



**Software Engineering**

# **Landscape depending parameter tuning for search-based software testing**

# Overview

1. Introduction
2. Fundamentals
3. Experimental
4. APC-DynaMOSA
5. Evaluation
6. Conclusion

# Introduction

- Unit tests
- maximize coverage (line, branch, exception)
- lack of sufficient tests
- costly and time-consuming
- => use search-based software testing

# Motivation

- Tools... => EvoSuite state-of-the-art
- may not terminate => search budget
- optimal only with optimal configuration
- No Free Lunch theorem
  - impossible to find optimal configuration for all problems
- EvoSuite's default configuration is fairly good, but not perfect

# Research goal



- wide variety of problem-cases
- concept landscape depending
- adaptive
- parameter control

# State-of-the-art



# Challenges



# Delimitation





# Search-based software testing

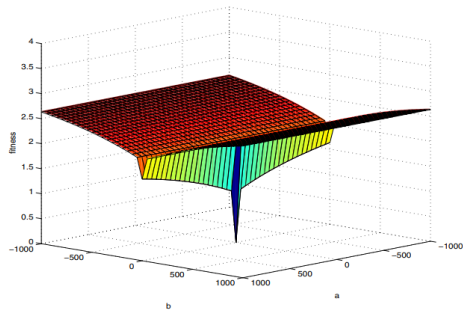
- tests for object oriented languages
- sequence of method calls
- 
- 
-

# Fitness function

- 
- function for e.g. coverage
- guidance for search algorithms
- 
- 

```
1 void bar(int x) {  
2     if (x == 1) {  
3         // uncovered code  
4     }  
5 }
```

# Fitness landscape



# Genetic algorithm

## Heading

- start with random population
- iterate till termination condition
- return last generation

## Heading

- 
- 
-

# DynaMOSA

## Heading

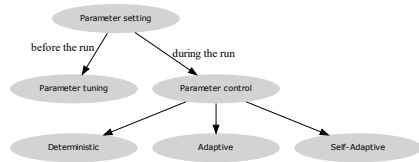
- start with random population
- multiple target
- keep track of target covering individuals

## Heading

- iterate till termination condition
- update targets
- update archive
- select by rank
- return archive as last generation

# Parameter tuning and control

- 
- 
- 
- 
- 



# Corpus

## SF110

- 110 open-source Java projects
- 23,894 Java Classes
- 
- 
- 

## Panichella et al.

- 117 open-source Java projects
- 346 Java Classes
- non-trivial and complex
- often used

# Prediction sample

- $S_1$
- 709 Java Classes
- randomly selected
- SF110

- 9.8 days on three machines

- 
- 
-



# Evaluation sample

- $S_2$
  - 346 Java Classes
  - the whole Panichella corpus
  -
- 4.8 days on tree machines
  - 
  - 
  -

# Sensitive sample

- $S_3$
- 20 Java Classes
- extracted from  $S_1$
- high Standard Deviation

- 6.6 hours on tree machines

- 

- 

-

# Comparisons

- 30 repeats for every Java Class
- Mann-Whitney U-test
- Vargha-Delaney effect size  $\hat{A}_{12}$
- 
- 
- 
-

# Concept

## Heading

- 
- 
- 

## Heading

- 
- 
-

# Landscape analysis

## Heading

- 
- 
- 

## Heading

- 
- 
-

# Targets

## Heading

- 
- 
- 

## Heading

- 
- 
-

# Classification

## Heading

- 
- 
- 

## Heading

- 
- 
-

# Parameter selection

## Heading

- 
- 
- 

## Heading

- 
- 
-





UNIVERSITÄT  
PADERBORN