Search Based Test Data Generation for Server-side Web Application Testing

Nadia Alshahwan and Mark Harman

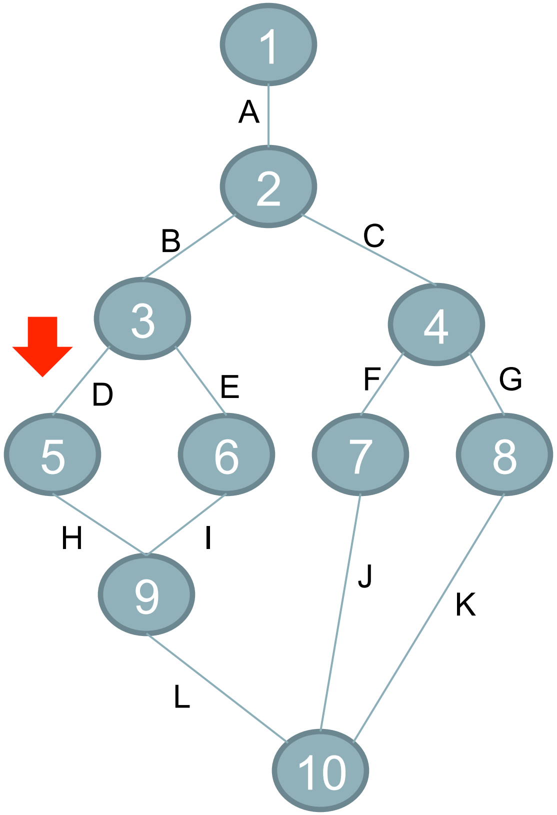
CREST Centre

University College London

Automated web application testing using search based software Engineering (ASE 2011)

- Hill Climbing (AVM)
- Maximise branch coverage
- Server-side code (PHP)

Overall strategy



C. C. Michael, G. McGraw, and M. A. Schatz. Generating software test data by evolution. IEEE Transactions on Software Engineering,2001.

Pros

- Only local distance → no need for approach level
- *Accidental* coverage (highest %)

Pros

- Only local distance → no need for approach level
- *Accidental* coverage (highest %)

Cons

- Not suitable for specific targets
- Distance calculations affect execution time

Pros

- Only local distance → no need for approach level
- *Accidental* coverage (highest %)

Cons

- Not suitable for specific targets
- Distance calculations affect execution time

→ Keep track of covered branches and skip

Web Specific Issues

- Identifying inputs ($_POST[‘inputname’])
- Dynamic includes
- Dynamic typing → check type at run-time
- Non determinism

Dynamic Value Seeding

```
if($x>=$y) {
.
.
}
```

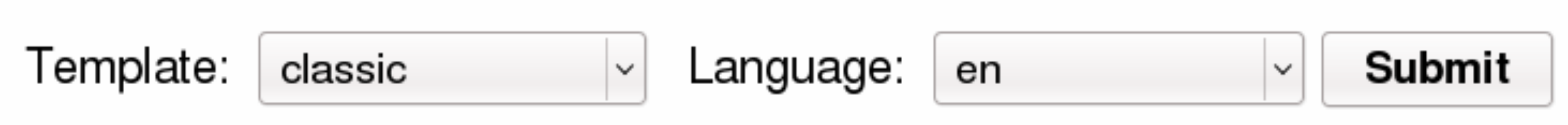
```
$x = 5
$y = 300
```

```
if($class==$result[0]) {
.
.
}
```

```
$class = CS
$result[0] = English
```

