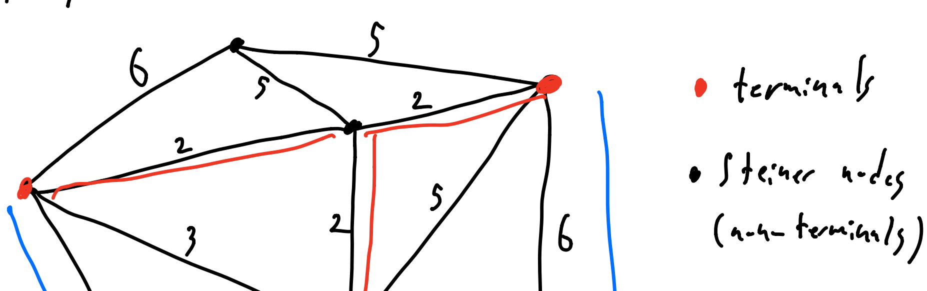
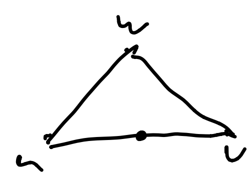
Steiner Tree:

- Input:
  - Graph $G = (V, E)$
  - Costs $c : E \to \mathbb{R}^+$
  - Terminals $T \subseteq V$
- Feasible solutions: $F \subseteq E$ s.t. F connected, spans all terminals
- Objective: $\min \sum_{e \in F} c(e) = \min_F c(F)$

MST: Steiner tree with $T = V$
shortest path: ST with $T = \{s, t\}$





Def: $d : V \times V \to \mathbb{R}_{\geq 0}$ is a metric space on V if:

- $d(u,v) = 0$ iff $u = v$
- $d(u,v) = d(v,u) \quad \forall u,v \in V$
- $d(u,v) \leq d(u,w) + d(w,v) \quad \forall u,v,w \in V$ (triangle inequality)

Metric Steiner Tree: (special case of ST on a metric space)

- Input: V, metric $c: V \times V \to \mathbb{R}_{\geq 0}$ on V, terminals $T \subseteq V$
- Feasible: $F \subseteq V \times V$ s.t. F connected, spans all terminals
- Objective: $\min \sum_{e \in F} c(e)$

Thm: If there is an α-approx for Metric ST, then there is an α-approx for Steiner Tree

Def: The metric completion c' of $(G=(V,E),c)$ is the metric on V where $c'(u,v)$ is the cost of the shortest path between u and v under edge lengths c

Lemma: Let H be a solution (a Steiner Tree) for Steiner Tree problem on input (G,c,T). Then H solution to Metric ST problem on input (V,c',T) with $c'(H) \leq c(H)$

Pf: H feasible for metric: ✓

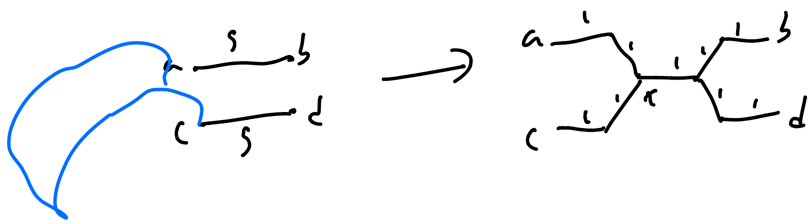
$c'(u,v) \leq c(u,v)$ by def of c' $\forall u,v \in V$

$\Rightarrow c'(H) = \sum_{e \in H} c'(e) \leq \sum_{e \in H} c(e) = c(H)$ ✓

Lemma: Let H' be a solution to Metric Steiner Tree on (V, c', T). Then there is some solution H to Steiner Tree on (G, c, T) with $c(H) \le c'(H')$, and given H' we can find H in polytime.

Pf: Replace each $\{u,v\} \in H'$ by shortest u–v path in G

$\Rightarrow$ subgraph $\hat{H}$ of G, $c(\hat{H}) \le c'(H')$



Let H arbitrary spanning tree of $\hat{H}$

$\Rightarrow c(H) \le c(\hat{H}) \le c'(H')$

Pf of reduction thm:

Let A α-approx for metric ST. Given input (G, c, T), run A on (V, c', T) to get H', use previous lemma to get H.

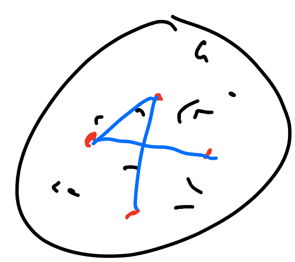
Let OPT_{metric} be opt solution for (V, c', T)

OPT be opt solution for (G, c, T)

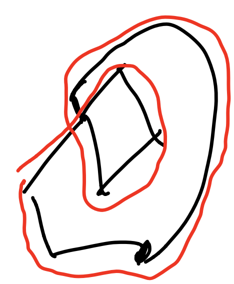
$$
\begin{aligned}
c(H) &\le c'(H') && \text{(lemma)} \\
&\le \alpha \cdot c'(OPT_{metric}) && \text{(def of } \mathcal{A}\text{)} \\
&\le \alpha \cdot c'(OPT) && \text{(def of } OPT_{metric}\text{)} \\
&\le \alpha \cdot c(OPT) && \text{(first lemma)}
\end{aligned}
$$

So just need to design good alg for metric case

ALG:



- Return $F = MST$ on terminals!



