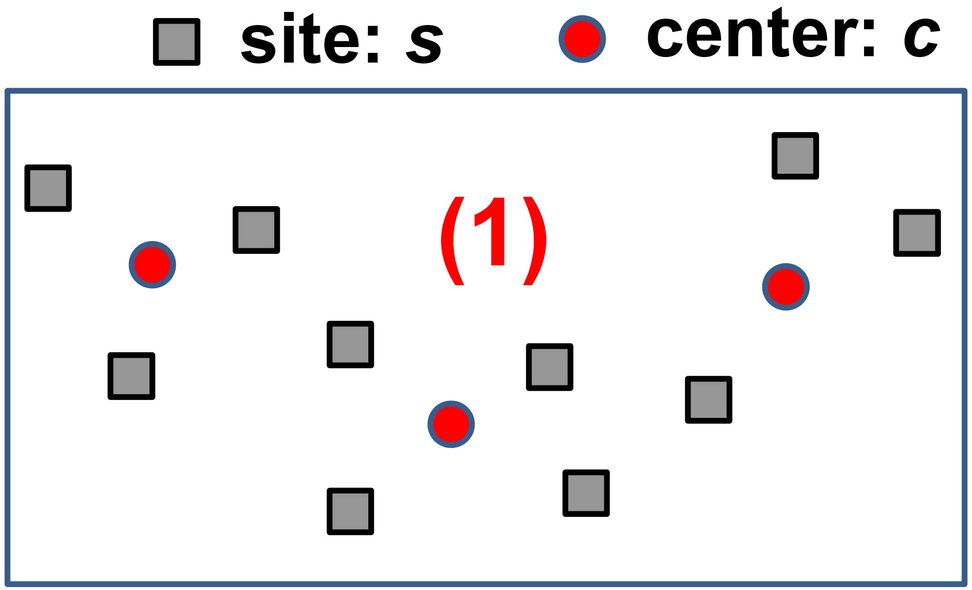
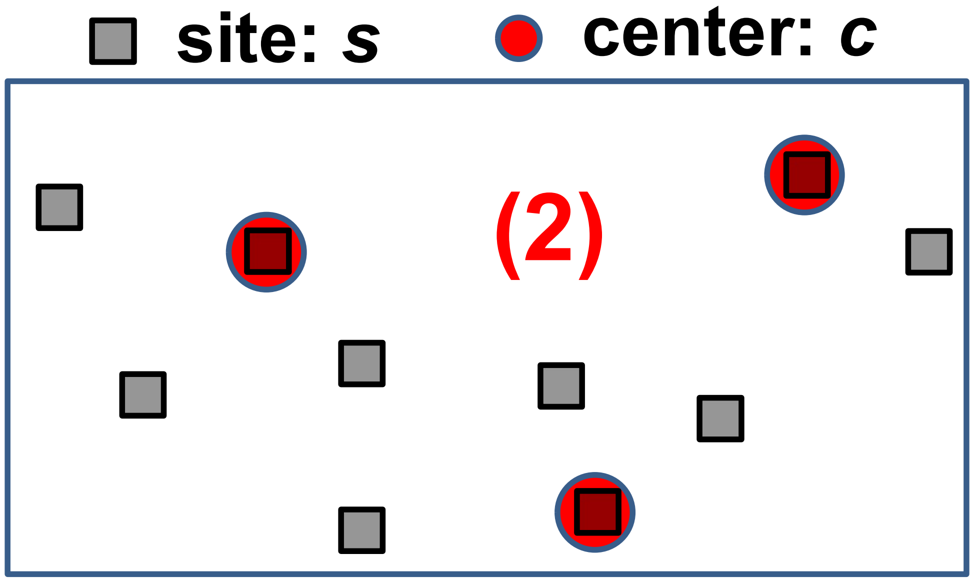
Center Selection Problem

