Approximation Algorithms

Lecture 4

Last Time

- Dual Fitting
- Randomized Rounding

Last Time

- ❑ Dual Fitting
- ❑ Randomized Rounding

Today

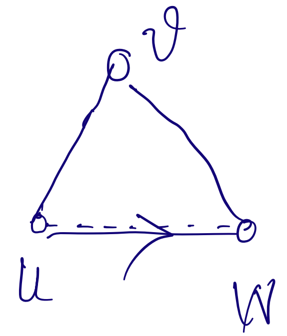
Greedy algorithms for:

- ❑ the k-Center problem
- ❑ the metric TSP problem

Christofides-Serdyukov approximation algorithm for metric TSP

k-Center problem

- Given a complete undirected graph on n vertices with lengths on edges
- Length of edge $\{u, v\}$ is $\ell(u, v)$ for $u \neq v$
- Lengths satisfy triangle inequality:
    - For $u, v, w \in [n]$, it holds that $\ell(u, v) + \ell(v, w) \geq \ell(u, w)$



-

k-Center problem

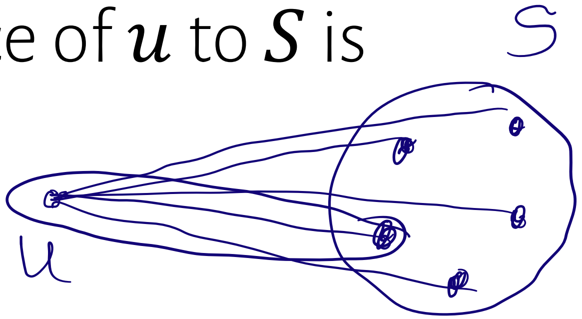
similarity measure b/w vertices

objects

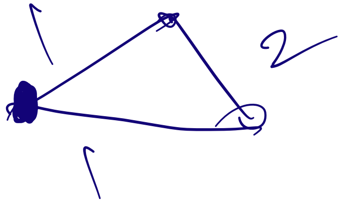
- Given a complete undirected graph on n vertices with lengths on edges
- Length of edge $\{u, v\}$ is $\ell(u, v)$ for $u \neq v$
- Lengths satisfy triangle inequality:
  - For $u, v, w \in [n]$, it holds that $\ell(u, v) + \ell(v, w) \geq \ell(u, w)$

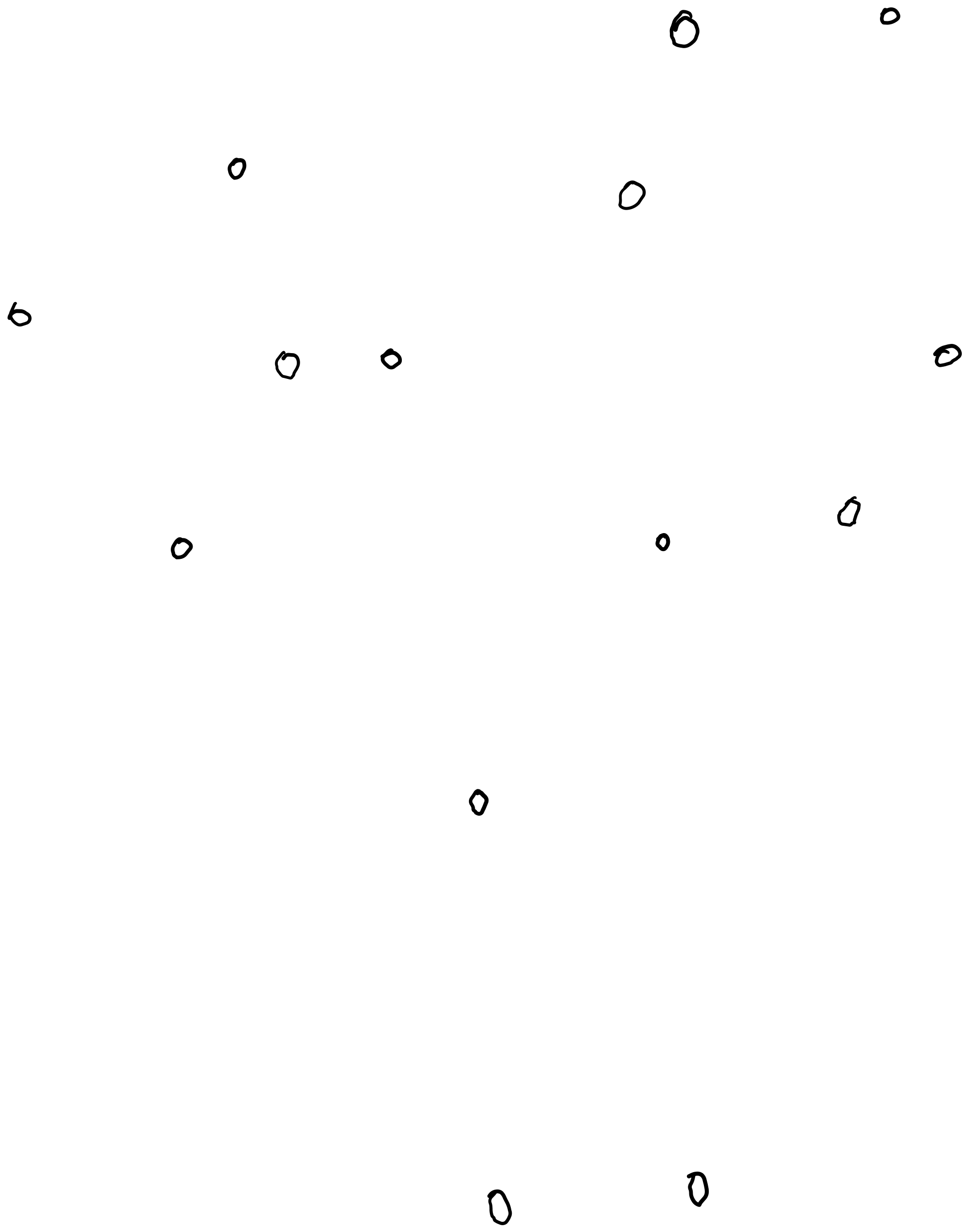
- For a vertex u and set $S \subseteq [n]$ of vertices, the distance of u to S is

