The University Of Sheffield.

Random or Genetic Algorithm Search for Object Oriented Test Suite Generation?

SINA SHAMSHIRI, JOSÉ MIGUEL ROJAS, GORDON FRASER, PHIL MCMINN

Automated Test Generation

Test Generation

```
Gecco gecco1 = new Gecco();
gecco1.isValid(1996);
gecco1.isValid(2013);
gecco1.isValid(2021);
```

```
public class Gecco{

    public boolean isValid(int year){

        if (year < 1999)
            return false;

        if (year <= 2015)
            return true;
        else
            return false;
    }

}
```

Test Generation

```
Gecco gecco1 = new Gecco();
assertFalse(gecco1.isValid(1996));
assertTrue(gecco1.isValid(2013));
assertFalse(gecco1.isValid(2021));
```

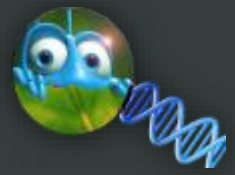
```
public class Gecco{

    public boolean isValid(int year){

        if (year < 1998)
            return false;

        if (year <= 2015)
            return true;
        else
            return false;
    }

}
```



Testful

an Evolutionary Testing Framework for Java

JTExpert

Tools

T3

Randoop

.class

EVOSUITE

Test Case

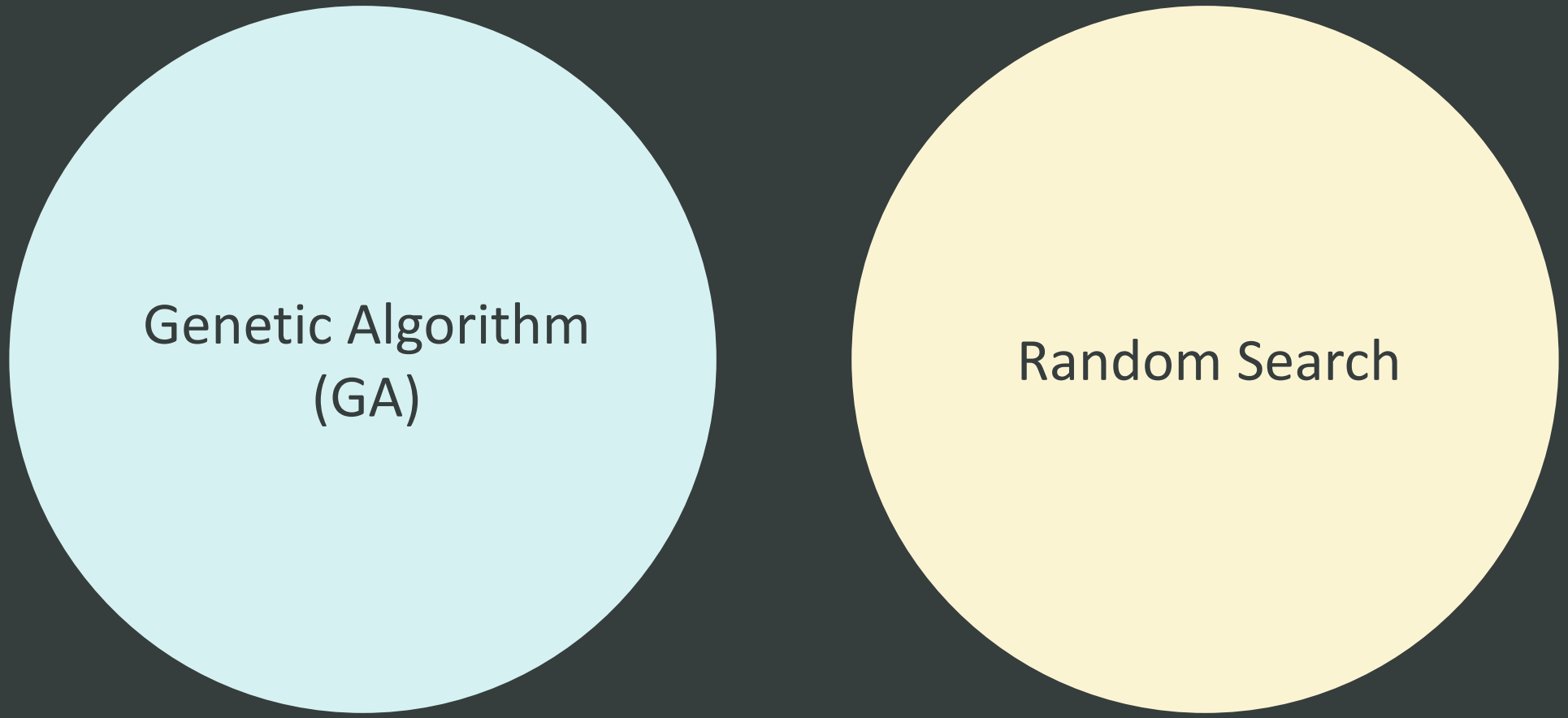
Test Case

Test Case

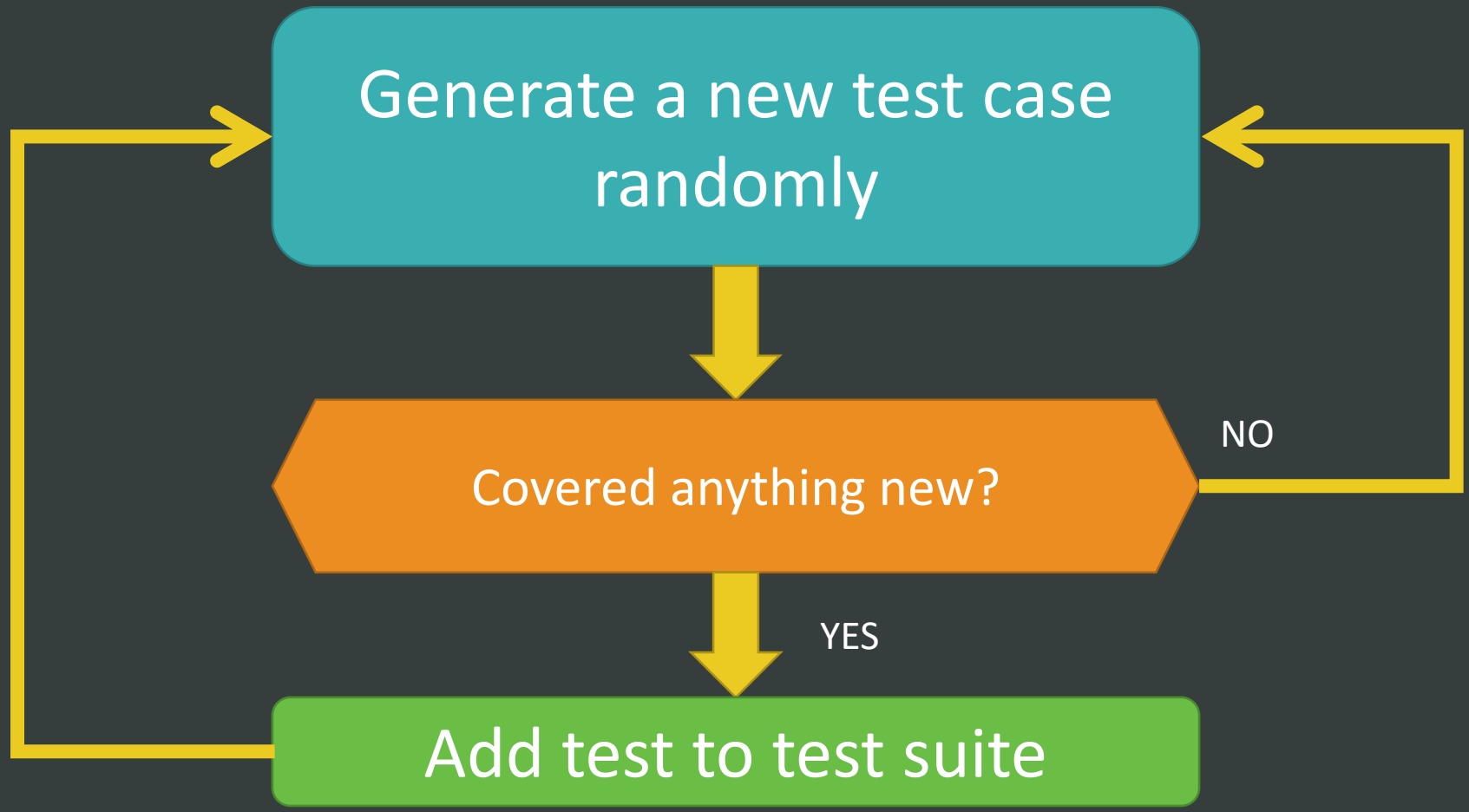
...

