

# Topic 2: Evolutionary Software Testing

Bogdan Marculescu and **Man Zhang**

November 21, 2019, MS340 Emerging Technologies 2019

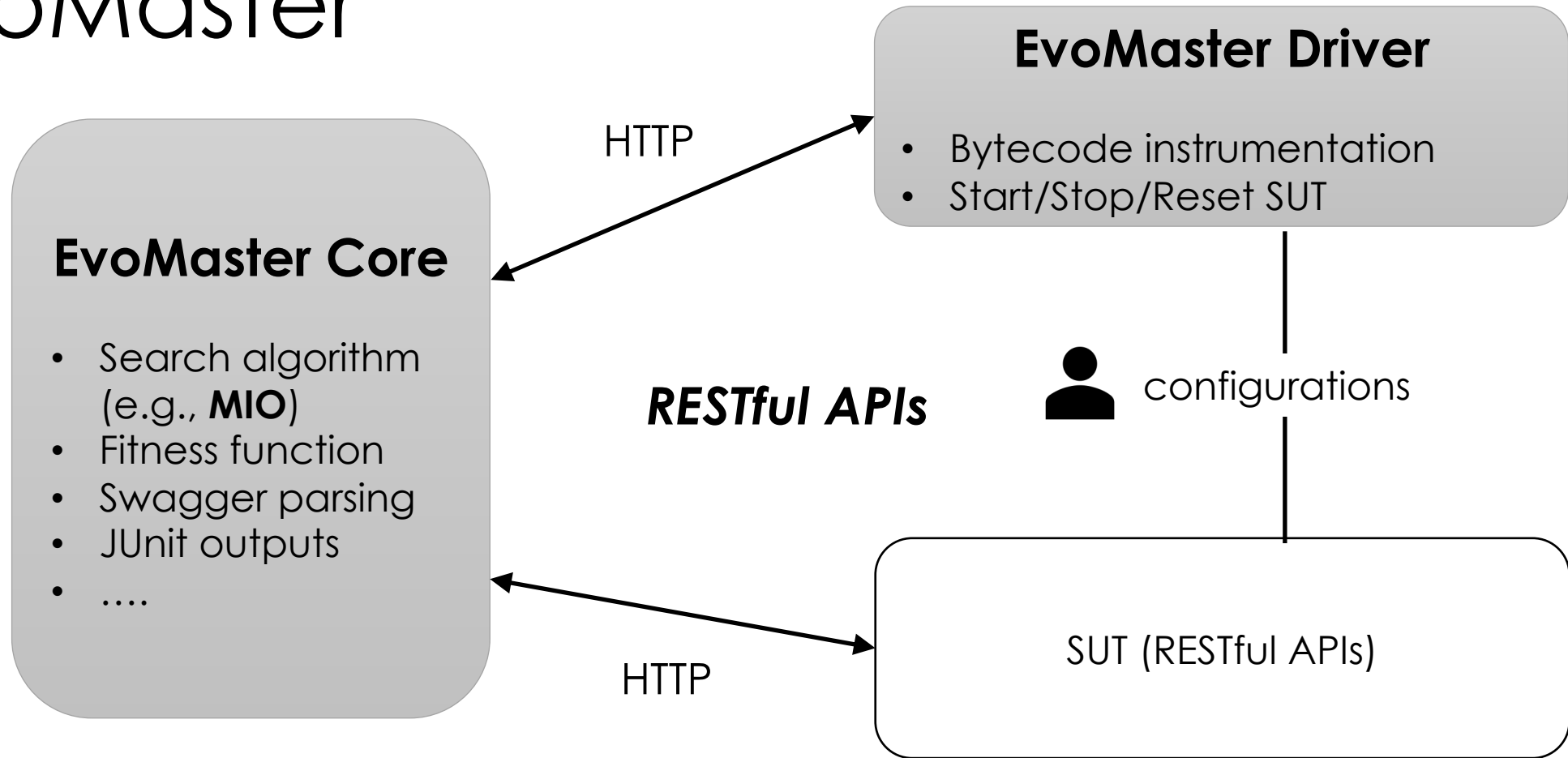
# EvoMaster

-- A Tool for Automatically Generating System-Level Test Cases

# EvoMaster

- Tool for automatically generating tests for REST APIs
- White box testing
  - code instrumentation which enables to get runtime info, e.g., line coverage
- Search-based testing
  - Many independent Objective (**MIO**)<sup>[1]</sup> Algorithm
  - MOSA...
- Open source [www.evomaster.org](http://www.evomaster.org)

# EvoMaster



Many independent Objective (**MIO**)<sup>[1]</sup> Algorithm is designed for system test case generation in the context of white box testing.

