Multi-Swap K-means++

Lorenzo Beretta (University of Copenhagen)
Vincent Cohen-Addad (Google Research)
Silvio Lattanzi (Google Research)
Nikos Parotsidis (Google Research)

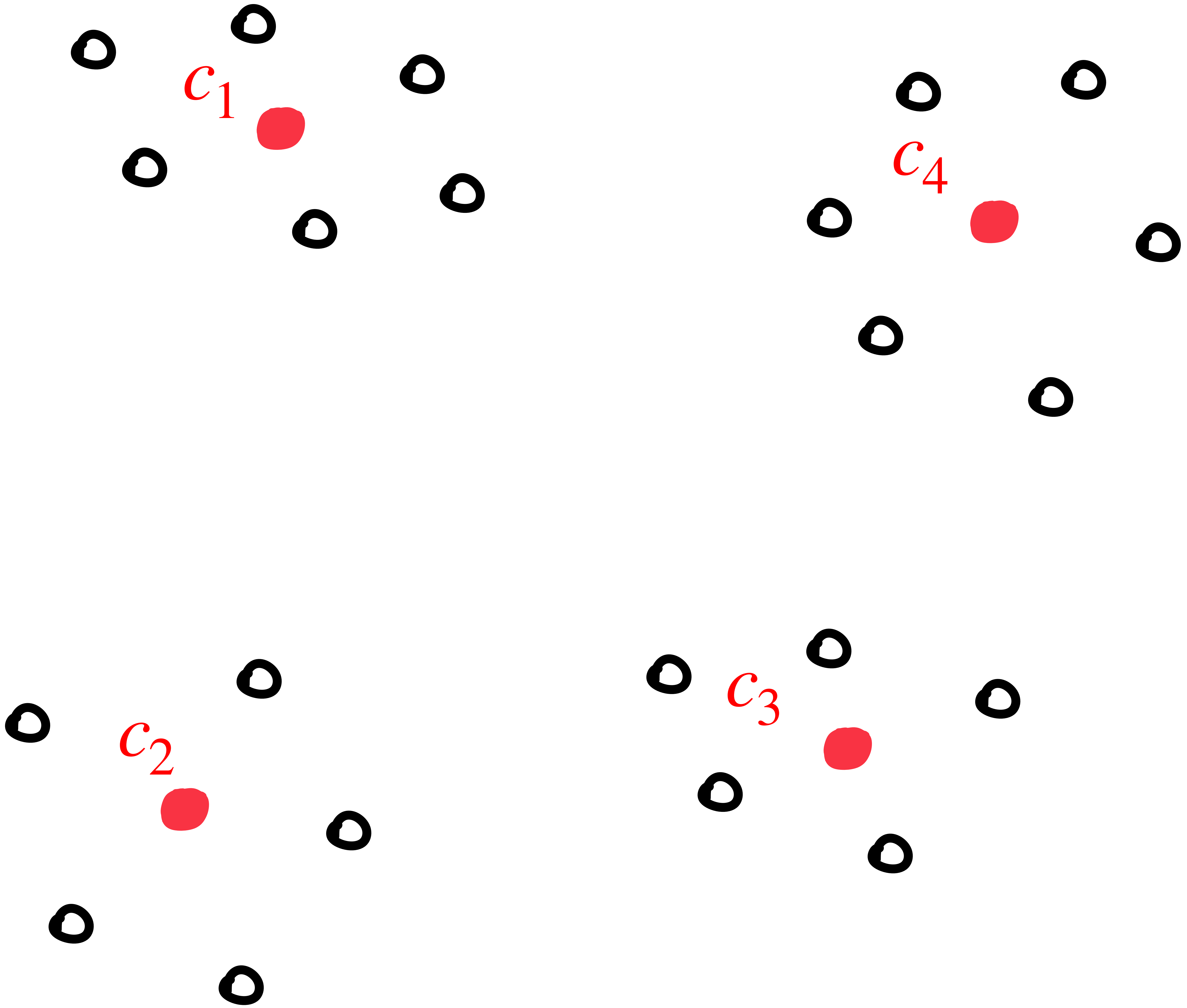
K-means

Input: $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$

Output: $c_1, c_2, \ldots, c_k \in \mathbb{R}^d$

that minimize

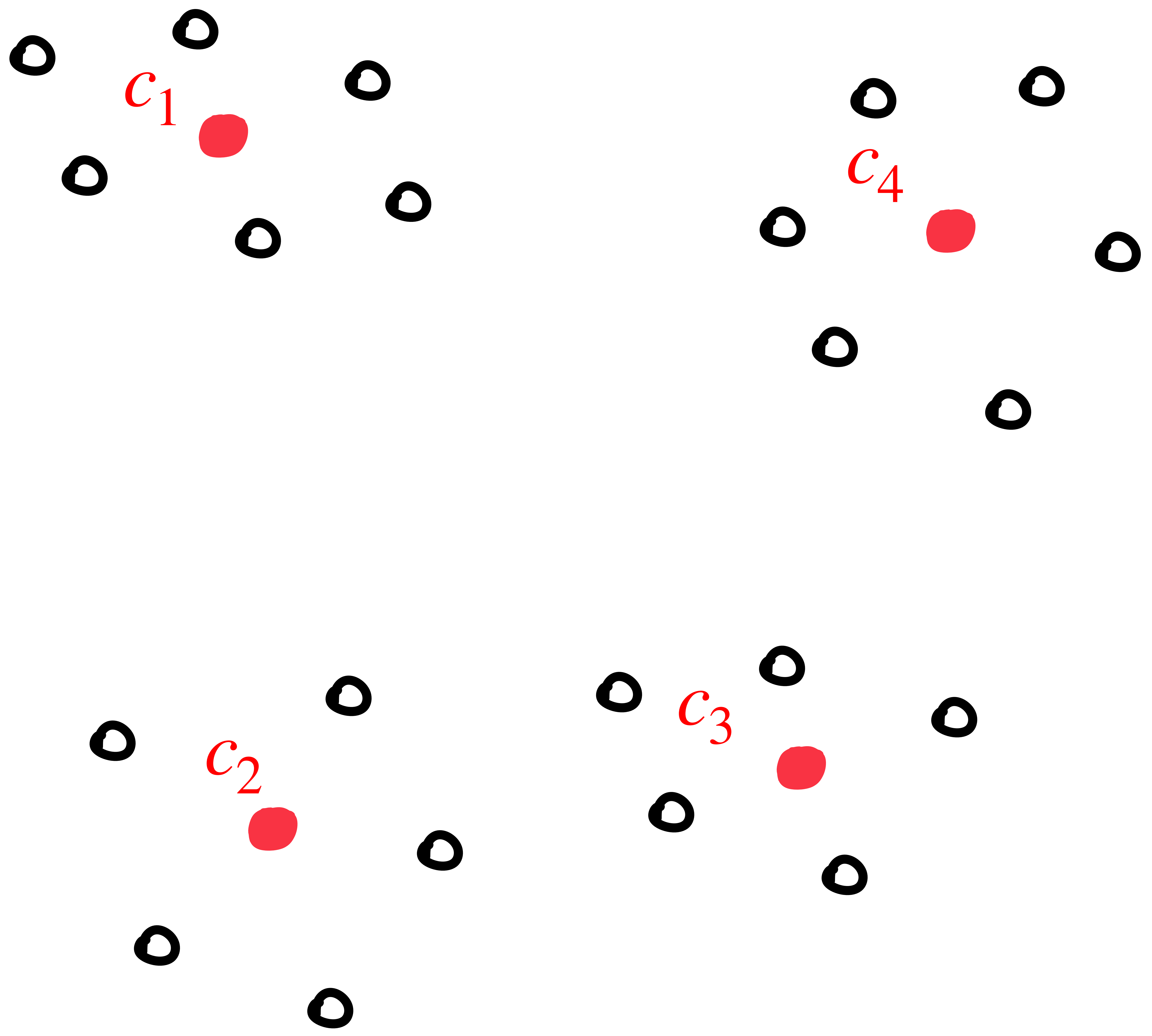
$$\sum_{i=1}^{n} \min_{j=1\ldots k} ||x_i - c_j||_2^2$$



