
Local Search Heuristics for Facility Location Problems

Rohit Khandekar

Dept. of Computer Science and Engineering,
Indian Institute of Technology Delhi

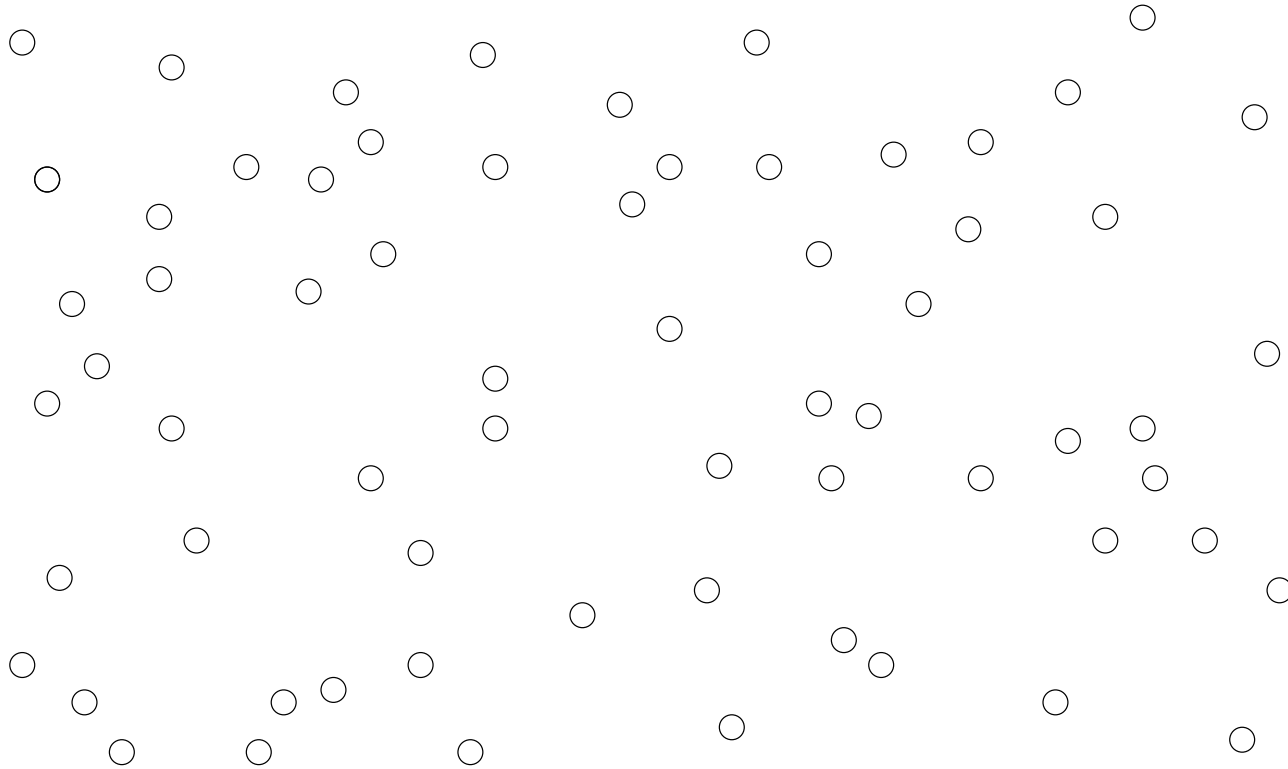
Joint work with: Vijay Arya, Naveen Garg, Vinayaka Pandit
Kamesh Munagala, Adam Meyerson

Outline

- ▶ Define the k -median problem
- ▶ Simple local search algorithm
- ▶ Analysis
- ▶ Generalization

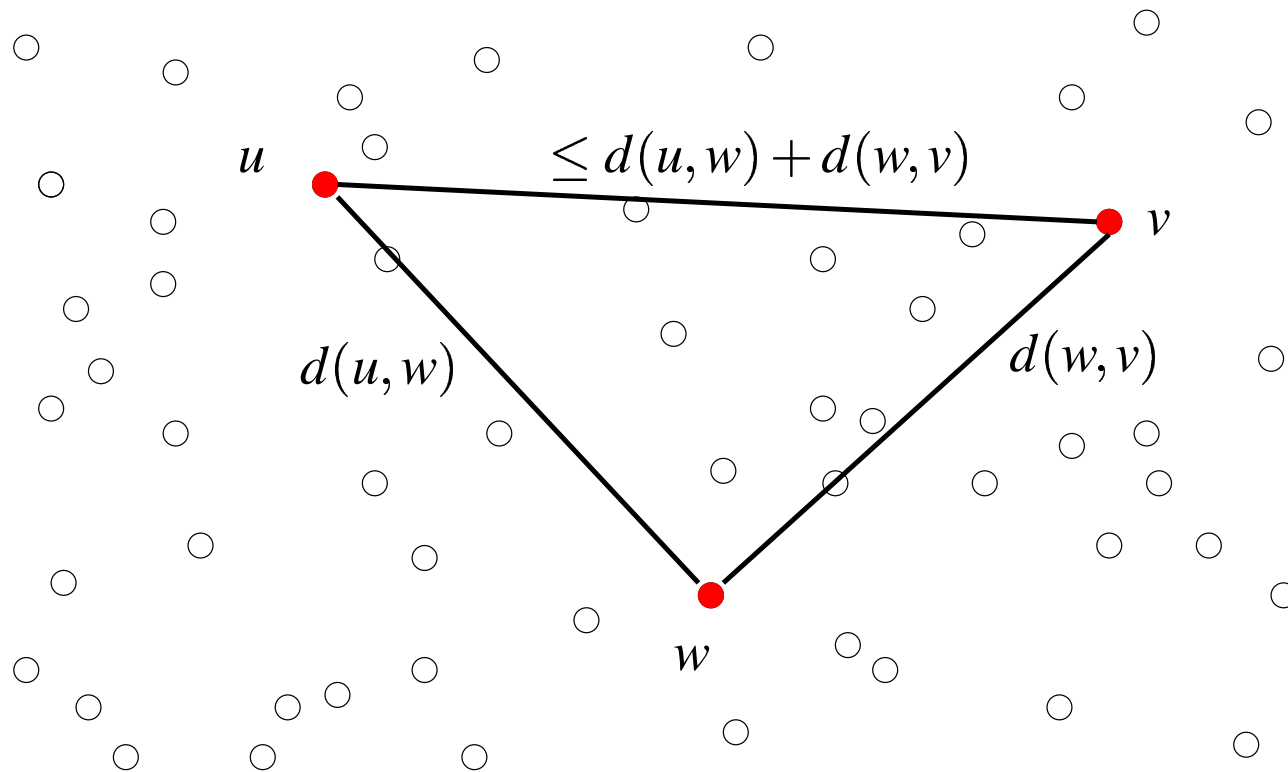
The k -median problem

We are given n points in a *metric space*.



The k -median problem

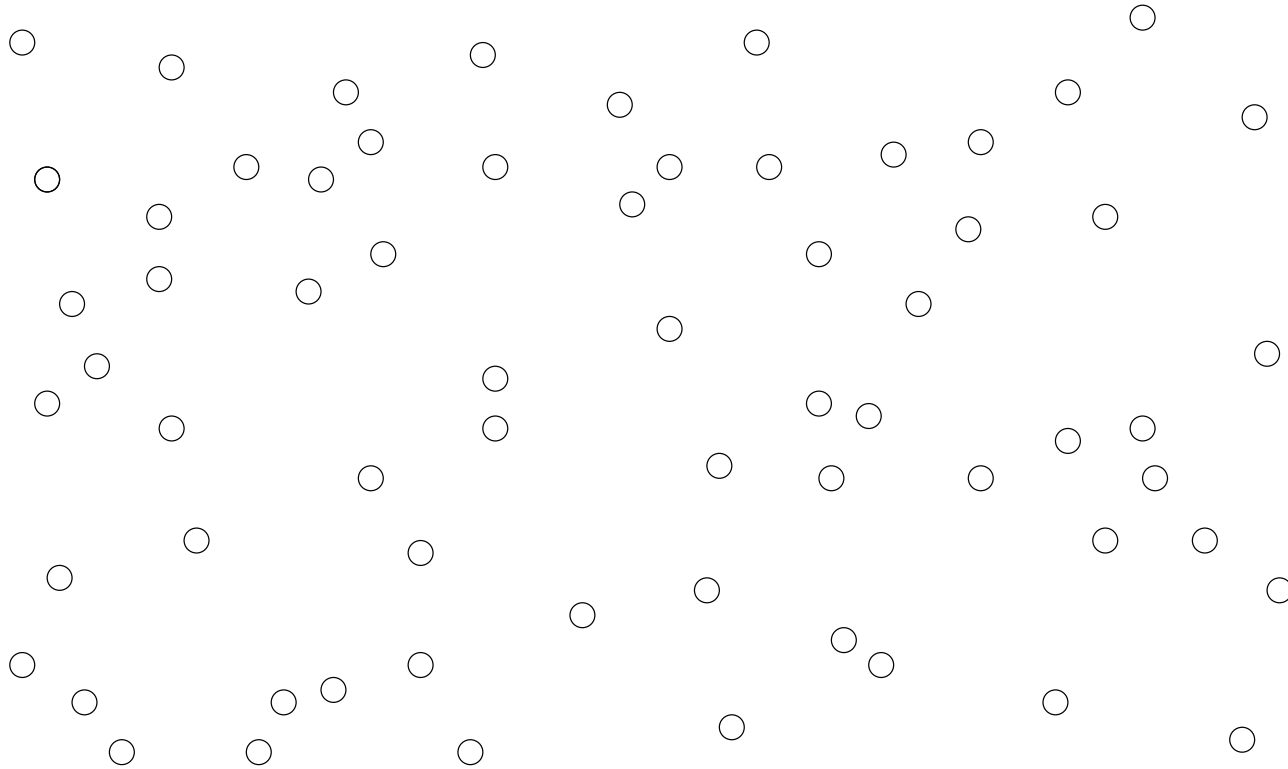
We are given n points in a *metric space*.



$$d(u, v) \geq 0, \quad d(u, u) = 0, \quad d(u, v) = d(v, u)$$

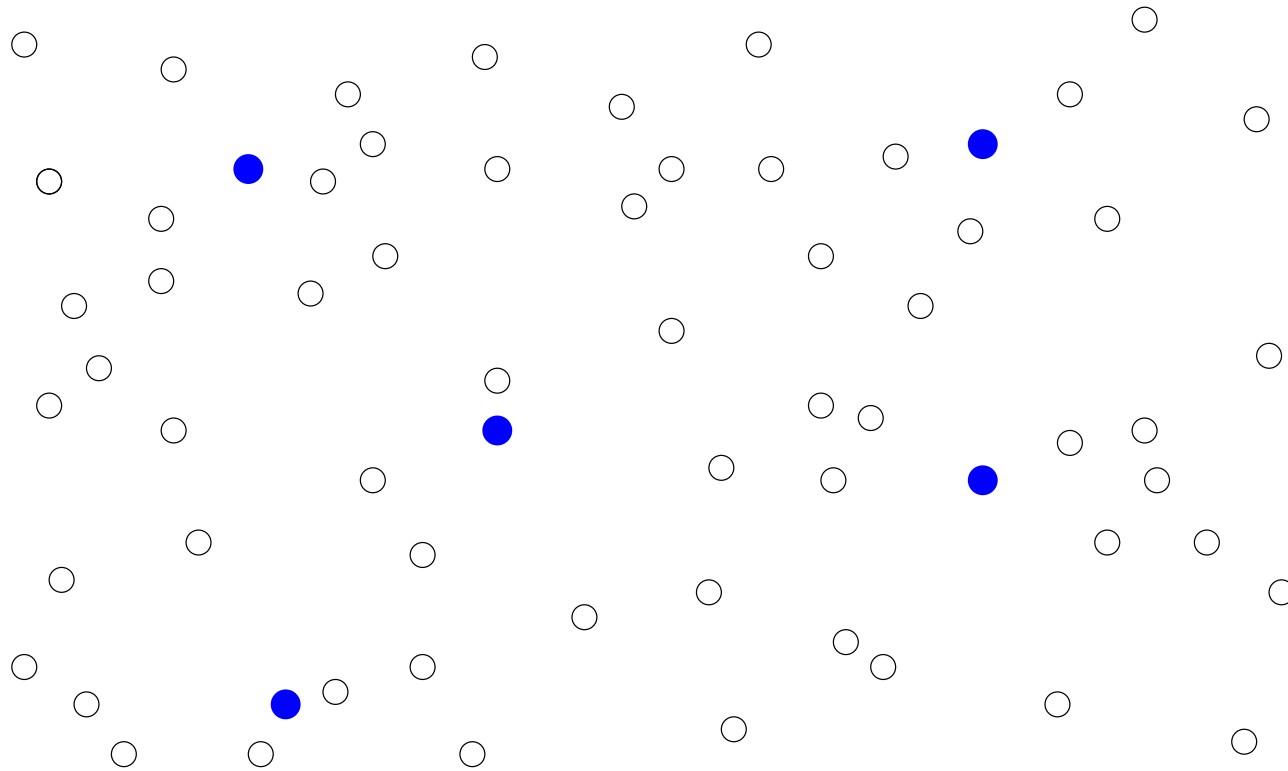
The k -median problem

We are given n points in a *metric space*.



The k -median problem

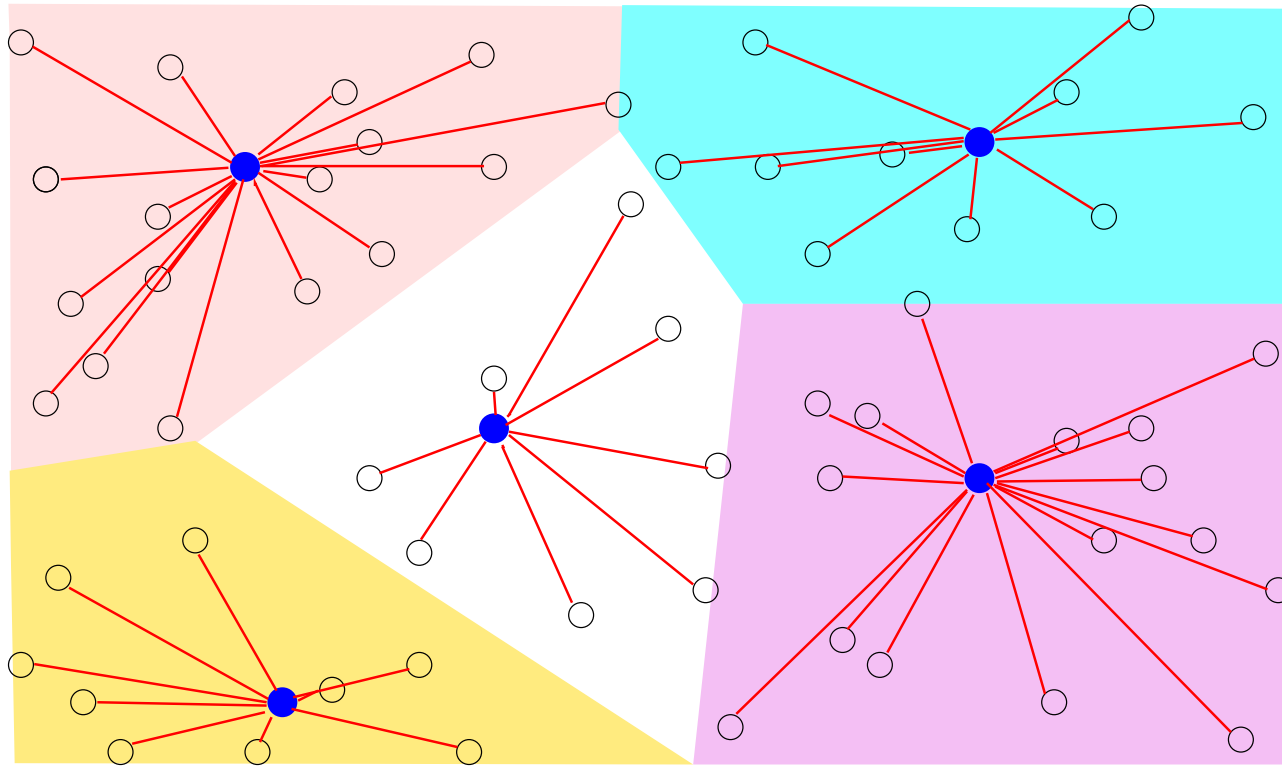
We are given n points in a *metric space*.



