Hard Problems

What do you do when your problem is **NP-Hard**? Give up?

(A) Solve a special case!
(B) Find the hidden parameter!
(Fixed parameter tractable problems)
(C) Find an approximate solution.
(D) Find a faster exponential time algorithm: $n^{O(n)}$, 3^n, 2^n, etc.

Part I

Traveling Salesperson Problem

TSP

TSP-Min

Instance: $G = (V, E)$ a complete graph, and $\omega(e)$ a cost function on edges of G. **Question:** The cheapest tour that visits all the vertices of G exactly once.

Solved exactly naively in $\approx n!$ time.
Using DP, solvable in $O(n^2 2^n)$ time.

TSP Hardness

Theorem

*TSP-Min can not be approximated within **any** factor unless* $NP = P$.

Proof.

1. Reduction from **Hamiltonian Cycle** into **TSP**.
2. $G = (V, E)$: instance of Hamiltonian cycle.
3. H: Complete graph over V.
$$\forall u, v \in V \quad w_H(uv) = \begin{cases} 1 & uv \in E \\ 2 & \text{otherwise.} \end{cases}$$
4. $\exists$ tour of price n in $H \iff \exists$ Hamiltonian cycle in G.
5. No Hamiltonian cycle $\implies$ TSP price at least $n + 1$.
6. But... replace 2 by cn, for c an arbitrary number

TSP Hardness - proof continued

Proof.

1. Price of all tours are either:
   (i) n (only if $\exists$ Hamiltonian cycle in $\mathbf{G}$),
   (ii) larger than $cn + 1$ (actually, $\geq cn + (n-1)$).
2. Suppose you had a poly time c-approximation to TSP-Min.
3. Run it on $\mathbf{H}$:
   (i) If returned value $\geq cn + 1 \implies$ no Ham Cycle since $(cn+1)/c > n$
   (ii) If returned value $\leq cn \implies$ Ham Cycle since $OPT \leq cn < cn + 1$
4. c-approximation algorithm to TSP $\implies$ poly-time algorithm for **NP-Complete** problem. Possible only if **P = NP**.

□

TSP with the triangle inequality

Because it is not that bad after all.

$\mathrm{TSP}_{\triangle\neq}$-Min

Instance: $\mathbf{G} = (V, E)$ is a complete graph. There is also a cost function $\omega(\cdot)$ defined over the edges of $\mathbf{G}$, that complies with the triangle inequality.
Question: The cheapest tour that visits all the vertices of $\mathbf{G}$ exactly once.

triangle inequality: $\omega(\cdot)$ if

$$\forall u, v, w \in \mathsf{V}(\mathsf{G}), \qquad \omega(u,v) \leq \omega(u,w) + \omega(w,v).$$

Shortcutting

σ: a path from s to t in $\mathbf{G} \implies \omega(st) \leq \omega(\sigma)$.

TSP with the triangle inequality

Continued...

Definition

Cycle in $\mathbf{G}$ is ***Eulerian*** if it visits every edge of $\mathbf{G}$ exactly once.

Assume you already seen the following:

Lemma

A graph $\mathbf{G}$ has a cycle that visits every edge of $\mathbf{G}$ exactly once (i.e., an Eulerian cycle) if and only if $\mathbf{G}$ is connected, and all the vertices have even degree. Such a cycle can be computed in $O(n+m)$ time, where n and m are the number of vertices and edges of $\mathbf{G}$, respectively.

TSP with the triangle inequality

Continued...

1. C_{opt} optimal **TSP** tour in $\mathbf{G}$.
2. **Observation**:
   $\omega(C_{\mathrm{opt}}) \geq \mathrm{weight}\big(\text{cheapest spanning graph of } \mathbf{G}\big)$.
3. MST: cheapest spanning graph of $\mathbf{G}$.
   $\omega(C_{\mathrm{opt}}) \geq \omega(\mathrm{MST}(\mathbf{G}))$
4. $O(n \log n + m) = O(n^2)$: time to compute MST.
   $n = |\mathsf{V}(\mathsf{G})|$, $m = \binom{n}{2}$.

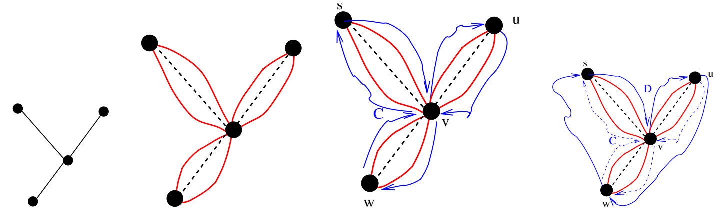
TSP with the triangle inequality

2-approximation

1. $\boldsymbol{T} \leftarrow \mathrm{MST}(\mathbf{G})$
2. $\mathbf{H} \leftarrow$ duplicate every edge of $\boldsymbol{T}$.
3. $\boldsymbol{H}$ has an Eulerian tour.
4. $\mathbf{C}$: Eulerian cycle in $\boldsymbol{H}$.
5. $\omega(\mathbf{C}) = \omega(\boldsymbol{H}) = 2\omega(\boldsymbol{T}) = 2\omega(\boldsymbol{MST}(\mathbf{G})) \leq 2\omega(\boldsymbol{C}_{\mathrm{opt}})$.
6. π: Shortcut $\mathbf{C}$ so visit every vertex once.
7. $\omega(\pi) \leq \omega(\mathbf{C})$