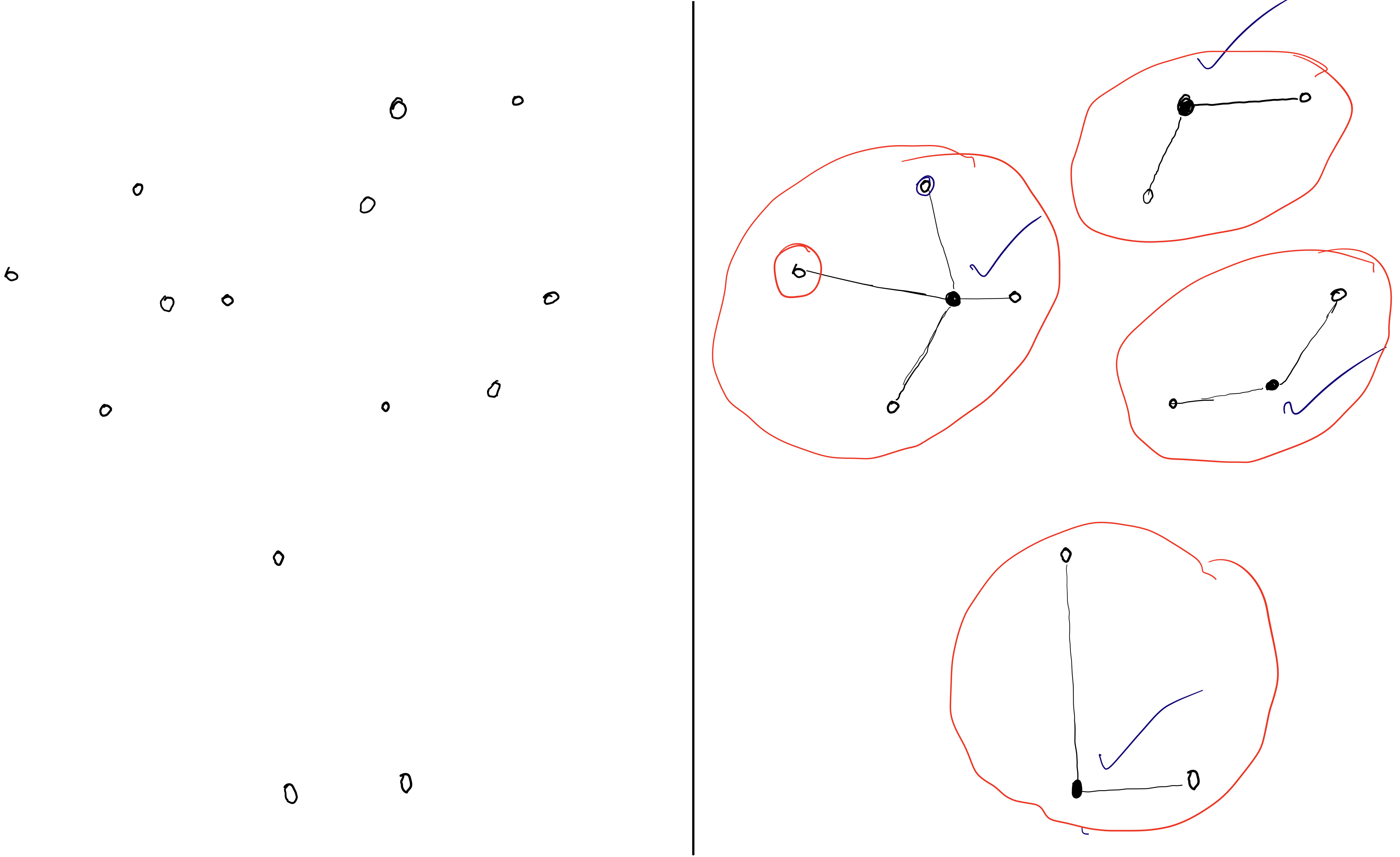
$$\ell(u, S) = \min_{v \in S} \ell(u, v)$$



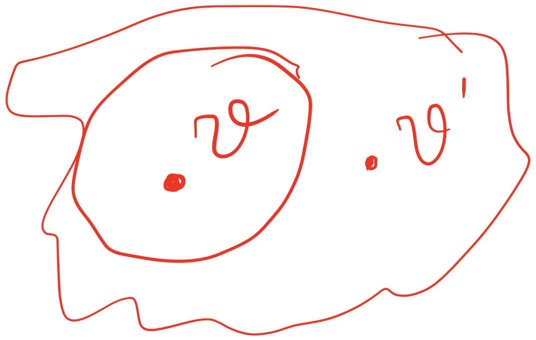
- **Goal**: Output a set S of k "centers" such that $\max_{u \in [n]} \ell(u, S)$ is minimized