We want to identify k “medians” such that the sum of distances of all the points to their nearest medians is minimized.

The k -median problem

We are given n points in a *metric space*.



We want to identify k “medians” such that the sum of lengths of all the red segments is minimized.

A brief bio-sketch of the k -median problem

- ▶ NP-hard

A brief bio-sketch of the k -median problem

- ▶ NP-hard
- ▶ Known to OR community since 60's.

A brief bio-sketch of the k -median problem

- ▶ NP-hard
- ▶ Known to OR community since 60's.
- ▶ Used for locating warehouses, manufacturing plants, etc.

A brief bio-sketch of the k -median problem

- ▶ NP-hard
- ▶ Known to OR community since 60's.
- ▶ Used for locating warehouses, manufacturing plants, etc.
- ▶ Used also for clustering, data mining.

A brief bio-sketch of the k -median problem

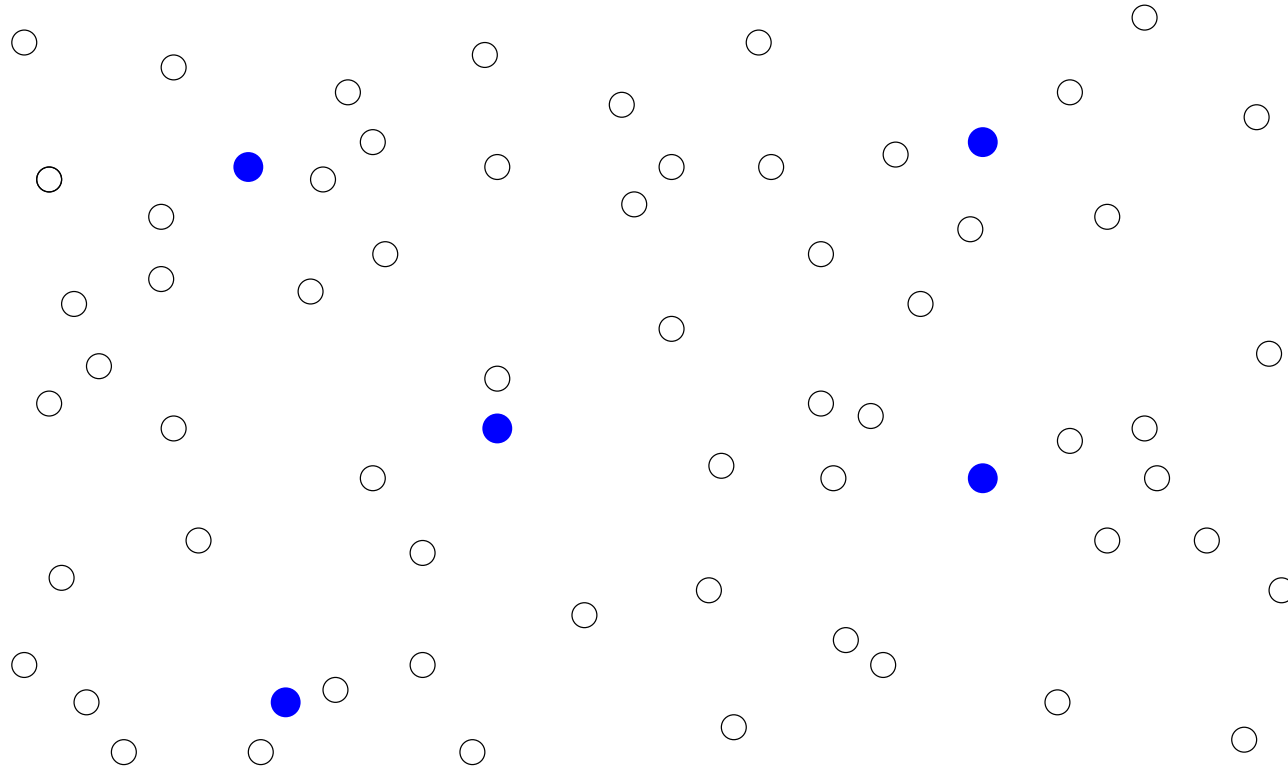
- ▶ NP-hard
- ▶ Known to OR community since 60's.
- ▶ Used for locating warehouses, manufacturing plants, etc.
- ▶ Used also for clustering, data mining.
- ▶ Received the attention of the Approximation algorithms community in early 90's.

A brief bio-sketch of the k -median problem

- ▶ NP-hard
- ▶ Known to OR community since 60's.
- ▶ Used for locating warehouses, manufacturing plants, etc.
- ▶ Used also for clustering, data mining.
- ▶ Received the attention of the Approximation algorithms community in early 90's.
- ▶ Various algorithms via LP-relaxation, primal-dual scheme, etc.

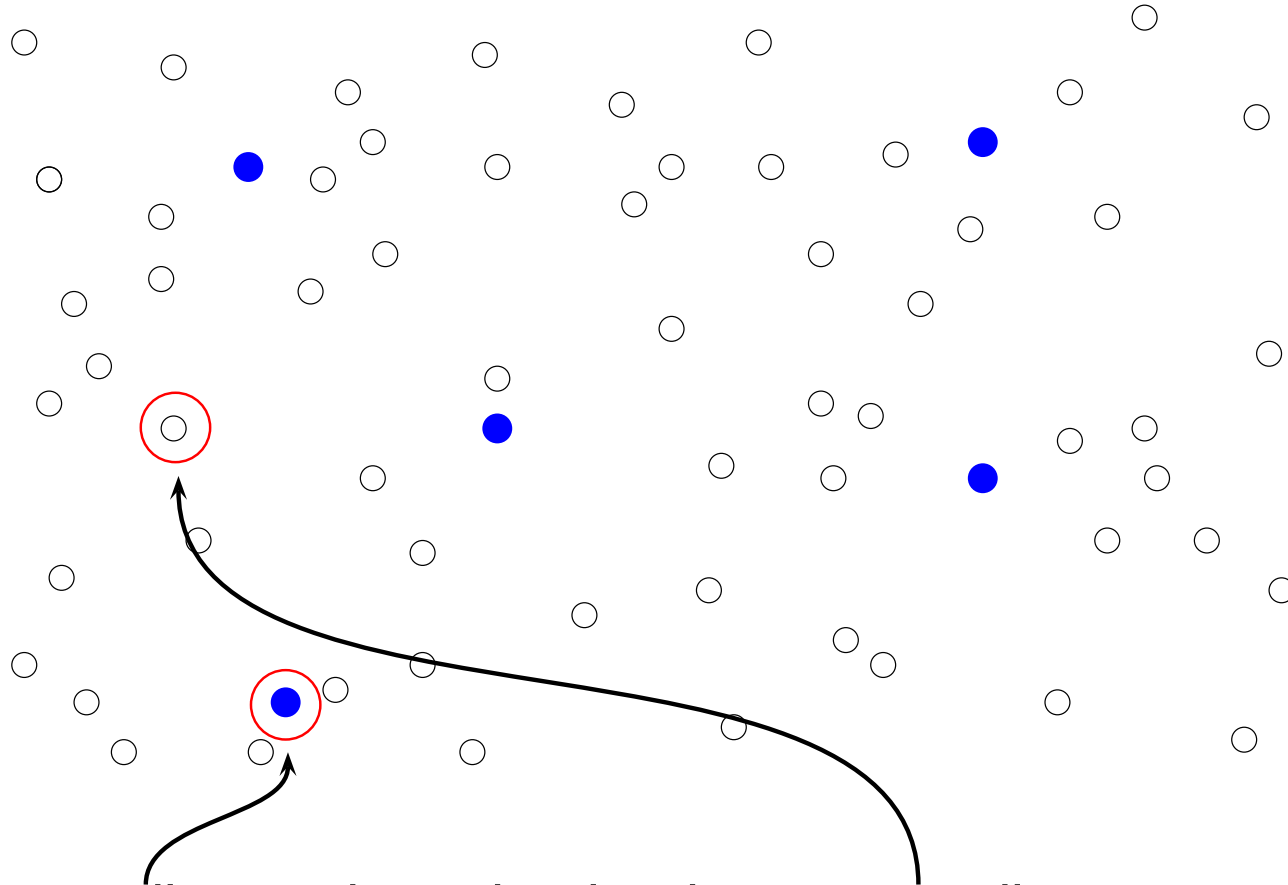
A local search algorithm

A local search algorithm



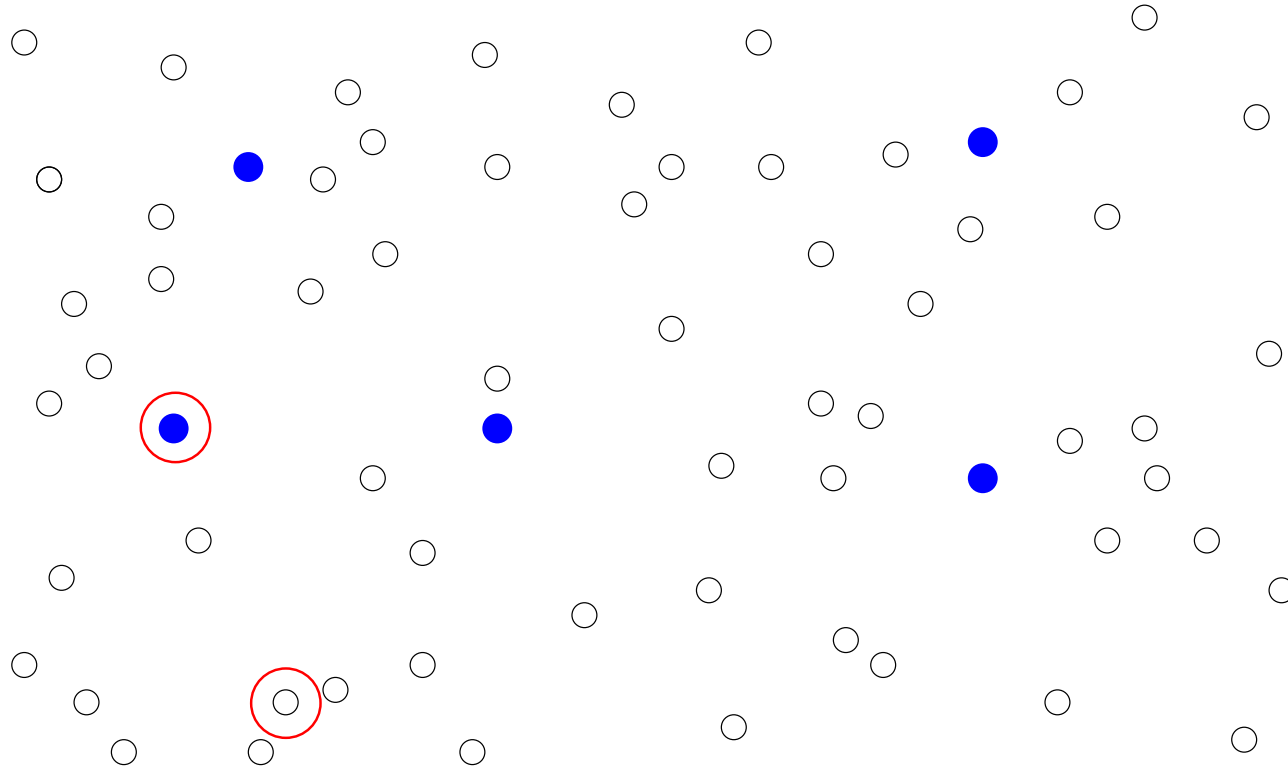
Start with **any** set of k medians.

A local search algorithm



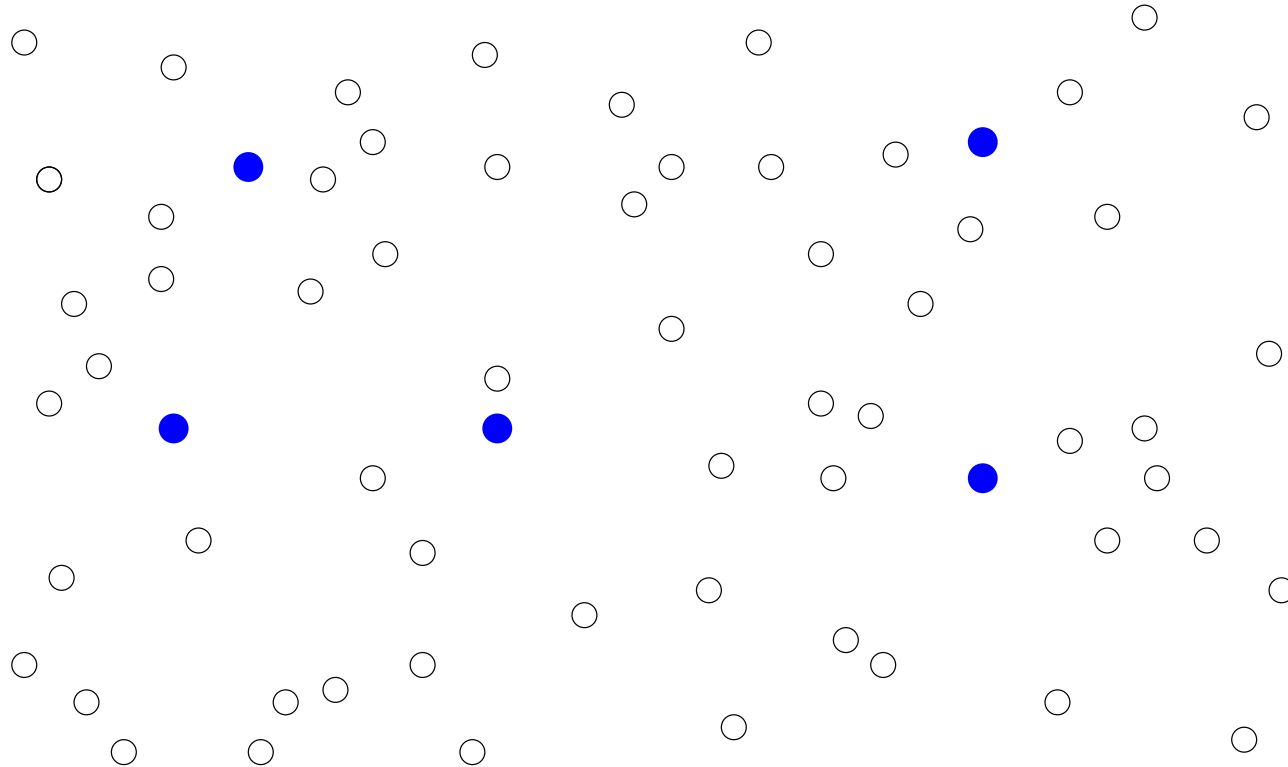
Identify a median and a point that is not a median.