Coverage: 84%

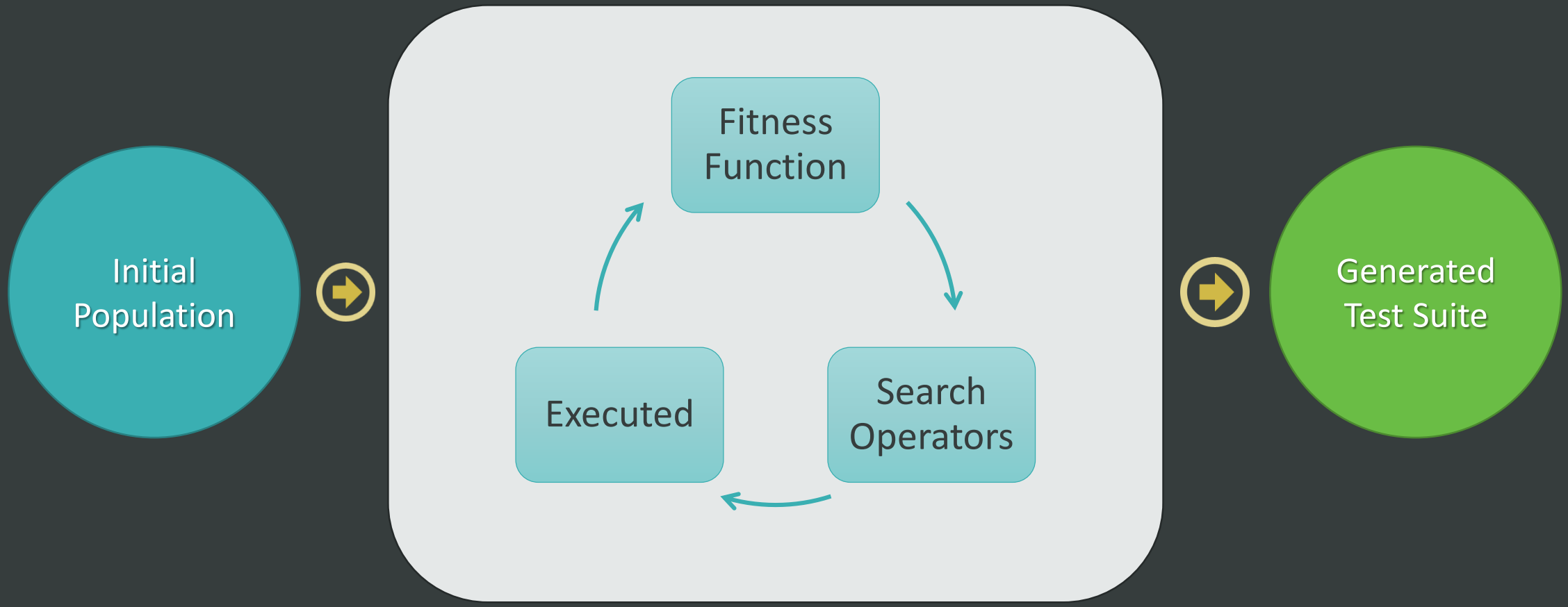
Search-Based Test Suite Generation



Random Search

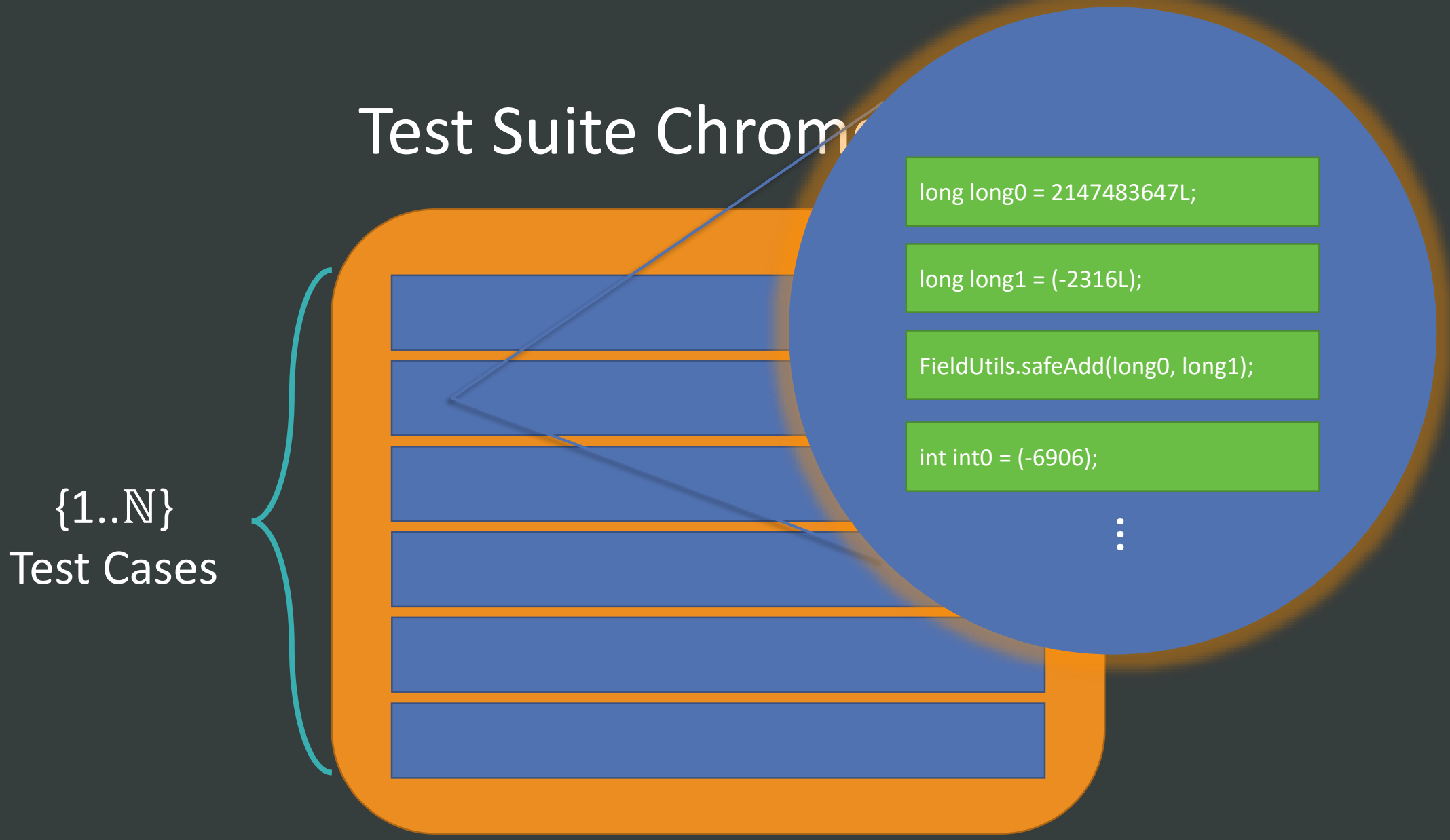


Genetic Algorithm (GA)

