608</td>
<td>5</td>
</tr>
<tr>
<td>schedule2</td>
<td>307</td>
<td>146</td>
<td>2700</td>
<td>5</td>
</tr>
<tr>
<td>tot_info</td>
<td>406</td>
<td>136</td>
<td>1052</td>
<td>5</td>
</tr>
<tr>
<td>schedule</td>
<td>412</td>
<td>166</td>
<td>2650</td>
<td>5</td>
</tr>
<tr>
<td>replace</td>
<td>562</td>
<td>263</td>
<td>5542</td>
<td>5</td>
</tr>
<tr>
<td>print_tokens</td>
<td>563</td>
<td>194</td>
<td>4130</td>
<td>5</td>
</tr>
<tr>
<td>print_tokens2</td>
<td>570</td>
<td>197</td>
<td>4115</td>
<td>5</td>
</tr>
<tr>
<td>flex</td>
<td>12,421</td>
<td>567</td>
<td>548</td>
<td>5</td>
</tr>
<tr>
<td>LogicBlox</td>
<td>570,595</td>
<td>29204</td>
<td>393</td>
<td>5</td>
</tr>
<tr>
<td>Eclipse</td>
<td>1,892,226</td>
<td>35903</td>
<td>3621</td>
<td>5</td>
</tr>
</tbody>
</table>
# Experimental Subjects and Solvers Considered

## Subjects:

<table>
<thead>
<tr>
<th>Subject</th>
<th>LOC</th>
<th>COV</th>
<th>#Test Cases</th>
<th>#Versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcas</td>
<td>173</td>
<td>72</td>
<td>1608</td>
<td>5</td>
</tr>
<tr>
<td>schedule2</td>
<td>307</td>
<td>146</td>
<td>2700</td>
<td>5</td>
</tr>
<tr>
<td>tot_info</td>
<td>406</td>
<td>136</td>
<td>1052</td>
<td>5</td>
</tr>
<tr>
<td>schedule</td>
<td>412</td>
<td>166</td>
<td>2650</td>
<td>5</td>
</tr>
<tr>
<td>replace</td>
<td>562</td>
<td>263</td>
<td>5542</td>
<td>5</td>
</tr>
<tr>
<td>print_tokens</td>
<td>563</td>
<td>194</td>
<td>4130</td>
<td>5</td>
</tr>
<tr>
<td>print_tokens2</td>
<td>570</td>
<td>197</td>
<td>4115</td>
<td>5</td>
</tr>
<tr>
<td>flex</td>
<td>12,421</td>
<td>567</td>
<td>548</td>
<td>5</td>
</tr>
<tr>
<td>LogicBlox</td>
<td>570,595</td>
<td>29204</td>
<td>393</td>
<td>5</td>
</tr>
<tr>
<td>Eclipse</td>
<td>1,892,226</td>
<td>35903</td>
<td>3621</td>
<td>5</td>
</tr>
</tbody>
</table>

## Solvers:

Four SAT-based pseudo-Boolean and two pure ILP solvers
RQ1: How often can MINTS find an optimal solution quickly? (setup)
RQ1: How often can MINTS find an optimal solution quickly? (setup)

Test-related data
- Code coverage (gcov, cobertura)
- Running time (UNIX’s time utility)
- Fault-detection ability (#faults detected in previous version)
RQ1: How often can MINTS find an optimal solution quickly? (setup)

Test-related data
- Code coverage (gcov, cobertura)
- Running time (UNIX’s time utility)
- Fault-detection ability (#faults detected in previous version)

Minimization criteria
- One absolute: maintain statement coverage
- Three relatives: min size test suite, min execution time, max fault-detection capability
RQ1: How often can MINTS find an optimal solution quickly? (setup)

Test-related data
- Code coverage (gcov, cobertura)
- Running time (UNIX’s time utility)
- Fault-detection ability (#faults detected in previous version)

Minimization criteria
- One absolute: maintain statement coverage
- Three relatives: min size test suite, min execution time, max fault-detection capability

Minimization policies
- Seven weighted: same weight; 0.6, 0.3, 0.1 (all combinations)
- One prioritized: (1) min size test suite, (2) min execution time, (3) max fault-detection capability
RQ1: How often can MINTS find an optimal solution quickly? (setup)

Test-related data
- Code coverage (gcov, cobertura)
- Running time (UNIX’s time utility)
- Fault-detection ability (#faults detected in previous version)