K-means

Output: $c_1, c_2, \ldots, c_k \in \mathbb{R}^d$
that minimize

$$\sum_{i=1}^{n} \min_{j=1\ldots k} ||x_i - c_j||_2^2$$



K-means

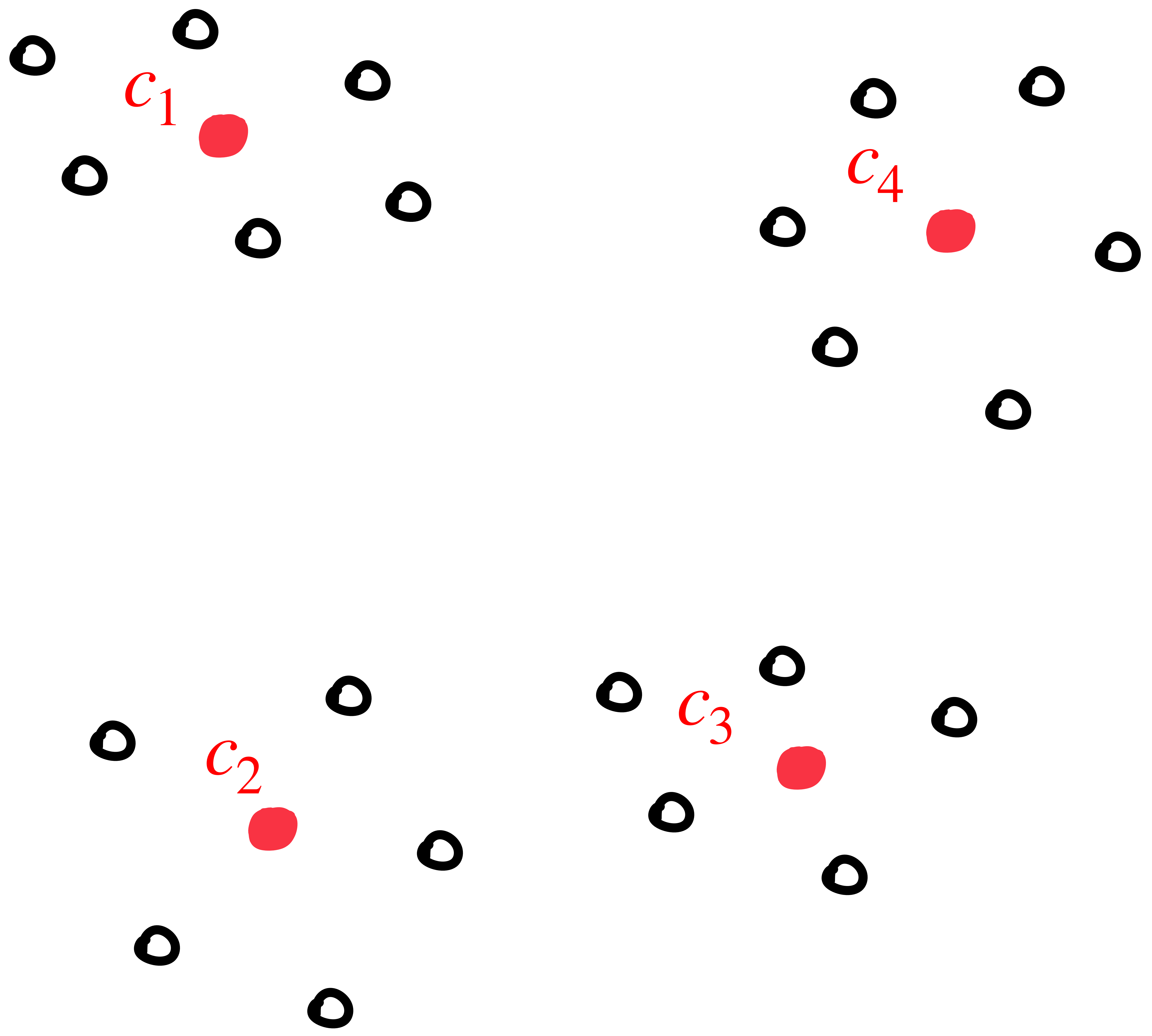
Output: $c_1, c_2, \ldots, c_k \in \mathbb{R}^d$
that minimize
$$\sum_{i=1}^{n} \min_{j=1\ldots k} ||x_i - c_j||_2^2$$

Lloyd's Algorithm

maintain $c_1 \ldots c_k$ and alternate between

1) $C_j \leftarrow \{x_i \text{ captured by } c_j\}$ for each j

2) $c_j \leftarrow mean(C_j)$ for each j



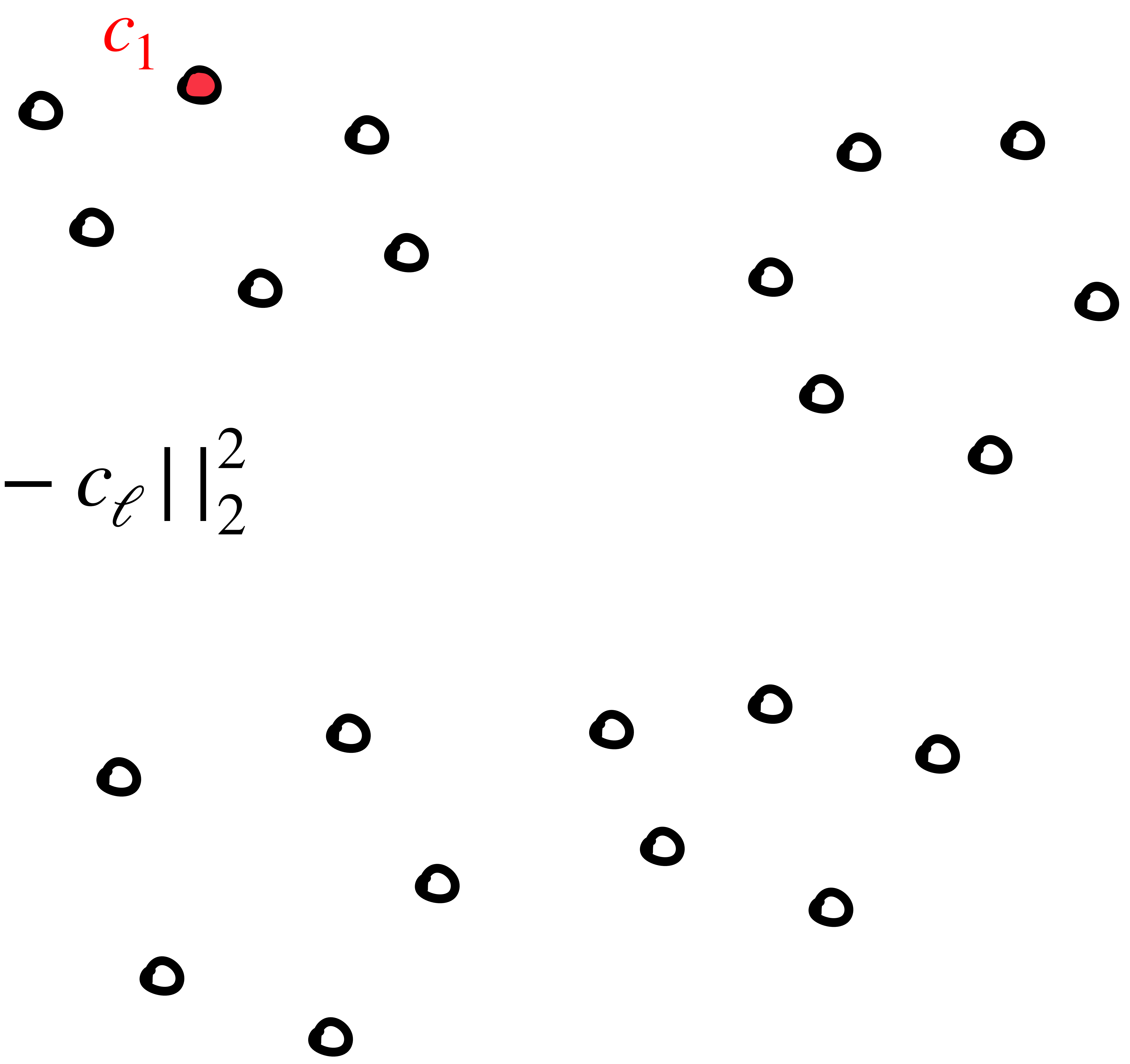
K-means++ [AV07]

