Lecture 26

Coping with NP-Completeness

Is your real problem a special subcase?

- E.g. 3-SAT is NP-complete, but 2-SAT is not; ditto 3- vs 2-coloring
- E.g. you only need planar graphs, or degree 3 graphs, …?

Guaranteed approximation good enough?

- E.g. Euclidean TSP within 2 * Opt in poly time

Fast enough in practice (esp. if n is small),

- E.g. clever exhaustive search like backtrack, branch & bound, pruning

Heuristics – usually a good approximation and/or usually fast

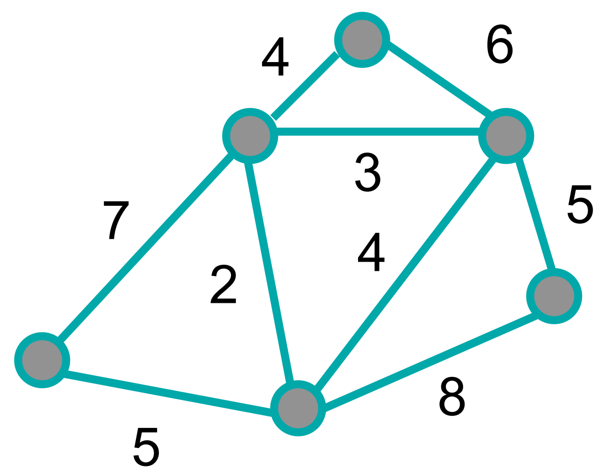
NP-complete problem: TSP

Input: An undirected graph G=(V,E) with integer edge weights, and an integer b.

Output: YES iff there is a simple cycle in G passing through all vertices (once), with total cost ≤ b.

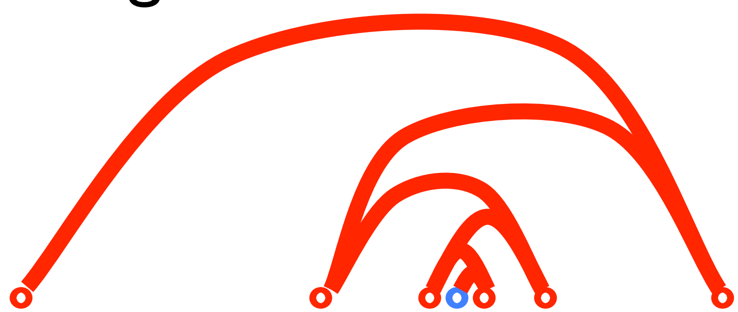
Example:

b = 34



TSP - Nearest Neighbor Heuristic

NN Heuristic –go to nearest unvisited vertex



Fact: NN tour can be about (log n) x opt, i.e.

$$\lim_{n \to \infty} \frac{NN}{OPT} \to \infty$$

(above example is not that bad)

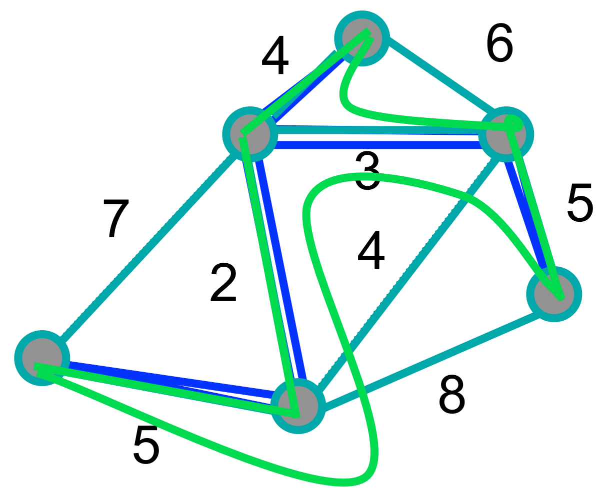
2x Approximation to EuclideanTSP

A TSP tour visits all vertices, so contains a spanning tree, so TSP cost is > cost of min spanning tree.

Find MST

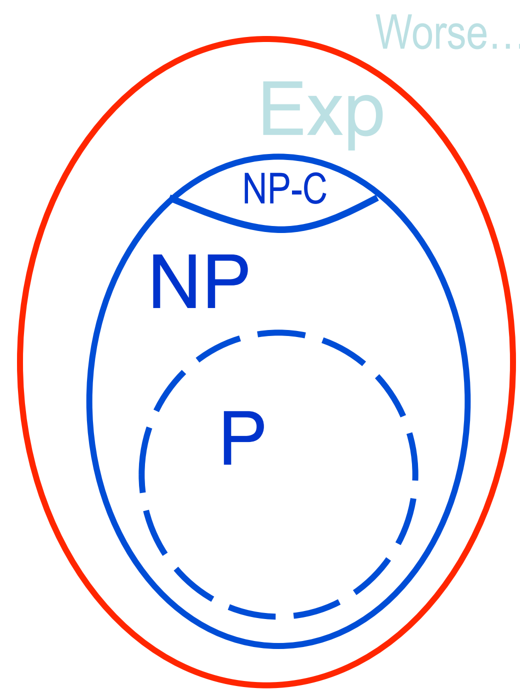
Find "DFS" Tour

Shortcut



TSP ≤ shortcut < DFST = 2 * MST < 2 * TSP

Summary



Big-O – good

P – good

Exp – bad

Exp, but hints help? NP

NP-hard, NP-complete – bad (I bet)

To show NP-complete – reductions

NP-complete = hopeless? – no, but you need to lower your expectations: heuristics & approximations.

Beyond NP

Many complexity classes are worse, e.g. time 2^{2^n}, $2^{2^{2^n}}$, …

Others seem to be "worse" in a different sense, e.g., not in NP, but still exponential time. E.g., let

L_P = "assignment y satisfies formula x", $\in P$

Then :

SAT = $\{ x \mid \exists y \; \langle x,y \rangle \in L_P \}$

UNSAT = $\{ x \mid \forall y \; \langle x,y \rangle \in L_P \}$

$QBF_k = \{ x \mid \exists y_1 \forall y_2 \exists y_3 \ldots \partial_k \; \langle x, y_1 \ldots y_k \rangle \in L_P \}$

$QBF_\infty = \{ x \mid \exists y_1 \forall y_2 \exists y_3 \ldots \quad \langle x, y_1 \ldots \quad \rangle \in L_P \}$