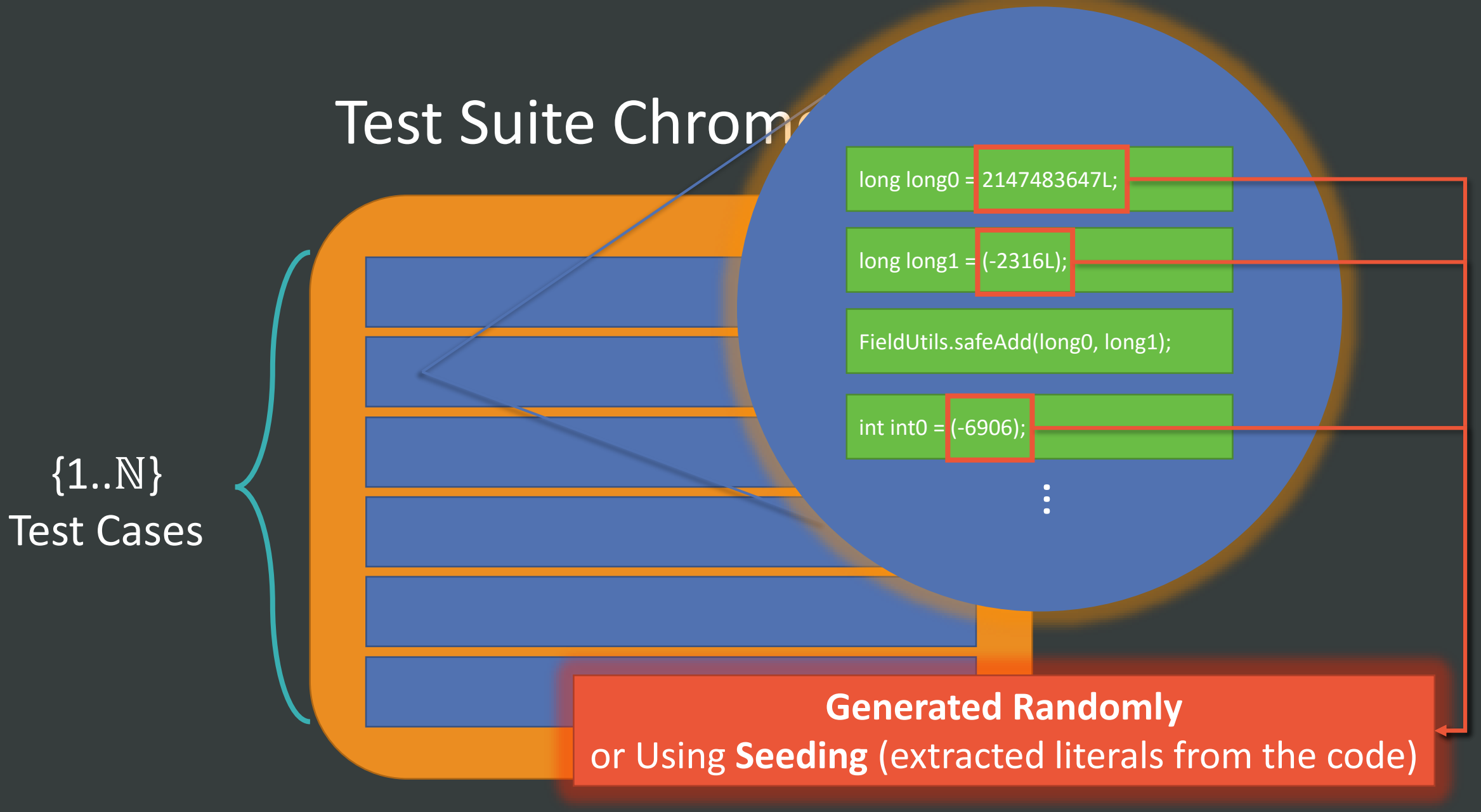


Test Suite Chromosomes

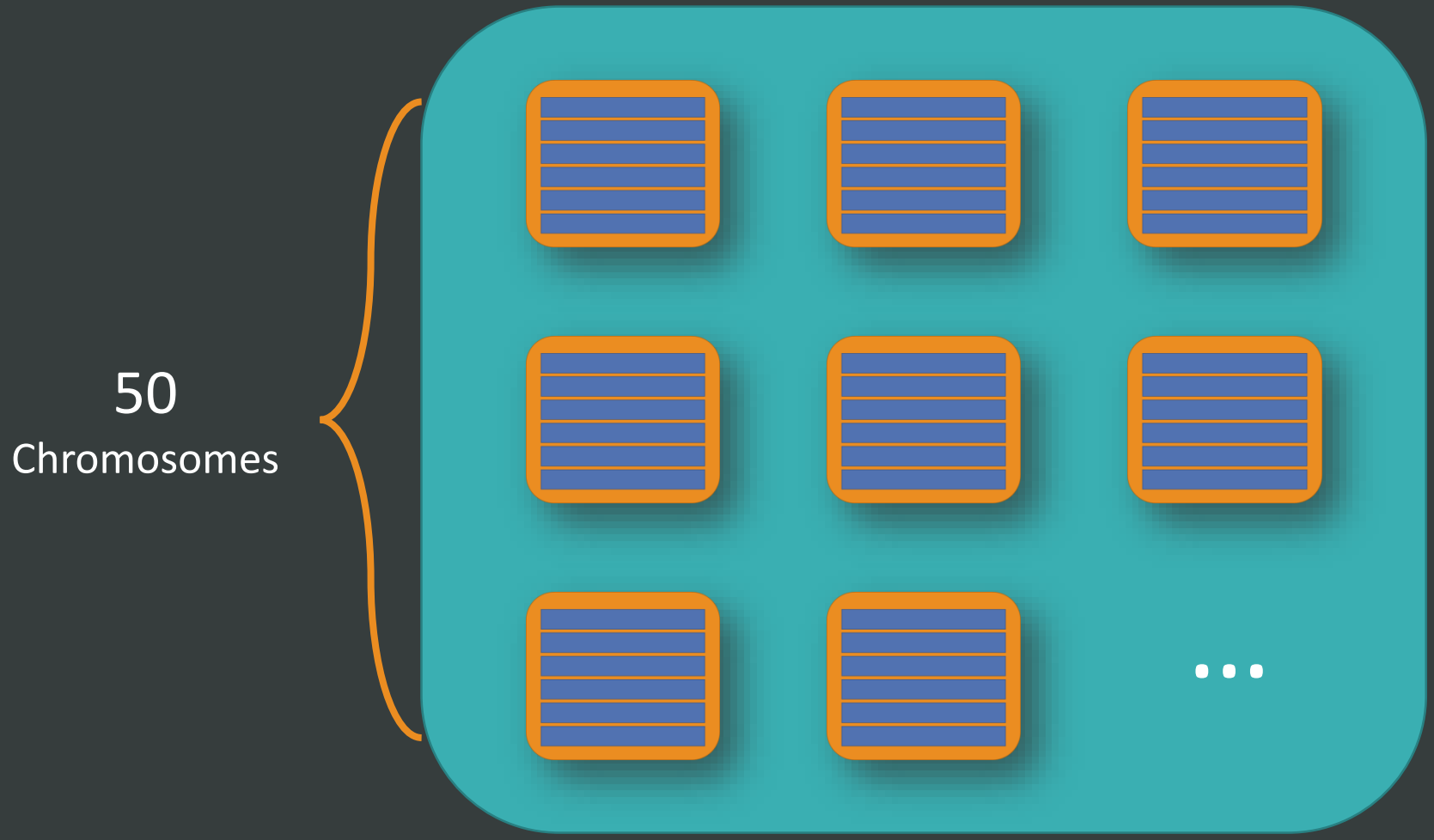
Test Suite Chrom



Test Suite Chrom



Population of Test Suite Chromosomes



Search Operators

Mutation

Crossover

Search Operators

Mutation Examples

- A test case is replaced with a new one
- A statement within a test is added, mutated or deleted

Crossover Example

- Two individual test suite chromosomes are combined to form two new offsprings

Fitness Function

Branch Coverage

Fitness Function

Coverage Goals

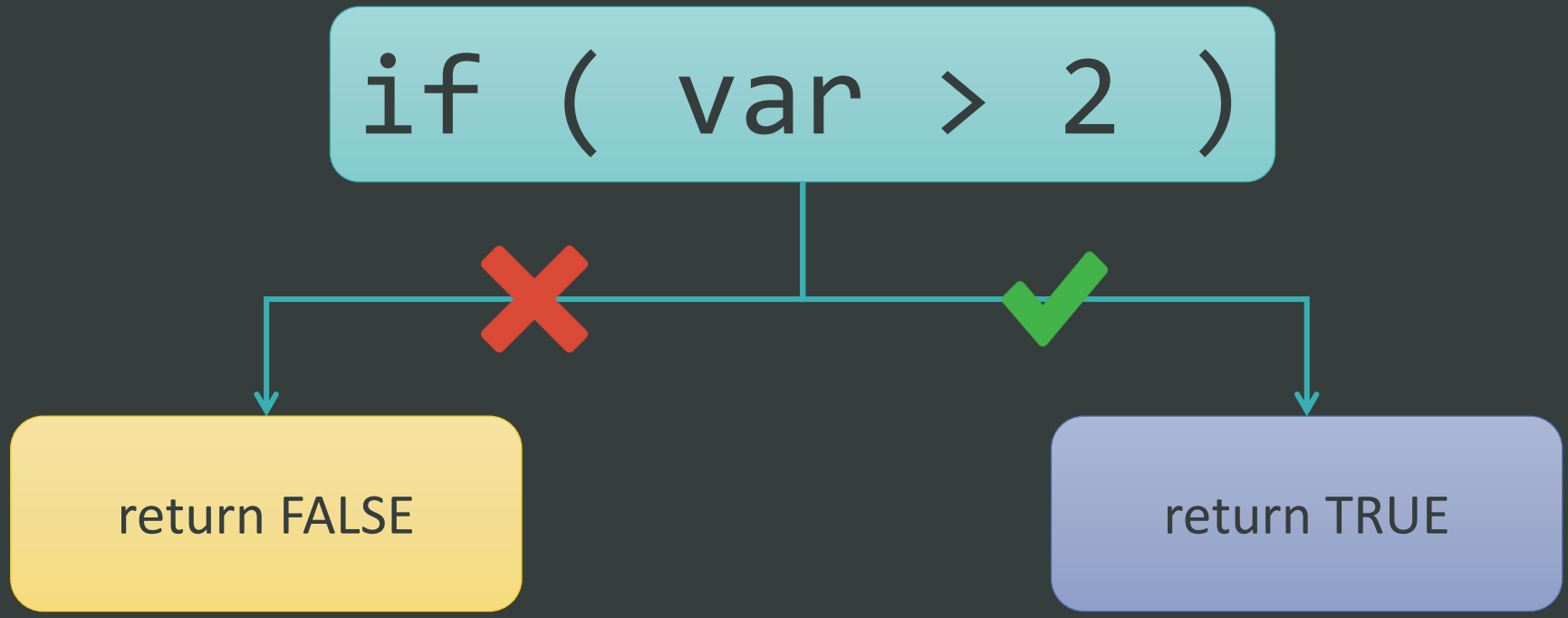
- All Branches
- All Branch-less Methods

Fitness Function

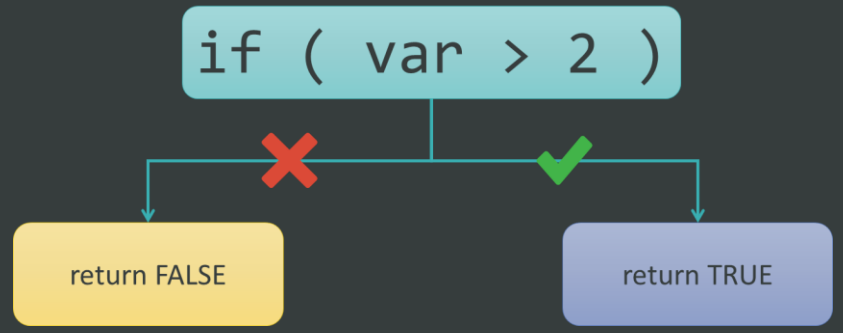
Calculation

Sum of the minimum distance value per branch

Branch Coverage



Branch Distance for Branch Coverage in GA



Distance Values

var = 1	0	$\lvert 1 - 3 \rvert = 2$
var = 2	0	$\lvert 2 - 3 \rvert = 1$
var = 3	$\lvert 3 - 2 \rvert = 1$	0

JAVA Bytecode

JAVA Bytecode

Source Code

Java Compiler

Byte Code
(.class)

JVM
(Java Virtual Machine)

