Clustering with Constraints

Practical Theory for Geometric Center Based Clustering

Melanie Schmidt

3rd Workshop on Geometry and Machine Learning, Budapest

June 11th, 2018

Clustering

Clustering

- Unsupervised learning task:

  Find structure in data in form of clusters without labels

Clustering

- Unsupervised learning task:

  Find structure in data in form of clusters without labels

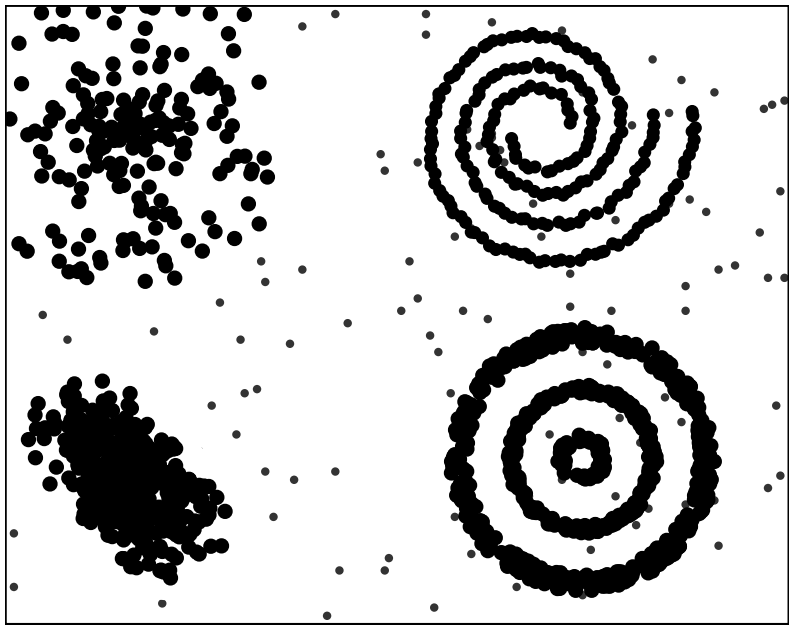
- Clusters can take many shapes.

Clustering

- Unsupervised learning task:

  Find structure in data in form of clusters without labels

- Clusters can take many shapes. A quote from Jain's much cited survey [J10]: *'. . . none of the available clustering algorithms can detect all these clusters'* [in Fig 2. from [J10], redrawn below]:



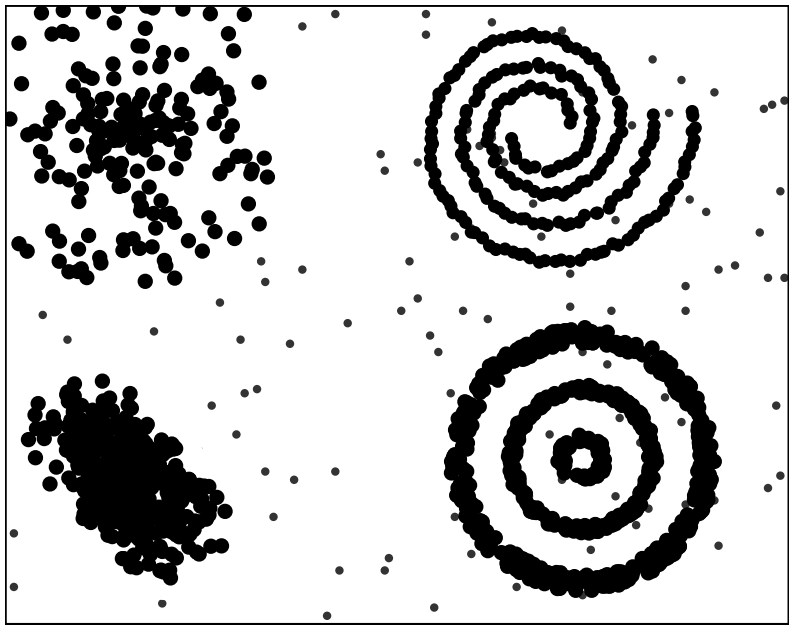
[J10] Jain: 50 years beyond *k*-means, Pattern Recognition Letters, 2010.

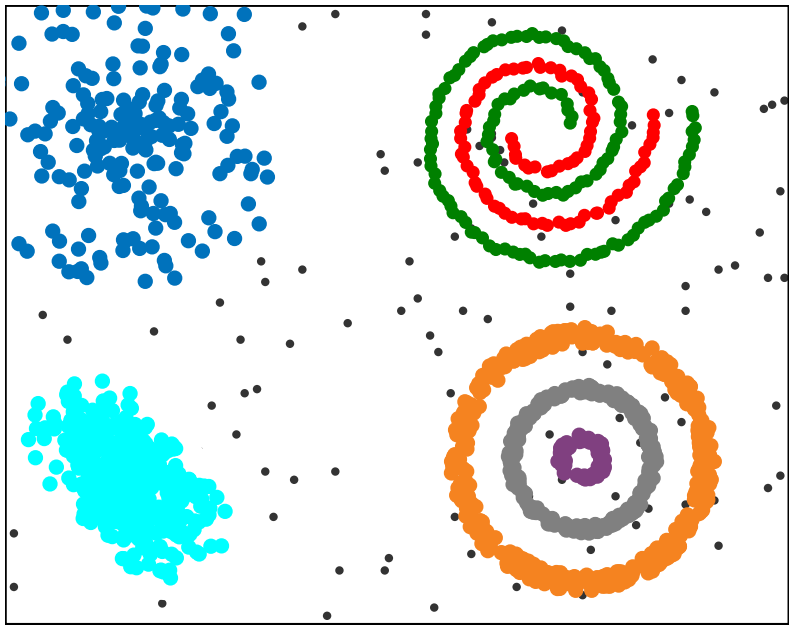
Clustering

- Unsupervised learning task:

  Find structure in data in form of clusters without labels

- Clusters can take many shapes. A quote from Jain's much cited survey [J10]: *'. . . none of the available clustering algorithms can detect all these clusters'* [in Fig 2. from [J10], redrawn below]:





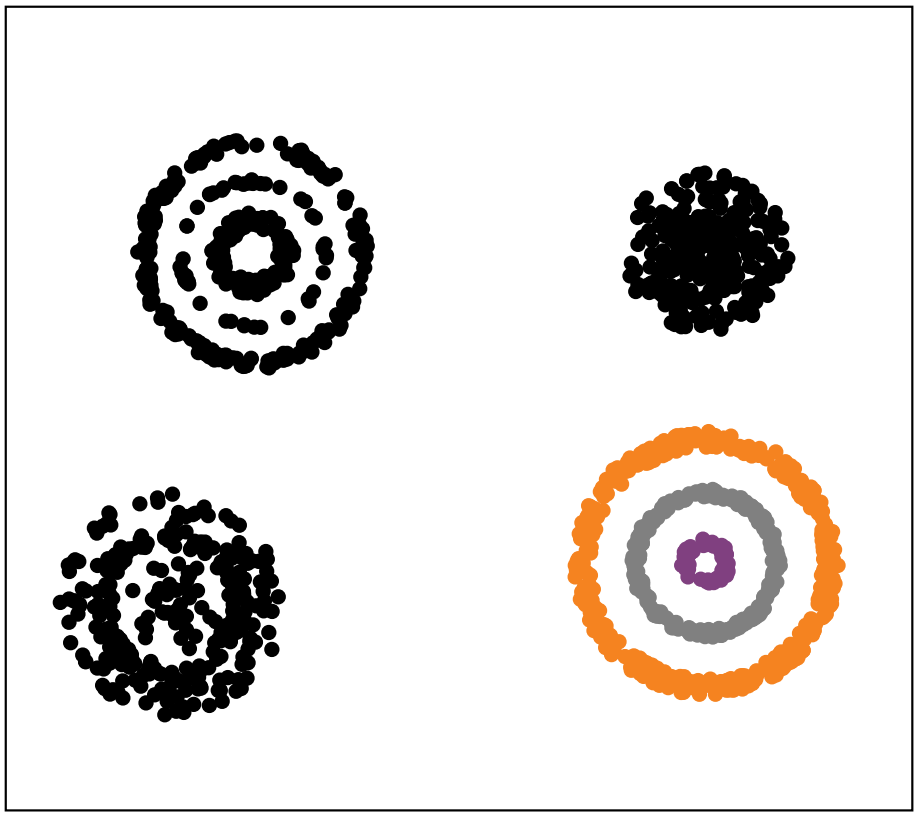
[J10] Jain: 50 years beyond *k*-means, Pattern Recognition Letters, 2010.

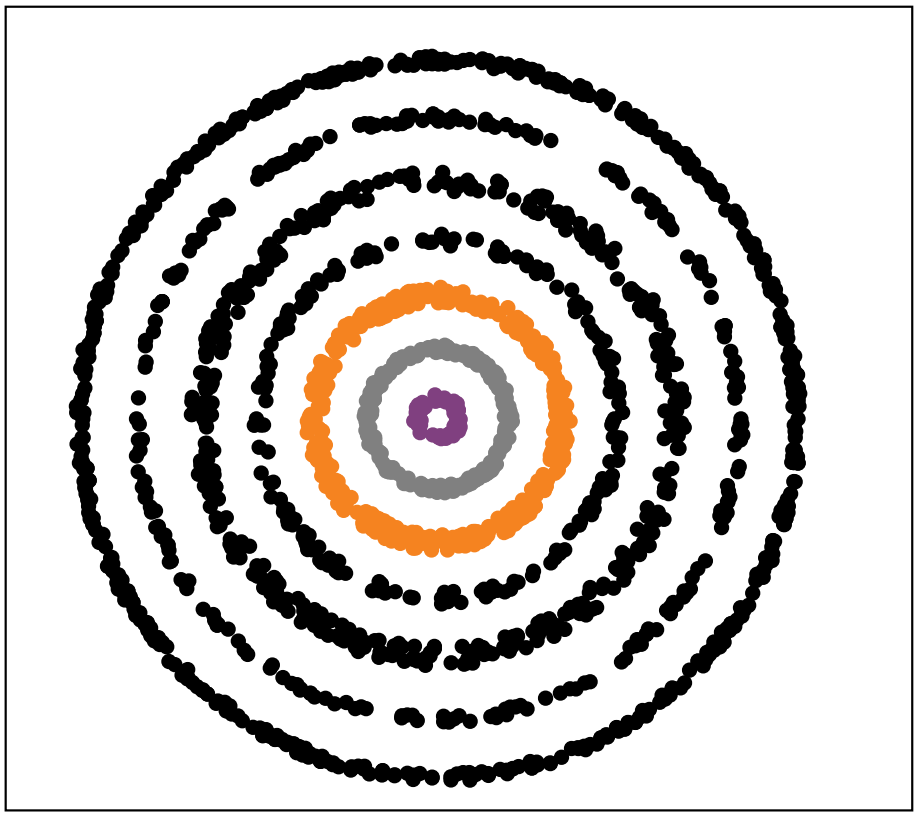
Even worse:

- The correct output may depend on the data set
- The ‘natural’ clustering may even lie in the eye of the beholder

Even worse:

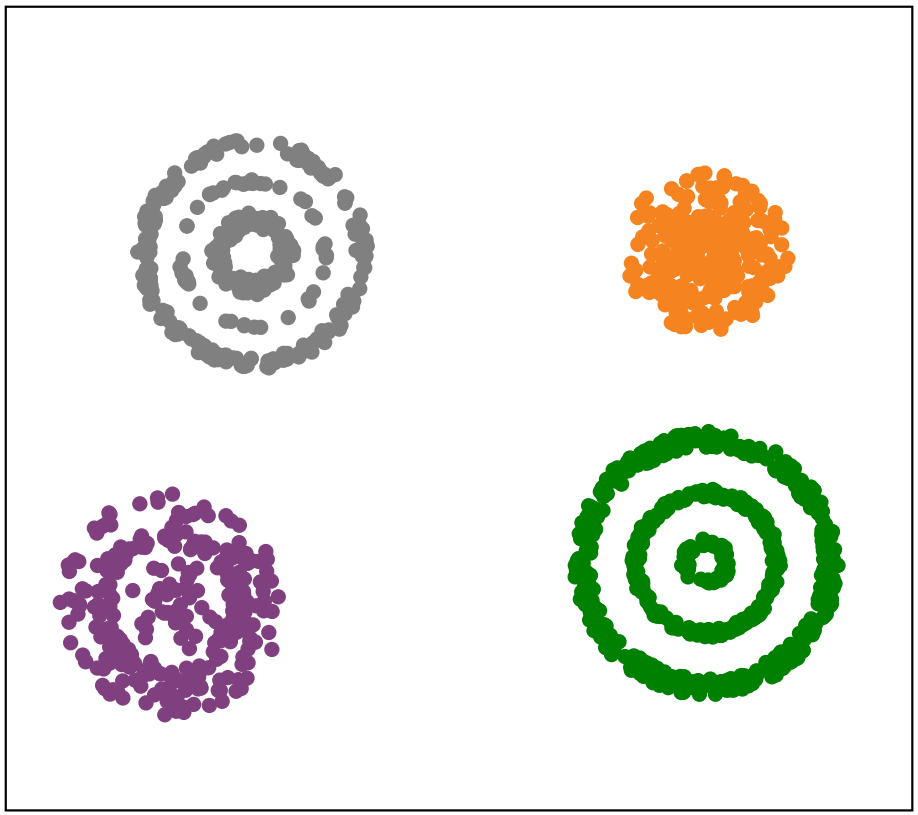
- The correct output may depend on the data set
- The ‘natural’ clustering may even lie in the eye of the beholder

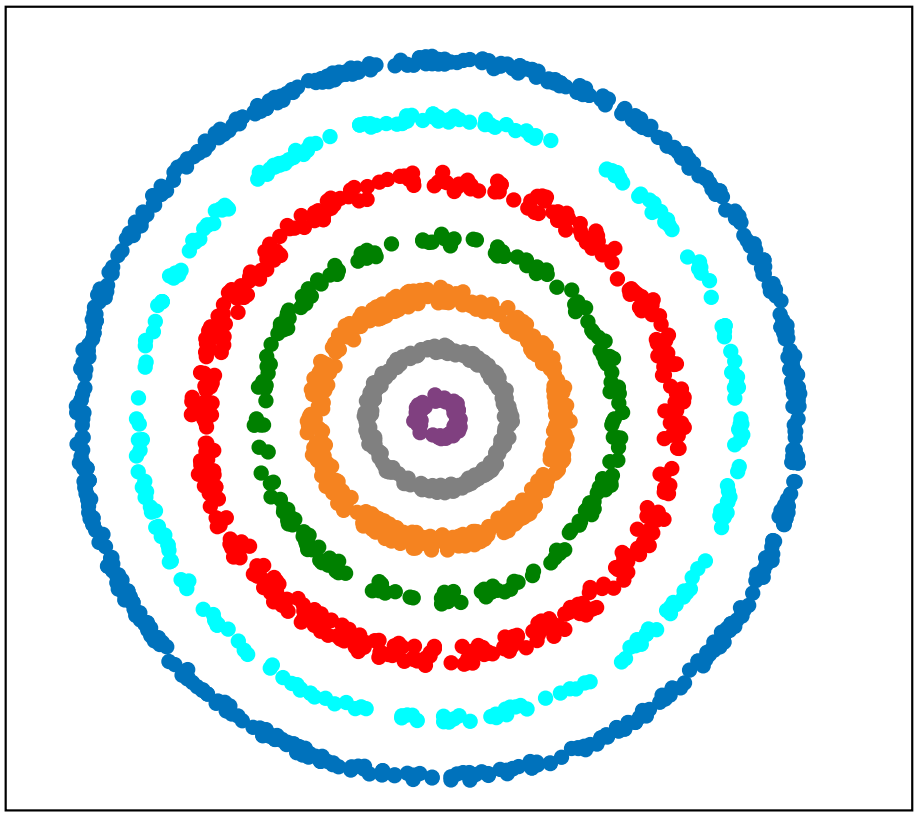




Even worse:

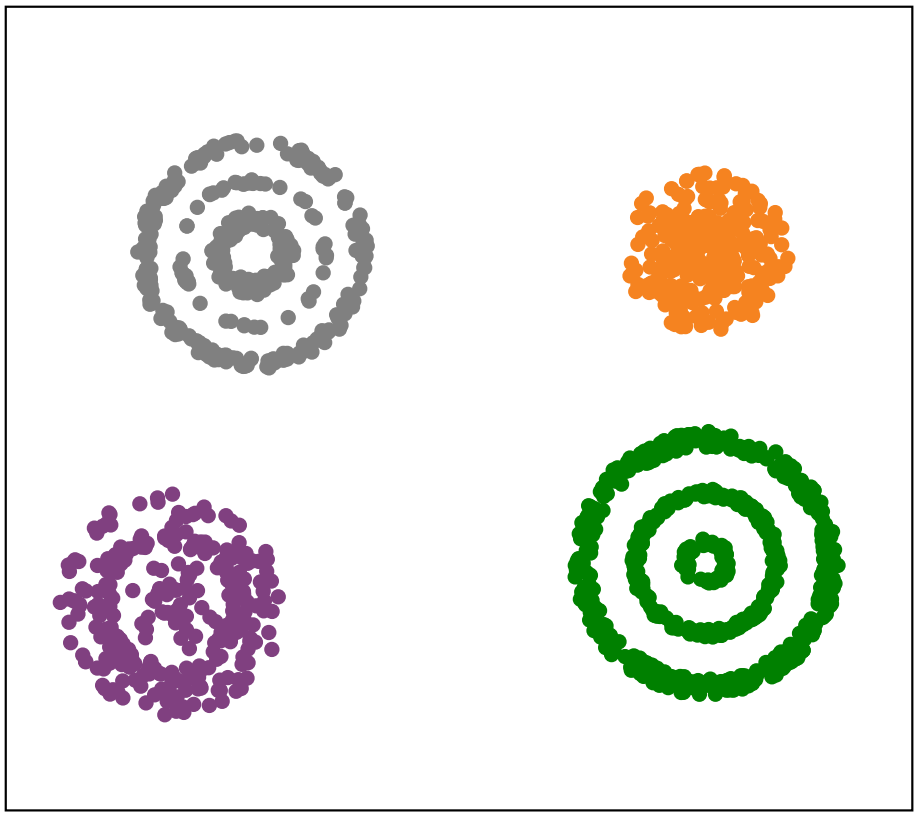
- The correct output may depend on the data set
- The 'natural' clustering may even lie in the eye of the beholder

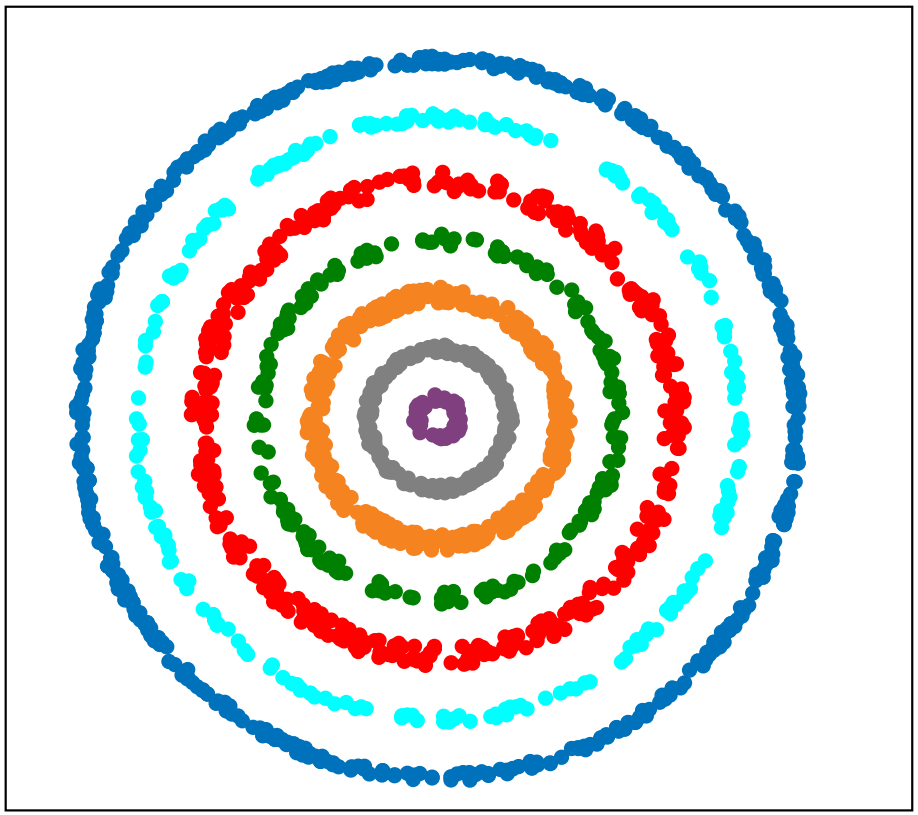




Even worse:

- The correct output may depend on the data set
- The ‘natural’ clustering may even lie in the eye of the beholder

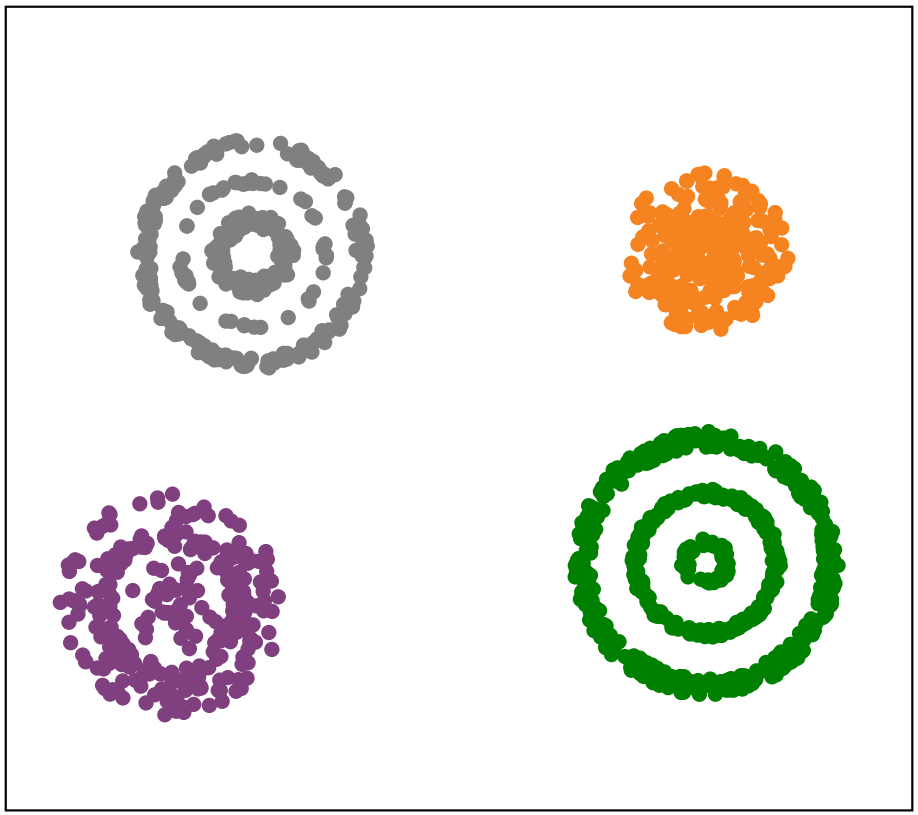


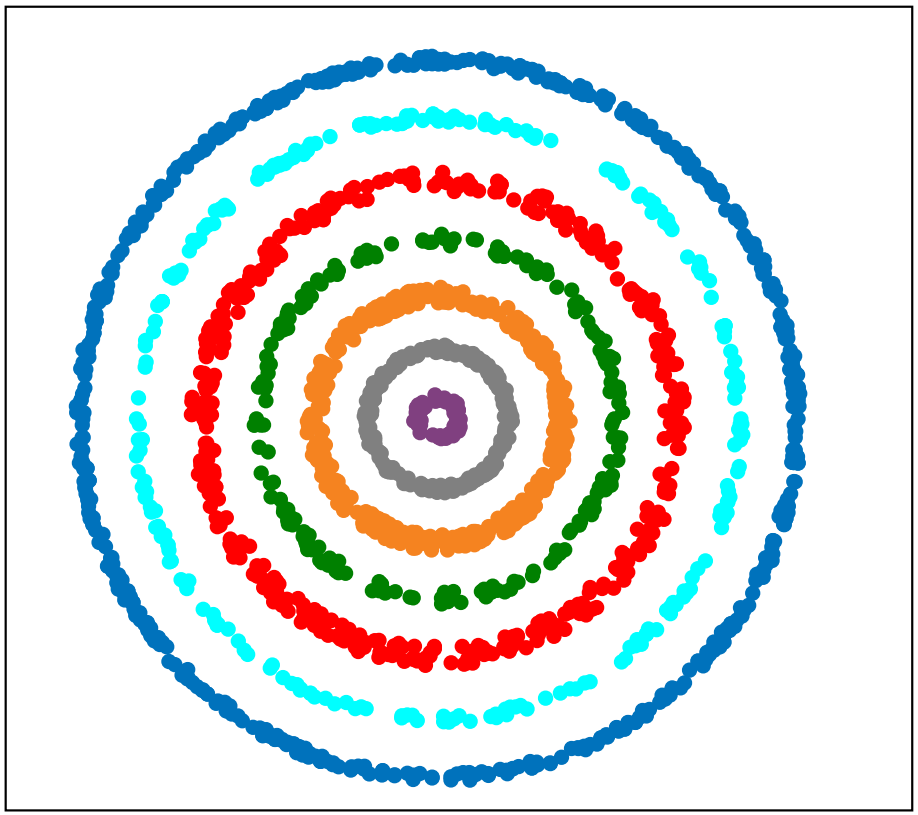


Consequence: Tons of different objectives and algorithms.

Even worse:

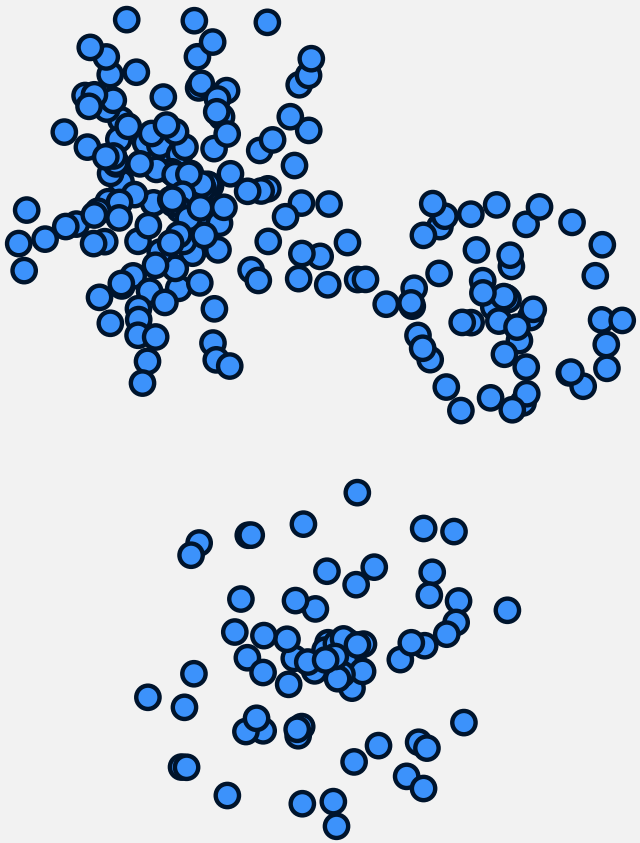
- The correct output may depend on the data set
- The 'natural' clustering may even lie in the eye of the beholder





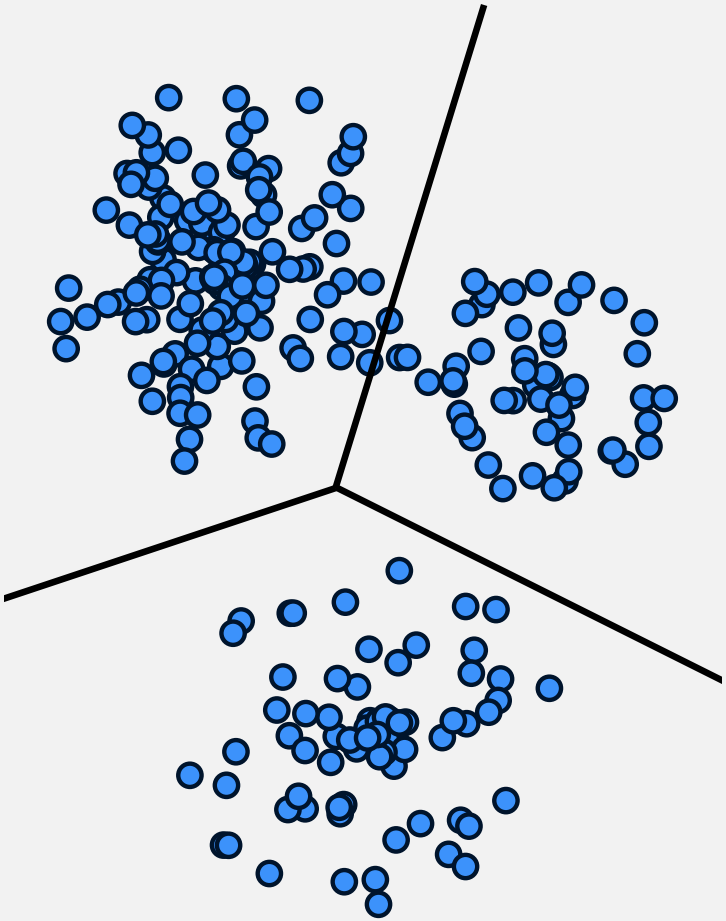
This talk: Partitional Clustering, geometric setting

Partitional Clustering



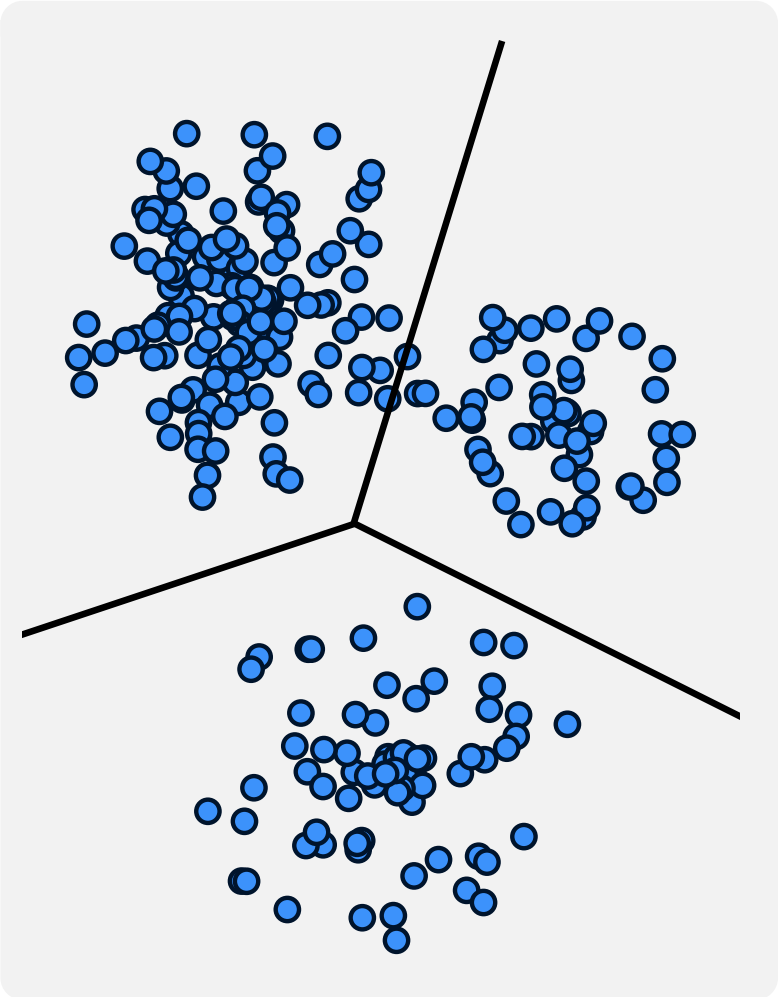
Partitional Clustering

- Given a point set P (and $k \in \mathbb{N}$)
- and some pairwise distances,



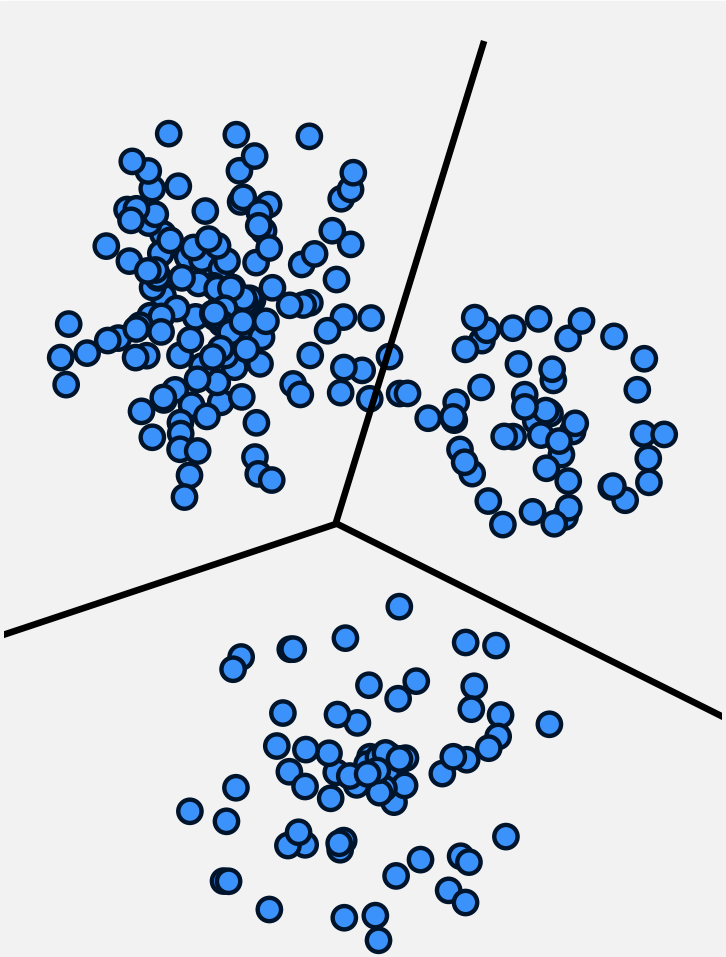
Partitional Clustering

- Given a point set P (and $k \in \mathbb{N}$)
- and some pairwise distances,
- partition P into $C_1, \ldots, C_k$



