UECS2053 UECS2153 UEMH3073 UEMH3163 Artificial Intelligence

May 2022 Trimester

Lab 2: Genetic Algorithm

Requirements

Python + Jupyter Notebook

Methodology

1. Download the 'Lab2GeneticAlgorithm.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	Marks
#TODO1	Fitness	Write a new Fitness function 2, which is based on (Peng Chen, 2013).	10
#TODO2	Population initialization	Read a set of cities from the filename when creating an initial population.	10
#TODO3	Parent selection	Replace a dummy parent selection function with Tournament Selection.	10
#TODO4		Replace a dummy parent selection function with Proportional Selection.	10

#TODO5	Survival selection	Replace the dummy survival selection function with either Fitness Based Selection or Merge, Sort & Truncate.	10
#TODO6	Crossover	Replace the dummy crossover function with appropriate crossover approach generating valid paths. The Partially Mapped Crossover and Order Crossover approaches are examples of appropriate crossover approaches as explained in (Peng Chen, 2013).	10
#TODO7	Mutation	Replace the dummy mutation function with an appropriate mutation approach generating valid paths. Shift Mutation is a suitable mutation approach as explained in (Peng Chen, 2013).	10
#TODO8	Performance evaluation	Present performance evaluation for the different options created in this lab, either: a) Fitness function; or b) Parent Selection function. Examples of performance include route distance, convergence rate, and others.	10
			80*

* 20 marks are allocated for Report Presentation and Formatting (10%) and Code Quality and Comments (10%).

3. Apart from the ipynb file, you will need:
   a. Text files providing city coordinates (i.e., cities8 and cities500).
   b. Reference:

      Peng Chen, "An Improved Genetic Algorithm for Solving the Traveling Salesman Problem", *In Proceedings of 9th International Conference on Natural Computation*, Shenyang, China, 2013.
4. Test your genetic algorithm on small data sets provided (i.e., cities8).
5. Once you have chosen the best combination of functions, try to obtain the best solution to the large data set provided (i.e., cities500).
6. Remember to gather and save your results for performance evaluation.

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:
  1. ONE (1) Jupyter notebook with file name being <Your name>_jupyter.ipynb
  2. ONE (1) Report in PDF format with file name being <Your name>_report.pdf (Word files are not accepted). 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 exactly 7 days after your lab finishes. For Week 6 labs, submission deadline is more than 7 days.

Lab session	Deadline (Date)
19 July (Tuesday) 2:00pm-5:00pm	27 July (Wednesday) 5:00pm
22 July (Friday) 2:30pm-5:30pm	29 July (Friday) 5:30pm
26 July (Tuesday) 2:00pm-5:00pm	2 August (Tuesday) 5:00pm
29 July (Friday) 2:30pm-5:30pm	5 August (Friday) 5:30pm
2 August (Tuesday) 2pm-5pm	9 August (Tuesday) 5pm
5 August (Friday) 2:30pm-5:30pm	12 August (Friday) 5:30pm
9 August (Tuesday) 2pm-5pm	16 August (Tuesday) 5pm
12 August (Friday) 2:30pm-5:30pm	19 August (Friday) 5:30pm

Lab 2 – Genetic Algorithms Marking Rubric

Label	Excellent (7 to 10 marks)	Above average (5 to <7 marks)	Below average (>0 to <5 marks)	0 marks
#TODO1	Correct implementation.	Minor errors (i.e., slight effects on the fitness value).	Major errors (i.e., significant effects on the fitness value).	No answer is given.
#TODO2	Correct implementation.	Minor errors.	Major errors.	No answer is given.
#TODO3				
#TODO4				
#TODO5				
#TODO6				
#TODO7				
#TODO8	In-depth performance evaluation. Optimal routes are found.	Clear understanding of performance evaluation.	Inaccurate or unclear understanding of performance evaluation.	No answer is given.
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.