Input: n sites: $S = \{s_1, s_2, ..., s_n\}$

Output: Locations of k centers: $C = \{c_1, c_2, ..., c_k\}$



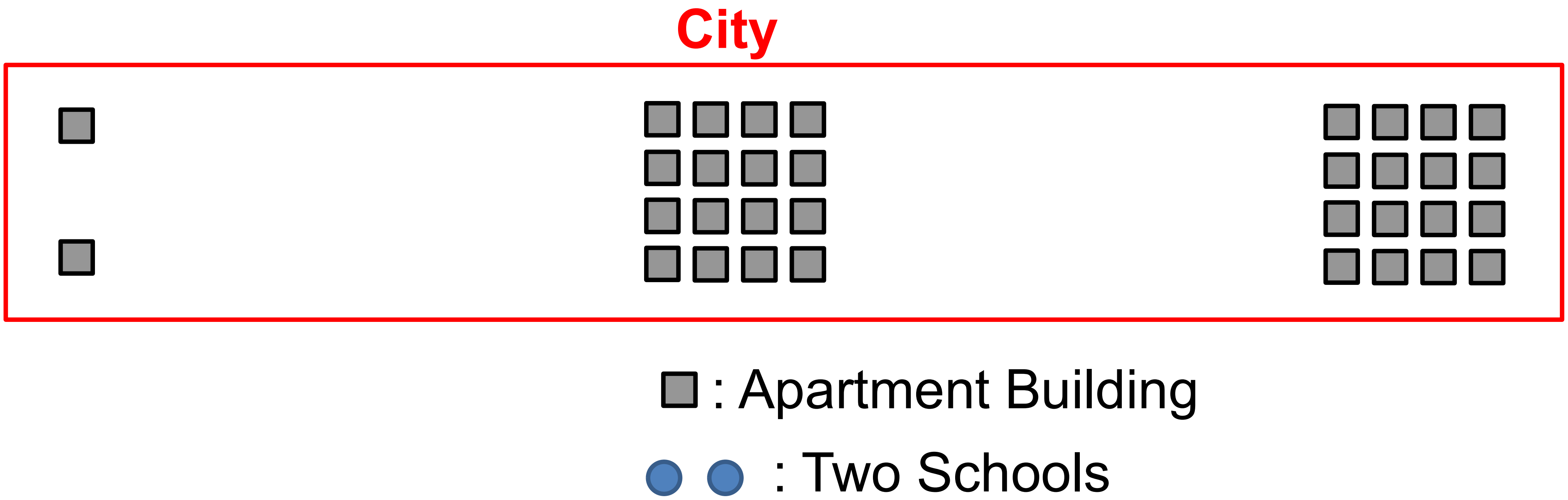


Two interpretations of this formulation

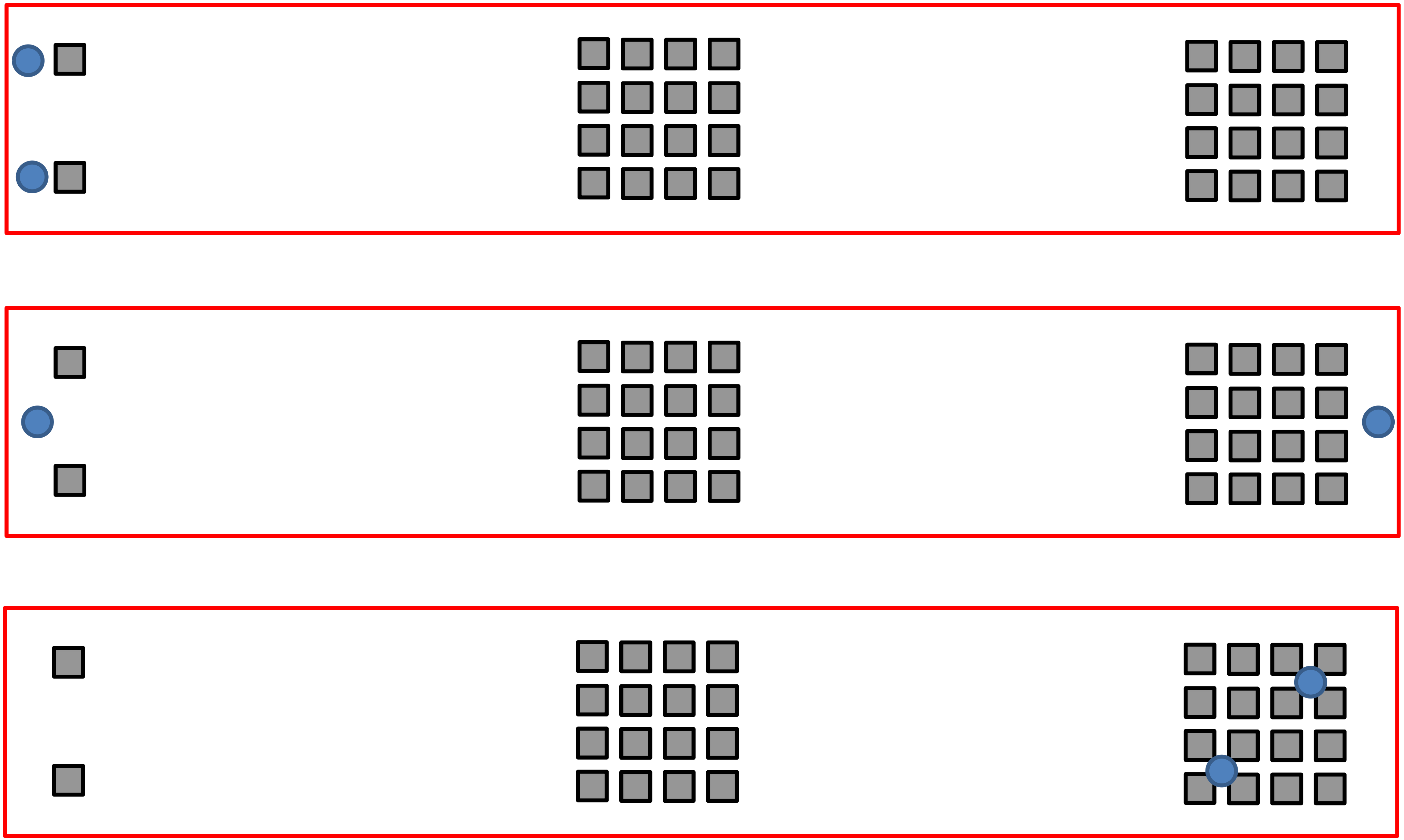
(1) Without any additional constraint conditions.

