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.

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

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

Center Selection Algorithm (2-approximation)

(i) Select a site.

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

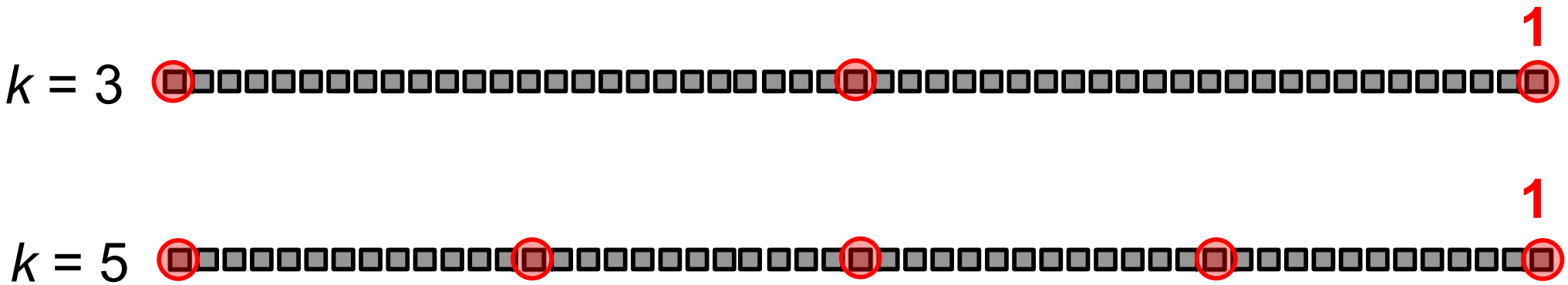
Observation: This algorithm tries to maximize the minimum distance between the selected sites (instead of minimizing the maximum distance from each site to the nearest center).

Center Selection Algorithm (2-approximation)

(i) Select a site.

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

Maximization of the minimum distance between the selected sites, which is different from the original objective function .

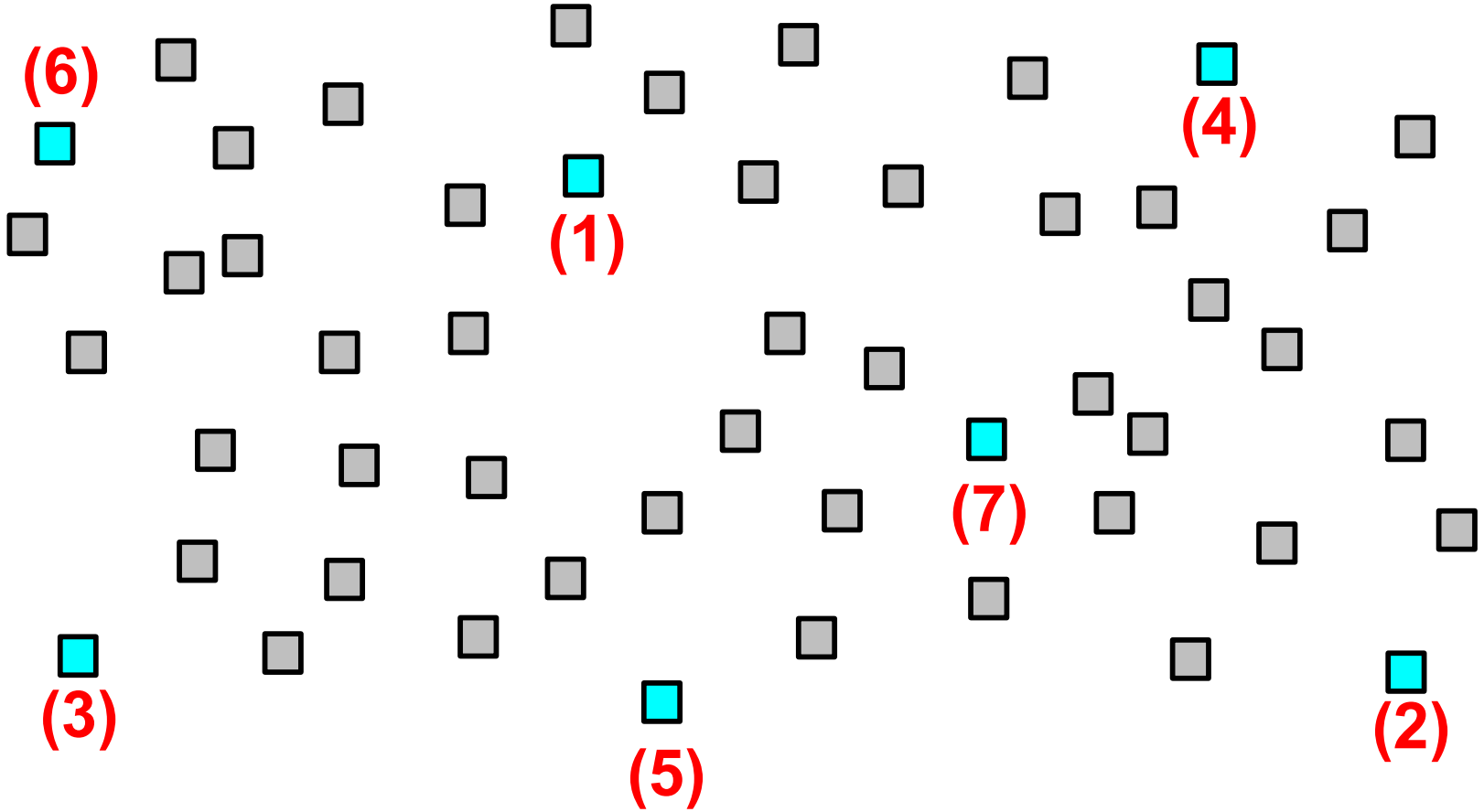


Center Selection Algorithm (2-approximation)

(i) Select a site.

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

Maximization of the minimum distance between the selected sites, which is different from the original objective function .

