Fall 2016, MATH-566

Traveling Salesman Problem

Let $G = (V, E)$ be a complete graph on n vertices. Let $c : E \to \mathbb{R}^+$. Find a closed cycle/circuit C through all vertices of minimum cost.

$$\min \left\{ \sum_{e \in C} c(e) : C \text{ is a circuit of all vertices on } G \right\}$$

The problem is NP-complete. We will try for vertices on the plane (triangle inequality satisfied and distance is positive)

Heuristics

- *Nearest neighbor:* Build a path, always include the nearest neighbor. On test data gives 1.26 of optimum.

  1: Show that the nearest neighbor algorithm can do arbitrarily bad it no triangle-inequality

  Worst case if triangle-inequality is satisfied $\frac{1}{3}(\log_2(n+1) + \frac{4}{9})$ times optimum.

- *Cheapest insertion:* Start with an edge and keep adding vertices one by one that are cheapest to insert. At most 2 times optimum.

- *Furthest insertion:* Start with longest edge and keep adding vertices one by one that are furthest away. At most $\log_2 n + 1$ times optimum. Better in experiments than previous.

  note: No instance is known, where insertion method would do worse than 4 times the optimum.

- *Christofides Heuristics:* Start with Minimum Spanning Tree. Add Minimum Matching to vertices of odd degree. Vertices of degree at least 4 can split off. Does at most $\frac{3}{2}$ of optimum.

  2: Show that the upper bound of the algorithm is at most $\frac{3}{2}$ of optimum.

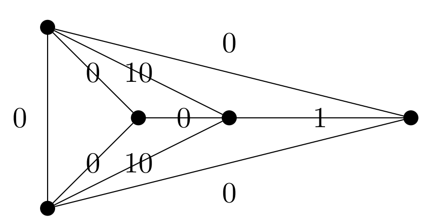
Tour improvements

- *2-optimal switch:* Replace 2 edges by different 2 edges. (more generally, k-switch)

- *Lin-Kernighan:* Better way of doing 2-switches.

Lower Bounds

- *Held-Karp:* Find a vertex v and minimum spanning tree T in $G - v$, then add v to T by using to smallest cost edges adjacent to v. Modify cost of edges/vertices and rerun. Try to make costs such that every vertex is in exactly two edges.



- *Linear-programming:* Can be used to provide a relaxation of integer programing version. (can be modified to match Held-Karp)