# Improving Search-based Test Data Generation

**Project Report**

**Group 6**

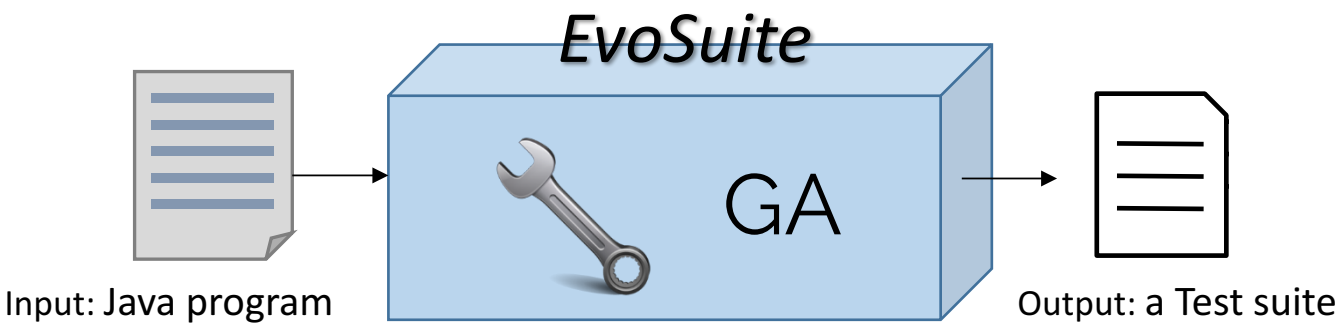
**Mingyu Jin, Youlim Jung, Jiyoung Song**

**Dec 13th, 2016**

# Search-based Test Data Generation

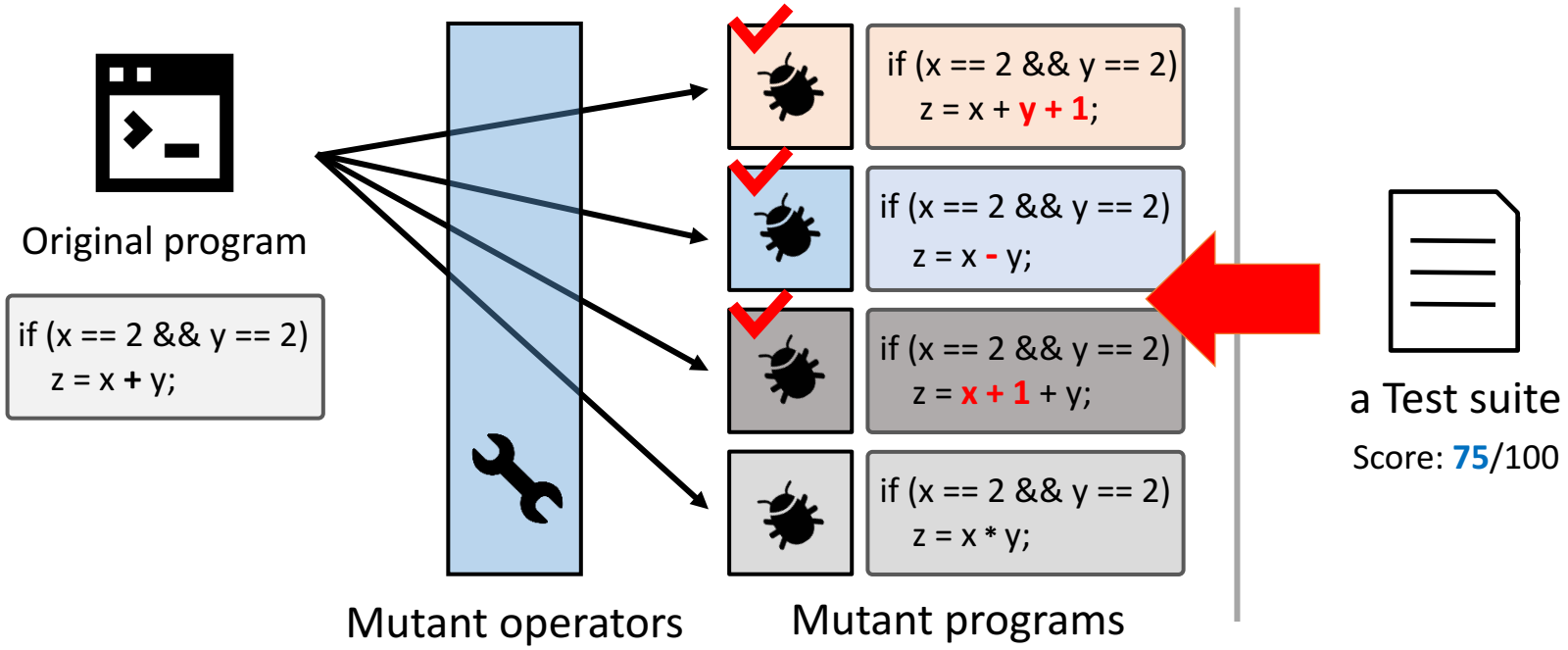
- **Automation in test data generation**
  - It is a bothersome and error-prone task.
  - The search-based technique is one way to automate it.