Branch Types in JAVA Bytecode

Integer-Integer

```
if_icmpeq   if_icmpne
if_icmpgt   if_icmplt
if_icmpge   if_icmple
```

Reference-Reference

```
if_acmpeq
if_acmpne
```

Integer-Zero

```
Ifeq   ifne   iflt
ifgt   ifge   ifle
```

Reference-Null

```
ifnull
ifnonnull
```

Integer-Integer branches

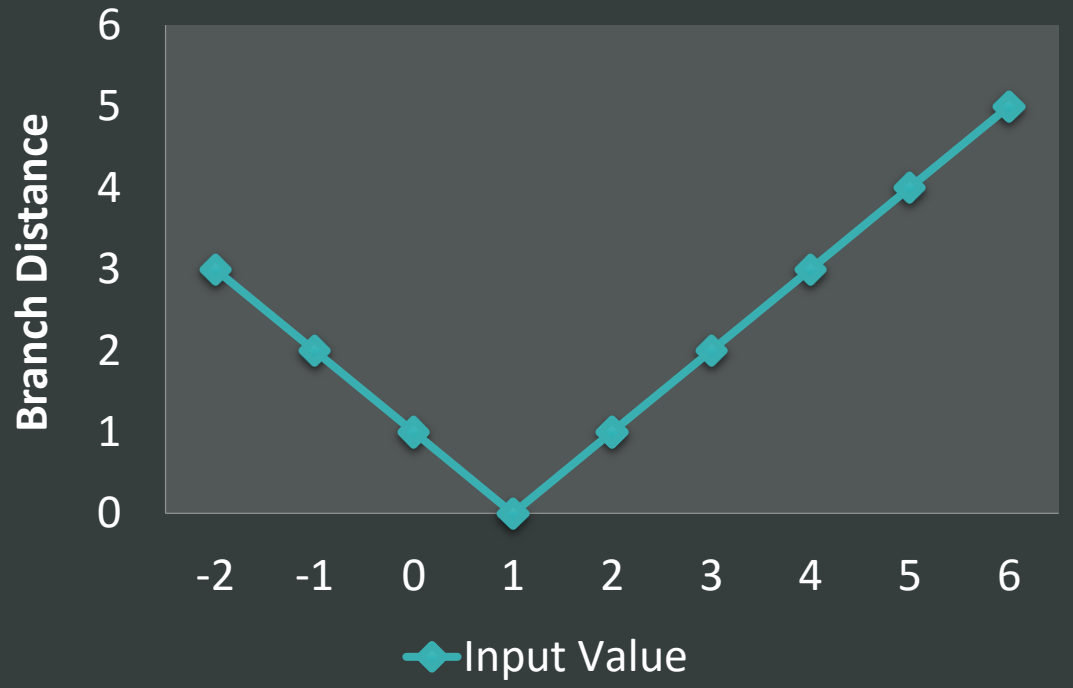
```
void m(int a){
  if (a == 1){
    // uncovered branch
  }
}
```

```
void m(int):

  0: iload_1
  1: iconst_1
  2: if_icompne 7
     [uncovered branch]
  7: return
```

Integer-Integer branches

```
void m(int a){
   if (a == 1){
      // uncovered branch
   }
}
```



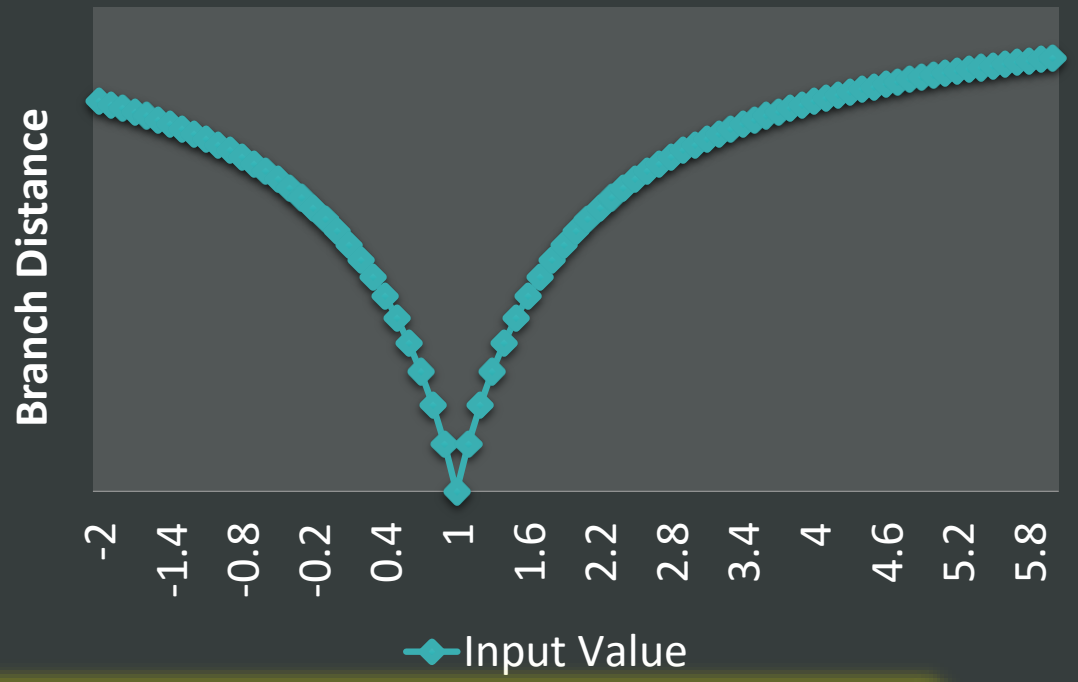
Integer-Zero branches

```
void m(int a){
   boolean x = false;
   if (a == 1) x = true;
   if (x){
     // uncovered branch
   }
}
```

**In Bytecode, boolean values are represented as 0 or 1 integers.*

Double, Float and Long Transformations

```
void m(double a){
   if (a == 1.0){
      // uncovered branch
   }
}
```



While they fall under the Integer-Zero bytocode branch type, in EvoSuite they are transformed to their respective values

Reference-Reference branches

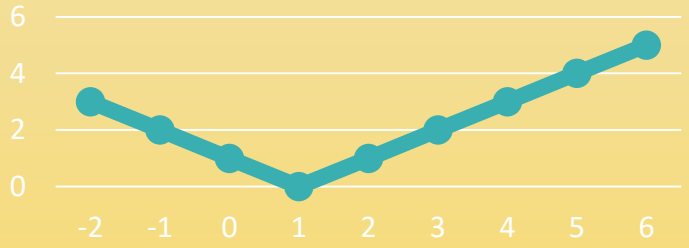
```
void m(int a){
    object x = null
    if (a == 1) x = this;
    if (this == x){
      // uncovered branch
    }
}
```

Reference-Null branches

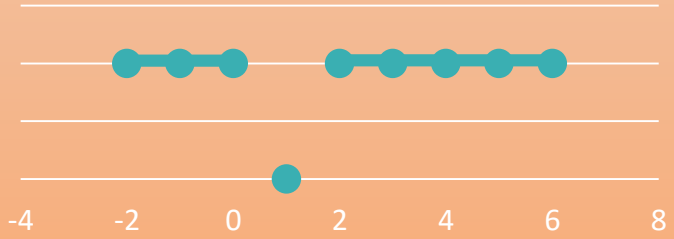
```
void m(int a){
    object x = null
    if (a != 1) x = new Object();
    if (x == null){
      // uncovered branch
    }
}
```