Dynamic Value Seeding

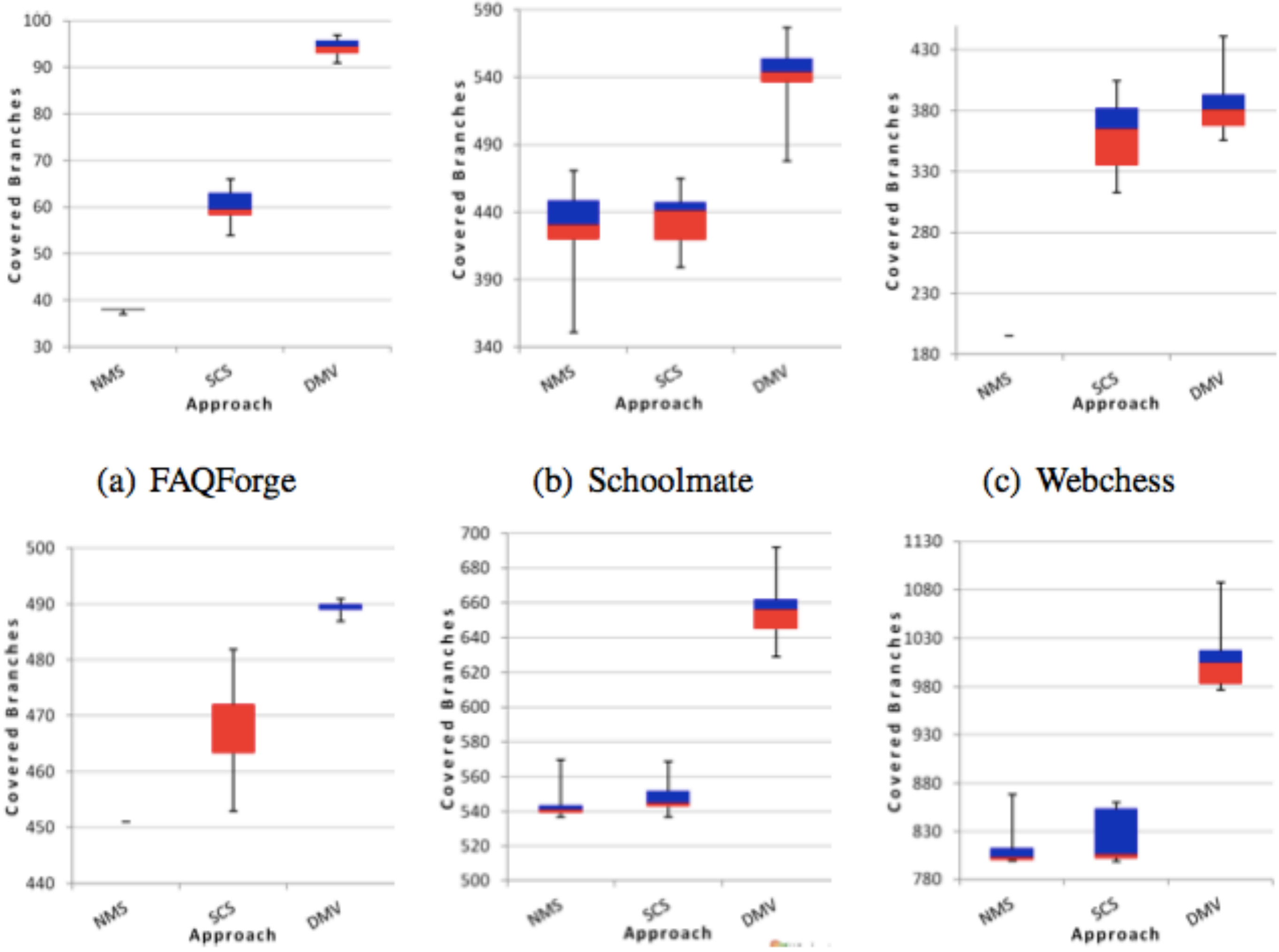
```
if (file_exists($lng.’.php’)) {
..}
```