A local search algorithm



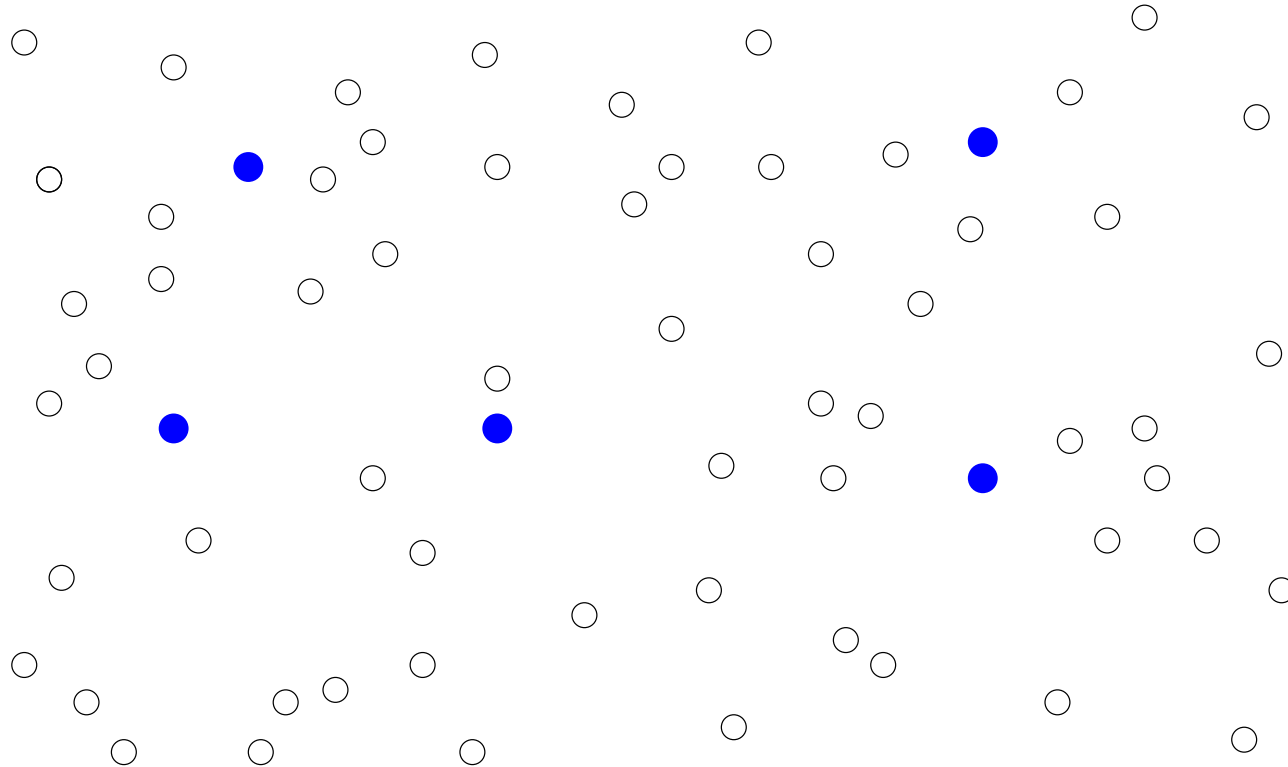
And SWAP tentatively!

A local search algorithm



Perform the swap, only if the new solution is “better” (has less cost) than the previous solution.

A local search algorithm



Perform the swap, only if the new solution is “better” (has less cost) than the previous solution.

Stop, if there is no swap that improves the solution.

The algorithm

Algorithm Local Search.

1. $S \leftarrow$ any k medians
2. While $\exists s \in S$ and $s' \notin S$ such that,
 $cost(S - s + s') \leq cost(S)$,
do $S \leftarrow S - s + s'$
3. return S

The algorithm

Algorithm Local Search.

1. $S \leftarrow$ any k medians
2. While $\exists s \in S$ and $s' \notin S$ such that,
 $cost(S - s + s') \leq (1 - \epsilon)cost(S)$,
do $S \leftarrow S - s + s'$
3. return S

Main theorem

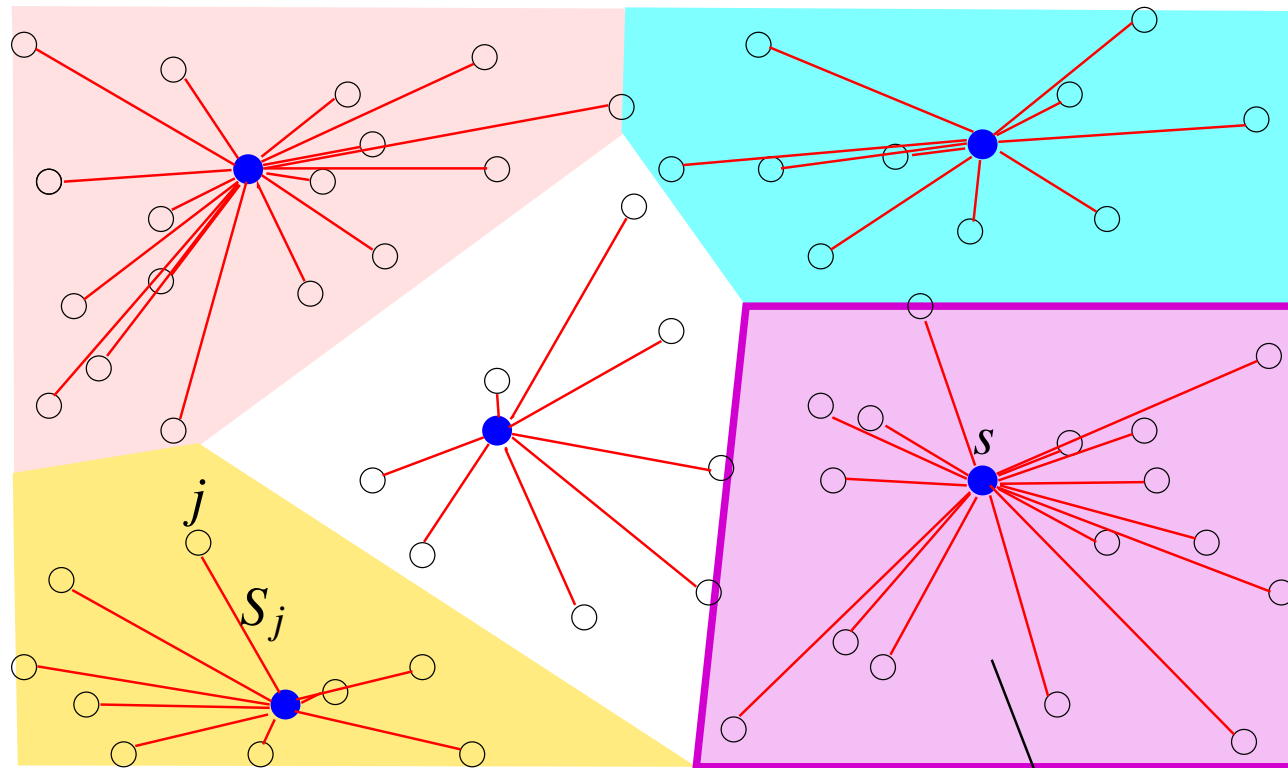
The local search algorithm described above computes a solution with cost (the sum of distances) at most 5 times the minimum cost.

Main theorem

The local search algorithm described above computes a solution with cost (the sum of distances) at most 5 times the minimum cost.

Korupolu, Plaxton, and Rajaraman (1998) analyzed a variant in which they permitted adding, deleting, and swapping medians and got $(3 + 5/\varepsilon)$ approximation by taking $k(1 + \varepsilon)$ medians.

Some notation



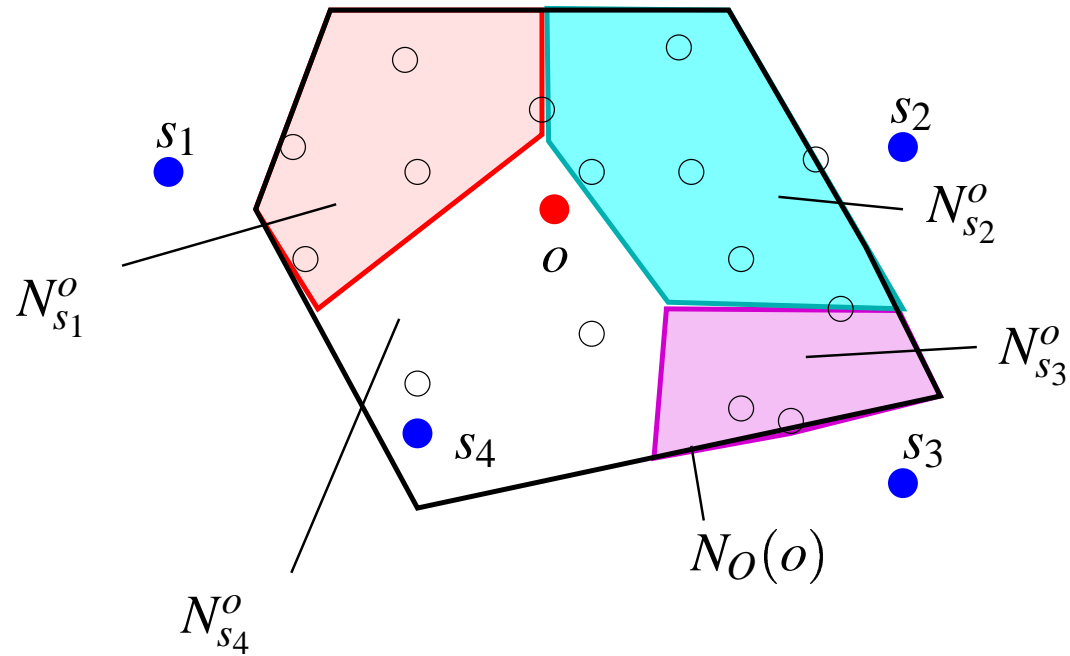
$$S = \{ \bullet \bullet \bullet \bullet \bullet \}$$

$$|S| = k$$

$$N_S(s)$$

$cost(S) =$ the sum of lengths of all the red segments

Some more notation



$$N_s^o = N_O(o) \cap N_S(s)$$

Local optimality of S

- ▶ Since S is a local optimum solution,

Local optimality of S

- ▶ Since S is a local optimum solution,
We have,

$$\text{cost}(S - s + o) \geq \text{cost}(S) \quad \text{for all } s \in S, o \in O.$$

Local optimality of S

- ▶ Since S is a local optimum solution,
We have,

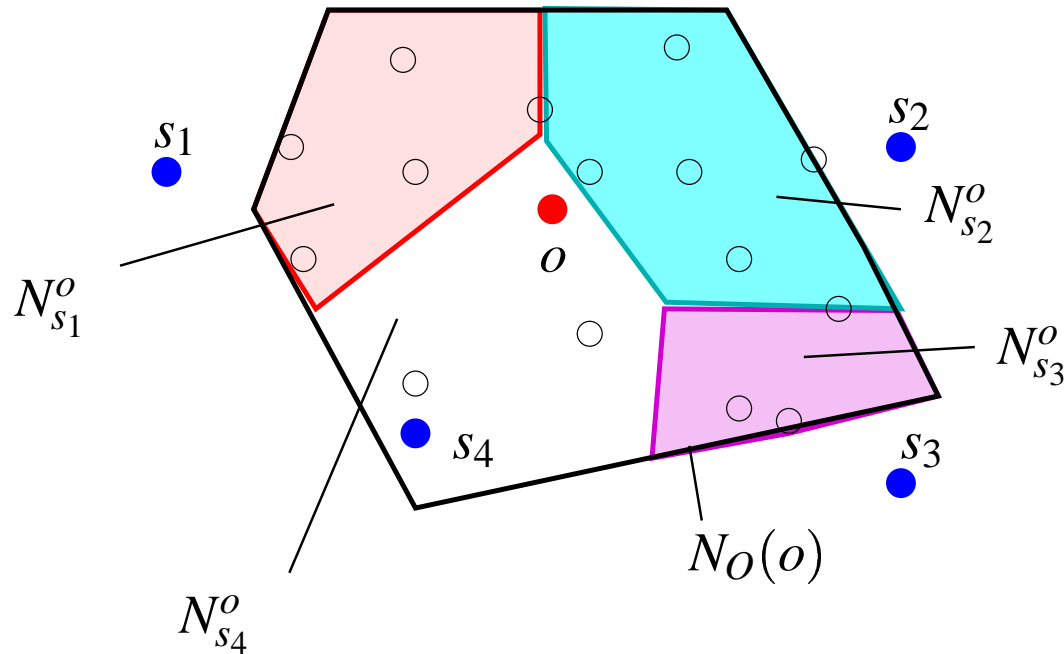
$$\text{cost}(S - s + o) \geq \text{cost}(S) \quad \text{for all } s \in S, o \in O.$$