(2) With the constraint condition: $c_j \in S,\ j = 1, 2, ..., k$

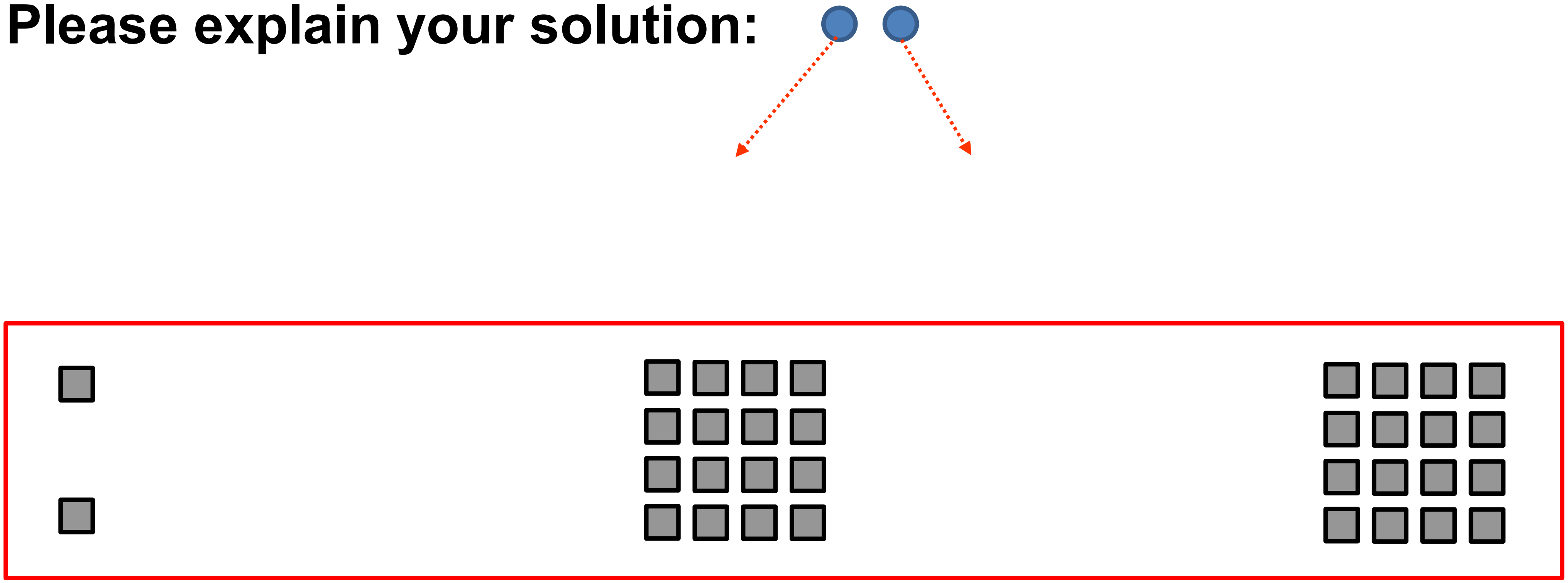
Problem (City Planning): Determine the locations of two schools in a city with 34 apartment buildings. The size of each apartment building is the same (e.g., for 100 families).



Examples of Bad Solutions



Please explain your solution:



Center Selection Problem

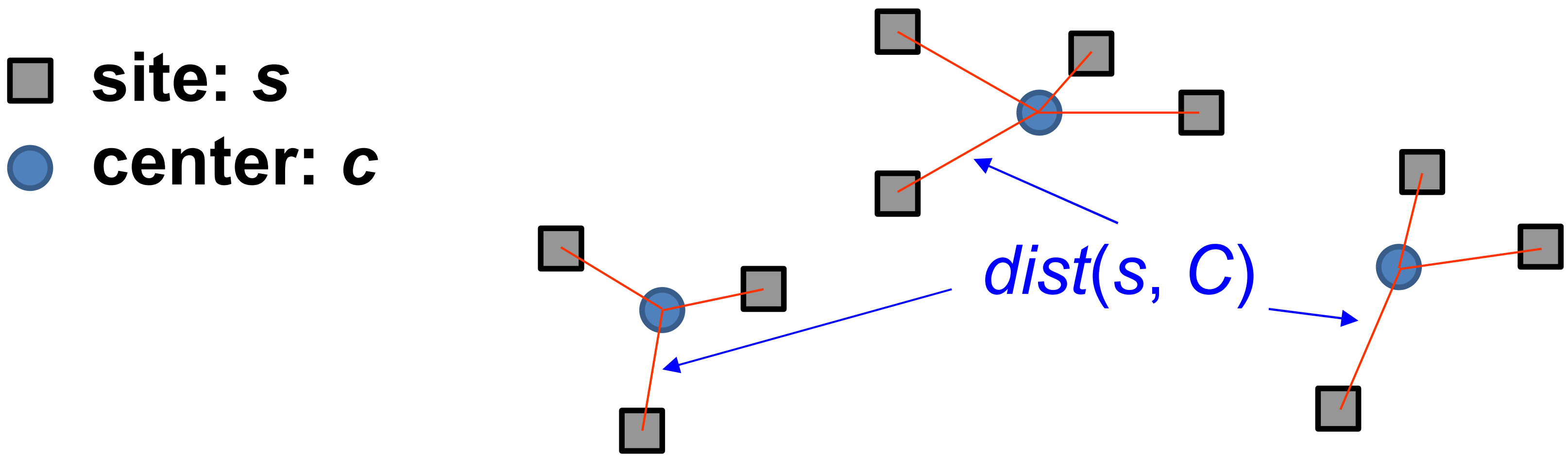
Input: n sites: $S = \{s_1, s_2, ..., s_n\}$

Output: Locations of k centers: $C = \{c_1, c_2, ..., c_k\}$

Objective: Minimize the maximum distance from each site to the nearest center.