Minimization criteria
- One absolute: maintain statement coverage
- Three relatives: min size test suite, min execution time, max fault-detection capability

Minimization policies
- Seven weighted: same weight; 0.6, 0.3, 0.1 (all combinations)
- One prioritized: (1) min size test suite, (2) min execution time, (3) max fault-detection capability

Overall, 400 minimization problems covering a wide spectrum
RQ1: How often can MINTS find an optimal solution quickly? (Process and results)

MINTS encoded each problem, submitted it to all solvers, and measured the time required to get the first solution.
RQ1: How often can MINTS find an optimal solution quickly? (Process and results)

MINTS encoded each problem, submitted it to all solvers, and measured the time required to get the first solution.

Ordered by complexity indicator – size of the subject x # test cases
**RQ1:** How often can MINTS find an optimal solution quickly? (Process and results)

MINTS encoded each problem, submitted it to all solvers, and measured the time required to get the first solution.

- MINTS always found an optimal solution.
- All solutions found within 40 sec.
- Less then 10 seconds for the majority of the most complex minimization problems.
- In most cases, less than two sec.

Ordered by complexity indicator – size of the subject x # test cases

Friday, May 22, 2009
RQ1: How often can MINTS find an optimal solution quickly? (Process and results)

MINTS encoded each problem, submitted it to all solvers, and measured the time required to get the first solution.

- MINTS always found an optimal solution.
- All solutions found within 40 sec.
- Less then 10 seconds for the majority of the most complex minimization problems.
- In most cases, less than two sec.
- Clear correlation between complexity and time required.
- Almost linear; promising wrt scalability.

Ordered by complexity indicator – size of the subject x # test cases
RQ2: How does MINTS compare with a heuristic approach?
RQ2: How does MINTS compare with a heuristic approach?

Process
1. Single criterion: maintain statement coverage
2. Implemented HGS [HGS93] – well known, simple
3. Measured
   1. time to solve minimization problems
   2. size of resulting test suite
**RQ2: How does MINTS compare with a heuristic approach?**

**Process**

1. Single criterion: maintain statement coverage
2. Implemented HGS [HGS93] – well known, simple
3. Measured
   1. time to solve minimization problems
   2. size of resulting test suite

**Results**

- Both found solutions to all problems in a few seconds
- MINTS sometimes faster than HGS
- Minimized test suites of the same size for the Siemens programs and flex, of similar size for LogicBlox, and fairly different for Eclipse
RQ2: How does MINTS compare with a heuristic approach?

<table>
<thead>
<tr>
<th>Eclipse version</th>
<th>Original T’s size</th>
<th>HGS</th>
<th>MINTS</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.0.1</td>
<td>2460</td>
<td>656</td>
<td>418</td>
<td>238 (36%)</td>
</tr>
<tr>
<td>3.0.2</td>
<td>2467</td>
<td>651</td>
<td>423</td>
<td>228 (35%)</td>
</tr>
<tr>
<td>3.1</td>
<td>3621</td>
<td>851</td>
<td>553</td>
<td>298 (35%)</td>
</tr>
<tr>
<td>3.1.1</td>
<td>3681</td>
<td>833</td>
<td>532</td>
<td>301 (36%)</td>
</tr>
<tr>
<td>3.1.2</td>
<td>3681</td>
<td>656</td>
<td>406</td>
<td>250 (38%)</td>
</tr>
</tbody>
</table>

Results

- Both found solutions to all problems in a few seconds
- MINTS sometimes faster than HGS
- Minimized test suites of the same size for the Siemens programs and flex, of similar size for LogicBlox, and fairly different for Eclipse
Outline

- Introduction
- Our technique
- Empirical evaluation
- Conclusion and future work
Conclusion and Future Work
Conclusion and Future Work

Summary

- MINTS is a technique and tool for test suite minimization that allows for specifying a wide range of multi-criteria minimization problems.
- Computes (when successful) optimal solutions.
- Empirical results show usefulness and applicability of the approach.
Conclusion and Future Work

Summary

- MINTS is a technique and tool for test suite minimization that
- Allows for specifying a wide range of multi-criteria minimization problems
- Computes (when successful) optimal solutions
- Empirical results show usefulness and applicability of the approach

Future work

- Additional experimentation
- Study solvers’ performance to go beyond the black box
- Extension of MINTS to include prioritization
Thank You!