Seeding Strategy

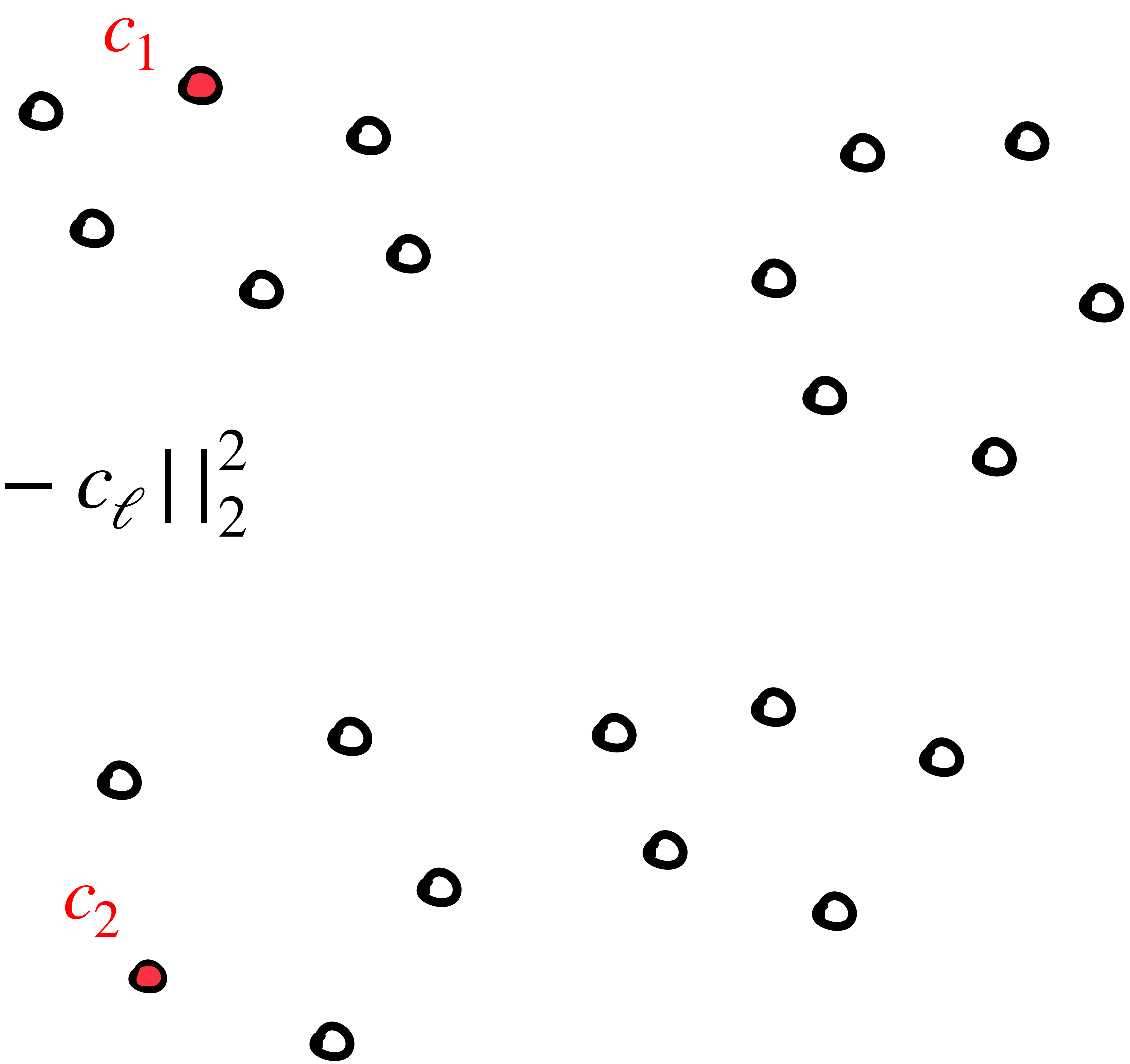
For $j = 1 \ldots k$:

Sample x_i proportionally to $\min_{\ell < j} ||x_i - c_\ell||_2^2$

Set $c_j \leftarrow x_i$



K-means++ [AV07]



Seeding Strategy

For $j = 1 \ldots k$:

Sample x_i proportionally to $\min_{\ell < j} ||x_i - c_\ell||_2^2$

Set $c_j \leftarrow x_i$

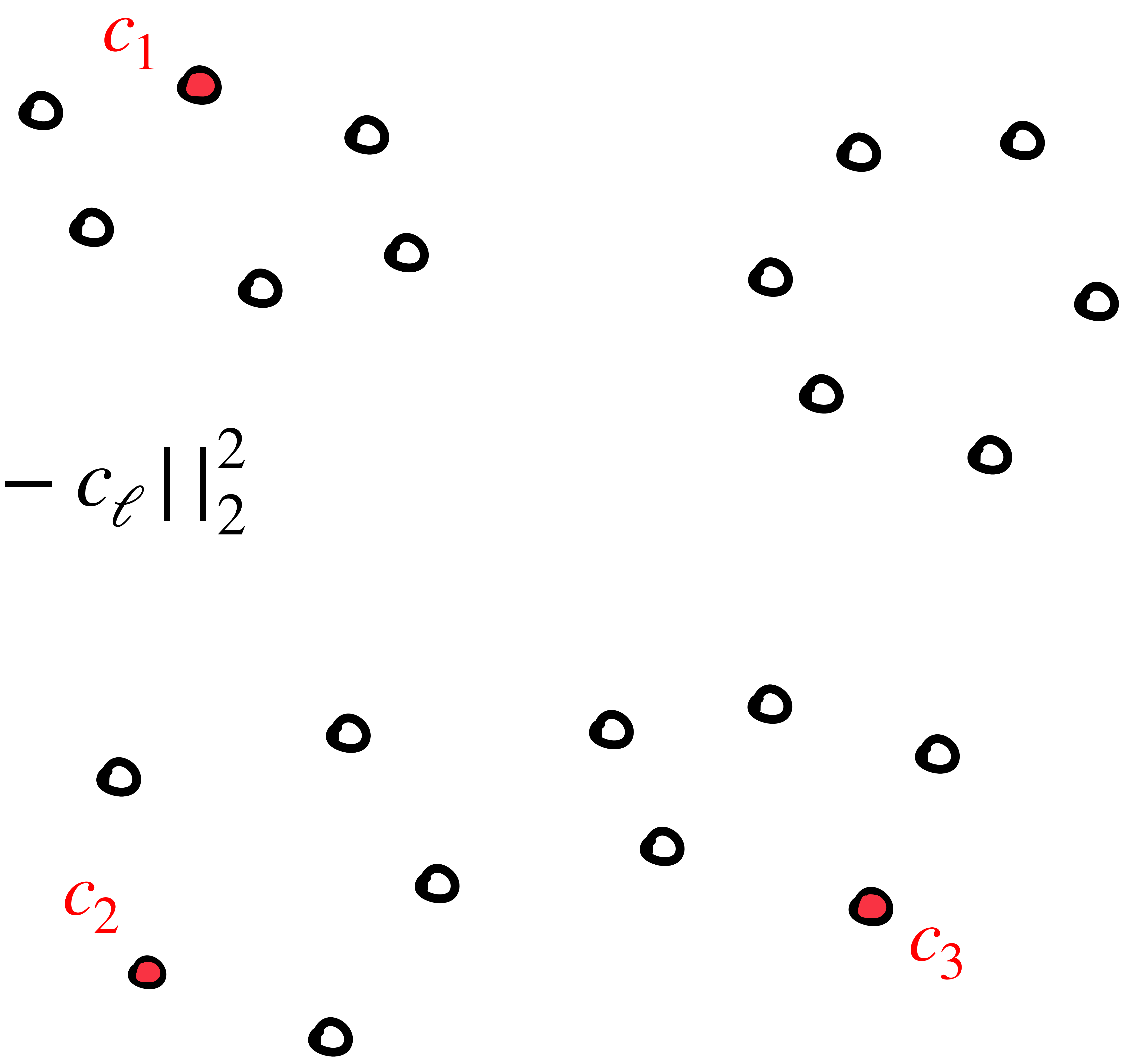
K-means++ [AV07]

Seeding Strategy

For $j = 1 \ldots k$:

Sample x_i proportionally to $\min_{\ell < j} ||x_i - c_\ell||_2^2$

Set $c_j \leftarrow x_i$



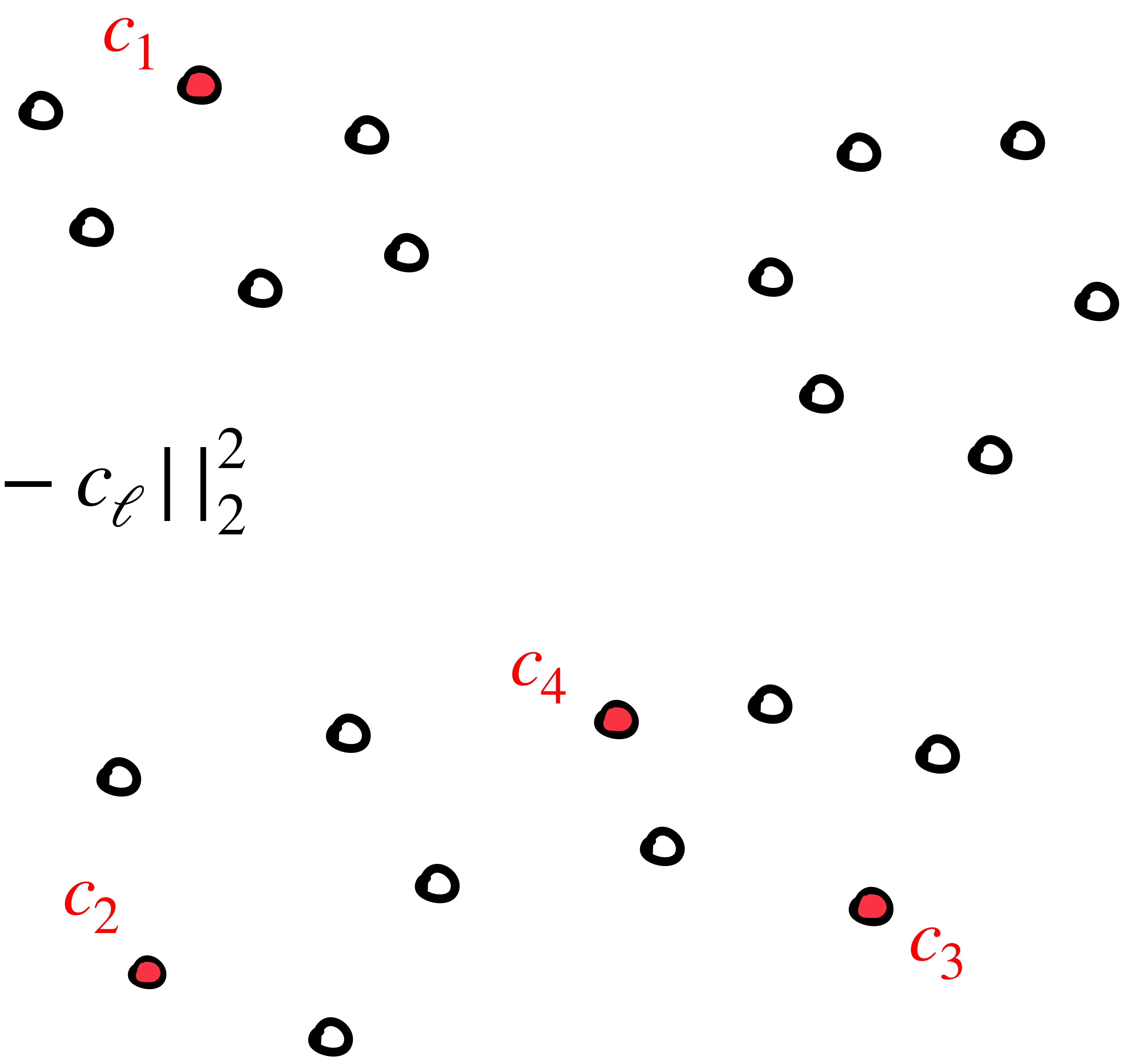
K-means++ [AV07]

Seeding Strategy

For $j = 1 \ldots k$:

Sample x_i proportionally to $\min_{\ell < j} ||x_i - c_\ell||_2^2$

Set $c_j \leftarrow x_i$



K-means++ [AV07]

Seeding Strategy

For $j = 1 \ldots k$:

Sample x_i proportionally to $\min_{\ell < j} ||x_i - c_\ell||_2^2$

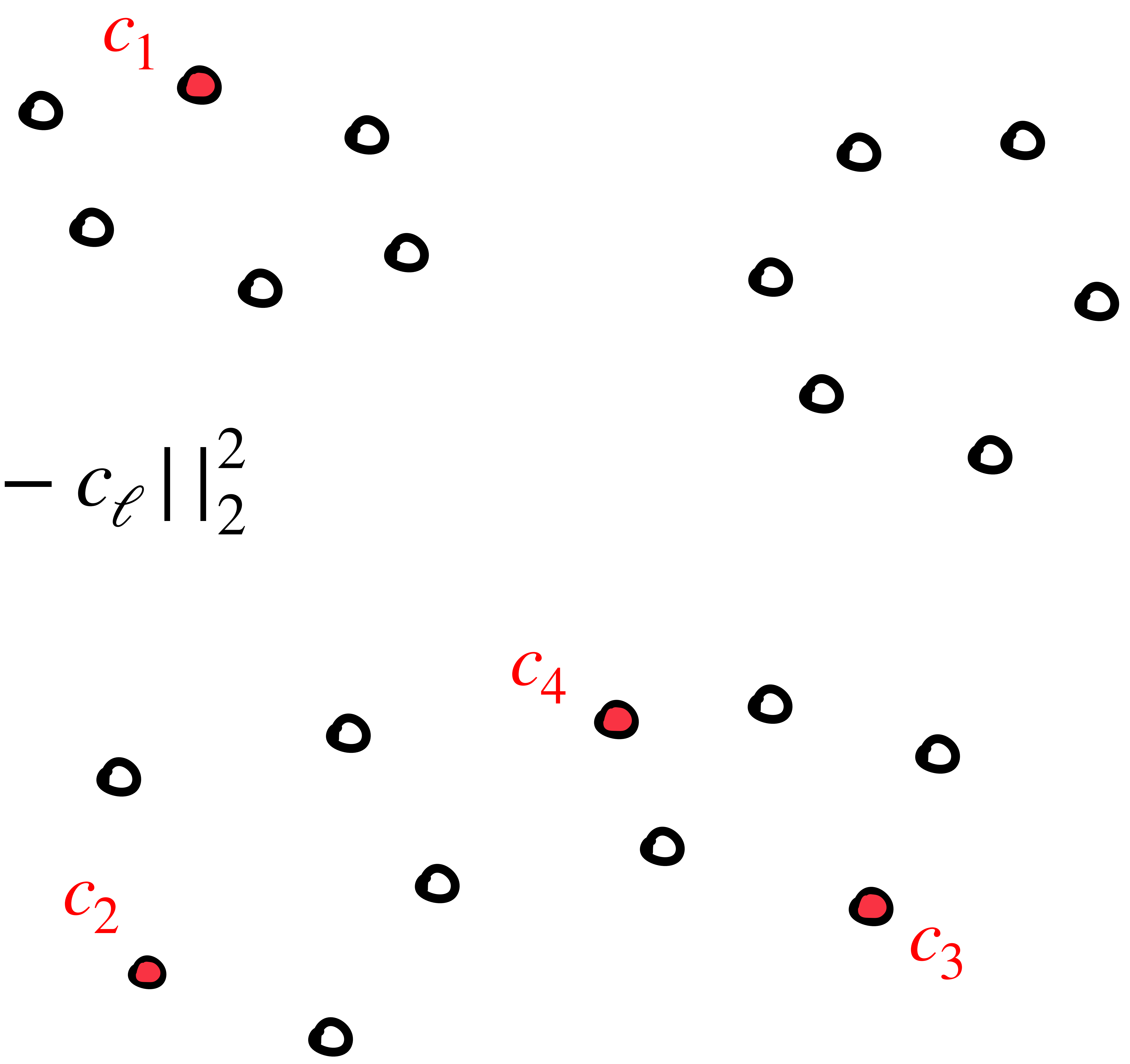
Set $c_j \leftarrow x_i$

Lloyd's Algorithm

Only improves current solution.

K-means++ seeding

$O(\log k)$-approximation!



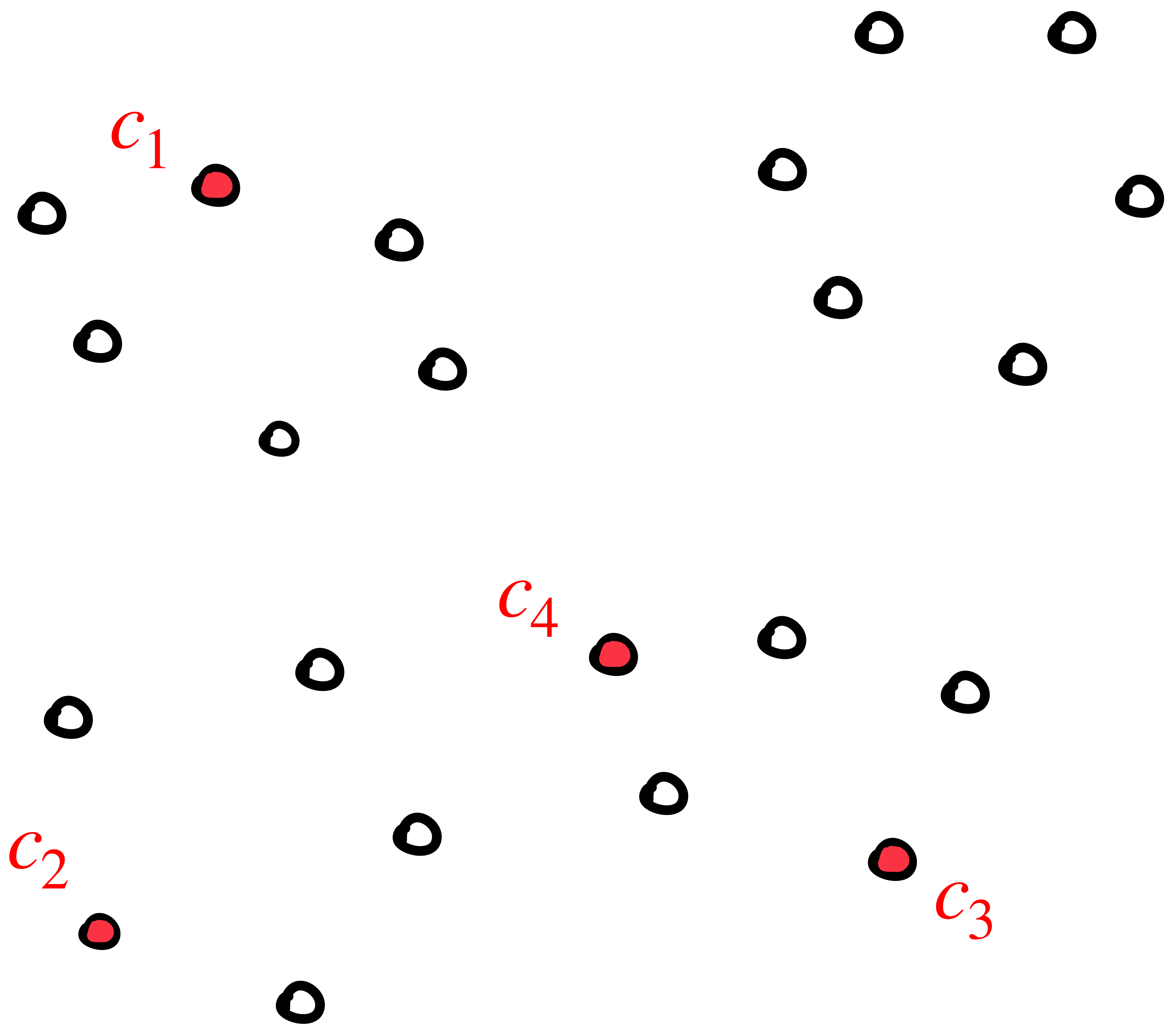
Single-Swap K-means++ [LS19]