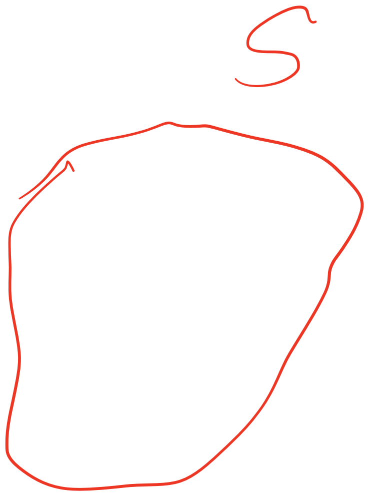
Algorithm



- ❑ Pick an arbitrary vertex $v \in [n]$ and $S \leftarrow \{v\}$
- ❑ while $|S| \leq k$:
    - ❑ Pick the vertex $w \in [n]$ that maximizes $\ell(w, S)$
    - ❑ $S \leftarrow S \cup \{w\}$

w

- ❑ Output S



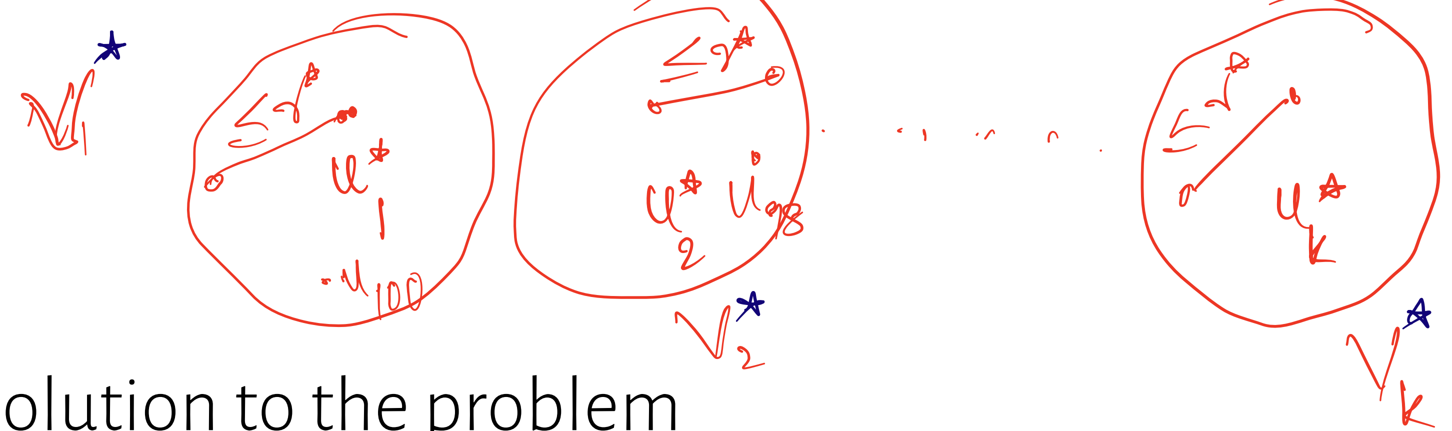
Algorithm

Suppose the optimal radius is r^*

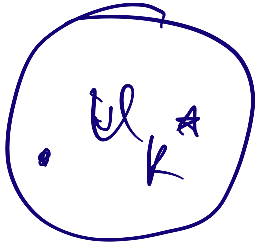
- Pick an arbitrary vertex $v \in [n]$ and $S \leftarrow \{v\}$
- while $|S| \leq k$:
  - Pick the vertex $w \in [n]$ that maximizes $\ell(w, S)$
  - $S \leftarrow S \cup \{w\}$
- Output S

Our solution has radius $\leq 2r^*$

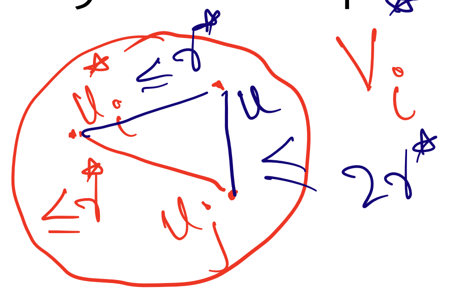
This greedy algorithm gives a 2-approximation for the k-center problem.



Analysis



- Let $u_1^*, \ldots, u_k^*$ be an optimal solution to the problem
- These centers partition $[n]$ into k clusters $V_1^*, \ldots, V_k^*$
- Let optimal radius be r^*
- Let $u_1, \ldots, u_k$ be the centers picked by the greedy
- If there is one greedy-center per optimal cluster, then

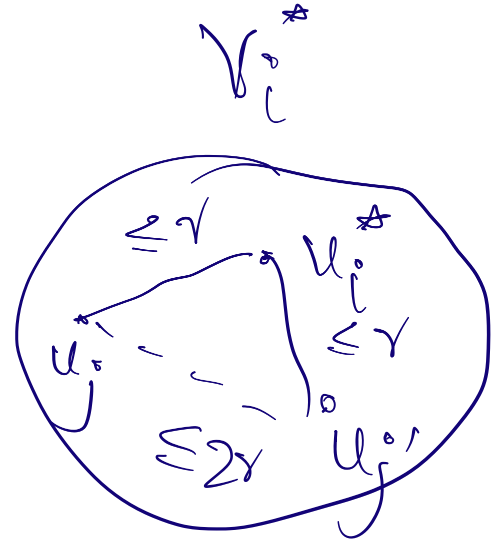


In each optimal cluster, distance of vertices to greedy center $\le 2r^*$

- Otherwise

Consider the first greedy center u_j s.t.

$u_j \in V_i^*$ and $u_{j'} \in V_i^*$ for $j' < j$

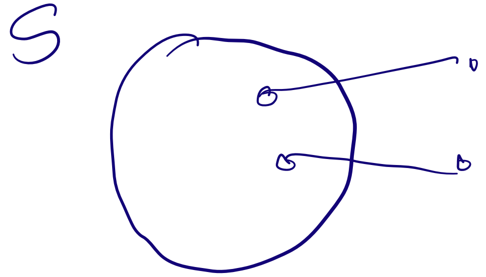


greedy-centers

$$l(u_j, u_{j'}) \leq 2r^*$$

Now, u_j was the point that was furthest from all greedy-centers so far

$$\Rightarrow \forall v \in [n] \quad l(v, S) \leq l(u_j, S) \leq l(u_j, u_{j'})$$



- It is NP-hard to approximate within a factor better than $\mathbf{2}$

- Reduction from dominating set problem: Given a graph G and a parameter k, is there a dominating set in G of size at most k ?