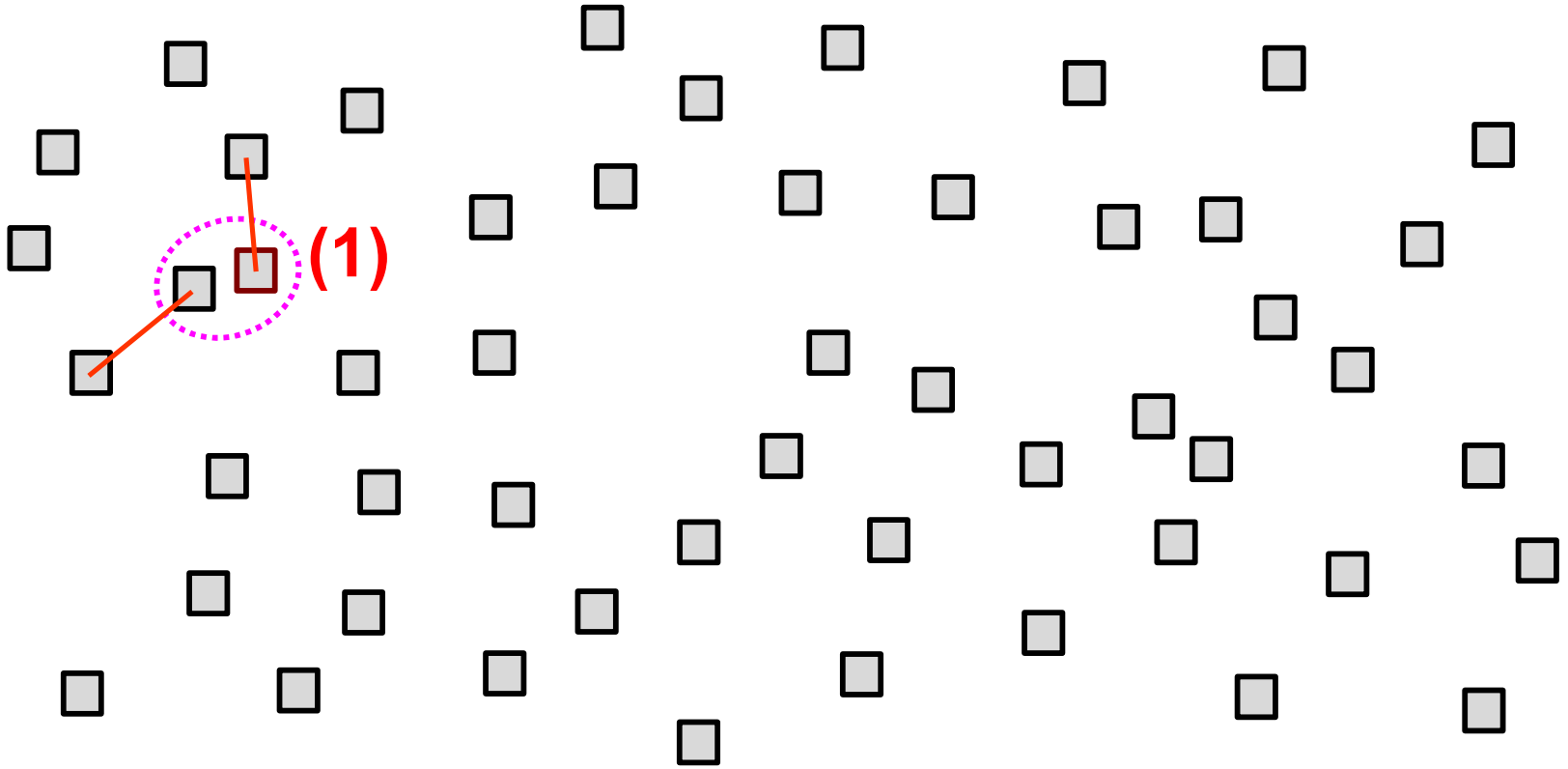


If our objective is to maximize the minimum distance between the selected sites, we can also use distance-based **greedy removal**:

Iterate the following two steps:

1. Select a pair of sites with the minimum distance among the remaining sites.

2. Remove the site with a shorter distance to the second neighbor from the selected pair.

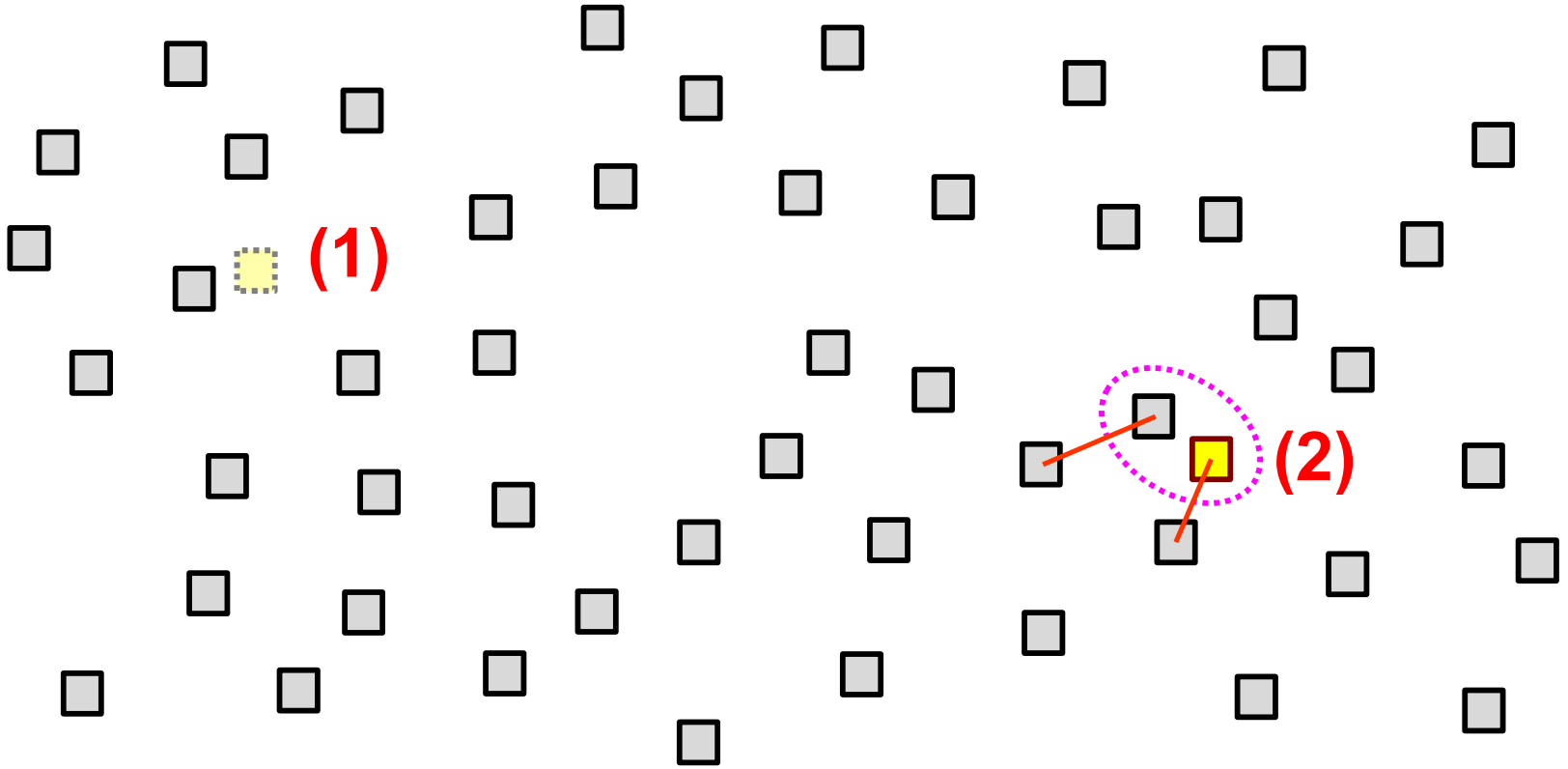


If our objective is to maximize the minimum distance between the selected sites, we can also use distance-based **greedy removal**:

Iterate the following two steps:

1. Select a pair of sites with the minimum distance among the remaining sites.

2. Remove the site with a shorter distance to the second neighbor from the selected pair.

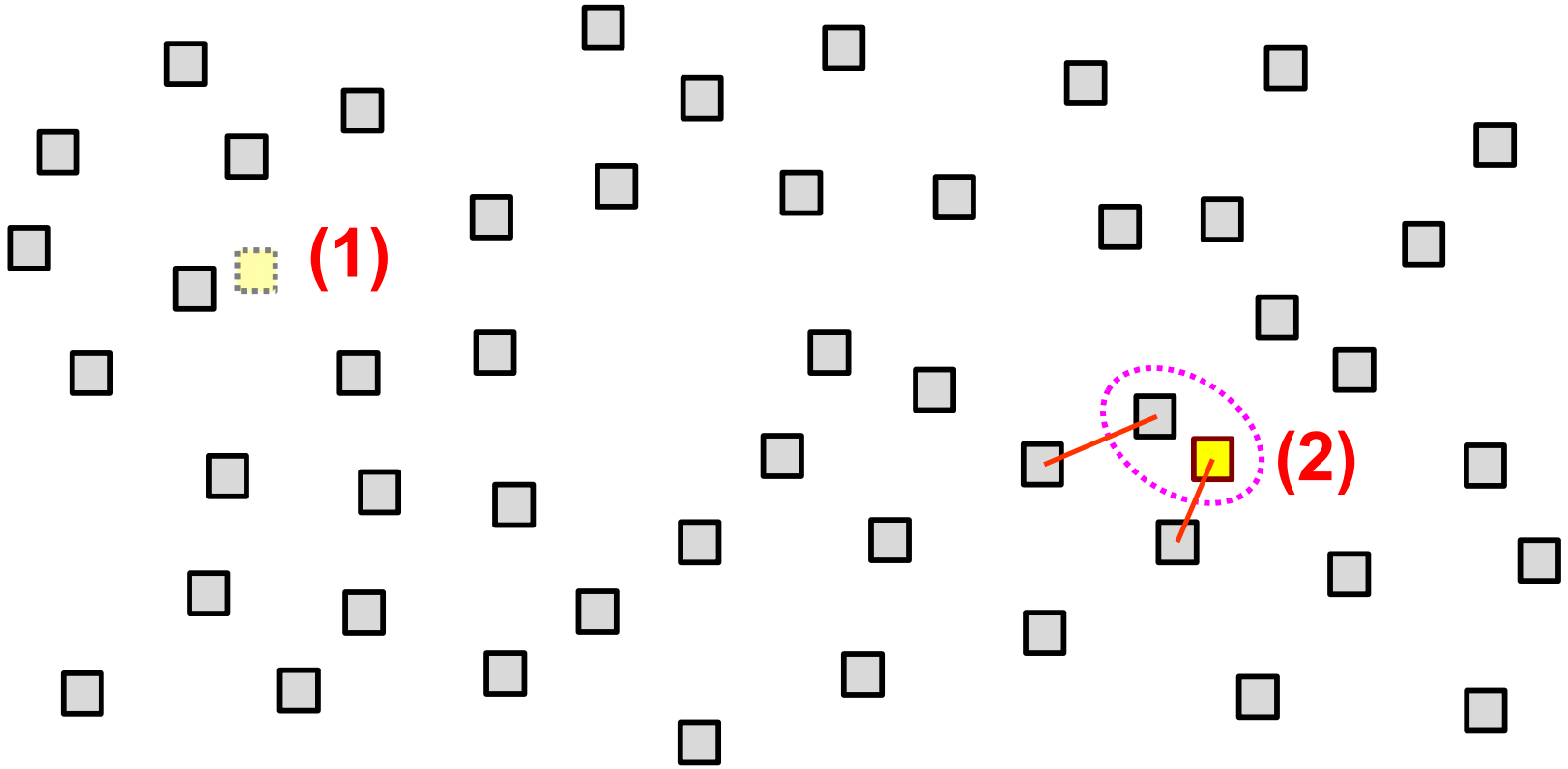


Distance-based greedy removal algorithm:

Iterate the following two steps:

1. Select a pair of sites with the minimum distance among the remaining sites.

2. Remove the site with a shorter distance to the second neighbor from the selected pair.



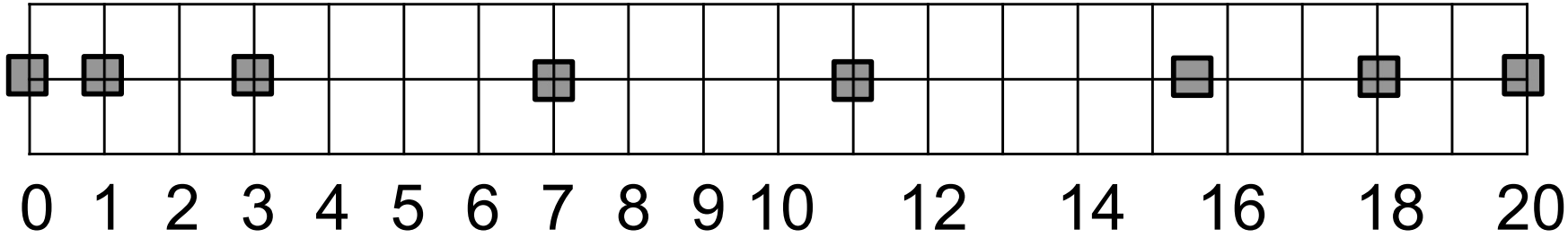
Distance-based greedy removal algorithm:

Iterate the following two steps:

1. Select a pair of sites with the minimum distance among the remaining sites.

2. Remove the site with a shorter distance to the second neighbor from the selected pair.

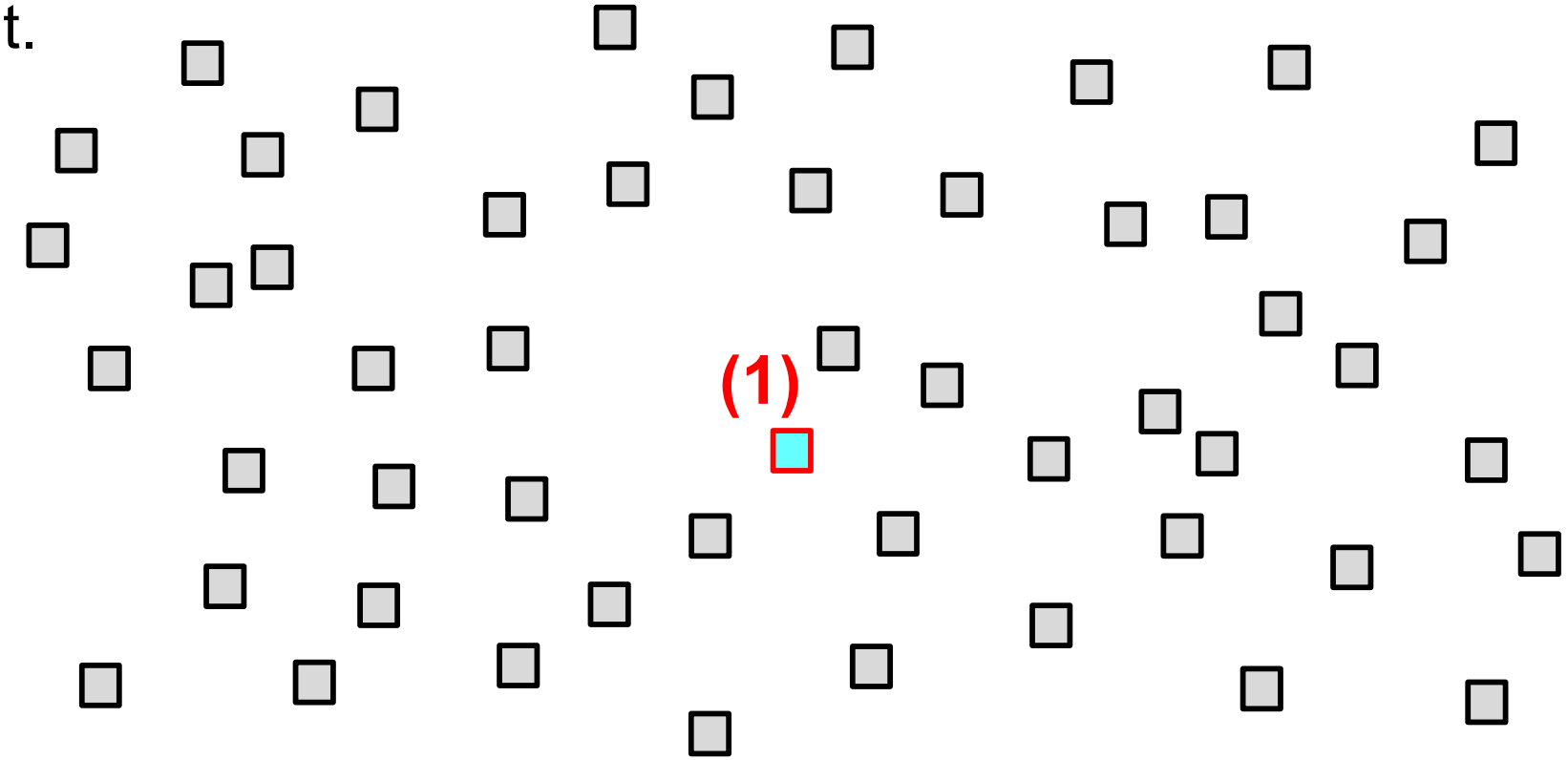
Example (k = 4):