Partitional Clustering

- Given a point set P (and $k \in \mathbb{N}$)
- and some pairwise distances,
- partition P into $C_1, \ldots, C_k$
- to optimize some objective



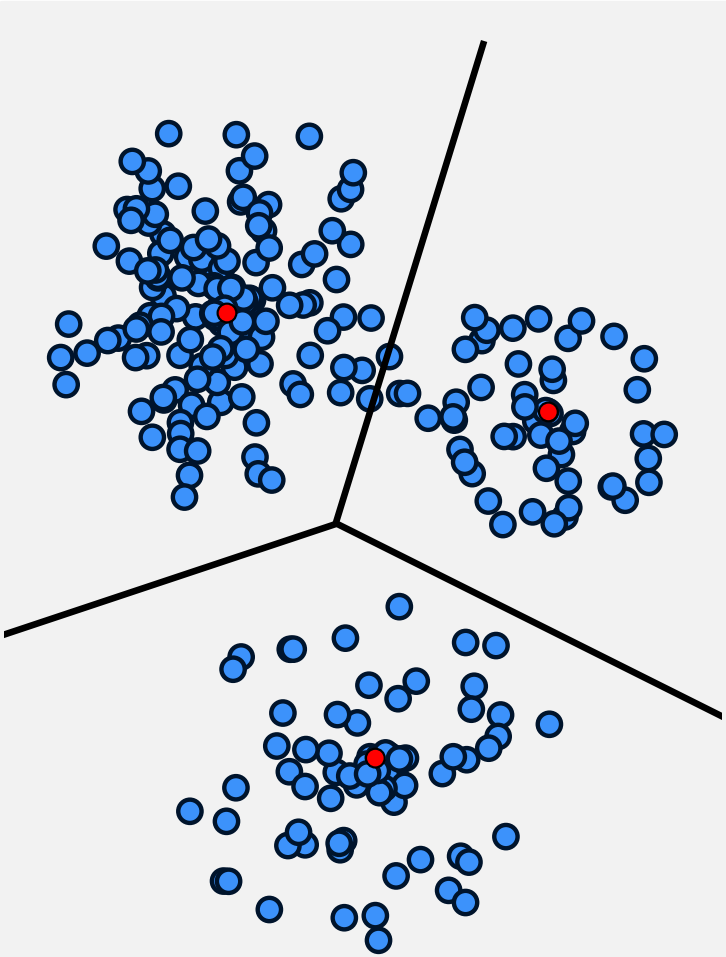
Partitional Clustering

- Given a point set P (and $k \in \mathbb{N}$)
- and some pairwise distances,
- partition P into $C_1, \ldots, C_k$
- to optimize some objective

The three 'k'-objectives

Pick k centers $\rightsquigarrow$ induced partitioning

- k-center: maximum distance, metric
- k-median: sum of distances, metric
- k-means: sum of squared distances, Euclidean space $\mathbb{R}^d$



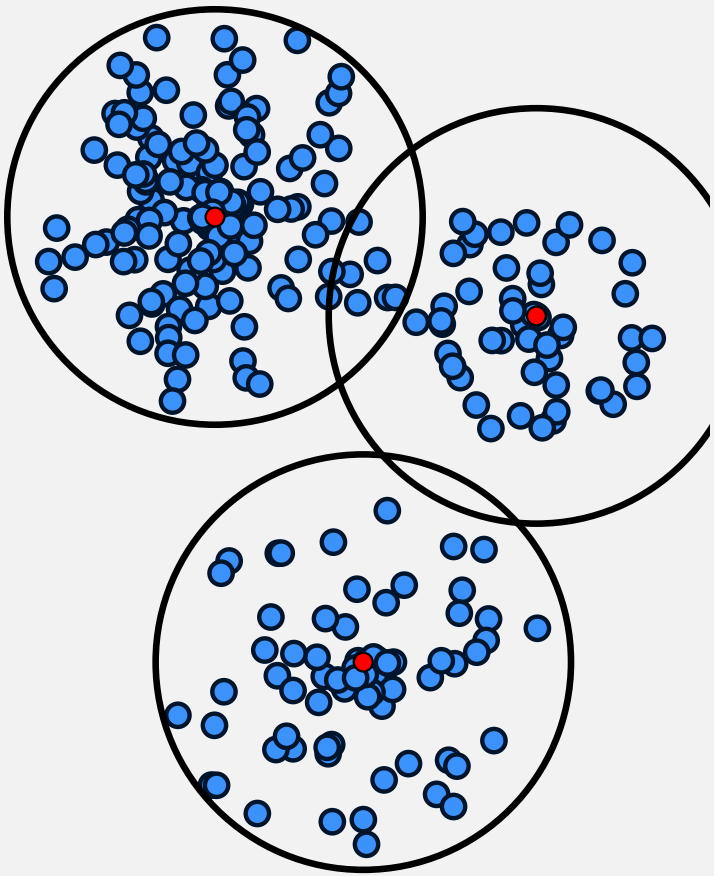
Partitional Clustering

- Given a point set P (and $k \in \mathbb{N}$)
- and some pairwise distances,
- partition P into $C_1, \ldots, C_k$
- to optimize some objective

The three 'k'-objectives

Pick k centers $\rightsquigarrow$ induced partitioning

- k-center: maximum distance, metric
- k-median: sum of distances, metric
- k-means: sum of squared distances, Euclidean space $\mathbb{R}^d$



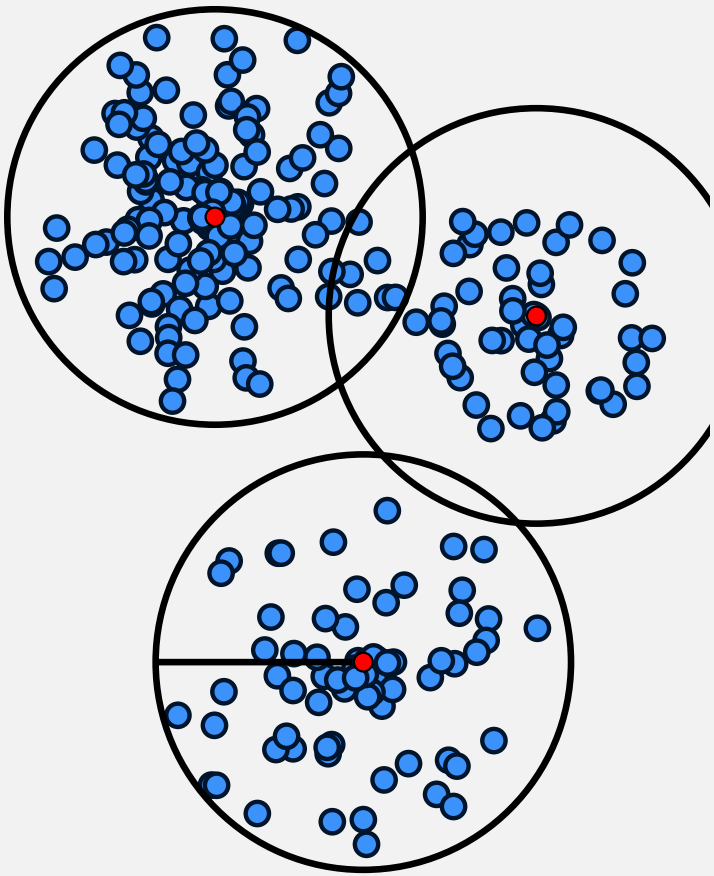
Partitional Clustering

- Given a point set P (and $k \in \mathbb{N}$)
- and some pairwise distances,
- partition P into $C_1, \ldots, C_k$
- to optimize some objective

The three 'k'-objectives

Pick k centers $\rightsquigarrow$ induced partitioning

- k-center: maximum distance, metric
- k-median: sum of distances, metric
- k-means: sum of squared distances, Euclidean space $\mathbb{R}^d$



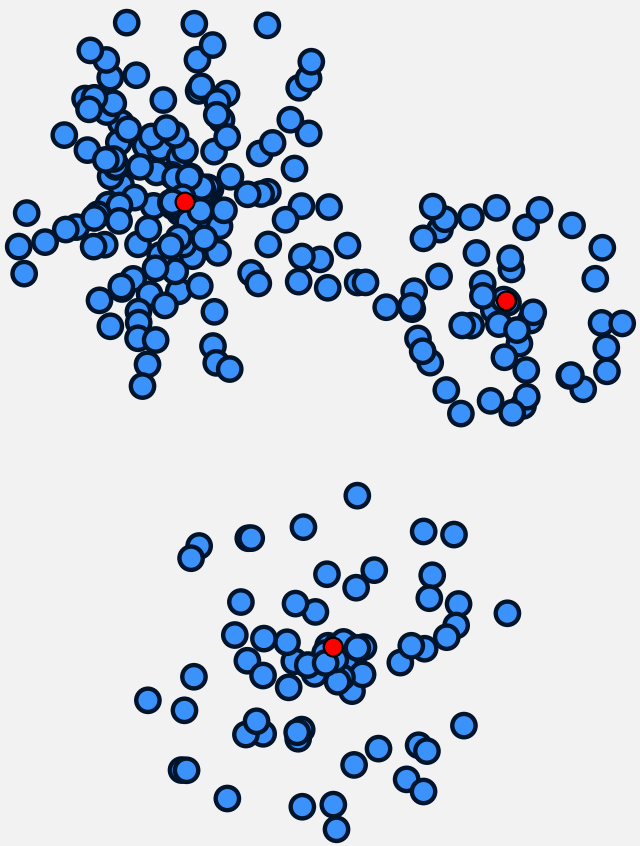
Partitional Clustering

- Given a point set P (and $k \in \mathbb{N}$)
- and some pairwise distances,
- partition P into $C_1, \ldots, C_k$
- to optimize some objective

The three 'k'-objectives

Pick k centers $\rightsquigarrow$ induced partitioning

- k-center: maximum distance, metric
- k-median: sum of distances, metric
- k-means: sum of squared distances, Euclidean space $\mathbb{R}^d$



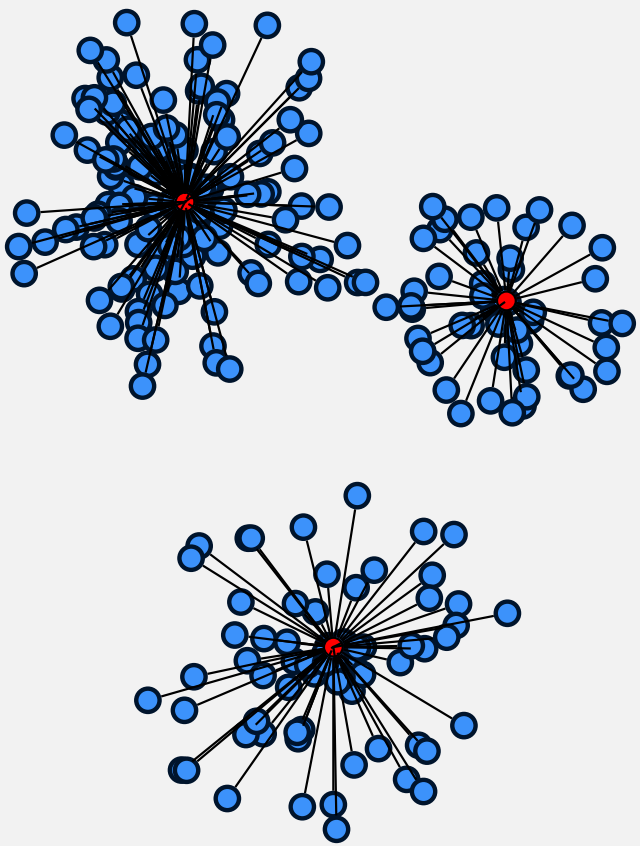
Partitional Clustering

- Given a point set P (and $k \in \mathbb{N}$)
- and some pairwise distances,
- partition P into $C_1, \ldots, C_k$
- to optimize some objective

The three 'k'-objectives

Pick k centers $\rightsquigarrow$ induced partitioning

- k-center: maximum distance, metric
- k-median: sum of distances, metric
- k-means: sum of squared distances, Euclidean space $\mathbb{R}^d$



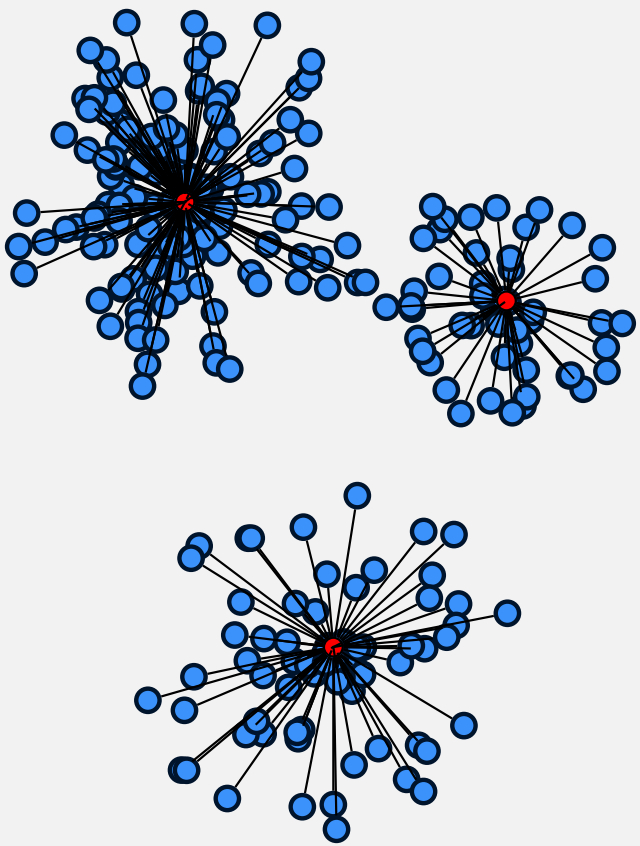
Partitional Clustering

- Given a point set P (and $k \in \mathbb{N}$)
- and some pairwise distances,
- partition P into $C_1, \ldots, C_k$
- to optimize some objective

The three 'k'-objectives

Pick k centers $\rightsquigarrow$ induced partitioning

- k-center: maximum distance, metric
- k-median: sum of distances, metric
- k-means: sum of squared distances, Euclidean space $\mathbb{R}^d$



Partitional Clustering

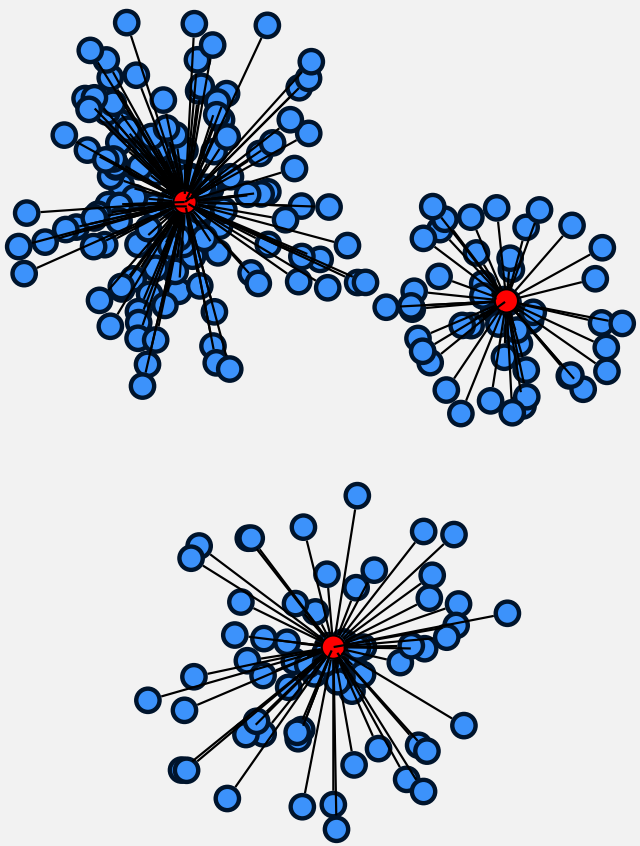
- Given a point set P (and $k \in \mathbb{N}$)
- and some pairwise distances,
- partition P into $C_1, \ldots, C_k$
- to optimize some objective

The three 'k'-objectives

Pick k centers $\rightsquigarrow$ induced partitioning

- k-center: maximum distance, metric
- k-median: sum of distances, metric
- k-means: sum of squared distances, Euclidean space $\mathbb{R}^d$

Facility location: like k-median, but opening cost instead of k



Partitional Clustering

- Given a point set P (and $k \in \mathbb{N}$)
- and some pairwise distances,
- partition P into $C_1, \ldots, C_k$
- to optimize some objective

The three 'k'-objectives

Pick k centers $\rightsquigarrow$ induced partitioning

- k-center: maximum distance, metric
- k-median: sum of distances, metric
- k-means: sum of squared distances, Euclidean space $\mathbb{R}^d$

Facility location: like k-median, but opening cost instead of k

not center-based: min sum k-clustering, aversion k-clustering

Approximation: State of the art

	Approximation	Hardness
k-center		
facility location		
k-median		
k-means		
min sum k-clustering		
aversion k-clustering		

[G85] Gonzalez. Clustering to minimize the maximum intercluster distance, Theoretical Computer Science 1985.
[GK99] Guha, Khuller. Greedy strikes back: Improved facility location algorithms, J. Algorithms 1999.

[HN79] Hsu, Nemhauser, Easy and hard bottleneck location problems. Discrete Applied Mathematics 1979.
[HS85] Hochbaum, Shmoys, A best possible heuristic for the k-center problem, Mathematics of Operations Research 1985.

[L11] Li. A 1.488-approximation algorithm for the uncapacitated facility location problem. ICALP 2011.

Approximation: State of the art

	Approximation	Hardness
k-center	2 [G85,HS85]	2 [HN79]
facility location	1.488 [L11]	1.463 [GK99]
k-median	$3+\varepsilon$ [AGK+01]	$1+2/e \approx 1.74$ [HN79]
k-means	$9+\varepsilon$ [KMN+02]	
min sum k-clustering	$\mathcal{O}(\epsilon^{-1}\log^{1+\epsilon} n)$ [BZR01]	
aversion k-clustering		

[AGK+01] Arya, Garg, Khandekar, Meyerson, Munagala, Pandit. Local search heuristic for k-median [...], STOC 2001.

[BZR01] Bartal, Charikar, Raz. Approximating min-sum k-Clustering in metric spaces. STOC 2001.
[G85] Gonzalez. Clustering to minimize the maximum intercluster distance, Theoretical Computer Science 1985.
[GK99] Guha, Khuller. Greedy strikes back: Improved facility location algorithms, J. Algorithms 1999.

[HN79] Hsu, Nemhauser, Easy and hard bottleneck location problems. Discrete Applied Mathematics 1979.
[HS85] Hochbaum, Shmoys, A best possible heuristic for the k-center problem, Mathematics of Operations Research 1985.
[KMN+02] Kanungo, Mount, Netanyahu, Piatko, Silverman, Wu. A local search approx. alg. for k-means clustering, SoCG 2002.
[L11] Li. A 1.488-approximation algorithm for the uncapacitated facility location problem. ICALP 2011.

Approximation: State of the art

	Approximation	Hardness
k-center	2 [G85,HS85]	2 [HN79]
facility location	1.488 [L11]	1.463 [GK99]
k-median	$2.675 + \varepsilon$ [BPRST15]	$1 + 2/e \approx 1.74$ [HN79]
k-means	6.357 [ASFW17]	1.0013 [ACKS15], [LSW15]
min sum k-clustering	$\mathcal{O}(\log n)$ [BFSS15]	
aversion k-clustering	$\mathcal{O}(1)$ [GGS16]	

[ACKS15] Awasthi, Charikar, Krishnaswamy, Sinop. The hardness of approximation of euclidean k-means, SoCG 2015.
[AHSW17] Ahmadian, Norouzi-Fard, Svensson, Ward: Better Guarantees for k-Means and Euclidean k-Median by Primal-Dual Algori
[BFSS15] Behsaz, Friggstad, Salavatipour, Sivakumar. Appr. Alg. for Min-Sum k-Clustering and Balanced k-Median, ICALP 2015.
[BPRST15] Byrka, Pensyl, Rybicki, Srinivasan, Trinh. An Improved Approximation for k-median [...], SODA 2015.
[G85] Gonzalez. Clustering to minimize the maximum intercluster distance, Theoretical Computer Science 1985.
[GK99] Guha, Khuller. Greedy strikes back: Improved facility location algorithms, J. Algorithms 1999.
[GGS16] Gupta, Guruganesh, S. Approximation Algorithms for Aversion k-Clustering via Local k-Median, ICALP 2016.
[HN79] Hsu, Nemhauser, Easy and hard bottleneck location problems. Discrete Applied Mathematics 1979.
[HS85] Hochbaum, Shmoys, A best possible heuristic for the k-center problem, Mathematics of Operations Research 1985.
[LSW15] Lee, S. Wright. Improved and Simplified Inapproximability for k-means, IPL 2017.
[L11] Li. A 1.488-approximation algorithm for the uncapacitated facility location problem. ICALP 2011.

Many techniques

Center based

- k-center somewhat easier
- Chain: first k-center or facility location, then k-median, and k-means is last

Many techniques

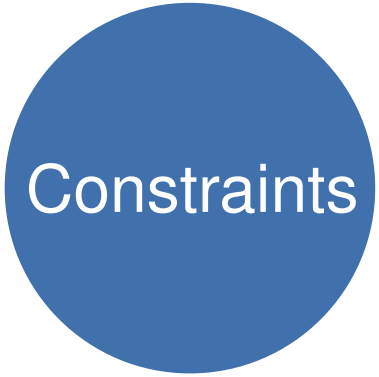
Center based

- k-center somewhat easier
- Chain: first k-center or facility location, then k-median, and k-means is last

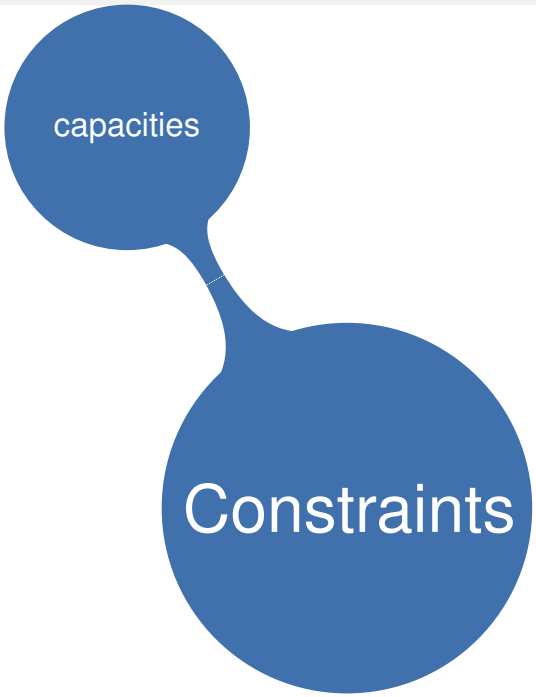
- Embedding into tree metrics
- Local Search
- LP-based rounding
  - Filtering
  - Dual fitting
  - Randomized rounding
- Primal-dual framework

Any problems?

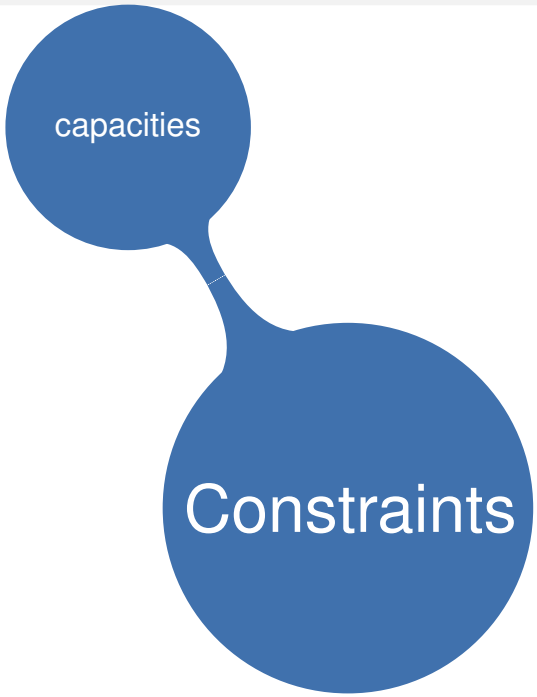
Any problems?



Any problems?

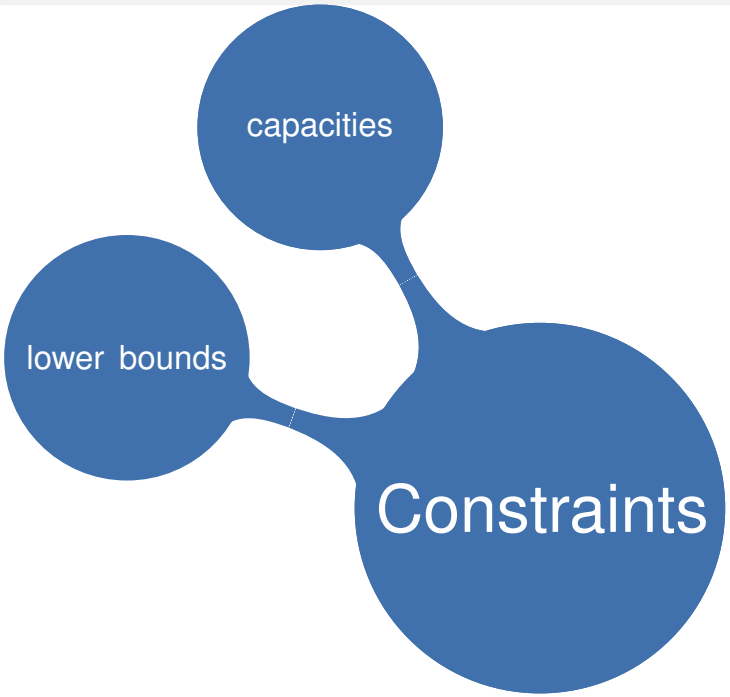


Any problems?



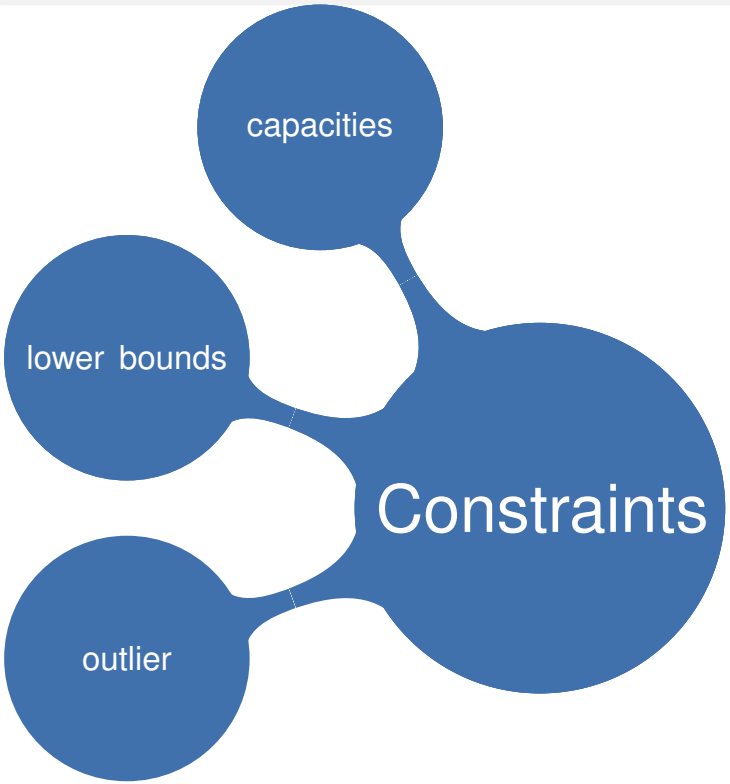
pictures: creative commons from flickr and pixabay

Any problems?



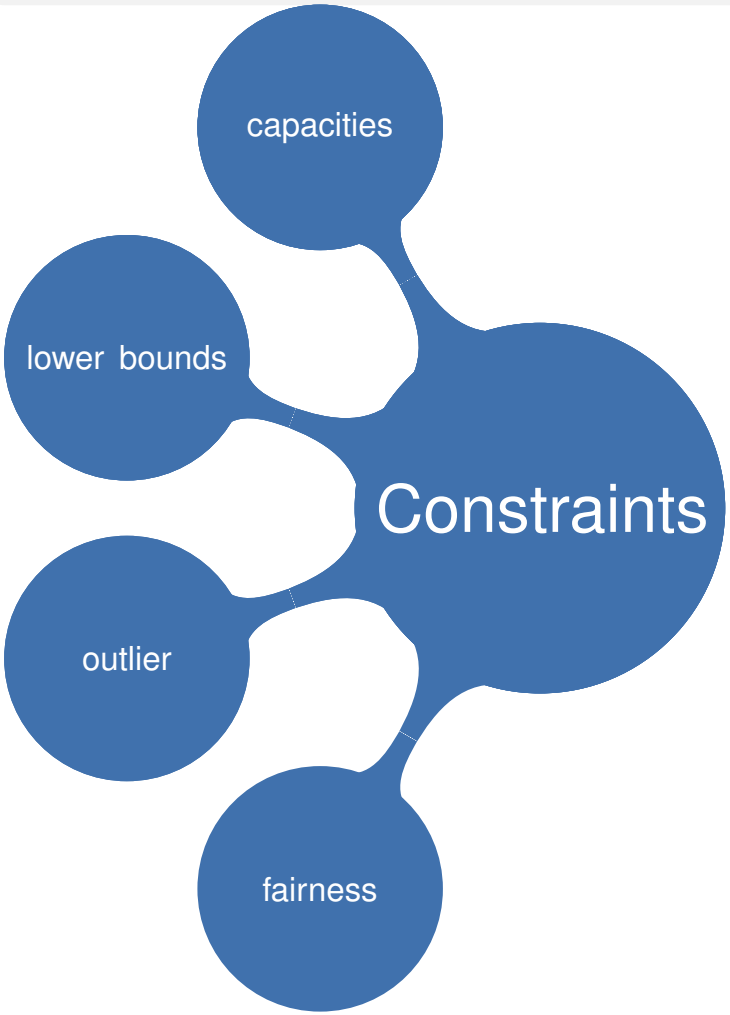
pictures: creative commons from flickr and pixabay

Any problems?



pictures: creative commons from flickr and pixabay

Any problems?



pictures: creative commons from flickr and pixabay

Another motivation

Another motivation

Techniques for non-center based objectives

- Reductions to center-based objectives
- min sum k-clustering: balanced k-median (cluster cost times $|C_i|$)
- aversion k-clustering: local k-median (centers have a radius)

Basic clustering objectives are well studied,
but clustering with constraints is not

Basic clustering objectives are well studied,
but clustering with constraints is not

Capacitated clustering

Every center c has a maximal capacity $u(c)$ for assigned points

Basic clustering objectives are well studied,
but clustering with constraints is not

Capacitated clustering

Every center c has a maximal capacity $u(c)$ for assigned points

Lower bounded clustering

It is not worthwhile to open a center c unless a lower bound $\ell(c)$ is met

Basic clustering objectives are well studied,
but clustering with constraints is not

Capacitated clustering

Every center c has a maximal capacity $u(c)$ for assigned points

Lower bounded clustering

It is not worthwhile to open a center c unless a lower bound $\ell(c)$ is met

Clustering with outliers

Up to z outliers may be ignored for the cost

Basic clustering objectives are well studied,
but clustering with constraints is not

Capacitated clustering

Every center c has a maximal capacity $u(c)$ for assigned points

Lower bounded clustering

It is not worthwhile to open a center c unless a lower bound $\ell(c)$ is met

Clustering with outliers

Up to z outliers may be ignored for the cost

(Exact) Fair clustering

Assume points have a sensitive attribute.
Clusters shall have the same composition wrt this attribute as P.