[1] Andrea Arcuri. 2018. Test suite generation with the Many Independent Objective (MIO) algorithm. *Information and Software Technology (IST)* 104 (2018), 195--206.

---

**Algorithm 1:** Pseudo-code of the MIO Algorithm [5]

---

**Input** : Stopping condition  $C$ , Fitness function  $\delta$ , Population size  $n$ , Probability for random sampling  $P_r$ , Start of focused search  $F$

**Output**: Archive of optimised individuals  $A$

```

1  $T \leftarrow SetEmptyPopulations()$ 
2  $A \leftarrow \{\}$ 
3 while  $\neg C$  do
4   if  $P_r > rand()$  then
5      $p \leftarrow RandomIndividual()$ 
6   else
7      $p \leftarrow SampleIndividual(T)$ 
8      $p \leftarrow Mutate(T)$ 
9   end
10  foreach element  $k$  of  $ReachedTargets(p)$  do
11    if  $IsTargetCovered(k)$  then
12       $UpdateArchive(A, p)$ 
13       $T \leftarrow T \setminus T_k$ 
14    else
15       $T_k \leftarrow T_k \cup \{p\}$ 
16      if  $|T_k| > n$  then
17         $RemoveWorstTest(T_k, \delta)$ 
18      end
19    end
20  end
21   $UpdateParameters(F, P_r, n)$ 
22 end

```

**Sampling with  $P_r$**

**Mutation with  $(1 - P_r)$**

**Fitness Evaluation**

# MIO Algorithm

- Kind of a multi-population (1+1)EA
- Many objectives
  - testing targets (e.g., lines or a branch to be covered)
  - a population (a set of individuals) for a target
- Dynamic number of populations of tests
- output a solution
  - Best individuals in each population

# Examples

- Constant Problem (general problem)
- Branch (white box testing)
  - EMController
- EMB
  - <https://github.com/EMResearch/EMB>

# Constant problem

- Seek a number which is 'close' to a target within a given budget
  - e.g., target = 123

- **Individual**
- **Fitness function**
- Operator
  - Sampler
  - Mutation
- Termination

Individual is composed of **one integer gene**

$$\text{distance} = |\text{target} - \text{value}|$$
$$F = 1.0 - (\text{distance} / (1.0 + \text{distance}))$$

**Optimal is F = 1.0**



# Constant problem cont.

- Seek a number which is equal with a target within a given budget
  - e.g., target = 123
- Individual
- Fitness function
- **Operator**
  - Sampler
  - Mutation
- **Termination**

sample an individual with an integer generated randomly

modify the integer to be mutated  
(e.g., +1 or -1 with a probability)

a certain number of fitness evaluation  
a certain time

# white box testing

- Seek a test suite (a set of tests) which achieves a '*high*' coverage of codes within a given budget
- Individual

An Individual is composed of **a set of test**.  
A test is sequence of methods/HTTP calls.

- Code snippet
  - pos (x,y) : int
    - If  $x > 0$  **return** 0
    - If  $y \geq 0$  **return** 1
    - **return** 2
  - neg (x,y) : int
    - If  $x < 0$  **return** 3
    - If  $y \leq 0$  **return** 4
    - **return** 5
  - eq (x,y) : int
    - If  $x == 0$  **return** 6
    - If  $y != 0$  **return** 7
    - **return** 8

# individual

```
1. testA(){  
2.     pos(-1, 0) // pos method, x= -1  
3. }
```

```
1. testB(){  
2.     pos(-5, 0) // pos method, x= -5  
3.     neg(-5, 0) // neg method  
4. }
```

# White box testing

- Seek a test suite (a set of tests) which achieves a '*high*' coverage of codes within a given budget
- Individual
- **Fitness function**
- Operator
  - Sampler
  - Mutation
- Termination

## Many testing targets

covered/reached lines, branches, classes

Faults detection

...

# Heuristics: Branch Distance

- `pos (x,y) : int`
  - If `x > 0` **return** 0
  - If `y >= 0` **return** 1
  - **return** 2

For example,  
Test A {`pos(-1, 0)` }  
Test B {`pos(-5, 0)`}

Branch `y >= 0` is **covered** by both A and B

Branch `x > 0` is not covered but **reached** by both A and B.