$$\text{Minimize} \quad r = Max_{s \in S}\{dist(s, C)\}$$

$$\text{where} \quad dist(s, C) = Min_{c \in C}\{dist(s, c)\}$$

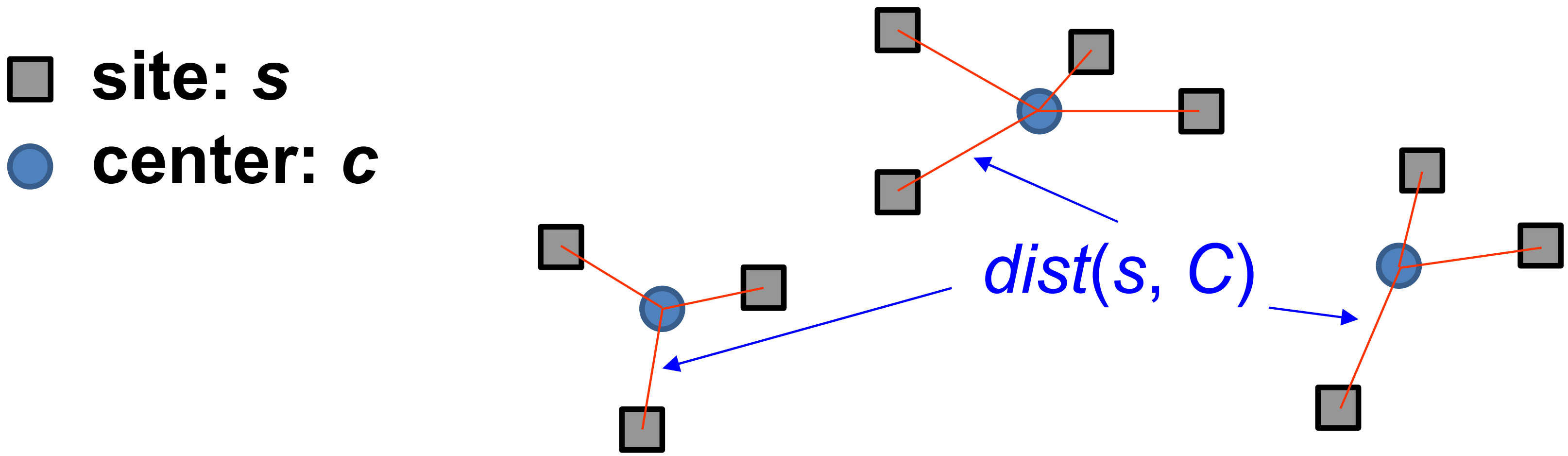


Input: n **sites:** $S = \{s_1, s_2, ..., s_n\}$

Output: **Locations of** k **centers:** $C = \{c_1, c_2, ..., c_k\}$

Objective: **Minimize the maximum distance from each site to the nearest center.**

$$\text{Minimize} \quad r = Max_{s \in S}\{dist(s, C)\}$$



Two interpretations of this formulation

(1) Without any additional constraint conditions.

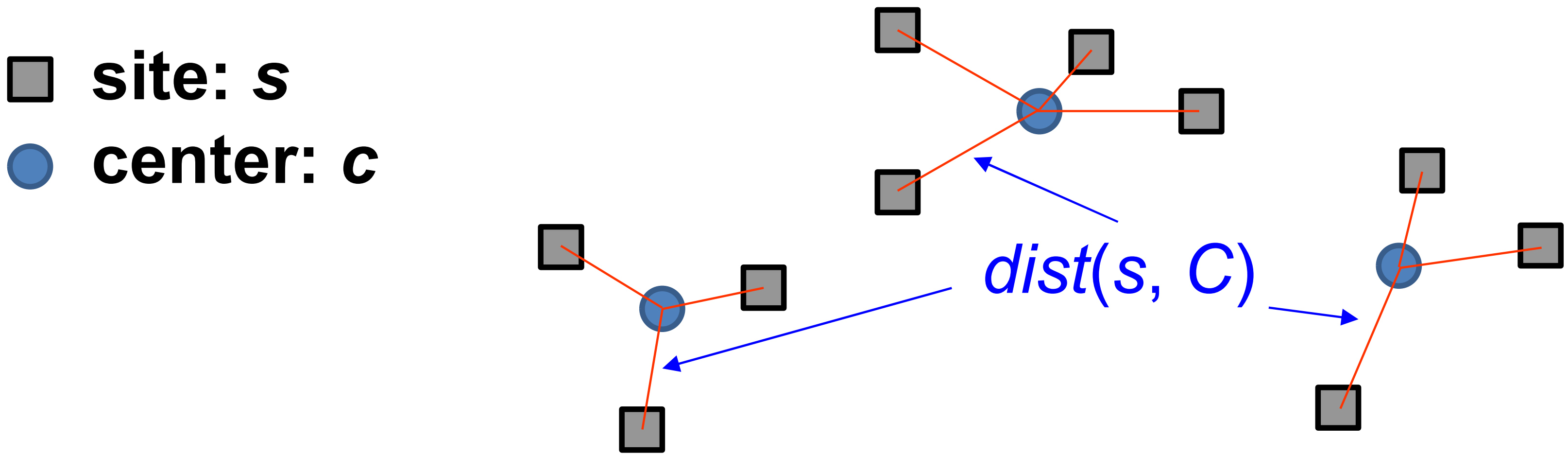
(2) With the constraint condition: $c_j \in S,\ j = 1, 2, ..., k$

Input: n sites: $S = \{s_1, s_2, ..., s_n\}$

Output: Locations of k centers: $C = \{c_1, c_2, ..., c_k\}$

Objective: Minimize the maximum distance from each site to the nearest center.

