CS 312 Assignment 4 Report; Team - 21

Josyula V N Taraka Abhishek (200010021), M V Karthik (200010030)

February 6, 2022

1 Introduction

In this task - Travelling salesman problem, with a given a set of cities (coordinates) and distances between them, find the best (shortest) tour given that visiting all cities exactly once and returning to the origin city in a 300 seconds had to be found.

2 Algorithm

Pseudocode:

```
    Ants = []
for i in range(N):
    Ants.append(ANT(g))

g = Graph()

def colonize():
    for ant in Ants:
        ant.getPath(g)

    g.updatePhermone()
    g.updateProb()
```

By Brute force, we found this set of values are optimal:

- Q = 100
- alpha = 0.9
- beta = 20
- rho = 0.3

Time required is 300 seconds.

2.1 Avoiding local minima:

If best solution has not changed for more than 30 seconds then we reset the probability matrix to switch.

Pseudocode:

```
if time.time() - Graph.solchange > timedelay:
            self.Phermones = numpy.ones((N, N))
```

3 How to run?

The code should be run as:

./run.sh input.txt

Output will be printed in output.txt.

4 Our approaches

we have tried these algorithms:

4.1 Ant colony optimization:

General ACO algorithm runs till 300 seconds and outputs a solution with cost nearly 1700

4.2 A* algorithm

Taken too much time to run and couldn't complete in given 300 seconds

4.3 Lin Keringhan algorithm

we explored a new algorithm here. Not enough time to check the correctness and optimality of the implementation,

4.4 Modified ACO

Finally, we made few changes to our initial ACO algoritm which are avoiding local minima Best costs till now:

- euc_100 : 1690
- neuc_100 : 5280