A is **closer** to the target than B. thus,  
 $H(A) > H(B)$

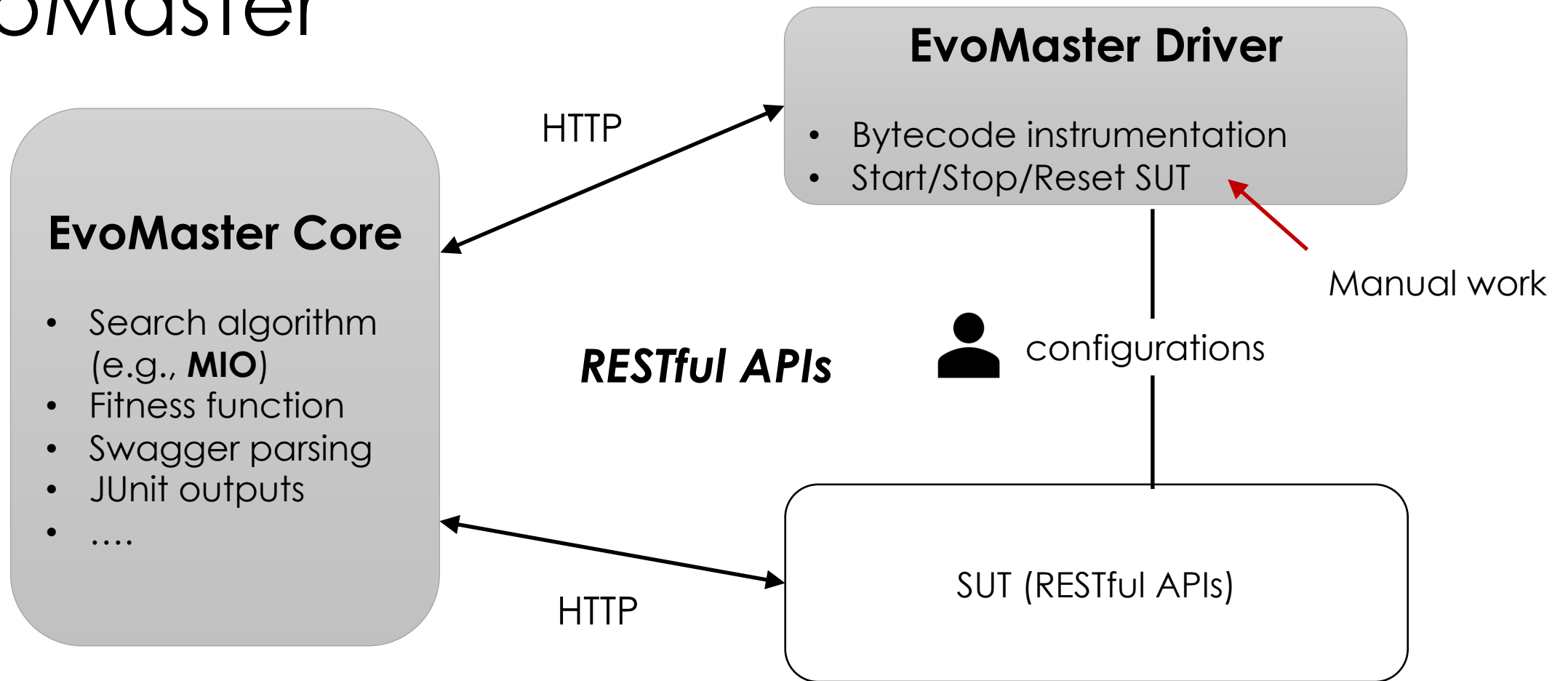
# White box testing cont.

- Seek a test suite (a set of tests) which achieves a '*high*' coverage of codes within a given budget
- Individual
- Fitness function
- Operator
  - Sampler
  - Mutation
- Termination

sample a test with a set of HTTP calls generated randomly

Mutate genes with a probability

# EvoMaster



Our main goal is to generate tests **automatically**.

