CS 312: Artificial Intelligence Laboratory

Lab 4 report

- Ashwin Waghmare 210010060
- Tejal Ladage 210010026

Introduction

The objective of this assignment is to find the shortest tour, visiting all the given cities. This problem is commonly known as the 'Travelling Salesman Problem'.

Methodology

We have used Ant Colony Optimization to build a tour. For each iteration of the algorithm, each ant builds a tour based on the following factors:

1) Visibility of the city
2) Amount of pheromone trail
3) Whether the city is visited or not

Probability of ant k, currently at city i, choosing a next city j is given by,

$$P_{ij}^{k}(t) = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum_{s=Allowed_k} \tau_{is}^{\alpha}(t)\eta_{is}^{\beta}(t)} & \text{if } j \in Allowed_k(t) \\ 0 & \text{Otherwise} \end{cases}$$

$Allowed_k$: feasible neighbourhood of ant k when being at city i, which are the set of cities that ant k has not visited yet.

$\tau_{ij}(t)$: Intensity of pheromone w.r.t. time.

$\eta_{ij}(t)$: visibility of city j from i.

$\eta_{ij} = 1/d_{ij}$: heuristic value that gives more priority to closer city

d_{ij} : length of arc (i, j)

α, β : hyper-parameters

$\alpha = 0, \beta > 1$: the closest city is selected

$\alpha > 1, \beta = 0$: only pheromone is used

$\tau_{ij}(t + n) = \rho\tau_{ij}(t) + \Delta\tau_{ij}(t, t + n)$, where ρ is evaporation rate.

$$\Delta\tau_{ij}(t, t + n) = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t, t + n)$$

Where $\tau_{ij}^{k} = Q/L_k$, if ant k visits city j from i, and 0 otherwise.

Pseudocode

Initialize pheromone level in all edges to a small value.

```
while (run 500 times):
        for each ant:
                construct tour using nextCity()
        store best tour
        update pheromone matrix
return best tour
```

Iterative Improvements

To improve our results we have varied the visibility factor, pheromone factor and evaporation factor.

When considering Euclidean distances, values of parameters like alpha=0.05, beta=25 and rho=0.70 were set to get optimum results.

As for Non-Euclidean distances we set alpha=25, beta=100 and rho=0.70 to optimize the solution.

Also, we concluded that Ants colony Optimization can find best solution for graphs with lesser number of nodes. But for larger graphs the solution is stuck at local optimum and does not reach the global optimum.