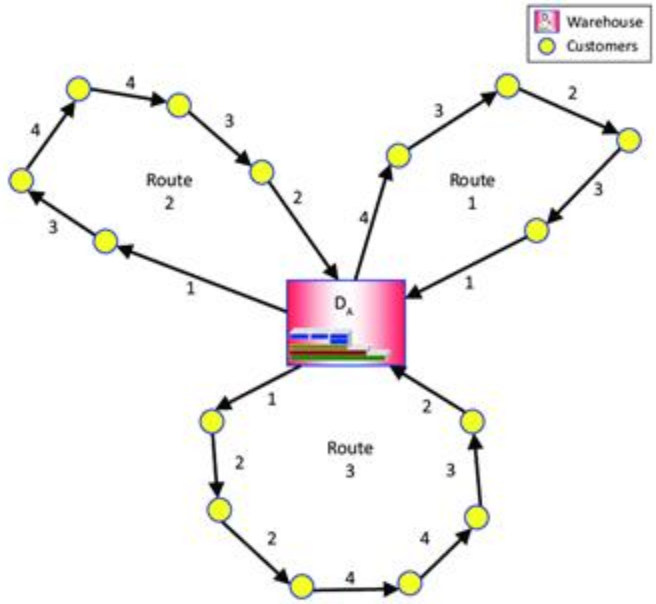
CS 451 – Computational Intelligence
Spring’ 2022
Assignment # 2 – Optimization using Swarm Intelligence
Due On: Mar 11 2022, 11:59 pm

Objective:

The assignment focuses on swarm intelligence and provides students hands-on with Ant Colony optimization(ACO) and Particle Swarm Optimization(PSO) techniques to solve complex optimization problems.

Q-1 – Capacitated Vehicle Routing using ACO [25 Points]

Vehicle routing problem (VRP) is a combinatorial optimization problem which asks "What is the optimal set of routes for a fleet of vehicles to traverse in order to deliver to a given set of customers?".



In this question, you will implement Ant Colony Optimization (ACO) technique to solve the capacitated vehicle routing problem. The problem instances to test your implementation are attached with the assignment (instances_CVRP.zip).

You can start with the following values for different parameters (α, β,γ, No. of ants) and are required to fine-tune these values to come up with the best set of parameters for the given problem:

- α: 2
- β: 2

- γ: 0.5
- No. of ants: 10

You also need to plot the following graphs to show the behavior of your implementation during the optimization process:

- Iteration vs best fitness so far
- Iteration vs avg fitness so far

You can take help from the following online resources to understand the problem and its formulation. However, once understood, you have to implement it entirely on your own.

- Solving Vehicle Routing Problem using Ant Colony Optimization
- Solving Vehicle Routing Problem with ACO

Submission:

Along with the code, you will also submit a report outlining your problem formulation and best solutions achieved for given problem instances with their graphical visualization.

Q-2 Evolving a Tic-tac-toe Agent [25 Points]

You must have played tic-tac-toe several times. In this question, you will make your program play tic-tac-toe 'without explicitly teaching it how to play'.

You will use any of the nature inspired algorithms learned so far (EA, EP, GA, ACO, or preferably PSO) to evolve the strategy of a tic-tac-toe player. The implementation of game is provided with the assignment with all game rules incorporated. You will use this implementation to evolve the strategy of your tic-tac-toe agent.

- Your strategy will be implemented in the function getMove(..) that will take the current state of the game and will return a move to be made (i.e. the cell to be marked).
- You can play multiple matches using your strategy and the number of wins/losses/ties will define the fitness of your solution.
- Your agent will start with a random strategy. You are not supposed to incorporate any game specific heuristic in your program to teach it how to play. It should purely evolve itself just by playing games.

Some heuristic based strategies are already incorporated in the attached program that you can use as an adversary to play and evolve.

Submission:

You will submit the final game and a report outlining your problem formulation, your best-so-far and avg-so-far curves and final results (in terms of % times you win against your adversary). Once submitted, we may run your strategy against the strategies evolved by your peers in the class to see which strategy is doing best.

Grading

The grading will be based on the following components:

Problem Formulation	20%
Algorithm Implementation (Correctness of code, Proper Structuring of code, Generic Implementation)	40%
Fine-tuning of parameters and final solution	10 %
Gathering statistics and plotting graph	10 %
Report – describing problem formulation, final results, solution visualization	20 %