Direct use of the original objective function: Minimization of the maximum distance from each site to the nearest center.

Greedy inclusion algorithm (with random tie-break):

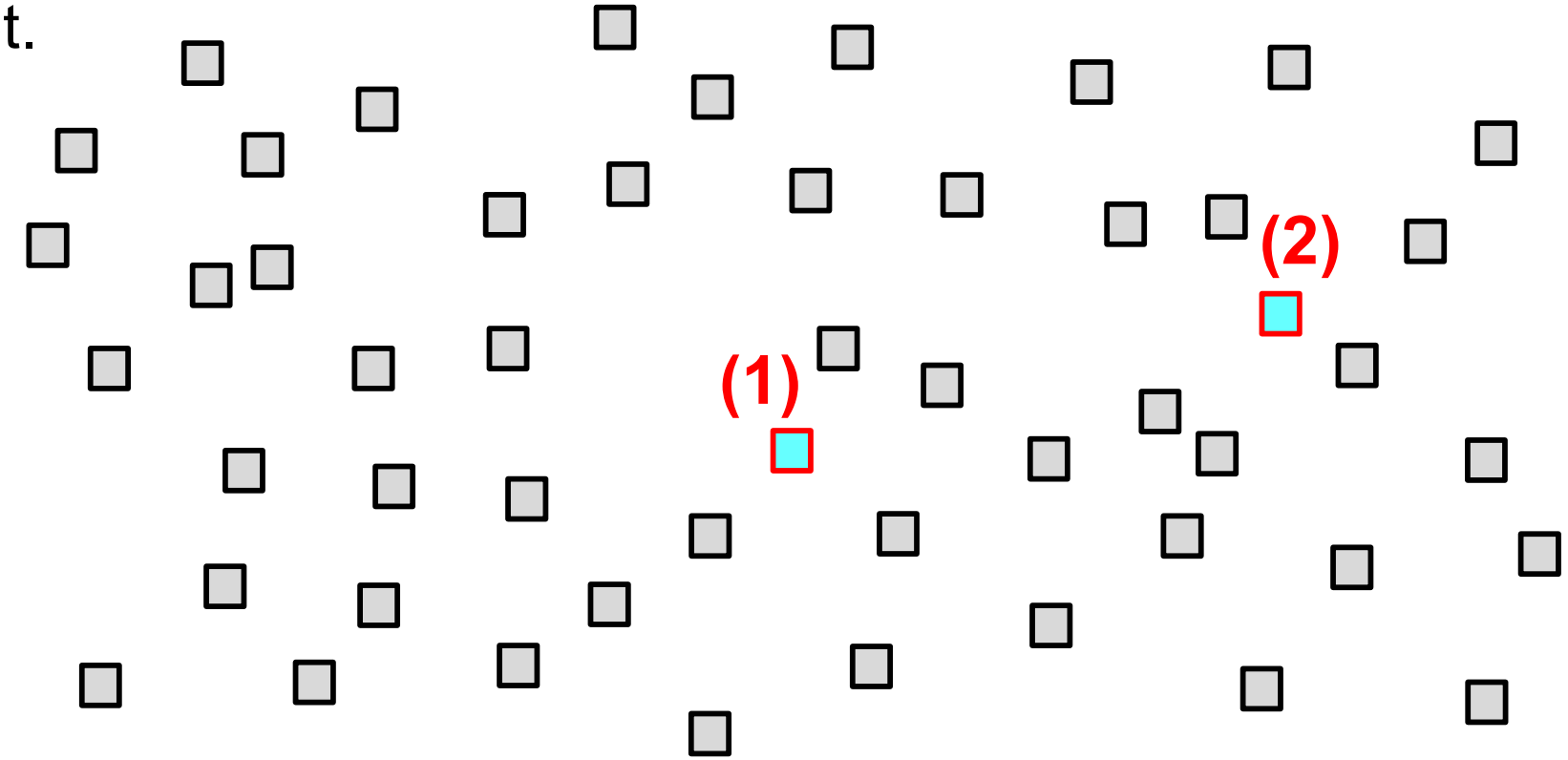
1. Select a single site as the first center (by examining all sites) to minimize the maximum distance from each site to the nearest center.
2. Iterated the following procedure: Select a single site as the next center (by examining all the remaining sites) to minimize the maximum distance from each site to the nearest center after adding it to the center set.



Direct use of the original objective function: Minimization of the maximum distance from each site to the nearest center.

Greedy inclusion algorithm (with random tie-break):

