Maze Simulation

Prakhar Aggarwal, Dhairya Gupta

July 29, 2021

1 Travelling Salesman Problem

We are given a list of cities and the distances between each pair of cities. Our goal is to find the shortest possible route that visits each city exactly once and returns to the origin city covering the least distance. The Travelling Salesman Problem is NP-complete i.e. there is no known polynomial time solution for this problem.

1.1 Task

We are given an weighted undirected Graph G = (V,E). Every edge (u,v) ϵ E is assigned a non-negative cost c(u,v). We have to find a **hamiltonian cycle** (A path that visits each vertex exactly once and there is an edge from the last vertex to the starting vertex). We will define c(A) as the total cost of the edges in A where A $\subseteq$ E.

1.2 Representation

We can represent the graph as an adjacency matrix. The (i,j) element of this matrix denotes the c(u,v). c(u,u) is considered to be 0 i.e. there is no self-loop.

2 Approach

As Travelling Salesman Problem is NP-complete, there exists no polynomial time algorithm. We will look at some algorithms which give near optimum solutions with.

2.1 Brute Force Method

In brute force approach, we will explore all possible cycles which connect all nodes, calculate each cycle's cost and take the one with minimum cost.

Runtime Analysis: Time complexity for this approach = Total Number of Cycles * Time taken to calculate cycle's cost. As our source node is fixed, we have remaining n-1 nodes. As the graph is a complete graph, each node has n-1 edges. So source node has (n-1) options, selected node will have (n-2) options and so on. So, total number of cycles = (n-1)*(n-2)*.....*2*1 i.e. Total Number of Cycles = (n-1)!. Assuming that we calculate each cycle's cost as we traverse over it, we can find a cycle's cost in O(1) time.

Hence, **Time Complexity = O(n!)**. This solution is not feasible enough to work on real life examples.

2.2 Approximated Method

Using the approximated method, we can find a lower bound to the optimum solution.