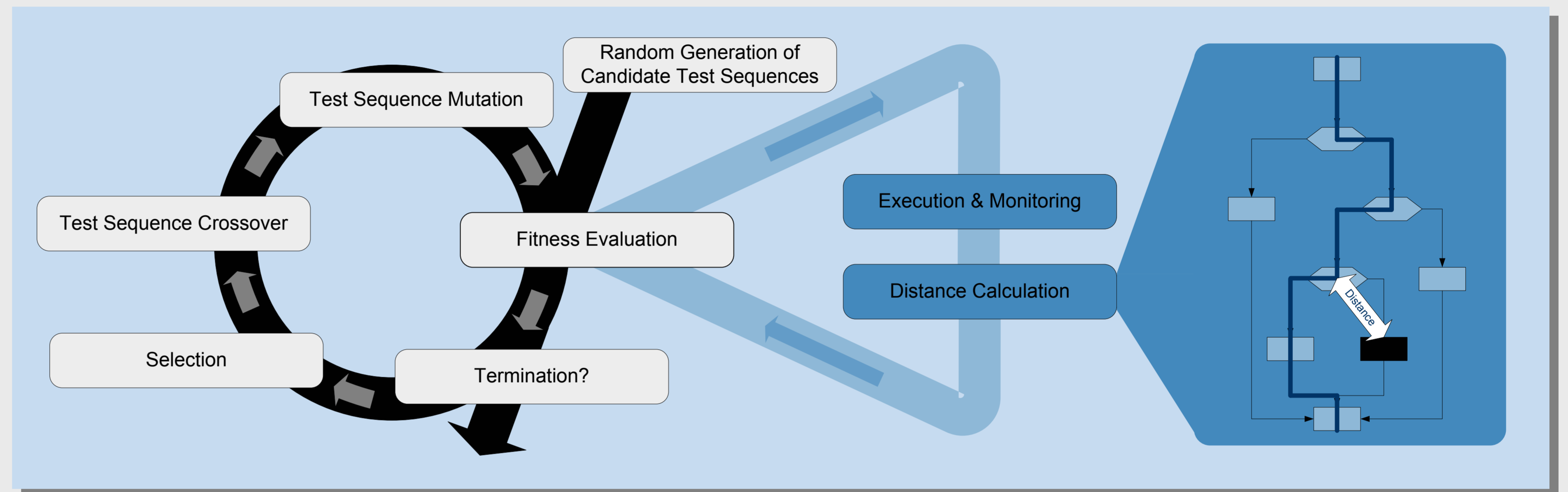
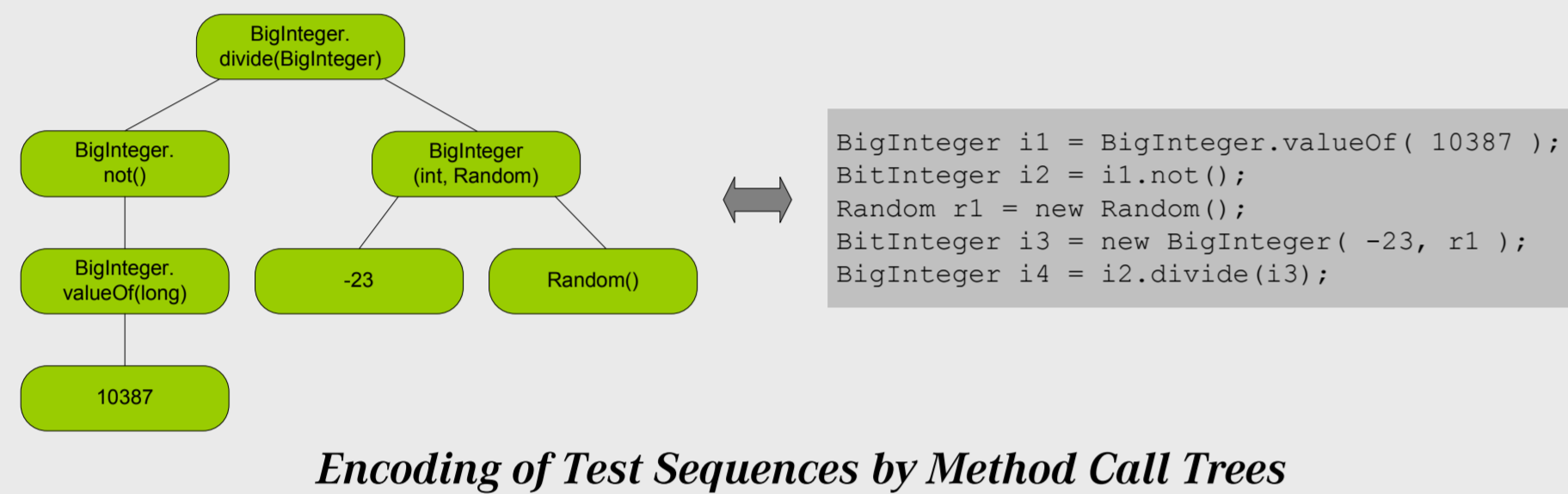


