TSP

Theorem

No α-approximation ~~algo~~ algorithm for TSP where α is a constant.

Proof

- reduce HC to α-approx TSP
- given an instance of HC with $G = (V, E)$ construct an instance of TSP with graph G' that is a complete graph with the same set of vertices.
- set edge weights for G' as

$$w(e) = \begin{cases} 1 & \text{if } e \in E \\ \alpha n & \text{otherwise} \end{cases}$$

- If G has a HC the cost of the optimal tour in G' has cost n
- If G does not have a HC then the optimal tour in G' must contain some edge not in G and hence has cost $> \alpha n$

Metric TSP

$$d(x,y) = d(y,x)$$ undirected

$$d(x,y) \geq 0$$

$$d(x,y) + d(y,z) \geq d(x,z)$$ triangle inequality

2- approx

$\frac{3}{2}$ - approx

Define

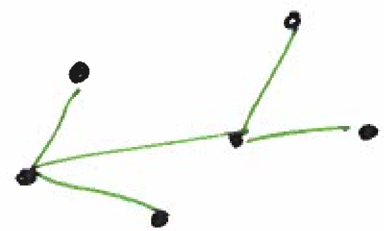
$c(S)$ total cost of a set of edges

H_G^* – minimum weight Hamiltonian Cycle

$c(H_G^*)$ – length of the optimal solution

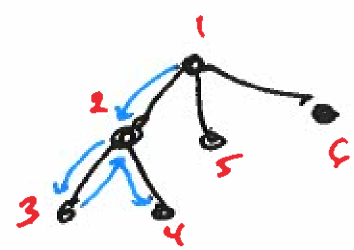
2-approx

- Using MST as our start



- clearly not a cycle
- DFS

  1, 2, 3, 2, 4, 2, 1, 5, 1, 6, 1



- Visits all vertices (some more than once)

- Bypass duplicate vertices

Given the triangle inequality this will not increase the path length.

$1, 2, 3, \cancel{2}, 4, \cancel{2}, \cancel{1}, 5, \cancel{1}, 6, 1$

$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 1$

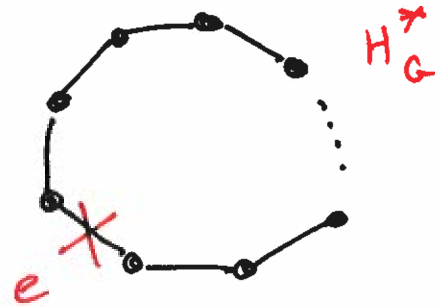
- Compute MST T
- Construct path P using DFS
- Strip duplicates to make P' $\quad c(P') \leq c(P)$

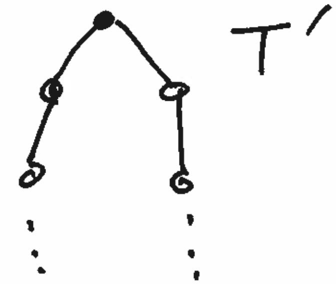
what is $c(P)$?

$$c(P) = 2c(T)$$

$$c(P') \leq 2c(T)$$

How does $c(T)$ relate to $c(H^*_G)$?





Deleting an edge from H^*_G creates a spanning tree. (not necessarily the MST)

$$c(T') \geq c(T)$$

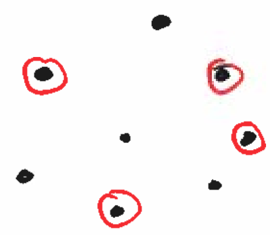
$$c(H^*_G) \geq c(H^*_G - e) \geq c(T)$$

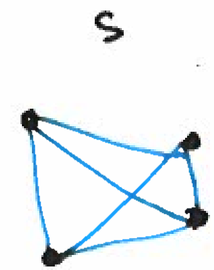
$$c(P') \leq 2c(H^*_G)$$

Christofides Algorithm

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

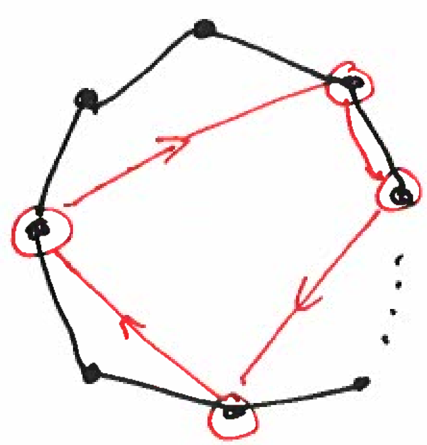
$S \subseteq V$





S has a Hamiltonian cycle

$c(H^*_S) \leq c(H^*_G)$ ← only true because of the triangle inequality



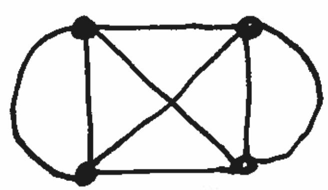
Perfect Matching

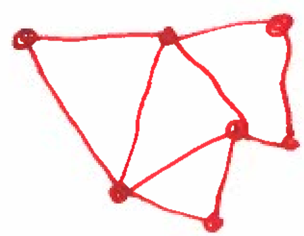
every vertex has exactly one edge

minimum cost perfect matching can be found in poly time

Euler Circuit

A circuit that traverses every edge exactly once

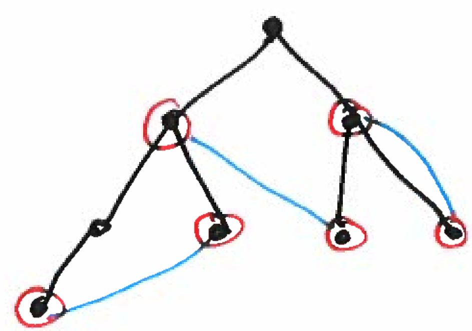




- The degree of every node must be even for a graph to have an Euler Circuit.

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

- Add edges to the MST so that the graph has an Euler circuit.



S - set of odd degree vertices

- use perfect matching on S to increase the degree of all vertices by 1

Can we have an odd number of vertices with odd degree? No

$\Sigma d_i = 2|E|$

- Compute MST T
- $S \leftarrow$ set of odd degree vertices
- Find perfect matching on S M
- add edges in M to T
- compute Euler Circuit E
- strip duplicate vertices E'

$$c(E) = c(T) + c(M)$$

$$c(E') \leq c(E)$$

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

$c(M)$?

H_S^*

Let M_1 be the red set

Let M_2 be the blue

The red and blue sets are both perfect matchings

$$c(M) \le c(M_1)$$

$$c(H_S^*) = c(M_1) + c(M_2)$$

$$c(M) \le c(M_2)$$

$$\begin{aligned} c(M) &\le \tfrac{1}{2}\big(c(M_1) + c(M_2)\big) \\ &\le \tfrac{1}{2} c(H_S^*) \\ &\le \tfrac{1}{2} c(H_G^*) \end{aligned}$$

$$\begin{aligned} c(E') &\le c(T) + c(M) \\ &\le c(H_G^*) + \tfrac{1}{2} c(H_G^*) \\ &\le \tfrac{3}{2} c(H_G^*) \end{aligned}$$