Constraints make problems difficult

Capacitated facility location

Constraints make problems difficult

Capacitated facility location, all distances zero

Constraints make problems difficult

Capacitated facility location, all distances zero

Should be simple?

Constraints make problems difficult

Capacitated facility location, all distances zero

Should be simple?

- find facilities with $\geq n$ total capacity

Constraints make problems difficult

Capacitated facility location, all distances zero

Should be simple?

- find facilities with $\geq n$ total capacity
- minimize total opening cost

Constraints make problems difficult

Capacitated facility location, all distances zero

Should be simple?

- find facilities with $\geq n$ total capacity
- minimize total opening cost

$\rightsquigarrow$ Minimum Knapsack Problem

Constraints make problems difficult

Capacitated facility location, all distances zero

Should be simple?

- find facilities with $\geq n$ total capacity
- minimize total opening cost

$\rightsquigarrow$ Minimum Knapsack Problem

Standard LP: integrality gap

Constraints make problems difficult

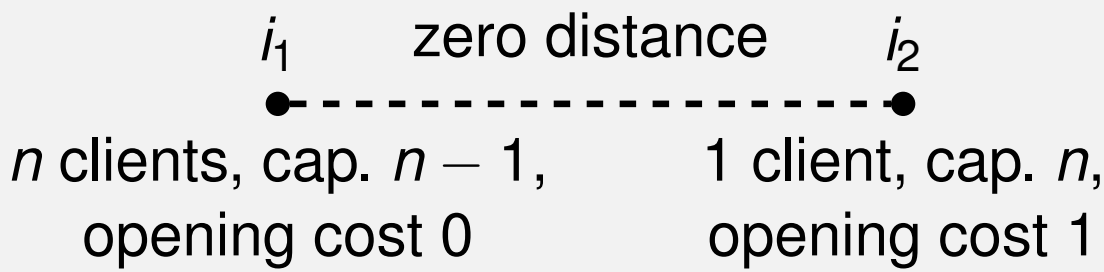
Capacitated facility location, all distances zero

Should be simple?

- find facilities with $\geq n$ total capacity
- minimize total opening cost

$\rightsquigarrow$ Minimum Knapsack Problem

Standard LP: integrality gap



Constraints make problems difficult

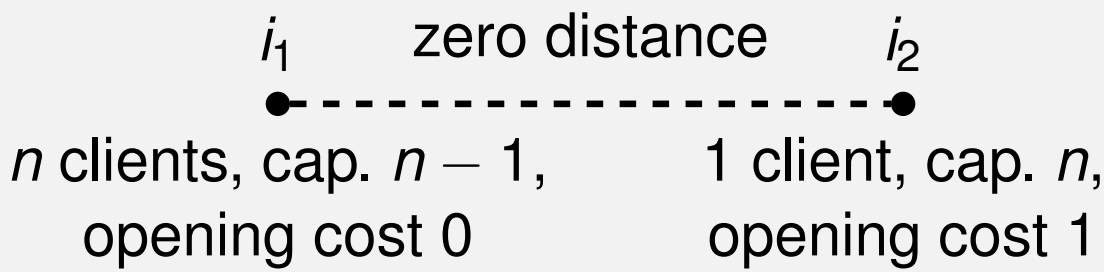
Capacitated facility location, all distances zero

Should be simple?

- find facilities with $\geq n$ total capacity
- minimize total opening cost

$\rightsquigarrow$ Minimum Knapsack Problem

Standard LP: integrality gap



- integral solution: Assign one client to i_2 $\rightsquigarrow$ cost 1
- fractional solution: Assign all clients to i_2 by $\frac{1}{n}$ $\rightsquigarrow$ cost $\frac{1}{n}$

Constraints make problems difficult

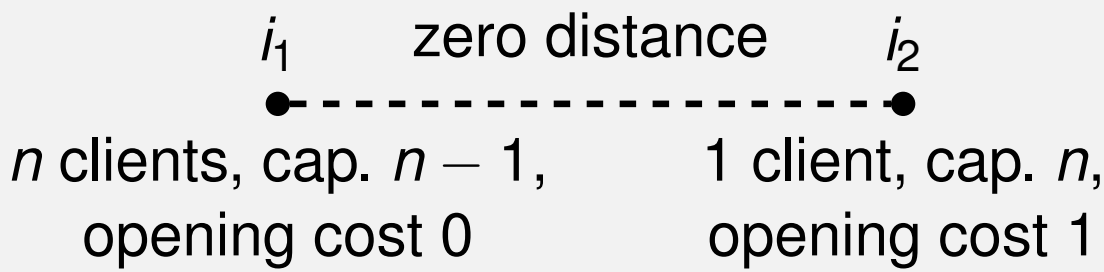
Capacitated facility location, all distances zero

Should be simple?

- find facilities with $\geq n$ total capacity
- minimize total opening cost

$\rightsquigarrow$ Minimum Knapsack Problem

Standard LP: integrality gap unbounded



- integral solution: Assign one client to i_2 $\rightsquigarrow$ cost 1
- fractional solution: Assign all clients to i_2 by $\frac{1}{n}$ $\rightsquigarrow$ cost $\frac{1}{n}$

Capacitated clustering: State of the art

Capacitated clustering: State of the art

- Local search based 5-approximation for cap. fac. location [BGG12]

[ABC+] An, Bhaskara, Chekuri, Gupta, Madan, Svensson. Centrality of trees for capacitated k-center, Math. Progr. 2015.
[ASS14] An, Singh, Svensson. LP-Based Algorithms for Capacitated Facility Location, FOCS 2014.
[BGG12] Bansal, Garg, Gupta. A 5-Approximation for Capacitated Facility Location, ESA 2012.
[CHK12] Cygan, Hajiaghayi, Khuller. LP rounding for k-centers with non-uniform hard capacities, FOCS 2012.

Capacitated clustering: State of the art

- Local search based 5-approximation for cap. fac. location [BGG12]
- $\mathcal{O}(1)$-approximation with knapsack based LP [ASS14]

[ABC+] An, Bhaskara, Chekuri, Gupta, Madan, Svensson. Centrality of trees for capacitated k-center, Math. Progr. 2015.
[ASS14] An, Singh, Svensson. LP-Based Algorithms for Capacitated Facility Location, FOCS 2014.
[BGG12] Bansal, Garg, Gupta. A 5-Approximation for Capacitated Facility Location, ESA 2012.
[CHK12] Cygan, Hajiaghayi, Khuller. LP rounding for k-centers with non-uniform hard capacities, FOCS 2012.

Capacitated clustering: State of the art

- Local search based 5-approximation for cap. fac. location [BGG12]
- $\mathcal{O}(1)$-approximation with knapsack based LP [ASS14]
- capacitated k-center: ≥ 3 [CHK12] and ≤ 9 [ABC+]

[ABC+] An, Bhaskara, Chekuri, Gupta, Madan, Svensson. Centrality of trees for capacitated k-center, Math. Progr. 2015.
[ASS14] An, Singh, Svensson. LP-Based Algorithms for Capacitated Facility Location, FOCS 2014.
[BGG12] Bansal, Garg, Gupta. A 5-Approximation for Capacitated Facility Location, ESA 2012.
[CHK12] Cygan, Hajiaghayi, Khuller. LP rounding for k-centers with non-uniform hard capacities, FOCS 2012.

Capacitated clustering: State of the art

- Local search based 5-approximation for cap. fac. location [BGG12]
- $\mathcal{O}(1)$-approximation with knapsack based LP [ASS14]
- capacitated k-center: ≥ 3 [CHK12] and ≤ 9 [ABC+]
- No constant factor known for capacitated k-median or k-means

[ABC+] An, Bhaskara, Chekuri, Gupta, Madan, Svensson. Centrality of trees for capacitated k-center, Math. Progr. 2015.
[ASS14] An, Singh, Svensson. LP-Based Algorithms for Capacitated Facility Location, FOCS 2014.
[BGG12] Bansal, Garg, Gupta. A 5-Approximation for Capacitated Facility Location, ESA 2012.
[CHK12] Cygan, Hajiaghayi, Khuller. LP rounding for k-centers with non-uniform hard capacities, FOCS 2012.

Capacitated clustering: State of the art

- Local search based 5-approximation for cap. fac. location [BGG12]
- $\mathcal{O}(1)$-approximation with knapsack based LP [ASS14]
- capacitated k-center: ≥ 3 [CHK12] and ≤ 9 [ABC+]
- No constant factor known for capacitated k-median or k-means
- Various bicriteria approximations

[ABC+] An, Bhaskara, Chekuri, Gupta, Madan, Svensson. Centrality of trees for capacitated k-center, Math. Progr. 2015.
[ASS14] An, Singh, Svensson. LP-Based Algorithms for Capacitated Facility Location, FOCS 2014.
[BGG12] Bansal, Garg, Gupta. A 5-Approximation for Capacitated Facility Location, ESA 2012.
[CHK12] Cygan, Hajiaghayi, Khuller. LP rounding for k-centers with non-uniform hard capacities, FOCS 2012.

Capacitated clustering: State of the art

- Local search based 5-approximation for cap. fac. location [BGG12]
- $\mathcal{O}(1)$-approximation with knapsack based LP [ASS14]
- capacitated k-center: ≥ 3 [CHK12] and ≤ 9 [ABC+]
- No constant factor known for capacitated k-median or k-means
- Various bicriteria approximations
- Research on easier variants: uniform or soft capacities

[ABC+] An, Bhaskara, Chekuri, Gupta, Madan, Svensson. Centrality of trees for capacitated k-center, Math. Progr. 2015.
[ASS14] An, Singh, Svensson. LP-Based Algorithms for Capacitated Facility Location, FOCS 2014.
[BGG12] Bansal, Garg, Gupta. A 5-Approximation for Capacitated Facility Location, ESA 2012.
[CHK12] Cygan, Hajiaghayi, Khuller. LP rounding for k-centers with non-uniform hard capacities, FOCS 2012.

Lower bounds are even worse..

Lower bounds are even worse..

- Standard LP for facility location has unbounded integrality gap

Lower bounds are even worse..

- Standard LP for facility location has unbounded integrality gap
- Standard local search approach yields arbitrarily bad solutions

Lower bounds are even worse..

- Standard LP for facility location has unbounded integrality gap
- Standard local search approach yields arbitrarily bad solutions

Lower bounds: State of the art

- 82.6-approximation for uniform lower bounds and facility loc. [AS12]

[AS12] Ahmadian, Swamy: Improved approximation guarantees for lower-bounded facility location, WAOA 2012.
[AS16] Ahmadian, Swamy: Approximation Algorithms for Clustering Problems with Lower Bounds and Outliers, ICALP 2016.

Lower bounds are even worse.. (except for k-center)

- Standard LP for facility location has unbounded integrality gap
- Standard local search approach yields arbitrarily bad solutions

Lower bounds: State of the art

- 82.6-approximation for uniform lower bounds and facility loc. [AS12]
- $\mathcal{O}(1)$ for non-uniform bounds only known for k-center [AS16]

[AS12] Ahmadian, Swamy: Improved approximation guarantees for lower-bounded facility location, WAOA 2012.
[AS16] Ahmadian, Swamy: Approximation Algorithms for Clustering Problems with Lower Bounds and Outliers, ICALP 2016.

The happy world of k-center 

State of the art for k-center

- just k-center: 2-approx. [G85][HS86]

The happy world of k-center 

State of the art for k-center

- just k-center: 2-approx. [G85][HS86]
- capacitated k-center: 9-approx. [ABCGMS15]

[ABCGMS15] An, Bhaskara, Chekuri, Gupta, Madan, Svensson: Centrality of trees for capacitated k-center. Math. Prog. 2015.
[CGK16] Chakrabarty, Goyal, Krishnaswamy. The non-uniform k-center problem, ICALP 2016.
[CKLV17] Chierichetti, Kumar, Lattanzi, Vassilvitskii. Fair clustering through fairlets, NIPS 2017.

The happy world of k-center 

State of the art for k-center

- just k-center: 2-approx. [G85][HS86]
- capacitated k-center: 9-approx. [ABCGMS15]
- lower bounded k-center: 2-approx. [CGK16]

[ABCGMS15] An, Bhaskara, Chekuri, Gupta, Madan, Svensson: Centrality of trees for capacitated k-center. Math. Prog. 2015.
[CGK16] Chakrabarty, Goyal, Krishnaswamy. The non-uniform k-center problem, ICALP 2016.
[CKLV17] Chierichetti, Kumar, Lattanzi, Vassilvitskii. Fair clustering through fairlets, NIPS 2017.

The happy world of k-center 

State of the art for k-center

- just k-center: 2-approx. [G85][HS86]
- capacitated k-center: 9-approx. [ABCGMS15]
- lower bounded k-center: 2-approx. [CGK16]
- k-center with outliers: 2-approx. [CGK16]

[ABCGMS15] An, Bhaskara, Chekuri, Gupta, Madan, Svensson: Centrality of trees for capacitated k-center. Math. Prog. 2015.
[CGK16] Chakrabarty, Goyal, Krishnaswamy. The non-uniform k-center problem, ICALP 2016.
[CKLV17] Chierichetti, Kumar, Lattanzi, Vassilvitskii. Fair clustering through fairlets, NIPS 2017.

The happy world of k-center 

State of the art for k-center

- just k-center: 2-approx. [G85][HS86]
- capacitated k-center: 9-approx. [ABCGMS15]
- lower bounded k-center: 2-approx. [CGK16]
- k-center with outliers: 2-approx. [CGK16]
- (exact and balanced) fair k-center with 2 colors: 3-approx. [CKLV17]

[ABCGMS15] An, Bhaskara, Chekuri, Gupta, Madan, Svensson: Centrality of trees for capacitated k-center. Math. Prog. 2015.
[CGK16] Chakrabarty, Goyal, Krishnaswamy. The non-uniform k-center problem, ICALP 2016.
[CKLV17] Chierichetti, Kumar, Lattanzi, Vassilvitskii. Fair clustering through fairlets, NIPS 2017.

The happy world of k-center 

State of the art for k-center

- just k-center: 2-approx. [G85][HS86]
- capacitated k-center: 9-approx. [ABCGMS15]
- lower bounded k-center: 2-approx. [CGK16]
- k-center with outliers: 2-approx. [CGK16]
- (exact and balanced) fair k-center with 2 colors: 3-approx. [CKLV17]
- (exact) fair k-center with multiple colors: 14-approx.

[ABCGMS15] An, Bhaskara, Chekuri, Gupta, Madan, Svensson: Centrality of trees for capacitated k-center. Math. Prog. 2015.
[CGK16] Chakrabarty, Goyal, Krishnaswamy. The non-uniform k-center problem, ICALP 2016.
[CKLV17] Chierichetti, Kumar, Lattanzi, Vassilvitskii. Fair clustering through fairlets, NIPS 2017.

The happy world of k-center 

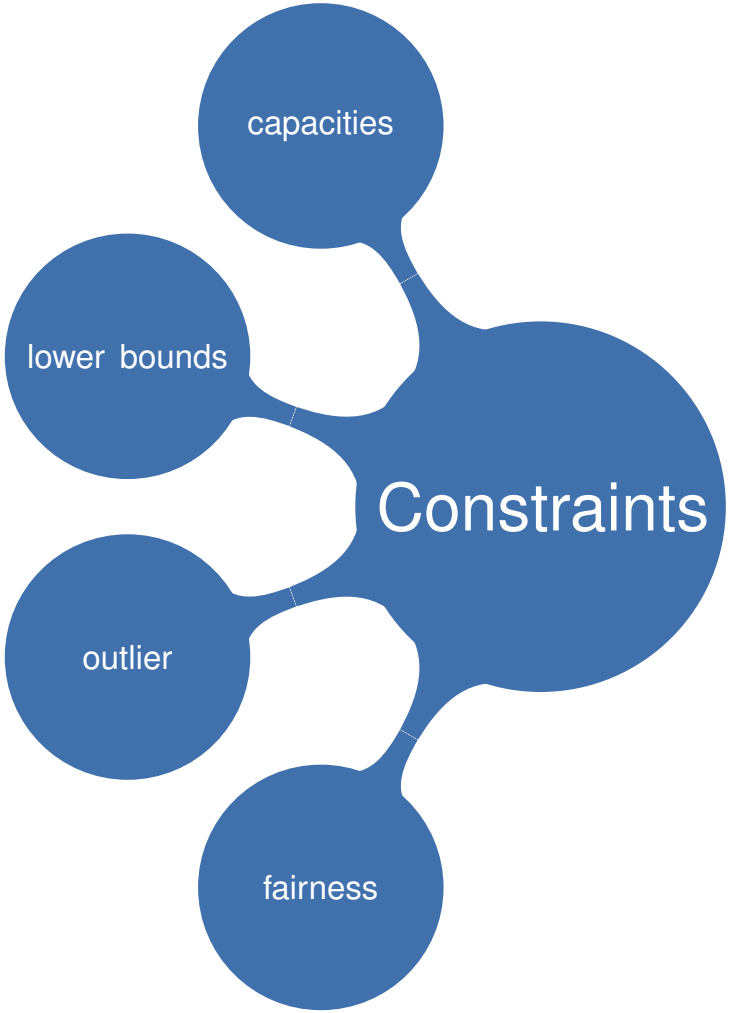
State of the art for k-center

- just k-center: 2-approx. [G85][HS86]
- capacitated k-center: 9-approx. [ABCGMS15]
- lower bounded k-center: 2-approx. [CGK16]
- k-center with outliers: 2-approx. [CGK16]
- (exact and balanced) fair k-center with 2 colors: 3-approx. [CKLV17]
- (exact) fair k-center with multiple colors: 14-approx.

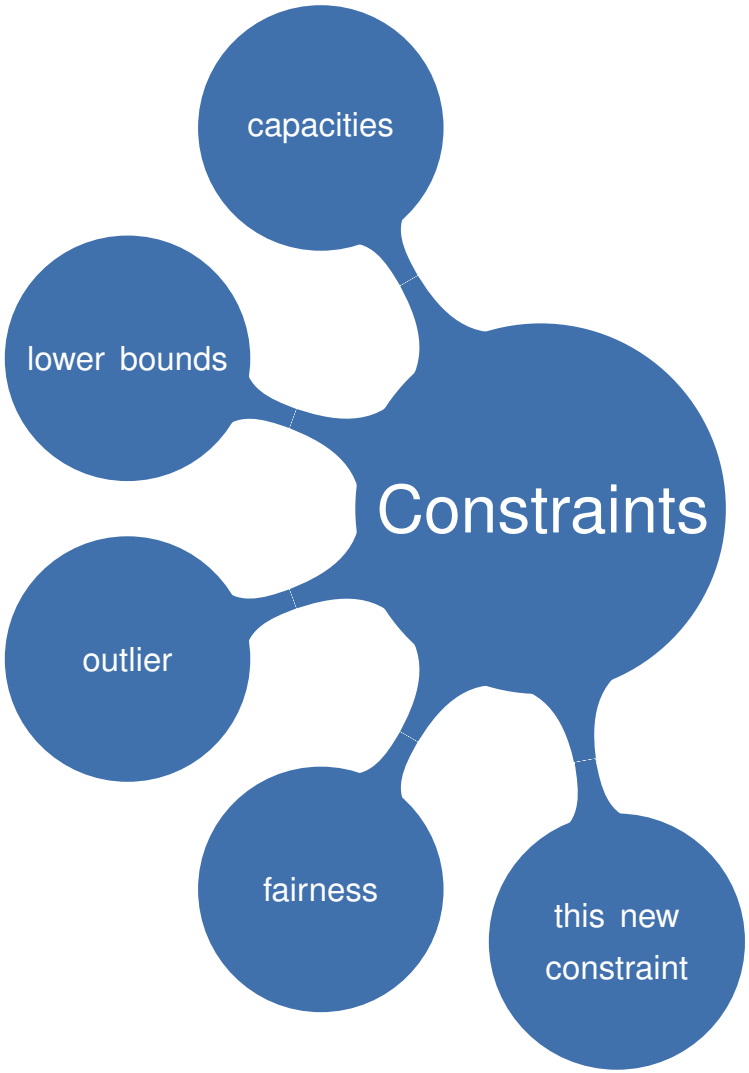
[ABCGMS15] An, Bhaskara, Chekuri, Gupta, Madan, Svensson: Centrality of trees for capacitated k-center. Math. Prog. 2015.
[CGK16] Chakrabarty, Goyal, Krishnaswamy. The non-uniform k-center problem, ICALP 2016.
[CKLV17] Chierichetti, Kumar, Lattanzi, Vassilvitskii. Fair clustering through fairlets, NIPS 2017.

Solve new problems on k-center first.

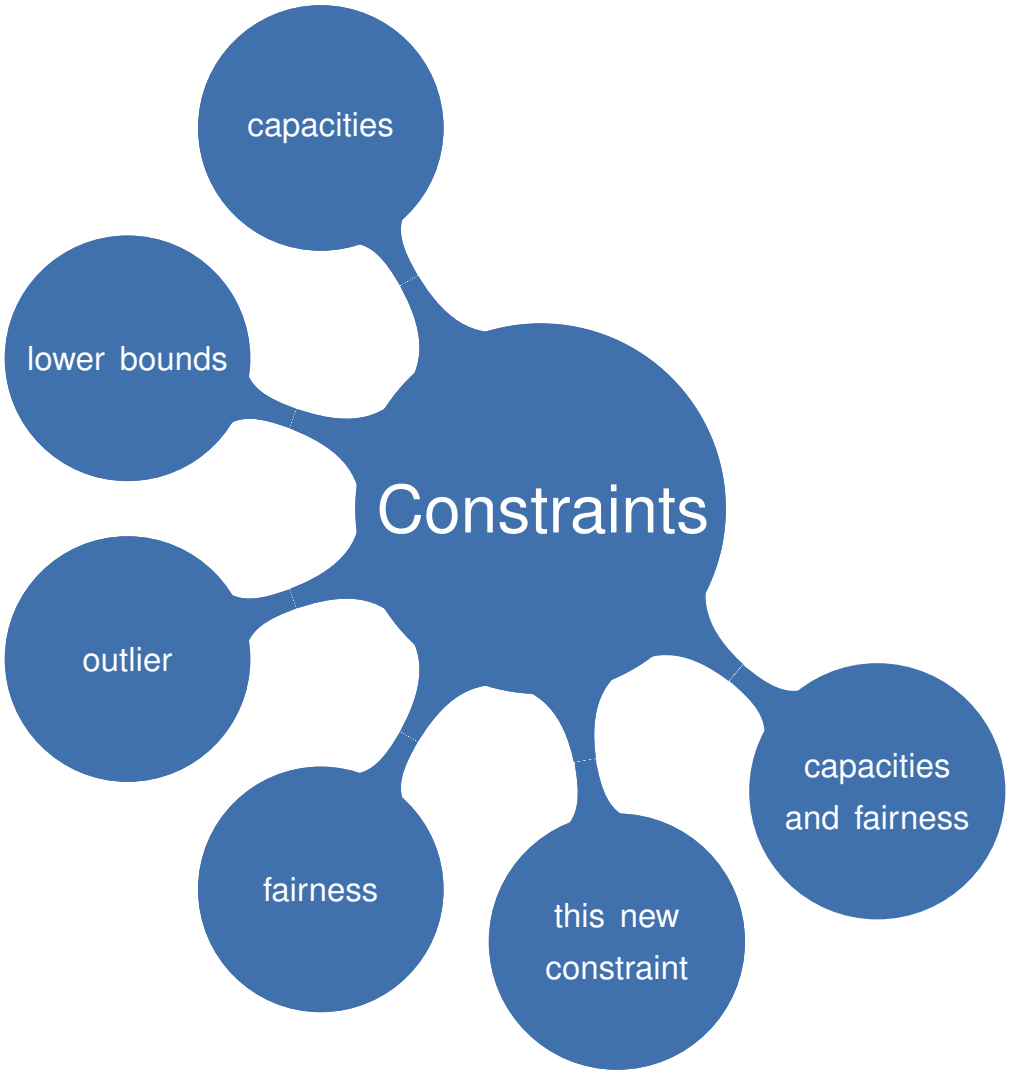
Do we want to track all these problems?



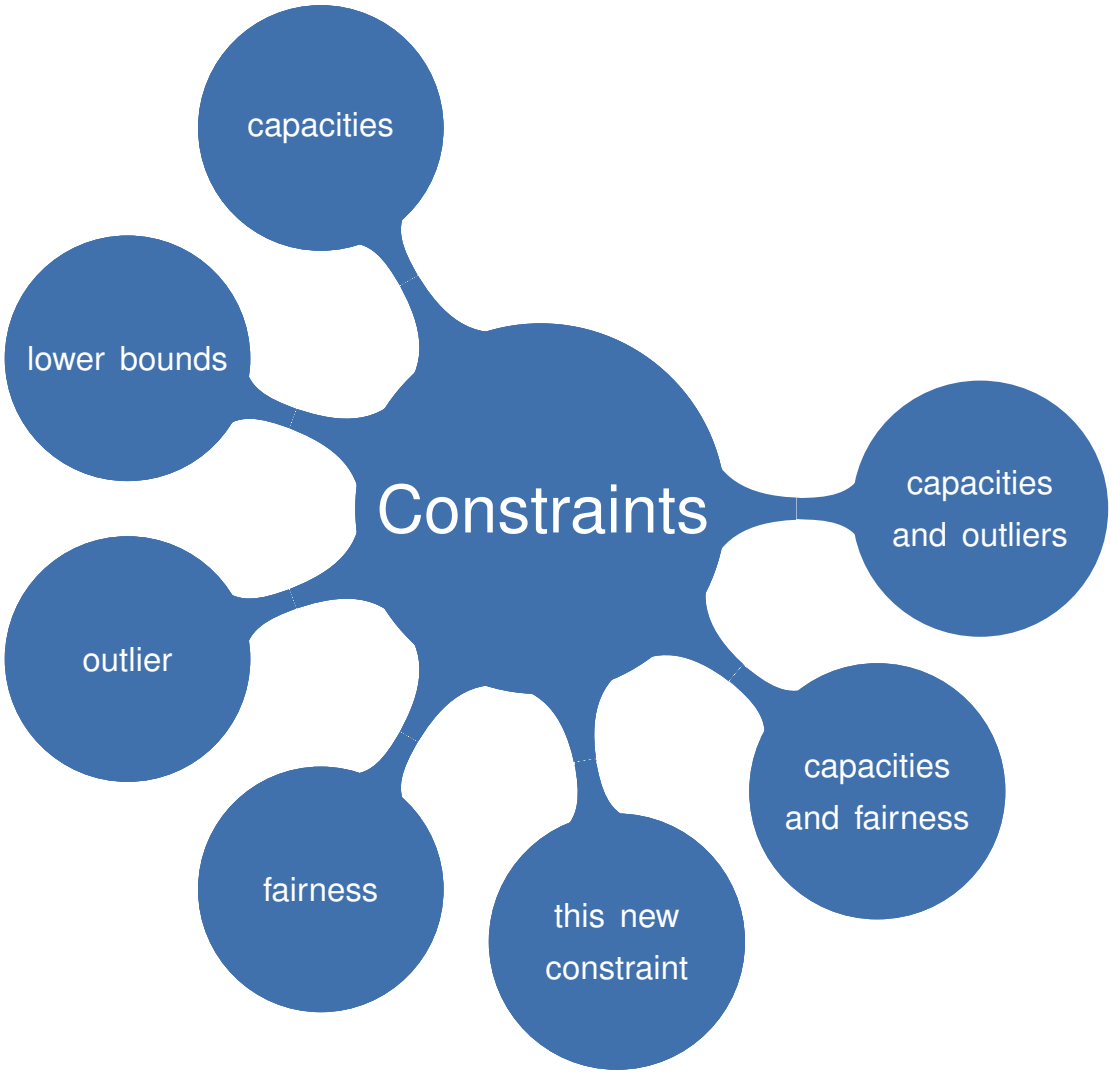
Do we want to track all these problems?



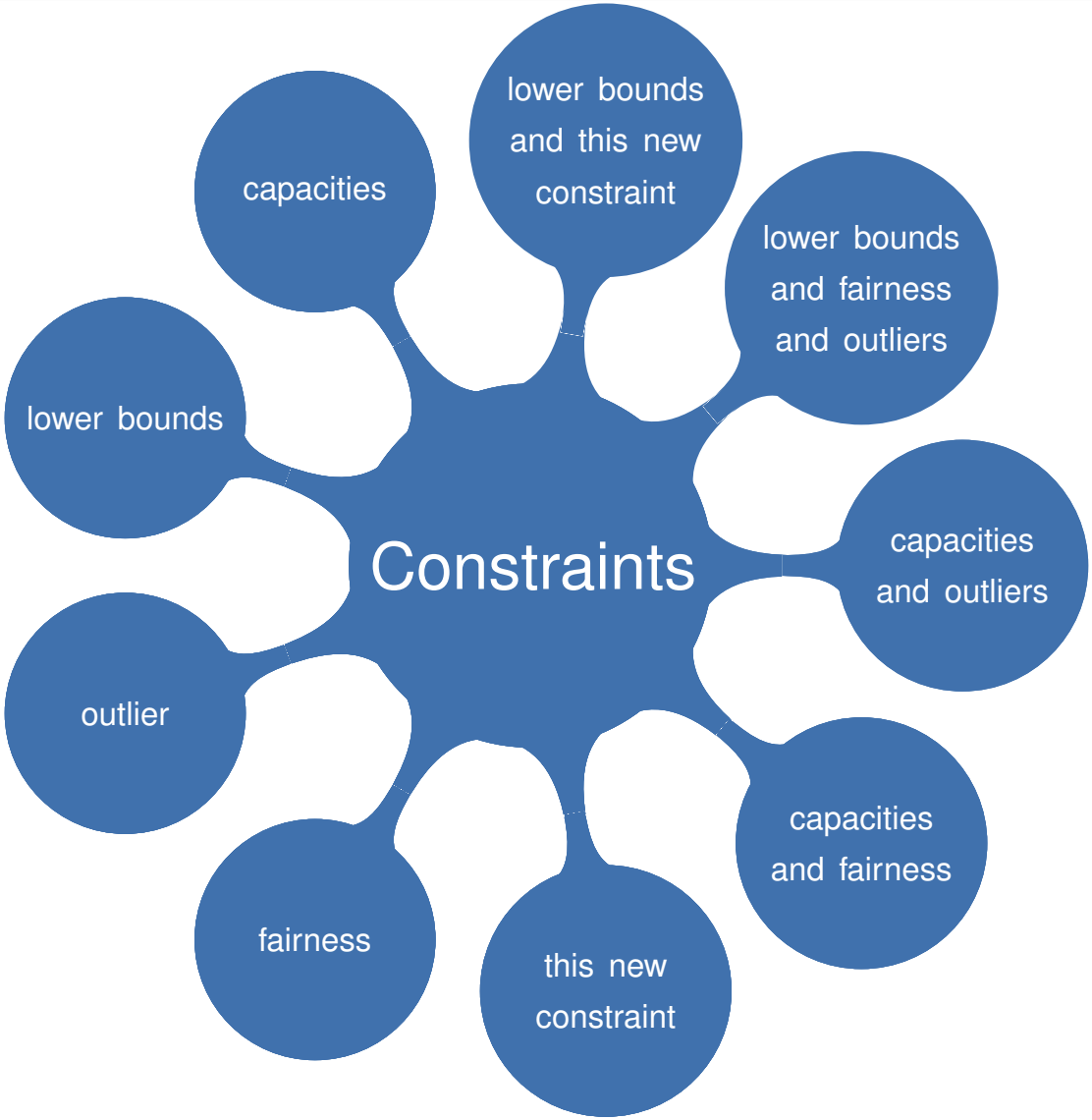
Do we want to track all these problems?



Do we want to track all these problems?



Do we want to track all these problems?



Adding constraints

Adding constraints

- Approximation ratios have been optimized for centuries
- Constraints require new techniques to transfer results
- What if a better algorithm for the unconstrained problem appears?
- And what if we want combinations of constraints?

Adding constraints

- Approximation ratios have been optimized for centuries
- Constraints require new techniques to transfer results
- What if a better algorithm for the unconstrained problem appears?
- And what if we want combinations of constraints?

$\rightsquigarrow$ endless chain of improvements and new problem combinations

Adding constraints

- Approximation ratios have been optimized for centuries
- Constraints require new techniques to transfer results
- What if a better algorithm for the unconstrained problem appears?
- And what if we want combinations of constraints?

$\rightsquigarrow$ endless chain of improvements and new problem combinations

Question

Can we add constraints?

Adding constraints

- Approximation ratios have been optimized for centuries
- Constraints require new techniques to transfer results
- What if a better algorithm for the unconstrained problem appears?
- And what if we want combinations of constraints?

$\rightsquigarrow$ endless chain of improvements and new problem combinations

Question

Can we add constraints?

Given an approximation algorithm as a subroutine,
can we establish an additional constraint?

Let's do that for k-center

Let's do that for k-center

Goal

Add uniform lower bound L to k-center with outliers

Let's do that for k-center

Goal

Add uniform lower bound L to k-center with outliers

What makes k-center easy?

Let's do that for k-center

Goal

Add uniform lower bound L to k-center with outliers

What makes k-center easy?