Branch Types in JAVA Bytecode

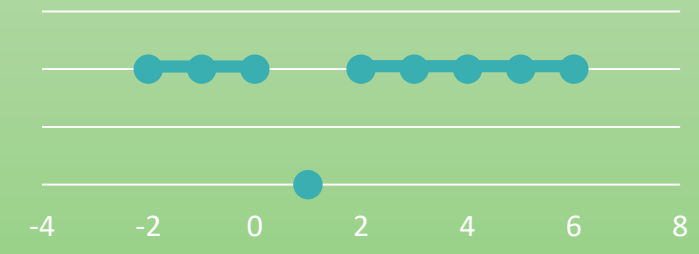
Integer-Integer



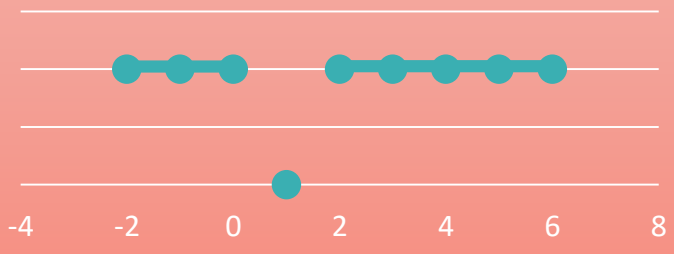
Reference-Reference



Integer-Zero

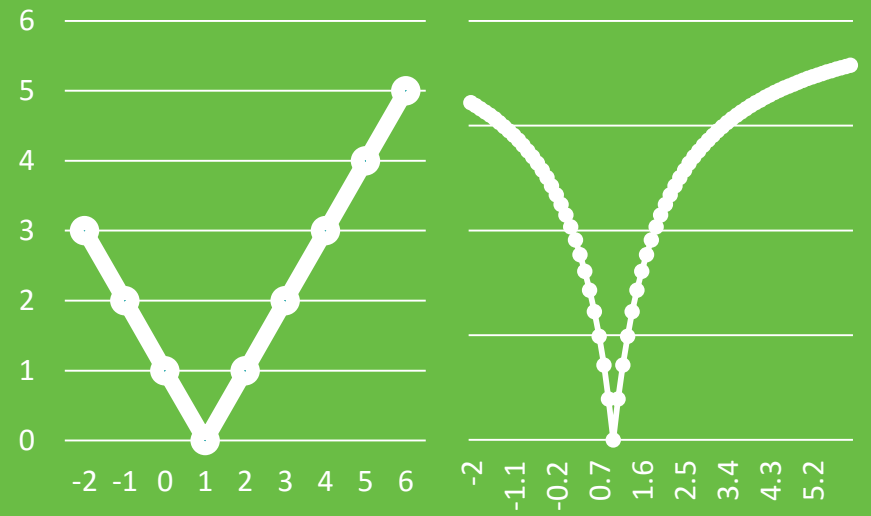


Reference-Null



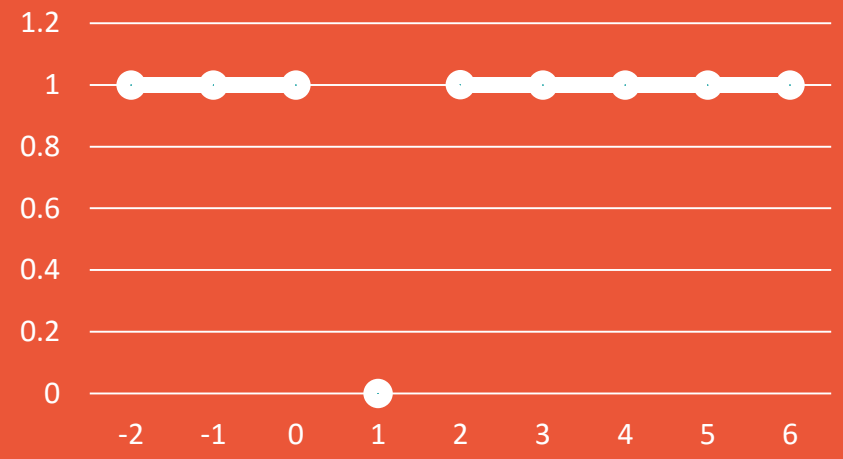
Summary of Branch Types

Gradient Branches



Can be used by the GA for guidance

Plateau Branches



Cannot be used by the GA for guidance

Research Questions

- RQ1. Is the use of a GA more effective at generating unit tests than random search?
- RQ2. How do the results of the comparison depend on the types of branches found in the code under test?
- RQ3. How do the results of the comparison depend on the time allowed for the search?

Study Subjects

- 1,000 classes selected randomly from the SF110 corpus of 110 open source projects selected from the sourceforge repository

G. Fraser and A. Arcuri, "A Large Scale Evaluation of Automated Unit Test Generation Using EvoSuite," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 24, 2014.

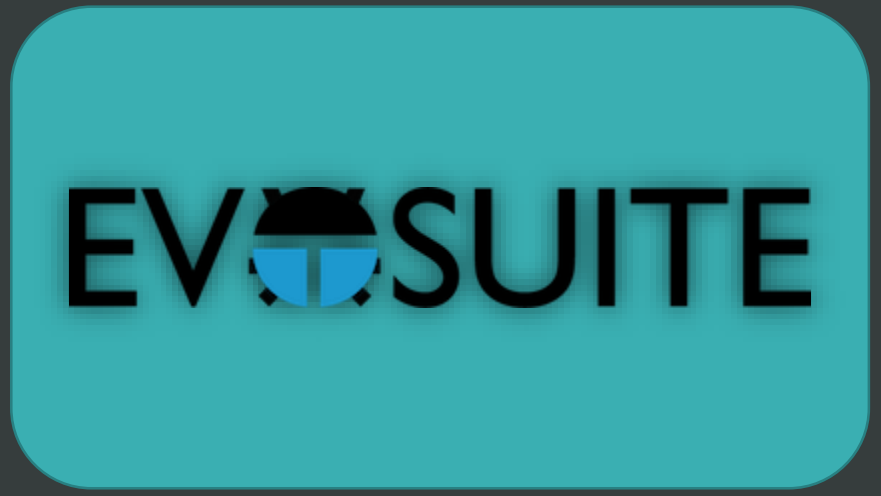
Study Subjects

	Min	Average	Max	Std. Deviation
Total Branches	0	26.91	1,020	79.2
Branchless Methods	0	7.2	155	11.4
Total Goals	1	34.1	1,030	84.0

- 22 classes excluded, since they either did not have testable methods (e.g., enumerated types or no public methods) or caused exceptions on load

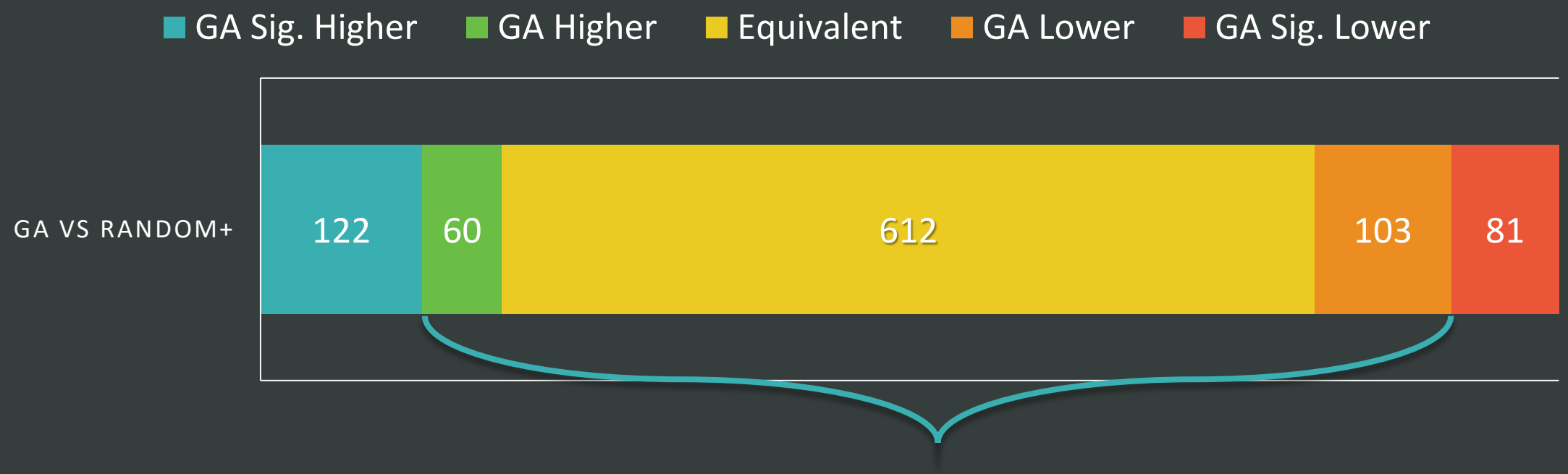
Experiments

- 50 repetitions for each experiment
- 2 minutes as the search budget for RQ1 and RQ2
- 10 minutes as the search budget for RQ3