Lloyd's Algorithm

Only improves current solution.

K-means++ seeding

$O(\log k)$-approximation!



Single-Swap K-means++ [LS19]

Lloyd's Algorithm

Only improves current solution.

K-means++ seeding

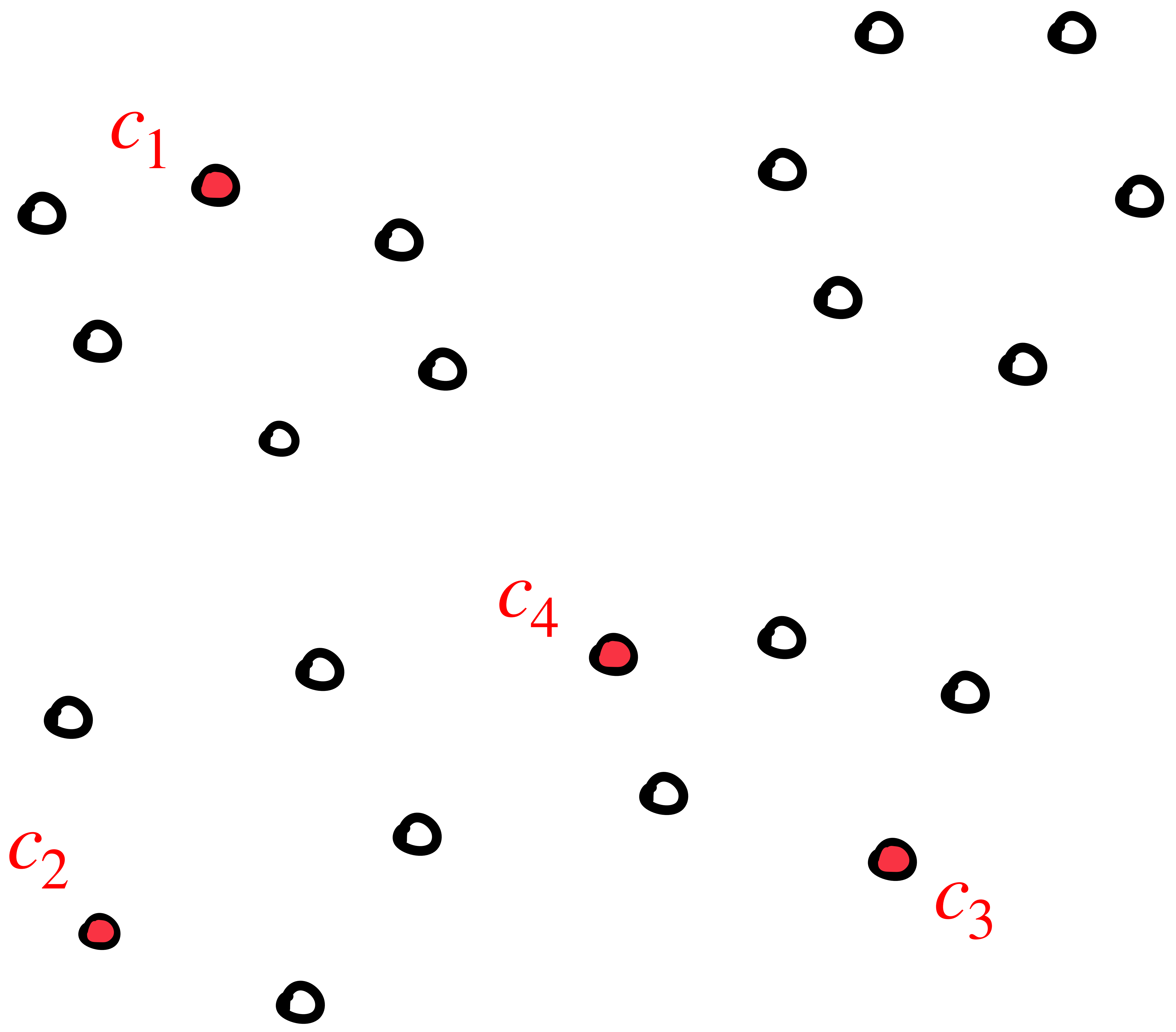
$O(\log k)$-approximation!

Single-Swap KM++

Repeat $\tilde{O}(k)$ times:

Sample 1 more center c_{k+1}

Swap-out the least useful c_j



Single-Swap K-means++ [LS19]

Lloyd's Algorithm

Only improves current solution.

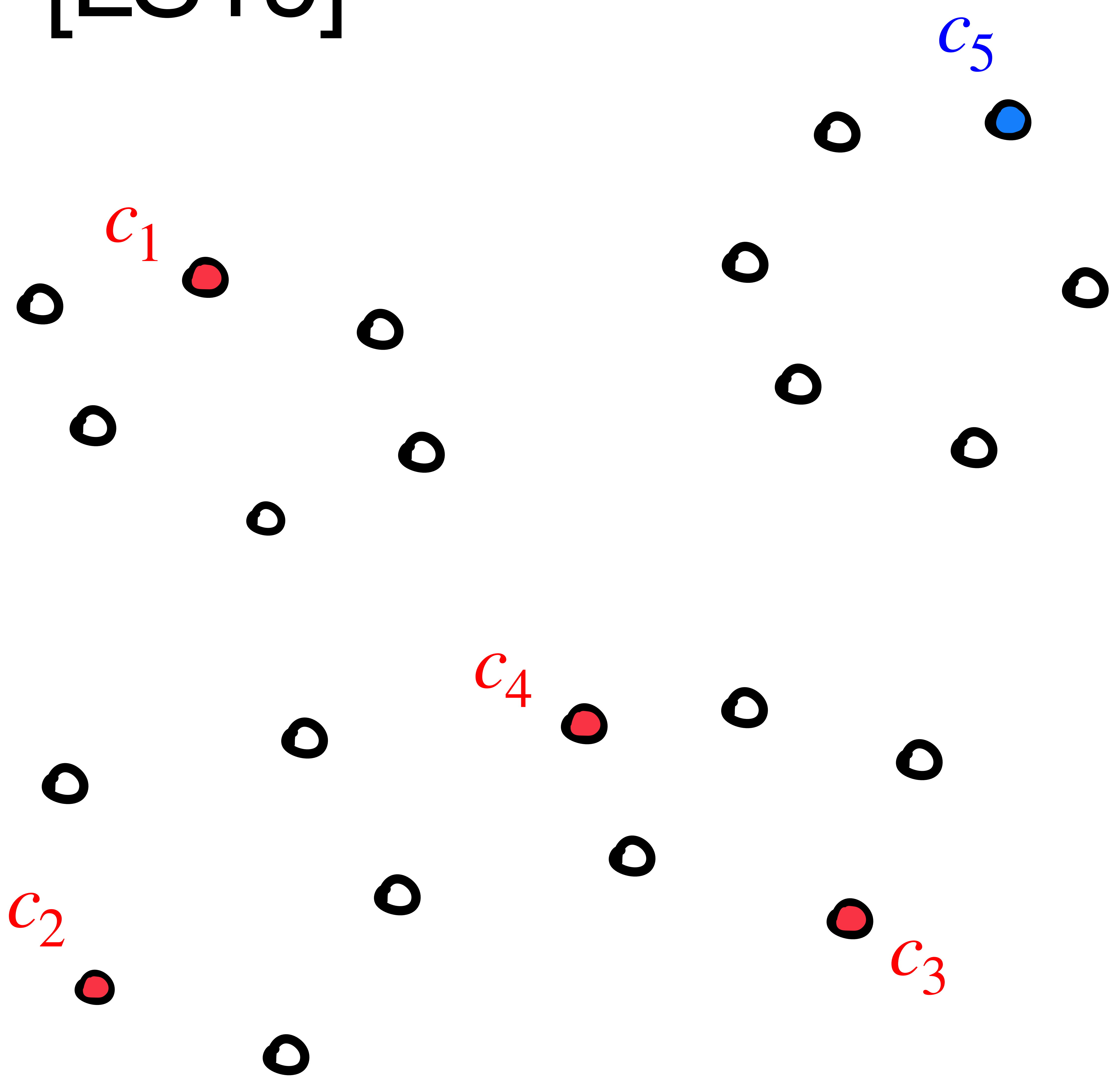
K-means++ seeding

$O(\log k)$-approximation!