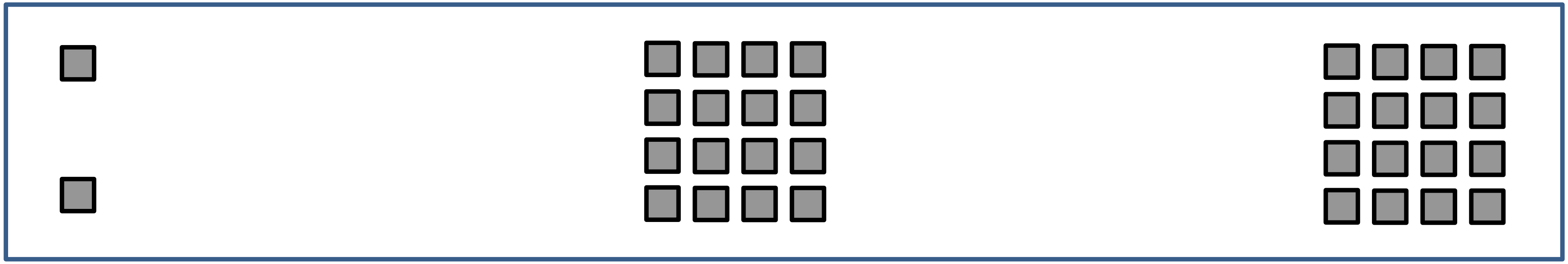
$$\text{Minimize} \quad r = Max_{s \in S}\{dist(s, C)\}$$



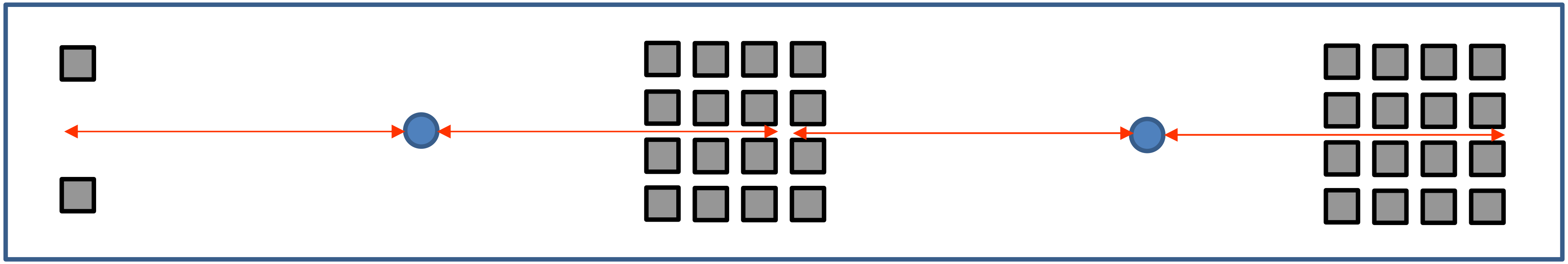
In center selection algorithms, centers are selected from the given sites. However, in general, centers can be any points (e.g., the midpoint between two sites).

Optimal Solution of the Center Selection Problem:

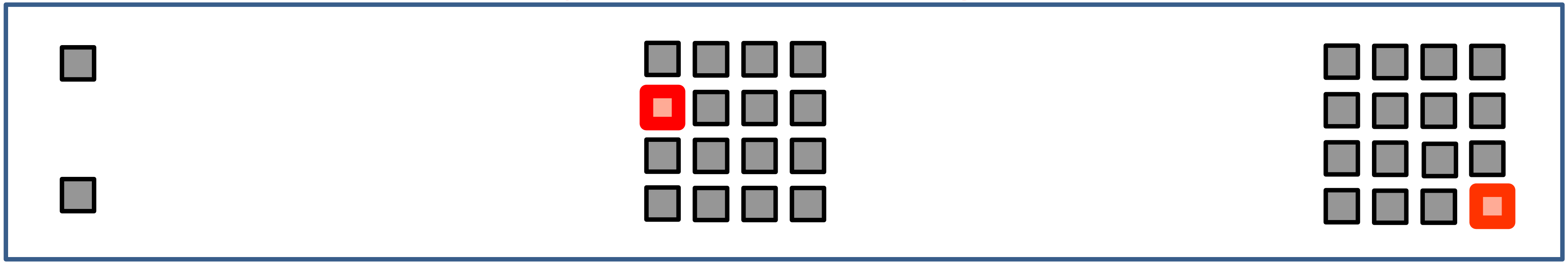
Problem



Optimal Solution

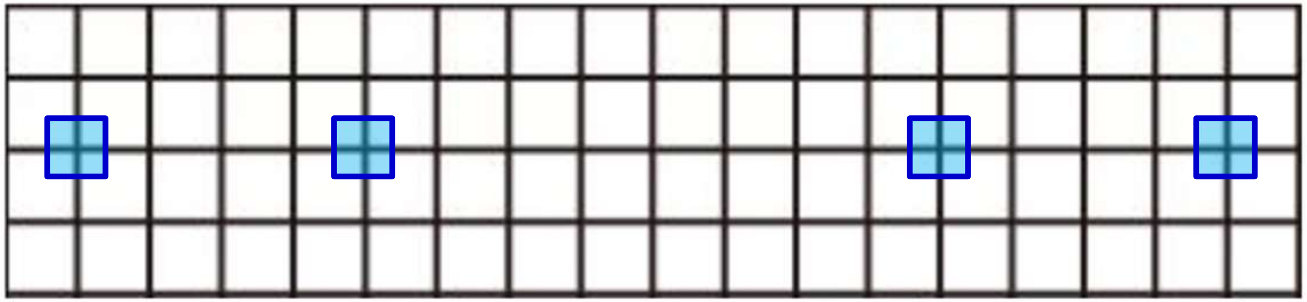


Example of a result by a selection algorithm.