- Branch-type statistics were collected during the search

RQ1: Coverage Effectiveness

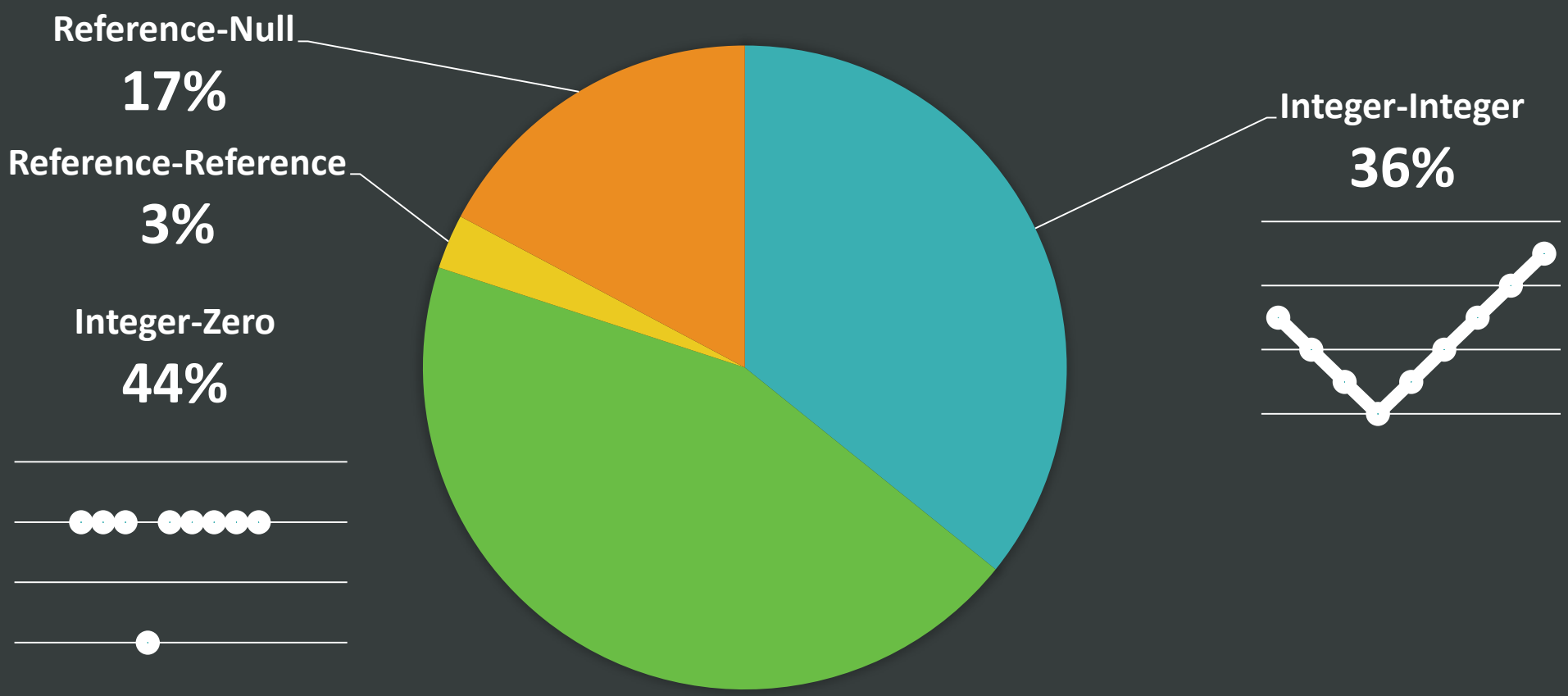


78%

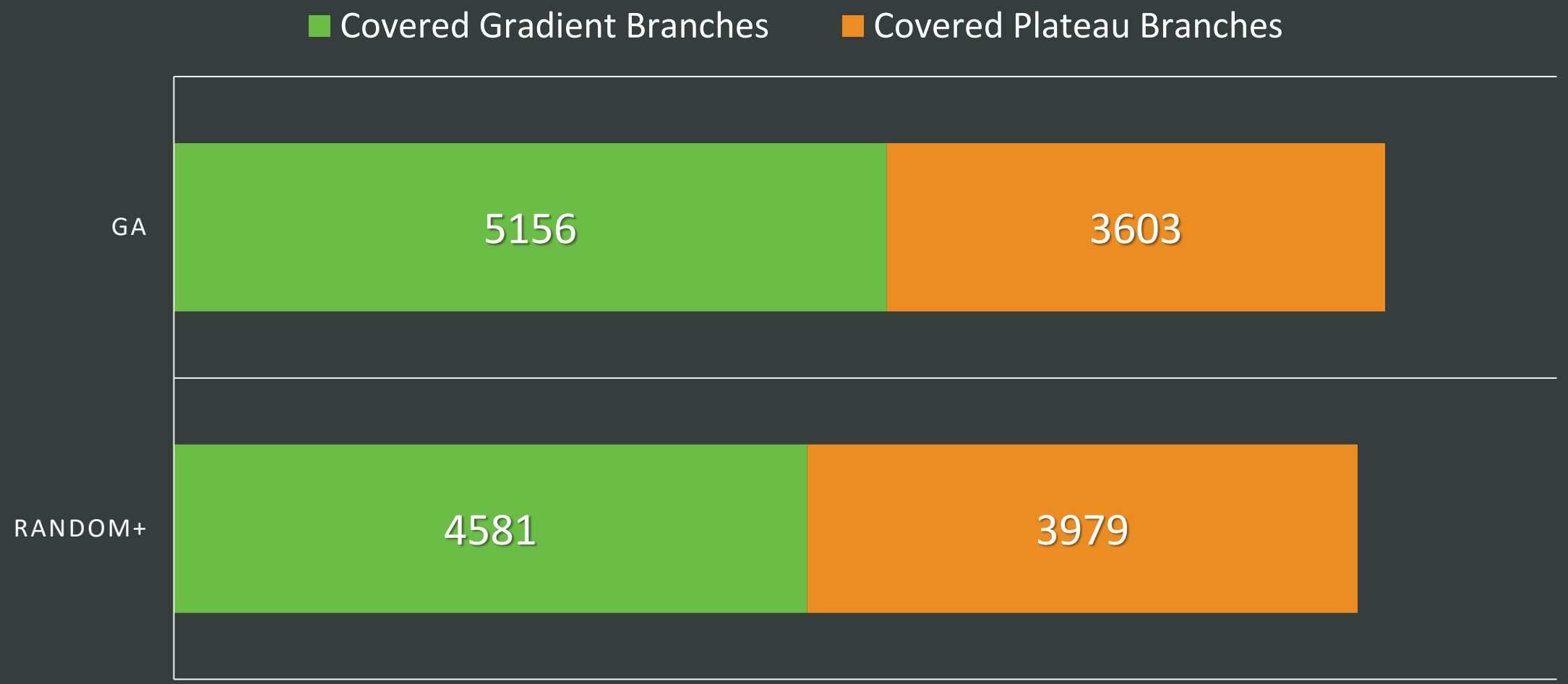
of all classes

for which **no significant difference** was observed

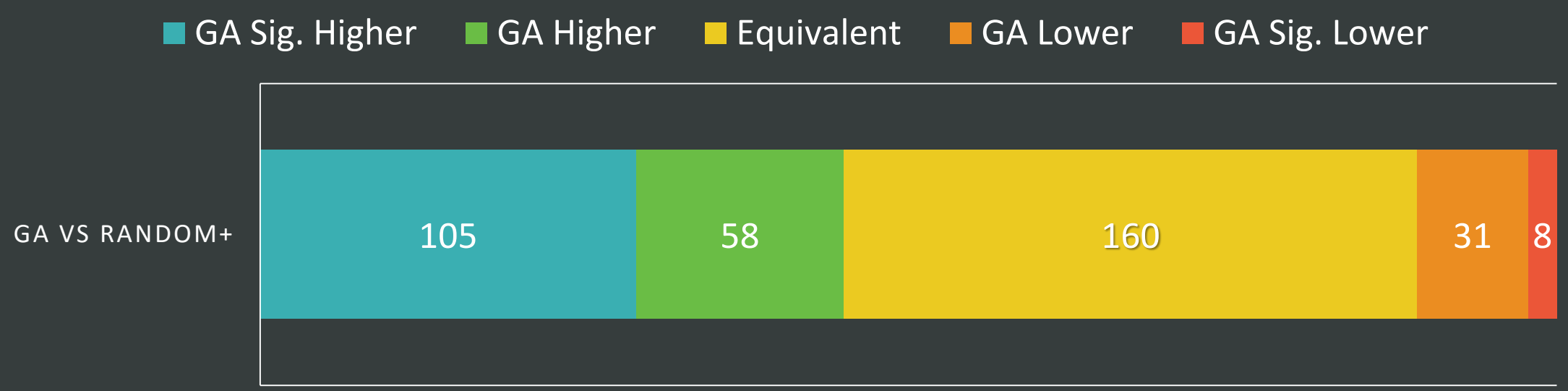
RQ2: Influence of Branch Types



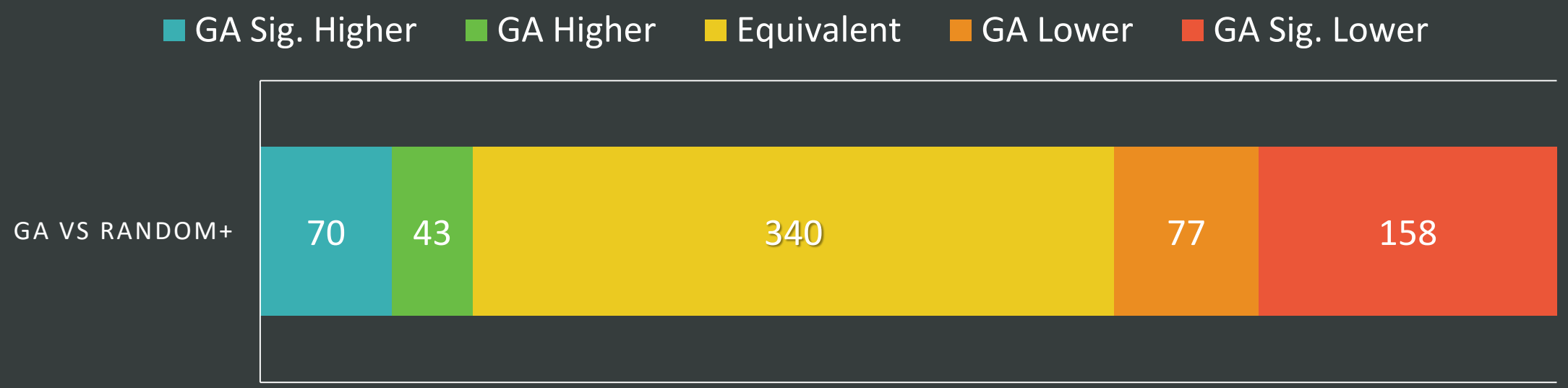
RQ2: Influence of Branch Types



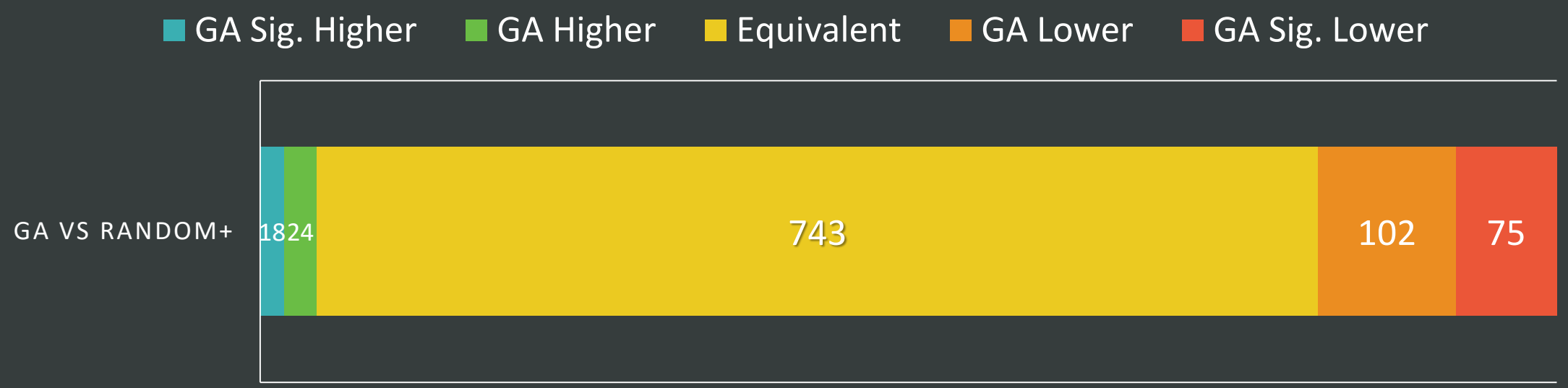
Effectiveness on Covering **Gradient Branches** (RQ2)