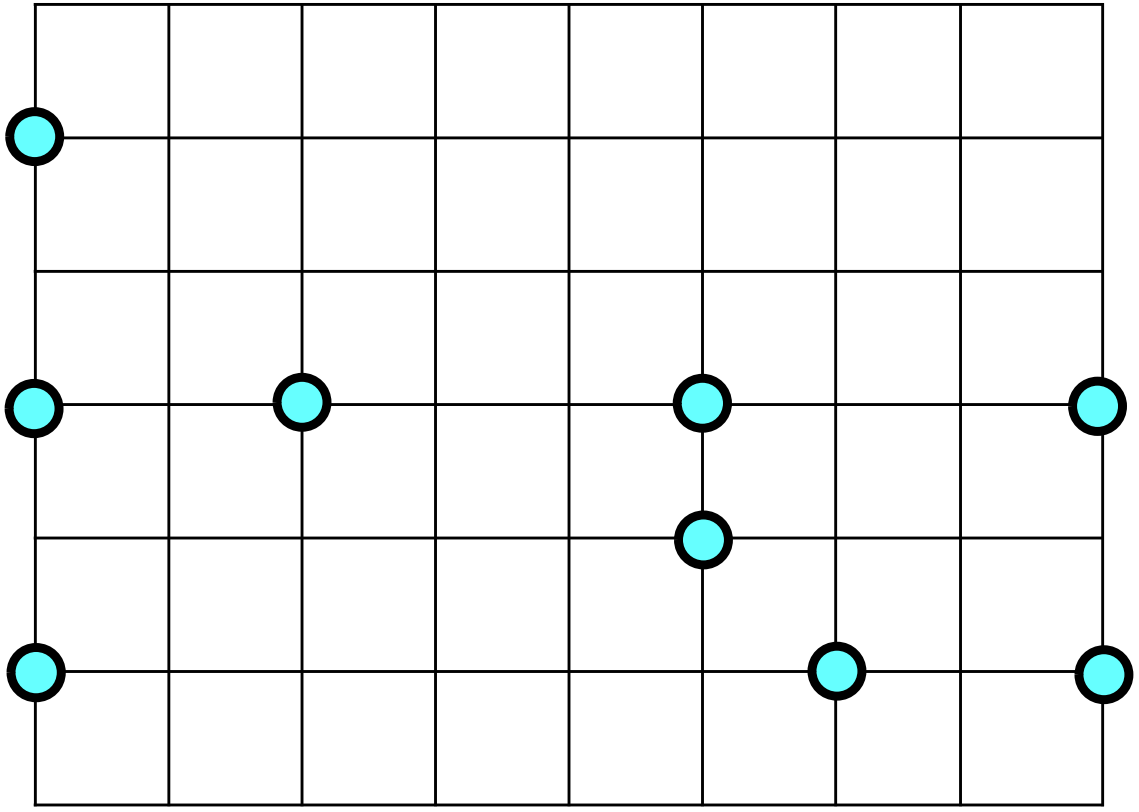
1. Select a single site as the first center (by examining all sites) to minimize the maximum distance from each site to the nearest center.
2. Iterated the following procedure: Select a single site as the next center (by examining all the remaining sites) to minimize the maximum distance from each site to the nearest center after adding it to the center set.



Greedy inclusion algorithm based on the original objective:

1. Select a single site as the first center (by examining all sites) to minimize the maximum distance from each site to the nearest center.
2. Iterated the following procedure: Select a single site as the next center (by examining all the remaining sites) to minimize the maximum distance from each site to the nearest center after adding it to the center set.

Example (k = 3):



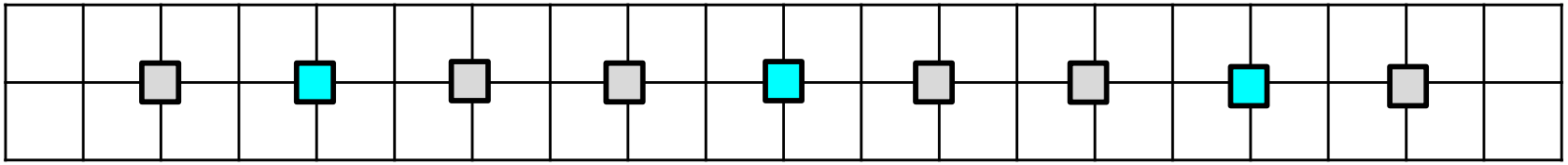
Greedy inclusion algorithm based on the original objective:

1. Select a single site as the first center (by examining all sites) to minimize the maximum distance from each site to the nearest center.
2. Iterated the following procedure: Select a single site as the next center (by examining all the remaining sites) to minimize the maximum distance from each site to the nearest center after adding it to the center set.

Difficulties: In each iteration of the second step, there are many sites with the same best evaluation.

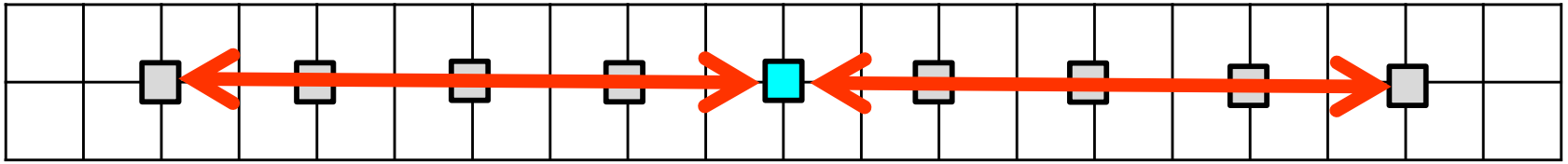
Example: Selection of 3 sites from the given 9 sites.

Best solution:

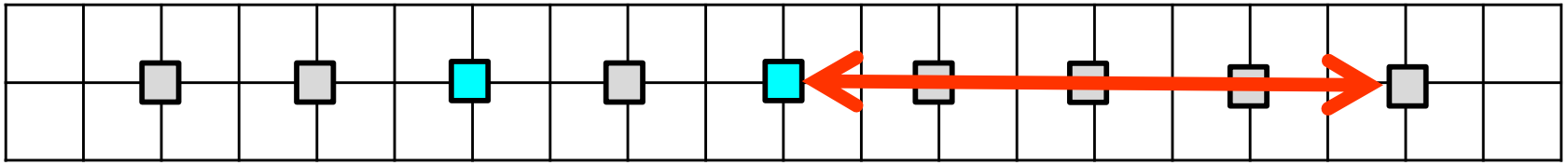


Greedy inclusion:

First step:

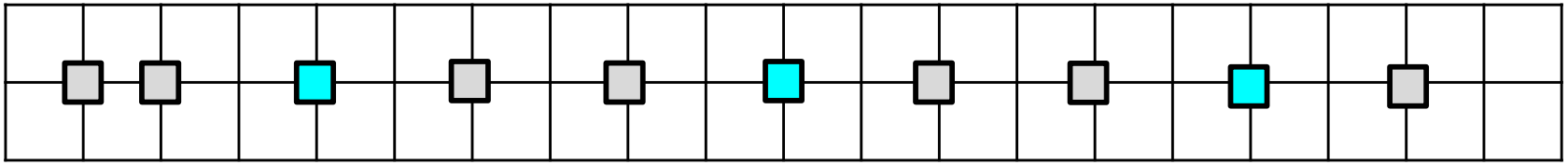


Second step: **No solution can improve the objective function value.**



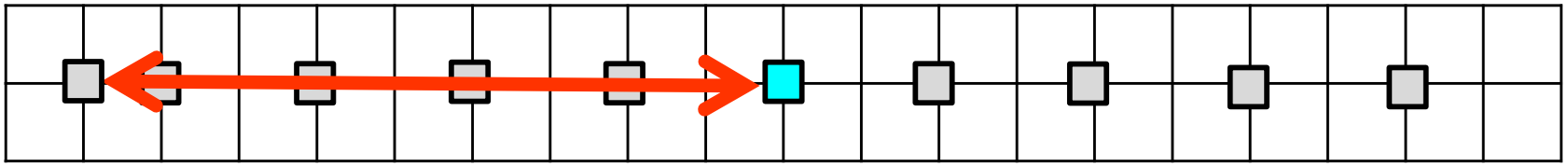
Another Example: Selection of 3 sites from the given 10 sites.