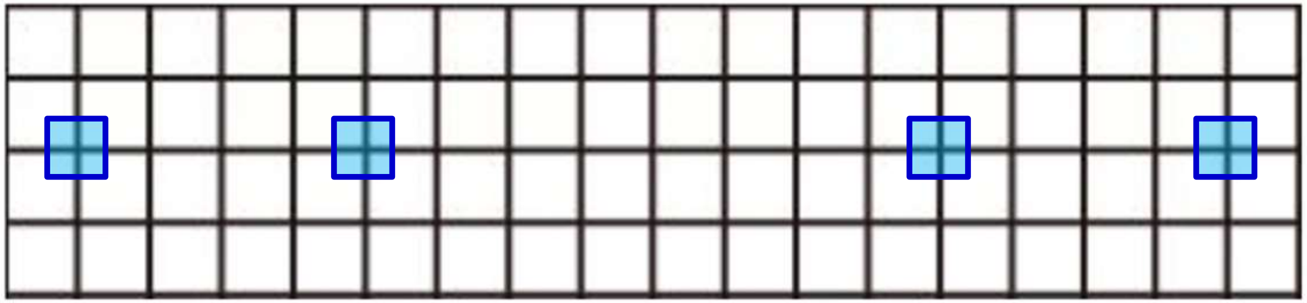


Problem

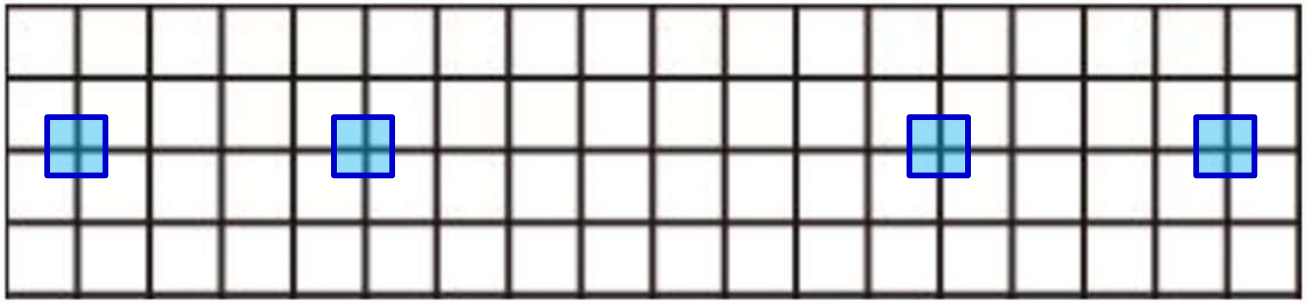
$k = 2$



Optimal solution under no constraint condition:

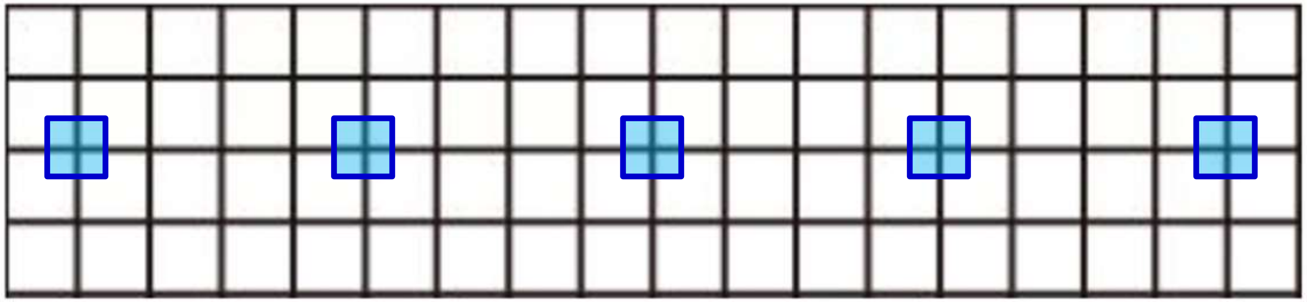


Optimal Solution under $c_j \in S,\ j = 1, 2, \ldots, k$

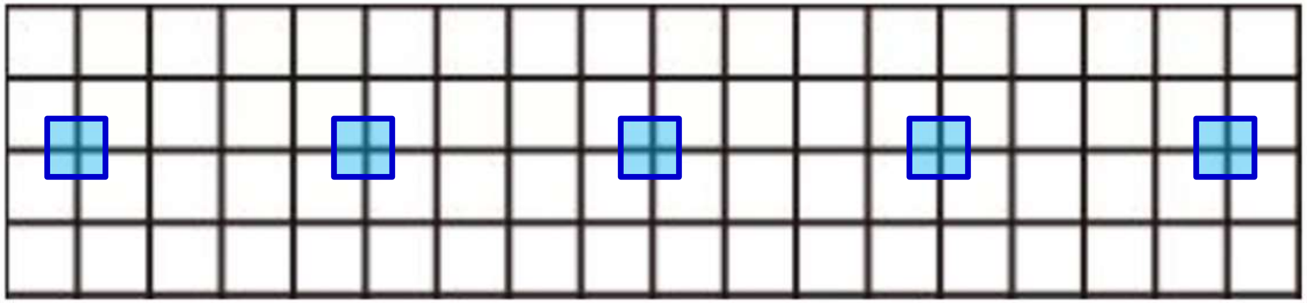


Problem

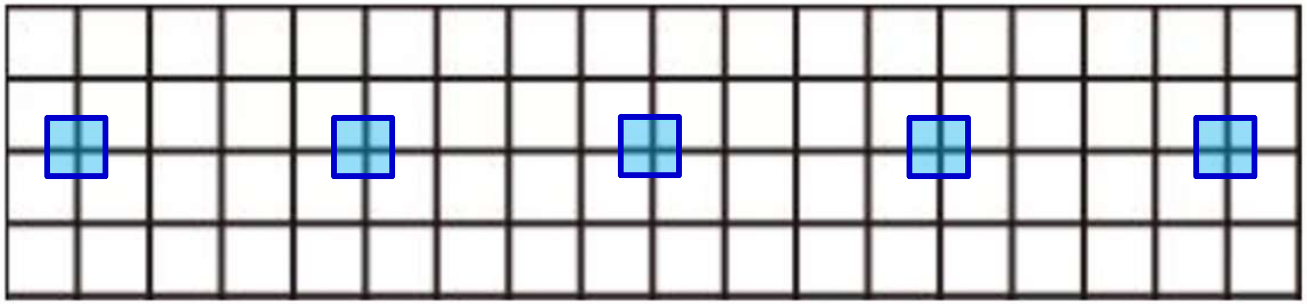
$k = 2$



Optimal solution under no constraint condition:



