UEMH3073 / UECS2053/ UECS2153

Artificial Intelligence, June 2025 Trimester

Lab 2: Genetic Algorithm

Requirements

Python + Jupyter Notebook

Methodology

1. Download the 'Lab 2_Genetic Algorithm.ipynb' Jupyter notebook and open it up.

2. Implement a genetic algorithm for solving the Travelling Salesman Problem (TSP). You will encounter #TODO in the code cells explaining tasks you need to complete or codes you need to write so that the genetic algorithm functions well and runs correctly. Look for "Replacement starts here" and "Replacement ends here" to know the parts of the codes requiring your revision and inputs. The #TODO list is as follows:

Label	Class/Functions	Tasks
#TODO1	Population Initialization	Read a set of cities from the filename when creating an initial population
#TODO2	Parent Selection	Replace a dummy parent selection function with Tournament Selection.
#TODO3	Parent Selection	Replace a dummy parent selection function with Proportional Selection.
#TODO4	Survival Selection	Replace the dummy survival selection function with Merge, Sort and Truncate.
#TODO5	Crossover	Replace the dummy crossover function with Partially Mapped Crossover approach.
#TODO6	Mutation	Replace the dummy mutation function with Insertion Mutation approach.
#TODO7	Performance evaluation	Present performance evaluation for different Parent Selection functions created in this lab. Examples of performance include route distance, convergence rate, and others.

3. Apart from the ipynb file, you will need text files providing city coordinates (i.e., cities40 and cities400).

4. Test your genetic algorithm on small data sets provided (i.e., cities40).

5. Once you have chosen the best combination of functions, try to obtain the best solution to the large data set provided (i.e., cities400).

6. Remember to gather and save your results for performance evaluation.

Note: This practical is an open-ended lab. After the demonstration and briefing by the instructor, students are expected to plan and conduct proper procedures to achieve the aim and objectives of the experiments.

Report

- You should complete the report in a group of four persons. Only in cases you cannot form a four-person group, you can work in a group of three persons. Group members must be from the same practical group.
- Your report should present:
  - Your new codes and your explanation of those codes.
  - Your results and your analysis of those results.
- Your report should *not* present:
  - Existing codes
- Pay special attention to the relative performance and effectiveness of the functions you implemented.
- Page limit is 8 pages.
- Be sure to report the best distance in the report.

Submission

- Your submission comprises of ONE (1) Jupyter notebook and ONE (1) PDF document (Word files not accepted).
- Name the documents Lab2_P<X>_<Group Number>.<file extension> where ‘X’ is your practical number (P1, P2, P3). The first page is the title page with the following details:

  a. Your practical group number (e.g., Practical group 2)

  b. Names of group members

  c. Students IDs of group members

  d. Year/Trimester of group members

  e. Programmes of group members

- Submission is through the WBLE website.

- Submission deadline is as follows.

Lab session	Deadline
31st July 2025 (Thursday, AM/PM Sessions)	7th August 2025 (Thursday) 11.55PM
7th August 2025 (Thursday, AM/PM Sessions)	14th August 2025 (Thursday) 11.55PM
14th August 2025 (Thursday, AM/PM Sessions)	21st August 2025 (Thursday) 11.55PM

Lab 2 – Genetic Algorithm Marking Rubric

Label	Excellent (7 to 10 marks)	Above average (5 to <7 marks)	Below average (>0 to <5 marks)	0 marks
#TODO1 #TODO2 #TODO3 #TODO4 #TODO5 #TODO6	Correct implementation.	Minor errors.	Major errors.	No answer is given.
#TODO7	In-depth performance evaluation. Optimal routes are found.	Clear understanding of performance evaluation.	Inaccurate or unclear understanding of performance evaluation.	No answer is given.
Label	Excellent (12 to 15 marks)	Above average (8 to <12 marks)	Below average (>0 to <8 marks)	0 marks
Report Presentation and Formatting	Good readability, appropriate use of graphics/tables. Minimal grammatical and formatting errors.	Acceptable, with some errors in formatting, grammar etc.	Difficult to read, with obvious errors in formatting, grammar etc.	Unreadable report.
Code Quality and Comments	Code is easy to read because it is very well organised, showing proper planning.	Code is well organised and commented.	Working code.	Very poor code (no cells, hard to read etc.) or provided code does not work.