- ▶ We shall add k of these inequalities (chosen carefully) to show that,

$$\text{cost}(S) \leq 5 \cdot \text{cost}(O)$$

> >

Capture

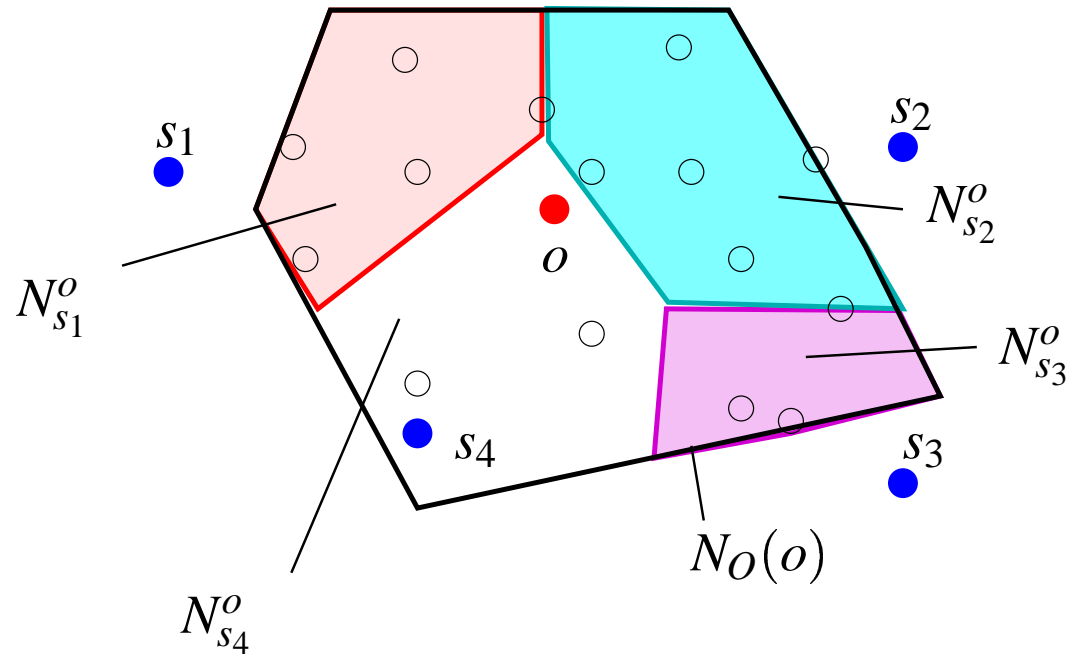


We say that $s \in S$ captures $o \in O$ if

$$|N_s^o| > \frac{|N_O(o)|}{2}.$$

Capture graph

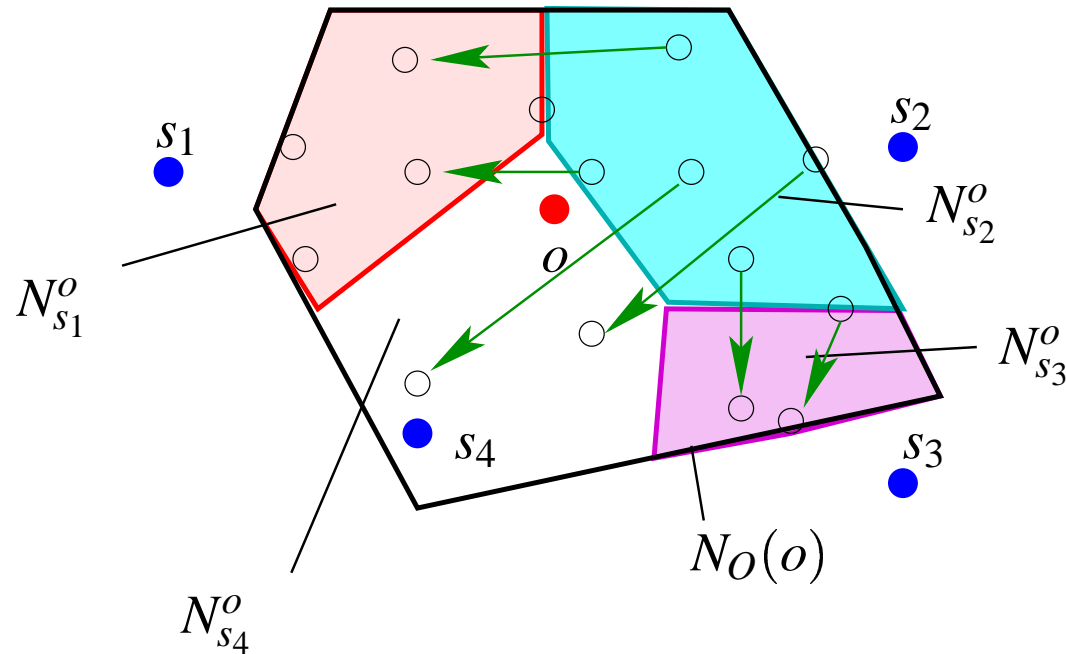
A mapping π



We consider a permutation $\pi : N_O(o) \rightarrow N_O(o)$ that satisfies the following property:

if s does not capture o then a point $j \in N_s^o$ should get mapped outside N_s^o .

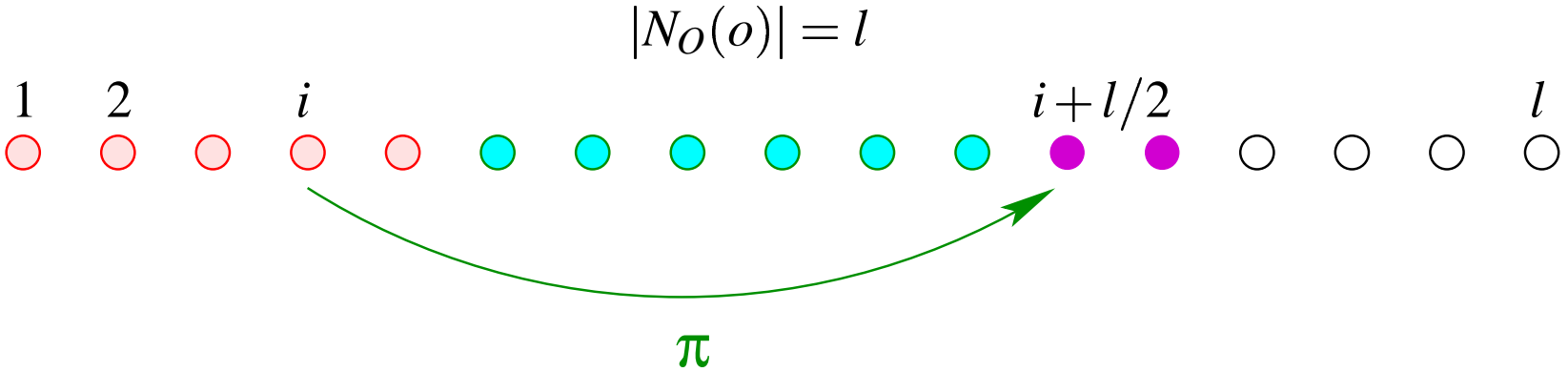
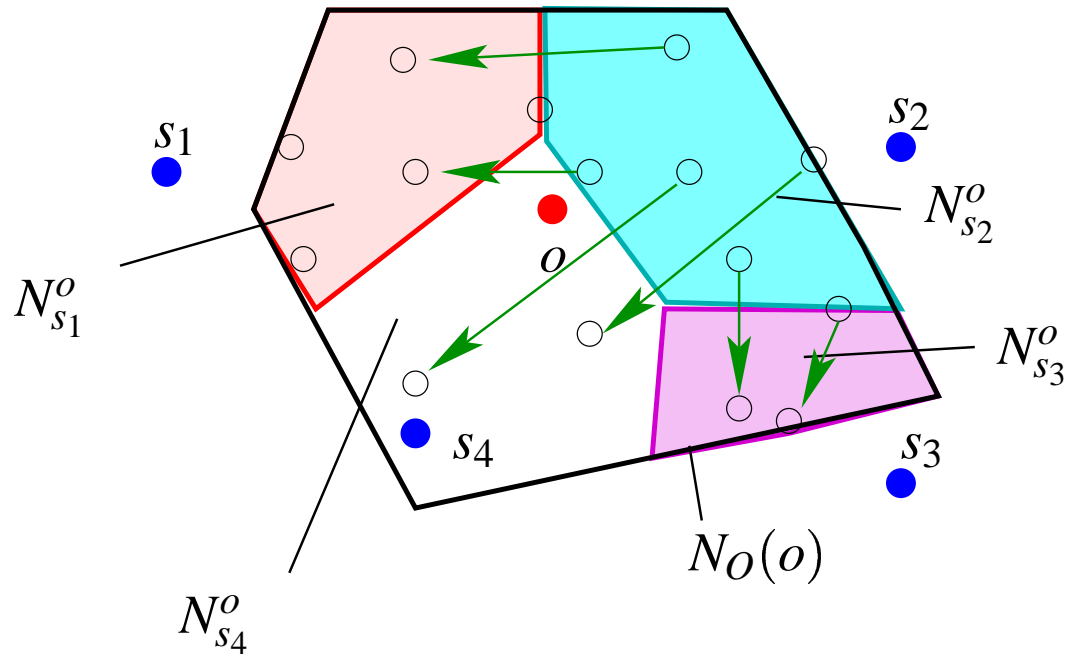
A mapping π



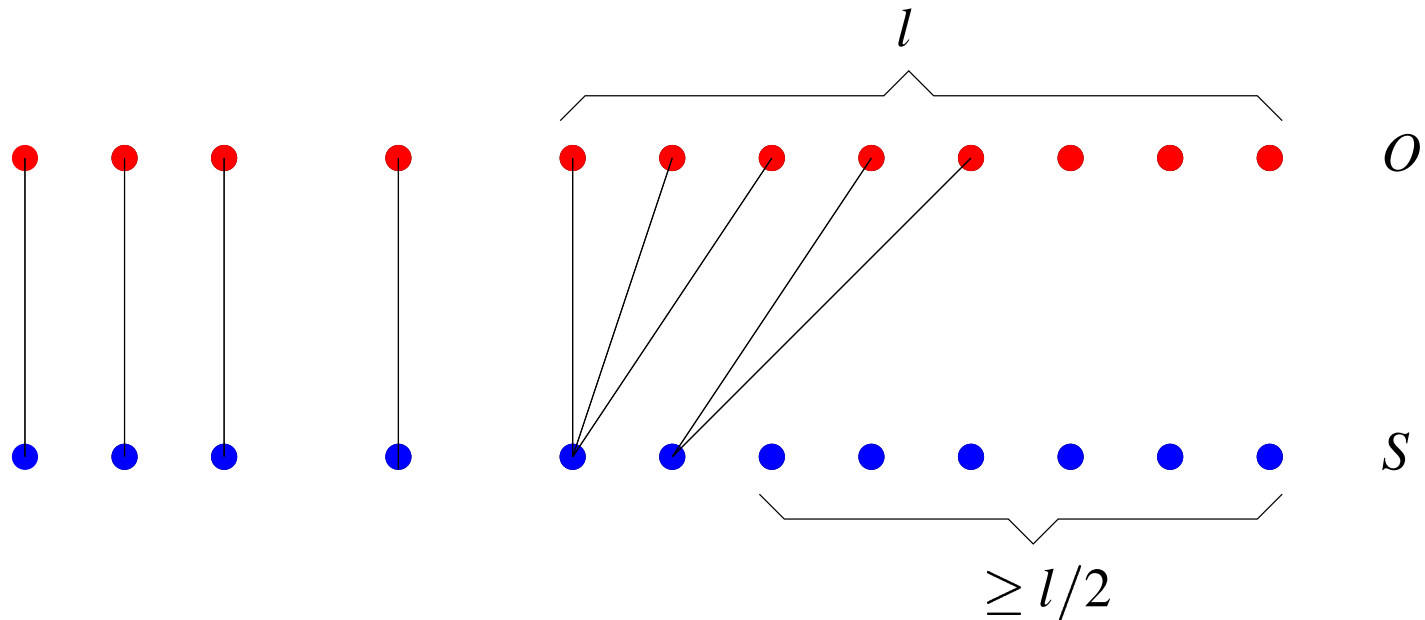
We consider a permutation $\pi : N_O(o) \rightarrow N_O(o)$ that satisfies the following property:

if s does not capture o then a point $j \in N_s^o$ should get mapped outside N_s^o .

A mapping π



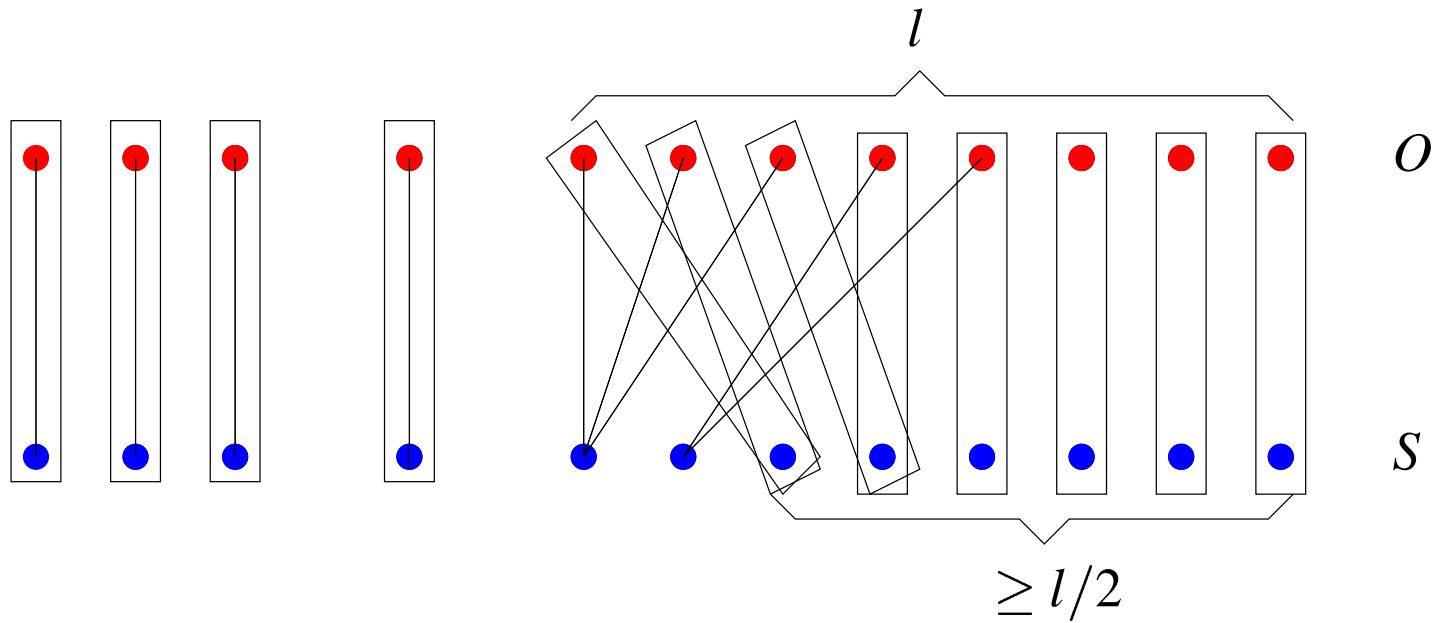
Capture graph



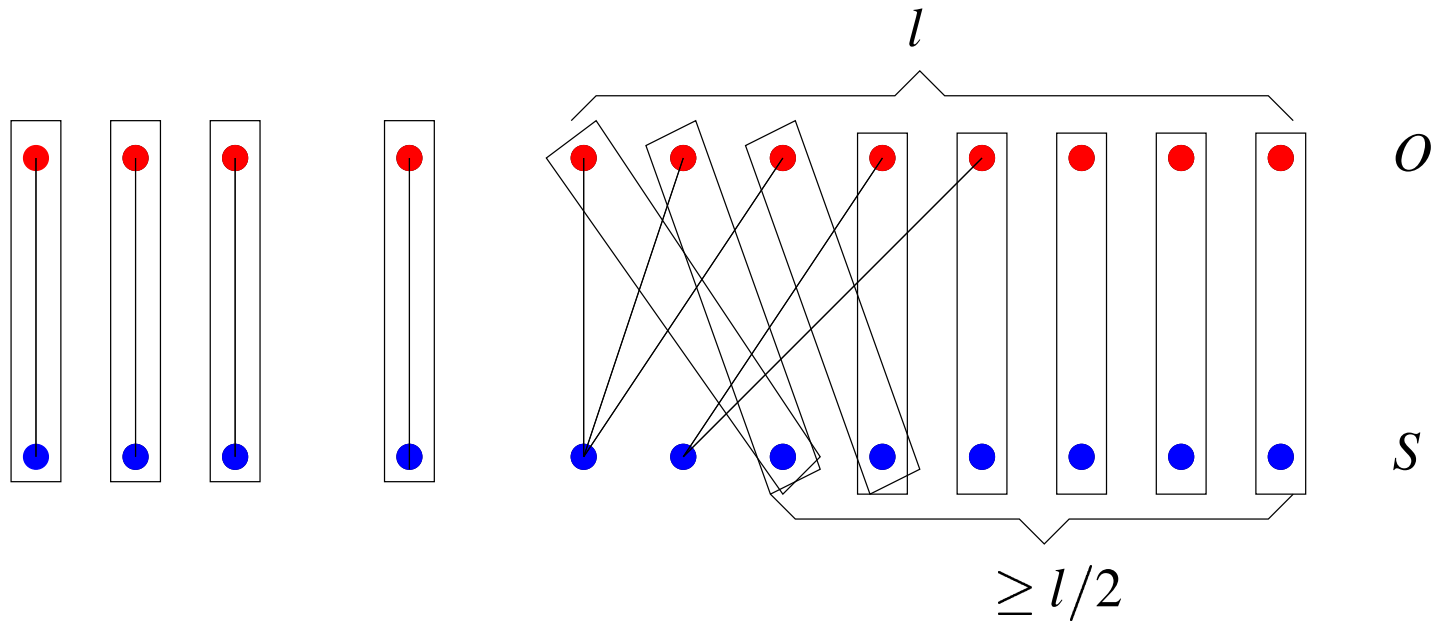
Construct a bipartite graph $G = (O, S, E)$ where there is an edge (o, s) if and only if $s \in S$ captures $o \in O$.