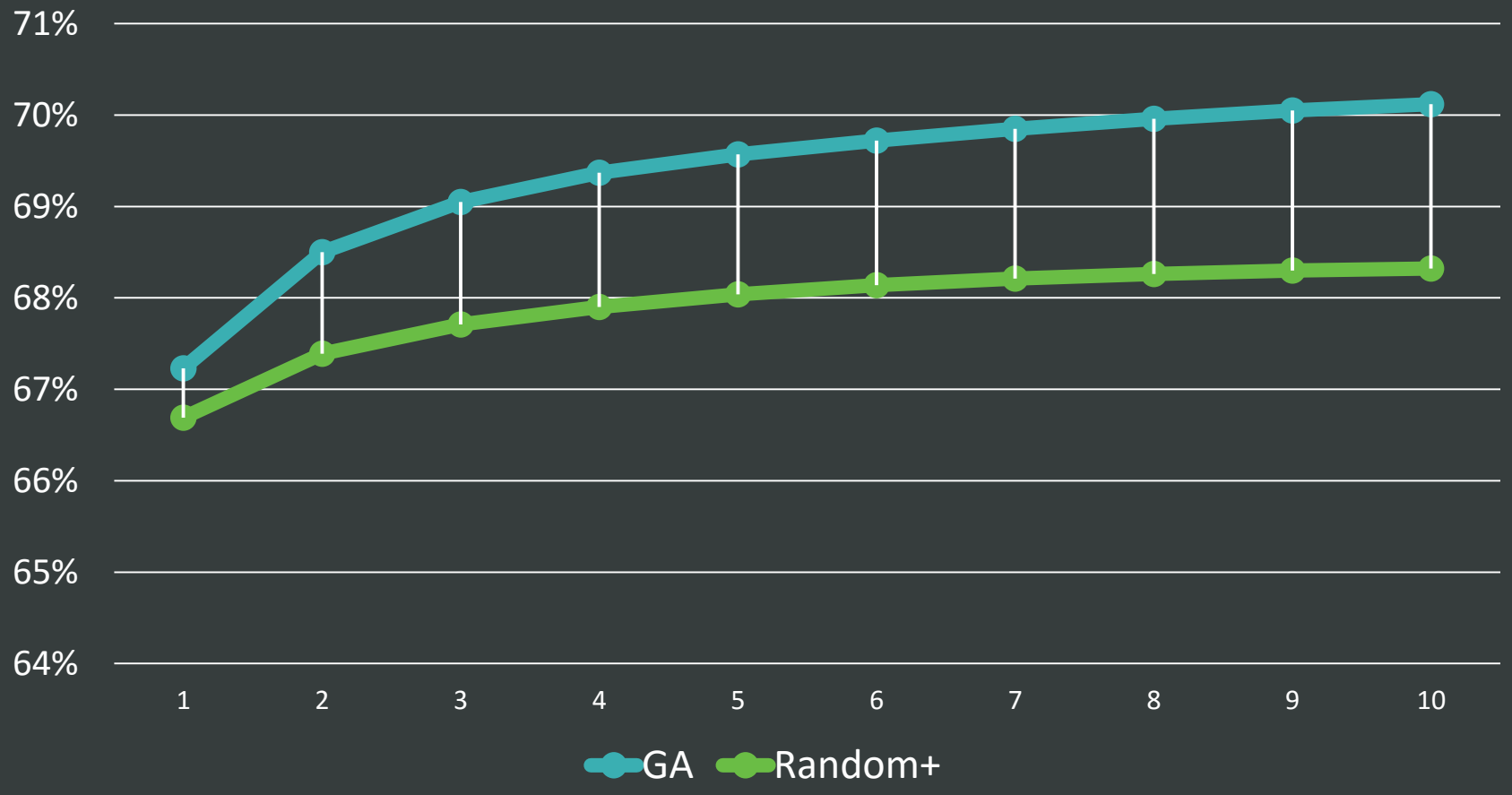
Effectiveness on Covering **Plateau Branches** (RQ2)



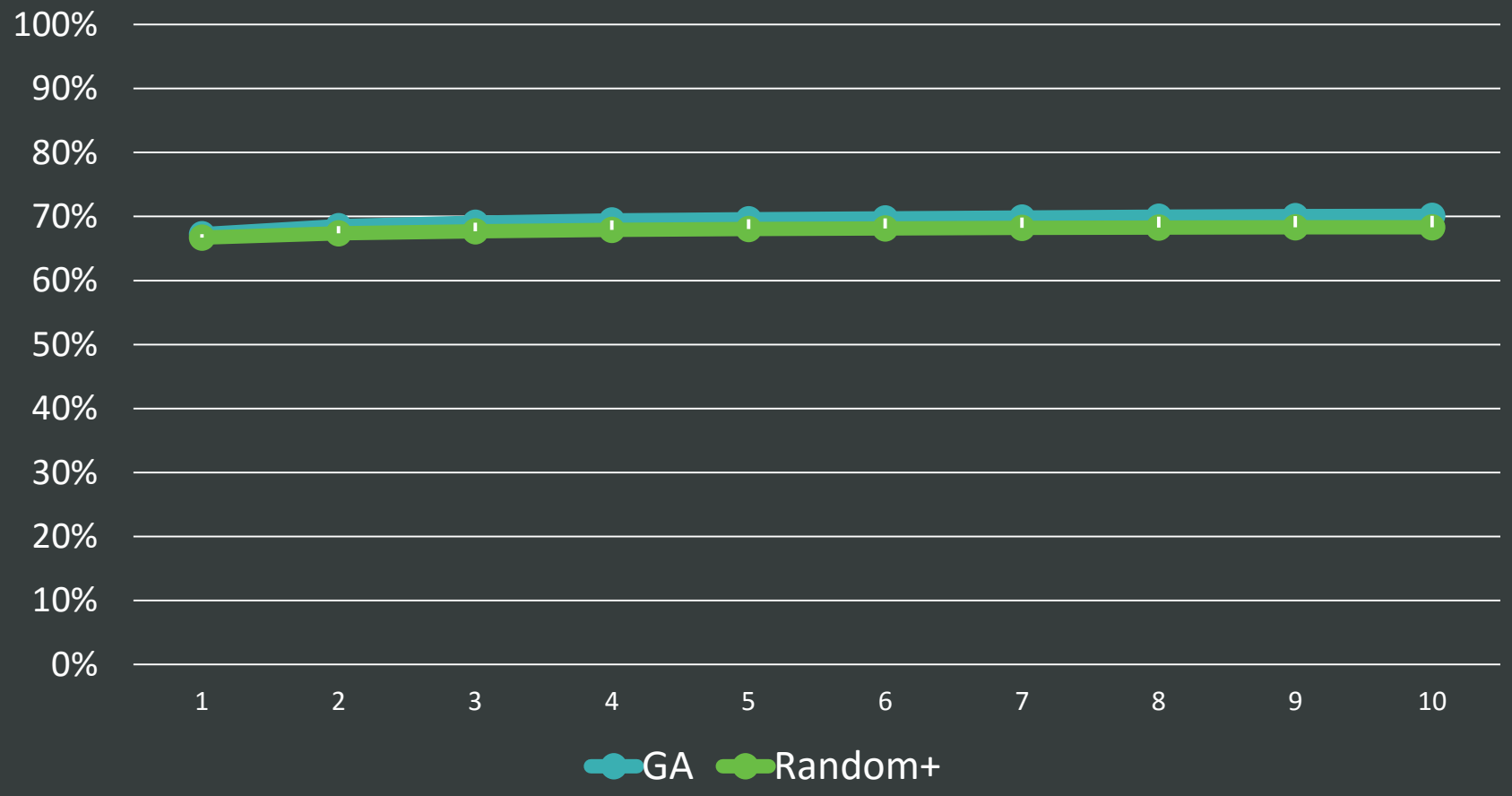
Effectiveness on Covering **Branchless Methods** (RQ2)



