

Traveling Salesman

Given a set of cities $(\{1, \dots, n\})$ and a symmetric matrix $C = (c_{ij})$, $c_{ij} \geq 0$ that specifies for every pair $(i, j) \in [n] \times [n]$ the cost for travelling from city i to city j . Find a permutation π of the cities such that the round-trip cost

$$c_{\pi(1)\pi(n)} + \sum_{i=1}^{n-1} c_{\pi(i)\pi(i+1)}$$

is minimized.

Traveling Salesman

Theorem 3

There does not exist an $O(2^n)$ -approximation algorithm for TSP.

Hamiltonian Cycle:

For a given undirected graph $G = (V, E)$ decide whether there exists a simple cycle that contains all nodes in G .

- ▶ Given an instance to HAMPATH we create an instance for TSP.
- ▶ If $(i, j) \notin E$ then set c_{ij} to $n2^n$ otw. set c_{ij} to 1. This instance has polynomial size.
- ▶ There exists a Hamiltonian Path iff there exists a tour with cost n . Otw. any tour has cost strictly larger than $n2^n$.
- ▶ An $O(2^n)$ -approximation algorithm could decide btw. these cases. Hence, cannot exist unless $P = NP$.

Metric Traveling Salesman

In the metric version we assume for every triple $i, j, k \in \{1, \dots, n\}$

$$c_{ik} \leq c_{ij} + c_{jk} .$$

It is convenient to view the input as a complete undirected graph $G = (V, E)$, where c_{ij} for an edge (i, j) defines the distance between nodes i and j .

TSP: Lower Bound I

Lemma 4

The cost $\text{OPT}_{\text{TSP}}(G)$ of an optimum traveling salesman tour is at least as large as the weight $\text{OPT}_{\text{MST}}(G)$ of a minimum spanning tree in G .

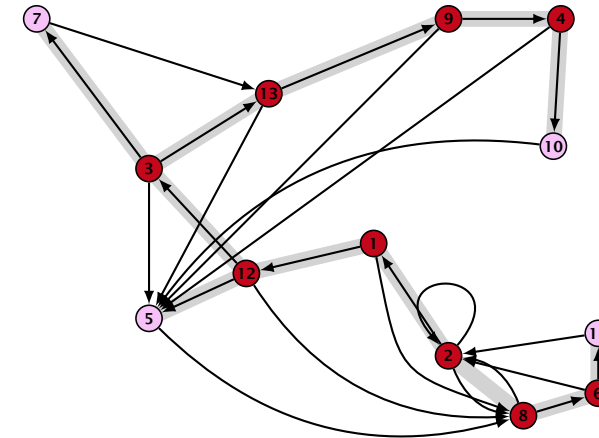
Proof:

- ▶ Take the optimum TSP-tour.
- ▶ Delete one edge.
- ▶ This gives a spanning tree of cost at most $\text{OPT}_{\text{TSP}}(G)$.

TSP: Greedy Algorithm

- ▶ Start with a tour on a subset S containing a single node.
- ▶ Take the node v closest to S . Add it S and expand the existing tour on S to include v .
- ▶ Repeat until all nodes have been processed.

TSP: Greedy Algorithm



TSP: Can we do better?

Duplicating all edges in the MST seems to be rather wasteful.

We only need to make the graph Eulerian.

For this we compute a Minimum Weight Matching between odd degree vertices in the MST (note that there are an even number of them).

TSP: Can we do better?

An optimal tour on the odd-degree vertices has cost at most $\text{OPT}_{\text{TSP}}(G)$.

However, the edges of this tour give rise to two disjoint matchings. One of these matchings must have weight less than $\text{OPT}_{\text{TSP}}(G)/2$.

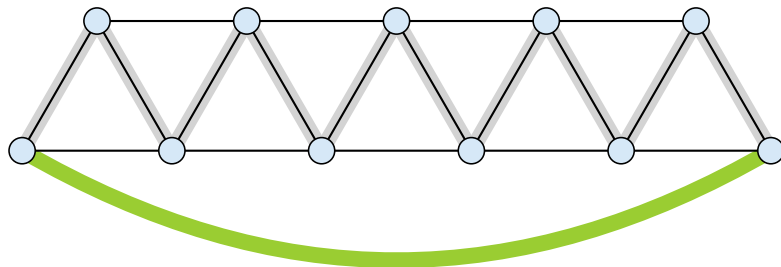
Adding this matching to the MST gives an Eulerian graph with edge weight at most

$$\text{OPT}_{\text{MST}}(G) + \text{OPT}_{\text{TSP}}(G)/2 \leq \frac{3}{2} \text{OPT}_{\text{TSP}}(G) ,$$

Short cutting gives a $\frac{3}{2}$ -approximation for metric TSP.

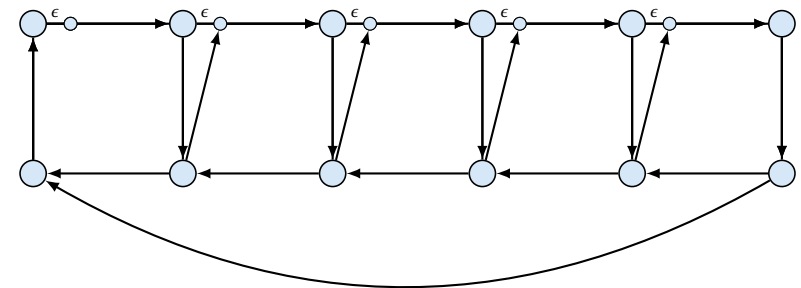
This is the best that is known.

Christofides. Tight Example



- ▶ optimal tour: n edges.
- ▶ MST: $n - 1$ edges.
- ▶ weight of matching $(n + 1)/2 - 1$
- ▶ MST+matching $\approx 3/2 \cdot n$

Tree shortcutting. Tight Example



- ▶ edges have Euclidean distance.