Approximation algorithms

Let P be an optimization problem, and I be an input.

- P is computationally hard (e.g. NP-hard) to solve optimally.
- aim to design efficient algorithms that output a "good" solution for P.

General idea:

Identify a different optimization problem P', s.t.

- $P' \approx P$ on every input I
- P' can be solved efficiently

Then solve P' instead of P.

How do we define "$P' \approx P$"?

- Suppose P is a minimization problem. $OPT(I)$ be the value of the optimal solution to input I for P. Then, if the value of the optimal solution for P', $OPT'(I)$

satisfies

$$OPT(I) \le OPT'(I) \le C \cdot OPT(I).$$

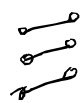
then this is a C-approximation.

Vertex Cover

Given $G=(V,E)$, find the smallest $S \subseteq V$ s.t. every edge is incident on a vertex in S.

(NP-hard)

P': matching.



$VC \ge |\text{matching}|$

Lemma: Let M be any matching of G, S be a vertex cover. $|S| \ge |M|$.

<u>Proof</u>. Every edge in M must be covered by a different vertex in S. □

<u>Def</u>. A matching M is called <u>maximal</u> if no edge e can be added to M s.t. $M \cup \{e\}$ is also a matching.

A maximal matching can be computed in $O(m+n)$ time.

- Greedily add edges, as long as the new edge does not share a vertex with the previous edges.

Lemma: Let M be a maximal matching. S be the set of vertices consisting both endpoints of edges in M. Then S is a vertex cover.

<u>Proof</u>. If e is not covered by S, then e doesn't share vertices with all edges in M. $M \cup \{e\}$ would be a matching. Contradiction. $\square$

Two lemmas $\Rightarrow$

$$\mathrm{OPT} \leq 2 \cdot |M| \leq 2 \cdot \mathrm{OPT}.$$

Outputting endpoints of a maximal matching is 2-approximation.

Travelling Salesman Problem

Given an undirected $G=(V,E)$ with edge

cost $c: E \to \mathbb{R}^{\geq 0}$, find a min cost Hamiltonian Cycle.

Metric TSP

Further assume c satisfies the triangle inequality.

$$c(x,y) \leq c(x,z) + c(z,y)$$

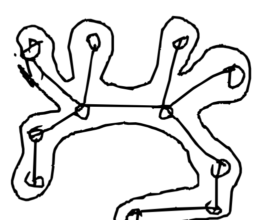
P': MST

Lemma: $|MST| \leq |TSP|$.

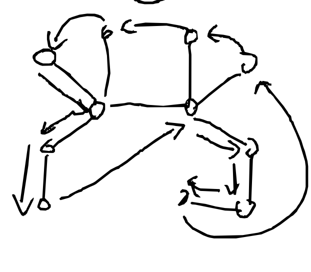
Proof. Any TSP solution is a Hamiltonian path + one edge.

A Hamiltonian Path is a spanning tree $\geq$ MST. $\square$

Consider a DFS tour of the MST.



$\Rightarrow$



Shortcut the tour by skipping visited vertices.

- Triangle inequality $\Rightarrow$ shortcutting reduces the total cost.
- Turn into a Hamiltonian Cycle.

The DFS tour visits each edge of MST exactly twice.

$$\underbrace{2|MST|}_{\leq\, 2\cdot|TSP|} = |DFS\ tour| \geq |Short\ cut\ DFS\ tour|.$$

This gives a 2-approximation.

Christofides Algo.

DFS tour $\Leftrightarrow$ double each edge & find an Eulerian tour.

It suffices to add some edges to MST s.t. an Eulerian tour exists.

Add a matching to all odd deg vertices!

Approx-TSP

$T :=$ the MST of G

$S :=$ the set of odd deg vertices in T.

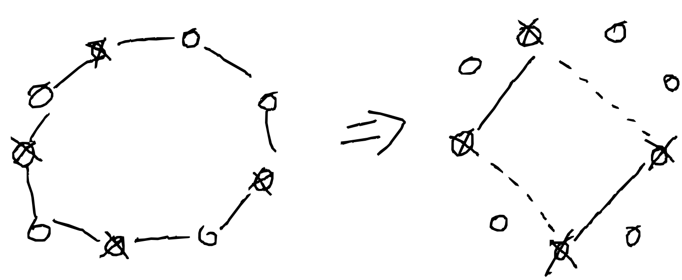
$M :=$ the min cost perfect matching between vertices in S.

Compute an Eulerian tour of $T \cup M$.

Return the shortcut Eulerian tour.

Lemma: $|M| \leq \frac{1}{2} \cdot |TSP|$.

Proof sketch: A Hamiltonian Cycle
$\Rightarrow$ two matchings.



$\otimes \in S$

$|\text{Approx-TSP}| \leq |T| + |M| \leq 1.5 |TSP|$.

1.5-approximation.