- ***EvoSuite*, automated test generation tool**
  - Exploits the genetic algorithm to generate test data automatically.
  - Supports various kinds of criteria (coverage of line, branch, method, exception, and etc.) including mutation criterion.



# Mutation Criterion

- **Assumption**
  - The better a test suite is in detecting faults in mutants, the more it is powerful to detect real faults.
- **How to evaluate a test suite under this criterion?**
  - Simply, the number of the killed mutants is used.



# Diversity-Aware Mutation Criterion

- **Limitation of traditional mutation criterion**
  - focuses on how many mutants are killed by the test suite.
  - fails to leverage the diversity of mutants.
- **To fully exploit the diversity**
  - Shin et al.[1] proposed the new mutation criterion distinguishing mutants along their kill-patterns.
  - The diversity info. can raise fault detection capability.

| Test | m1 | m2 | m3 | m4 |
|---|---|---|---|---|
| t1 | 1 | 1 | 1 | 1 |
| t2 | 0 | 1 | 1 | 1 |
| t3 | 0 | 0 | 0 | 1 |

* 1: kill, the test killed the mutant
  0: alive, the mutant survived against the test

[1] D. Shin, S. Yoo, and D.H. Bae, "Diversity-Aware Mutation Adequacy Criterion for Improving Fault Detection Capability," Software Testing, Verification and Validation Workshops (ICSTW), 2016.