Threshold graph

Let's do that for k-center

Goal

Add uniform lower bound L to k-center with outliers

What makes k-center easy?

Threshold graph

- Value of the optimal solution is a pairwise distance!

Let's do that for k-center

Goal

Add uniform lower bound L to k-center with outliers

What makes k-center easy?

Threshold graph

- Value of the optimal solution is a pairwise distance!
- There are $\Theta(n^2)$ pairwise distances $\rightsquigarrow$ can guess optimum value τ

Let's do that for k-center

Goal

Add uniform lower bound L to k-center with outliers

What makes k-center easy?

Threshold graph

- Value of the optimal solution is a pairwise distance!
- There are $\Theta(n^2)$ pairwise distances $\rightsquigarrow$ can guess optimum value τ
- Allows to build threshold graph

Let's do that for k-center

Goal

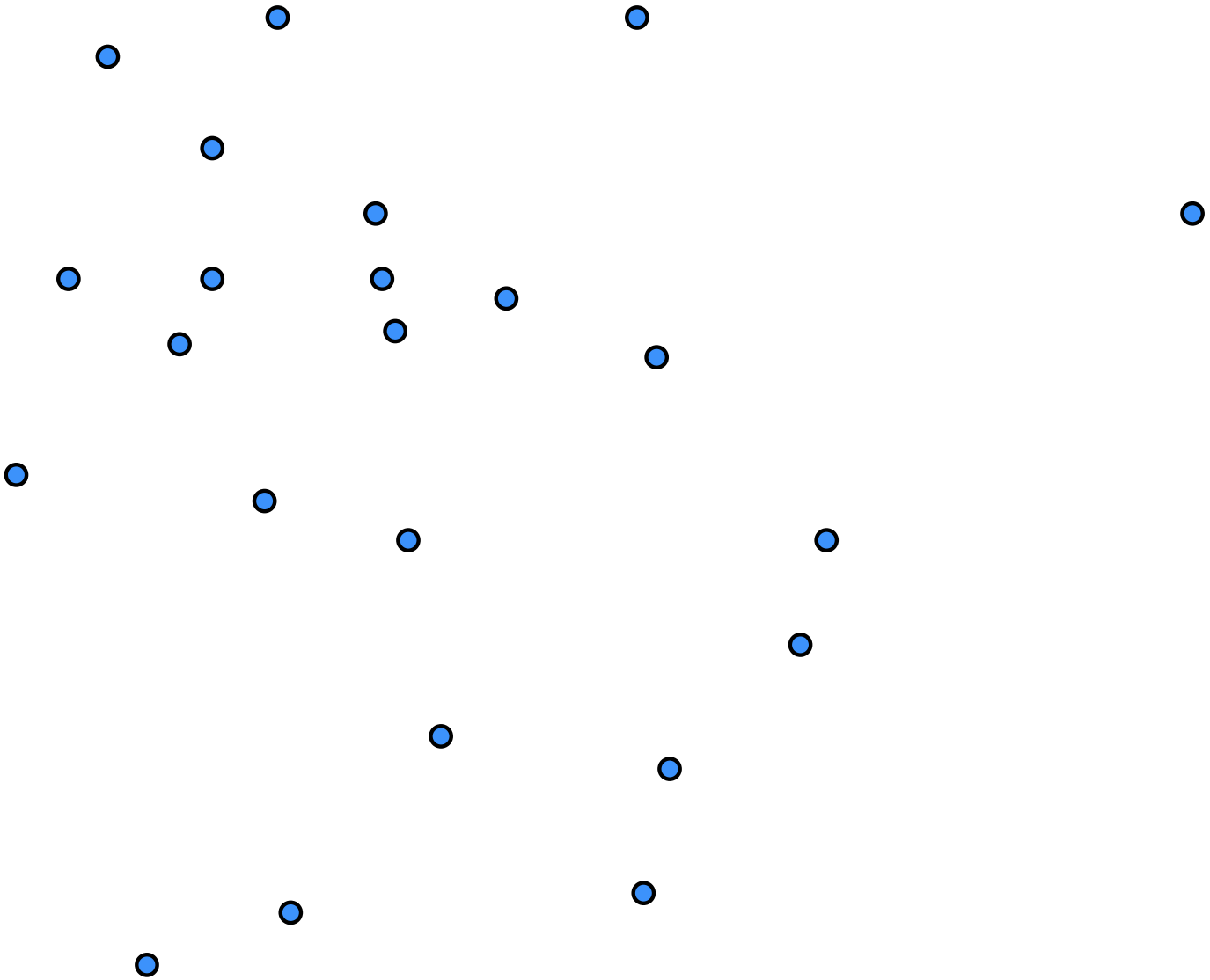
Add uniform lower bound L to k-center with outliers

What makes k-center easy?

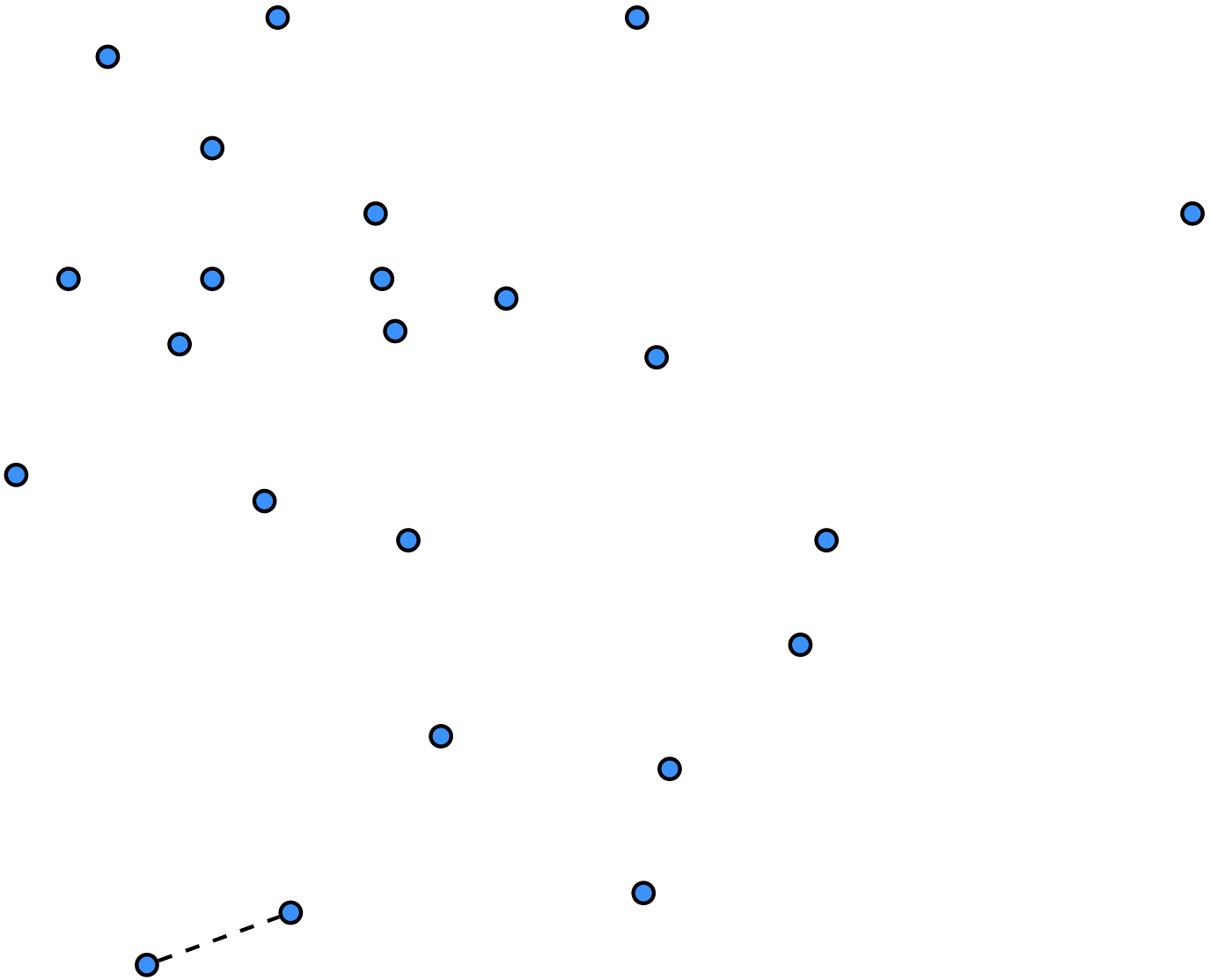
Threshold graph

- Value of the optimal solution is a pairwise distance!
- There are $\Theta(n^2)$ pairwise distances $\rightsquigarrow$ can guess optimum value τ
- Allows to build threshold graph
- At most two hops between two points in the same optimum cluster

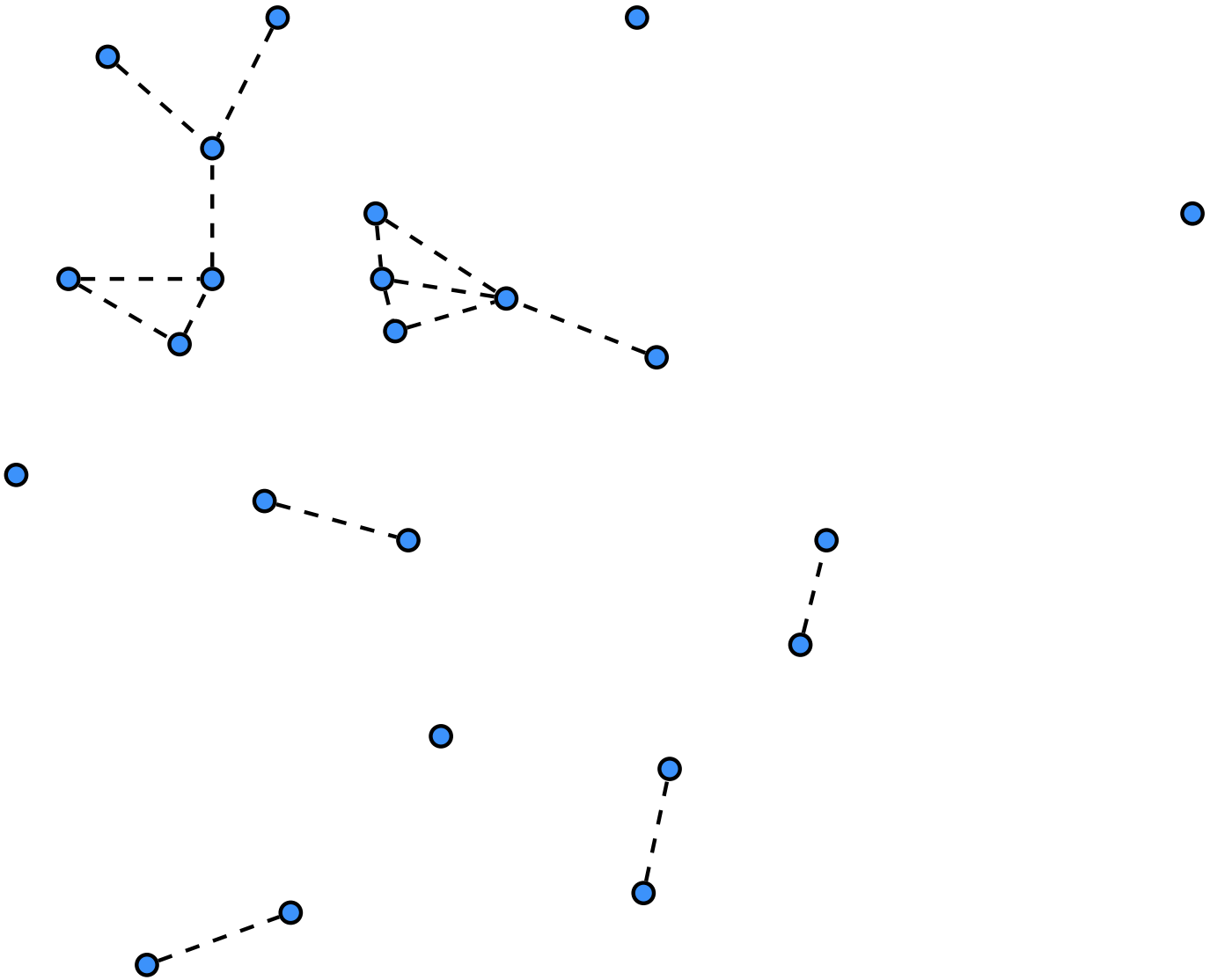
Threshold graph



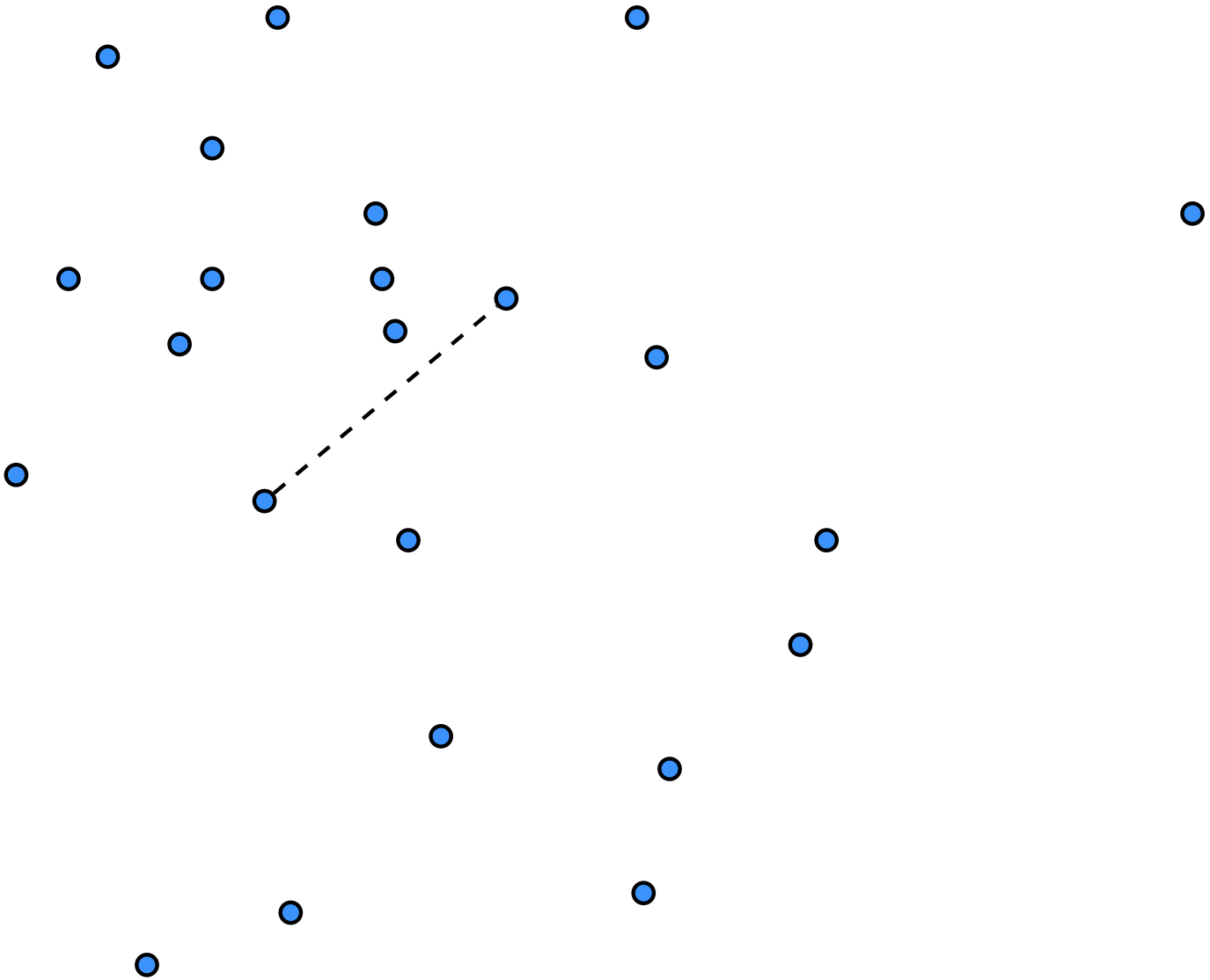
Threshold graph



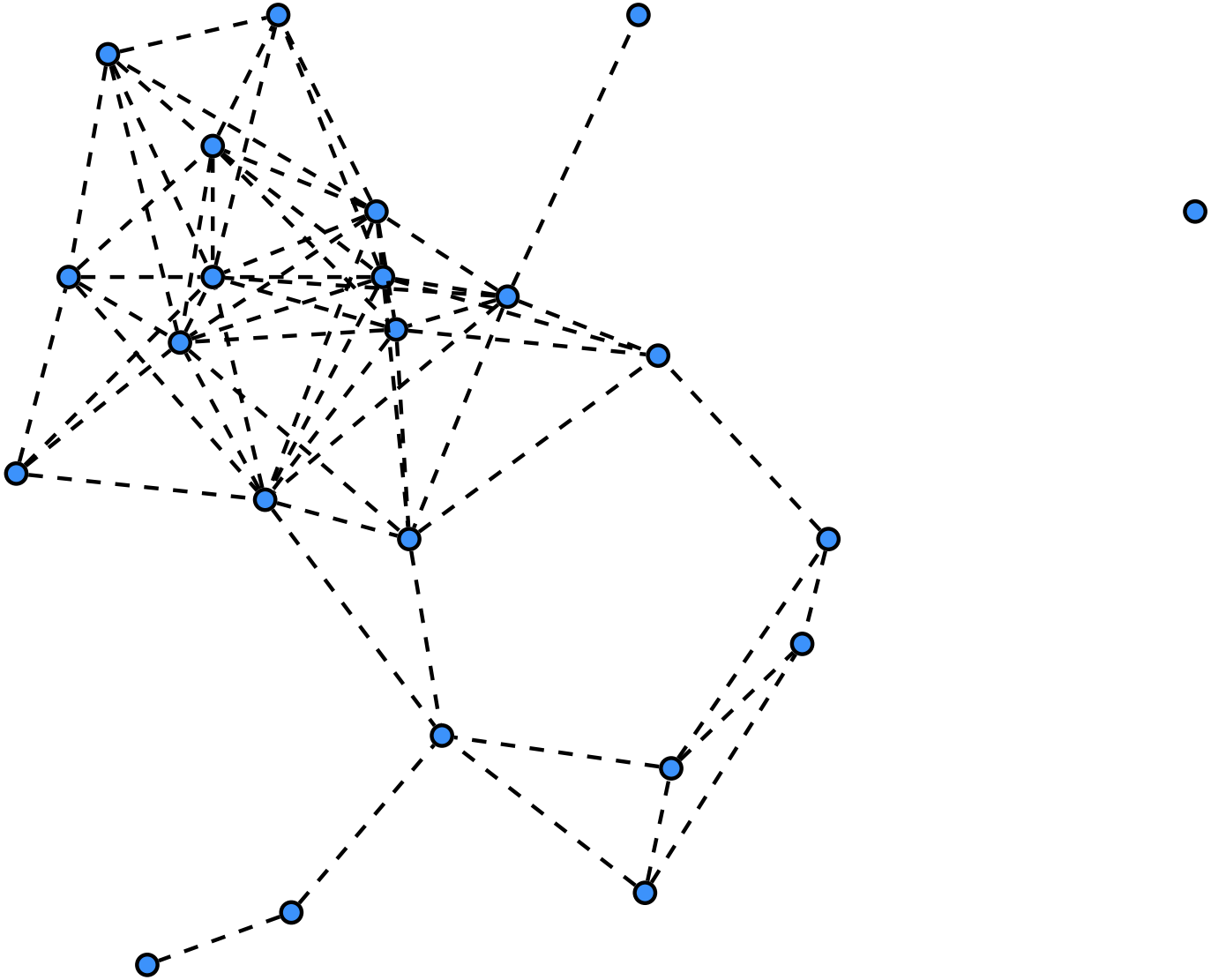
Threshold graph



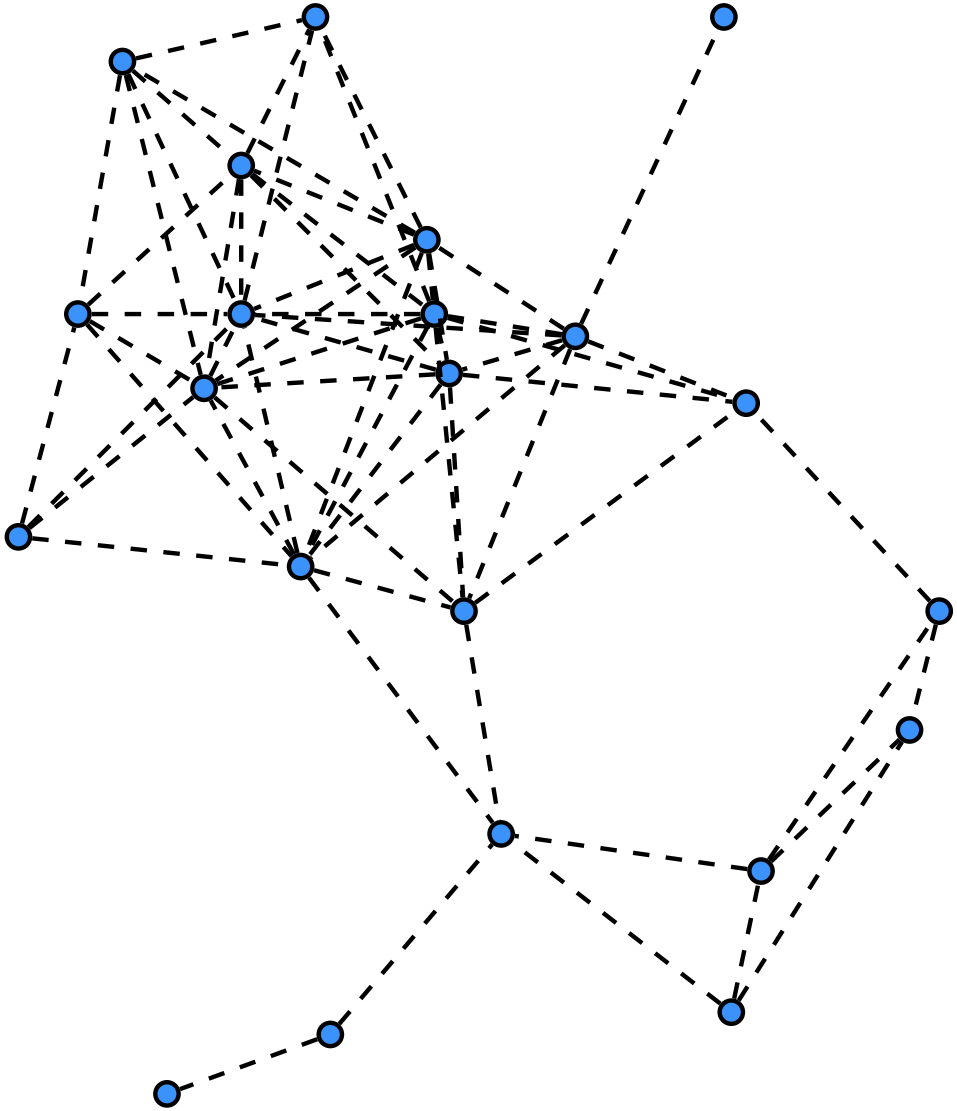
Threshold graph



Threshold graph

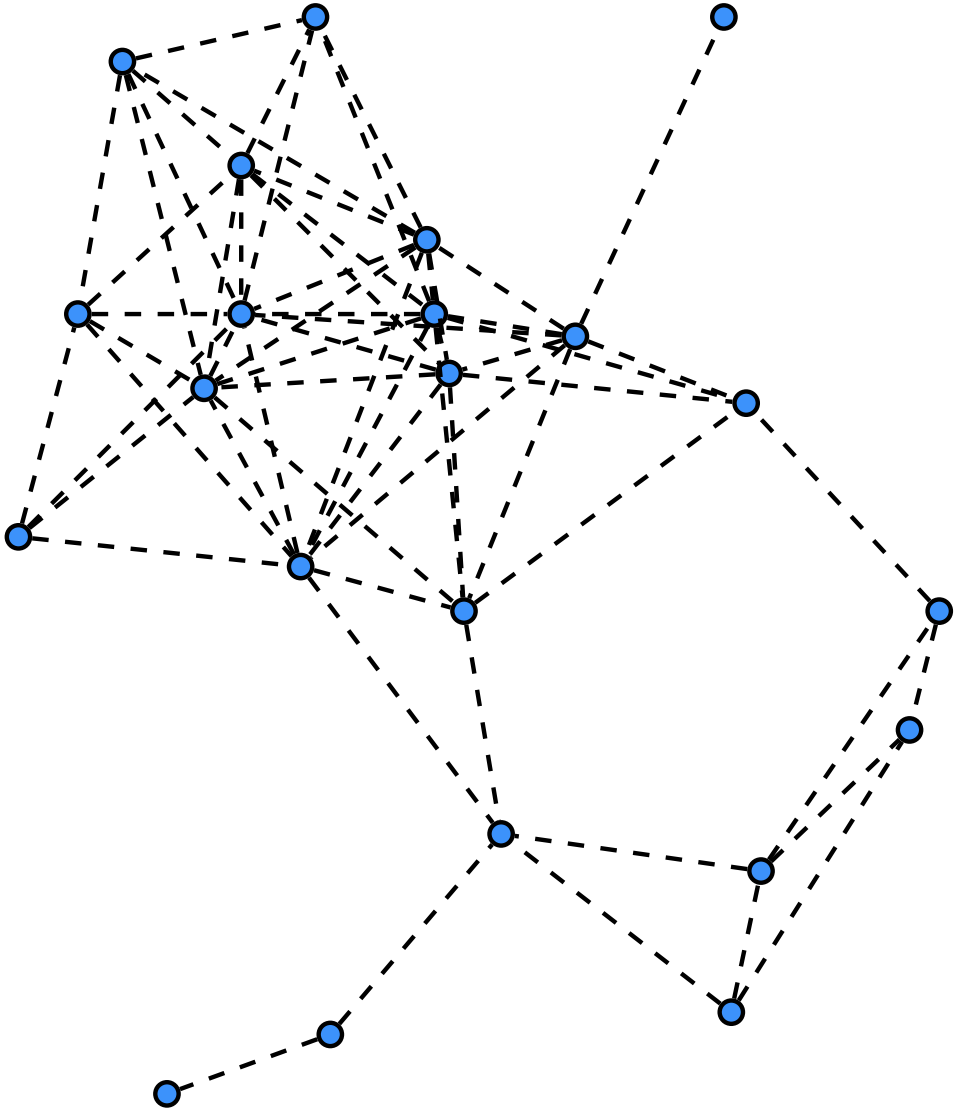


Threshold graph



Approximation in [HS86]

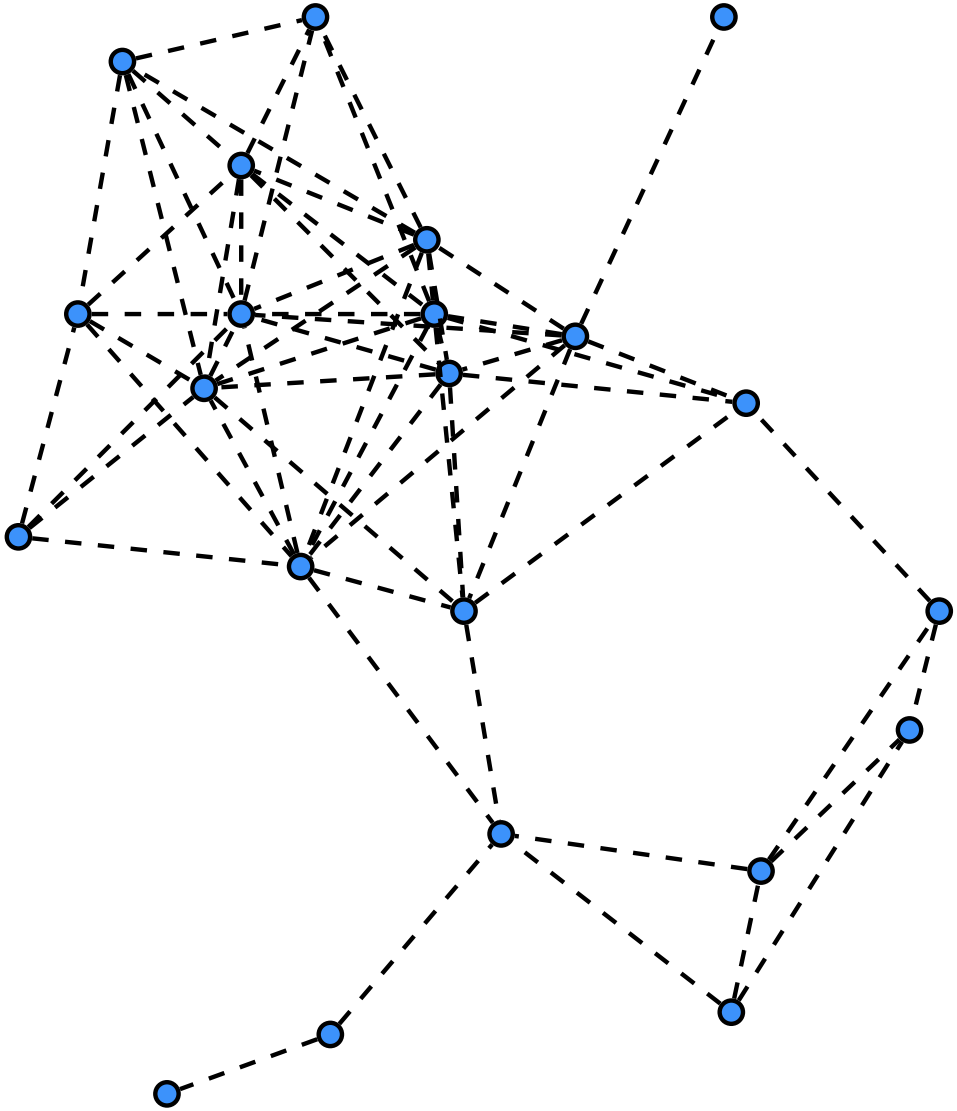
Threshold graph



Approximation in [HS86]

- Guess correct threshold

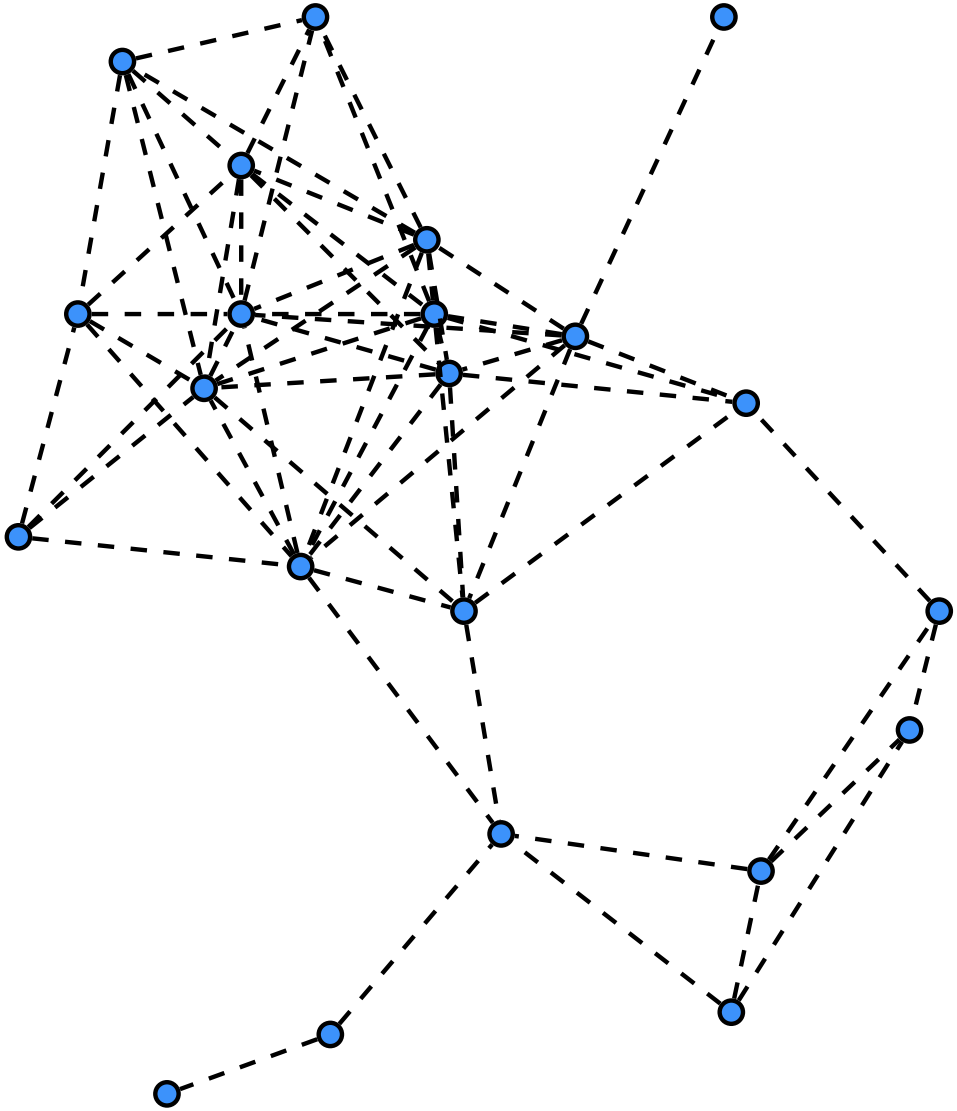
Threshold graph



Approximation in [HS86]