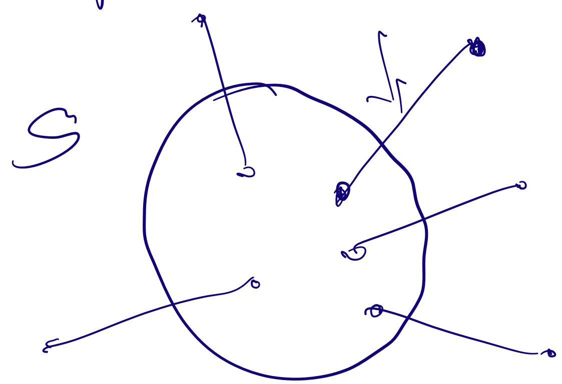
every vertex in $V \setminus S$ has some nbr in S

- Construct a complete graph with weight $\mathbf{1}$ for edges in G and weight $\mathbf{2}$ for nonedges in G

- If we can approximate k-center in this graph to a factor of $\rho < 2$, then we can solve the dominating set problem exactly.

$$(G, k) \longrightarrow K_n \text{ with weights}$$

* Suppose G has a dom. set of size $\leq k$



this same set (plus some spurious vertices) is a soln. to k-center problem instance with radius $\underline{1}$.

* Suppose G does not have a dom. set of size $\leq k$, then the opt. radius is 2 for k-center instance

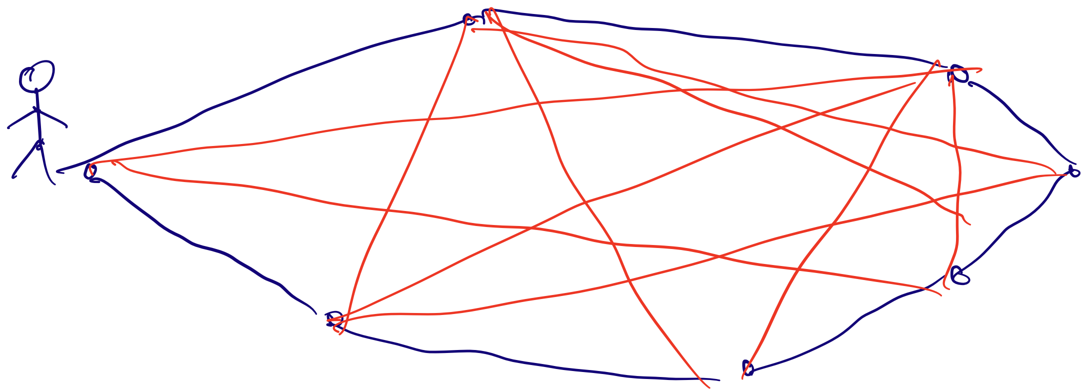
Metric Travelling Salesman Problem (TSP)

- Given a complete undirected graph on n vertices with costs on edges
- Cost of edge $\{u, v\}$ is $\ell(u, v)$ for $u \neq v$
- Costs satisfy triangle inequality:
    - For $u, v, w \in [n]$, it holds that $\ell(u, v) + \ell(v, w) \geq \ell(u, w)$
-

Metric Travelling Salesman Problem (TSP)

- Given a complete undirected graph on n vertices with costs on edges
- Cost of edge $\{u, v\}$ is $\ell(u, v)$ for $u \neq v$
- Costs satisfy triangle inequality:
  - For $u, v, w \in [n]$, it holds that $\ell(u, v) + \ell(v, w) \geq \ell(u, w)$

- **Goal**: Find a tour of minimum total cost that visits every vertex exactly once and returns to the starting vertex



- What is the state of the art?

- NP hard to approximate within a factor of $\frac{123}{122}$ $= 1.008\ldots$

  [Karpinski, Lampis, Schmied '15]

- $\frac{3}{2} \to 1.5$

  $\frac{3}{2} - \epsilon$ factor approximation algorithm for some $\epsilon > 10^{-36}$ $<$

  [Karlin, Klein, Oveis Gharan '21]

-

- What is the state of the art?

- NP hard to approximate within a factor of $\frac{123}{122}$

  [Karpinski, Lampis, Schmied '15]

- $\frac{3}{2} - \epsilon$ factor approximation algorithm for some $\epsilon > 10^{-36}$

  [Karlin, Klein, Oveis Gharan '21]

1.5

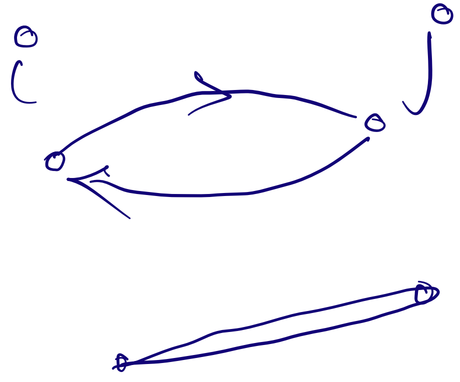
- Today: $\frac{3}{2}$ - approximation algorithm by

  [Christofides '76] & [Serdyukov '78]

