AI Lab 3: Travelling Salesman Problem

Divy Jain, Karthik Hegde (Group 15)

February 7, 2023

1 Introduction

Given a set of cities (coordinates) and distances between them, we have to find the shortest tour (visiting all cities exactly once and returning to the origin city) in a given amount of time.

1.1 State Space

For a given complete graph with N cities, there are $(N-1)!$ possible tours. This is the state space of TSP. For 100 cities the state space would comprise of 99! states.

2 Algorithms

We have implemented ant colony optimization to find the shortest tour of the given complete graph. All the ants in the colony move in an independent tour where the next city is chosen with some probability that is calculated using the pheromone concentration and the visibilty factor. We chose ant colony optimization technique because there is only a small chance for the algorithm to get stuck in a local optima.

Then we optimized the solution further from ACO using various local optimization techniques. These techniques are 2-city exchange, 3-city exchange, 2-Opt, 3-Opt and city insert.

1. **2-city exchange -** picks and swap two cities at random from path
2. **3-city exchange -** picks and swap three cities at random from path
3. **2-Opt -** picks two edges at random and interconnects them
4. **3-Opt -** picks three edges at random and interconnects them
5. **City insert -** picks a city at random and insert it somewhere in path

2.1 Pseudo code: ACO

```
function ACO(graph)
    Initialize()
    while not terminated do
        foreach ant do
            start city = random city
            while cities in ant's path < total cities do
                next city = selectNextCity()
            updateBestTour()
        pheromoneUpdate()
    repeat
end procedure
```

2.2 Pseudo code: Local Optimization

```
curr = solution
best = solution

while not terminated do
    p=random(0,1)
    if p < 0.1 do
        next = twoCitySwap(curr)
    else if p < 0.2 do
        next = threeCitySwap(curr)
    else if p < 0.4 do
        next = 2_Opt(curr)
    else if p < 0.9 do
        next = 3_Opt(curr)
    else do
        next = insertCity(curr)

    if cost(next) < cost(best) do
        best = next
```

2.3 Formulae

Transition probability to choose next city j for kth ant is given by:-

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum_{j \in allowed_k} \tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}, & j \in allowed_k \\ 0, & otherwise \end{cases}, \tag{1}$$

The change in pheromone level for every edge at iteration t+1 is:-

$$\Delta\tau_{ij}^{k}(t+1) = \begin{cases} \frac{Q}{L_k}, & \text{k}th\ ant\ passes\ route\ (i,\ j) \\ 0, & otherwise \end{cases}, \tag{2}$$

Update the pheromone level by the following formula:-

$$\tau_{ij}^{k}(t+1) = (1-p)\tau_{ij}^{k}(t) + \Delta\tau_{ij}^{k}(t+1). \tag{3}$$

3 Observations

Choosing the right parameters is very crucial to get an optimal cost. We tried running the code for various combinations of these parameters and here are our observations :-

- α **is pheromone factor**, which denotes the influence of pheromone concentration to the path choosing When it equals to 0, the ant currently selects completely according to greedy rule for path planning
- Weight β **is the heuristic factor**, denoted as the influence of distance of two cities to the path choosing. When it equals to 0, the path choosing depends entirely on the pheromone concentration.
- for Local optimization each optimization technique is choosen uniformly with equal probability.