- Guess correct threshold
- Consider two-hop graph

Threshold graph



Approximation in [HS86]

- Guess correct threshold
- Consider two-hop graph
- Find maximal independent set

Goal

Add uniform lower bound L to k-center with outliers

Goal

Add uniform lower bound L to k-center with outliers

Main idea

Goal

Add uniform lower bound L to k-center with outliers

Main idea

- Use subroutine for approximating k-center with outliers

Goal

Add uniform lower bound L to k-center with outliers

Main idea

- Use subroutine for approximating k-center with outliers
- Build a network to move points using the threshold idea

Goal

Add uniform lower bound L to k-center with outliers

Main idea

- Use subroutine for approximating k-center with outliers
- Build a network to move points using the threshold idea
- One of two outcomes:

Goal

Add uniform lower bound L to k-center with outliers

Main idea

- Use subroutine for approximating k-center with outliers
- Build a network to move points using the threshold idea
- One of two outcomes:
  - We can successfully establish the lower bound L or

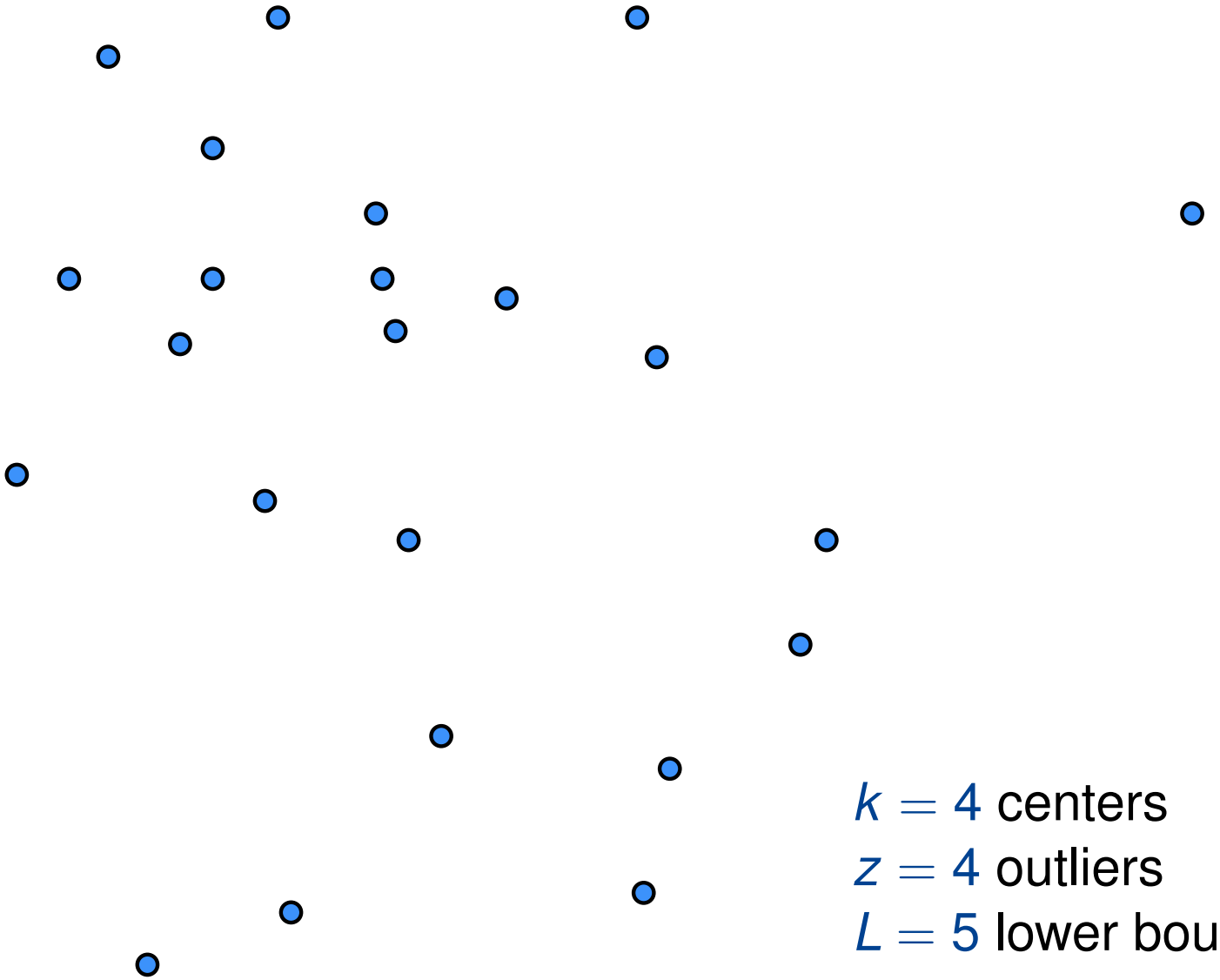
Goal

Add uniform lower bound L to k-center with outliers

Main idea

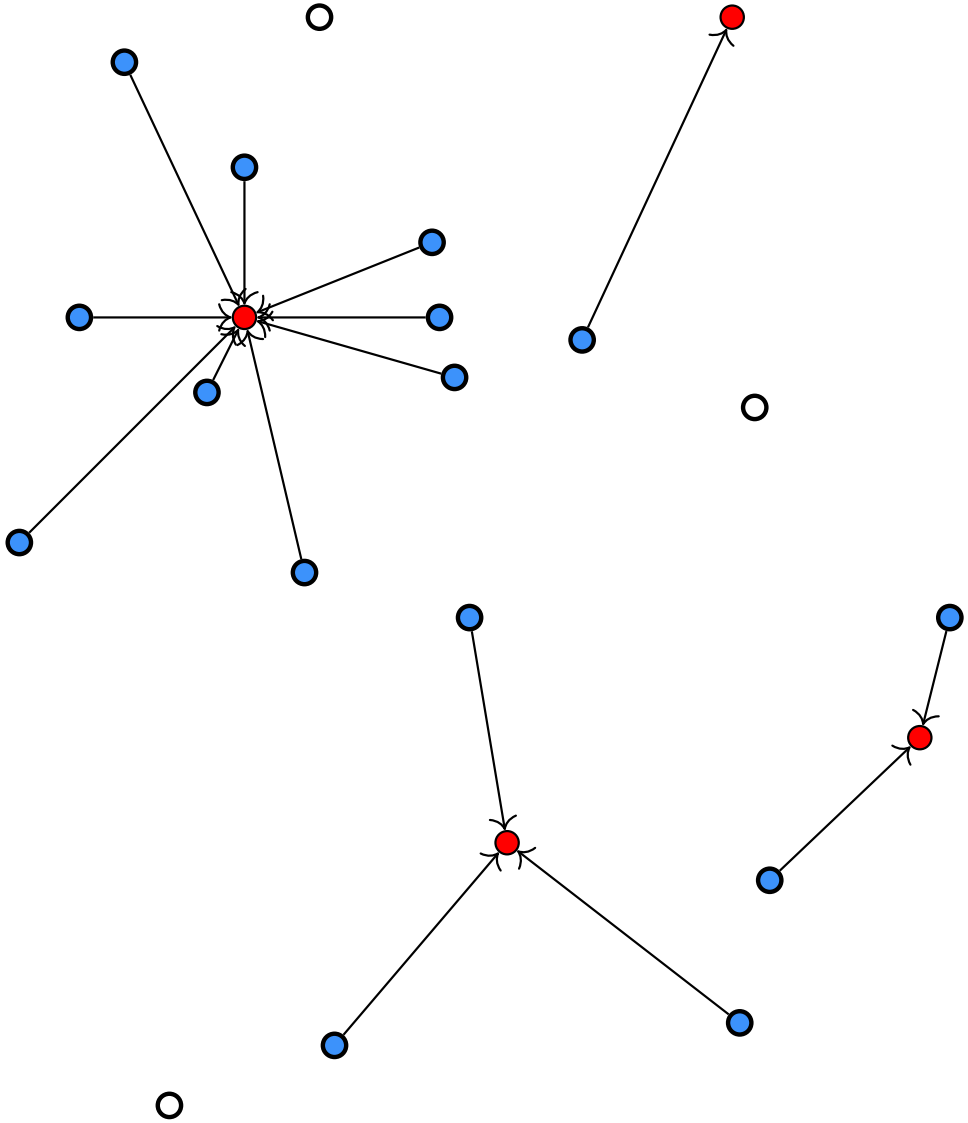
- Use subroutine for approximating k-center with outliers
- Build a network to move points using the threshold idea
- One of two outcomes:
  - We can successfully establish the lower bound L or
  - We find a set P' that we recluster

Step 1: Use subroutine to get initial solution



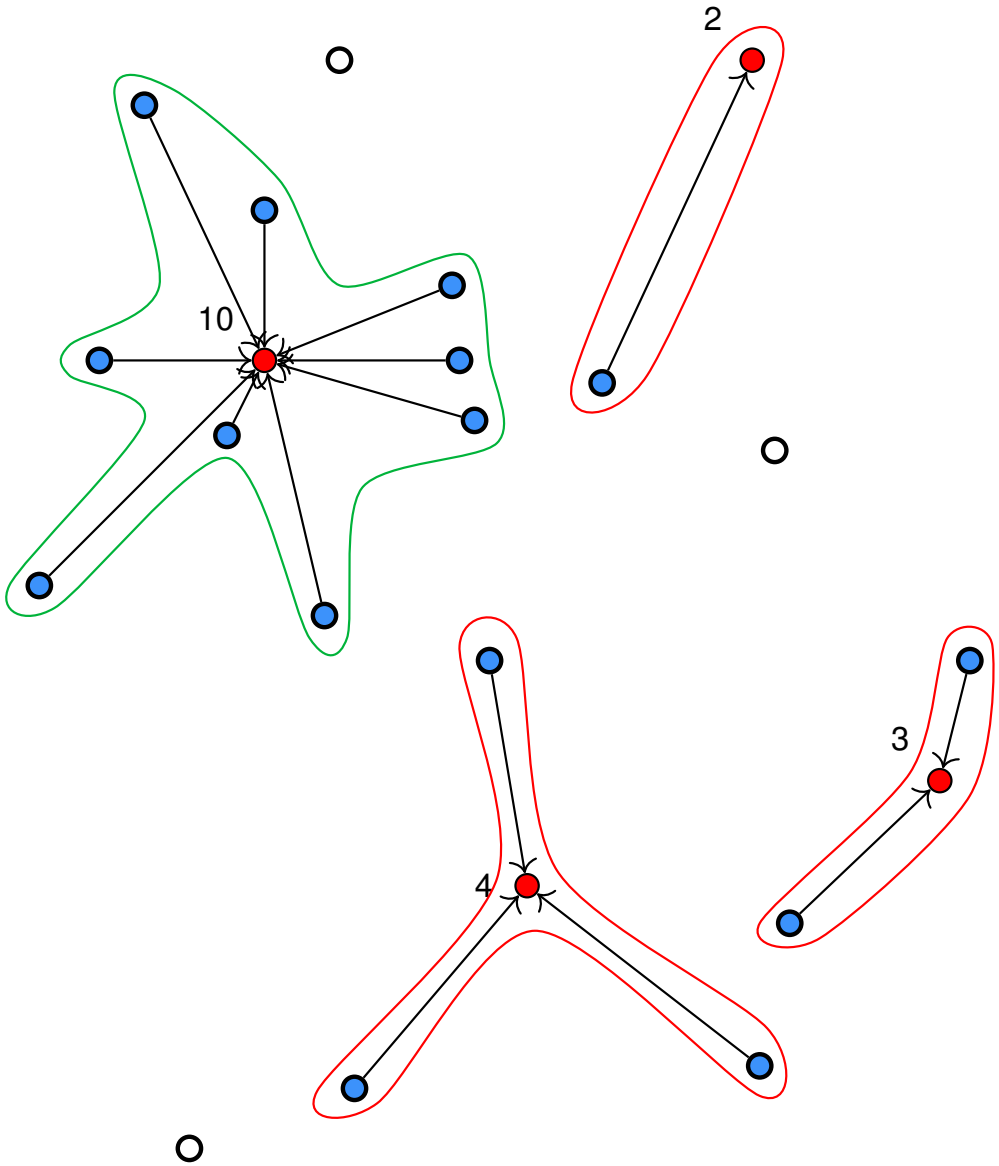
$k = 4$ centers
$z = 4$ outliers
$L = 5$ lower bound

Step 1: Use subroutine to get initial solution



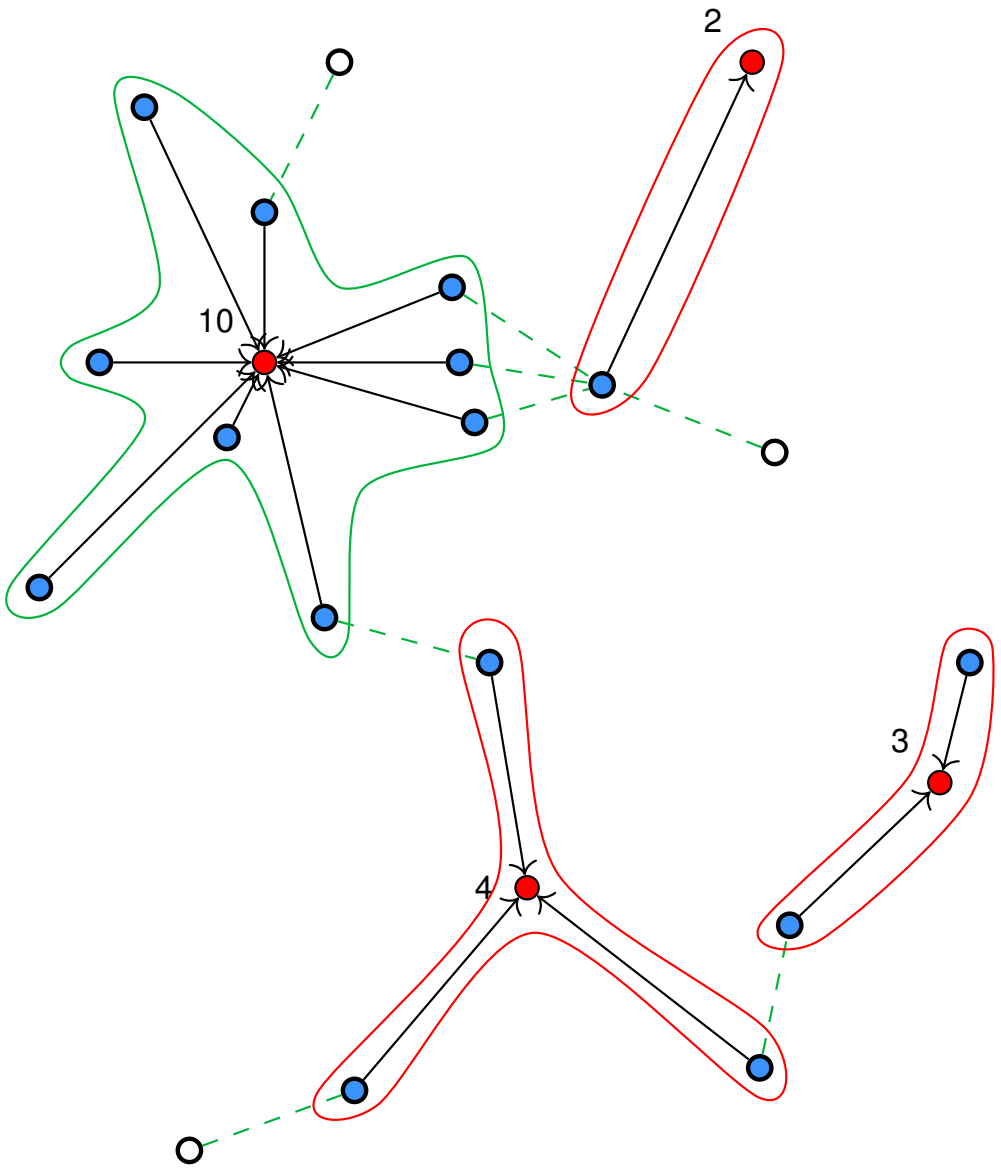
$k = 4$ centers
$z = 4$ outliers
$L = 5$ lower bound

Step 1: Use subroutine to get initial solution



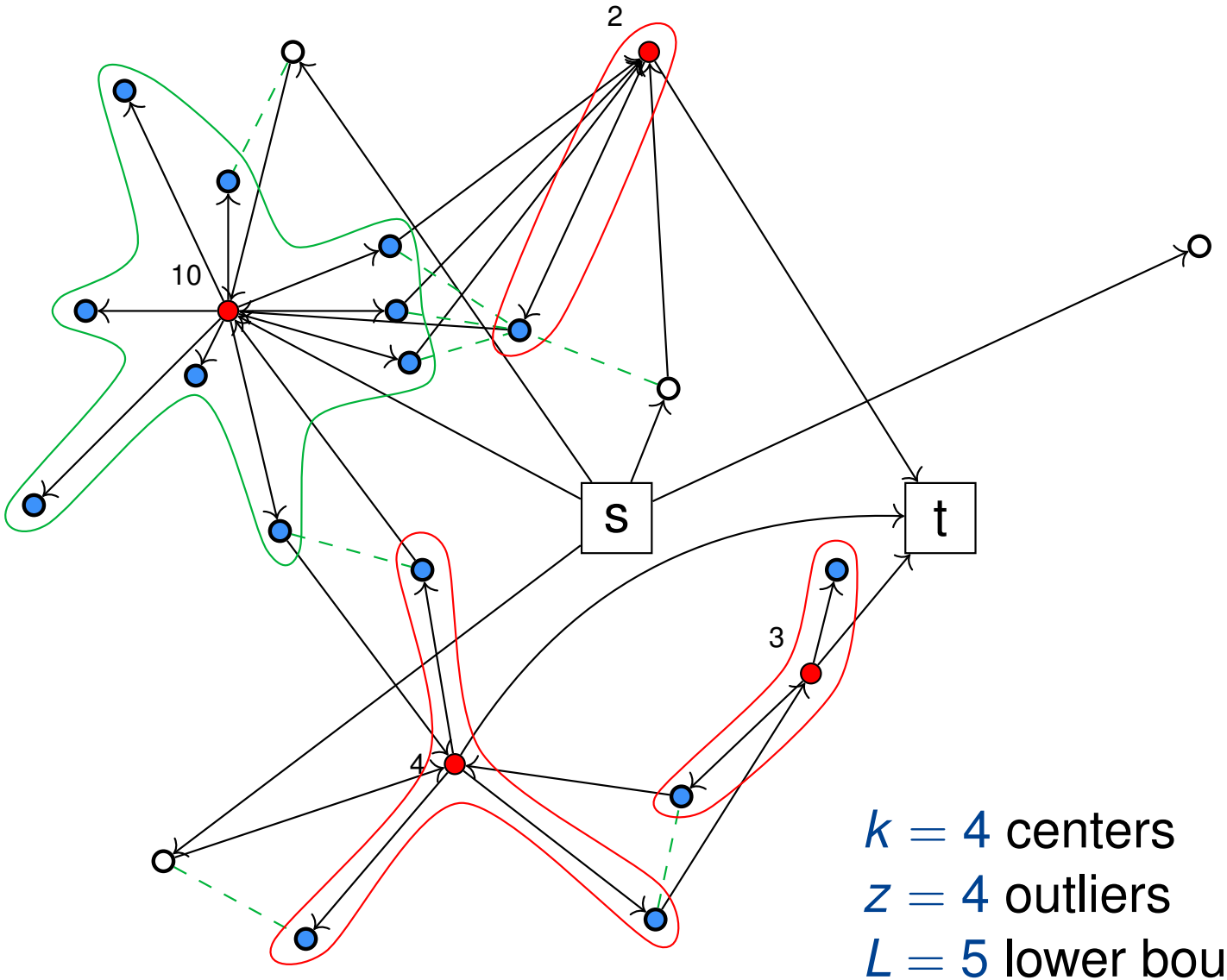
$k = 4$ centers
$z = 4$ outliers
$L = 5$ lower bound

Step 2: Compute threshold edges



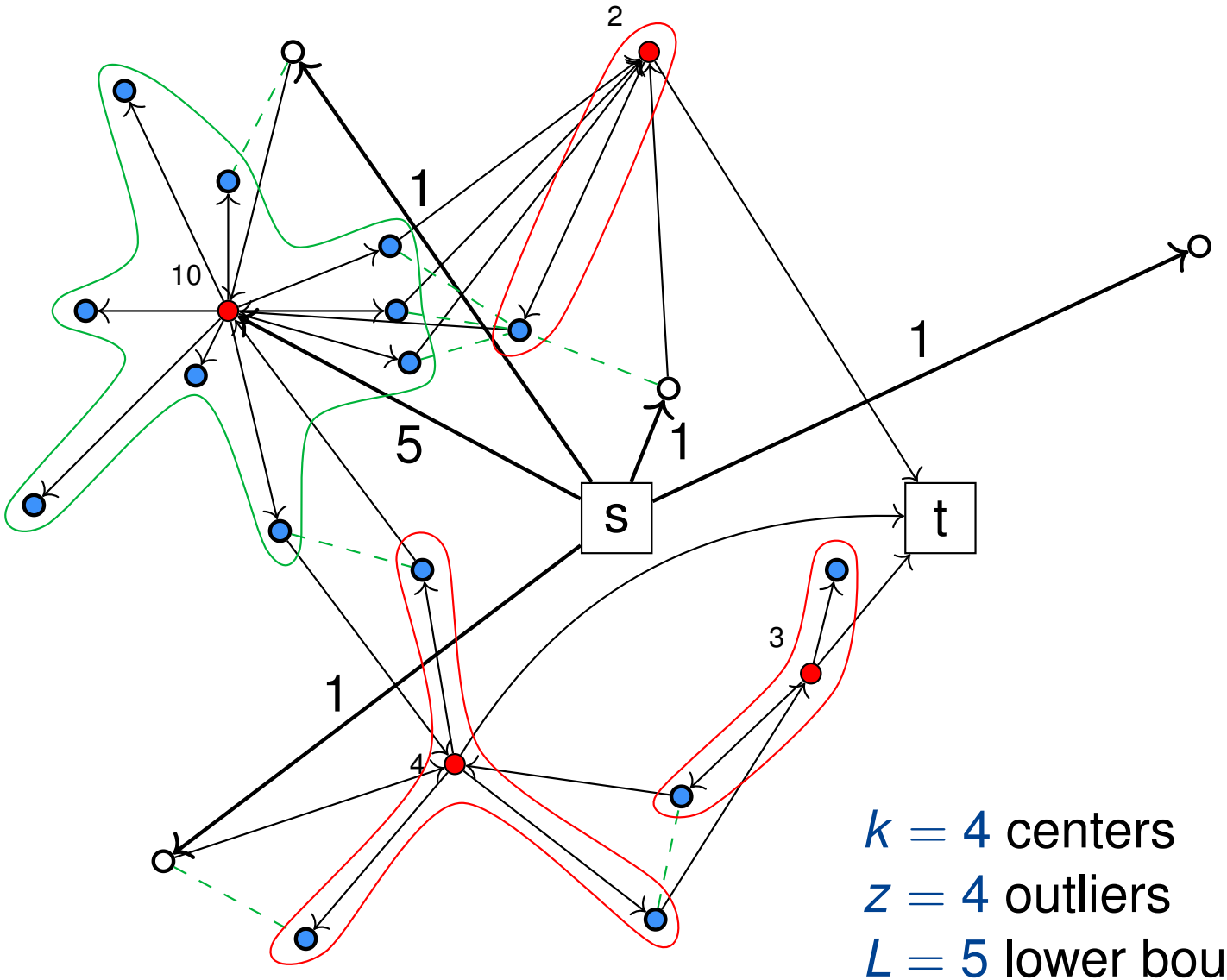
$k = 4$ centers
$z = 4$ outliers
$L = 5$ lower bound

Step 3: Build network



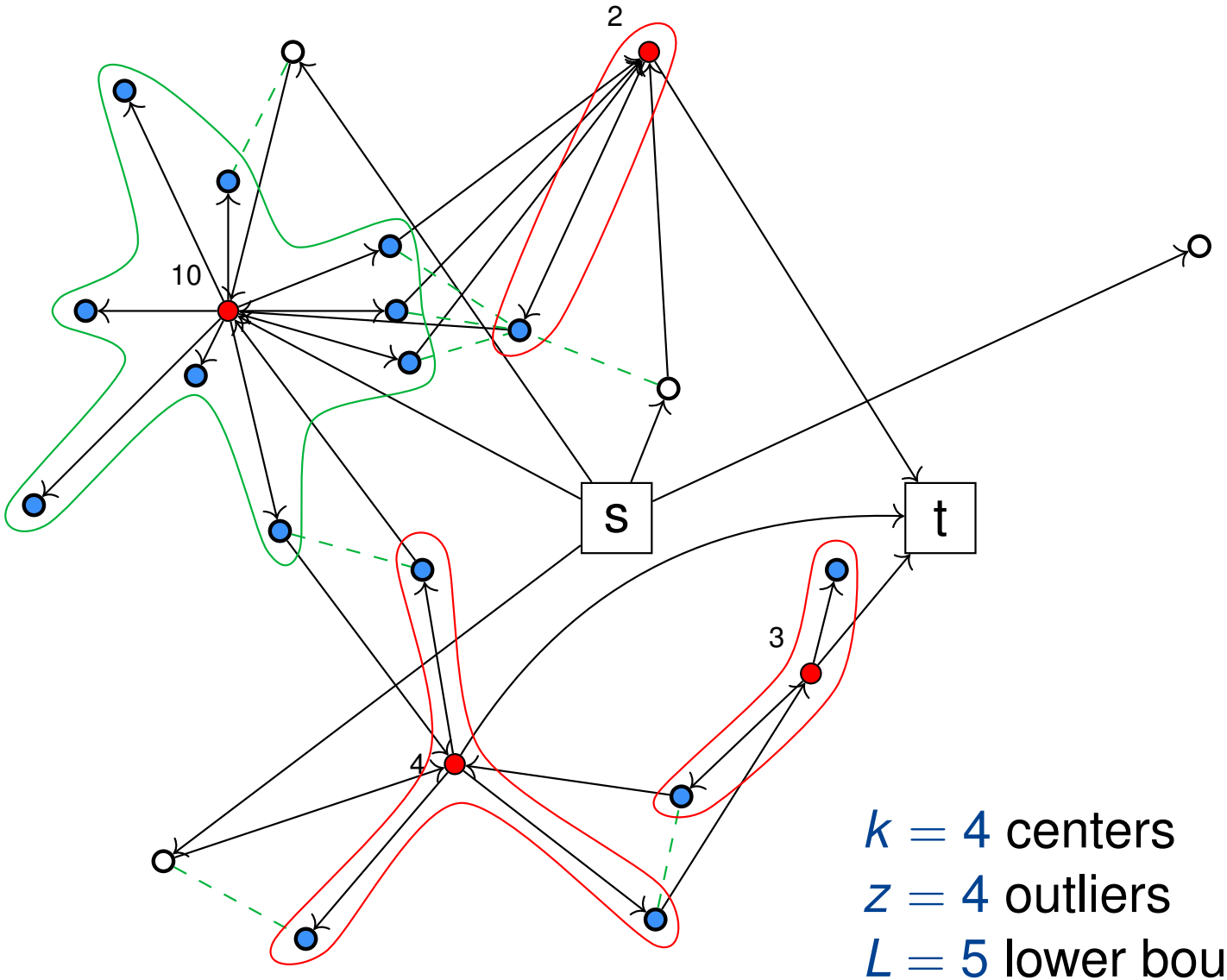
$k = 4$ centers
$z = 4$ outliers
$L = 5$ lower bound

Step 3: Build network



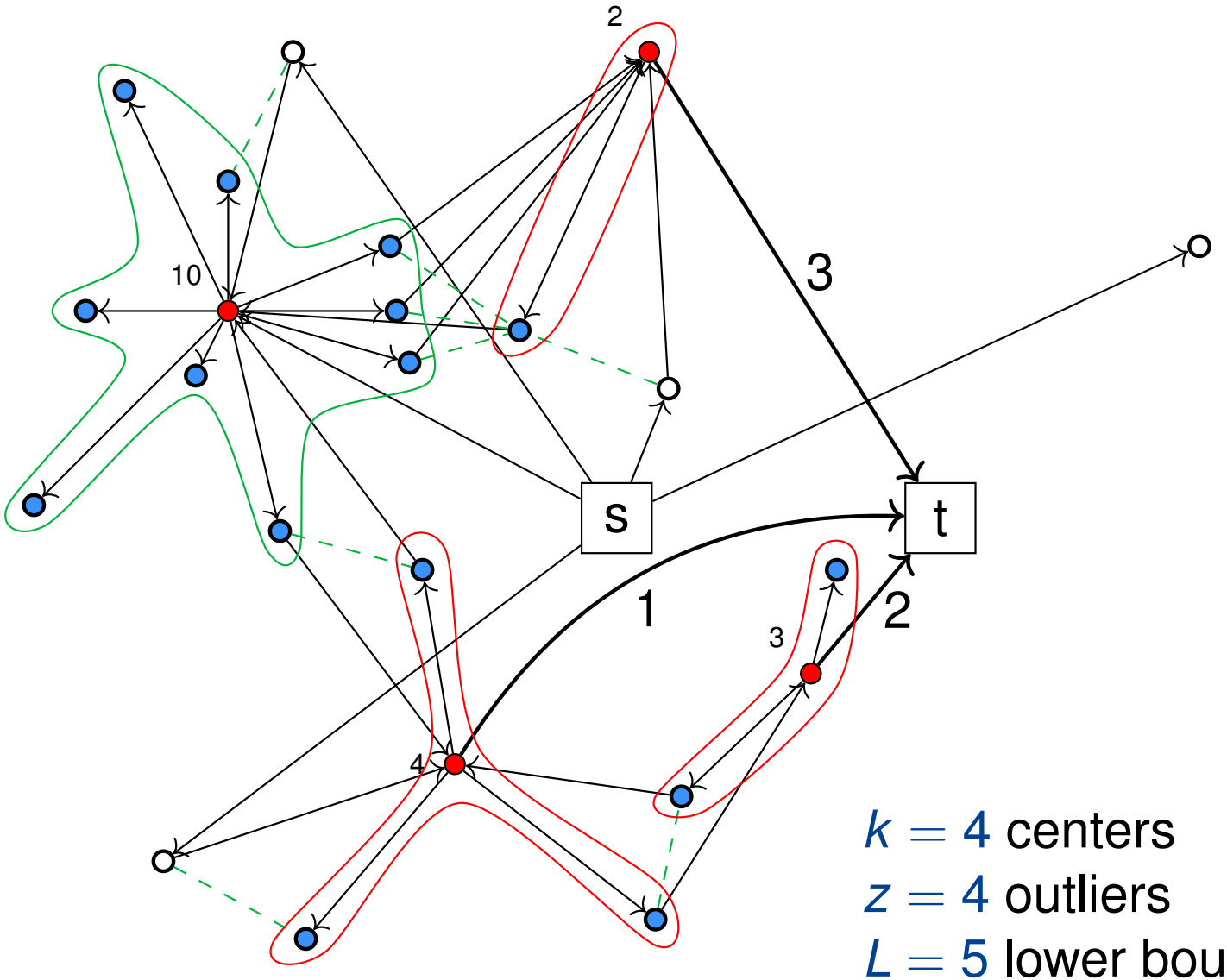
$k = 4$ centers
$z = 4$ outliers
$L = 5$ lower bound