US

USSR

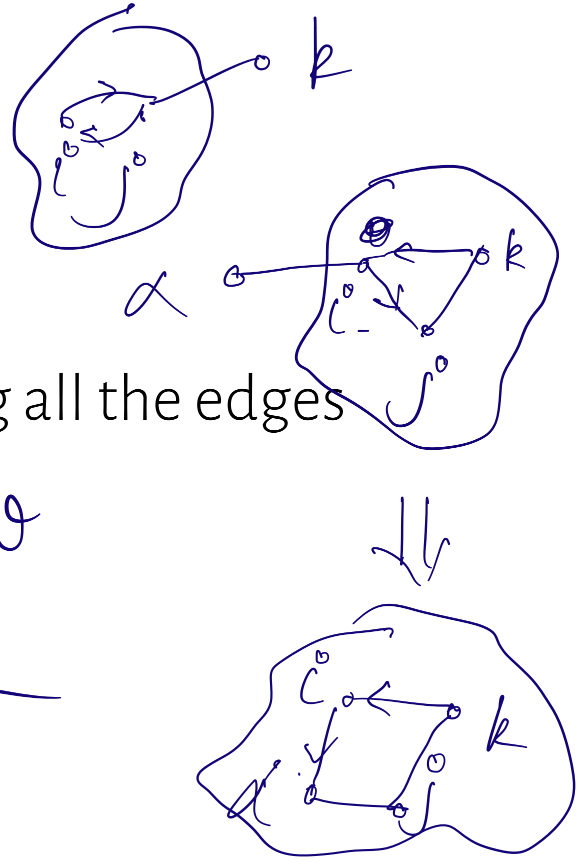
- First result: 2-factor approximation for metric TSP

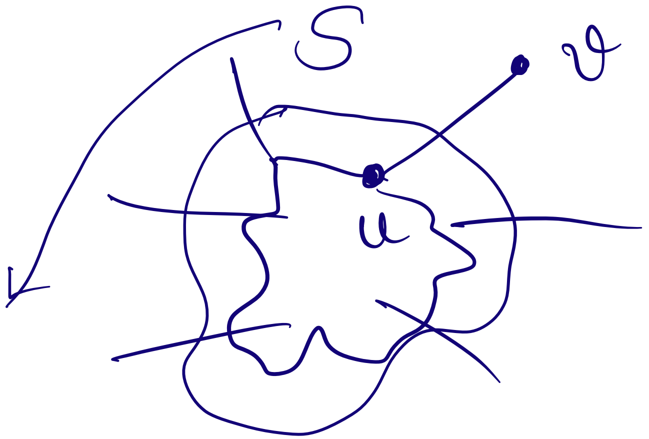


Greedy algorithm

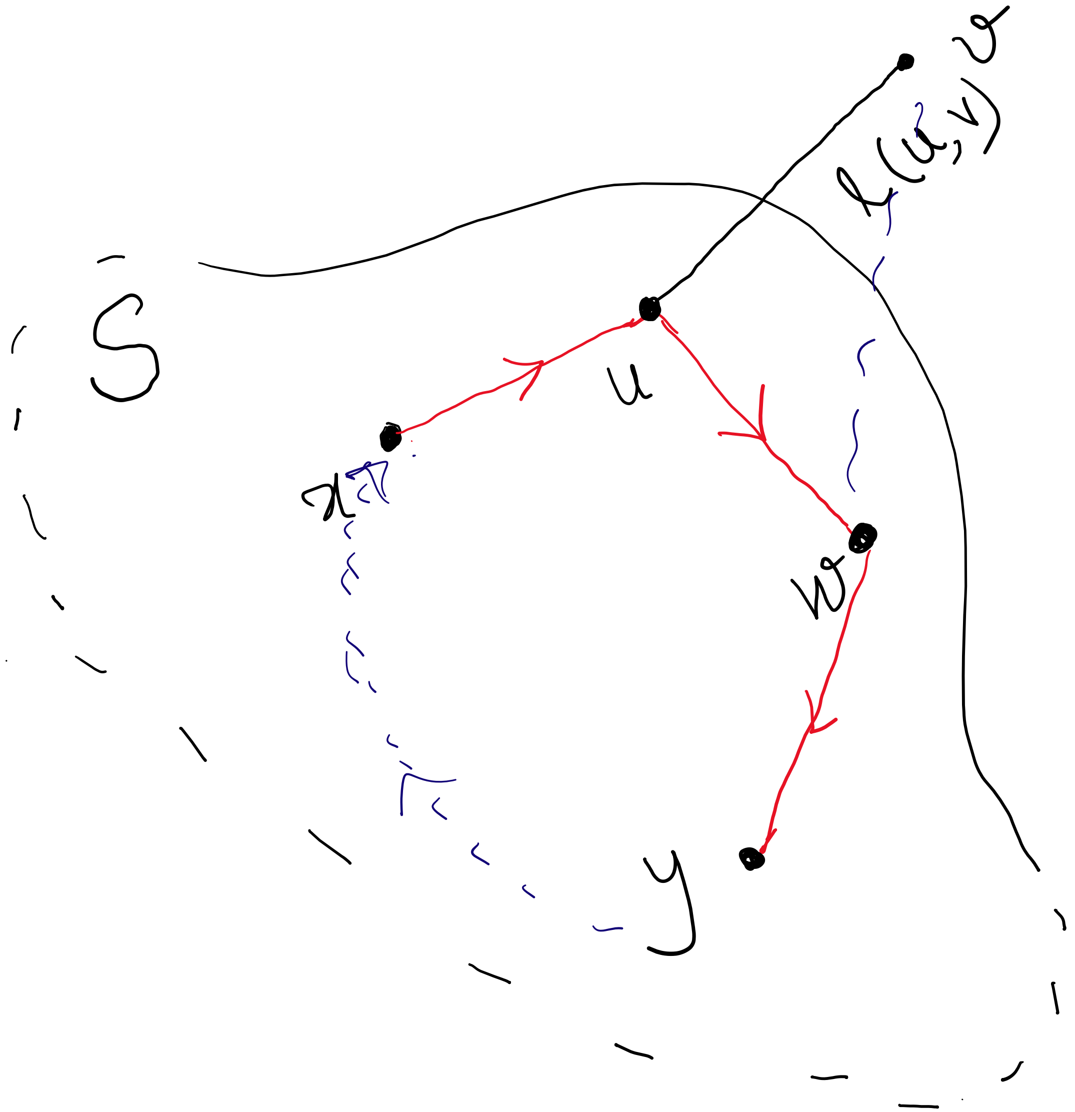
- Start with a least cost edge $\ell(i,j)$, set $S \leftarrow \{i,j\}$ and let the tour be $i \to j \to i$
- while $|S| < n$
  - Let $u \in S$, $v \notin S$ be such that $\ell(u,v)$ is the least cost edge among all the edges crossing the cut $(S, V \setminus S)$
  - Modify the tour to visit v and add v to S