# Motivation & Goal

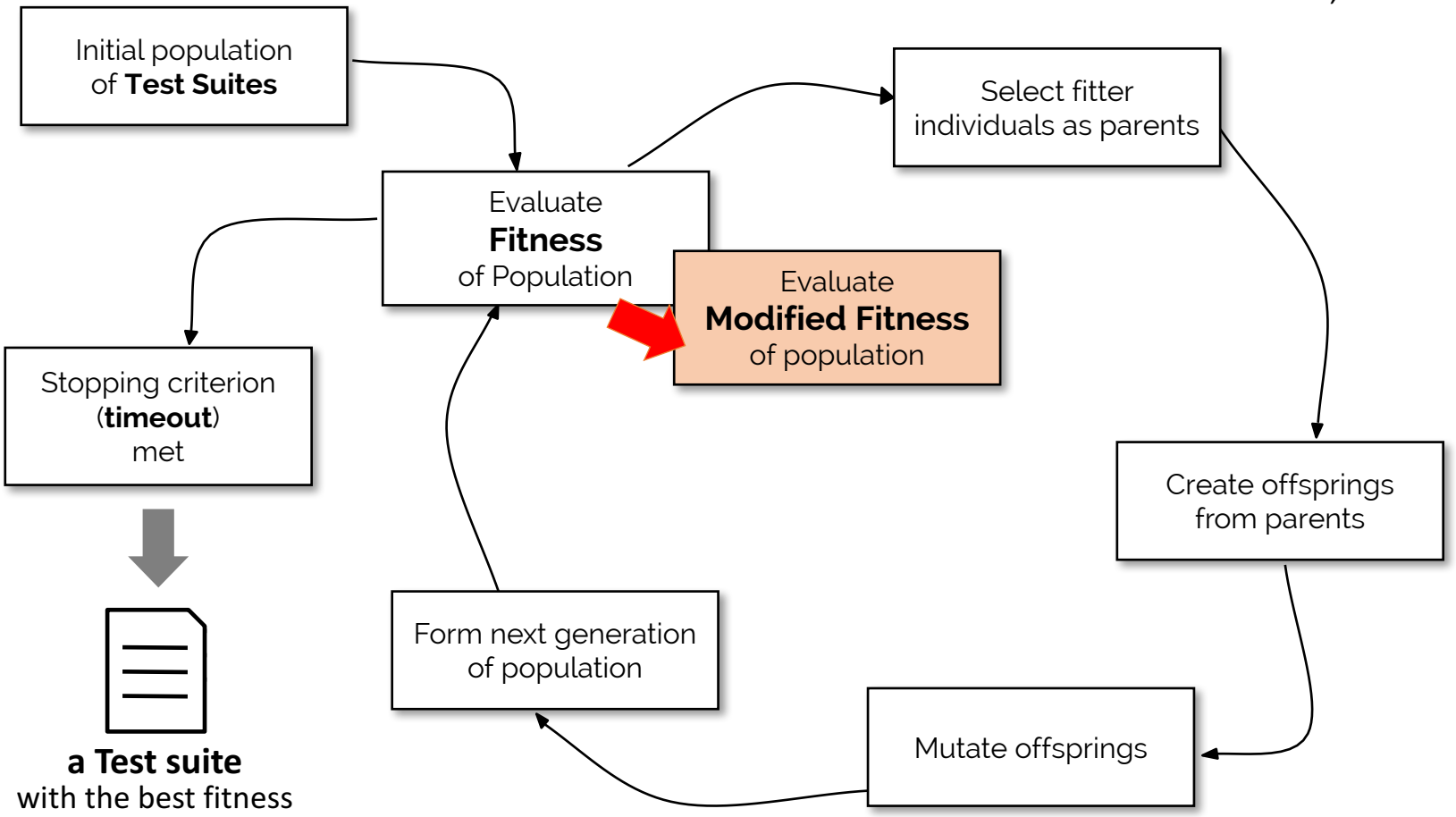
- **Motivation**
  - Shin et al. insist the diversity should be considered to raise fault detection capability in mutation criterion.
  - EvoSuite, a automated tool, currently uses kill-based criterion only.

- **Goal**
  - To redesign the fitness function, guiding EvoSuite to utilize the diversity information to produce a better test suite.
    - Adding one more dimension to the fitness (diversity)
  - To see this concept is adaptable to real world program.
    - Expected to be more effective (in fault detection capability)

# Overall Approach

**EvoSuite**: 224,318 SLOC

**GA-related**: 113,357 SLOC



# Existing Fitness Function

- **Structure**
  - Minimization problem (e.g. 0 is better than 999)

fitness = Branch fitness + Traditional mutation fitness

- **Branch fitness**
  - Depends on the number of reached predicates/methods
    - Every unreached predicate or method got a 1 penalty, otherwise (reached) 0.
- **Traditional mutation fitness**
  - Depends on how and how many mutants are killed
  - Sums up the penalties of every mutant.
    - Alive penalty: 3 > Weakly killed penalty: 1~2 > Strongly killed penalty: 0~1

# New Fitness Function

- **Structure**
  - Minimization problem (e.g. 0 is better than 999)

fitness = Branch fitness + Traditional fitness + Kill-based fitness / Diversity fitness

