SMU | LYLE SCHOOL OF ENGINEERING

CS 3353 ALGORITHMS: LAB 4

Overview: Using both Simulated Annealing and Genetic Algorithm (GA) techniques, solve the traveling salesman problem (TSP) (find a Hamiltonian circuit) for a given list of nodes and positions (graph). Compare and contrast your results with those from Lab 3.

Code Repositor Link: https://classroom.github.com/a/F6d7o5X0

Due: December 4, 2020 @ 11:59pm.

Code Requirements:

- Always use Node 1 as your start/finish point of TSP
- Read node list in from a text file which uses same format as the positions.txt file from Lab 2
  - NodeID x, y, z → 1,1.23,3.45,4.112 → int,float,float,float
  - You can derive your own list of nodes and positions
  - Code will be tested against a private list of nodes/positions. Code will only be tested up to the number of nodes identified in writeup as total nodes able to analyze.
- Return optimal (shortest) path for given list of nodes
- Start with a list of 4 nodes and increase to the maximum number that is feasible with your computer for both Tabu and GA algorithms
- When coding GA, you must identify and test multiple techniques for:
  - Selection
  - Mutation
  - Crossover
- When coding Simulated Annealing, you must identify and test multiple values/techniques for:
  - Temperature value
  - Temperature scaling over search
- Design your code to be reusable for new algorithms, incorporating the file loader and output system into a single interface. Utilize patterns (or modified patterns) to make informed decisions on how to architect your project. How your code was designed will hold a substantial weight on overall project grade. Specifically, in this lab you will need to consider how you design an interface for GA and Simulated Annealing to test the multiple techniques and configurations required.

Report Requirements:

- Plot timing with excel for each algorithm (and its different configurations) in a single graph. X axis should be number of nodes in graph, Y axis should be total timing.
  - Also provide a table to summarize data in graph
  - This should include results from Lab 3 for comparison
- Provide a summary of results obtained and be sure to explain time complexity for each algorithm
  - Heuristic techniques should be compared to timing (actual and asymptotic) of naïve and dynamic programming approaches in Lab 3.
  - Timing comparison should be for all graph sizes achieved in Lab 3.
- For graph sizes larger than those obtained in Lab 3 produce the following learning curve plots for the top performing configurations for each heuristic technique.
  - Plot the learning curves for multiple graph sizes on the same plot.
  - The x axis should be epoch/step, y axis should be best length of path found for given epoch/step.
  - Execution time for each individual algorithm/graph should be at least 10min

- Explain the learning curve results
- Explain and document your design decision. Should include write up that explains architecture decisions and how you designed for functionality and extensibility as well as a UML diagram.
- Explain and discuss in detail how the variations of GA and Simulated Annealing caused changes in the outcomes, and be sure to explain why those modification/changes caused the output obtained.

All code used in project should be in the "Code" folder, raw data files generated should be in "Data" and your final report should be in "Report" folder