TSP with the triangle inequality

2-approximation algorithm in figures



(a) (b) (c) (d)

Euler Tour: vuvwvsv
First occurrences: vuvwvsv
Shortcut String: vuwsv

TSP with the triangle inequality

2-approximation - result

Theorem

$\mathbf{G}$: *Instance of* $TSP_{\triangle\neq}$*-Min.*
$\boldsymbol{C}_{\mathrm{opt}}$: *min cost TSP tour of* $\mathbf{G}$.
$\implies$ *Compute a tour of* $\mathbf{G}$ *of length* $\leq 2\omega(\boldsymbol{C}_{\mathrm{opt}})$.
Running time of the algorithm is $O\left(n^2\right)$.

$\mathbf{G}$: $\boldsymbol{n}$ vertices, cost function $\omega(\cdot)$ on the edges that comply with the triangle inequality.

TSP with the triangle inequality

3/2-approximation

Definition

$\mathbf{G} = (\boldsymbol{V}, \boldsymbol{E})$, a subset $\boldsymbol{M} \subseteq \boldsymbol{E}$ is a ***matching*** if no pair of edges of $\boldsymbol{M}$ share endpoints.
A ***perfect matching*** is a matching that covers all the vertices of $\mathbf{G}$.
$\boldsymbol{w}$: weight function on the edges. ***Min-weight perfect matching***, is the minimum weight matching among all perfect matching, where

$$\omega(\boldsymbol{M}) = \sum_{e \in M} \omega(e)\,.$$

TSP with the triangle inequality

3/2-approximation

The following is known:

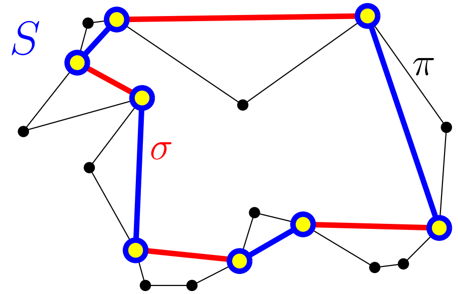
Theorem

Given a graph **G** *and weights on the edges, one can compute the min-weight perfect matching of* **G** *in polynomial time.*

Min weight perfect matching vs. TSP

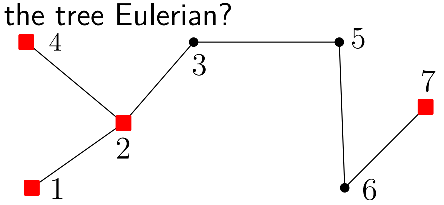
Lemma

$\mathbf{G} = (\mathbf{V}, \mathbf{E})$: *complete graph.*
$S \subseteq \mathbf{V}$: *even size.*
$\omega(\cdot)$: *a weight function over* $\mathbf{E}$.
$\implies$ *min-weight perfect matching in* $\mathbf{G}_S$ *is* $\leq \omega(\mathrm{TSP}(\mathbf{G}))/2$.



A more perfect tree?

1. How to make the tree Eulerian?



2. Pesky odd degree vertices must die!
3. Number of odd degree vertices in a graph is even!
4. Compute min-weight matching on odd vertices, and add to MST.
5. $\mathbf{H} = \mathrm{MST} + \mathbf{min} - \mathbf{weight} - \mathbf{matching}$ is Eulerian.
6. Weight of resulting cycle in $\mathbf{H} \leq (3/2)\omega(\mathrm{TSP})$.

Even number of odd degree vertices

Lemma

The number of odd degree vertices in any graph G' *is even.*

Proof:

$\mu = \sum_{v \in V(G')} d(v) = 2|E(G')|$ and thus even.
$U = \sum_{v \in V(G'), d(v) \text{ is even}} d(v)$ even too.
Thus,

$$\alpha = \sum_{v \in V, d(v) \text{ is odd}} d(v) = \mu - U = \text{even number},$$

since μ and U are both even.
Number of elements in sum of all odd numbers must be even, since the total sum is even.

$3/2$-approximation algorithm for TSP

The result

Theorem

Given an instance of TSP with the triangle inequality, one can compute in polynomial time, a $(3/2)$*-approximation to the optimal TSP.*