Step 3: Build network



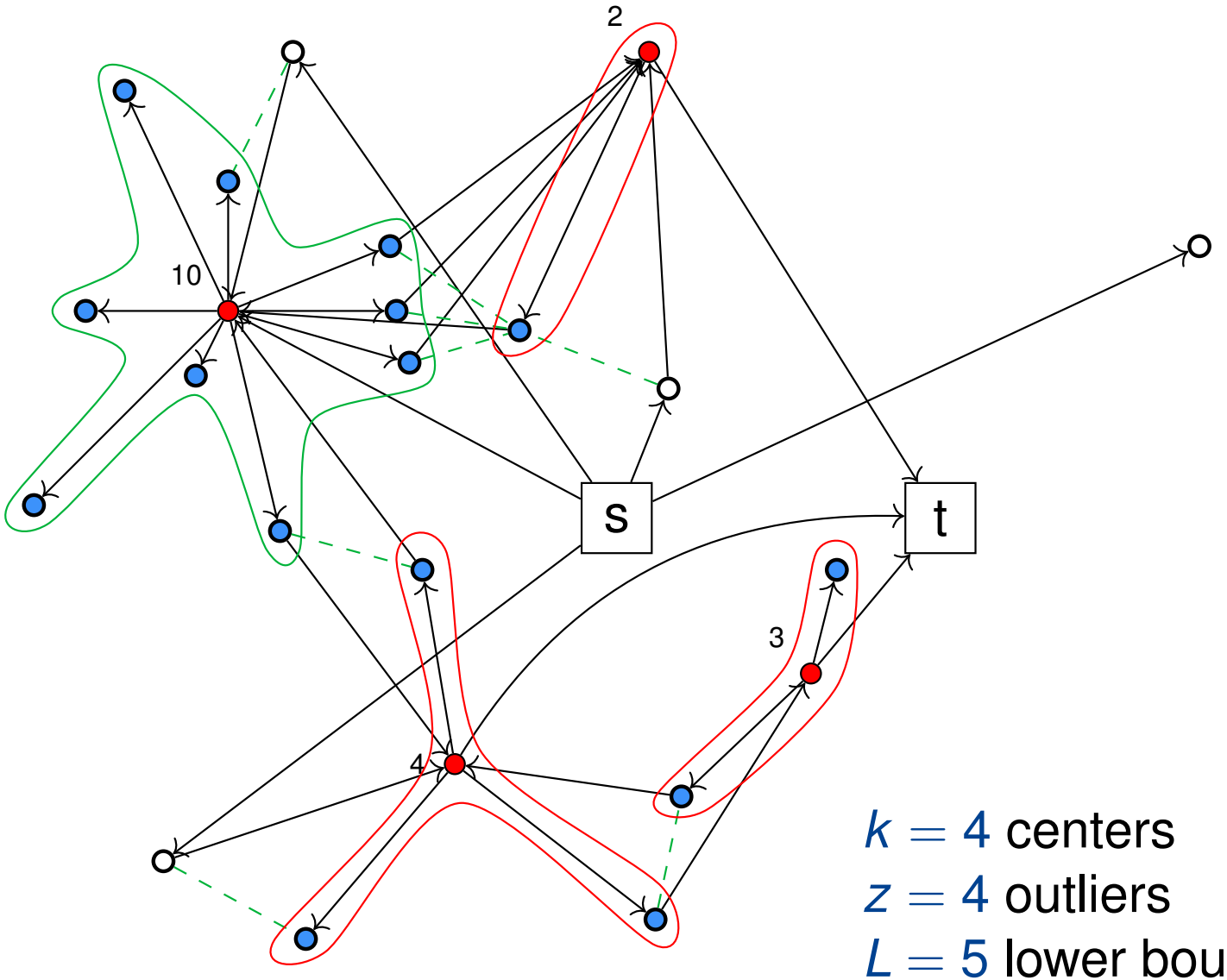
$k = 4$ centers
$z = 4$ outliers
$L = 5$ lower bound

Step 3: Build network



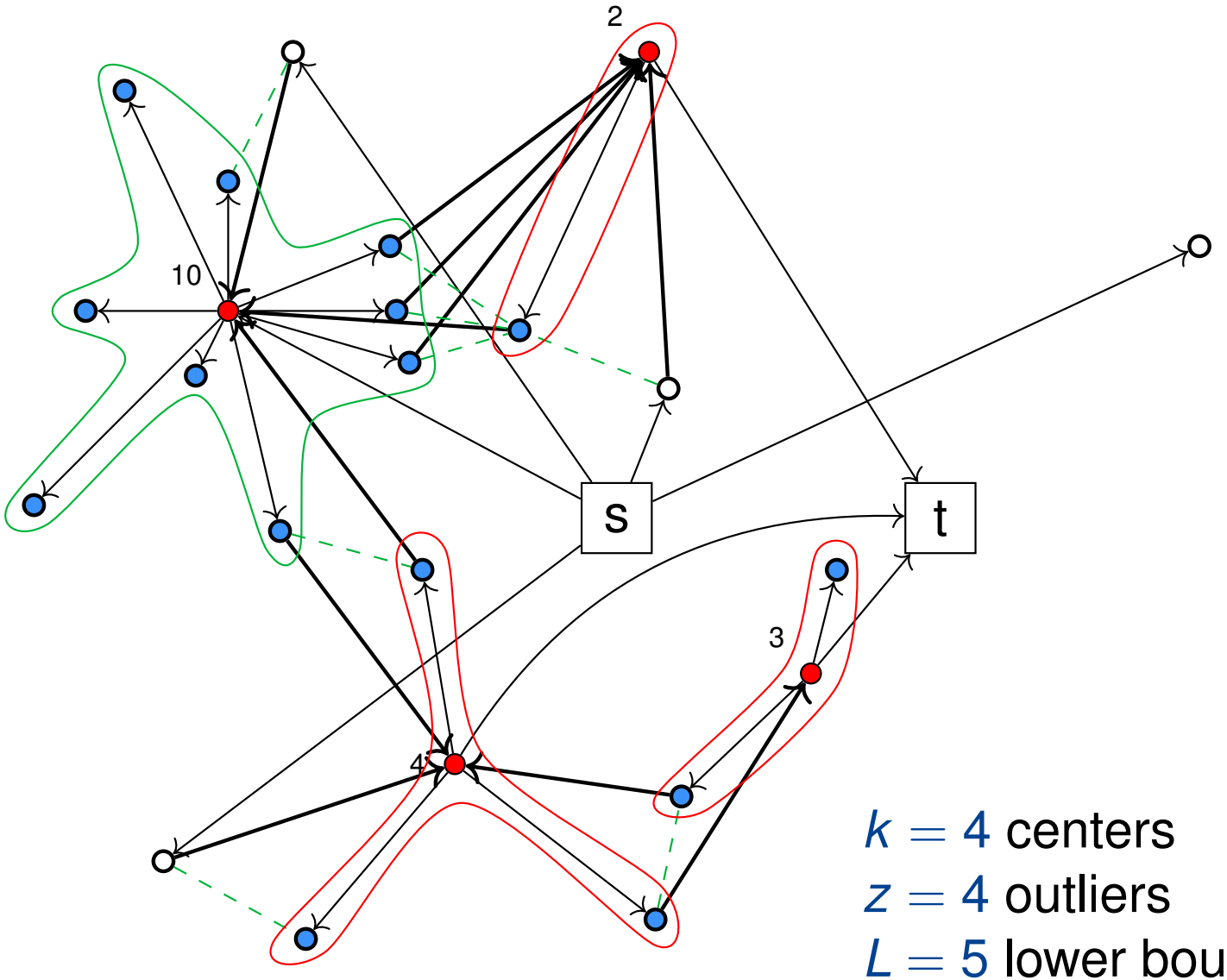
$k = 4$ centers
$z = 4$ outliers
$L = 5$ lower bound

Step 3: Build network



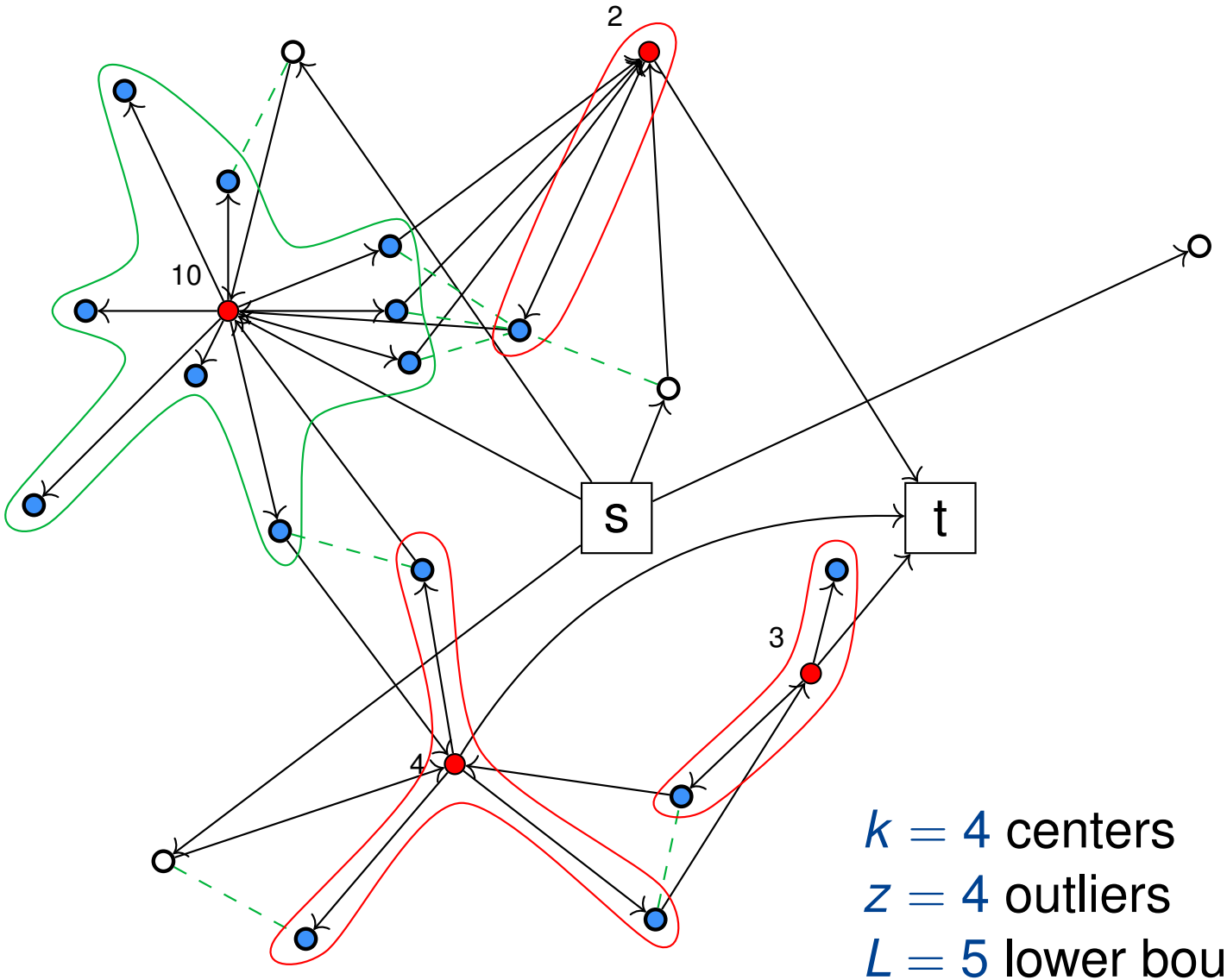
$k = 4$ centers
$z = 4$ outliers
$L = 5$ lower bound

Step 3: Build network



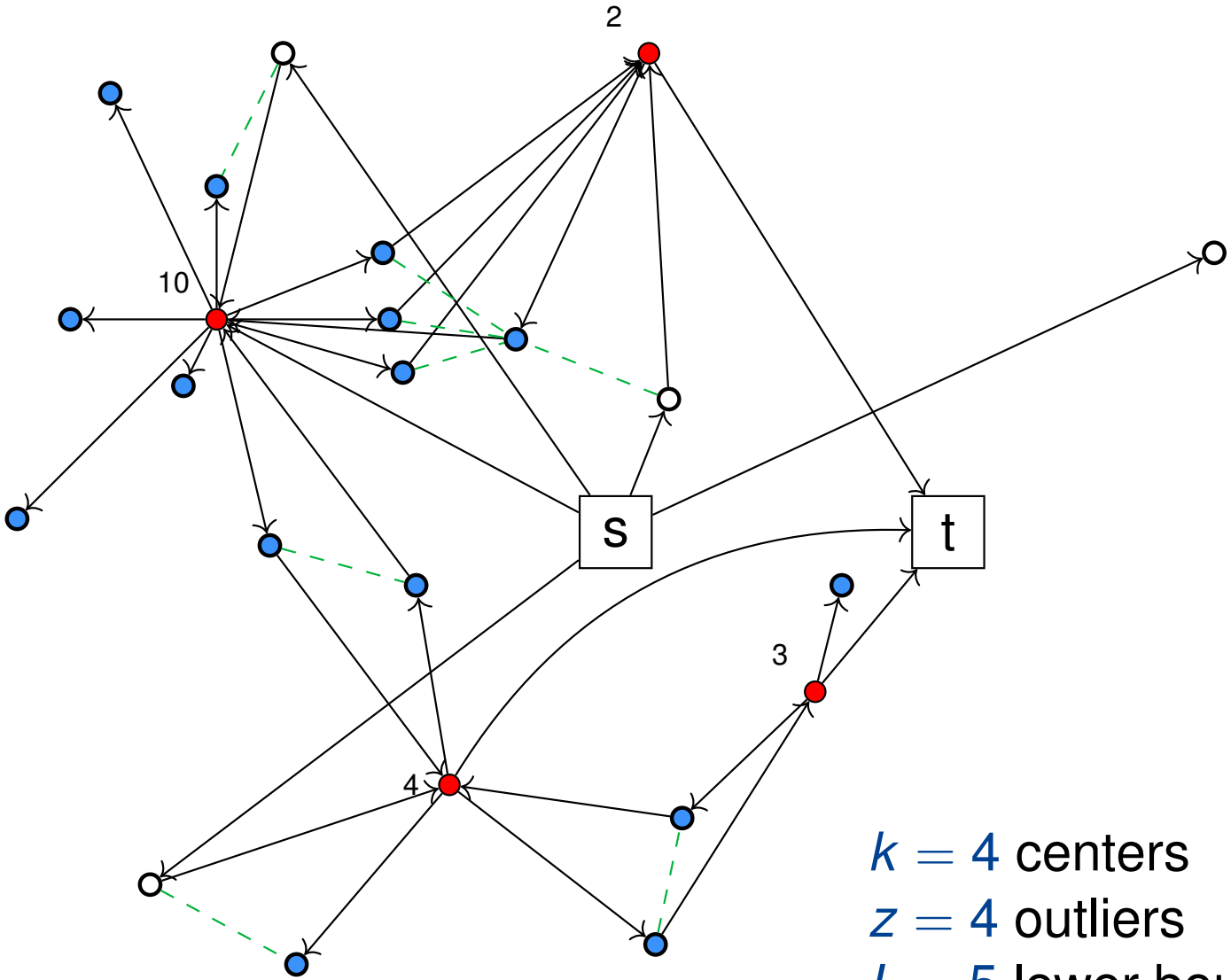
$k = 4$ centers
$z = 4$ outliers
$L = 5$ lower bound

Step 3: Build network



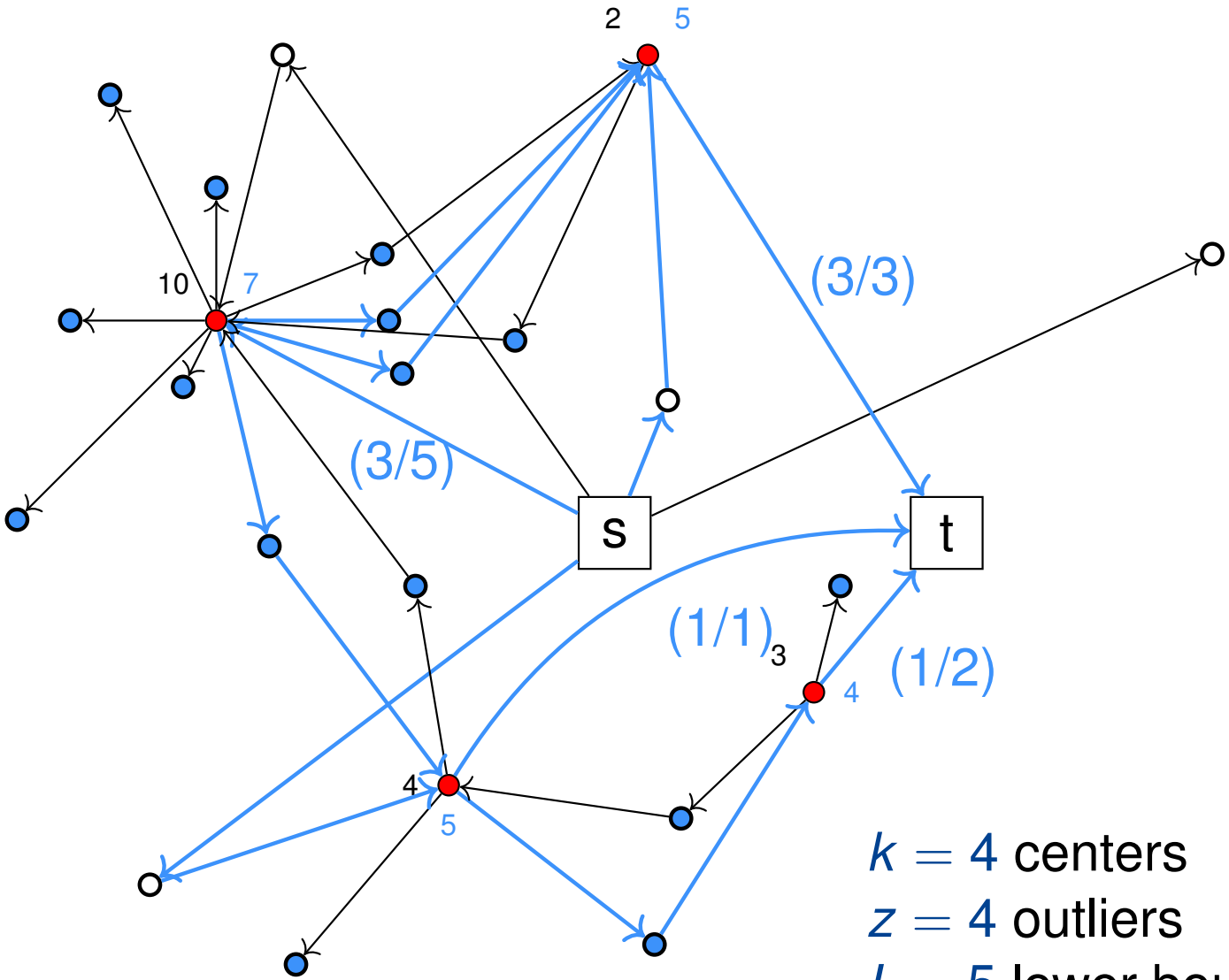
$k = 4$ centers
$z = 4$ outliers
$L = 5$ lower bound

Step 3: Build network



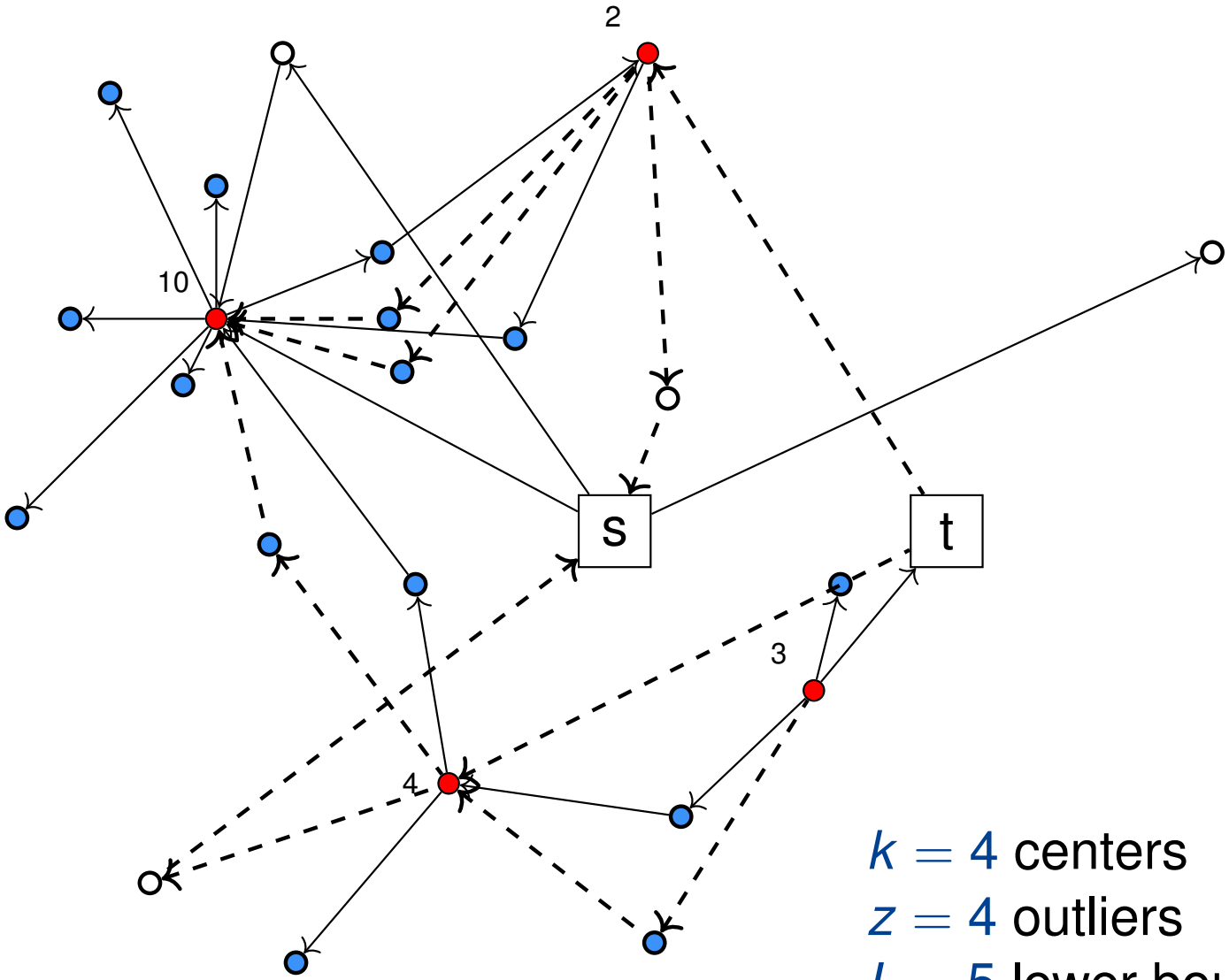
$k = 4$ centers
$z = 4$ outliers
$L = 5$ lower bound

Step 4: Compute an integral maximum flow



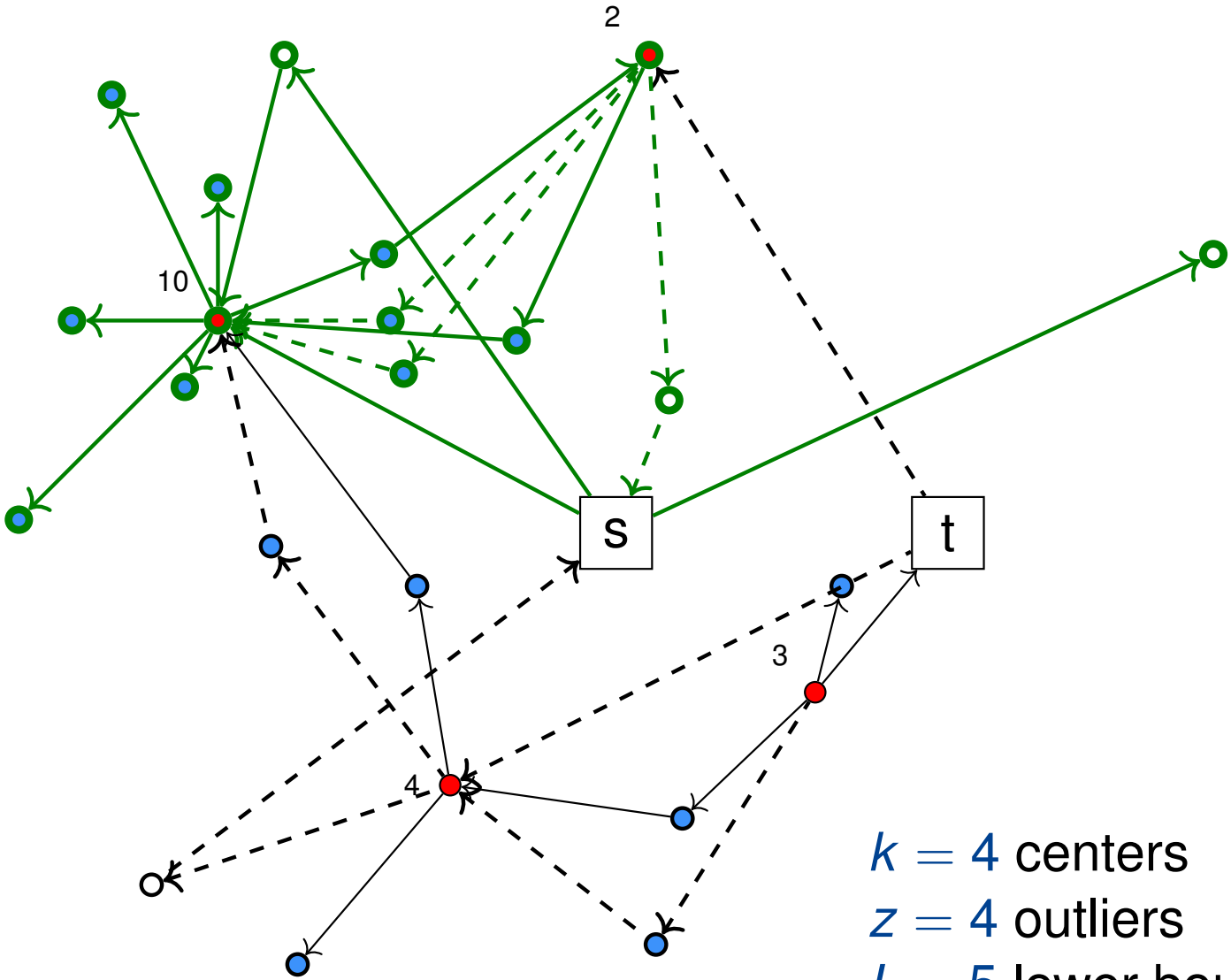
$k = 4$ centers
$z = 4$ outliers
$L = 5$ lower bound

Step 4: Compute an integral maximum flow



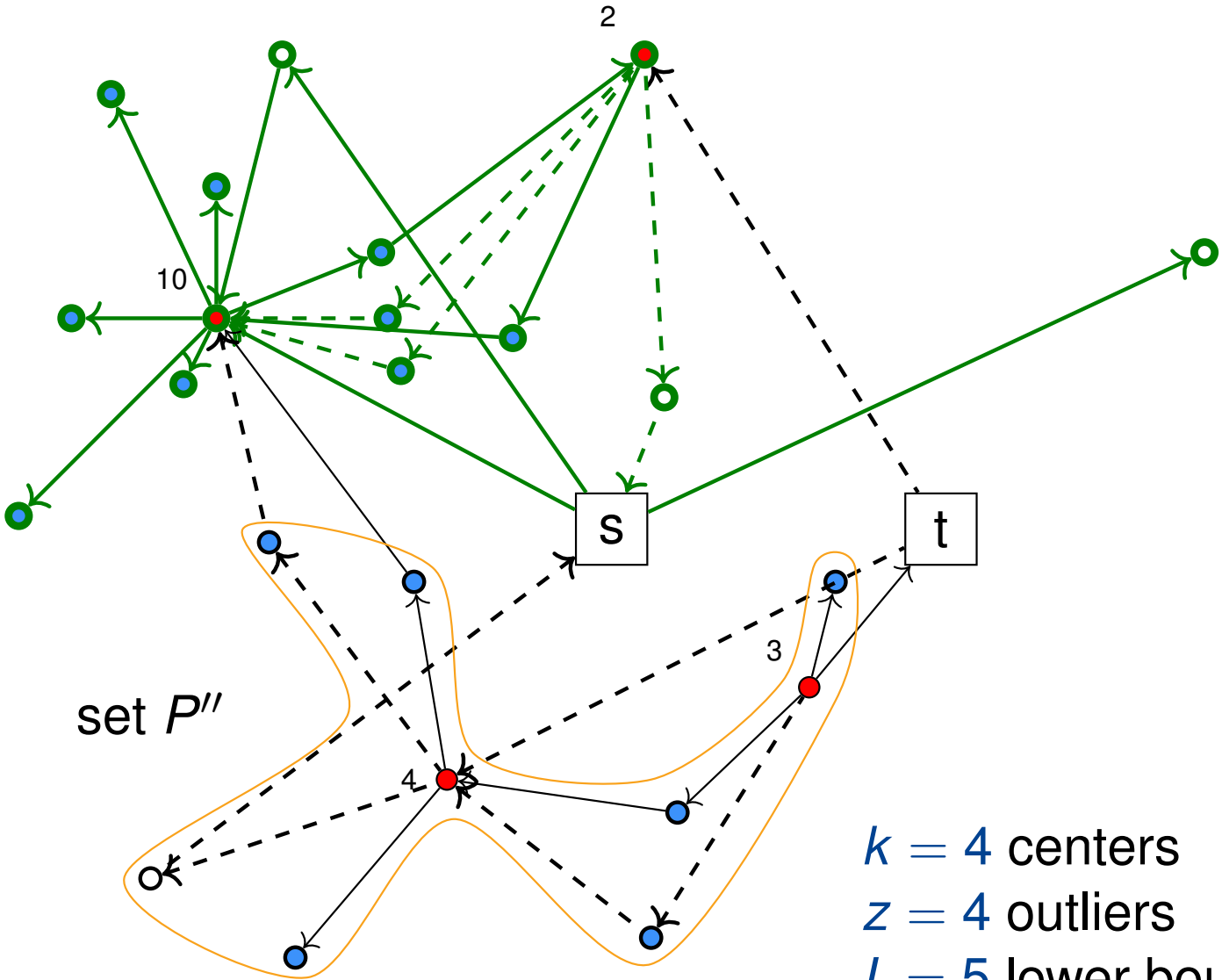
$k = 4$ centers
$z = 4$ outliers
$L = 5$ lower bound

Step 5: Compute reachable points



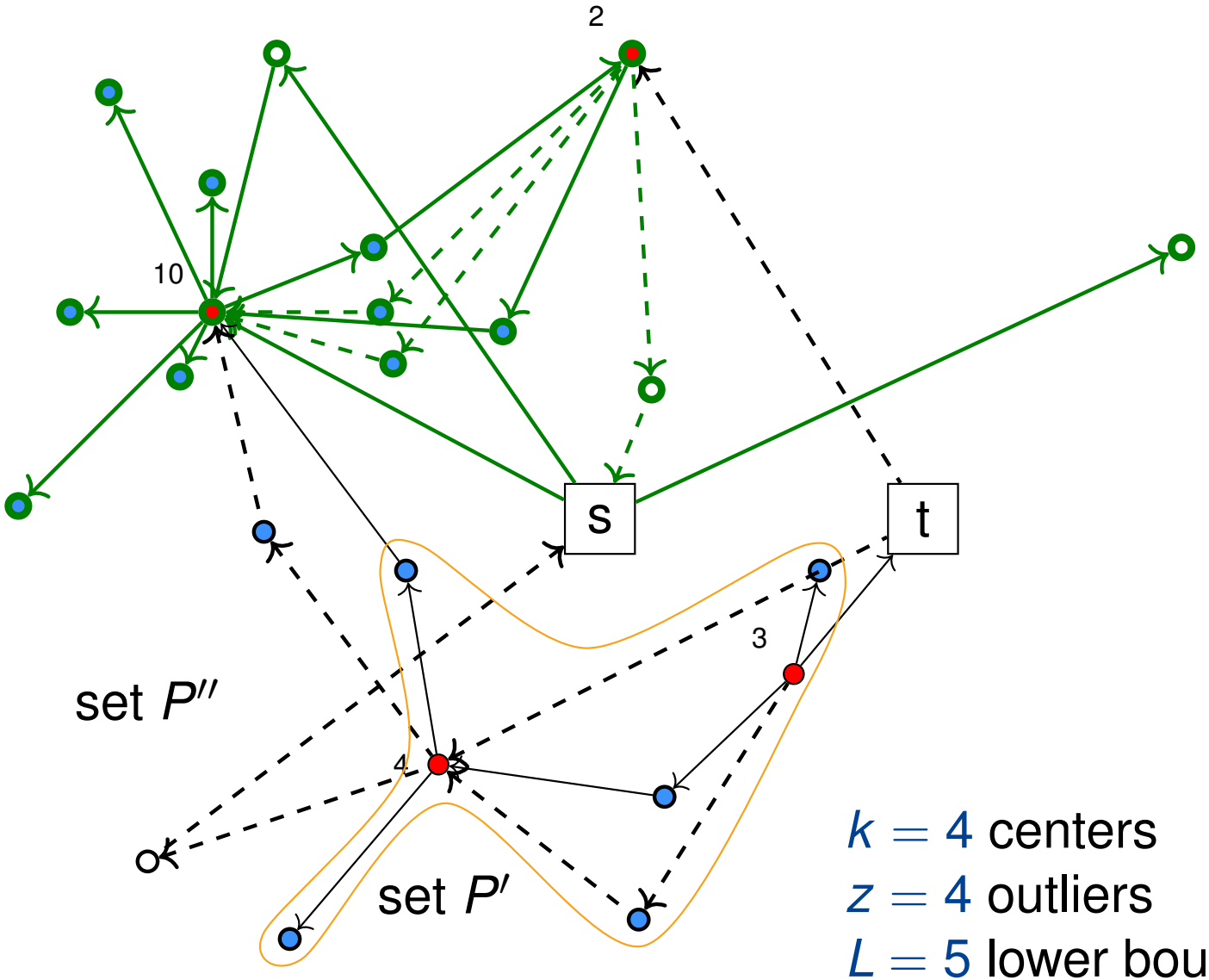
$k = 4$ centers
$z = 4$ outliers
$L = 5$ lower bound

