AI Lab 4 Report

Om Patil (200010036)
Hrishikesh Pable (200010037)

February 2022

1 Final Submission

Our final submission is a Ant Colony optimization based solution to TSP.

1.1 Overview

Pheromone levels of all paths are initially set to a small +ve value. A set of ants are run on this graph such that the probability that they take an edge is proportional to μ^α/d^β where μ is pheromone level, d is cost of path and α, β are constants. These ant then increase the pheromone levels on the edges which they traverse, and this increase is inversely proportional to path cost along all the edge . The same loop is run a few times until the path found by the ants stabilizes. Every iteration the pheromone is decayed a bit by ρ amount.

1.2 Constant Values

After trial and error we were able to narrow down the constant values to $\alpha = 1$, $\beta = 15$, Number of Ants = 300, $\rho = 0.2$ such that cost of the path found is minimized.

1.3 Result

As Ant Colony is stochastic a fixed path length is not found. But in general with our set of constants the path cost come out to around 1620 to 1680.

To ensure that minimum path is found on final run the algorithm is run several times until 150 seconds elapse and the minimum path found is outputted. This reduces randomness and makes sure program terminates within 300 seconds.

2 Other Methods Tried

2.1 Genetic Algorithm

In this method, we first construct a set of n candidates (randomly), which is the initial population. Then, we calculate the fitness function value (i.e. the heuristic value) for each of the candidates in the population. We create a new population from this population by selecting candidates randomly with the probability being proportional to the fitness value of the candidate. Then, we split this new population into two equal halves, and pair up the two parts. Then we do crossover between the two parents, using techniques like PMX(Partially Mapped Crossover). This gives us a new population of candidates. We repeat this procedure until the average heuristic of the population is high.We also do some mutations (2 city exchange) with a low frequency. Then we print the candidate, whose heuristic is highest among the final population.

The path cost of the final solution is around 4200, and the lowest was around 3500. Hence we had to discard this method (As it is higher than Ant Colony's path cost)

2.2 Simulated Annealing

We first set a high value of the Temperature parameter(T).We create some candidate path, and also set a heuristic function(E) to guide the search. Then, we use MoveGen function to find some random neighbour of the current path. We decide whether to select this path with probability which is given by the following value:

$$Probability = \frac{1}{1 + e^{\frac{-\Delta E}{T}}}$$

where T is the Temperature and $\Delta E = E_{final\ node} - E_{current\ node}$

After each iteration of the program, we go on decreasing the value of T using some monotonically decreasing function.

The path obtained by this method gave a path cost of around 11,000 to 12,000 which is too high compared to that obtained from Ant Colony. Hence, we discarded this method.

2.3 Nearest Neighbour

In this method we choose an initial city after which we choose to go to the nearest city and continue going to the nearest city out of the cities yet to be visited until all cities are visited. Such a path is found bringing from all cities and the minimum is considered the answer.

Using this method we were able to achieve a score of 1879.33246113244. This is much larger than Ant Colony hence the method was abandoned.

2.4 Greedy Heuristic (Modified Prims Algorithm)

This method is a modification of the Prims algorithm which is used to find the minimum spanning tree in a graph. We first sort the edges in the increasing order of edge lengths. Then we go on including the smallest edges, while making sure that it doesn't form a mini-cycle beforehand . Finally, after all 100 cities are connected by edges, we obtain the final path.

Using this method a score of 1874.7522897147103 was achieved. This is slightly better than Nearest Neighbour approach but still falls short of Ant Colony.

2.5 A*

We tried to implement A* for TSP using a state space exploration approach. That is we chose cities which had least value of f to further continue the path on. The value of f is the sum of g and h where g is the cost of the path till now whereas h is the estimated highest lower bound for remaining cost. We calculated h as the sum of average of lowest two edges of a city yet to be visited divided for all remaining cities in the path.

The method has a very high time complexity and was not able to find the solution in even in 1 hour. Hence we dropped the method.