Template: classic

Language: en

Evaluation - Coverage



(a) FAQForge

(b) Schoolmate

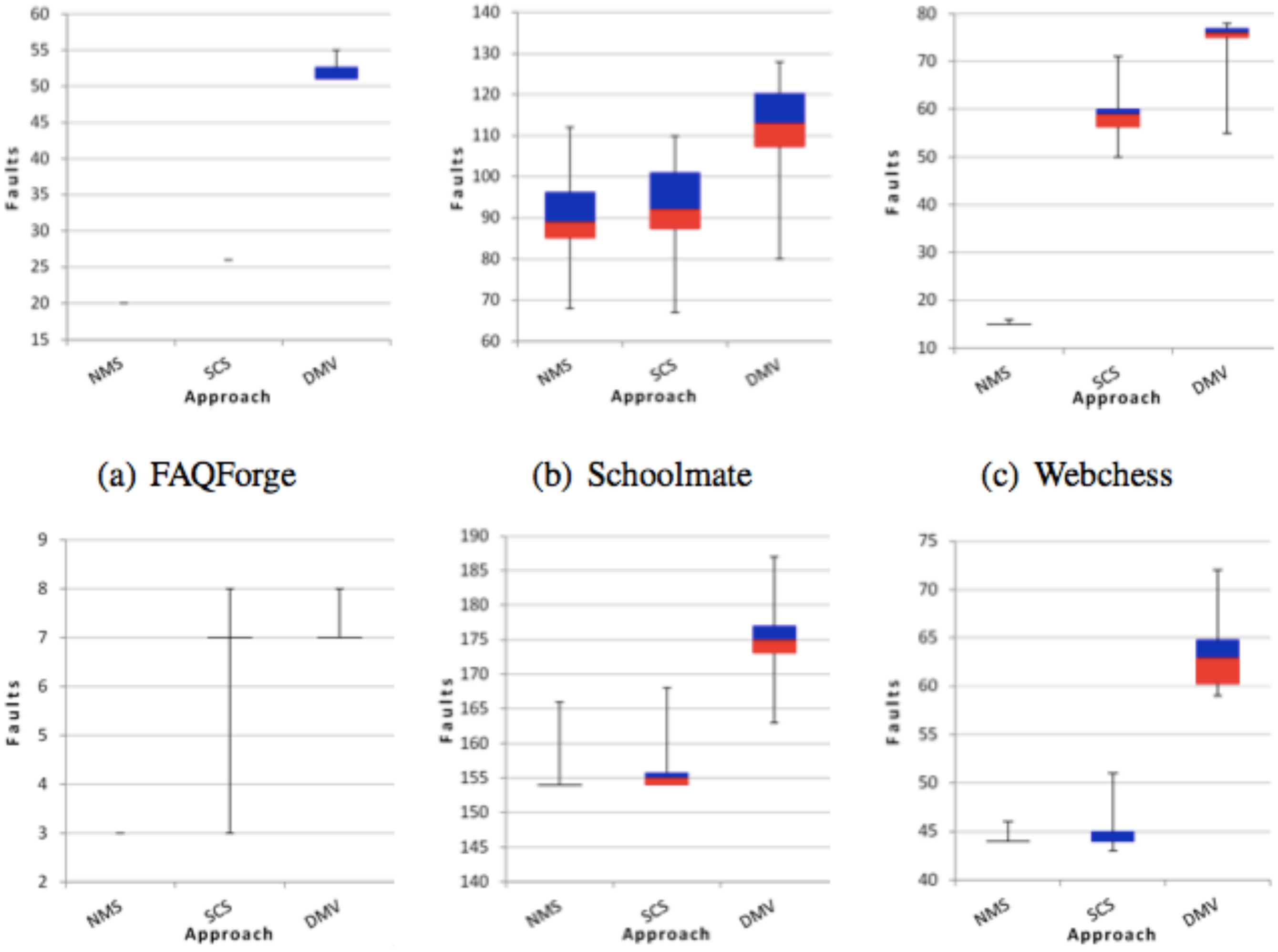
(c) Webchess

(d) PHPSysInfo

(e) Timeclock

(f) PHPBB2

Evaluation - Faults



(a) FAQForge

(b) Schoolmate

(c) Webchess

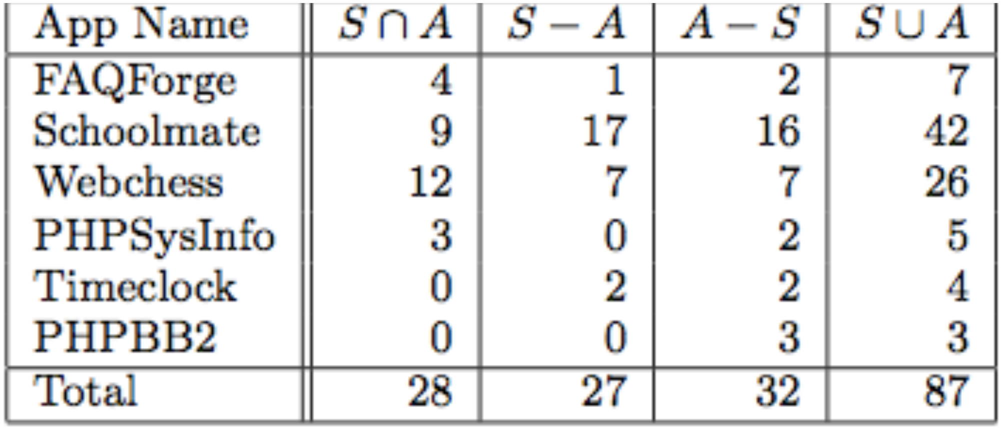
(d) PHPSysInfo

(e) Timeclock

(f) PHPBB2

SBST vs DSE

Bugs found by SBST (SWAT) and DSE (APOLLO)



App Name	$S \cap A$	$S - A$	$A - S$	$S \cup A$
FAQForge	4	1	2	7
Schoolmate	9	17	16	42
Webchess	12	7	7	26
PHPSysInfo	3	0	2	5
Timeclock	0	2	2	4
PHPBB2	0	0	3	3
Total	28	27	32	87

Results

- Impact of seeding higher with string predicates
- Constant seeding might mislead the search
- Test suites with the same coverage perform differently in fault detection

Results

- In some applications coverage is low (20%), similar results for DSE
- Branches that are not covered:
  - Database dependent
  - Environment dependent: time, OS, browser..etc
  - Configuration: infeasible?
  - Multi-user dependent

Future Directions

- Easy to apply, better performance
- Different goals not just coverage
- Oracle problem: automated might not be possible but reduce the cost