Claim: F is valid solution

Pf: Trivial: connected and spans T

Def: G is Eulerian if there is a closed tour that uses every edge exactly once

Thm: G is Eulerian iff connected, all degrees even (even holds for multigraphs).

Thm: ALG is a $2(1-\frac{1}{|T|})$-approximation

Pf:

Let F^* optimal solution.

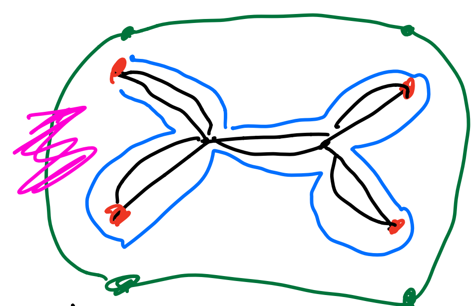
WTS: $c(F) \le 2(1-\frac{1}{|T|}) \cdot c(F^*)$

Plan: Find <u>some</u> spanning tree $\hat{F}$ of T s.t.

$c(\hat{F}) \le 2(1-\frac{1}{|T|}) \cdot c(F^*)$

$\Rightarrow c(F) \le c(\hat{F})$ since F MST of T

start with F^*



Double every edge: $2F^*$

All degrees even: Eulerian!

Tour C which uses every edge:

$c(C) = c(2F^*) = 2c(F^*)$

"Shortcut" C to only use terminals, see each terminal once: cycle H

Triangle inequality:

$$c(H) \le c(C)$$

$c(H) = \sum_{e \in H} c(e)$

$\Rightarrow \max_{e \in H} c(e) \ge \frac{1}{|H|} c(H)$

Remove heaviest edge of H: path $\hat{F}$

$$c(\hat{F}) \le \left(1 - \frac{1}{|T|}\right) c(H) \le 2\left(1 - \frac{1}{|T|}\right) c(F^*) \quad \checkmark$$

Metric TSP:

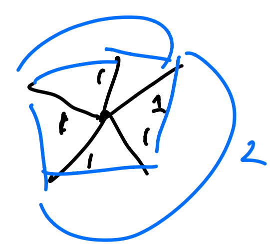
Input: Metric space (V, c)

Feasible: Hamiltonian cycle H ← cycle visiting all nodes once

Objective: $\min\ c(H) = \sum_{e \in H} c(e)$

Alg 1:

- Compute MST T
- Double T to get $2T$
- $2T$ Eulerian, so Eulerian tour C
- Shortcut C to get H



Thm: $2(1-\frac{1}{n})$-approx

Pf: Just like Steiner Tree!

Let H^* optimal solution,

F path from removing heaviest edge from H^*

$$\Rightarrow c(H) \underset{\text{shortcut}}{\leq} c(C) = c(2T) = 2c(T) \underset{F \text{ a spanning tree}}{\leq} 2c(F)$$

$$\leq 2(1-\tfrac{1}{n})\,c(H^*)$$

Want to do better: <u>Christofides' Algorithm</u>

why did we lose 2?

- Doubling MST

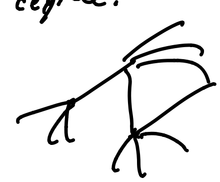
why did we do that?

- Make it Eulerian

cheaper way to make MST Eulerian?

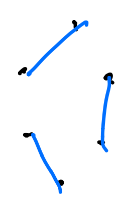
Problem: odd degree nodes

<u>Lemma</u>: Let $G=(V,E)$ be a graph. Then there are an even # nodes with odd degree.



<u>Pf</u>:

$$\sum_{v \in V} d(v) = 2|E| \quad (\text{even})$$



<u>Def</u>: A perfect matching of $S \subseteq V$ is a matching on S of size $\frac{|S|}{2}$ (every node in S matched to other node in S)

Fact: Can find min-cost perfect matchings in poly time

Christofides':

- Compute MST T
- Let D be odd-degree nodes in T
- Compute min-cost perfect matching M of D
- Let C be Eulerian tour of $T+M$
- Return $H=$ shortcutted C

Claim: Everything well-defined

Thm: $\frac{3}{2}$-approximation

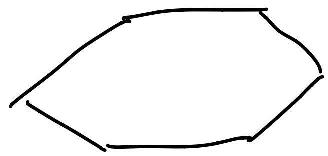
Pf: Let H^* optimal solution

$c(T) \leq c(H^*)$

$c(H) \leq c(C) = c(T) + c(M) \leq c(H^*) + c(M)$

So WTS $c(M) \leq \frac{1}{2} c(H^*)$

Shortcut H^* to D, get H_D



$|D|$ even, so partition into "evens" M_1 and "odds" M_2

- Each a perfect matching of D

$c(M_1) + c(M_2) = c(H_D)$

$c(M) \le \min(c(M_1), c(M_2)) \le \frac{1}{2} c(H_D) \le \frac{1}{2} c(H^*)$