Team Project: Swarm Intelligence Algorithms to Solve the Traveling Salesperson Problem

Let me quote form an essay on Swarm intelligence [1]: Swarm intelligence can be described as the collective behavior emerged from social insects working under very few rules. Self-organization is the main theme with limited restrictions from interactions among agents. Many famous examples of swarm intelligence come from the world of animals, such as birds flock, fish school and bugs swarm. The social interactions among individual agent help them to adapt to the environment more efficiently since more information are gathered from the whole swarm. A description of popular swarm intelligence algorithms can be found in [1, 2, 3].

In this project your goal is to solve the Traveling Salesperson Problem (TSP). TSP asks the following question: "Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city **exactly once** and returns to the origin city?". TSP can be modeled as an undirected weighted graph, such that cities are the graph's vertices, paths are the graph's edges, and a path's distance is the edge's weight. It is a minimization problem **starting and finishing at a specified vertex** after having visited each other vertex **exactly once**. Often, the model is a complete graph (i.e., each pair of vertices is connected by an edge). A number of real world problems can be modeled as TSP or some modifications to basic TSP (see [4] for a list of such problems).

Task Description: In this project your task is to implement **one** swarm intelligence algorithm to solve TSP and compare the results to a simple hill-climbing algorithm.

- This is a team project. You are allowed to work in groups of **at most 3 students**.
- Your team has to implement a hill-climbing algorithm and one swarm algorithm. Make sure the swarm algorithm your team chooses is different from algorithm chosen by some other group. Write the name of the team members and the algorithm your team wants to implement in the google doc.
- You are not allowed to choose to implement the PSO algorithm.
- You will run both algorithms on 3 different instances of symmetric TSP (distance from city A to city B is same as city B to city A) which you can obtain from the TSPLIB library (http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/tsp/).
- Download the following 3 files: berlin52.tsp.gz, att48.tsp.gz, pr76.tsp.gz site. The each file is a separate instance of the problem and contains the coordinates of the cities for that instance. The berlin52.opt.tour.gz, att48.opt.tour.gz and pr76.opt.tour.gz files contain the solution to each instance (that is the order of the cities in the optimal tour.)
- One issue that your code should handle is that a candidate solution should be a valid TSP tour that is the start and end city should be same and the tour should contain all the cities appearing exactly once. Also the tour can start from and end on any city.

Experiments to be done:

- For each instance of TSP: you should run the algorithms (hill-climbing and the chosen population based algorithms) for 100 times. You will execute each run for a maximum of Max_iter number of iterations.
- You will report the best result (shortest tour) you obtain within these 100 runs. You can compute the distance traveled in the optimal tour and the best tour produced by the algorithms by adding the euclidean distances of the successive cities in the tour. You will report the relative error of the solution: $error_{rel} = \frac{opt - best}{opt} * 100\%$, where opt = distance of the optimal tour and best = distance of the best tour found by the algorithm.
- Also your will repeat the process for different population sizes (50, 100, 250, 500) and plot the error-vs-population size for your algorithm. Set Max_iter to a suitably large number so that running your simulation does not take more than 5 minutes.

Report writing instructions:

- The report should contain a section on TSP and its applications, a section on the objective of this project, section on the description of the algorithms used, the experimental setup and results section and the conclusion.
- The report should not exceed two pages. Use font 11 with single line spacing.
- You should also mention the contributions of each team member. Try to use latex templates (in overleaf.com) to write your report (not mandatory).

Submission Instruction: You should upload your report and code as well as an output file that stores the best tour found.

Submission Deadline: **August 16 by 8 AM**

References:

[1] Hu Yichen, Swarm Intelligence, link https://guava.physics.uiuc.edu/~nigel/courses/569/Essays_Fall2012/Files/Hu.pdf
[2] Ab Wahab MN, Nefti-Meziani S, Atyabi A, A Comprehensive Review of Swarm Optimization Algorithms. PLoS ONE 10(5): e0122827. 2015 https://doi.org/10.1371/journal.pone.0122827 link: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0122827
[3] SwarmPackagePy, https://github.com/SISDevelop/SwarmPackagePy
[4] Matai R et al, Traveling Salesman Problem: An overview of applications, formulations, and Solution approaches. link