# EMController

```
public abstract class XXX extends EmbeddedSutController {  
  
    @Override  
    public String startSut(){}  
    @Override  
    public boolean isSutRunning{}  
    @Override  
    public void stopSut{}  
    @Override  
    public String getPackagePrefixesToCover() {return "com.foo."; }  
    @Override  
    public void resetStateOfSUT() {}  
    @Override  
    public ProblemInfo getProblemInfo() {  
        return new RestProblem(  
            "http://localhost:" + getSutPort() + "/v2/api-docs",  
            null  
        );  
    }  
}
```



# EMController cont.

```
public abstract class XXX extends EmbeddedSutController {  
    ....  
    @Override  
    public List<AuthenticationDto> getInfoForAuthentication() {}  
    // Database Handling  
    @Override  
    public Connection getConnection() {return null;}  
    @Override  
    public String getDatabaseDriverName() {return null;}  
  
    @Override  
    public SutInfoDto.OutputFormat getPreferredOutputFormat() {  
        return SutInfoDto.OutputFormat.JAVA_JUNIT_5;  
    }  
}
```

# Get started with EvoMaster

[www.evomaster.org](http://www.evomaster.org)

EMResearch / **EvoMaster** Watch 9 Unstar 27 Fork 13

[Code](#) [Issues 1](#) [Pull requests 1](#) [Projects 0](#) [Wiki](#) [Security](#) [Insights](#)

A tool for automatically generating system-level test cases

[testing](#) [evolutionary-algorithms](#) [rest](#) [java](#) [kotlin](#) [test-case-generation](#)

1,558 commits 26 branches 3 releases 5 contributors LGPL-3.0

Branch: master [New pull request](#) [Create new file](#) [Upload files](#) [Find File](#) [Clone or download](#)

# Get started with EvoMaster

- Download EvoMaster [www.evomaster.org](http://www.evomaster.org)
  - Release version
    - <https://github.com/EMResearch/EvoMaster/releases>
  - JVM 8 bytecode (e.g., Java and Kotlin)
- Develop a driver
  - Start/stop system under test (sut)
  - ....

# Generating tests...

- Start the **driver**
- Start **EvoMaster** with the command line  
*java -jar evomaster.jar --help*  
*java -jar evomaster.jar --maxTime=20s*

Enjoy!

Report issues

- <https://github.com/EMResearch/EvoMaster/issues>

# Reference for EvoMaster

## 2019

- M. Zhang, B. Marculescu, A. Arcuri. *Resource-based Test Case Generation for RESTful Web Services*. ACM Genetic and Evolutionary Computation Conference (GECCO).
- A. Arcuri, J.P. Galeotti. *SQL Data Generation to Enhance Search-Based System Testing*. ACM Genetic and Evolutionary Computation Conference (GECCO).
- **A. Arcuri. *RESTful API Automated Test Case Generation with EvoMaster*. ACM Transactions on Software Engineering and Methodology (TOSEM). --- for REST**

## 2018

- **A. Arcuri. *Test Suite Generation with the Many Independent Objective (MIO) Algorithm*. Information and Software Technology (IST). --- For MIO**
- **A. Arcuri. *EvoMaster: Evolutionary Multi-context Automated System Test Generation*. IEEE Conference on Software Testing, Validation and Verification (ICST). --- For the tool**
- A. Arcuri. *An Experience Report On Applying Software Testing Academic Results In Industry: We Need Usable Automated Test Generation*. Empirical Software Engineering (EMSE).

# Reference cont.

## 2017

- A. Arcuri. *RESTful API Automated Test Case Generation*. IEEE International Conference on Software Quality, Reliability & Security (QRS).
- A. Arcuri. *Many Independent Objective (MIO) Algorithm for Test Suite Generation*. Symposium on Search-based Software Engineering (SSBSE). **Best paper award**.

## Seminars/Presentations

- 2019: *The Many Independent Objective (MIO) Algorithm for Test Suite Generation*. Lecture given at ISTI-CNR, Italy.
- 2019: *Testing RESTful Web Services with EvoMaster*. Lecture given at the 3rd International Genoa Software Engineering PhD School on Automated Functional and Security Testing of Web and Mobile Application, Italy.
- 2019: *Using Evolutionary Algorithms to Test Software*. Lecture given at Kristiania University College, Norway.
- 2018: *EvoMaster: Evolutionary Multi-context Automated System Testing*. Seminar given at Mälardalen University, Sweden.
- 2017: *EvoMaster: Evolutionary Multi-context Automated System Testing*. Seminar given at the University of Luxembourg, Luxembourg.

# More ...

- Feedback-directed sampling
  - care about what targets are covered than how close to the target to be covered
- Adaptive exploration/exploitation control
- Sampling techniques for RESTful Web Services
- ...

# demo

You can find all codes online

- [www.evomaster.org](http://www.evomaster.org)

Constant problems

- **core** module
  - Test: `org.evomaster.core.search.algorithms.constant`

Branch

- **spring-examples** module under **e2etest**
  - Find *branches* package
    - Java: source code of sut
    - Test: controller and tests

More examples can be found in

<https://github.com/EMResearch/EMB>