Optimal Solution under $c_j \in S,\ j = 1,\ 2,\ \ldots,\ k$



Optimal solution: C^*

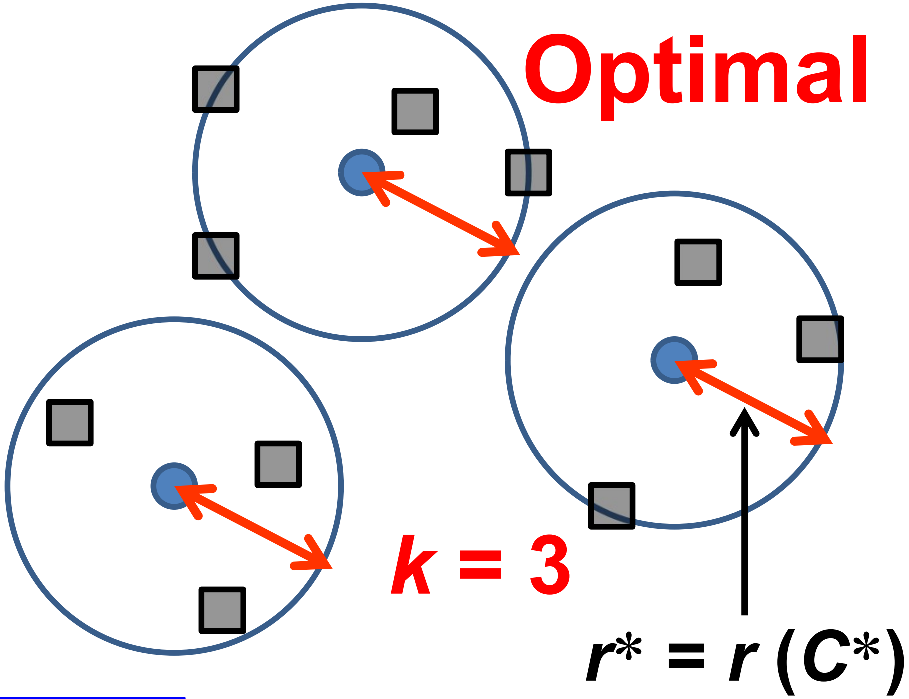
Optimal value: $r^* = r(C^*)$

Virtual Center Selection Algorithm

Iteration of the following:

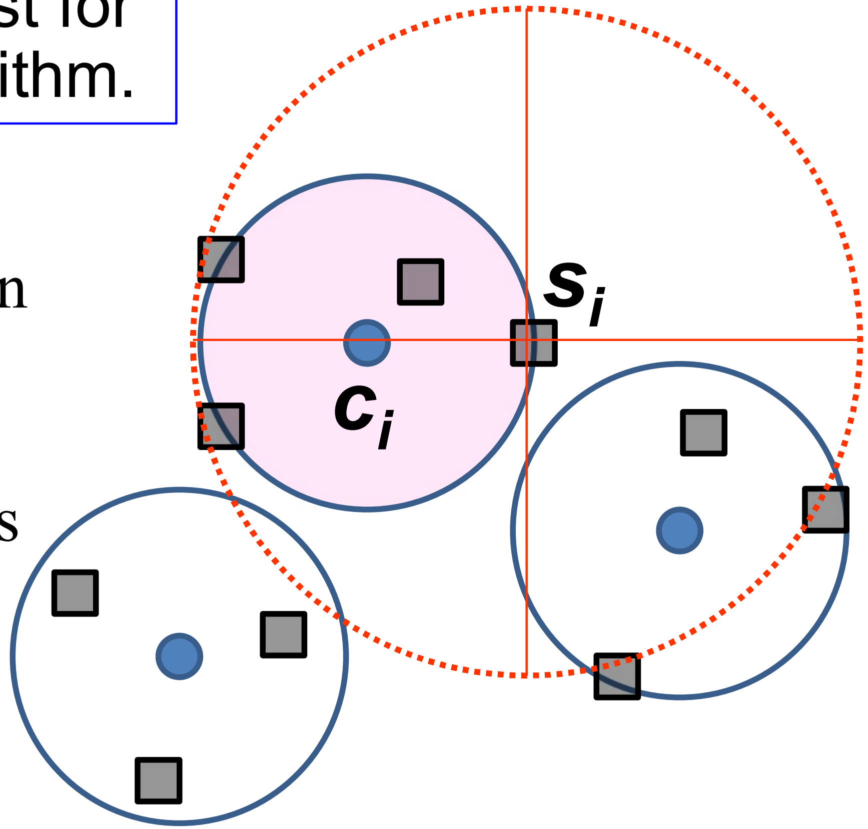
(i) Select a site s.

(ii) Remove all sites covered by s within distance $2r^*$



This is not a usable algorithm. This is just for the theoretical analysis of the next algorithm.

Idea behind this algorithm: All sites covered by the center c_i within distance r^* in the optimal selection are always covered by a selected site s_i within distance $2r^*$ if s_i is within distance r^* from c_i. Thus, after k sites are selected, all sites are removed.