- **Diversity fitness**
  - depends on how many mutants are distinguished

| Test | m1 | m2 | m3 | m4 |
|---|---|---|---|---|
| t1 | 1 | 1 | 1 | 1 |
| t2 | 0 | 1 | 1 | 1 |
| t3 | 0 | 0 | 0 | 1 |

* 1: kill, the test killed the mutant
0: alive, the mutant survived against the test

**In this case,**
Kill-based Fitness: (4 – 4) = **0**
Diversity Fitness: (4 – 3) = **1**

* Kill-based fitness also needs to be computed only for comparison with Diversity fitness.

# Experiment Setup (1/3)

- **Fitness functions**

| Fitness | Function Formula |
|---|---|
| Original | Branch fitness + Traditional fitness |
| KA (K-fitness, additional) | Branch fitness + Traditional fitness + Kill-based fitness |
| DA (D-fitness, additional) | Branch fitness + Traditional fitness + Diversity fitness |
| KO (K-fitness, only) | ~~Branch fitness + Traditional fitness +~~ Kill-based fitness |
| DO (D-fitness, only) | ~~Branch fitness + Traditional fitness +~~ Diversity fitness |

- **Five different fitness functions**
  - **Original** provides the reference value.
  - In **KA** and **DA**, we added Kill-based fitness or Diversity fitness to the original fitness function.
  - In **KO** and **DO**, we removed the original fitness function to see the pure impact of newly proposed fitnesses.

# Experiment Setup (2/3)

- **Target programs**

| Program | Description | SLOC | # Mutants |
|---|---|---|---|
| Pattern | Finds a string match | 67 | 193 |
| Sort | Collection of bubble, selection, insertion sorts | 43 | 259 |
| Statistic | Computes statistics info. of an integer array | 57 | 273 |
| Triangle | Decides the form of triangle with sides length info. | 58 | 291 |
| Stack | Simple implementation of stack data structure | 20 | 50 |

- **Where do those programs come from?**
  - We've collected those programs from textbooks, Wikipedia, or tutorials.
- **To measure fault detection capability of a test suite**
  - We manually made 5 real faulty versions for each program.

# Experiment Setup (3/3)

- **Environment**
  - **Microsoft Azure**
  - CPU: Quardcores, 2.4GHz Intel Xeon® E5-2673 v3(Haswell)
  - RAM: 14GB
  - OS: Ubuntu Server 16.04 (LTS)

- **Repetition**
  - We've run 30-times every combination of

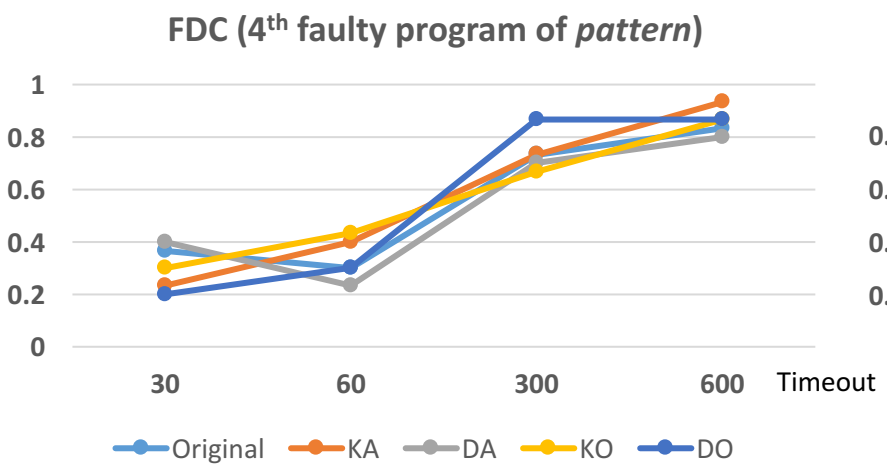
    (Target Programs) **X** (Fitness Function) **X** (Faulty Versions)