Best solution:

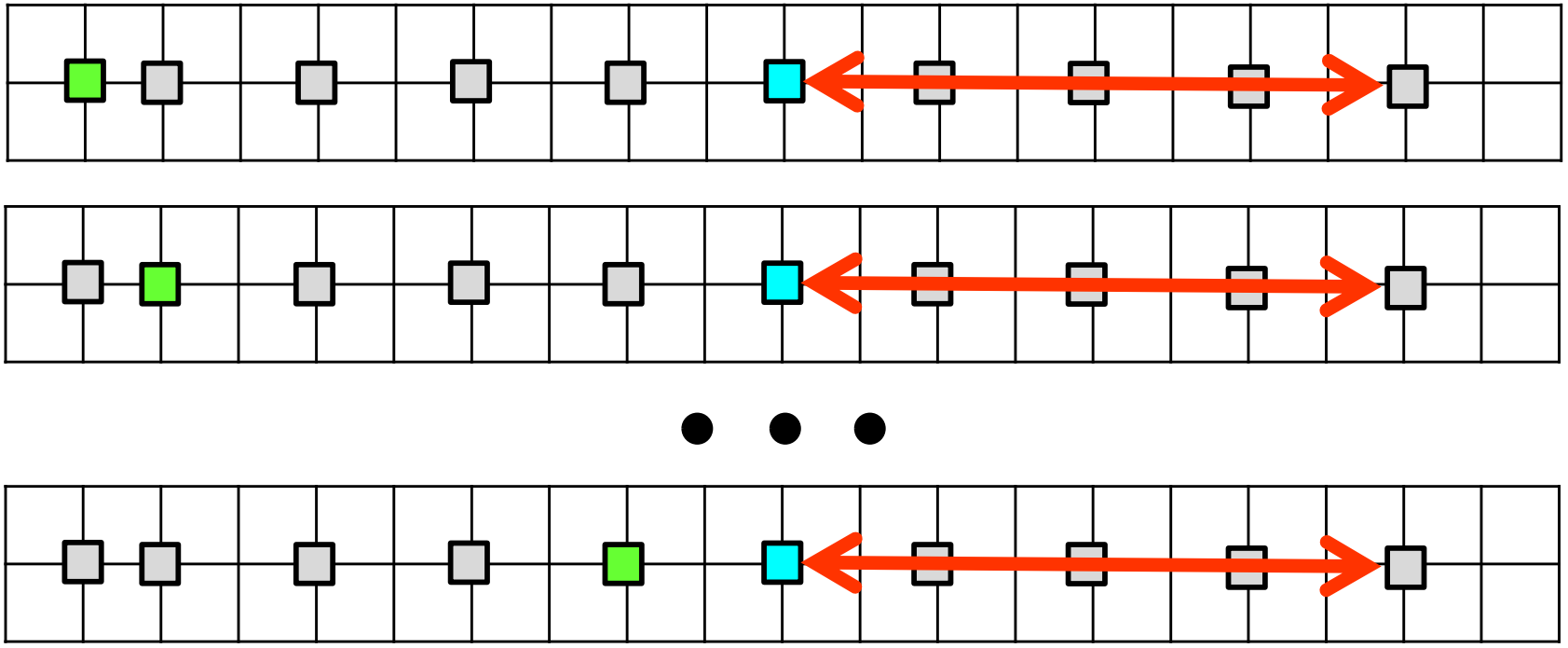


Greedy inclusion:

First step:



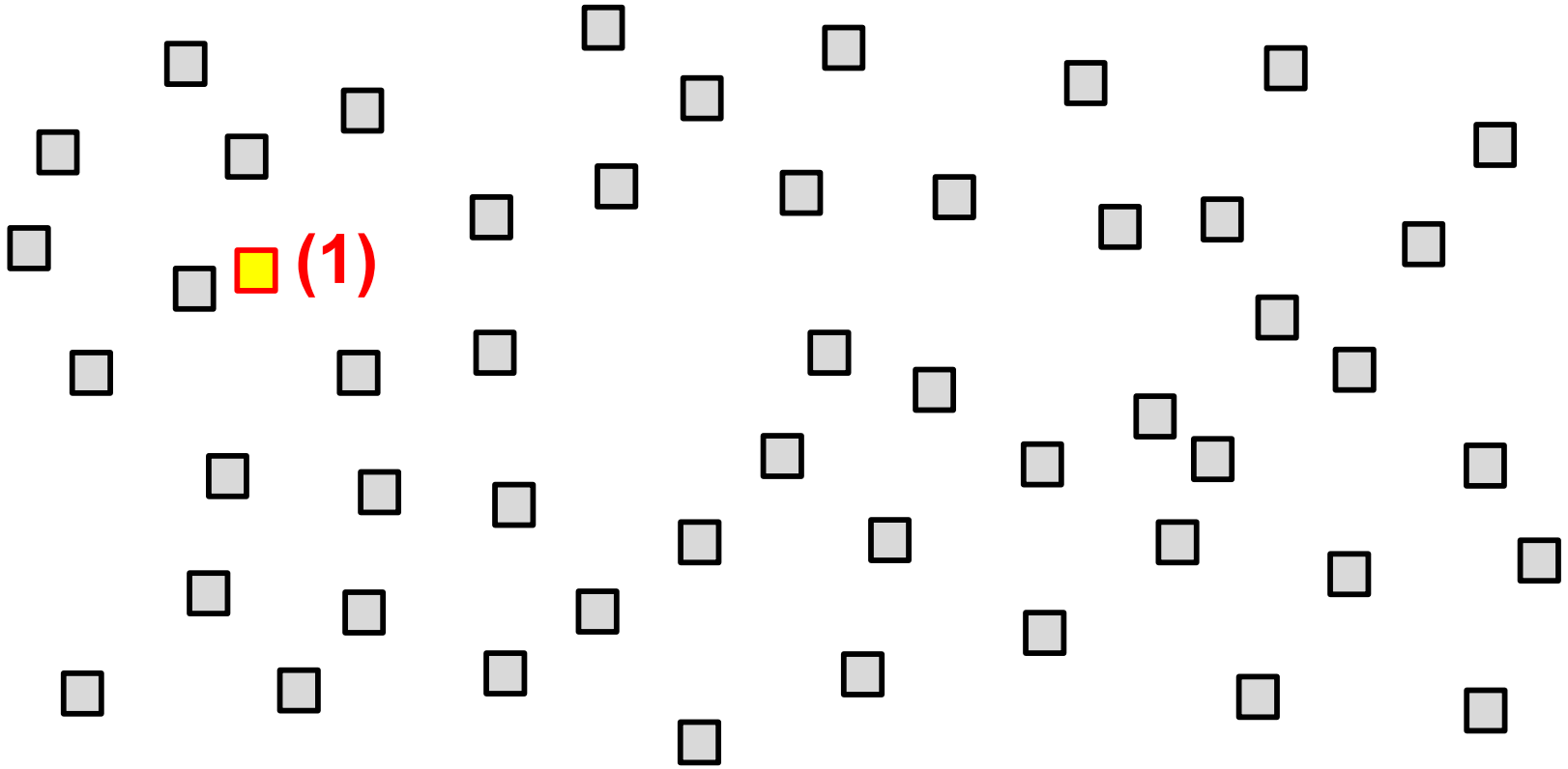
Second step: **Five solutions have the same evaluation.**



Direct use of the original objective function: Minimization of the maximum distance from each site to the nearest center.