Capture

Swaps considered

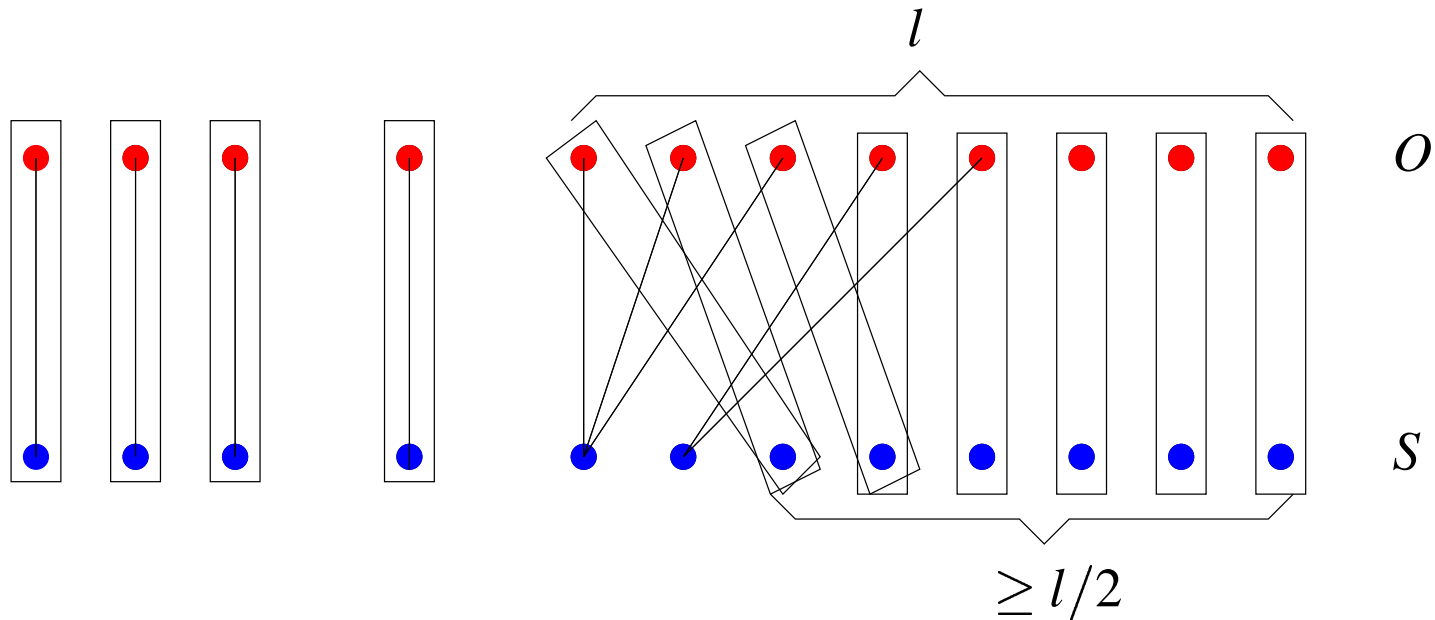


Swaps considered



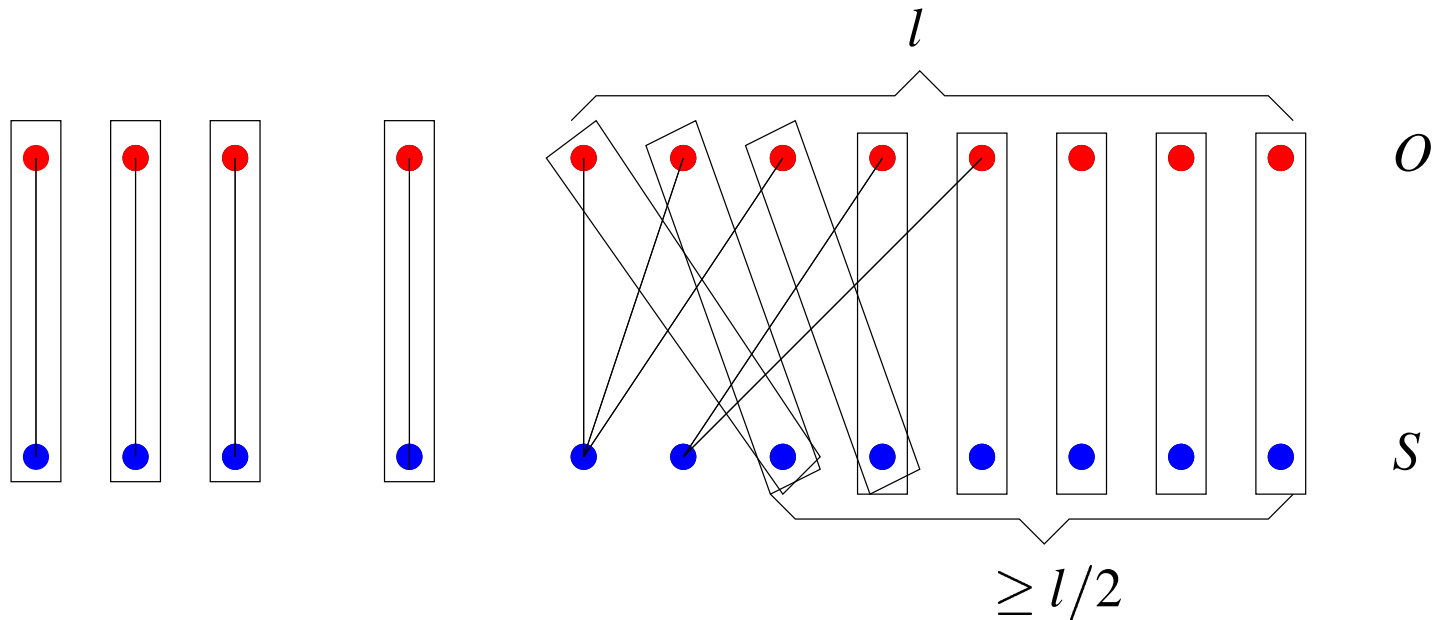
“Why consider the swaps?”

Properties of the swaps considered



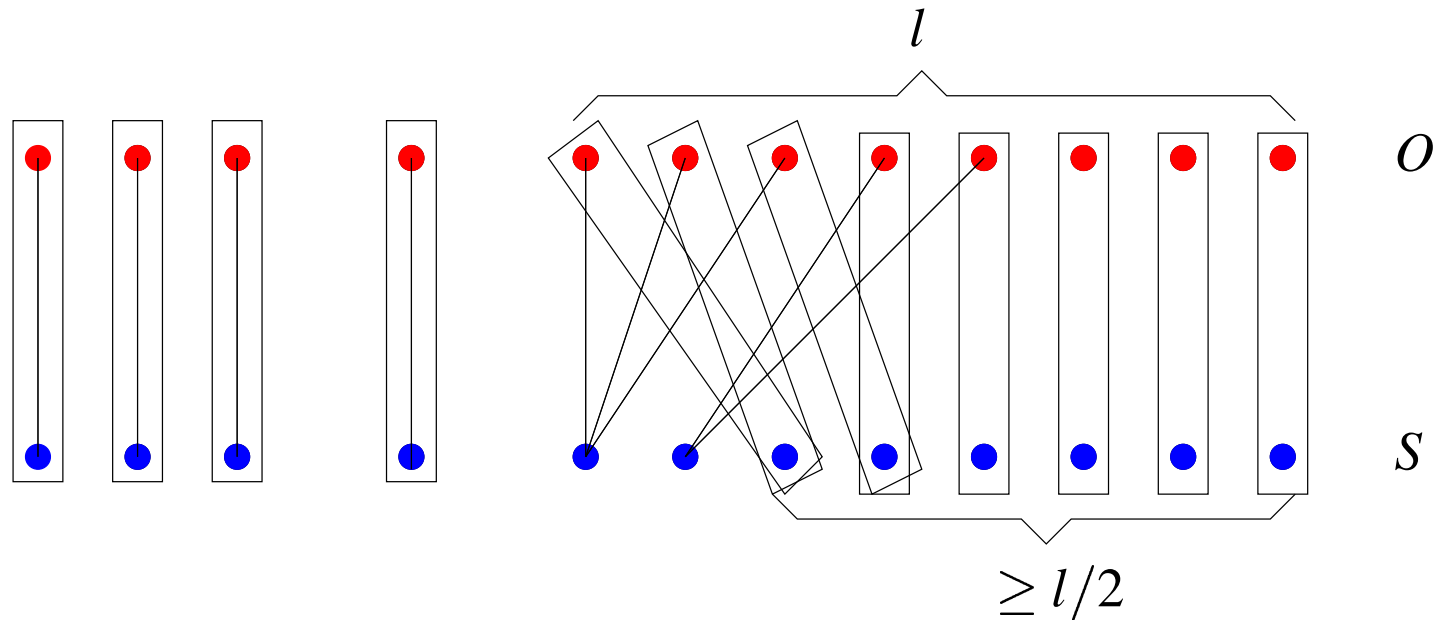
- ▶ If $\langle s, o \rangle$ is considered, then s does not capture any $o' \neq o$.

Properties of the swaps considered



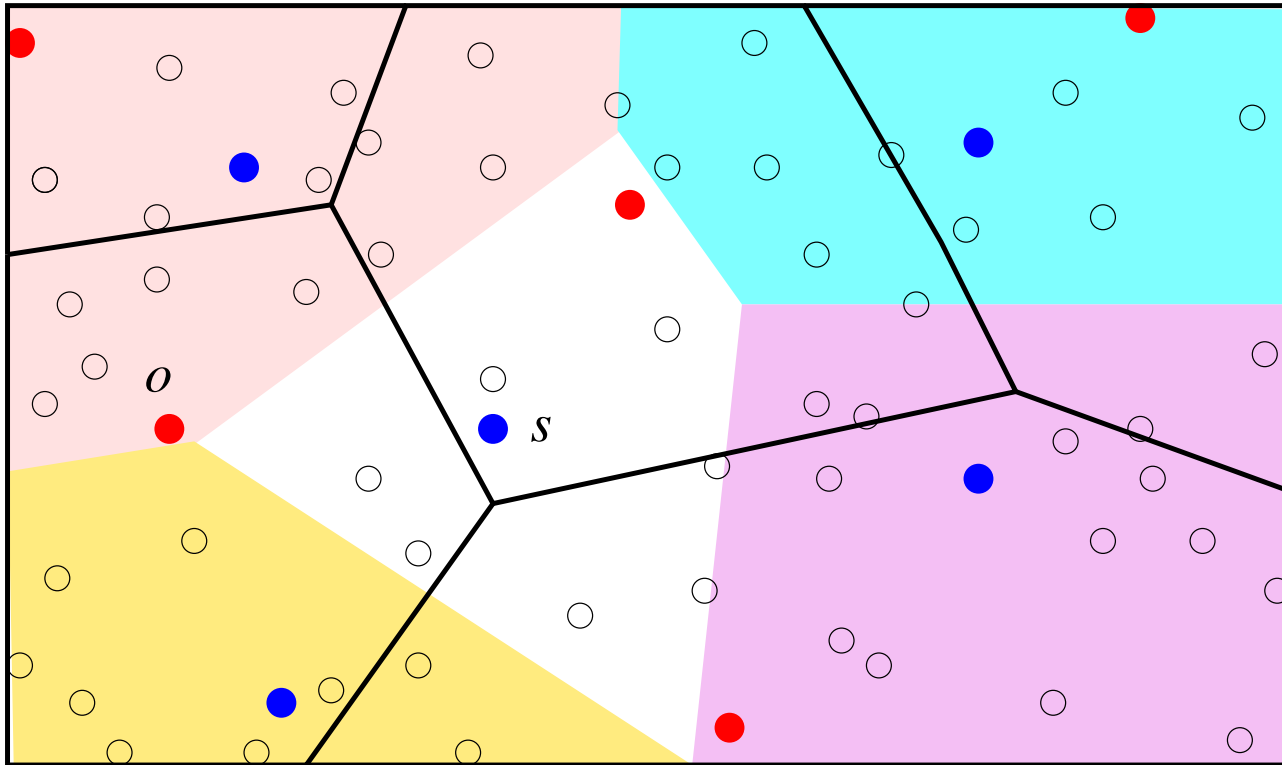
- ▶ If $\langle s, o \rangle$ is considered, then s does not capture any $o' \neq o$.
- ▶ Any $o \in O$ is considered in exactly one swap.

Properties of the swaps considered



- ▶ If $\langle s, o \rangle$ is considered, then s does not capture any $o' \neq o$.
- ▶ Any $o \in O$ is considered in exactly one swap.
- ▶ Any $s \in S$ is considered in at most 2 swaps.

Focus on a swap $\langle s, o \rangle$



Consider a swap $\langle s, o \rangle$ that is one of the k swaps defined above.

We know $cost(S - s + o) \geq cost(S)$.

Upper bound on $\text{cost}(S - s + o)$

- ▶ In the solution $S - s + o$, each point is connected to the closest median in $S - s + o$.

