Name: Joelle Hakim User-ID: ngnm22

Algorithm A: Ant Colony Optimisation

Algorithm B:

Description of enhancement of Algorithm A:

When looking into research papers I found that one of the most effective improvement algorithms was the min max algorithm to combat stagnation. The components of this were to set an upper bound and lower bound for the pheromones so that it would converge better.
I wrote a function to calculate stagnation, a function to apply pheromone smoothing, and added to my evaporation function to edit the matrix to apply the upper and lower bounds.
I also coded a function so that min max would only apply to points based on which iteration the loop is in, basing the figures on one of the research papers cited below.
Once we reached stagnation, I then had to decide whether I wanted to use mutation or the two opt algorithm. Though mutation is less computationally expensive, two opt gave the best result so applied this to my algorithm.
I also took the opportunity to test parameters based on the tour length, but found that as I coded originally based on the original parameters, that these were the best ones to stick to.

Description of enhancement of Algorithm B:

The first application I used was the nearest neighbour algorithm to get an original best tour and best cost as a starting point. Though this gave good results, my best result was from just using a random tour and its cost.
Secondly, instead of using an if statement for T>epsilon, I turned this into a while loop.
I also tested a range of starting temperatures and found that 45 was the best temperature to start with in my algorithm if I were to apply it to all tours.
However, on the larger tours such as 535 I found that a starting temperature of 6900 was effective.
Also, when looking at different research papers I also found different ways of updating the temperature other than the original formula, ie using linear, quadratic and logarithmic formulas. Therefore, I trialled five different formulas for temperature updates and found that for the larger tours using the starting temperature, and the number of iterations you can slow down the process of cooling, hence better overall tours.
I also tried to combat stagnation by keeping track of the number of times the best length has not been updated and used this data to decide when were the best points to start applying my mutation function.
I also found that by turning the probability elif statement into an inner if statement of the first if statement increased my tours for 535.

References:

Simulated Annealing | Eudaimonia (vamshij.com)
(3) Performance Analysis of Simulated Annealing Cooling Schedules in the Context of Dense Image Matching | Walid Mahdi - Academia.edu
FULLTEXT01.pdf (diva-portal.org)
tech-is_min-max_aco.pdf (ucsd.edu)
MAX-MIN Ant System and Local Search for theTraveling Salesman Problem - Evolutionary Computation, 1997., IEEE International Conference on (ufrj.br)