Single-Swap KM++

Repeat $\tilde{O}(k)$ times:

- Sample 1 more center c_{k+1}
- Swap-out the least useful c_j



Single-Swap K-means++ [LS19]

Lloyd's Algorithm

Only improves current solution.

K-means++ seeding

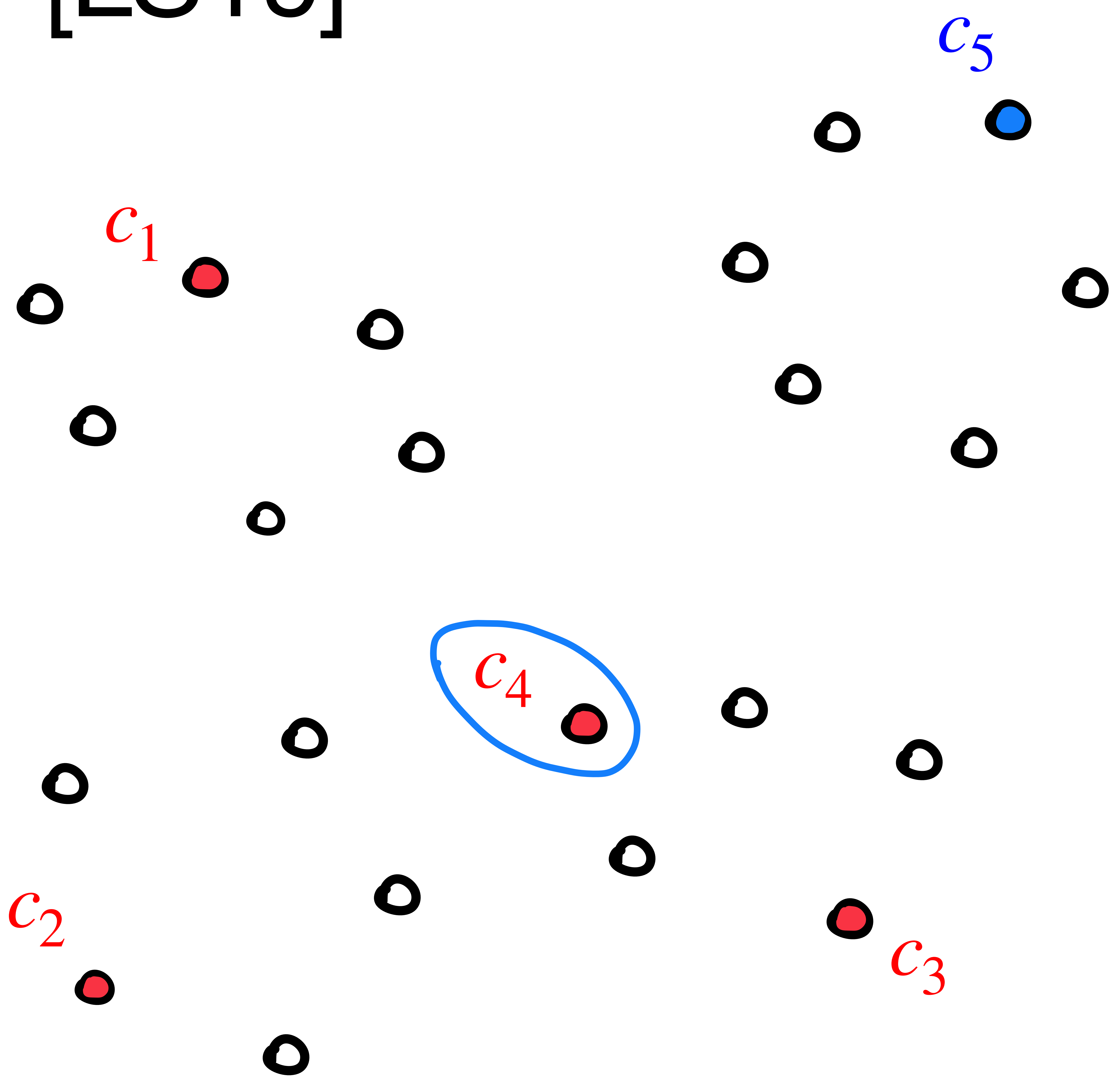
$O(\log k)$-approximation!

Single-Swap KM++

Repeat $\tilde{O}(k)$ times:

Sample 1 more center c_{k+1}

Swap-out the least useful c_j



Single-Swap K-means++ [LS19]

Lloyd's Algorithm

Only improves current solution.

K-means++ seeding

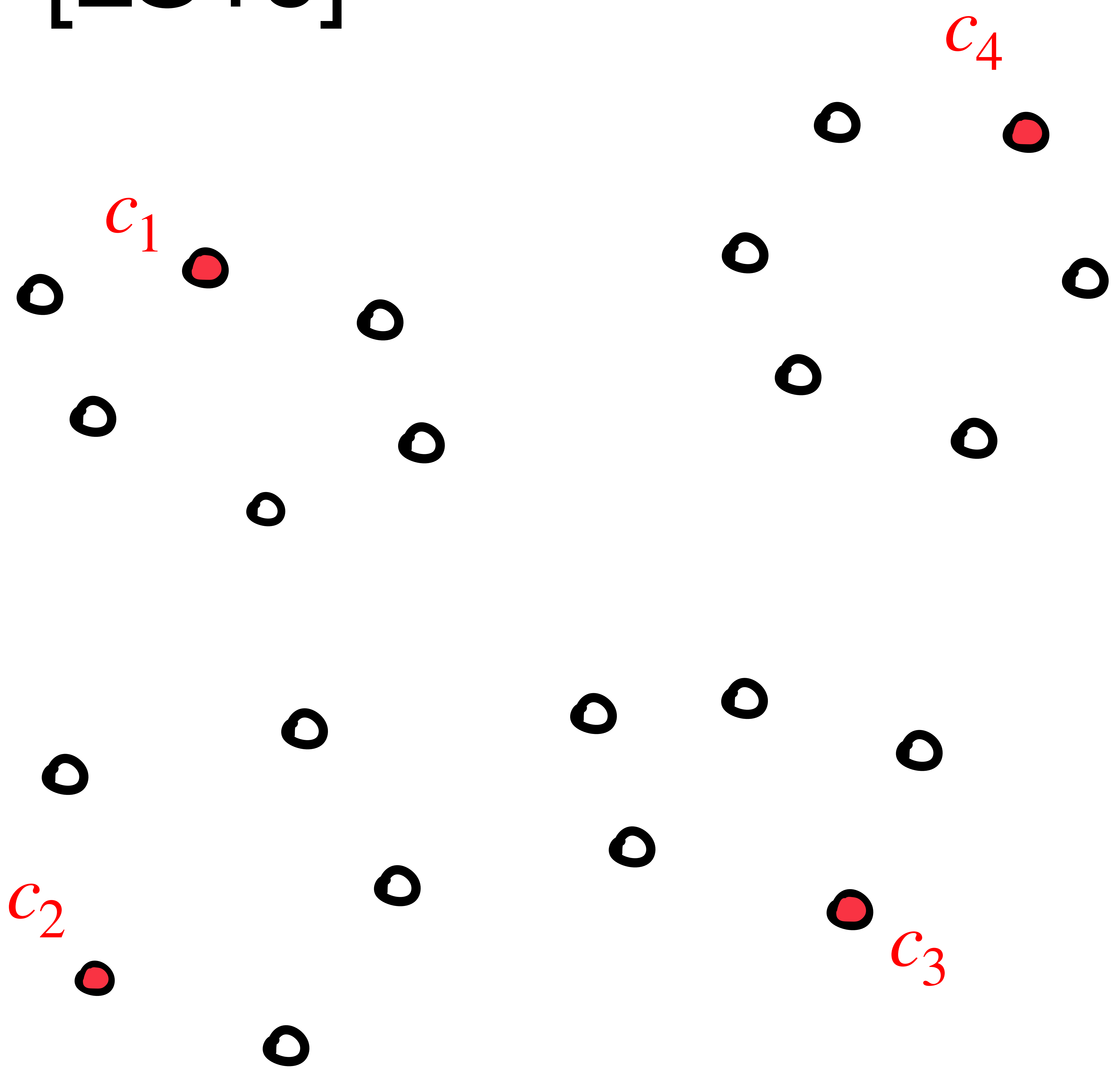
$O(\log k)$-approximation!

