ECM3412 - Coursework exercise

Deadline: 13th December 2023 12:00 (midday, Exeter time)

Lecturers: Dr. Ayah Helal (a.helal@exeter.ac.uk)
Dr. David Walker (d.j.walker2@exeter.ac.uk)

This assessment (CA) is worth 40% of the overall module assessment. This is an individual exercise and your attention is drawn to the faculty and University guidelines on collaboration and plagiarism, which are available from the faculty website.

Referencing, and academic conduct You are required to cite the work of others used in your solution and include a list of references, and must avoid plagiarism, collusion and any academic misconduct behaviours.
https://ele.exeter.ac.uk/course/view.php?id=1957

Coursework submission The submission of this coursework is via **ElE2**. The implementation can be in the programming language of your choice, such as Python, Java, C#, etc., and it must run without errors. Please ensure that your code is clear and well-commented, and that your answers are clearly labelled. The implemented code should have a README file to support the execution of the code.

In **ElE2**, you should submit a **single compressed (zipped) folder** which contains all the relevant files for this coursework exercise. The compressed folder can contain a PDF file with your report and your code. The report should have a **maximum of 4 pages (4 sides of A4, references do not count towards the limit)** which should include a description of your experiments where tables and/or graphs of results should take up no more than 2-3 pages, and your answers to the questions/descriptions in the remaining space. **Your code should be clearly commented and submitted as one part of the zip file.**

Marking scheme The marking scheme for this assessment is as follows.

Correct and efficient implementation of the algorithm	15%
Quality of the experiments design	15%
Documentation of code, using readme file and comments	5%
Correct results from the ACO runs	20%
Quality (e.g. readability & usefulness) of tables and graphs	20%
Answers to Questions	20%
Conclusions	5%
Overlength submissions (per page)	-10%

Question 1 | Ant Colony Optimization problem.

The Task is to optimize travel routes for a company operating in multiple cities, utilizing ant colony optimization (ACO) to address the Traveling Salesman Problem. You're provided with data in XML format specifying city details and travel costs between them for two sets of cities: one in Brazil and the other in Burma.

Your goal is to minimize the total cost, which is calculated using a distance matrix, denoted as D. This matrix represents the distances between between every pair of cities.. The total cost is defined by the equation:

$$cost = \sum_{i=1}^{n-1} D[C[i]][C[i+1]] + D[C[n]][C[1]]$$

Here, you're looking for an efficient route represented by a vector C. Each entry $C[i]$ is a unique integer value from 1 to n, denoting the location identifiers. The objective is to minimize this cost for optimal travel routes in both the Brazilian and Burmese cities.

To do this, you'll be implementing the ACO algorithm. The algorithm involves various parameters such as the number of ants, pheromone evaporation rate, pheromone importance factor, heuristic information, and more.

The Ant Colony Optimisation Algorithm

You'll need to build a construction graph to hold the pheromone values for each decision made by the ants. This graph will reflect the paths and associated pheromone levels for every possible city-to-city movement. The ACO algorithm can then be structured as follows:

- **Random Pheromone Initialization:** Initialize the construction graph by randomly assigning small amounts of pheromone (value between 0 and 1) on the edges representing the connections between cities.
- **Ant Path Generation:** Generate a set of
  - m ant paths from the starting node.
  - Each ant will traverse the construction graph, making decisions at each city based on the amount of pheromone on potential paths and a heuristic function.
  - The selection will be biased by the amount of pheromone on the paths ahead and the heuristic information.
  - Ants move to the next city until all cities are visited.
- **Pheromone Update:**
  - After an ant completes its path, update the pheromone in the construction graph based on its fitness.
  - Update the pheromone according to the formula $\frac{Q}{fittnes}$ to reward paths that decrease the fitness function. Here, Q is a parameter.
- **Pheromone Evaporation:** Evaporate the pheromone for all links in the graph by multiplying all pheromone values by the evaporation rate e.
- **Termination Criterion:** Terminate the algorithm when a maximum number of fitness evaluations (10,000) is reached. The result will be the best fitness found so far by the ants.

Implementation and Experimentation

Regarding experimentation, the goal is to explore how different parameters and operators affect the ACO's performance on the Traveling Salesman Problem. The experiment design involves varying one parameter at a time while keeping others constant. Potential parameters/operators to explore include:

- ACO approach variations (e.g., MMAS, Elitism)..
- Evaporation rate variations (e.g., 0.8, 0.7, 0.3).
- Colony size variations (e.g., 10, 50, 100).
- Local heuristic function variations (e.g., $\frac{1}{d}$, $\frac{Q}{d}$). Local search methods ((e.g., Tabu Search, hill climbing, etc).

Each experiment involves running the ACO algorithm with a specific parameter or operator variation and observing its impact on the algorithm's performance in finding optimal solutions for the Traveling Salesman Problem in the given city datasets.

Analysis

Hint: You should think carefully about how best to present your results to show the behaviours of the algorithm during your trials.

Question1: Which combination of parameters produces the best results?
Question 2: What do you think is the reason for your findings in Question 1?
Question 3: How does each of the parameter settings influence the performance of the algorithm?
Question 4: Can you think of a local heuristic function to add?
Question 5: Can you think if any variation for this algorithm to improve your results? Explain your answer.
Question 6: Do you think of any any other nature inspired algorithms that might have provided better results? Explain your answer.

In your answers, describe your observations of the results and explanations or conclusions you can draw from your results. In addition, you should describe any further experiments you conducted to better answer questions 1-5 above and to demonstrate your understanding of the parameters used in the ACO.