Problem 1: Traveling Salesman

A complete graph G has V vertices and E edges.
Find a Traveling Salesman Problem solution with k as max cost.
Prove it's non-deterministic polynomial (NP) complete.

Proof

Let G be a complete, undirected, weighted graph with V vertices and E edges:

$$G = (V, E), \text{ where } v \in V, \text{ and } e \in E$$

Let k be an integer, where $k \in \mathbb{Z}$ and k is the max cost or $\sum e$ to visit all vertices once.
Let $c(v_1, v_2)$ be the cost to visit each vertex, where $c \in \mathbb{Z}$ and v_1 and v_2 are some vertices.

This proof shows the Traveling Salesman Problem is NP-Complete by meeting two criteria:

1. TSP is NP — i.e., it has a solution verifiable as polynomial time;
2. TSP is NP-Hard — i.e., it's as hard as the hardest NP problem.

Sub-Proof 1: TSP is NP

G must have each vertex once, with edge costs summing to k at most.

G contains each vertex, since $G = (V, E)$, and $\sum e$ is at most k.

$$V = \{v_1, v_2, v_3, \dots\ v_n\}, \text{ where } n \text{ is the number of vertices}$$

These calculations can be trivially done in polynomial time, so TSP is NP.

Continued on next page

Problem 1 (Continued)

Proof (Continued)

Sub-Proof 2: TSP is NP-Hard

TSP is NP-Hard if another problem similar in complexity can be reduced to it. This sub-proof shows how G has a Hamiltonian cycle that visits all vertices once and has a cycle length $\leq k$.

Let G' be a weighted graph.

$$G' = (V, E'), \text{ where}$$
$$E' = \{(v_1, v_2) : v_1, v_2 \in V \text{ and } v_1 \neq v_2\}$$

If an edge is in G and G', assign weight zero.
Otherwise, assign weight one.

$$c(v_1, v_2) = \left\{ \begin{array}{l} 0, \text{ if } (v_1, v_2) \in E, \\ 1, \text{ if } (v_1, v_2) \notin E \end{array} \right\}$$

G has a Hamiltonian cycle if the cycle G' contains all vertices once with $k \leq 0$.

Because edges in G' have weight one, if a cycle visits all vertices once and has length ≤ 0, that means it only has edges from G.

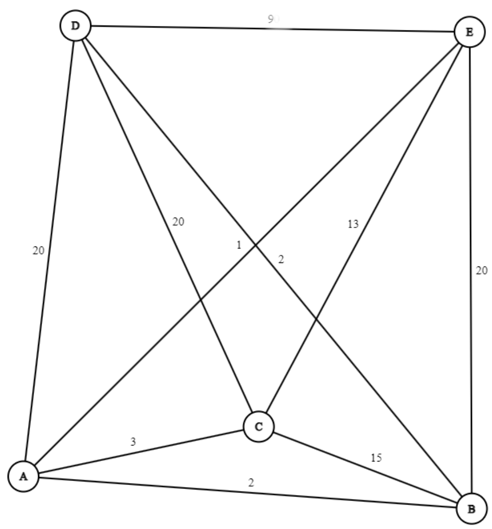
So, G has a Hamiltonian cycle G' of max cost $k \leq 0$. This proves a solution exists for TSP.

Problem 2: Nearest Neighbor Heuristic

	A	B	C	D	E
A	0	2	3	20	1
B	2	0	15	2	20
C	3	15	0	20	13
D	20	2	20	0	9
E	1	20	13	9	0

Cities $\{A, B, C, D, E\}$ and distances

2a) Weighted Graph



Continued on next page

Problem 2 (Continued)

Approximate Nearest Neighbor

With the nearest-neighbor heuristic:

$$\{A, E, D, B, C, A\}$$

$$\begin{aligned} 1 + 9 + 2 + 15 + 3 \\ &= 10 + 17 + 3 \\ &= 27 + 3 \\ &= 30 \end{aligned}$$

2c) Find Approximation Ratio

Optimal route:

$$\{A, C, E, D, B, A\}$$

$$\begin{aligned} 3 + 13 + 9 + 2 + 2 \\ &= 16 + 11 + 2 \\ &= 27 + 2 \\ &= 29 \end{aligned}$$

$$p(n) = \frac{C}{C^*} = \frac{30}{29} = 1.03\ (approx)$$

2d) Nearest Neighbor Implementation

See external file TSP.py

Bibliography

OpenDSA data structures and Algorithms Modules Collection. 28.19. Reduction of Hamiltonian Cycle to Traveling Salesman - OpenDSA Data Structures and Algorithms Modules Collection. (n.d.). https://opendsa-server.cs.vt.edu/ODSA/Books/

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2009. Introduction to Algorithms, Third Edition (3rd. ed.). The MIT Press.