

Università  
della  
Svizzera  
italiana



# Python test generator

**Paolo Tonella** (Software Institute, Università della Svizzera italiana, Lugano, Switzerland)

# Goal of the project

*Write a search based automated test generator for Python. The generator shall maximize condition coverage of the functions under test and will be compared against a random fuzzer used as baseline.*

1. Write an instrumentation script that transforms the Python code under test to enable computation of the coverage fitness function
2. Develop a fuzzer that generates new test cases randomly or by mutating/crossing over previously created tests
3. Use the library Deap to define a genetic algorithm that evolves test case inputs so as to maximize condition coverage
4. **Use the tool MutPy to inject artificial faults (mutations) into the benchmark functions under test and evaluate the fault detection capability of the genetic algorithm, considering the random fuzzer as baseline**

# MutPy

<https://pypi.org/project/MutPy/>

```
mut.py --target example.py --unit-test example_tests.py

[*] Start mutation process:
  - targets: example.py
  - tests: example_tests.py
[*] 4 tests passed:
  - example_tests [0.00027 s]
[*] Start mutants generation and execution:
  - [# 1] COI example: [0.00626 s] survived
  - [# 2] COI example: [0.00592 s] killed by test_f_3 (example_tests.Test_example)
  - [# 3] COI example: [0.00586 s] killed by test_f_2 (example_tests.Test_example)
  - [# 4] COI example: [0.00501 s] survived
  - [# 5] COI example: [0.00592 s] killed by test_f_2 (example_tests.Test_example)
  - [# 6] CRP example: [0.00495 s] survived
  - [# 7] CRP example: [0.00558 s] survived
  - [# 8] CRP example: [0.00493 s] survived
  - [# 9] CRP example: [0.00529 s] survived
  - [# 10] ROR example: [0.00538 s] survived
  - [# 11] ROR example: [0.00508 s] survived
  - [# 12] ROR example: [0.00606 s] killed by test_f_3 (example_tests.Test_example)
  - [# 13] ROR example: [0.00523 s] survived
  - [# 14] ROR example: [0.00590 s] killed by test_f_2 (example_tests.Test_example)
  - [# 15] ROR example: [0.00513 s] survived
  - [# 16] ROR example: [0.00506 s] survived
  - [# 17] ROR example: [0.00551 s] survived
  - [# 18] ROR example: [0.00559 s] killed by test_f_2 (example_tests.Test_example)
  - [# 19] ROR example: [0.00528 s] survived
[*] Mutation score [0.17480 s]: 31.6%
  - all: 19
  - killed: 6 (31.6%)
  - survived: 13 (68.4%)
  - incompetent: 0 (0.0%)
  - timeout: 0 (0.0%)
```

# Mutation score

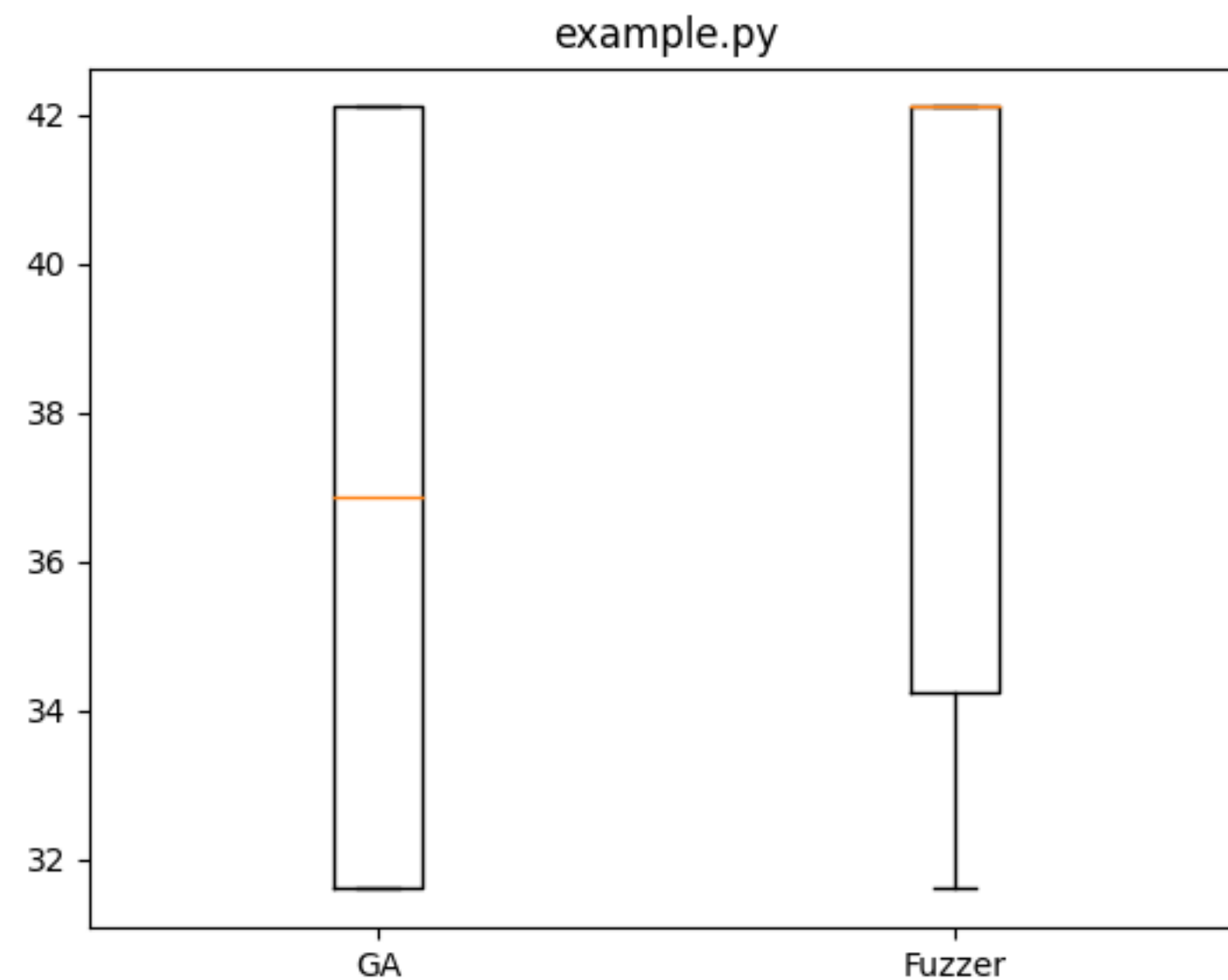
Run MutPy:

```
stream = os.popen(f'mut.py --target {py_file} --unit-test {py_test_file}')  
output = stream.read()
```

Collect mutation score:

```
re.search('Mutation score \[.*\]: (\d+\.?\d+)\%', output).group(1)
```

# Statistical comparison



Mean mu-score:

GA = 36.85 Fuzzer = 38.95

Effect size: -0.41 (small)

Wilcoxon's p-value: 0.31

## Experimental procedure:

- For each benchmark program P
  - Repeat the following experiment N times (e.g., with N = 10):
    - Generate random test cases for P using the GA generator
    - Measure the mutation score for P
    - Generate search based test cases for P using the Fuzzer
    - Measure the mutation score for P
  - Visualize the N mutations score values of Fuzzer and GA using boxplots
  - Report the average mutation score of Fuzzer and GA
  - Compute the effect size using the Cohen's d effect size measure
  - Compare the N mutation score values of Fuzzer vs GA using the Wilcoxon statistical test

# Python test generator

**Paolo Tonella**

*Software Institute, Università della Svizzera italiana, Lugano, Switzerland*