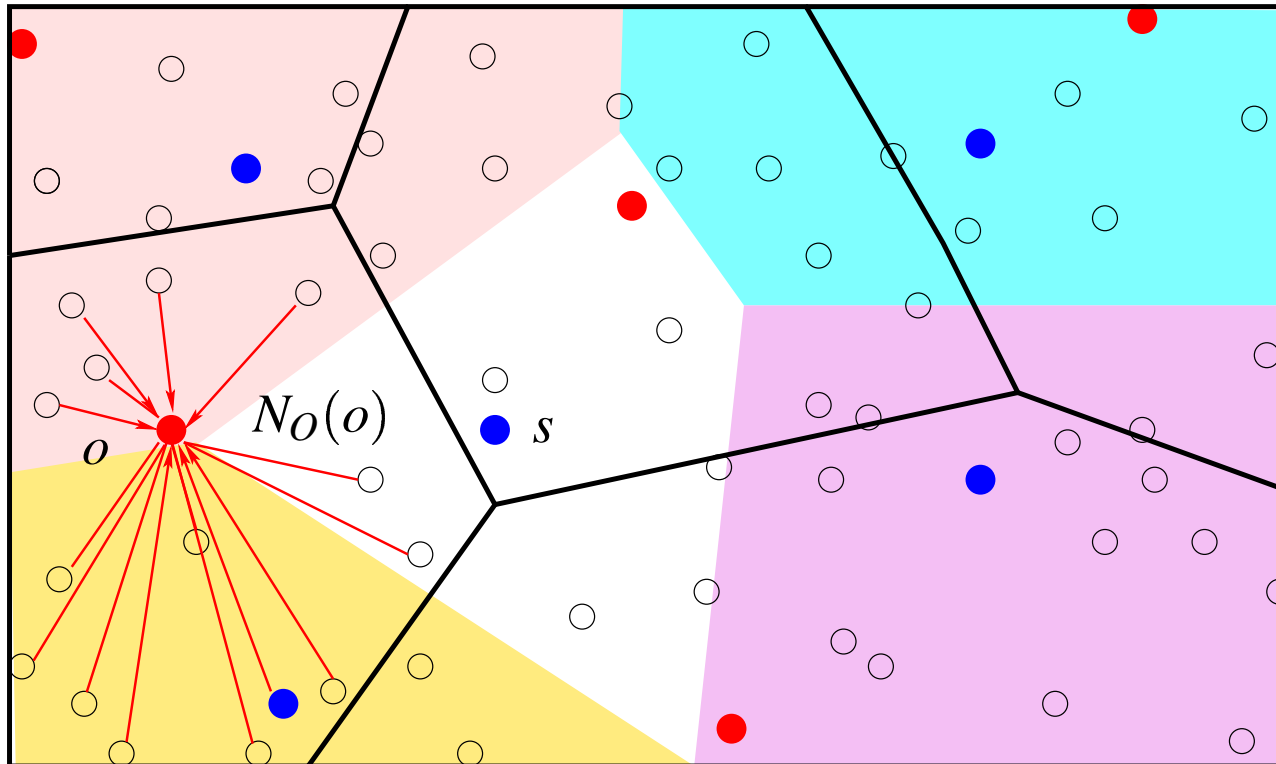
Upper bound on $cost(S - s + o)$

- ▶ In the solution $S - s + o$, each point is connected to the closest median in $S - s + o$.
- ▶ $cost(S - s + o)$ is the sum of distances of all the points to their nearest medians.

Upper bound on $cost(S - s + o)$

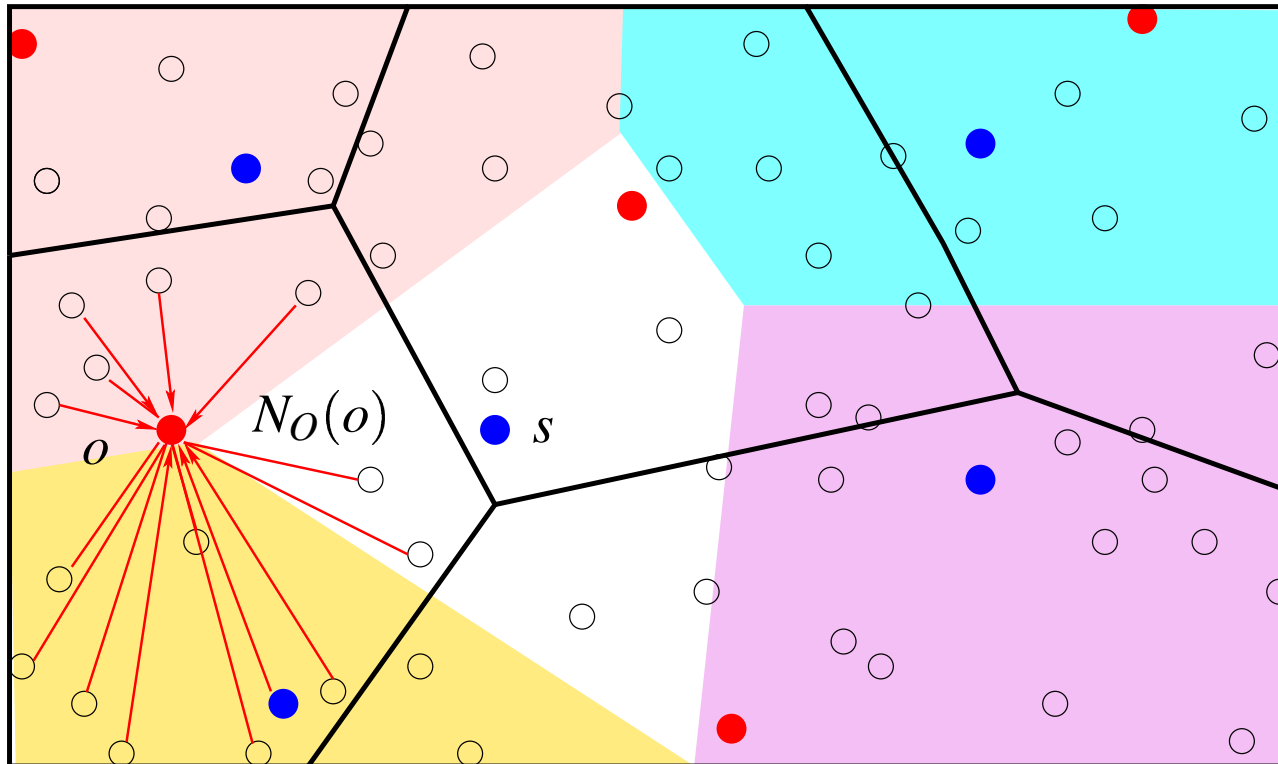
- ▶ In the solution $S - s + o$, each point is connected to the closest median in $S - s + o$.
- ▶ $cost(S - s + o)$ is the sum of distances of all the points to their nearest medians.
- ▶ We are going to demonstrate a possible way of connecting each client to a median in $S - s + o$ to get an upper bound on $cost(S - s + o)$.

Upper bound on $cost(S - s + o)$



Points in $N_O(o)$ are now connected to the new median o .

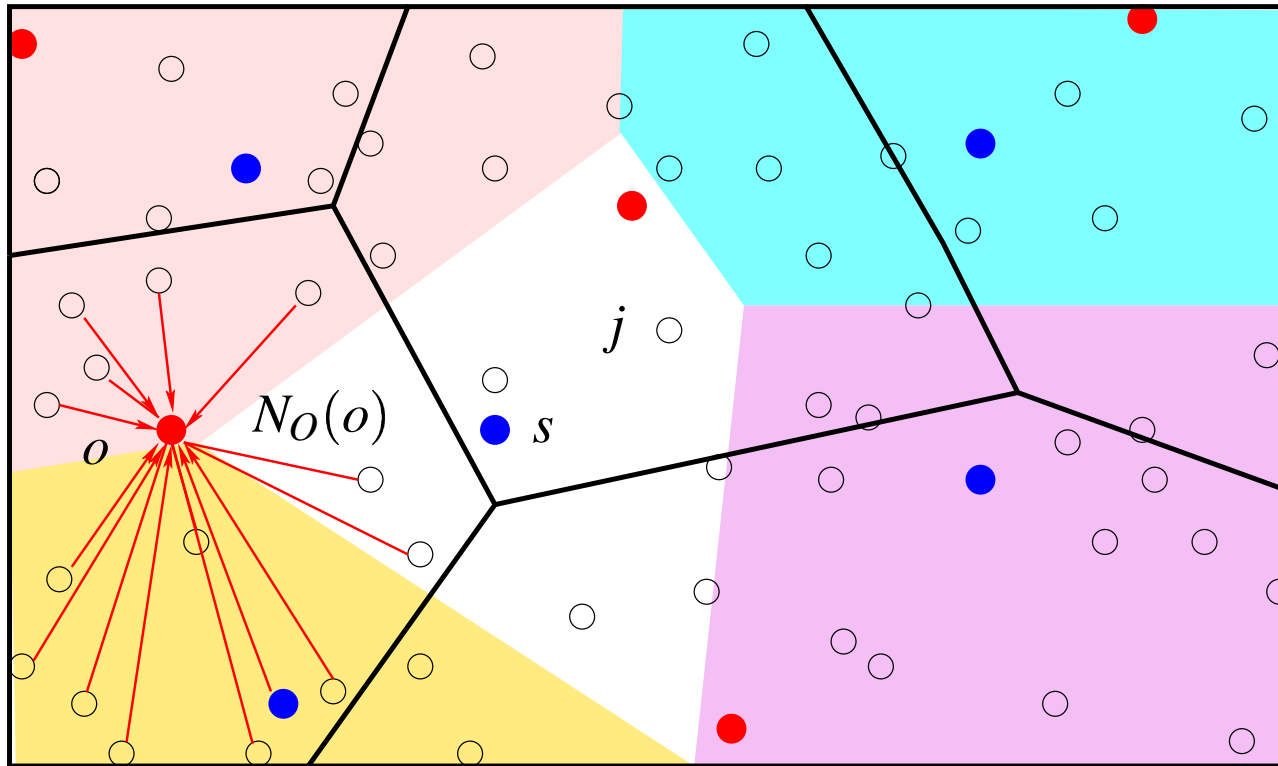
Upper bound on $cost(S - s + o)$



Thus, the increase in the distance for $j \in N_O(o)$ is at most

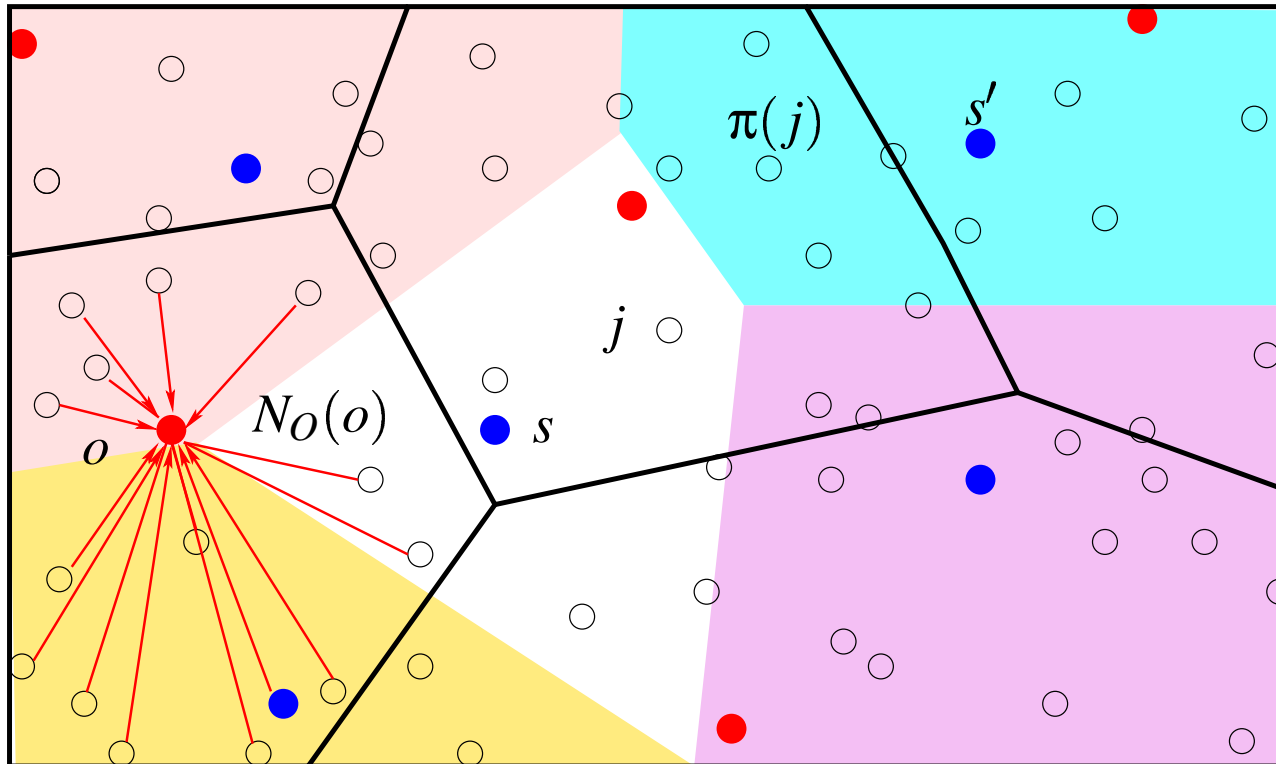
$$O_j - S_j.$$

Upper bound on $\text{cost}(S - s + o)$



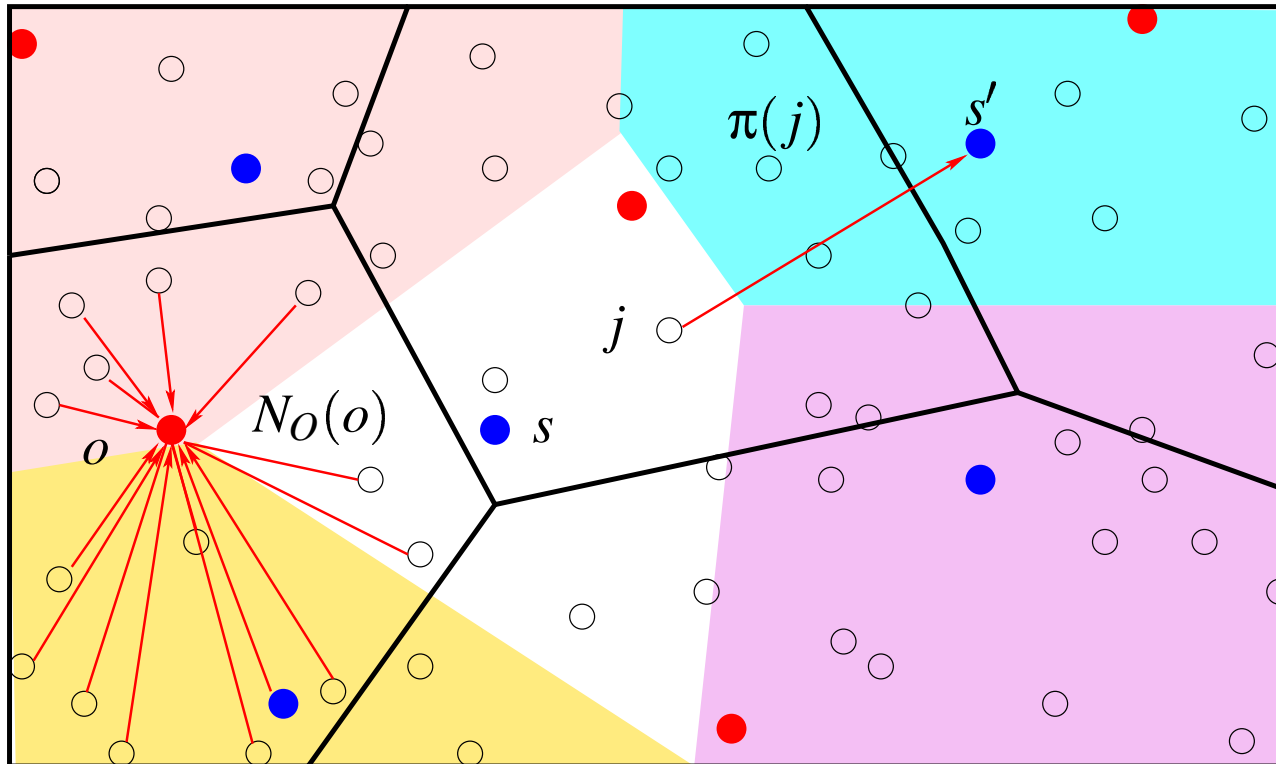
- ▶ Consider a point $j \in N_S(s) \setminus N_O(o)$.

