Approximation Algorithms

Center Selection Problem

Set S of n sites in $\mathbb{R}^2$.

Want to output k centers C such that the maximum distance from a site to its nearest center is as small as possible.

Let $r(C)$ be the smallest radius such that the circles of radius $r(C)$ cover S. (covering radius).

Try a Greedy Algorithm

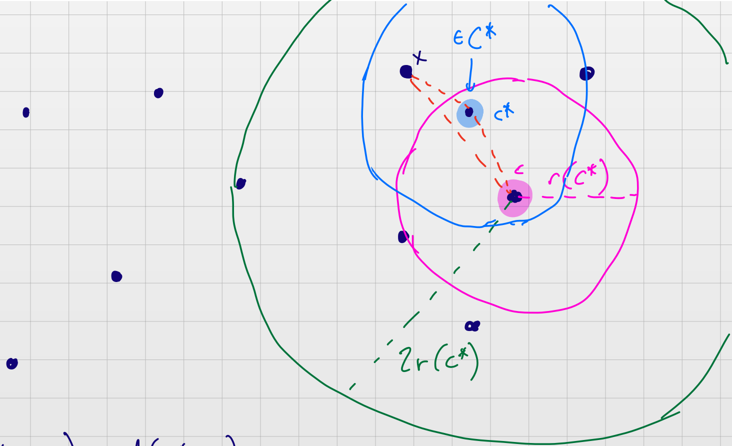
Pick centers to minimize the covering radius at each step.

covering radius of |x−y|
optimal is 0.
BAD approximation!

What if we know the optimal covering radius?

The exists a set C^* where every site is within $r(C^*)$ of some center.

Idea: Take a random site s. s is close to some center $c^* \in C^*$. Choose s as a center in C, and show that everything "close" to s must also be "close" to things close to c^*.



$$d(x,c) \le d(x,c^*) + d(c^*,c)$$

Greedy Algorithm Idea

$C = \emptyset$
$\mathcal{U} = S$ // uncovered sites
while $\mathcal{U} \neq \emptyset$:
  Choose c from $\mathcal{U}$.
  Remove all sites x w/ $d(x, c) \leq 2r(C^*)$ from $\mathcal{U}$.
return C.

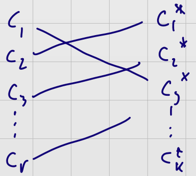
Does C have covering radius $r(C) \leq 2r(C^*)$?
Yes

Is $|C| \leq k$?

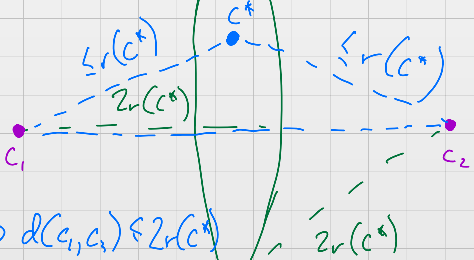
WTS $|C| \le k$. Then we'll have a 2-approximation.

Take $c \in C$. We know there is some $c^* \in C^*$, with $d(c, c^*) \le r(C^*)$

If each c's close to a unique c^*, then $|C| \le k$, because $|C^*|$ is k.



$\Rightarrow r \le k$



Δ inequality $\Rightarrow d(c_1, c_2) \leq 2r(c^*)$

$\Rightarrow$ there is no c^* s.t.

$d(c^*, c_1) \leq r(c^*)$

and $d(c^*, c_2) \leq r(c^*)$

Getting Around Not Knowing $r(C^*)$

Idea: always pick the furthest center from C.

```
C = ∅
U = S        // unpicked sites
while |C| < k,
      Choose s ∈ U furthest from C.

return C.
```

Suppose some site s is more than $2r(C^*)$ from every $c \in C$.

Let's look at the ith iteration of the loop, and C_i the C at that point.
We know that $d(s, C_i) \geq d(s, C) > 2r(C^*)$
$\Rightarrow$ whatever center we picked is also $> 2r(C^*)$ away.

So this is a valid execution of the algorithm that knows $r(C^*)$
$\Rightarrow$ the algo also returns a 2-approximation.

Weighted Vertex Cover

Input: graph $G=(V,E)$, $\{w_x \mid x \in V\}$.

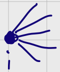
Output: minimal weight set $S \subseteq V$ such that every $e \in E$ is adjacent to some $s \in S$.

Pricing Algorithm

Associate prize p_e to each edge $e \in E$. "Cost of VC distributed among the edges"

Say prizes p_e are fair if $\sum_{e=(i,j)} p_e \leq w_i$.

"No edge is overpaying for its covering vertex"



If S^* is a vertex cover, and p_e is fair,
then $\sum_{e \in E} p_e \leq w(S^*)$

If S^* is a vertex cover, and p_e is fair,
then $\sum_{e \in E} p_e \leq w(S^*)$

Know that $\sum_{e=(i,j)} p_e \leq w_i$ for all nodes $i \in S^*$.

$$\Rightarrow \sum_{i \in S^*} \sum_{e=(i,j)} p_e \leq w(S^*)$$

b/c S^* is a vertex cover,
every edge appears in this sum

$$\Rightarrow \sum_{e \in E} p_e \leq \sum_{i \in S^*} \sum_{e=(i,j)} p_e$$