CSCI 3100

More approximation: using geometry

Announcements

- Oral grading on Friday
- Next HW due ~~Wednesday~~ after break

Traveling Salesman (TSP)

Given n cities with pairwise distances between them, find the shortest cycle visiting all cities.

This is NP-Hard:

Reduce Ham. ~~path~~ cycle to TSP!

An unweighted graph, goal is to see if there is a simple cycle visiting all vertices.

Build G': same vertices

Add all edges possible:

$$\begin{cases} w(e) = 1 \text{ in } G' & \text{if } e \in G \\ w(e) = 2 \text{ in } G' & \text{if } e \notin G \end{cases}$$

Set $k = n$.

Note: Nothing special about 1 & 2 here!

In fact, I can use different values + show even approximating TSP is hard:

Ex: Let $G' = \begin{cases} w(e) = 1 & \text{if } e \in G \\ w(e) = n+1 & \text{if } e \notin G \end{cases}$

Here: Still have

G has Ham. cycle

$\Longleftrightarrow$ G' has TSP tour of length n

But: If no Hamiltonian cycle, G' only has tours of length $\geq 2n$

So a 2-approx alg for TSP would give exact soln to Ham. cycle.

Thm: For any polynomial $f(n)$, there is no $f(n)$-approx algorithm for TSP (unless $P = NP$).

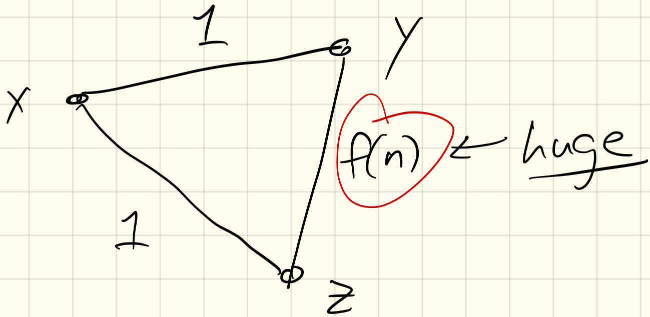
pf: Build G':

$w(e) = 1$ if $e \in G$

$w(e) = p(n)$ if $e \notin G$

However:

These are strange G' graphs:



If we have extra structure, can still approximate!

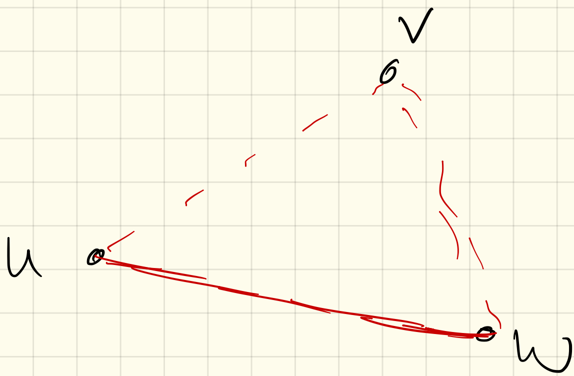
What are some common sources of graphs we might want to solve?

Roads!

Dfn: Triangle inequality:

For any $u, v, \& w \in V$,

$$\ell(u,w) \leq \ell(u,v) + \ell(v,w)$$



Note: Always get this for certain graphs:

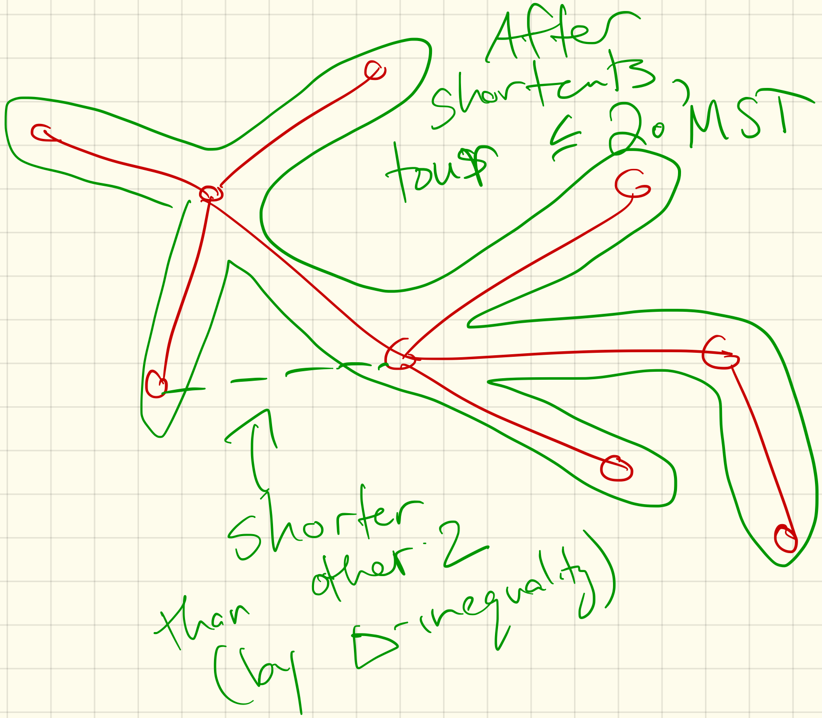
geometric graphs

$\hookrightarrow$ embedded in $\mathbb{R}^2$

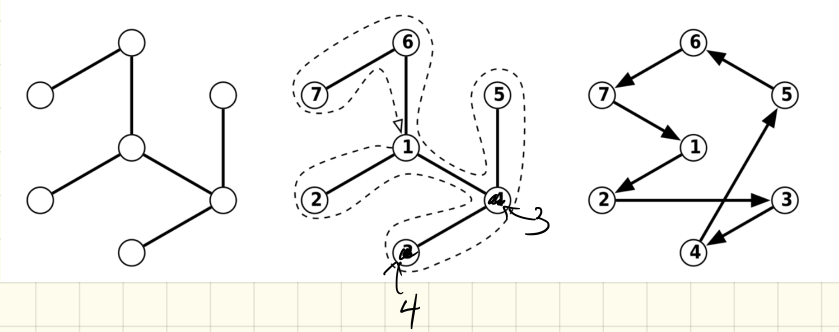
Thm: If G satisfies the triangle inequality, can compute a 2-approx for TSP.

Idea:

- Start w/ MST



The picture:



Steps:

- Compute MST
- Get DFS ordering
- Do shortcuts to avoid repeated edges + vertices

Claim: This algorithm is a 2-approximation.

pf: Let OPT be cost of the best TSP tour.

Let MST be the total weight of the min. spanning tree.

& Our algorithm's TSP length $= A$.

Bound A:

1. $A \leq 2\,\text{MST}$

(see prior slide & use $\triangle$-ineq.)

On the other hand,
OPT (best tour) is a cycle.
If you delete any edge from OPT, what do you have?
path!

$\Rightarrow$ Since any path is also a potential min. spanning tree,

(2) $MST \le OPT$

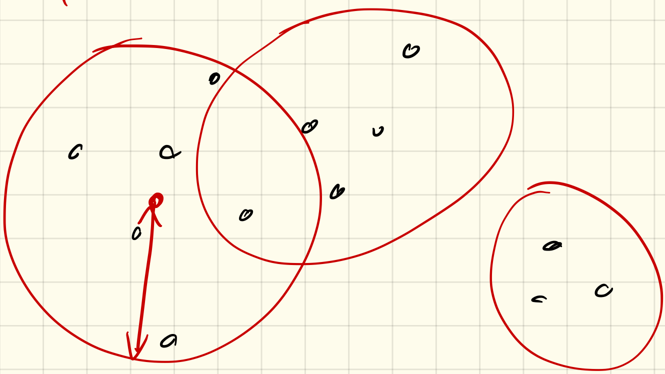
$$A \le 2\,MST \le 2\,OPT$$

Resut: we get a 2-approx.

Another: Clustering

Given a set of points $P = \{p_1, p_2, \ldots, p_n\}$ in $\mathbb{R}^2$ and an integer k, find a set of k circles that contain all n points, s.t. radius of largest circle is as small as possible.

$k=3$



Formally, find $C = \{c_1, \ldots, c_k\}$ of centers, s.t.

$$\text{cost} = \max_i \min_j |p_i c_j|$$

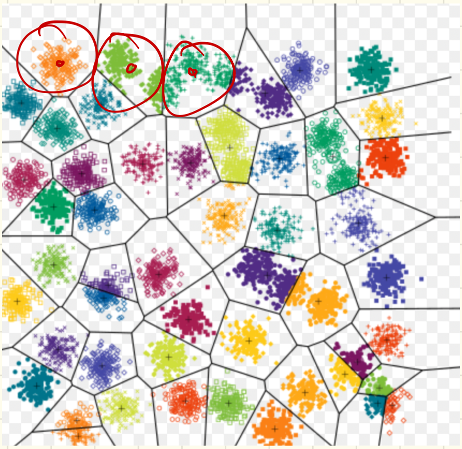
is minimized.

Why?

Assign each point to closest center!

(This is the "circle".)

Radius of circle is the max distance of center to point assigned to it

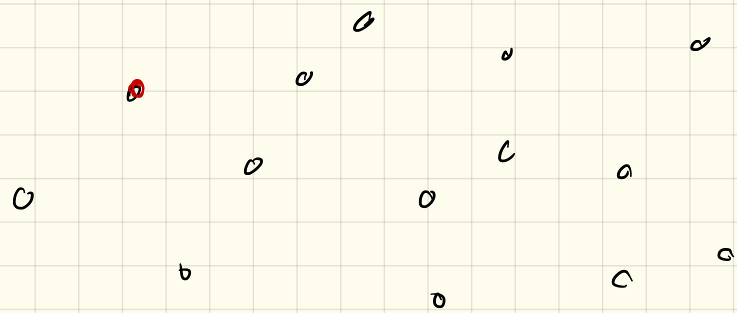


This problem is NP-Hard,
- even to approximate with a factor of ~1.8.