Upper bound on $\text{cost}(S - s + o)$



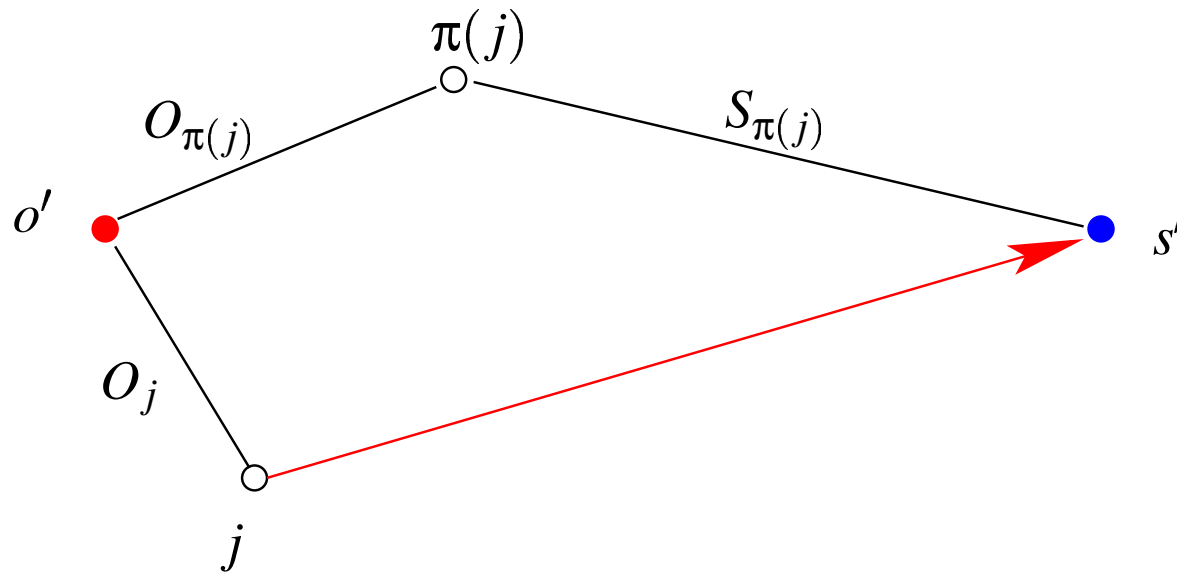
- ▶ Consider a point $j \in N_S(s) \setminus N_O(o)$.
- ▶ Suppose $\pi(j) \in N_S(s')$. (Note that $s' \neq s$.)

Upper bound on $\text{cost}(S - s + o)$



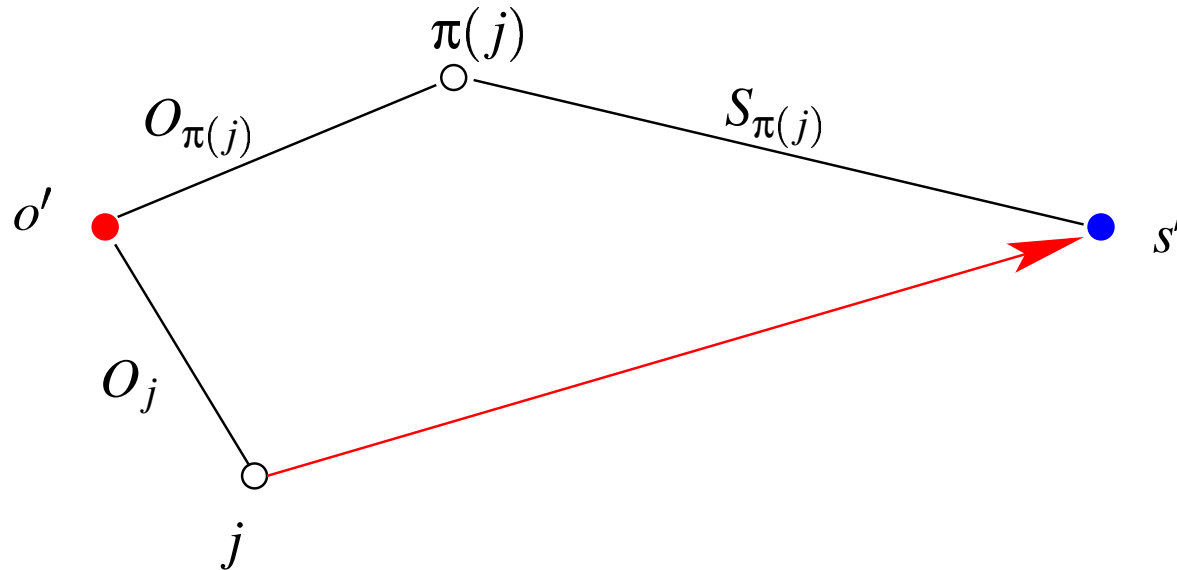
- ▶ Consider a point $j \in N_S(s) \setminus N_O(o)$.
- ▶ Suppose $\pi(j) \in N_S(s')$. (Note that $s' \neq s$.)
- ▶ Connect j to s' now.

Upper bound on $\text{cost}(S - s + o)$



- ▶ New distance of j is at most $O_j + O_{\pi(j)} + S_{\pi(j)}$.

Upper bound on $cost(S - s + o)$



- ▶ New distance of j is at most $O_j + O_{\pi(j)} + S_{\pi(j)}$.
- ▶ Therefore, the increase in the distance for $j \in N_S(s) \setminus N_O(o)$ is at most

$$O_j + O_{\pi(j)} + S_{\pi(j)} - S_j.$$

Upper bound on the increase in the cost

- ▶ Lets try to count the total increase in the cost.

Upper bound on the increase in the cost

- ▶ Lets try to count the total increase in the cost.
- ▶ Points $j \in N_O(o)$ contribute at most

$$(O_j - S_j).$$

Upper bound on the increase in the cost

- ▶ Lets try to count the total increase in the cost.
- ▶ Points $j \in N_O(o)$ contribute at most

$$(O_j - S_j).$$

- ▶ Points $j \in N_S(s) \setminus N_O(o)$ contribute at most

$$(O_j + O_{\pi(j)} + S_{\pi(j)} - S_j).$$

Upper bound on the increase in the cost

- ▶ Lets try to count the total increase in the cost.
- ▶ Points $j \in N_O(o)$ contribute at most

$$(O_j - S_j).$$

- ▶ Points $j \in N_S(s) \setminus N_O(o)$ contribute at most

$$(O_j + O_{\pi(j)} + S_{\pi(j)} - S_j).$$

- ▶ Thus, the total increase is at most,

$$\sum_{j \in N_O(o)} (O_j - S_j) + \sum_{j \in N_S(s) \setminus N_O(o)} (O_j + O_{\pi(j)} + S_{\pi(j)} - S_j).$$

Upper bound on the increase in the cost

$$\sum_{j \in N_O(o)} (O_j - S_j) + \sum_{j \in N_S(s) \setminus N_O(o)} (O_j + O_{\pi(j)} + S_{\pi(j)} - S_j)$$

Upper bound on the increase in the cost

$$\begin{aligned} \sum_{j \in N_O(o)} (O_j - S_j) + \sum_{j \in N_S(s) \setminus N_O(o)} (O_j + O_{\pi(j)} + S_{\pi(j)} - S_j) \\ \geq \text{cost}(S - s + o) - \text{cost}(S) \end{aligned}$$

Upper bound on the increase in the cost

$$\begin{aligned} & \sum_{j \in N_O(o)} (O_j - S_j) + \sum_{j \in N_S(s) \setminus N_O(o)} (O_j + O_{\pi(j)} + S_{\pi(j)} - S_j) \\ & \geq \text{cost}(S - s + o) - \text{cost}(S) \\ & \geq 0 \end{aligned}$$

Plan

- ▶ We have one such inequality for each swap $\langle s, o \rangle$.

$$\sum_{j \in N_O(o)} (O_j - S_j) + \sum_{j \in N_S(s) \setminus N_O(o)} (O_j + O_{\pi(j)} + S_{\pi(j)} - S_j) \geq 0.$$

Plan

- ▶ We have one such inequality for each swap $\langle s, o \rangle$.

$$\sum_{j \in N_O(o)} (O_j - S_j) + \sum_{j \in N_S(s) \setminus N_O(o)} (O_j + O_{\pi(j)} + S_{\pi(j)} - S_j) \geq 0.$$

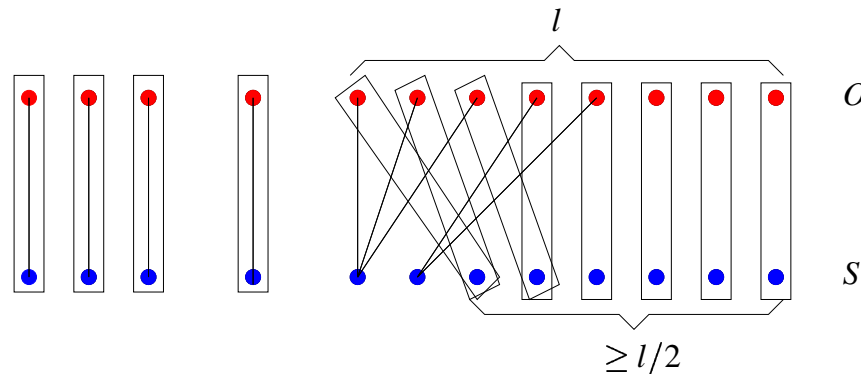
- ▶ There are k swaps that we have defined.

Plan

- ▶ We have one such inequality for each swap $\langle s, o \rangle$.

$$\sum_{j \in N_O(o)} (O_j - S_j) + \sum_{j \in N_S(s) \setminus N_O(o)} (O_j + O_{\pi(j)} + S_{\pi(j)} - S_j) \geq 0.$$

- ▶ There are k swaps that we have defined.

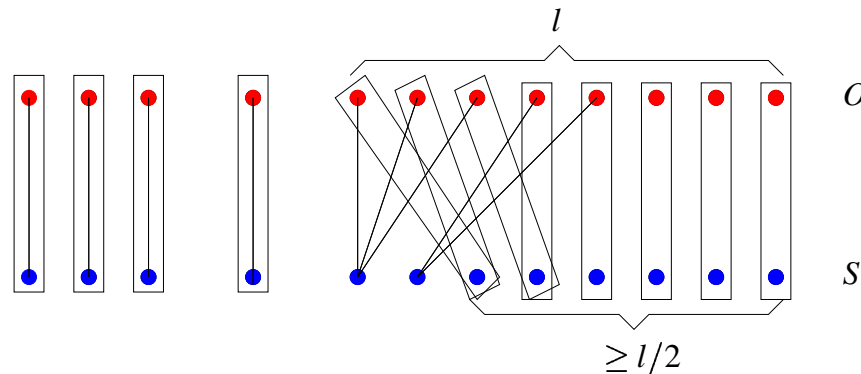


Plan

- ▶ We have one such inequality for each swap $\langle s, o \rangle$.

$$\sum_{j \in N_O(o)} (O_j - S_j) + \sum_{j \in N_S(s) \setminus N_O(o)} (O_j + O_{\pi(j)} + S_{\pi(j)} - S_j) \geq 0.$$

- ▶ There are k swaps that we have defined.



- ▶ Lets add the inequalities for all the k swaps and see what we get!

The first term . . .

$$\left[\sum_{j \in N_O(o)} (O_j - S_j) \right] + \sum_{j \in N_S(s) \setminus N_O(o)} (O_j + O_{\pi(j)} + S_{\pi(j)} - S_j) \geq 0.$$

The first term . . .

$$\left[\sum_{j \in N_O(o)} (O_j - S_j) \right] + \sum_{j \in N_S(s) \setminus N_O(o)} (O_j + O_{\pi(j)} + S_{\pi(j)} - S_j) \geq 0.$$

Note that each $o \in O$ is considered in exactly one swap.

The first term . . .

$$\left[\sum_{j \in N_O(o)} (O_j - S_j) \right] + \sum_{j \in N_S(s) \setminus N_O(o)} (O_j + O_{\pi(j)} + S_{\pi(j)} - S_j) \geq 0.$$

Note that each $o \in O$ is considered in exactly one swap.
Thus, the first term added over all the swaps is

$$\sum_{o \in O} \sum_{j \in N_O(o)} (O_j - S_j)$$

The first term . . .

$$\left[\sum_{j \in N_O(o)} (O_j - S_j) \right] + \sum_{j \in N_S(s) \setminus N_O(o)} (O_j + O_{\pi(j)} + S_{\pi(j)} - S_j) \geq 0.$$

Note that each $o \in O$ is considered in exactly one swap.
Thus, the first term added over all the swaps is

$$\begin{aligned} & \sum_{o \in O} \sum_{j \in N_O(o)} (O_j - S_j) \\ &= \sum_j (O_j - S_j) \end{aligned}$$

The first term . . .

$$\left[\sum_{j \in N_O(o)} (O_j - S_j) \right] + \sum_{j \in N_S(s) \setminus N_O(o)} (O_j + O_{\pi(j)} + S_{\pi(j)} - S_j) \geq 0.$$

Note that each $o \in O$ is considered in exactly one swap.
Thus, the first term added over all the swaps is

$$\begin{aligned} & \sum_{o \in O} \sum_{j \in N_O(o)} (O_j - S_j) \\ &= \sum_j (O_j - S_j) \\ &= \text{cost}(O) - \text{cost}(S). \end{aligned}$$

The second term ...

$$\sum_{j \in N_O(o)} (O_j - S_j) + \left[\sum_{j \in N_S(s) \setminus N_O(o)} (O_j + O_{\pi(j)} + S_{\pi(j)} - S_j) \right] \geq 0.$$

The second term ...

$$\sum_{j \in N_O(o)} (O_j - S_j) + \left[\sum_{j \in N_S(s) \setminus N_O(o)} (O_j + O_{\pi(j)} + S_{\pi(j)} - S_j) \right] \geq 0.$$

Note that

$$O_j + O_{\pi(j)} + S_{\pi(j)} \geq S_j.$$

The second term ...

$$\sum_{j \in N_O(o)} (O_j - S_j) + \left[\sum_{j \in N_S(s) \setminus N_O(o)} (O_j + O_{\pi(j)} + S_{\pi(j)} - S_j) \right] \geq 0.$$

Note that

$$O_j + O_{\pi(j)} + S_{\pi(j)} \geq S_j.$$

Thus

$$O_j + O_{\pi(j)} + S_{\pi(j)} - S_j \geq 0.$$

The second term ...

$$\sum_{j \in N_O(o)} (O_j - S_j) + \left[\sum_{j \in N_S(s) \setminus N_O(o)} (O_j + O_{\pi(j)} + S_{\pi(j)} - S_j) \right] \geq 0.$$

Note that

$$O_j + O_{\pi(j)} + S_{\pi(j)} \geq S_j.$$

Thus

$$O_j + O_{\pi(j)} + S_{\pi(j)} - S_j \geq 0.$$

Thus the second term is at most

$$\sum_{j \in N_S(s)} (O_j + O_{\pi(j)} + S_{\pi(j)} - S_j).$$

The second term ...