Step 5: Compute reachable points



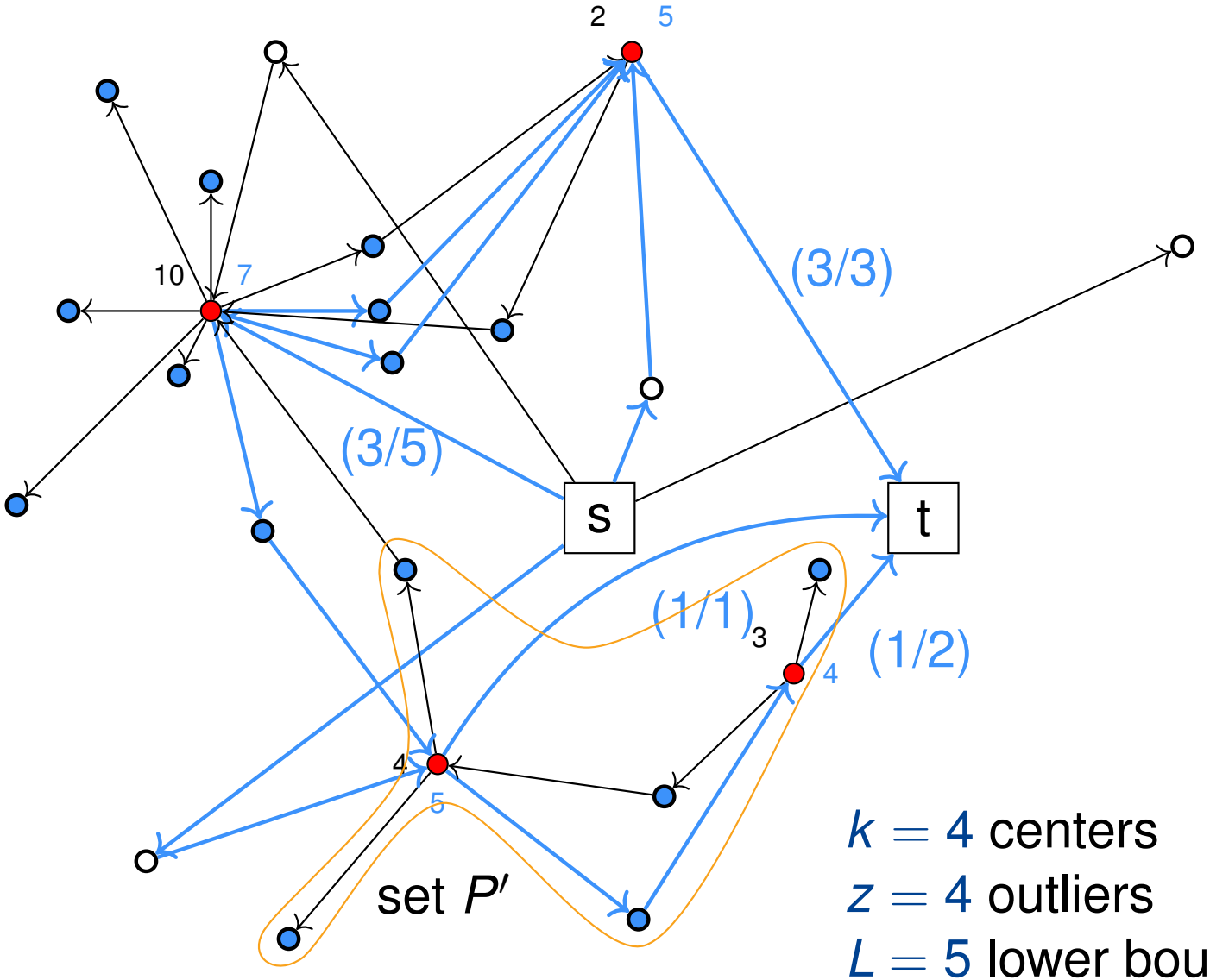
$k = 4$ centers
$z = 4$ outliers
$L = 5$ lower bound

Step 5: Compute reachable points and clusters



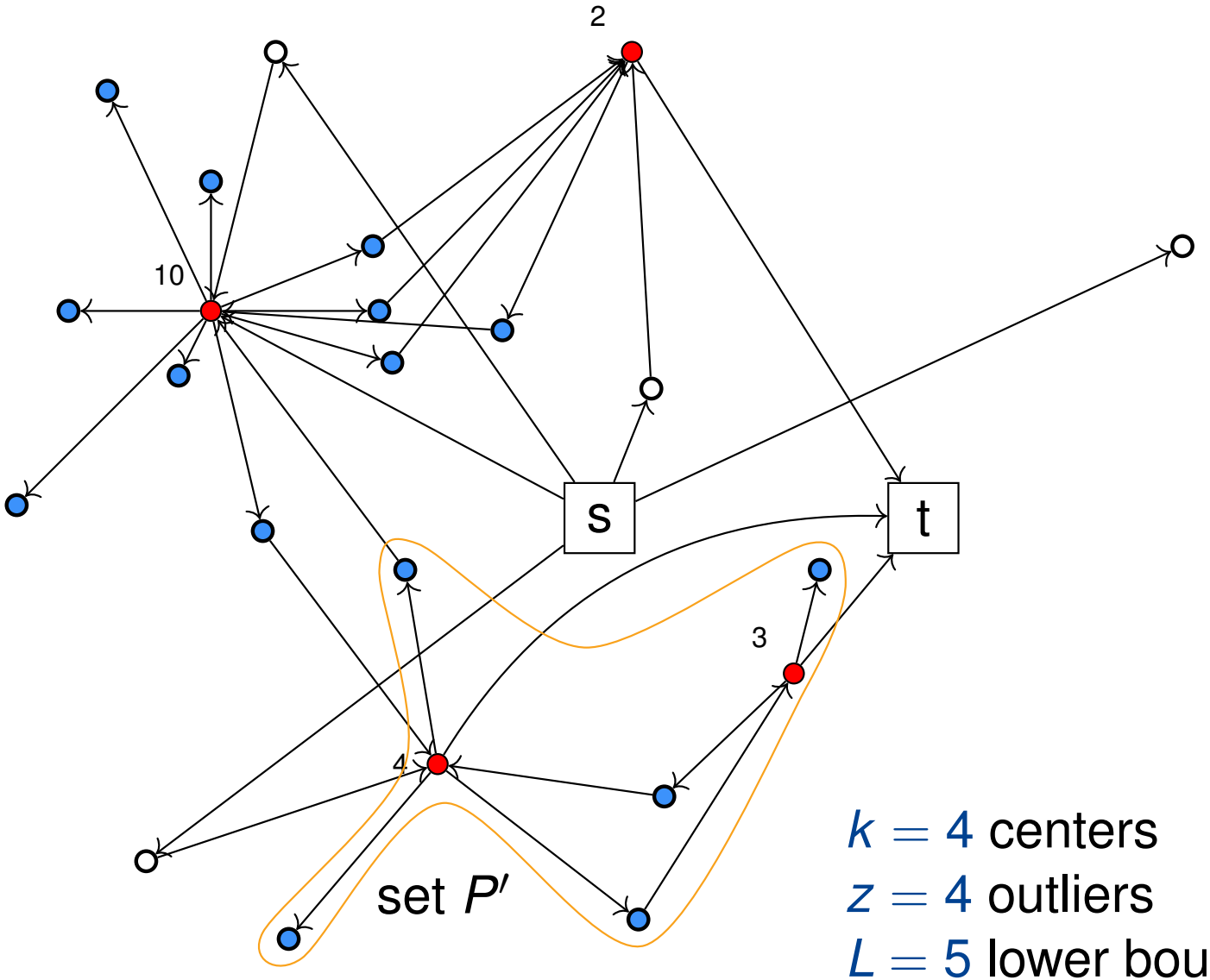
$k = 4$ centers
$z = 4$ outliers
$L = 5$ lower bound

Step 5: Compute reachable points and clusters



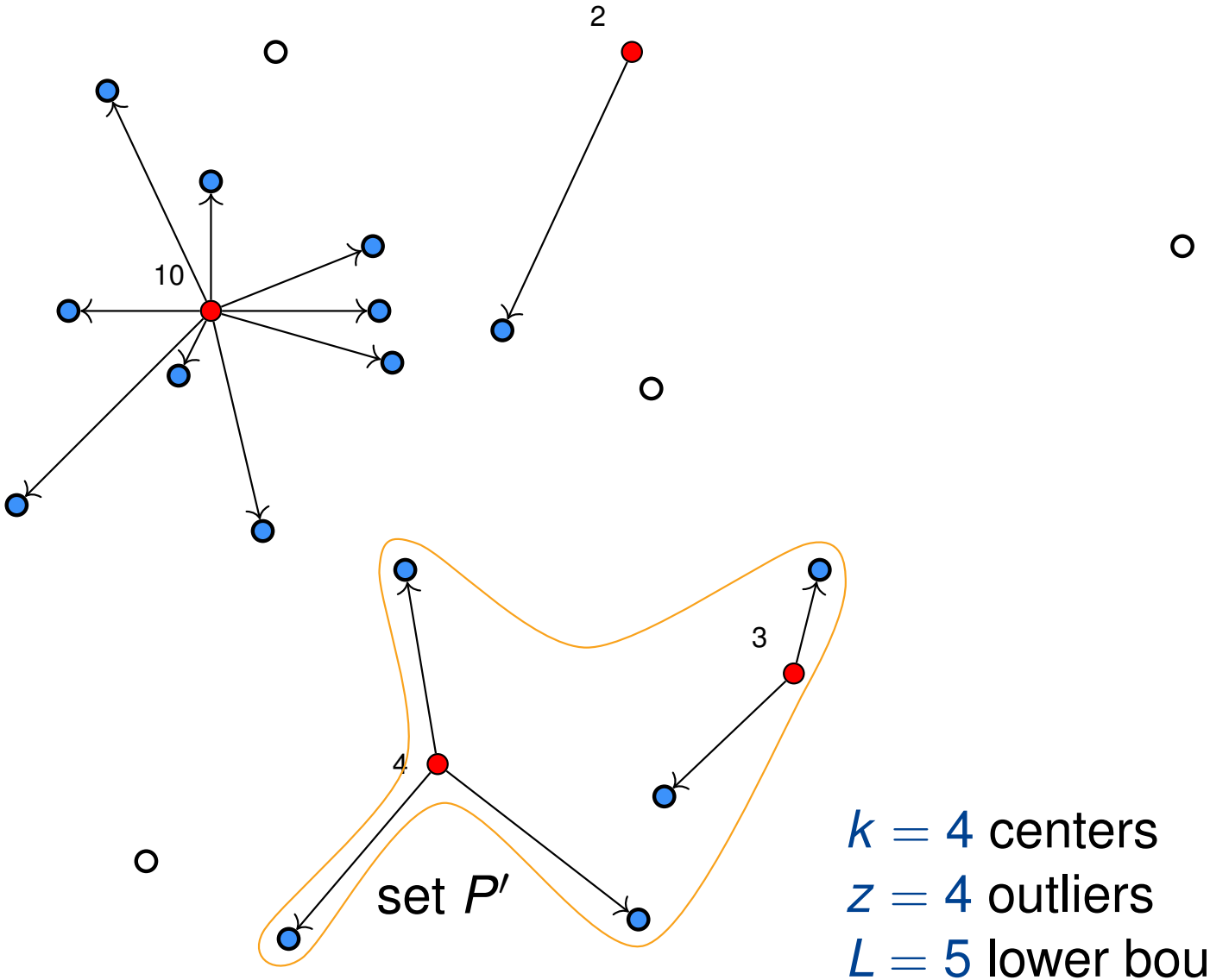
$k = 4$ centers
$z = 4$ outliers
$L = 5$ lower bound

Step 5: Recompute solution on P'!



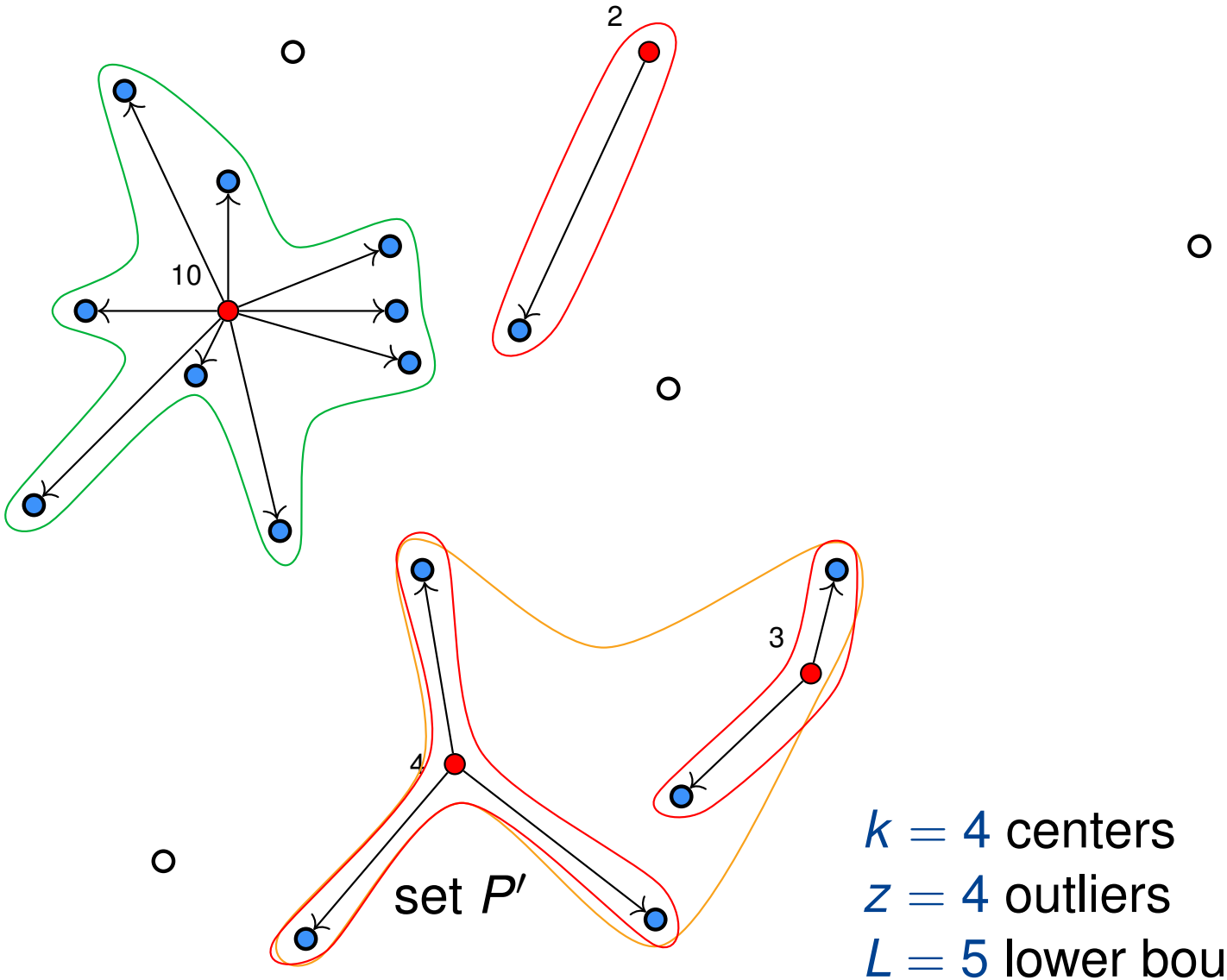
$k = 4$ centers
$z = 4$ outliers
$L = 5$ lower bound

Step 5: Recompute solution on P'!



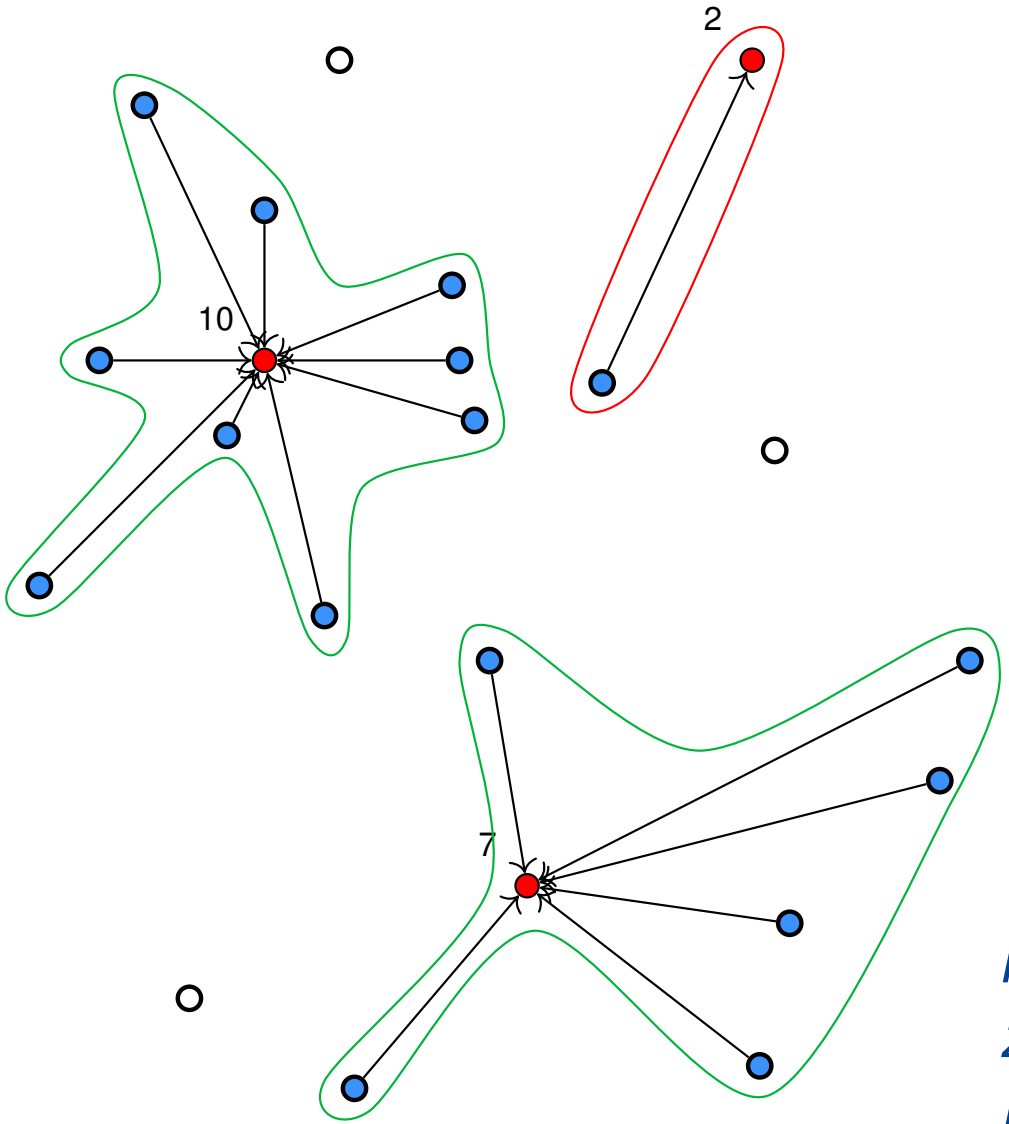
$k = 4$ centers
$z = 4$ outliers
$L = 5$ lower bound

Step 5: Recompute solution on P'!



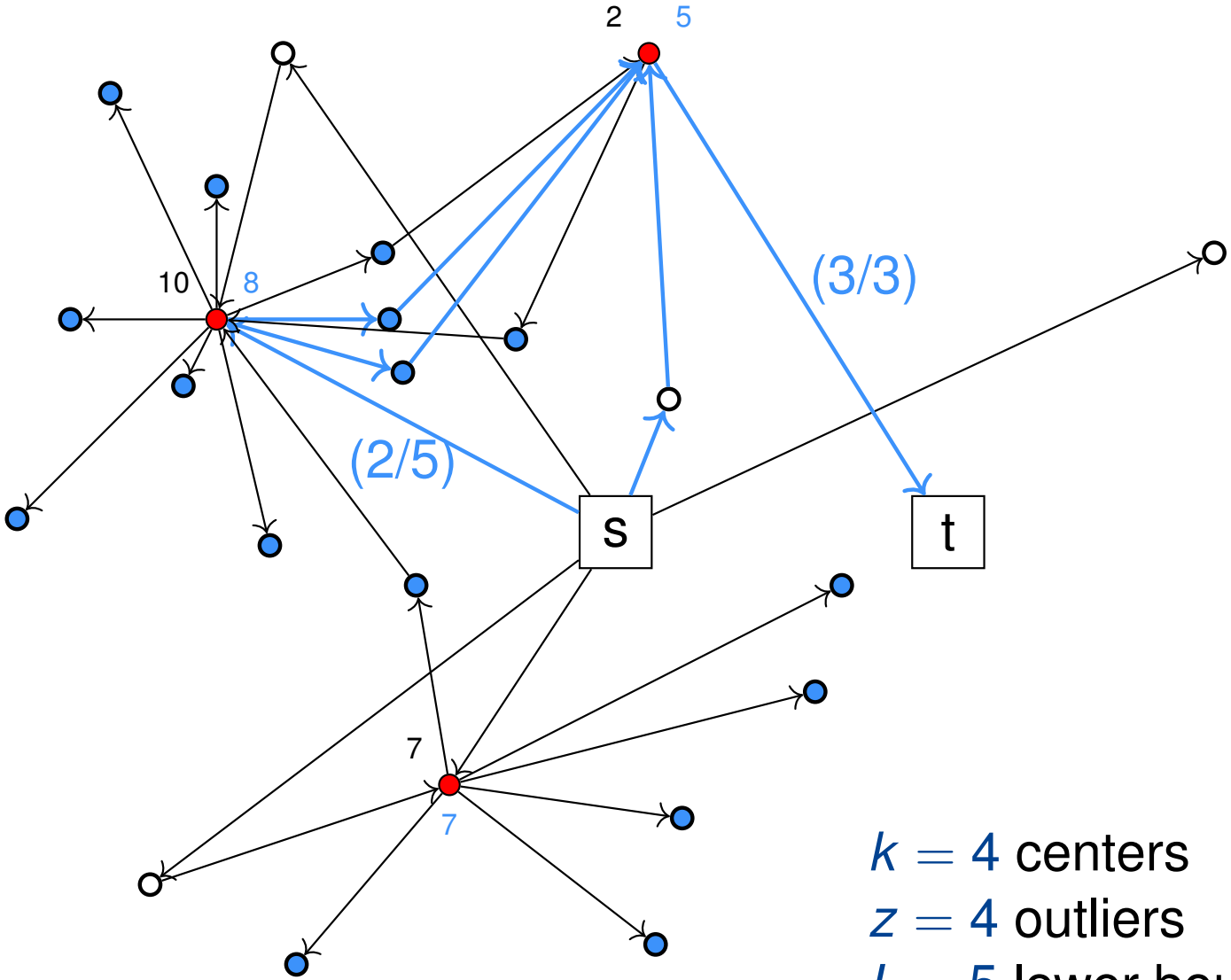
$k = 4$ centers
$z = 4$ outliers
$L = 5$ lower bound

Step 5: Recompute solution on P'!



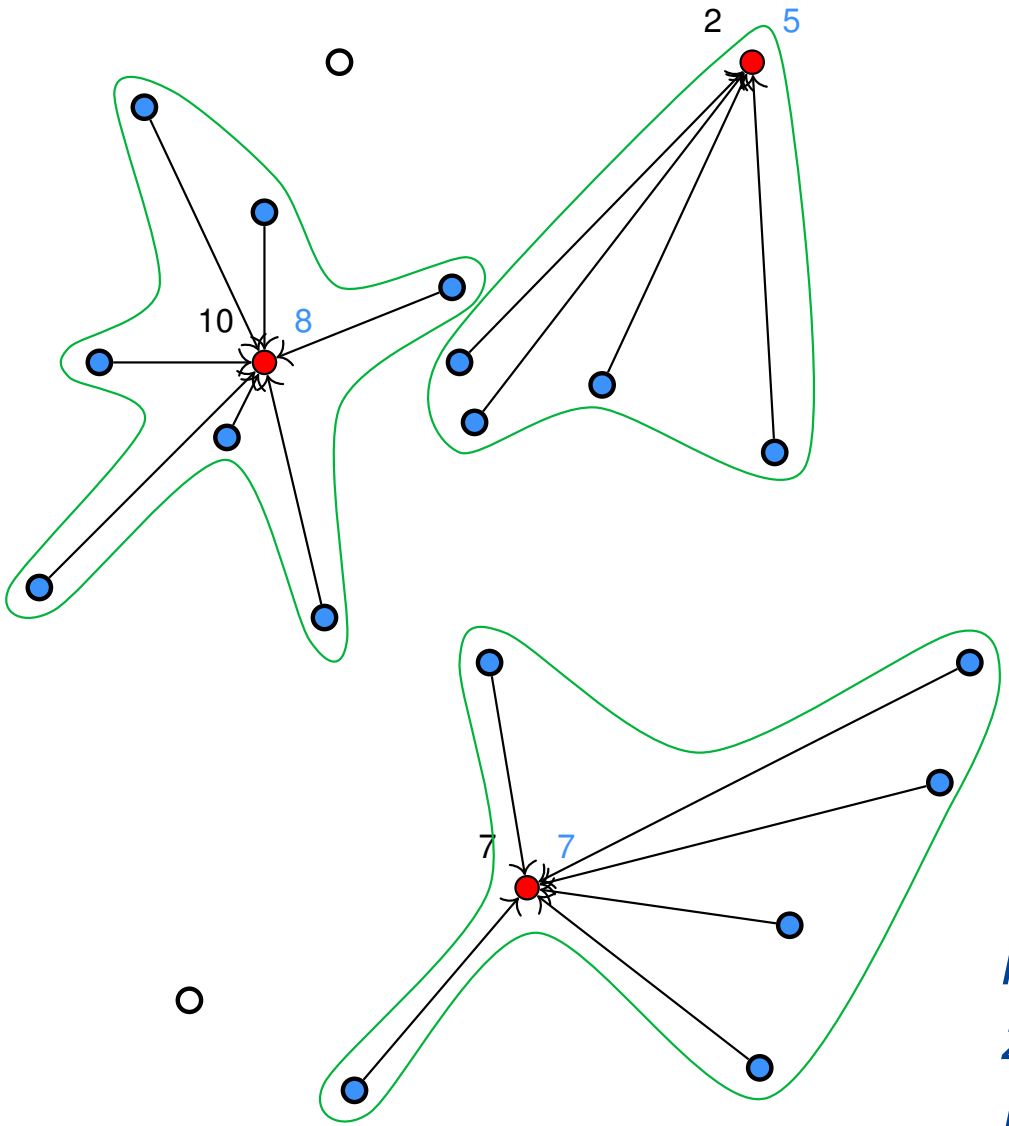
$k = 4$ centers
$z = 4$ outliers
$L = 5$ lower bound

Step 4 (again): Compute an integral maximum flow



$k = 4$ centers
$z = 4$ outliers
$L = 5$ lower bound

Step 6: Success! Move the points wrt the maximum flow



$k = 4$ centers
$z = 4$ outliers
$L = 5$ lower bound

Some details

Some details

- Points are only moved once, at the very end!

Some details

- Points are only moved once, at the very end!
- Recomputations do not increase the factor

Some details

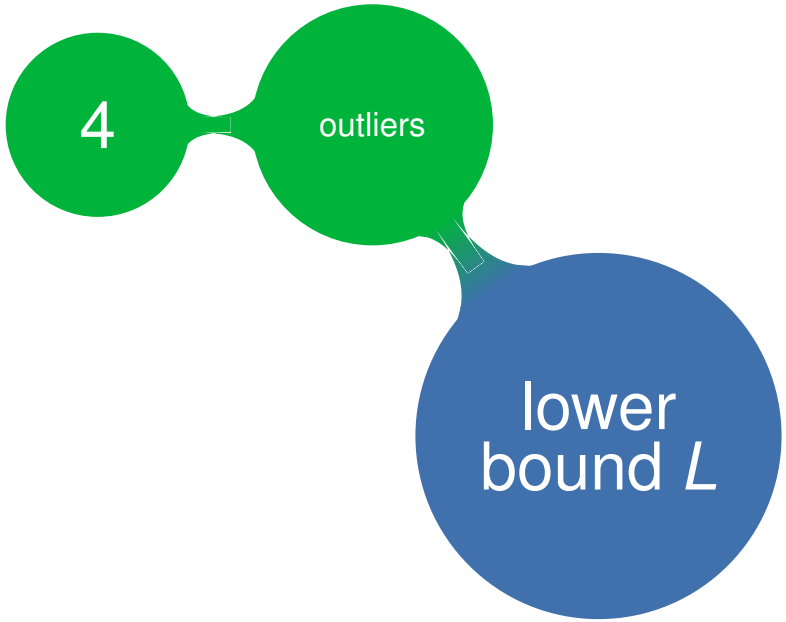
- Points are only moved once, at the very end!
- Recomputations do not increase the factor
- After every flow computation, number of clusters (or number of outliers) decreases $\rightsquigarrow \leq k \cdot z$ flow computations

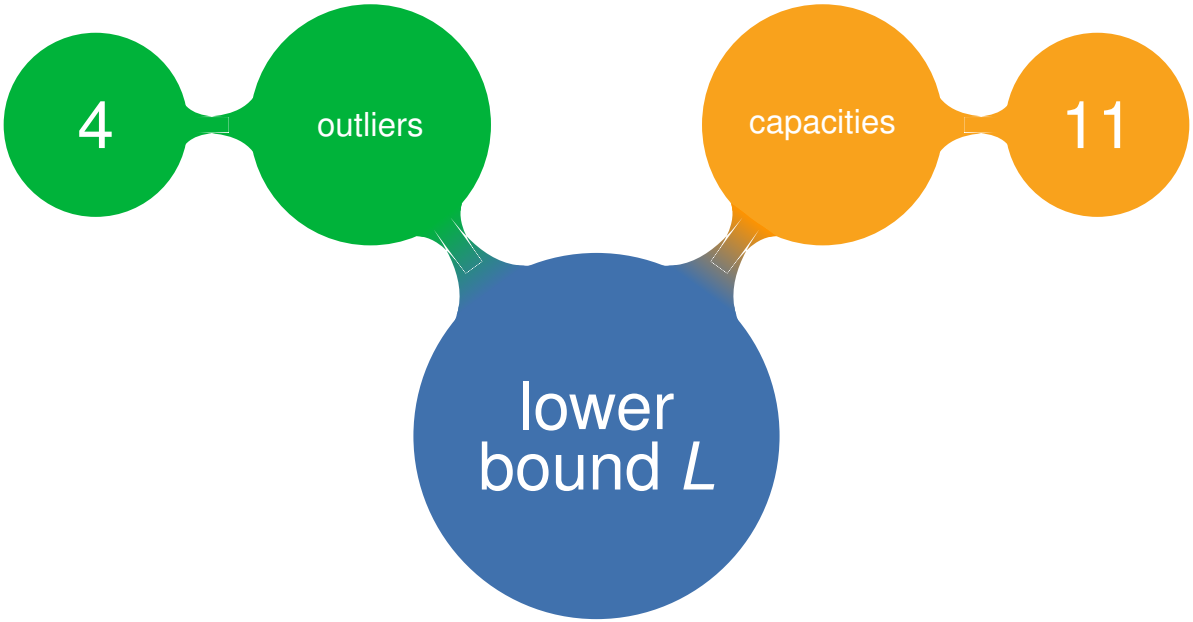
Some details

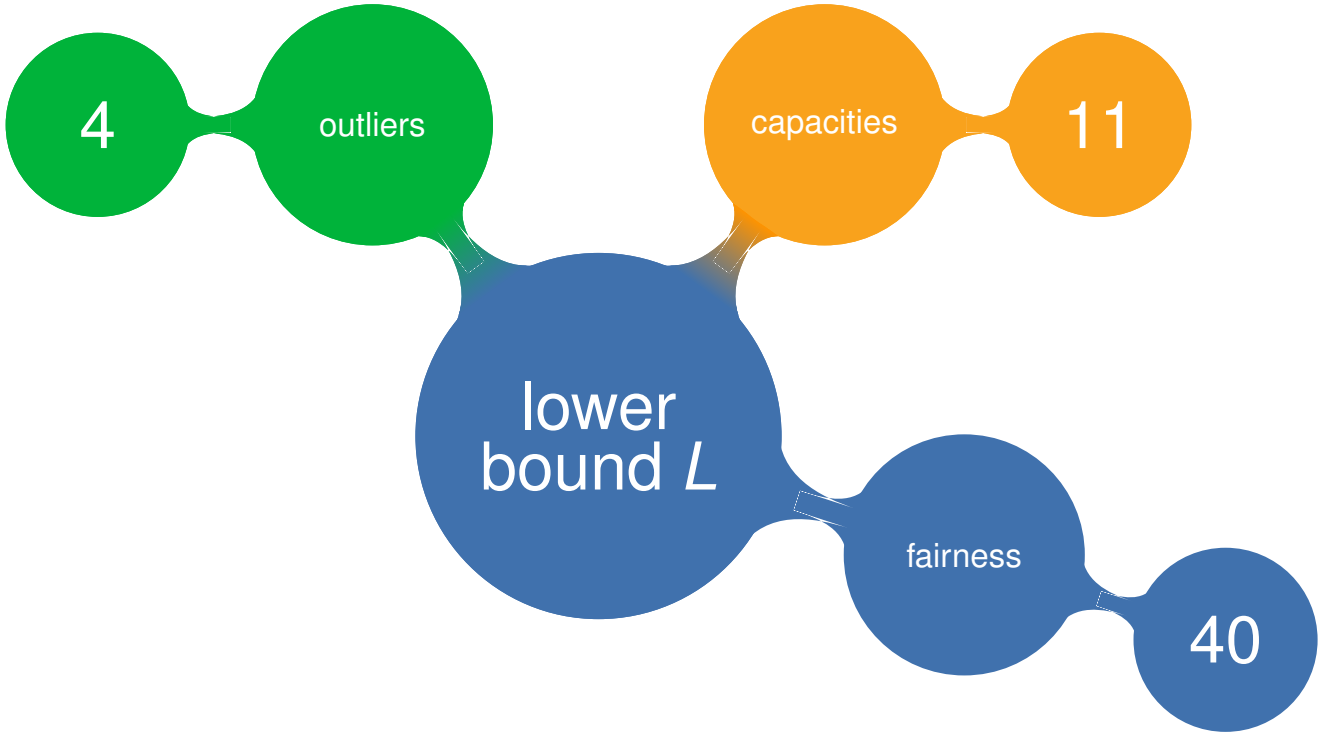
- Points are only moved once, at the very end!
- Recomputations do not increase the factor
- After every flow computation, number of clusters (or number of outliers) decreases $\rightsquigarrow \leq k \cdot z$ flow computations

