

# A Partial Replication of Parameter Tuning in SBSE

## Original Paper ([Web Link](#)):

### Title:

- Parameter tuning or default values? An empirical investigation in search-based software engineering

### Authors:

1. Andrea Arcuri \*
  - a. Email: [andrea.arcuri@kristiania.no](mailto:andrea.arcuri@kristiania.no)
  - b. Github ID: [@arcuri82](#)
2. Gordon Fraser
  - a. Email: [gordon.fraser@uni-passau.de](mailto:gordon.fraser@uni-passau.de)
  - b. Github ID: [@gofraser](#)

## Replicated Paper ([Web Link](#)):

### Title:

- Revisiting Hyper-Parameter Tuning for Search-Based Test Data Generation

### Authors:

1. Shayan Zamani \*
  - a. Email: [shayan.zamani1@ucalgary.ca](mailto:shayan.zamani1@ucalgary.ca)
  - b. Github ID: [@shayanzamani](#)
2. Hadi Hemmati
  - a. Email: [hadi.hemmati@ucalgary.ca](mailto:hadi.hemmati@ucalgary.ca)
  - b. Github ID:

**Why:** There are many tools such as EvoSuite that employs search-based methods to generate unit tests for software programs. Effectiveness of search-based techniques is affected by the choice of the hyper-parameters of the meta-heuristic in use. The right choice of hyper-parameters can result in higher branch coverage and potentially finding more bugs in the software under test.

**What:** The original paper addressed the question of “if tuning these hyper-parameters is needed?”. It includes three case studies. The first one focuses on illustrating the impact of tuning. The two other case studies then investigate the effectiveness of a particular tuning method (which showed no improvement compared to the default settings). In our paper, we only focus on the impact of tuning, thus only the first case study of their paper (from now on called “the baseline paper”) will be (partially) replicated.

In particular, we focus on the first two research questions of the first case study, in the baseline paper, where the main findings are:

- Different parameter settings cause very large variance in the performance
- Default parameter settings perform relatively well, but are far from optimal

**How:** The baseline paper used a set of 20 hand-picked Java classes, which we argue that they do not represent the entire project's classes.

**Where:** We exhaustively studied the impact of tuning, with the similar hyper-parameter search space as the baseline paper (1,200 different configurations), on all 177 classes of three random projects from the very same SF100 and showed that different parameter settings do NOT have the significant impact that is claimed.

**Discussion:** In terms of challenges that we faced during this study; the time-consuming nature of this study, especially when we wanted to extend it to a larger set of classes, was the main challenge.

In addition, we were required to use the same search method (grid search) while it is better to do a random. However, it was easier to reach to the relevant literature and to design the experiments when replicating a well-written paper.