



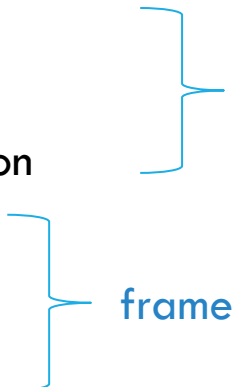
# TUNING AN AUTOMATED TEST CASE GENERATOR -- EVOSUITE

CS-HU 374 Lecture 9

# PARAMETERS TUNING

ATG tools depend on several parameters

EvoSuite has different groups of parameters

- Population size
  - Crossover rate
  - Fitness function calculation
  - Search time
  - Test suite size
  - Length of each test suite
- 
- genetic algorithm parameters
- framework
- The diagram uses blue curly braces to group parameters. The first brace groups 'Population size', 'Crossover rate', and 'Fitness function calculation', with the label 'genetic algorithm parameters' to its right. The second brace groups 'Search time', 'Test suite size', and 'Length of each test suite', with the label 'framework' to its right.

# NO FREE LUNCH THEOREM (NFL)\*

Is it possible to find an optimal parameter setting?

- Find the best values for many parameters
- Use them for TC generation for all programs

Formally proven that it is not possible

- Among all programs some parameter values do better than default some do worse
- On average there is no outstanding value combination

Parameter values should be tuned from default settings

Objective: find good parameter settings for specific problem types

\*Wolpert, D.H., Macready, W.G. "No free lunch theorems for optimization", IEEE Trans on Evolutionary Computation, 1997

# EVOSUITE HAS MANY PARAMETERS

Explore EvoSuiteParams excel in “files”

- all options tab with at least 350 algorithm-related parameters (some are inter-depended)
- investigate tab – one we will work with today

search_budget	long	60	Maximum search duration
Parameter name	Value type	Default value	Description

coverage criterion. Can define more than one criterion by using a ':' separated list

[LINE,METHOD, METHODNOEXCEPTION, CBRANCH]

Criterion[]

```
public static void main(String[] args) {  
    String[] evoArgs = { "-class", "code.Class34", "-projectCP", "./bin/", "-Dcriterion=branch", "-Dsearch_budget=10", };  
    String[] helpArgs = {"-listParameters"};  
    EvoSuite.main(helpArgs);  
}
```

# *PARAMETER TUNING OR DEFAULT VALUES?* BY A. ARCURI AND G. FRASER, SSBSE 2011

## Findings:

- Different parameter settings cause very large variance in the performance.
- Default parameter setting perform relatively well, but are far from optimal on individual problem instances.
- Tuned parameters (for the set of problems) can improve upon default values on average.
- The available search\_budget has the strongest impact on the parameter settings that should be used.
- Tuning on a large set of classes improves performance, the improvement is low.
- No significant improvements on real-world programs, but tuning is more expensive.

## Conclusions:

- Tuning can have a strong impact: not properly tuned parameters results in worsened from default performance.
- Ok to use default params from the literature vs setting them randomly

# DESIGNING AN EXPERIMENT

## Framework parameters

- search\_budget=10
- criterion=branch
- max\_length = 25 (Maximum length of a test suite)

## TCG parameters

- dynamic\_seeding
- dynamic\_pool
- dynamic\_size
- random\_seed

## Search parameters

- algorithm
- alternative\_fitness\_calculation\_mode
- population
- selection\_function

Let's go over them in the spreadsheet

# IN-CLASS WORK P5: TUNING EVOSUITE

Update the repository: week5 -> w5\_code

Use coverage provided by EvoSuite

Class11, Class34 and EvoSuiteArgs (the driver)

1. Fix framework parameters (set search\_budget to a small number)
2. Run EvoSuite with other parameters as default on Class11. Coverage – cov1?
3. Pick one parameter from each of two categories (2 total) and try to change their values to increase the coverage in Class11 comparing to default. Coverage – cov2?
4. Run EvoSuite with (other) default parameters on Class34. Coverage – cov3?
5. Run EvoSuite with the best parameters for Class11 on Class34. Coverage – cov4?
6. Report cov1, cov2, cov3 and cov4.
  1. Compare default coverage between two programs: cov1 vs cov2
  2. By how much cov2 does improve cov1? Describe your effort that you put to improve the coverage.
  3. Compare cov3 vs. cov4. Do fine—tuned parameters for Class11 result in better coverage for Class34?

# ASSIGNMENT 3

## Work tuning Evosuite

- Evaluate its default behavior on two programs
- Try to tune-up Evosuite to improve its coverage
  - Explore 3 parameters
  - Tune them for each program
- Document your experience

Will be posted after the class

Due on Sunday (hard deadline – grades are due on the following Tuesday)

## Advice of the week:

Even with 100% covered branches a program can have a bug.





# LAST CLASS

Other aspects and types of testing

Last 30 minutes of class – write a reflection on testing approaches learned in class and what we can do to improve the class, what worked what did not.

**Please complete the course evaluation!**