Single-Swap KM++

Repeat $\tilde{O}(k)$ times:

  Sample 1 more center c_{k+1}

  Swap-out the least useful c_j



Single-Swap K-means++ [LS19]

Lloyd's Algorithm

Only improves current solution.

K-means++ seeding

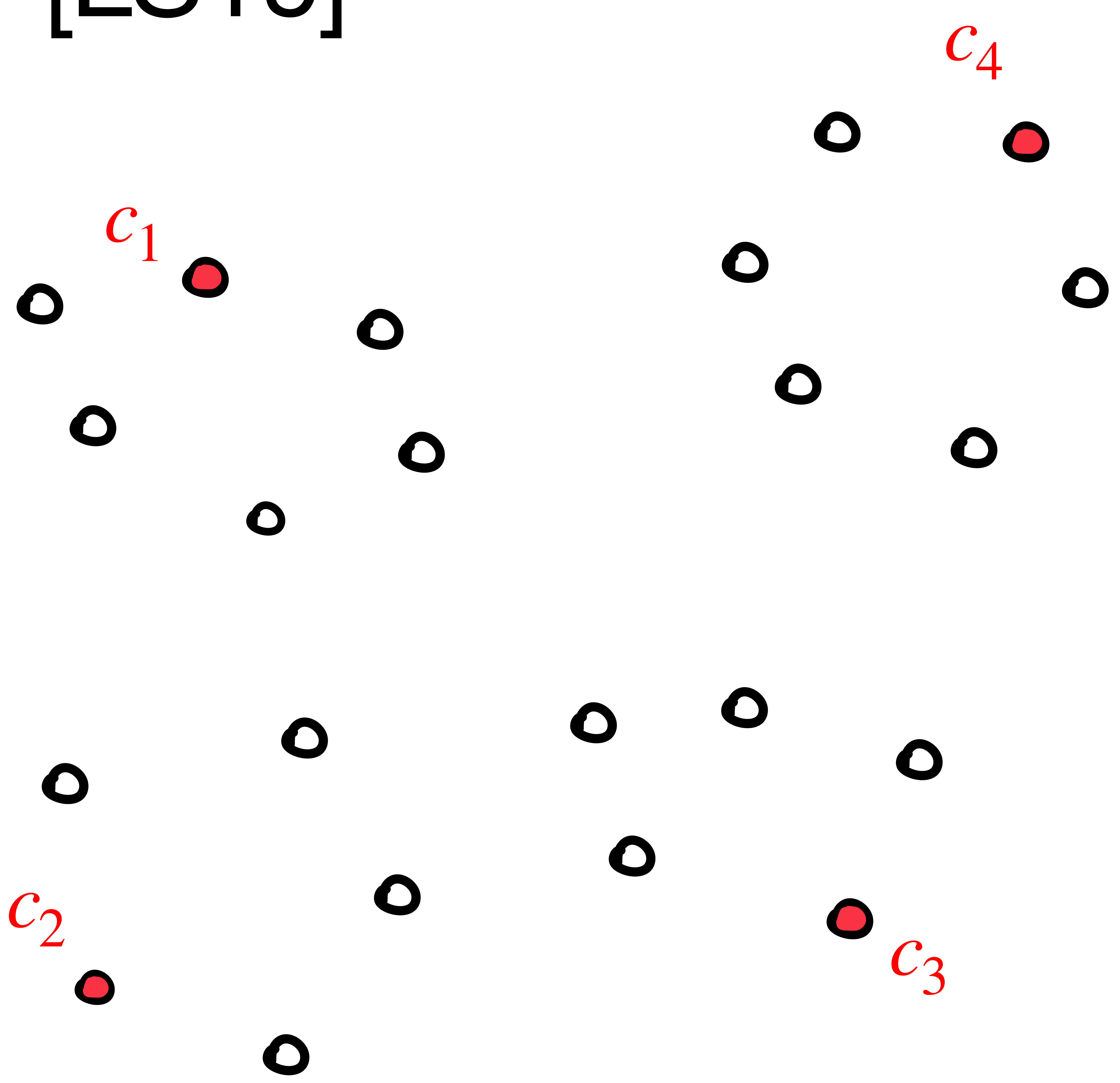
$O(\log k)$-approximation!

Single-Swap KM++

Repeat $\tilde{O}(k)$ times:

- Sample 1 more center c_{k+1}
- Swap-out the least useful c_j

C-approximation! ($C \approx 500$)



Multi-Swap K-means++ [this work]

Lloyd's Algorithm

Only improves current solution.

Single-Swap K-means++

C-approximation! ($C \approx 500$)

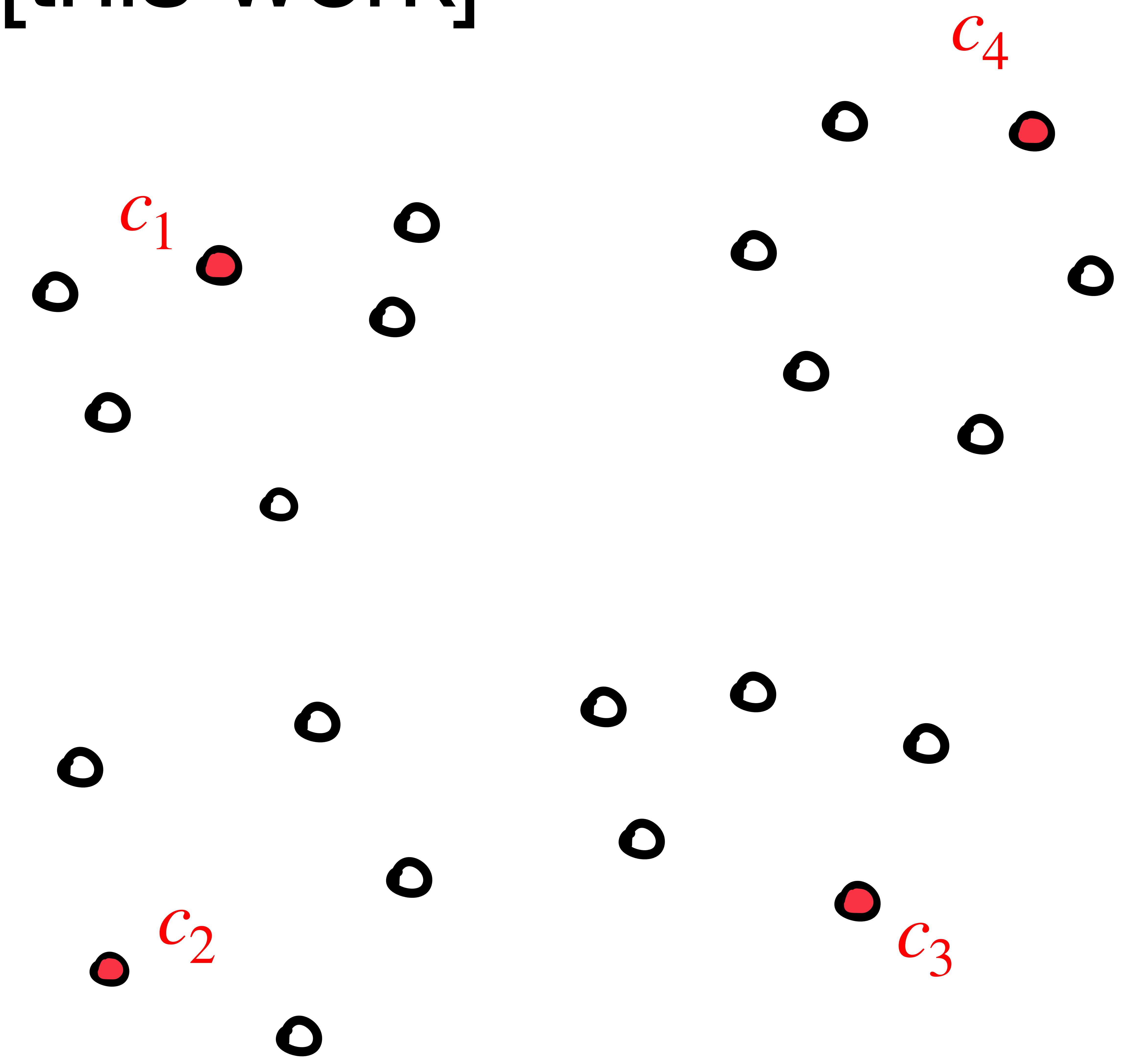
Multi-Swap KM++

Repeat $poly(k)$ times:

Sample p more centers $c_{k+1} \dots c_{k+p}$

Swap-out the least useful $c_{j_1} \dots c_{j_p}$

9-approximation!