Greedy removal algorithm (with random tie-break):

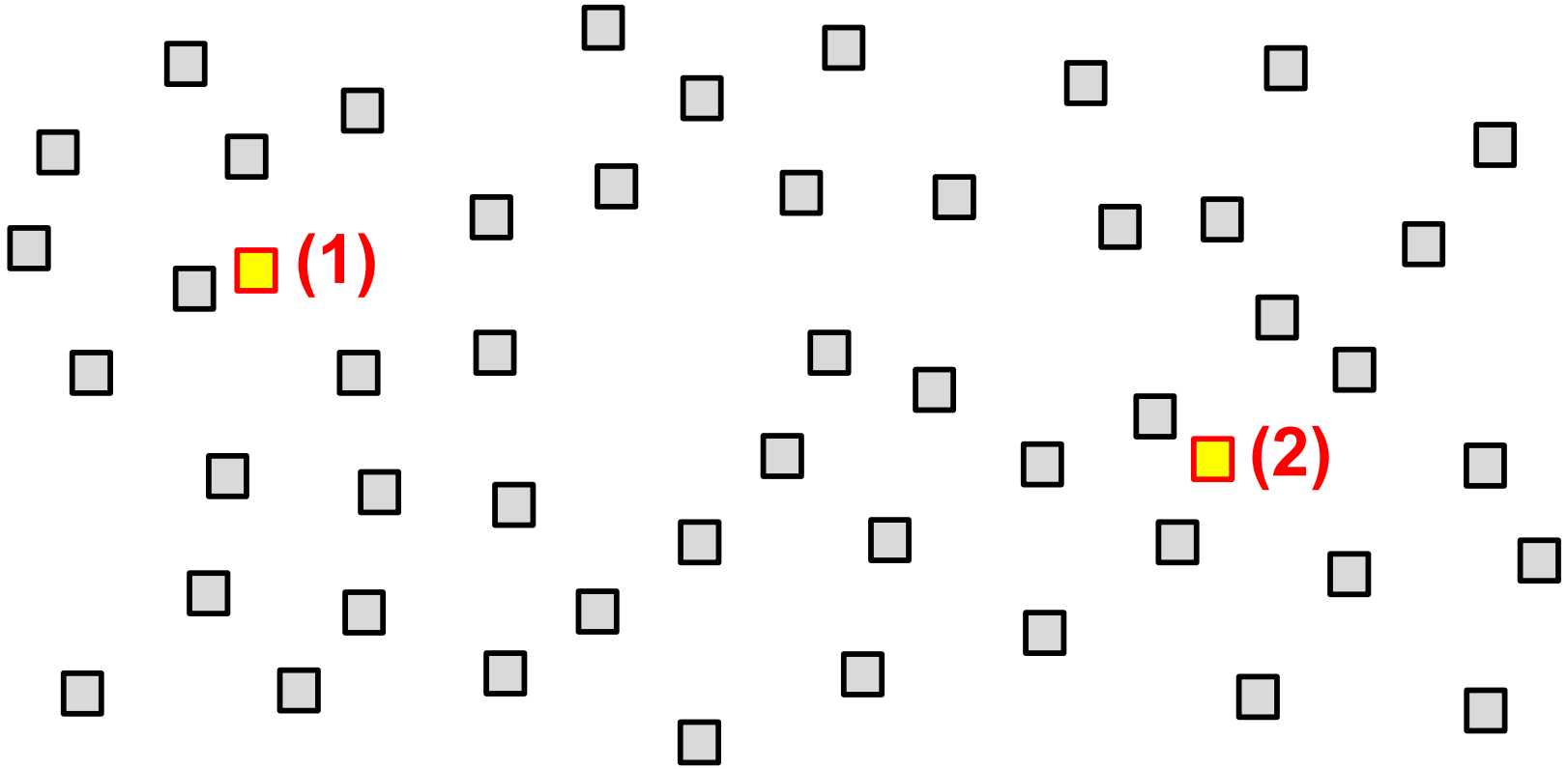
1. Let all cites be centers (i.e., the initial center set *C* is the site set *S*).
2. Iterated the following procedure: Select a single site to be removed from *C* (by examining all sites in *C*) to minimize the maximum distance from each site to the nearest center after removing it from *C*.



Direct use of the original objective function: Minimization of the maximum distance from each site to the nearest center.

Greedy removal algorithm (with random tie-break):

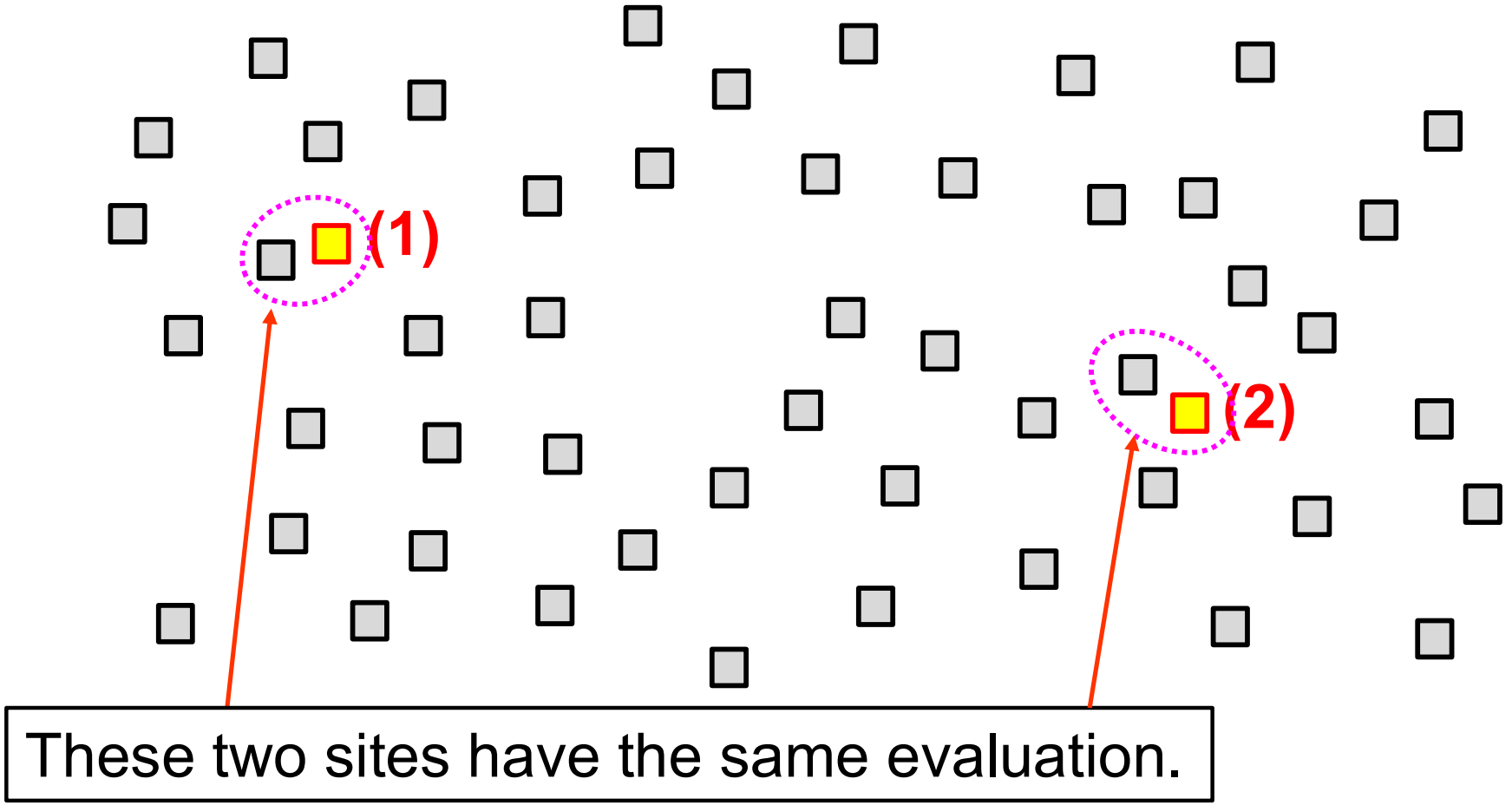
1. Let all cites be centers (i.e., the initial center set *C* is the site set *S*).
2. Iterated the following procedure: Select a single site to be removed from *C* (by examining all sites in *C*) to minimize the maximum distance from each site to the nearest center after removing it from *C*.