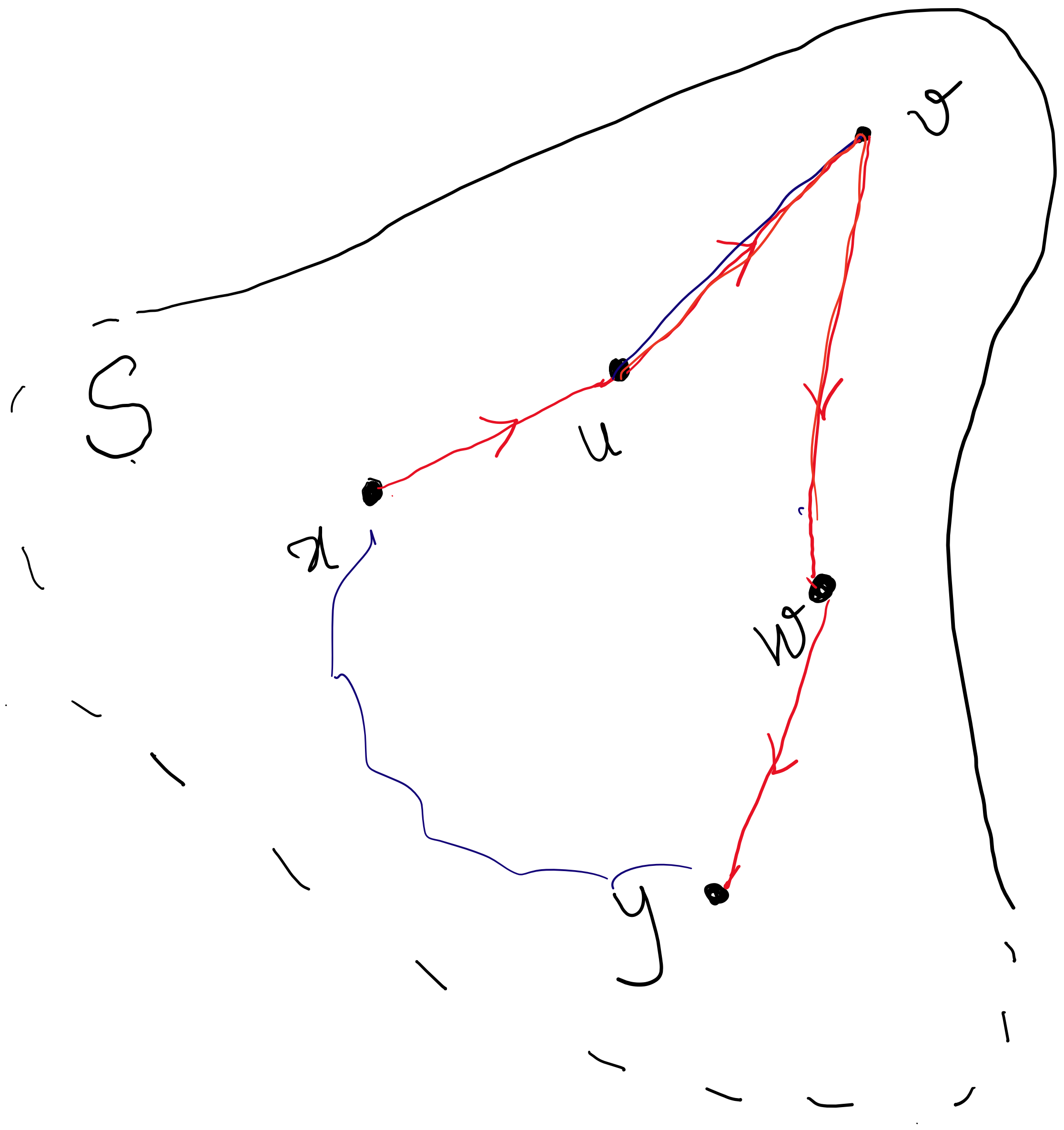
?