Note that each $s \in S$ is considered in at most two swaps.

The second term ...

Note that each $s \in S$ is considered in at most two swaps.

Thus, the second term added over all the swaps is at most

$$2 \sum_{s \in S} \sum_{j \in N_S(s)} (O_j + O_{\pi(j)} + S_{\pi(j)} - S_j)$$

The second term ...

Note that each $s \in S$ is considered in at most two swaps.

Thus, the second term added over all the swaps is at most

$$\begin{aligned} & 2 \sum_{s \in S} \sum_{j \in N_S(s)} (O_j + O_{\pi(j)} + S_{\pi(j)} - S_j) \\ &= 2 \sum_j (O_j + O_{\pi(j)} + S_{\pi(j)} - S_j) \end{aligned}$$

The second term ...

Note that each $s \in S$ is considered in at most two swaps.

Thus, the second term added over all the swaps is at most

$$\begin{aligned} & 2 \sum_{s \in S} \sum_{j \in N_S(s)} (O_j + O_{\pi(j)} + S_{\pi(j)} - S_j) \\ &= 2 \sum_j (O_j + O_{\pi(j)} + S_{\pi(j)} - S_j) \\ &= 2 \left[\sum_j O_j + \sum_j O_{\pi(j)} + \sum_j S_{\pi(j)} - \sum_j S_j \right] \end{aligned}$$

The second term ...

Note that each $s \in S$ is considered in at most two swaps.

Thus, the second term added over all the swaps is at most

$$\begin{aligned} & 2 \sum_{s \in S} \sum_{j \in N_S(s)} (O_j + O_{\pi(j)} + S_{\pi(j)} - S_j) \\ &= 2 \sum_j (O_j + O_{\pi(j)} + S_{\pi(j)} - S_j) \\ &= 2 \left[\sum_j O_j + \sum_j O_{\pi(j)} + \sum_j S_{\pi(j)} - \sum_j S_j \right] \\ &= 4 \cdot \text{cost}(O). \end{aligned}$$

Putting things together

$$0 \leq \sum_{\langle s, o \rangle} \left[\sum_{j \in N_O(o)} (O_j - S_j) + \sum_{j \in N_S(s) \setminus N_O(o)} (O_j + O_{\pi(j)} + S_{\pi(j)} - S_j) \right]$$

Putting things together

$$0 \leq \sum_{\langle s, o \rangle} \left[\sum_{j \in N_O(o)} (O_j - S_j) + \sum_{j \in N_S(s) \setminus N_O(o)} (O_j + O_{\pi(j)} + S_{\pi(j)} - S_j) \right]$$
$$\leq [\text{cost}(O) - \text{cost}(S)] + [4 \cdot \text{cost}(O)]$$

Putting things together

$$\begin{aligned} 0 &\leq \sum_{\langle s,o \rangle} \left[\sum_{j \in N_O(o)} (O_j - S_j) \quad + \quad \sum_{j \in N_S(s) \setminus N_O(o)} (O_j + O_{\pi(j)} + S_{\pi(j)} - S_j) \right] \\ &\leq [\text{cost}(O) - \text{cost}(S)] \quad + \quad [4 \cdot \text{cost}(O)] \\ &= 5 \cdot \text{cost}(O) - \text{cost}(S). \end{aligned}$$

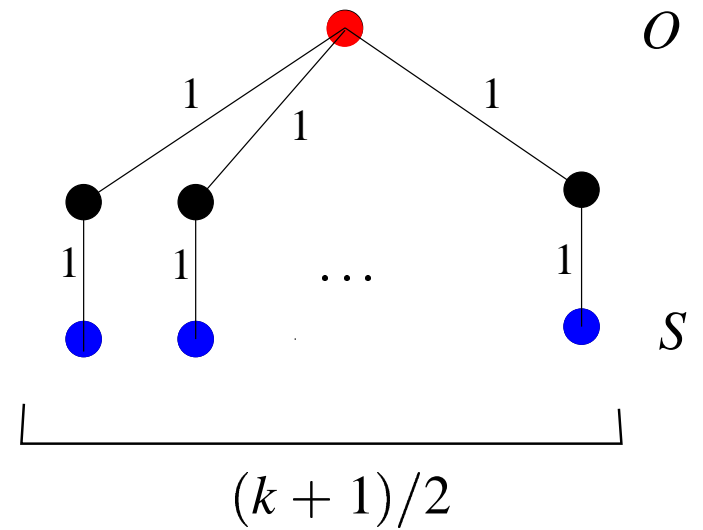
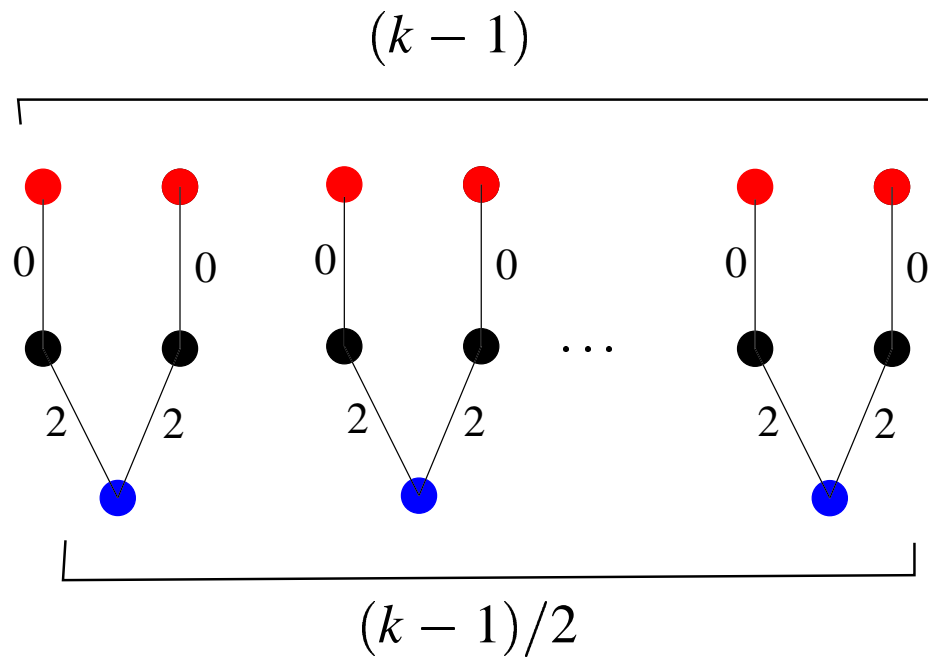
Putting things together

$$\begin{aligned} 0 &\leq \sum_{\langle s,o \rangle} \left[\sum_{j \in N_O(o)} (O_j - S_j) + \sum_{j \in N_S(s) \setminus N_O(o)} (O_j + O_{\pi(j)} + S_{\pi(j)} - S_j) \right] \\ &\leq [\text{cost}(O) - \text{cost}(S)] + [4 \cdot \text{cost}(O)] \\ &= 5 \cdot \text{cost}(O) - \text{cost}(S). \end{aligned}$$

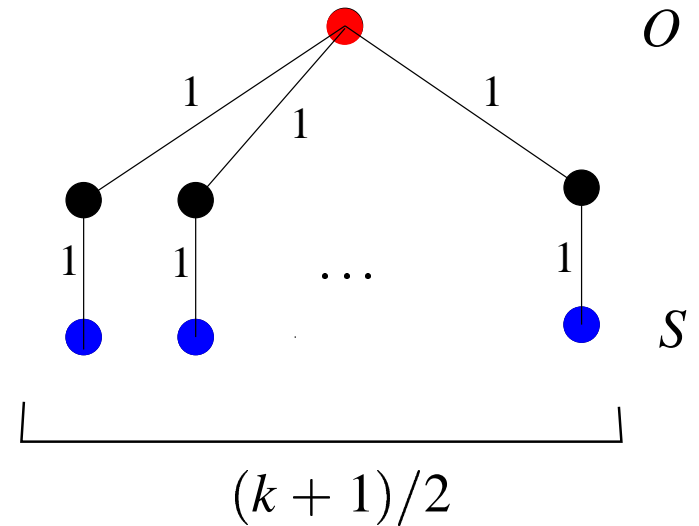
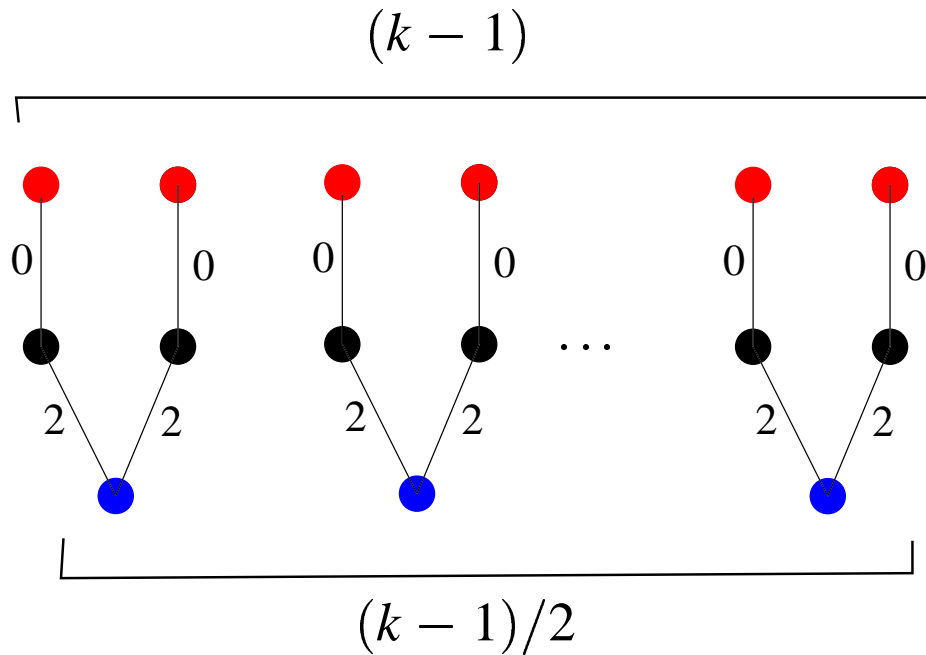
Therefore,

$$\text{cost}(S) \leq 5 \cdot \text{cost}(O).$$

A tight example

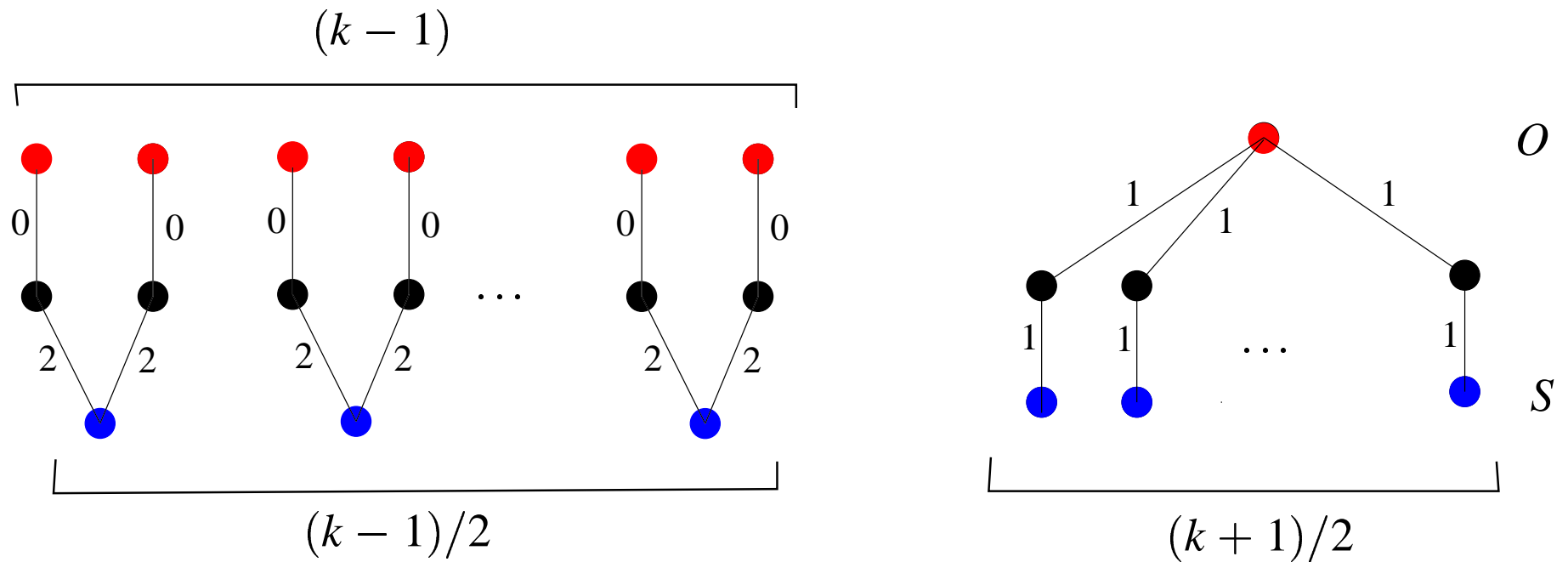


A tight example



► $cost(S) = 4 \cdot (k - 1)/2 + (k + 1)/2 = (5k - 3)/2$

A tight example



► $cost(S) = 4 \cdot (k-1)/2 + (k+1)/2 = (5k-3)/2$

► $cost(O) = 0 + (k+1)/2 = (k+1)/2$

Doing multiple swaps

- ▶ Doing single swaps yields 5 approximation.

Doing multiple swaps

- ▶ Doing single swaps yields 5 approximation.
- ▶ Doing p -way swaps yields $(3 + 2/p)$ approximation.