Our Theorems

Multi-Swap K-means++

$(9 + \varepsilon)$-approx in time $nd \cdot poly(k)$.

A practical 10.5-approx in time $nd \cdot poly(k)$.

Our Theorems

Multi-Swap K-means++

$(9 + \varepsilon)$-approx in time $nd \cdot poly(k)$.

A <u>practical</u> 10.5-approx in time $nd \cdot poly(k)$.

Single-Swap K-means++

Improved analysis: 26.7-approx rather than (≈ 500)-approx.

Our Theorems

Multi-Swap K-means++

$(9+\varepsilon)$-approx in time $nd \cdot poly(k)$.

A practical 10.5-approx in time $nd \cdot poly(k)$.

Single-Swap K-means++

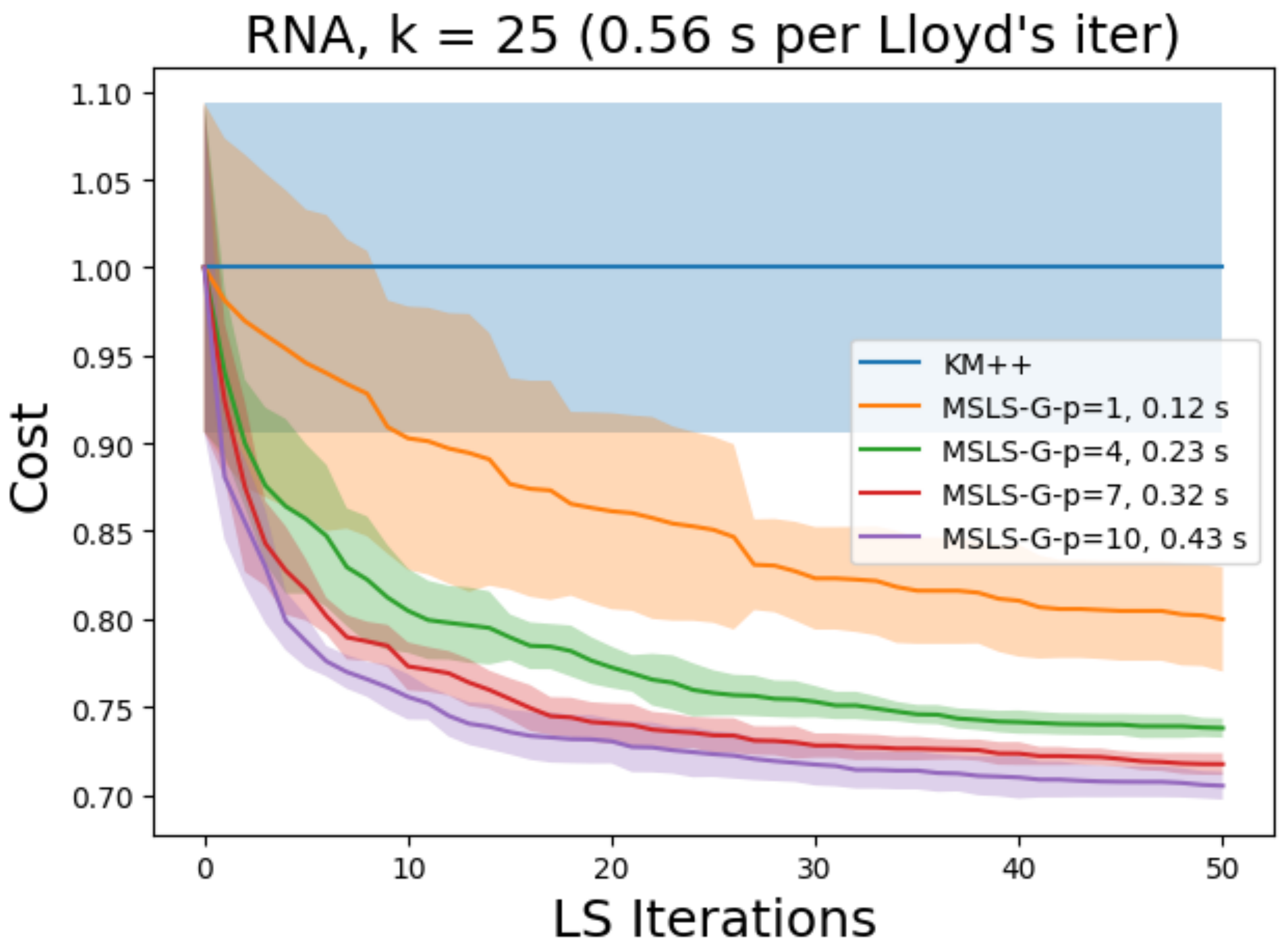
Improved analysis: 26.7-approx rather than (≈ 500)-approx.

A Tight Result for Local Search

[KMNPSW SoCG 02] proved that 9-approx is tight for local search.

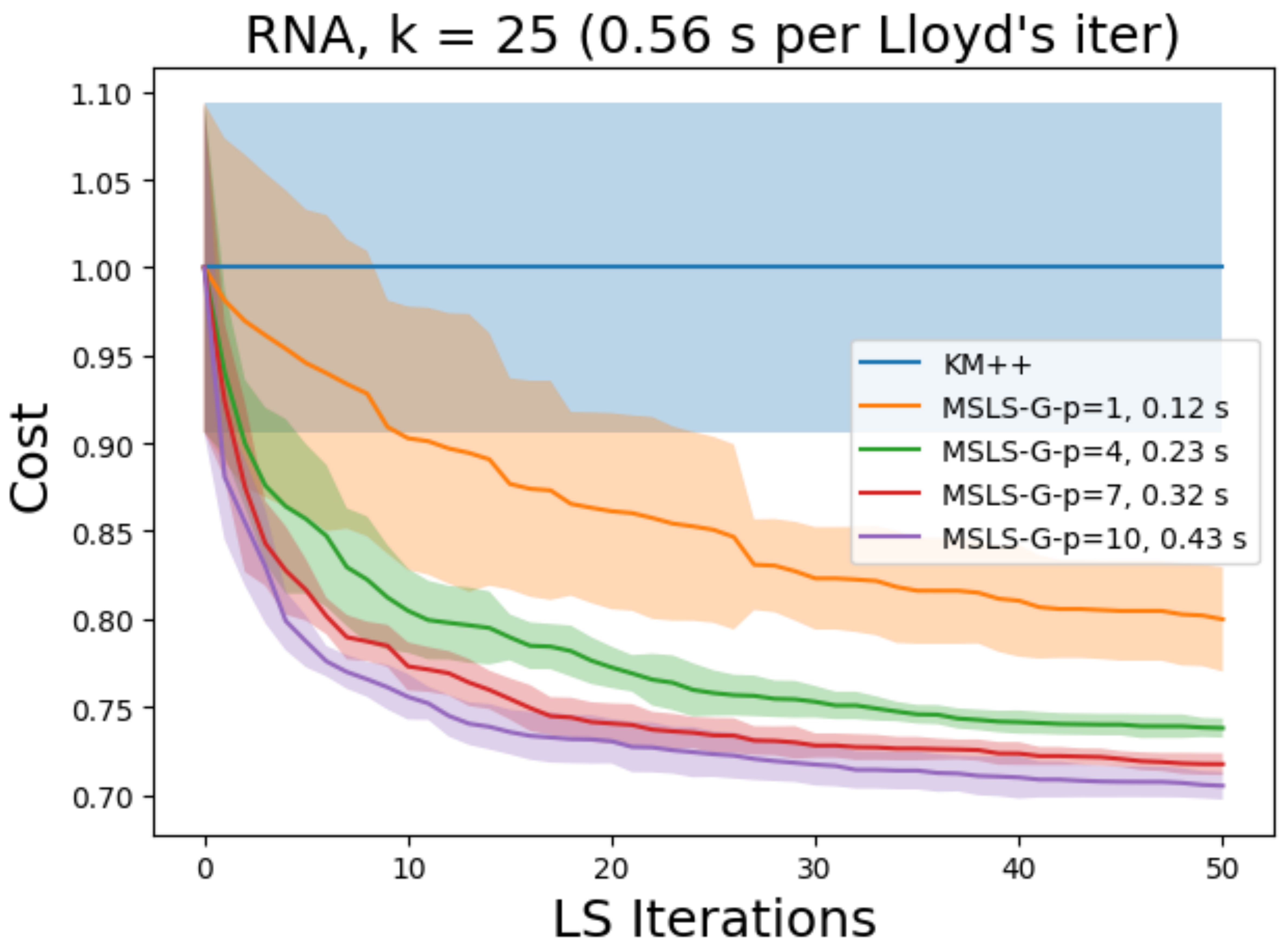
Our Experiments

Multi-Swap K-means++ seeding



Our Experiments

Multi-Swap K-means++ seeding



Seeding + Lloyd's postprocessing