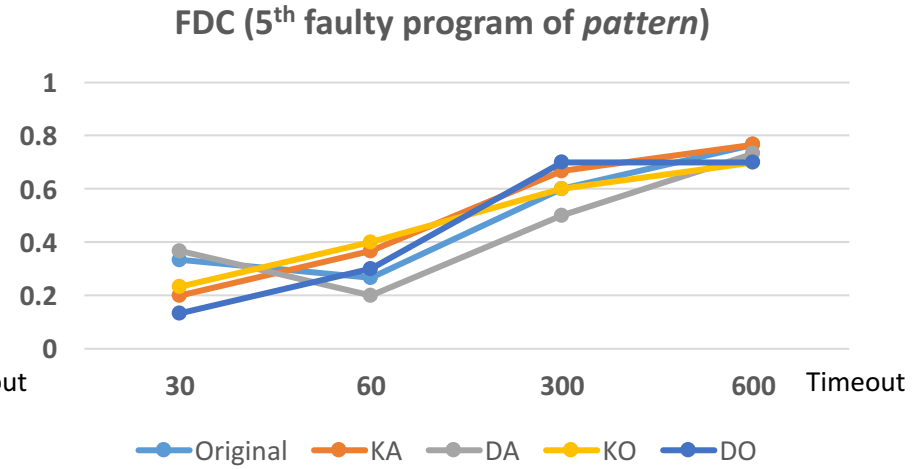
# Research Questions

- **RQ1. Timeout issue**
  - How much time would be enough for a test suite to evolve?
    - It would be unfair to compare premature test suites.
- **RQ2. Test suite size**
  - How different are sizes of the test suites generated by different fitness functions?
    - Diversity fitness may lead to produce a larger test suite, requiring more test cases to distinguish mutants.
- **RQ3. Fault Detection Capability**
  - How effective are test suites generated by different fitness functions, in terms of fault detection capability?
    - The referred paper said, test suites considering the diversity of mutants shows higher effectiveness.

# RQ1. Timeout Issue

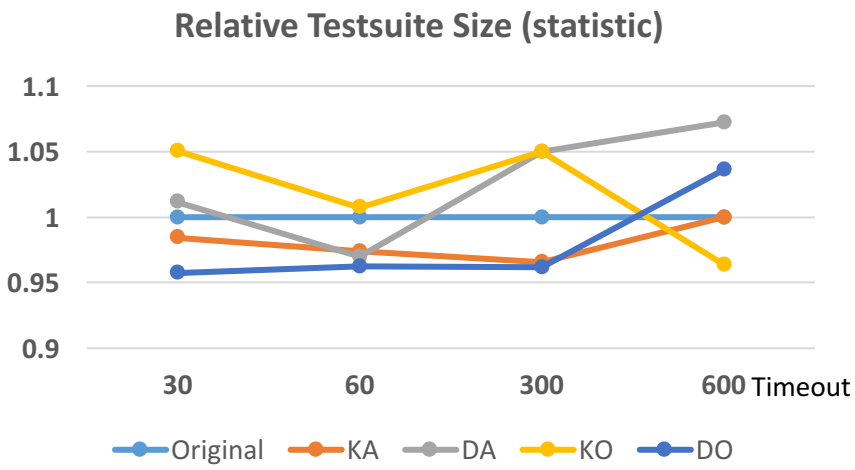
- **Before evaluating fitness functions**
  - We have to give enough time for test suites to evolve under each fitness function.
  - Matureness can be seen through fault detection capability (FDC) as different timeouts given.
  - ➢ The elbow of the graph is at about 600 seconds.