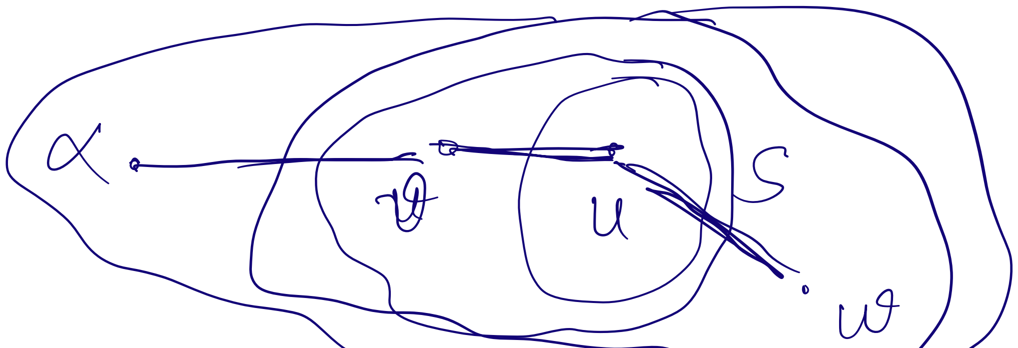


cities that are already part of tour so far

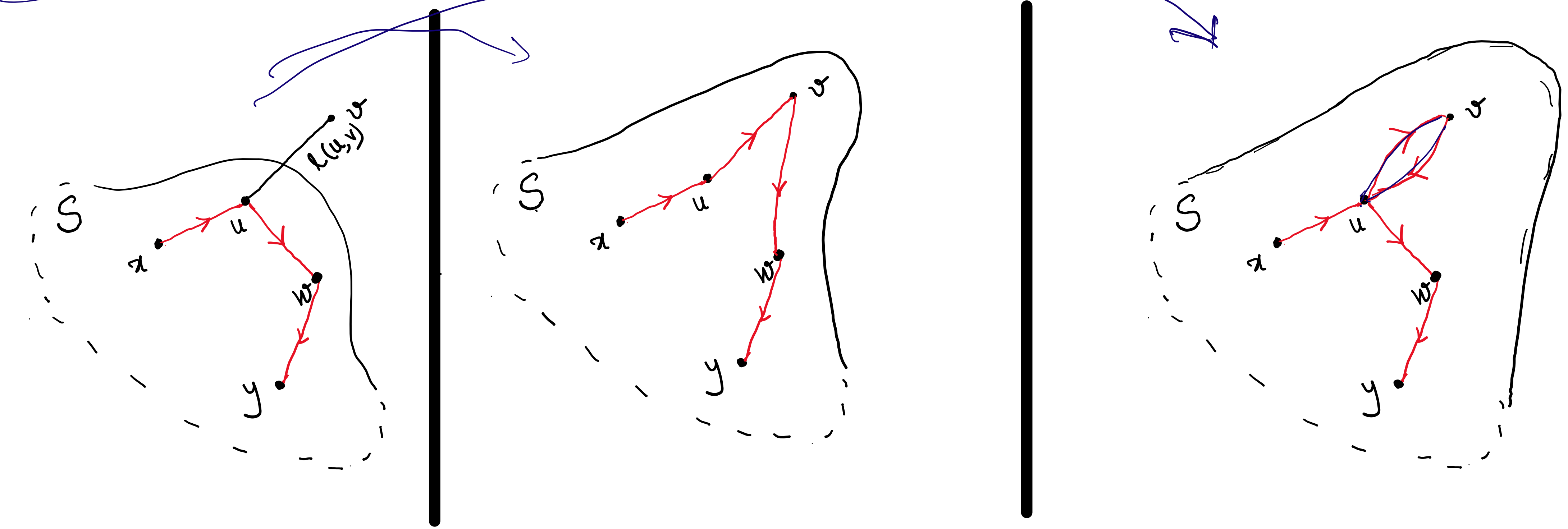

S
u
v
x
w
y
ℓ(u,v)

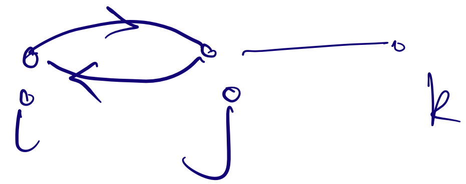

S
u
v
x
w
y

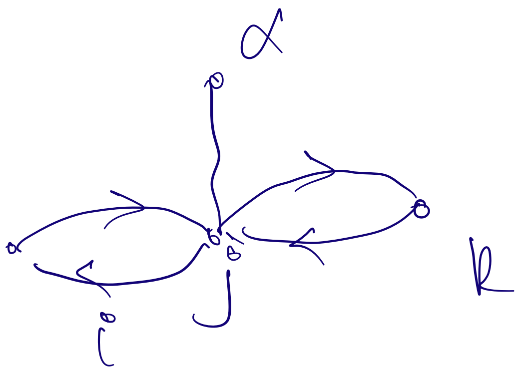
Why is this a **2**-approximation algorithm?

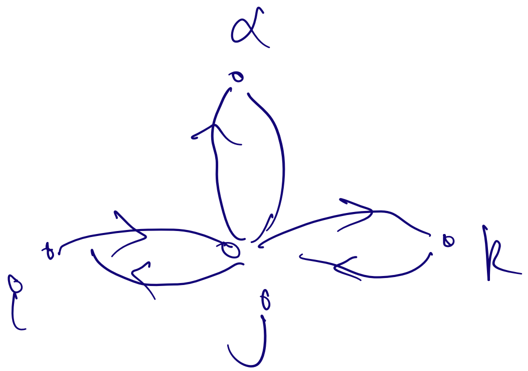


- ❑ Selecting the cut-edges in each iteration is identical to that of Prim's algorithm
- ❑ Consider a different walk whose total cost is an upper bound on the cost of the tour output by the algorithm

