

COMP3201 – Intro to Nature-Inspired Computing (Fall 2018)

Final Project - EA-TSP Competition

Objectives:

The goal of this project is to exercise on designing and implementing an evolutionary algorithm to solve the Traveling Salesman Problem (TSP) with different instances following the workflow:

1. Design a (standard/conventional) EA for TSP with operations discussed from the course
2. Implement at least one advanced technique from published research articles at your own choice (e.g., parameter self-adaptation, improved mutation/crossover, diversity preservation, etc.) to improve your EA
3. Test your EA on the TSP instance of Western Sahara (29 cities)
4. Test your EA on the Uruguay instance (734 cities) and optimize the runtime
5. For extra challenges (bonus points!), try the Canada instance (4663 cities)

Milestones:

1. Project proposal (5% into final grade, due on November 9th): a one-pager to describe

- a) the design of your EA
- b) what techniques you would like to use to advance your EA
- c) plan of runtime optimization of your EA
- d) team management (task distribution among members (3 to 4) in your

team)

2. Project demo (5%, November 27th and 29th, in class)

Six minute for each group, followed by a one-minute Q&A

3. Final project report and code (30%, due on December 10th)

The maximum 10-page final project report (excluding references) should follow the structure of Introduction - Methods - Results - Discussion - References. The use of figures and tables is highly encouraged. Source code must be written in Python and submitted with a readme.txt. Your EAs should be implemented from scratch.

Marking and notes:

The final projects are marked based on the relative effectiveness and efficiency of the EAs comparing to other teams (It is a competition!). Other factors that will be considered include the writing, organization, and figures and tables of the reports, as well as the ease of use and readability of the code.

The instance files are named with the format TSP_[country]_[#cities]. In the instance files, each row provides the two-dimensional coordinates of a city. The fitness should be computed as the complete tour length (starting and ending at the same city). The distance of two cities is the Euclidean distance of their coordinates.