

# Programming Assignment #3: Local Search Algorithms

---

*Assigned: 15.12.2023 Due: 31.12.2023*

## 1 Objective

With the help of this assignment, you will learn

- how to formulate a given problem as a search problem,
- how to solve it by applying local search algorithms,
- and how to evaluate the performance of search algorithms.

## 2 Requirements

- You are supposed to perform PA0, PA1, and PA2 before this assignment.
- You must implement your solution in Python and using the SimpleAI package. (Other solutions will not be accepted!)
- **Use your solution to PA2 as starter code.**
- From the SimpleAI package, you will specifically need *models.py* and *local.py* codes (located under `simpleai/search` directory). You are suggested to read their documentation (<https://simpleai.readthedocs.io/en/latest/>).

## 3 Specification

In this assignment, you will add new methods to your PA2 codes, to make the problem solvable by local search algorithms. Then you will call local search algorithms: Hill-climbing search, random restarts hill-climbing search, and genetic search.

### 3.1 Implementation

#### 3.1.1 Additions to NQueens in PA2

You need to implement the following methods of SearchProblem interface, in addition to the methods that you implemented in PA1 and PA2.

```
def value(self, state):
```

Returns the value of `state` as it is needed by optimization problems. Value is a number (integer or floating point).

As the value of a state, return the number of non-attacking pairs of queens for that state.

```
def generate_random_state(self):
```

Generates a random state

You must have already implemented this method in the previous assignments.

```
def crossover(self, state1, state2):
```

Crossover method for genetic search. It should return a new state that is the 'mix' (somehow) of `state1` and `state2`.

In this method, select a random crossover point and return a new state string that is the mixture of state1 and state2.

```
def mutate(self, state):
```

Mutation method for genetic search. It should return a new state that is a slight random variation of `state`.

In this method, select a random index from the state string and randomly modify it.

### 3.1.2 Testing

Add necessary function calls to print the results of hill-climbing, hill-climbing random restarts and genetic search algorithms, in addition to traditional search algorithms.

## 3.2 Report

You will also write a report including the below sections. Don't forget to write your name, surname, and student ID in the report.

### A – Development Environment

Briefly describe your development environment (Python version, OS, IDE, CPU properties, etc.).

### B – Results

Run your implementation with different parameters and inputs (N, restarts\_limit for hill climbing random restarts algorithm, population\_size for genetic algorithm, etc.). Put your output results in this section.

### C – Discussion

Discuss the results of different search algorithms in terms of *completeness*, *optimality*, and *time and space complexity* by referring to your results in Part B and our lectures. Compare local search algorithms to traditional search algorithms. Don't give purely theoretical information in this section, write your own observations and comments.

## 4 Submission

- The assignment can be done individually or in teams of a maximum of 3. (Teams can be a mixture of normal and evening education students.)
- You are free to use any Python development environment that supports Interactive Python Notebook (.ipynb). Some alternatives: Jupyter Notebook, JupyterLab, VS Code, ...
- You will submit your report in pdf format and source files (only the code you implemented, not the simpleai library codes).
- Place all your files in a zip archive with the name **AI\_PA3\_Surname1\_Surname2\_Surname3.zip** and submit using the Teams assignment module. Single submission for each team is sufficient.

- If you have further questions, you can send me an e-mail or send a message via MS Teams.

#### 4.1 Late Submission Policy

The deadline for homework submissions is **23:59** at the specified date. For each additional day, a **25% cut-off** will be applied.

### 5 Academic Honesty Policy

You cannot borrow others' ideas or portions of codes, without giving a proper citation. This can be an Internet source, AI chatbot, or your friend. Clearly indicate which part of your code/report/idea is borrowed and state its source. Of course, you cannot get all or most of your work from others. Otherwise, you will be penalized.

Dr. Zeynep ÇİPİLOĞLU YILDIZ