However, simple + natural <u>greedy</u> strategy (Gonzalez '85), which gives a 2-approx.

<u>Idea:</u>

- Suppose I start w/ a point I've chosen to be a center.



What's a natural next point to grab?

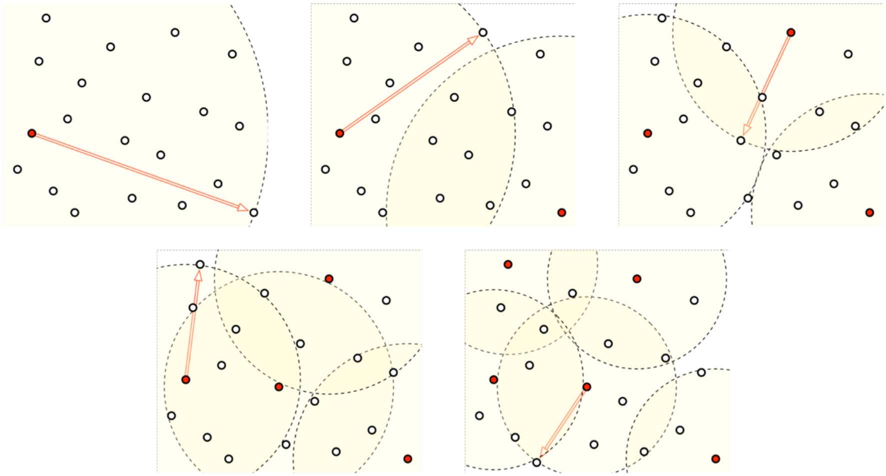
Furthest point → make this a center

Algorithm

GonzalezKCenter(P, k):

for $i \leftarrow 1$ to n
  $d_i \leftarrow \infty$ ← distance to a center
$c_1 \leftarrow p_1$
for $j \leftarrow 1$ to k
  $r_j \leftarrow 0$
  for $i \leftarrow 1$ to n
    $d_i \leftarrow \min\{d_i,\ |p_i c_j|\}$
    if $r_j < d_i$
      $r_j \leftarrow d_i;\ \ c_{j+1} \leftarrow p_i$
return $\{c_1, c_2, \ldots, c_k\}$

Runtime: $O(nk)$



The first five iterations of Gonzalez's k-center clustering algorithm.

Thm: Gonzales k-center gives a 2-approx.

Pf: Let OPT be optimal k-center solution. largest radius in best set of k circles

$\forall i$, let c_i + r_i be the i^{th} centers added by our algorithm.

c_1, | c_2, | c_3 | ... | c_k

(furthest point's distance)

r_1 | r_2 | r_3 ... | r_{k-1} | r_k

cost of A

c_j is at least r_{j-1} away from $c_1, c_2, \ldots, c_{j-1}$

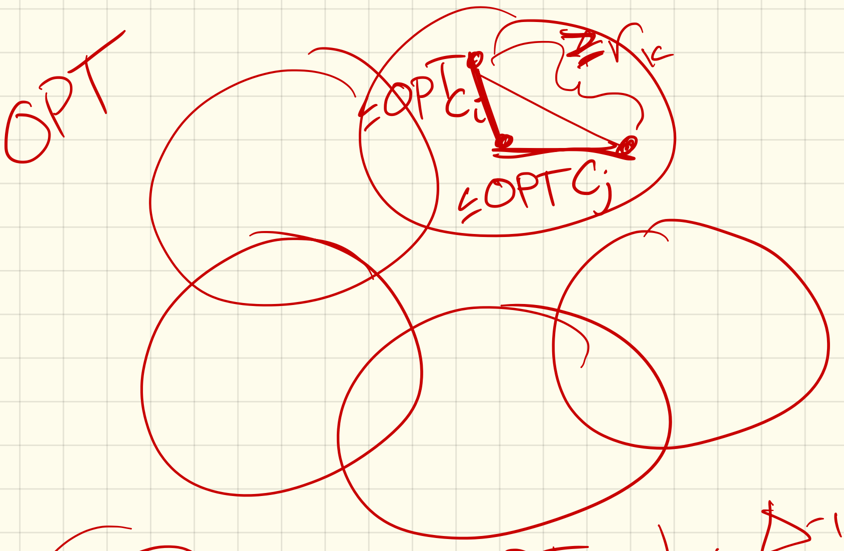
$$r_1 \geq r_2 \geq \cdots \geq r_{k-1}$$

$$\Rightarrow |c_i c_j| \geq r_k$$

Consider what c_{k+1} would have been (r_k away from one of $c_1, \dots c_k$)

$c_1, \dots, c_{k+1} \leftarrow k+1$ points (from P)

OPT soln covered these



$r_k \leq 2 \cdot OPT$ by Δ-ineq.

cost of my soln