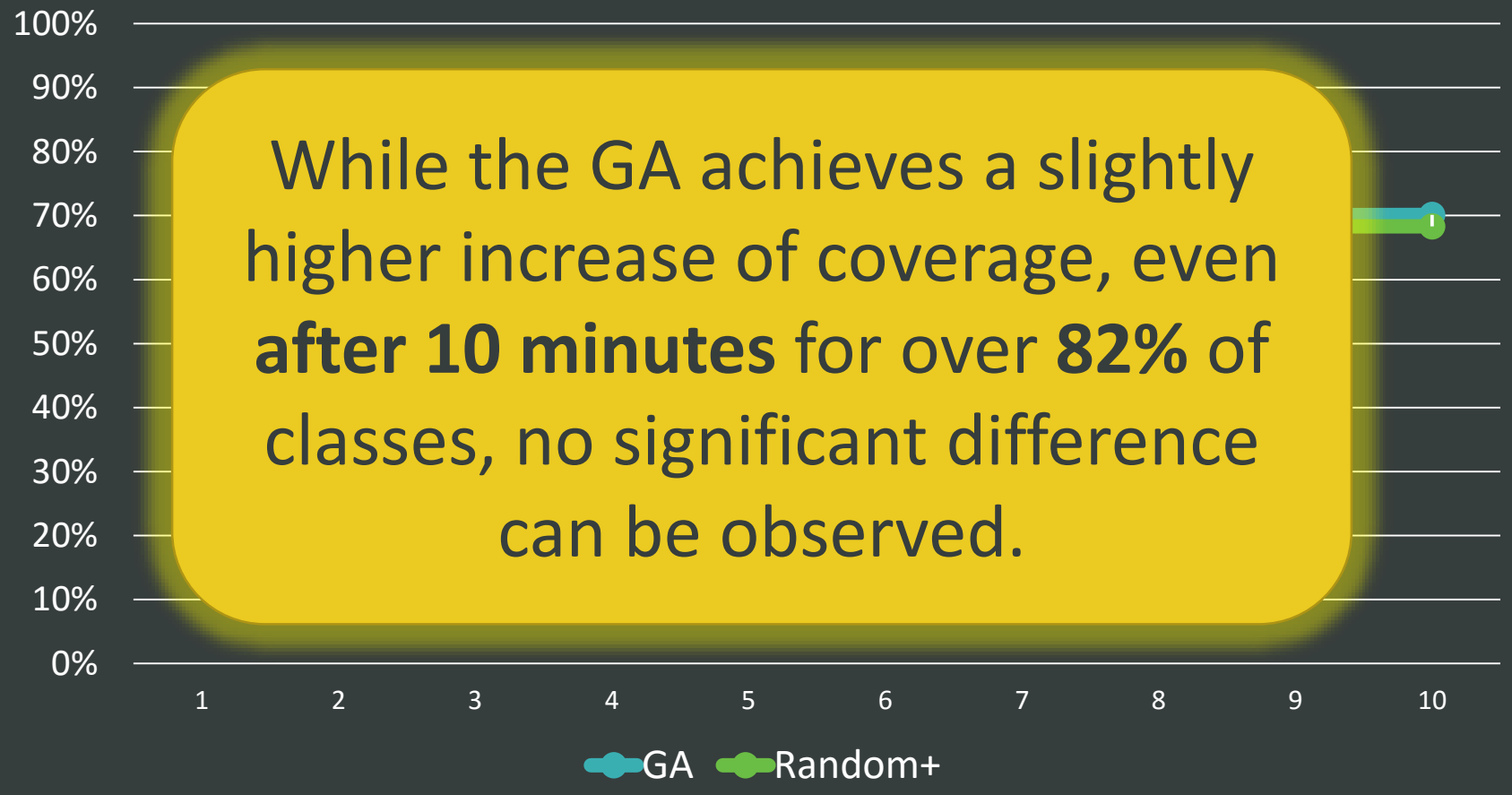
RQ3: Effect of the Time Allowed for the Search



RQ3: Effect of the Time Allowed for the Search



RQ3: Effect of the Time Allowed for the Search



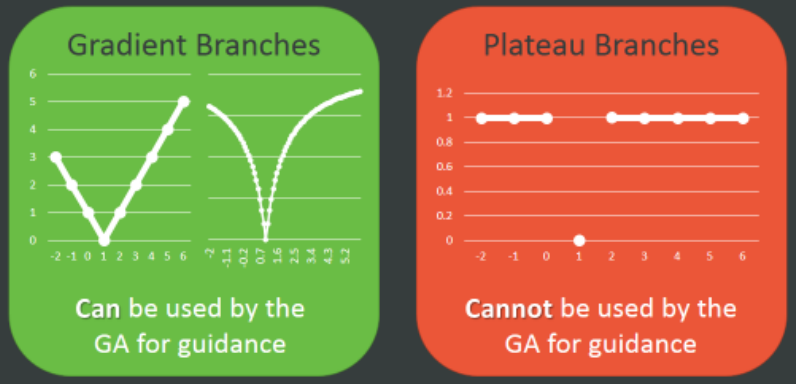
While the GA achieves a slightly higher increase of coverage, even **after 10 minutes** for over **82%** of classes, no significant difference can be observed.

How to Improve the GA?

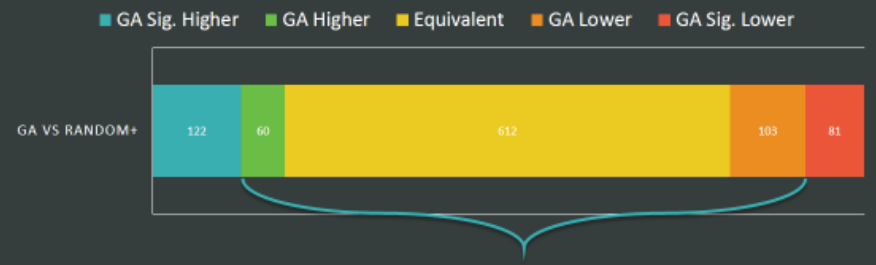
- Archive: Keep the found solutions and focus the GA on uncovered goals.
- Testability Transformation: Transform and inline flags, method calls, etc. to provide guidance to the GA
- Increase Diversity: Adaptive search operators

Search-Based Test Suite Generation

Summary of Branch Types



RQ1: Coverage Effectiveness

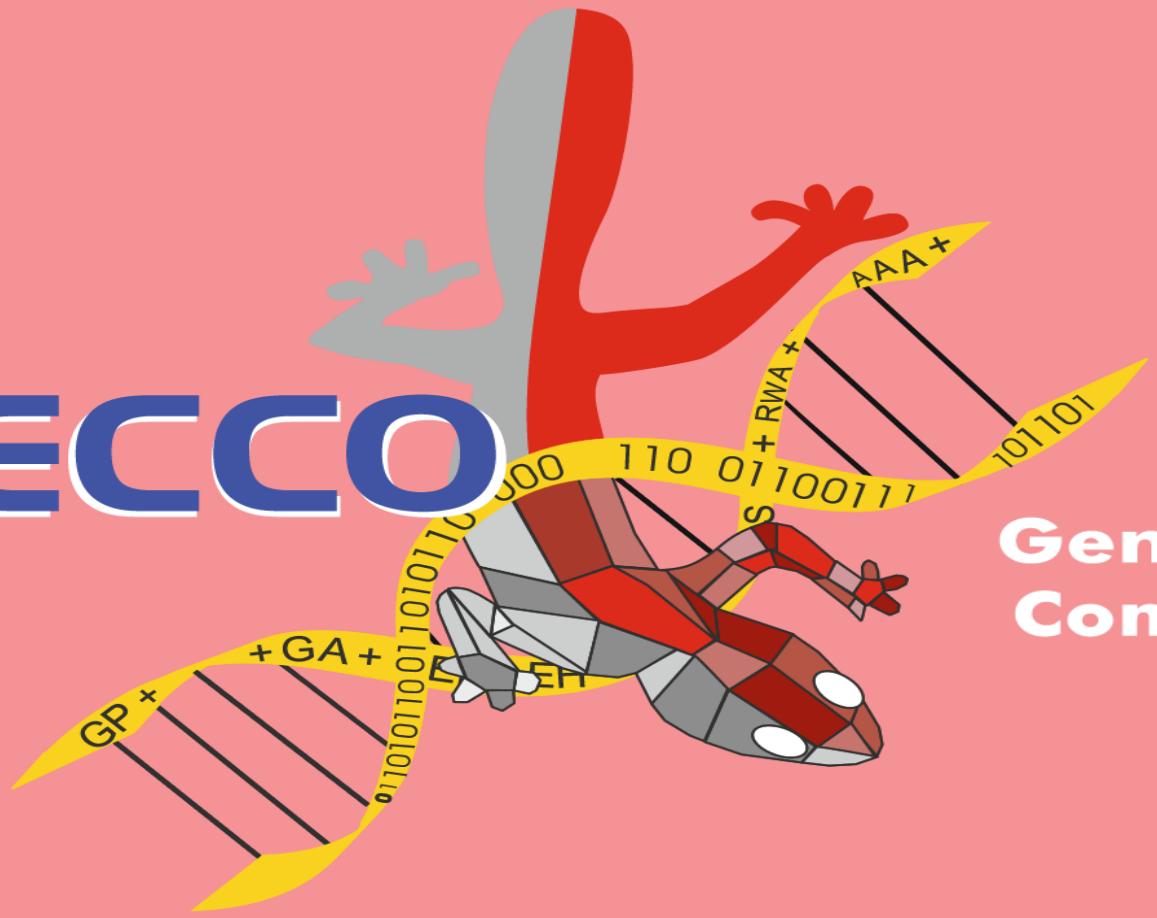


78%
of all classes
for which **no significant difference** was observed

Any Questions?

Sina Shamshiri
sina.shamshiri@sheffield.ac.uk

GECCO



Genetic and Evolutionary Computation Conference

2015

Madrid, Spain
July 11-15, 2015

acm