CS 214: Artificial Intelligence Lab
Assignment 3 - Report

Tanishq Trivedi (210010056), Prajwal Biradar (210020037)

February 7, 2023

1 Abstract

Given a set of cities (coordinates) and distances between them, find the best (shortest) tour (visiting all cities exactly once and returning to the origin city) in a given amount of time, viz. Travelling Salesman Problem using Ant Colony Optimization.

2 Domain Description

2.1 State Space

The search space consists of all the valid tours.

2.2 Goal

The goal of the travelling salesperson problem is to find the Hamiltonian cycle with the least cost.

3 Ant Colony Optimization

In the ant colony optimization algorithms, an artificial ant is a simple computational agent that searches for good solutions to a given optimization problem.

In the first step of each iteration, each ant stochastically constructs a solution, i.e. the order in which the edges in the graph should be traversed. In the second step, the paths found by the different ants are compared. The last step consists of updating the pheromone levels on each edge.

This will lead to the ants choosing better paths, i.e. paths with lower costs, in the later iterations.

After constructing a tour in n time steps (n is the number of cities), each ant k

deposits an amount of pheromone Q/L_k on the edges it traversed.
Here, L_k is the cost of the tour it found, and Q is an appropriate constant.
Now, the total pheromone deposited on the path between two cities i and j is $\Delta\tau_{ij}(t, t+n)$.

The total pheromone on edge i-j is updated as :
$\tau_{ij}(t+n) = (1-\rho) * \tau_{ij}(t) + \Delta\tau_{ij}(t, t+n)$.

On a city i, the k^{th} ant moves to city j with a probability given by:

$$P_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha * [\eta_{ij}]^\beta}{\sum_{h \in allowed_k(t)} [\tau_{ih}(t)]^\alpha * [\eta_{ih}]^\beta}$$

Here, η_{ij} is known as the visibility, which is inversely proportional to the distance between the two cities i and j.

Algorithm 1 ACO Function

```
procedure TSP-ACO()
    while some termination criteria do
        bestTour ← nil
        randomly place M ants on N cities
        for each ant a do                                        ▷ construct tour
            for n ← 1 to N do
                ant a selects an edge from the distribution P_n^a
            end for
        end for
        for each ant a do                                        ▷ update pheromone
            for each edge (u,v) in the ant's tour do
                deposit pheromone α 1/tour-length on edge (u,v)
            end for
        end for
        return bestTour
    end while
end procedure
```

4 Parameter Assignment and Optimal Values

4.1 For euclidean distances

The below table shows the best Hamiltonian cycle produced in different cases:

	$\alpha = 3, \beta = 3$	$\alpha = 3, \beta = 4$	$\alpha = 3, \beta = 5$	$\alpha = 3, \beta = 6$
euc 100	1631	1609	1580	1626
euc 250	2627	2702	2740	2683
euc 500	4158	4441	4116	4121

From the above table, it's clear that $\alpha = 3$ and $\beta = 6$ give the best results.

4.2 For non-euclidean distances

For the below table, $\alpha = 5$ and $\beta = 10$, the number of ants is M, and the number of cities is N.

	$M = N$	$M = N/2$	$M = N/3$	$M = N/4$
noneuc 100	5256	5237	5238	5239
noneuc 250	13402	12887	12817	12809
noneuc 500	27940	27918	27716	27705

It's clear that when $M = N/4$, the best results are obtained.

For the below table the number of ants = the number of cities.

	$\alpha = 5, \beta = 5$	$\alpha = 5, \beta = 10$	$\alpha = 5, \beta = 15$	$\alpha = 5, \beta = 20$
noneuc 100	5263	5256	5240	5231
noneuc 250	15093	13402	12783	12756
noneuc 500	31604	27940	26615	26335

It's clear that when $\alpha = 5, \beta = 20$, the best results are obtained.

Note:- α should always be greater than 1, β should always be greater than 0 and α should always be greater than β

5 Two Edge Exchange

Building the new route and calculating the distance of the new route can be a very expensive operation, usually O(n), where n is the number of vertices in the route. This can sometimes be skipped by performing a O(1) operation. Since a 2-opt operation involves removing two edges and adding two different edges, we can subtract and add the distances of only those edges.

6 Conclusion

Ant Colony Optimization is derived from the way ants behave in real life, and it's an extremely useful and effective algorithm which can be used to solve problems like the Travelling Salesperson Problem.

Algorithm 2 Two edge exchange

```
procedure 2-OPT()
    for city1 in range(1,N-2) do
        for city2 in range(city1+1, N) do
            old path = cost(city1-1,city1) + cost(city2-1,city2)
            new path = cost(city1-1,city2-1) + cost(city1,city2)
            lengthDelta = new path - old path
            if lengthDelta < 0 then swap edges
            end if
        end for
    end for
end procedure
```