==> **2-approximation algorithm.**

Virtual Center Selection Algorithm

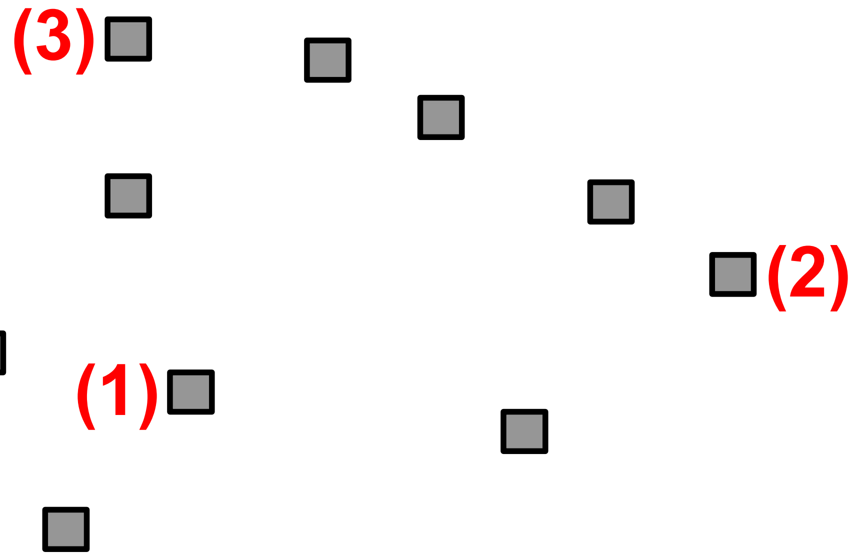
Assuming we know r: ($r = r^* = r(C^*)$ where C^* is the optimal selection)

procedure CENTER-SELECT-1
  // S' = sites still needing to be covered
  Init $S' = S$, $C = \emptyset$
  while $S' \neq \emptyset$ **do**
    Select any $s \in S'$ and add s to C
    Delete all $t \in S'$ where $\mathrm{dist}(t, s) \leq 2r$
  end while
  if $|C| \leq k$ **then**
    Return C as the selected set of sites
  else
    Claim there is no set of k centers with covering radius at most r
  end if
end procedure

Center Selection Algorithm (2-approximation)

(i) Select a site.

(ii) Iterate the following: Select a site with the largest distance from the selected sites.



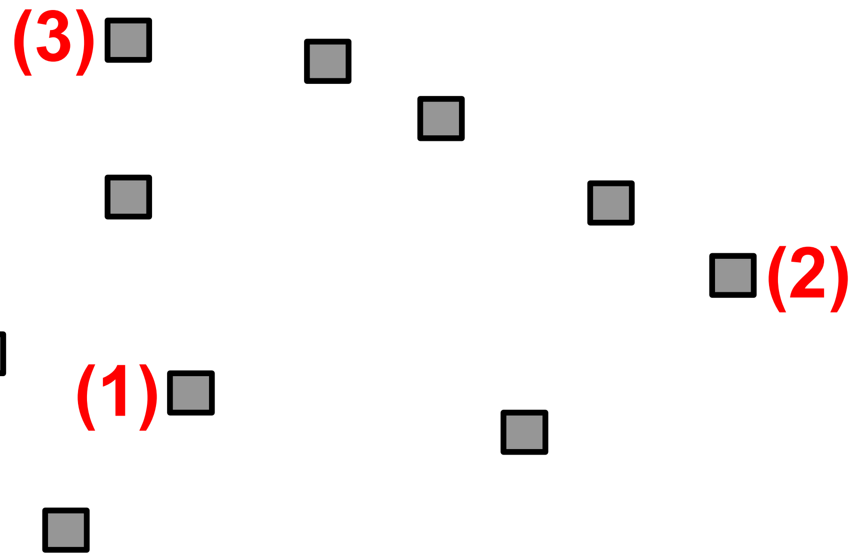
Center Selection Algorithm (2-approximation)

(i) Select a site.

(ii) Iterate the following: Select a site with the largest distance from the selected sites.

Idea behind this algorithm (Proof of 2-Approximation)

If the largest distance from the selected sites is larger than $2r^*$, the selection of the site can be viewed as being the same as in the virtual algorithm. Thus all sites will be covered when k sites are selected (thus 2-approximation). If it is not larger then $2r^*$, all sites have already been covered by the selected sites within $2r^*$ (thus 2-approximation).



```
Greedy-Center-Selection(k, n, s1,s2,...,sn) {

   C = φ
   repeat k times {
      Select a site si with maximum dist(si, C)
      Add si to C
   }
   return C
}
```

site farthest from any center

Q: How to select an initial site ?

procedure CENTER-SELECT
  Assume $k \leq |S|$ (else define $C = S$)
  Select any site s and let $C = \{s\}$
  while $|C| < k$ **do**
    Select a site $s \in S$ that maximizes $\mathrm{dist}(s, C)$
    Add s to C
  end while
  Return C as the selected set of sites
end procedure

<u>Exercise 5-1</u>:

Create two examples where the obtained value $r(C)$ by the algorithm is close to $2r(C^*)$. Create another two examples where the obtained value $r(C)$ by the algorithm is close to $r(C^*)$. There is no constraint condition on the location of each center.

<u>Exercise 5-2</u>:

Create two examples where the obtained value $r(C)$ by the algorithm is close to $2r(C^*)$. Create another two examples where the obtained value $r(C)$ by the algorithm is close to $r(C^*)$. There is the following additional constraint condition: Each center should be selected from the given sites.

Exercise 5-3

Design a method to select the first site in the center selection algorithm (instead of random selection).

Q. Why do you create examples?
A. To clearly understand the search behavior of the algorithm. For example, we can explain the advantages (strength) and disadvantages (weakness) of the algorithm using examples.

Center Selection Algorithm:

Advantages: One site from each cluster is always selected when the number of centers is the same as the number of clusters.
Disadvantages: An extreme site (instead of a center site) in each cluster is usually selected.