Improving Evolutionary Class Testing in the Presence of Non-Public Methods

Evolutionary Class Testing

Automatic Generation of Unit Test Suites with High Code Coverage:

- Transformation to Optimization Problems
- Application of Genetic Programming (Evolutionary Algorithms)



Testing Non-Public Methods

Increased Coverage of Non-Public Methods and hence Improved Test Thoroughness and Quality without Breaking Data Encapsulation (Using the Public Class Interface only):

- Static Call Point Analysis
- Extended Fitness Functions using Call Points

```
public class ClassA {
    private int x;

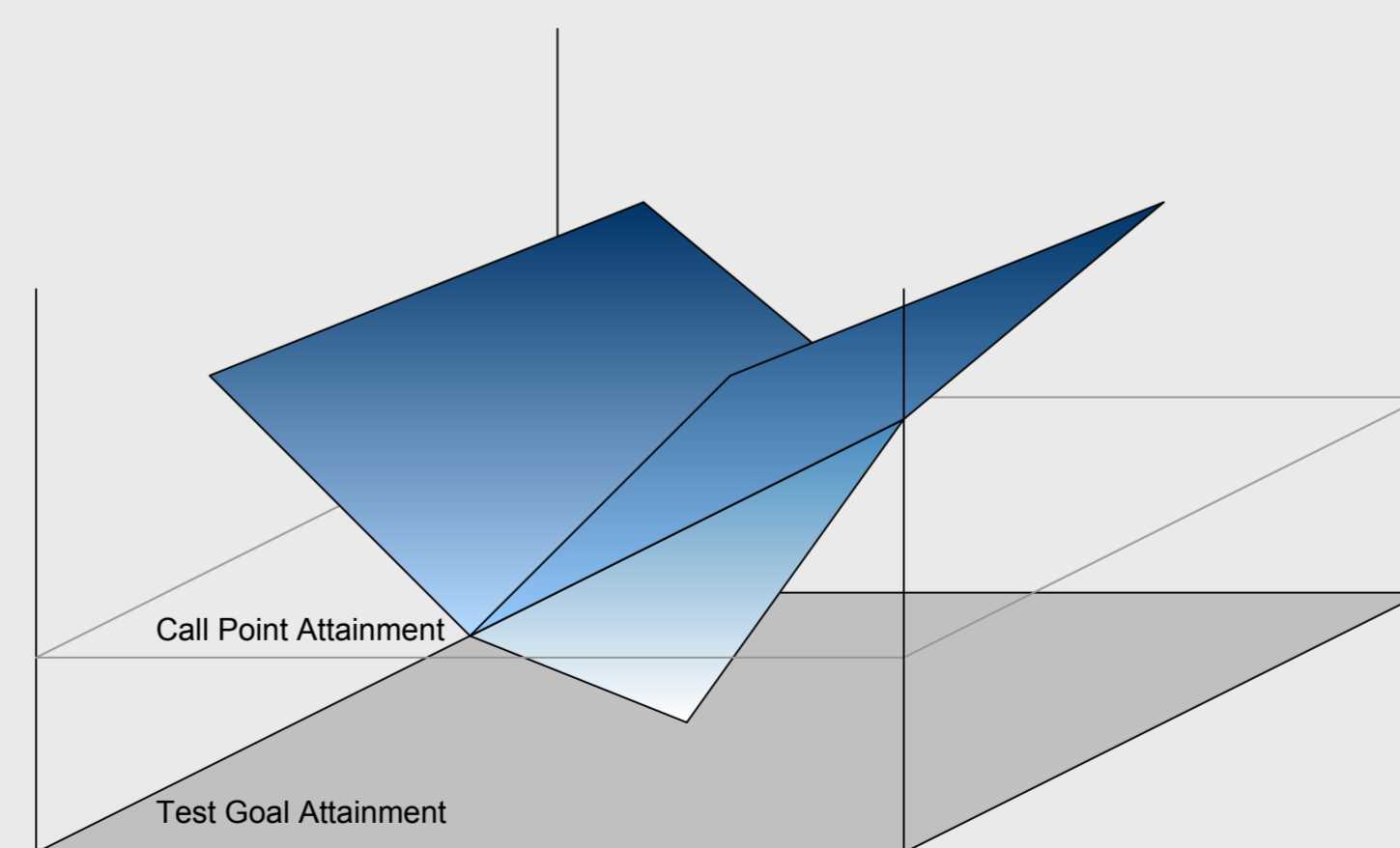
    public void m1(int a, int b) {
        if (a == 0)
            m3(b);
    }

    public void m2(int a, int b) {
        if (a == 1)
            m3(b);
    }

    private void m3(int x) {
        if (x > -10 && x < 10)
            this.x = x;
    }
}
```

Call Points

Call Point Identification



Fitness Function Landscape for Non-Public Targets

Benefit: Higher Code Coverage and Test Quality

Empirical Case Study suggests Effectiveness of Improvement