Greedy removal algorithm based on the original objective:

1. Let all cites be centers (i.e., the initial center set *C* is the site set *S*).
2. Iterated the following procedure: Select a single site to be removed from *C* (by examining all sites in *C*) to minimize the maximum distance from each site to the nearest center after removing it from *C*.

Difficulties: In each iteration, there are at least two sites with the same best evaluation.



Exercise 6-1:

For the center selection problem (where centers should be selected from the given sites), we have the following four algorithms.

1. **Center selection algorithm (i.e., Distance-based greedy inclusion algorithm).**
2. **Distance-based greedy removal algorithm.**
3. **Greedy inclusion algorithm based on the original objective function.**
4. **Greedy removal algorithm based on the original objective function.**

Create a simple example to clearly explain the characteristic features of each algorithm and also to clearly explain the differences among them.

Exercise 6-2:

Create another interesting example to clearly explain the characteristic features of each of the four algorithms and also to clearly explain the differences among them.

Similar Problem:

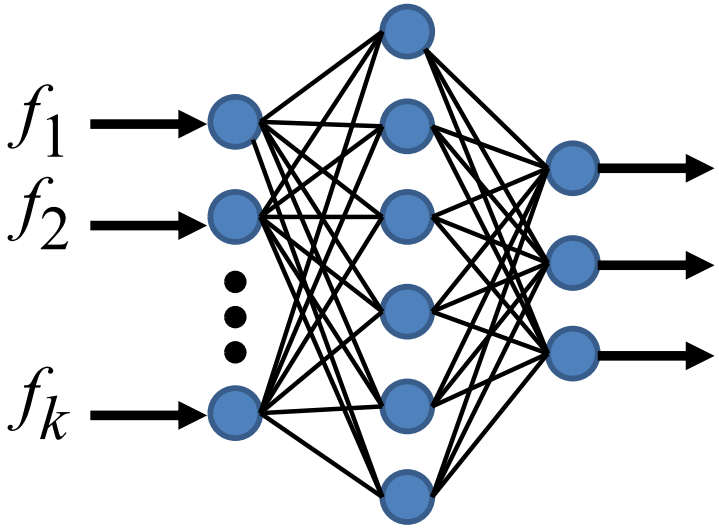
Feature Selection for Pattern Classification.

Set of *n* candidate features

$$G = \{g_1, g_2, \ldots, g_n\}$$

Set of *k* selected features

$$F = \{f_1, f_2, \ldots, f_k\} \subset G$$



Greedy Algorithms

- Greedy inclusion
- Greedy removal (exclusion)

Local search

Step 1: Generate an initial feature set *F* (|*F*| = *k*).

Step 2: Iterate the following.

Add *s* features to *F* and remove the least contributed *s* features from *F*.

(*s* is a parameter to determine the neighborhood size)

Population based approach: A singe individual is a feature set *F*

Binary coding approach: **1**000**11**000 (k = 3: f_1, f_5 and f_6 are selected)

- Single-objective algorithms
- Multi-objective algorithms

Permutation coding approach: **516**472839 (k = 3)

- Single-objective algorithms
- Multi-objective algorithms

Neural network-based reinforcement learning approach

- Permutation-based single-objective algorithms

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

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

Three Formulations

1. Minimization of the maximum distance from each site to the nearest center.

$$\text{Minimize} \ \max_{s \in S} dist(s, C)$$

2. Minimization of the total squared distance from each site to the nearest center.

$$\text{Minimize} \ \sum_{s \in S} dist(s, C)^2$$

3. Minimization of the total distance from each site to the nearest center.

$$\text{Minimize} \ \sum_{s \in S} dist(s, C)$$

No additional conditions such as $c_j \in S, \ j = 1, 2, ..., k$ are included.

Exercise 6-3:

For the center selection problem (with no additional constraint condition), we have the following three formulations.

(1) Minimization of the maximum distance from each site to the nearest center.

$$\text{Minimize} \ \max_{s \in S} dist(s, C)$$

(2) Minimization of the total squared distance from each site to the nearest center

$$\text{Minimize} \ \sum_{s \in S} dist(s, C)^2$$

(3) Minimization of the total distance from each site to the nearest center

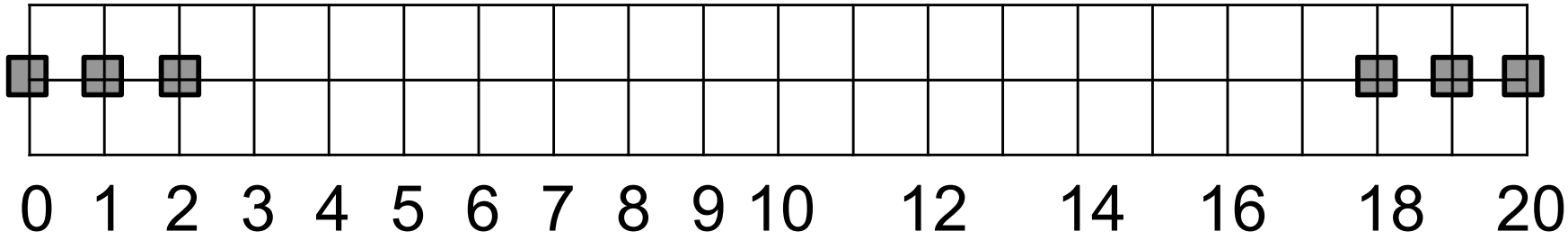
$$\text{Minimize} \ \sum_{s \in S} dist(s, C)$$

Create a simple example to clearly explain the characteristic features of each formulation and also to clearly explain the differences among them.

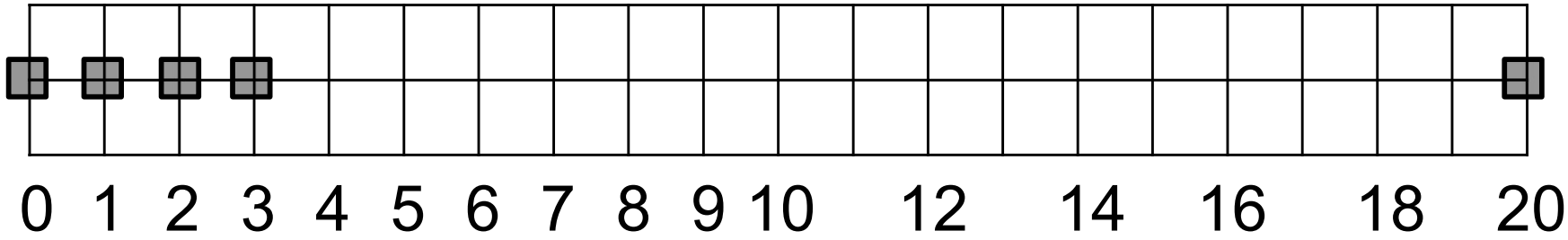
Exercise 6-4:

Create another interesting example to clearly explain the characteristic features of each of the three formulations and also to clearly explain the differences among them.

Example 1 (*k* = 1):



Example 2 (k = 1):



Example 3 ($k = 2$):