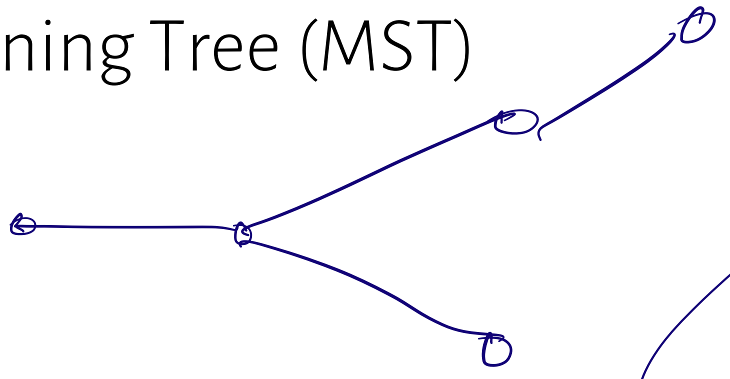


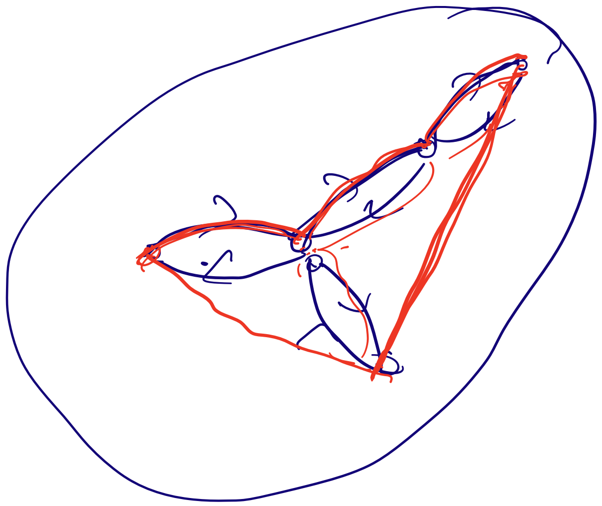

i
j
k


α
i
j
k

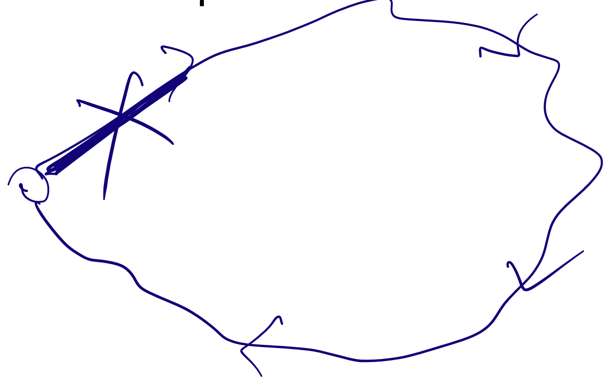

α
i
j
k

- [ ] Cost of new walk = $\mathbf{2} \cdot$ cost of Min Spanning Tree (MST)





- [ ] Cost of any MST $\leq$ Cost of optimal TSP tour = OPT



- [ ] Cost of tour output by greedy $\leq$ Cost of new walk

$$= 2 \cdot \text{Cost of MST} \leq 2 \cdot \text{OPT}$$

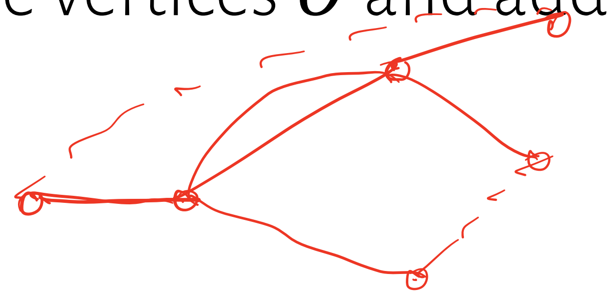
- Improving the approximation factor: A perspective change

- What is the algorithm doing?

  - Compute an MST of the graph
  - Replace each edge in MST with two copies of itself
  - Find a Eulerian tour on this graph
  - Use shortcuts to avoid revisiting vertices

- Can we modify the MST to be Eulerian without doubling its cost?

- Main problem: Odd degree vertices in the MST T
- We want all vertices to have even degree to have a Eulerian tour
- But there are an even number of odd degree vertices in any graph
- Idea: Find a perfect matching among odd degree vertices O and add the edges of the matching to T



  - Pick the min cost matching M

- Total cost of a Eulerian tour is $\overset{=}{\leq}$ Cost of M + Cost of T $\leq$ Cost of M + OPT

$T \cup M$ is Eulerian

$$\leq OPT + \frac{OPT}{2}$$

$$= \frac{3}{2} OPT$$

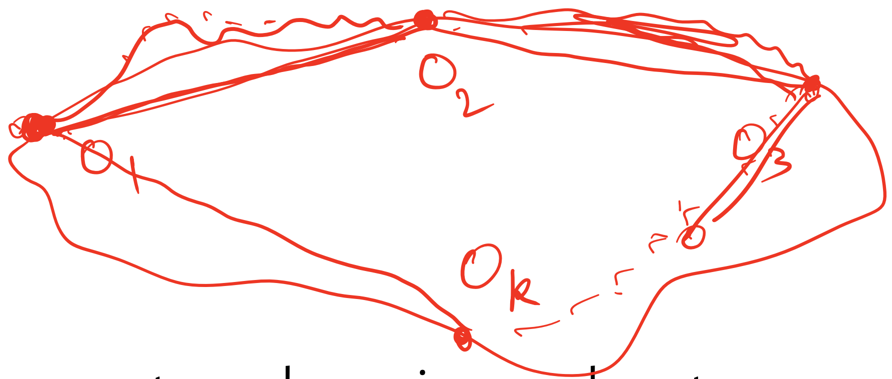
- **Claim**: Cost of $M \leq \mathrm{OPT}/2$

optimal tour

- Consider a TSP restricted to the odd degree vertices O

tour

- The cost of such a TSP is at most OPT



→ TSP optimal tour over all vertices

- If we pick alternate edges in such a tour, we get a perfect matching of O
- One of the two matchings has cost at most $\mathrm{OPT}/2$

$\Rightarrow \exists$ a perfect matching on O with cost $\leq \mathrm{OPT}/2$

- What about general TSP?

  ↳ no Δ^{le} inequality

- Even a $O(2^n)$-factor approximation algorithm will imply P = NP
- Proof?

  Reading Exercise !