

Assignment. Evolutionary optimization

Release: 8th September 2023
Supervision: 21st September 2023
Due: 5th October 2023

Vladimir Tarasov

1 Introduction

The assignment work is carried out in groups of 2–3 students. The assignment comprises **2.5 credits**. The work includes coding in a programming language and writing a report. The assignment ends with a short viva (oral examination).

The assignment falls within the area of evolutionary optimization. For the assignment work, the students are free to use any imperative or object-oriented programming language. Prolog or functional programming languages (e.g., LISP or Haskell) may not be used for these tasks.

2 Work Process

The assignment work is carried out in groups of 2–3 students, and you perform group division yourselves in Canvas. The assignment comprises **2.5 credits** and has the grading scale 5, 4, 3 or Fail.

There are three examination occasions, where the first is in October 2023, the second in December 2023 and the last one is in August 2024. The students who have not obtained a grade for the assignment on any of these occasions will have to carry out the assignment work next time the course is given.

The assignment work includes:

- A supervision occasion which is given group-wise on Thursday, 21st September 2023, 08:15–12:00.
- A short viva (oral examination) on Thursday, 5th October 2023, 08:15–12:00.

The supervision and viva sessions are scheduled in the room E1217 and E4304 respectively. Each group may book one time slot per occasion, and booking takes place via the Canvas calendar.

3 Tasks

This assignment is intentionally very open-ended, and the marking will depend on the quality of the submitted solution and report.

You shall implement and evaluate an evolutionary solution for an optimization problem of your own choice. Two possible candidate problems are n -queens and TSP (Traveling Salesperson Problem).

An important part of the task is to decide on the exact experimentation, but one thing that should be varied is *the size of the problem*; for n -queens the number of queens and the corresponding size of the board, for TSP the number of cities. You must choose a problem where this is possible.

In addition to the problem size, you may try different parameter values or vary some other aspect of your system. For the actual evaluation, you should look at the quality of the solution, but also at the running time and maybe some other metrics. Here, you should test your system “to its limits”, i.e., not just try a bunch of small problems.

The choice of programming language is free, with the exception that Prolog or functional languages like Haskell, LISP etc. may not be used.

There are of course many existing solutions to similar tasks online. While you may get some inspiration from those, you must write your own code from scratch, i.e., you may **not** copy-paste any code or exchange code between groups.

4 Report with Results Summary

You need to write a report where you describe:

- the problem,
- the system design (i.e., representation, fitness function, operations, parameter values etc.),
- an experimental setup,
- results from the experimentation and
- some analysis with conclusions.

The report should be well-organized into different sections describing the items listed above. Use graphs where you deem them appropriate to illustrate your results. Think carefully about the best design of the graphs.

5 Deliverables

The deliverable for this assignment are:

- A **pdf** document with your written report where you describe the problem, the system design (i.e., representation, fitness function, operations, parameter values etc.), an experimental setup, results from the experimentation and some analysis with conclusions.
- The numerical results from the experimentation saved in a text file.
- A **zip** file containing the source code for implementation of a solution of an optimization problem as well as the executable files.

The source code has to be thoroughly tested and the executable should be possible to run. Include any information about the setup needed to run your implementation. Moreover, the source code has to include reasonable amount of comments, that is the code is supposed to be well-readable. Do not forget to add **names of all the group members in a comment** at the beginning of the file with the main program.

6 Grading Criteria

The maximum score for the assignment is 30 points, and the grading is:

3	12–17.5	points
4	18–23.5	points
5	24–30	points

The points division is as follows :

Implementation	8	points
Experimentation	7	points
Report	10	points
Viva	5	points