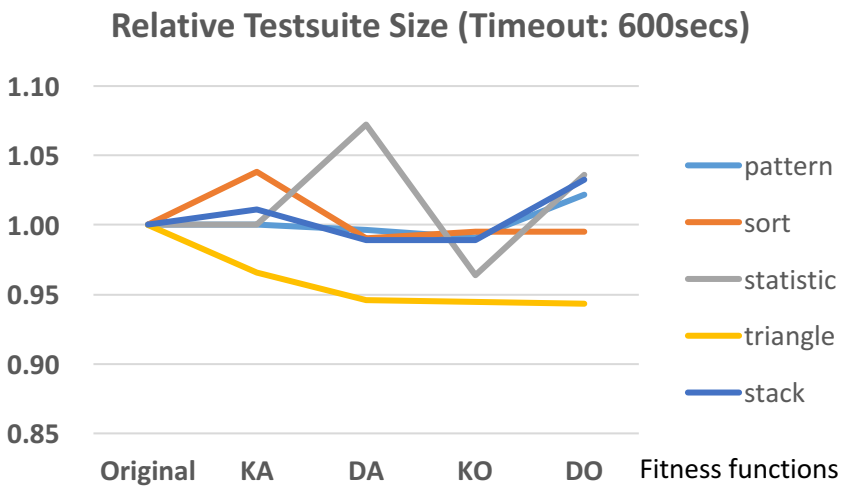


# RQ2. Test suite size

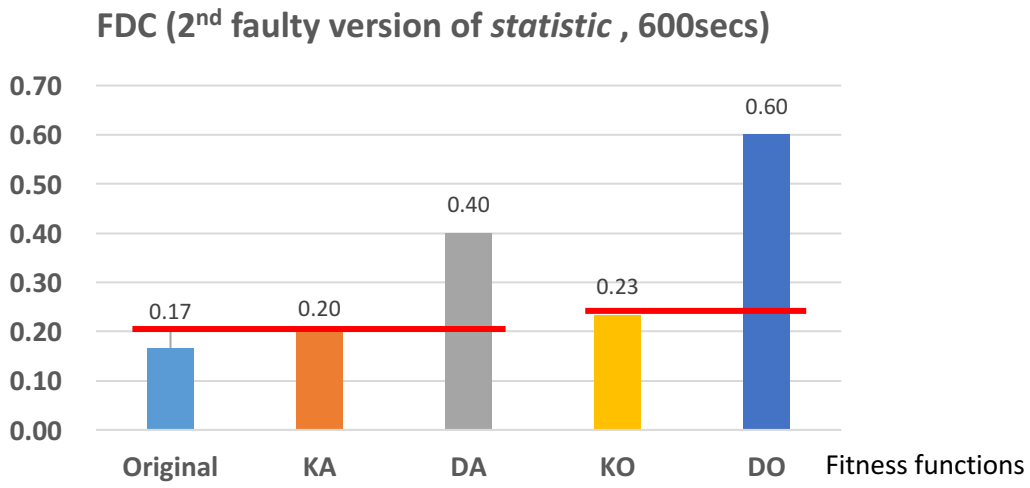
- **It is said, the diversity may enlarge a test suite size.**
  - However, there seems no correlation with fitness functions neither by time nor by programs.
  - ➢ It may be because EvoSuite evolves a test suite as a whole, not adding test cases one by one.



# RQ3. Fault Detection Capability

- **Fault Detection Capability**
  - As timeout is given more,
    fault detection capability of **DA** and **DO** keep increasing
    compared to **Original**, **RK**, and **PK**.
  - ➢ The diversity-based ones have shown higher capability
    which is about 2X more than **Original**, **KA**, or **KO** have done.



# Conclusion

- **Contribution**
  - Improved the quality of test data generation in search-based approach, with a new fitness function.
  - Seen the diversity-aware mutation criterion is adaptable to a real world tool and search-based algorithm.

- **Threats to validity**
  - 600 secs timeout can be extended to see dramatic results.
  - The size of programs might have impact.
    - Need of analysis on programs with larger size and complexity
  - Weighting of three different fitnesses was too naïve.

Thank you