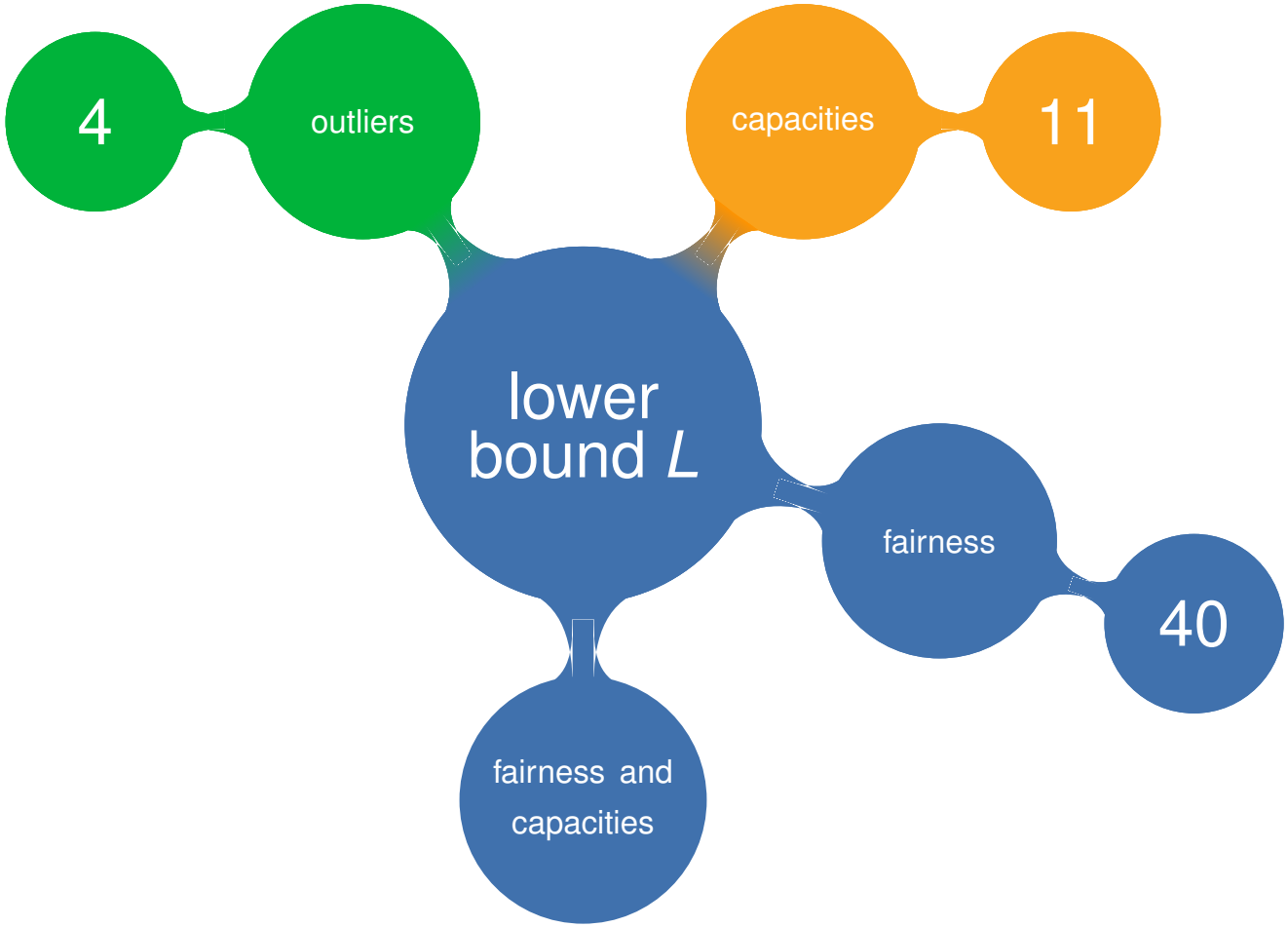
Theorem (Rösner, S., 2018)

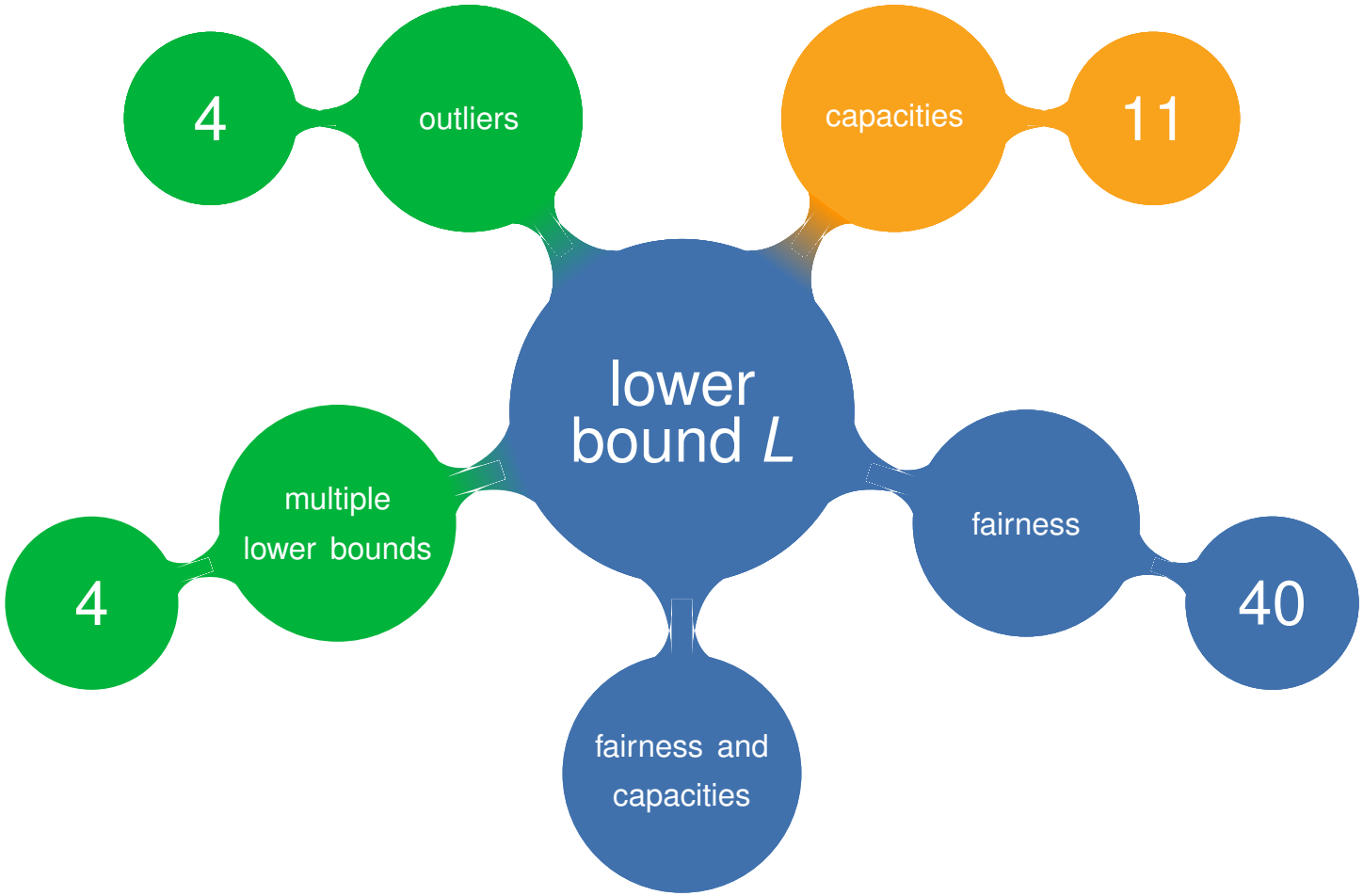
Let A be an α-approximation algorithm for k-center with outliers. Then we can compute an $(\alpha + 2)$-approximation for k-center with outliers and lower bound L in polynomial time.


4
outliers
lower
bound L


4
outliers
capacities
11
lower
bound L


4
outliers
capacities
11
lower
bound L
fairness
40


4
outliers
capacities
11
lower bound L
fairness
40
fairness and capacities


4
outliers
capacities
11
lower
bound L
multiple
lower bounds
4
fairness
40
fairness and
capacities

Follow-up questions

Follow-up questions

- Reduce increase of approximation factor

Follow-up questions

- Reduce increase of approximation factor
- Non-uniform lower bounds

Follow-up questions

- Reduce increase of approximation factor
- Non-uniform lower bounds
- Other constraints. Fairness works, what about outliers?

Follow-up questions

- Reduce increase of approximation factor
- Non-uniform lower bounds
- Other constraints. Fairness works, what about outliers?
- k-median? k-means?

Last Part: Data reduction

Last Part: Data reduction

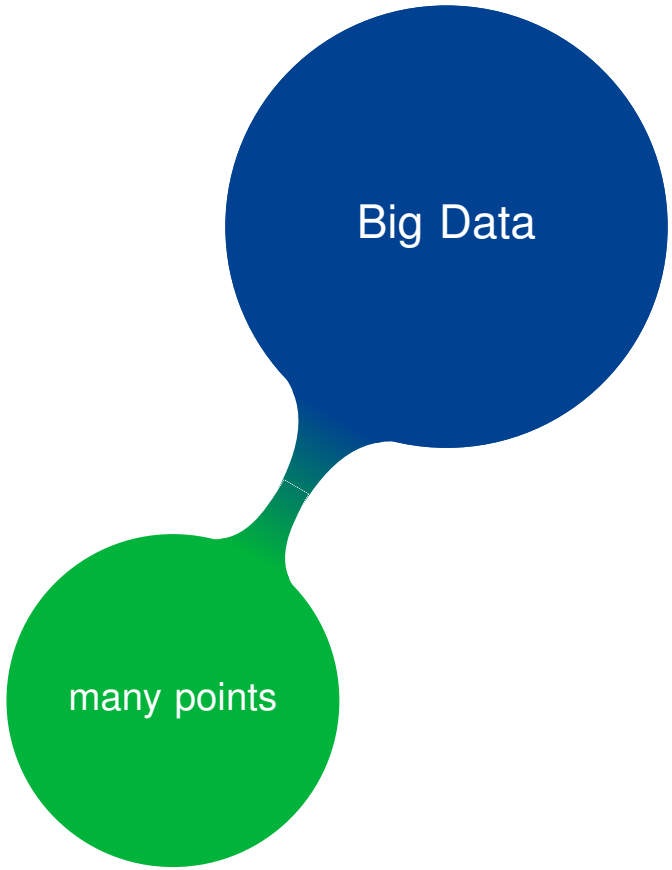
- Quick review for k-means

Last Part: Data reduction

- Quick review for k-means
- What about constraints?

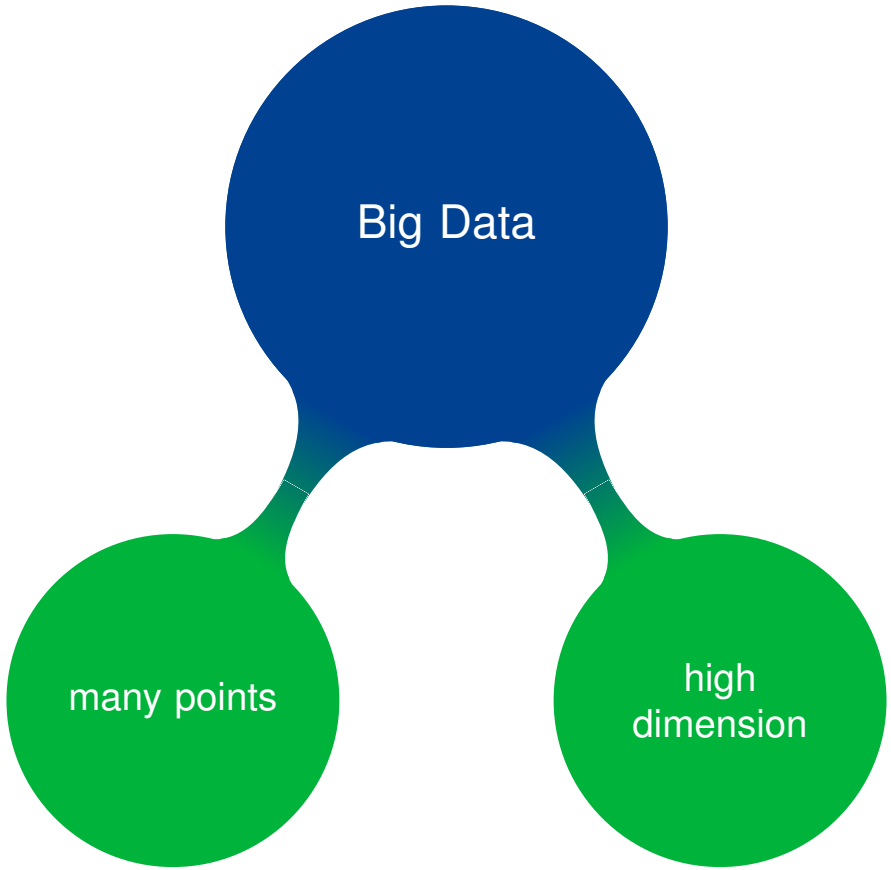
Last Part: Data reduction

- Quick review for k-means
- What about constraints?



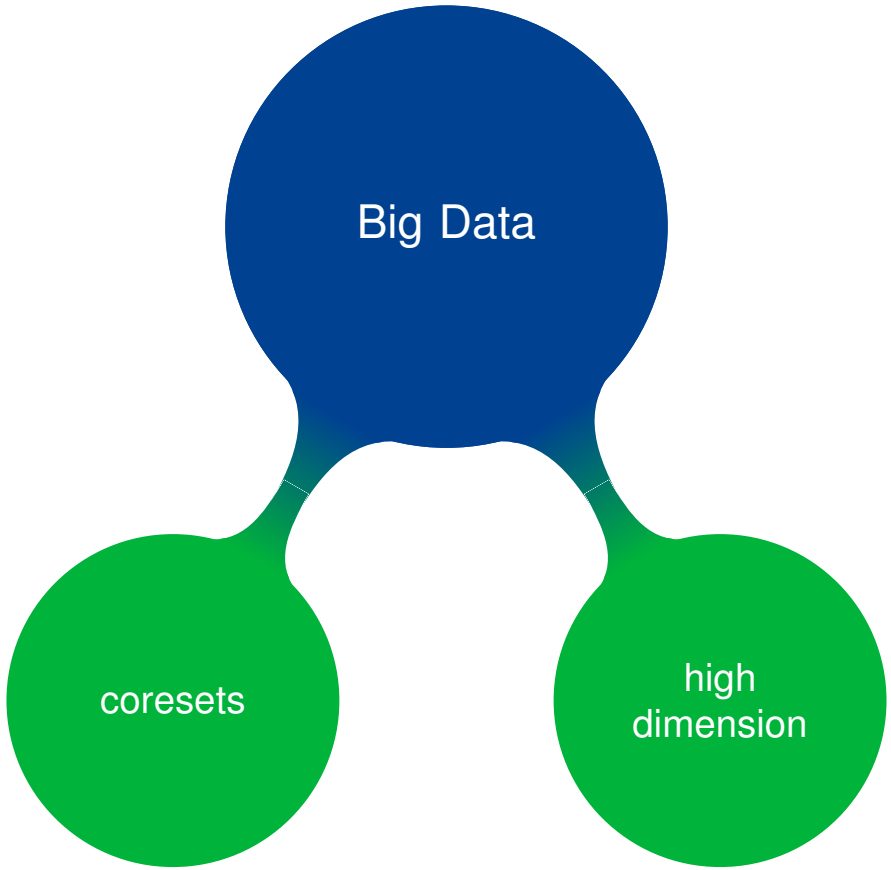
Last Part: Data reduction

- Quick review for k-means
- What about constraints?



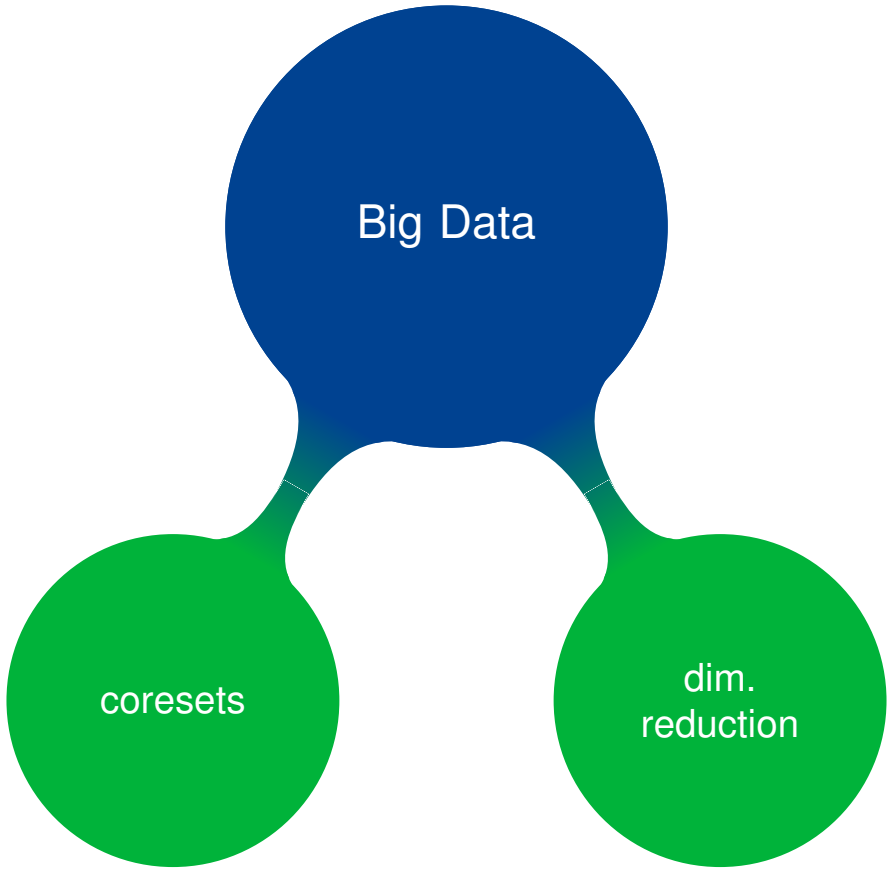
Last Part: Data reduction

- Quick review for k-means
- What about constraints?



Last Part: Data reduction

- Quick review for k-means
- What about constraints?



Reducing the dimension and number of input points

Coresets / Dim. Red

Reduce complexity,
but approximately preserve k-means cost for all sets of k centers

Reducing the dimension and number of input points

Coresets / Dim. Red

Reduce complexity,
but approximately preserve k-means cost for all sets of k centers

Definition

Let $P \subset \mathbb{R}^d$ be a point set. If $P' \subset \mathbb{R}^d$ sat. $\forall C \subset \mathbb{R}^d, |C| = k$:

$$|\operatorname{cost}(P, C) - \operatorname{cost}(P', C)| \leq \varepsilon \cdot \operatorname{cost}(P, C)$$

and P'...

Reducing the dimension and number of input points

Coresets / Dim. Red

Reduce complexity,
but approximately preserve k-means cost for all sets of k centers

Definition

Let $P \subset \mathbb{R}^d$ be a point set. If $P' \subset \mathbb{R}^d$ sat. $\forall C \subset \mathbb{R}^d, |C| = k$:

$$|\operatorname{cost}(P, C) - \operatorname{cost}(P', C)| \leq \varepsilon \cdot \operatorname{cost}(P, C)$$

and P'...

- ...satisfies that $|P'| \ll |P|$

Reducing the dimension and number of input points

Coresets / Dim. Red

Reduce complexity,
but approximately preserve k-means cost for all sets of k centers

Definition

Let $P \subset \mathbb{R}^d$ be a point set. If $P' \subset \mathbb{R}^d$ sat. $\forall C \subset \mathbb{R}^d, |C| = k$:

$$|\operatorname{cost}(P, C) - \operatorname{cost}(P', C)| \leq \varepsilon \cdot \operatorname{cost}(P, C)$$

and P'...

- ...satisfies that $|P'| \ll |P| \rightsquigarrow$ then P' is a coreset

Reducing the dimension and number of input points

Coresets / Dim. Red

Reduce complexity,
but approximately preserve k-means cost for all sets of k centers

Definition

Let $P \subset \mathbb{R}^d$ be a point set. If $P' \subset \mathbb{R}^d$ sat. $\forall C \subset \mathbb{R}^d, |C| = k$:

$$|\operatorname{cost}(P, C) - \operatorname{cost}(P', C)| \leq \varepsilon \cdot \operatorname{cost}(P, C)$$

and P'...

- ...satisfies that $|P'| \ll |P| \rightsquigarrow$ then P' is a coreset
- ...has intrinsic dimension s for an $s \ll d$

Reducing the dimension and number of input points

Coresets / Dim. Red

Reduce complexity,
but approximately preserve k-means cost for all sets of k centers

Definition

Let $P \subset \mathbb{R}^d$ be a point set. If $P' \subset \mathbb{R}^d$ sat. $\forall C \subset \mathbb{R}^d, |C| = k$:

$$|\operatorname{cost}(P, C) - \operatorname{cost}(P', C)| \leq \varepsilon \cdot \operatorname{cost}(P, C)$$

and P'...

- ...satisfies that $|P'| \ll |P| \rightsquigarrow$ then P' is a coreset
- ...has intrinsic dimension s for an $s \ll d \rightsquigarrow$ dim. reduction

Reducing the dimension and number of input points

Coresets / Dim. Red

Reduce complexity,
but approximately preserve k-means cost for all sets of k centers

Definition

Let $P \subset \mathbb{R}^d$ be a point set. If $P' \subset \mathbb{R}^d$ and $\Delta \in \mathbb{R}$ sat. $\forall C \subset \mathbb{R}^d, |C| = k$:

$$|\operatorname{cost}(P, C) - \operatorname{cost}(P', C) + \Delta| \leq \varepsilon \cdot \operatorname{cost}(P, C)$$

and P'...

- ...satisfies that $|P'| \ll |P| \rightsquigarrow$ then P' is a coreset
- ...has intrinsic dimension s for an $s \ll d \rightsquigarrow$ dim. reduction

Reducing the dimension and number of input points

Coresets / Dim. Red

Reduce complexity,
but approximately preserve k-means cost for all sets of k centers

Definition

Let $P \subset \mathbb{R}^d$ be a point set. If $P' \subset \mathbb{R}^d$ and $\Delta \in \mathbb{R}$ sat. $\forall C \subset \mathbb{R}^d, |C| = k$:

$$|\operatorname{cost}(P, C) - \operatorname{cost}(P', C) + \Delta| \leq \varepsilon \cdot \operatorname{cost}(P, C)$$

and P'...

- ...satisfies that $|P'| \ll |P| \rightsquigarrow$ then P' is a coreset
- ...has intrinsic dimension s for an $s \ll d \rightsquigarrow$ dim. reduction

Input and coreset look alike for every possible solution!

Coresets for k-means

Dimensionality reduction for k-means

[CMEP15] Cohen, Elder, Musco, Musco, Persu, Dim. reduction for k-means clustering and low rank approximation, STOC 2015.
[DFKVV04] Drineas, Frieze, Kannan, Vempala, Vinay, Clustering large graphs via the svd, Maschine Learning 2004.
[FL11] Feldman, Langberg, A unified framework for approximating and clustering data, STOC 2011.
[FSS13] Feldman, S., Sohler, Turning Big Data into Tiny Data: Constant-size C. for k-means, PCA and Pr. Cl., SODA 2013.
[HPM04] Har-Peled, Mazumdar, On coresets for k-means and k-median clustering, STOC 2004.
[JL84] Johnson, Lindenstrauss, Extensions of lipschitz mappings into a hilbert space, Contemporary Mathematics 1984.

Coresets for k-means

	Number of points	Remarks
[HPM04]	$\mathcal{O}(k\varepsilon^{-d}\log n)$	First coreset
[FL11]	$\mathcal{O}(dk\varepsilon^{-4})$	Independent of $\log n$
[FSS13], [CMEP15]	$\mathcal{O}(k^2\varepsilon^{-5})$	Independent of d, $\log n$

Dimensionality reduction for k-means

[CMEP15] Cohen, Elder, Musco, Musco, Persu, Dim. reduction for k-means clustering and low rank approximation, STOC 2015.
[DFKVV04] Drineas, Frieze, Kannan, Vempala, Vinay, Clustering large graphs via the svd, Maschine Learning 2004.
[FL11] Feldman, Langberg, A unified framework for approximating and clustering data, STOC 2011.
[FSS13] Feldman, S., Sohler, Turning Big Data into Tiny Data: Constant-size C. for k-means, PCA and Pr. Cl., SODA 2013.
[HPM04] Har-Peled, Mazumdar, On coresets for k-means and k-median clustering, STOC 2004.
[JL84] Johnson, Lindenstrauss, Extensions of lipschitz mappings into a hilbert space, Contemporary Mathematics 1984.

Coresets for k-means

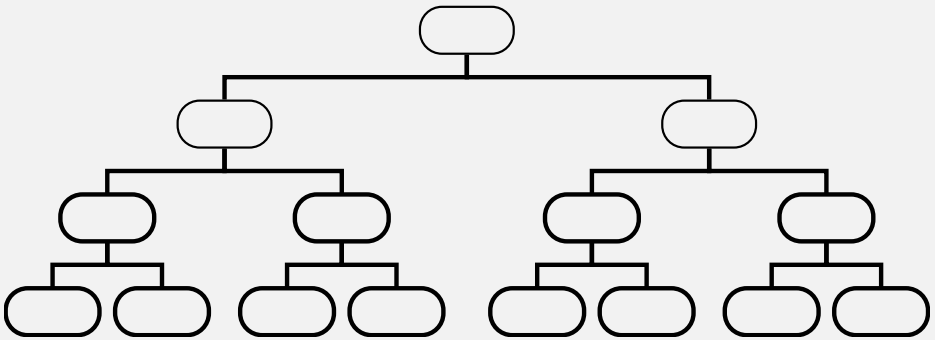
	Number of points	Remarks
[HPM04]	$\mathcal{O}(k\varepsilon^{-d}\log n)$	First coreset
[FL11]	$\mathcal{O}(dk\varepsilon^{-4})$	Independent of $\log n$
[FSS13], [CMEP15]	$\mathcal{O}(k^2\varepsilon^{-5})$	Independent of d, $\log n$

Dimensionality reduction for k-means

	Target dimension	Quality guarantee
[JL84]	$\Theta(\varepsilon^{-2}\log n)$	any $\varepsilon \in (0,1) \Rightarrow 1+\varepsilon$
[DFKVV04]	k	only $\varepsilon = 1 \quad \Rightarrow 2$
[FSS13], [CMEP15]	$\lceil k/\varepsilon \rceil$	any $\varepsilon \in (0,1] \Rightarrow 1+\varepsilon$

[CMEP15] Cohen, Elder, Musco, Musco, Persu, Dim. reduction for k-means clustering and low rank approximation, STOC 2015.
[DFKVV04] Drineas, Frieze, Kannan, Vempala, Vinay, Clustering large graphs via the svd, Maschine Learning 2004.
[FL11] Feldman, Langberg, A unified framework for approximating and clustering data, STOC 2011.
[FSS13] Feldman, S., Sohler, Turning Big Data into Tiny Data: Constant-size C. for k-means, PCA and Pr. Cl., SODA 2013.
[HPM04] Har-Peled, Mazumdar, On coresets for k-means and k-median clustering, STOC 2004.
[JL84] Johnson, Lindenstrauss, Extensions of lipschitz mappings into a hilbert space, Contemporary Mathematics 1984.

Conversion to a Streaming Algorithm: Merge & Reduce [BS80]

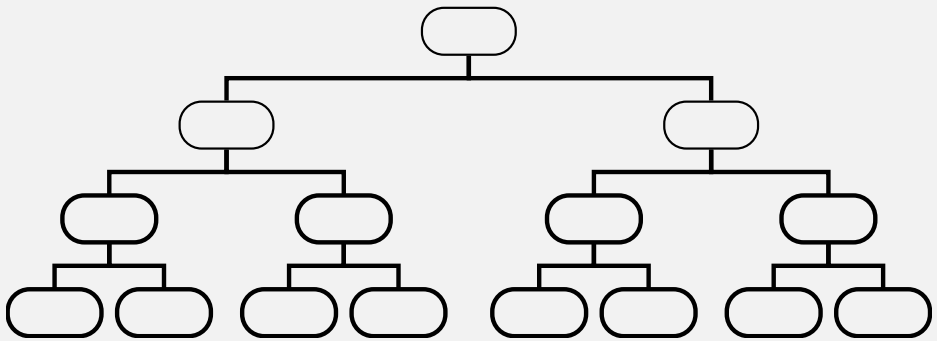


- read data in blocks
- compute a coreset for each block $\rightarrow s$
- merge and reduce coresets in a tree fashion
- $\rightsquigarrow$ space $s \cdot \log n$

Morally: Merge & Reduce causes overhead $\mathcal{O}(\log^c n)$

[BS80] Bentley, Saxe, Decomposable searching problems I: Static-to-dynamic transformation, Journal of Algorithms, 1980.

Conversion to a Streaming Algorithm: Merge & Reduce [BS80]



- read data in blocks
- compute a coreset for each block $\rightarrow s$
- merge and reduce coresets in a tree fashion
- $\rightsquigarrow$ space $s \cdot \log n$

Morally: Merge & Reduce causes overhead $\mathcal{O}(\log^c n)$

Necessary Requirement: Composability

P_1' and P_2' coresets for P_1 and $P_2 \Rightarrow P_1' \cup P_2'$ coreset for $P_1 \cup P_2$

[BS80] Bentley, Saxe, Decomposable searching problems I: Static-to-dynamic transformation, Journal of Algorithms, 1980.

Constraints and Composability

Constraints and Composability

For k-means, coresets are composable because the cost is linear:

$$\text{cost}(P_1 \cup P_2, C) = \text{cost}(P_1, C) + \text{cost}(P_2, C) \quad \forall P_1, P_2, C$$

Constraints and Composability

For k-means, coresets are composable because the cost is linear:

$$\text{cost}(P_1 \cup P_2, C) = \text{cost}(P_1, C) + \text{cost}(P_2, C) \quad \forall P_1, P_2, C$$

For problems with constraints, cost can increase or decrease!

Constraints and Composability

For k-means, coresets are composable because the cost is linear:

$$\text{cost}(P_1 \cup P_2, C) = \text{cost}(P_1, C) + \text{cost}(P_2, C) \quad \forall P_1, P_2, C$$

For problems with constraints, cost can increase or decrease!

lower-bounded k-means with $k = 2$ and $L = 5$

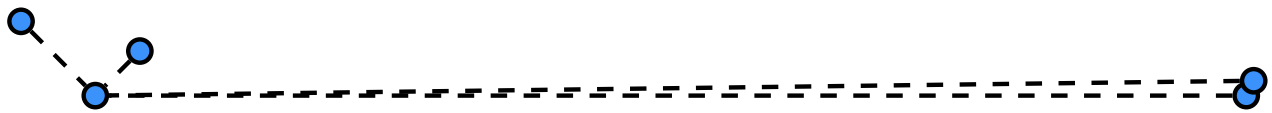
Constraints and Composability

For k-means, coresets are composable because the cost is linear:

$$\text{cost}(P_1 \cup P_2, C) = \text{cost}(P_1, C) + \text{cost}(P_2, C) \quad \forall P_1, P_2, C$$

For problems with constraints, cost can increase or decrease!

lower-bounded k-means with $k = 2$ and $L = 5$



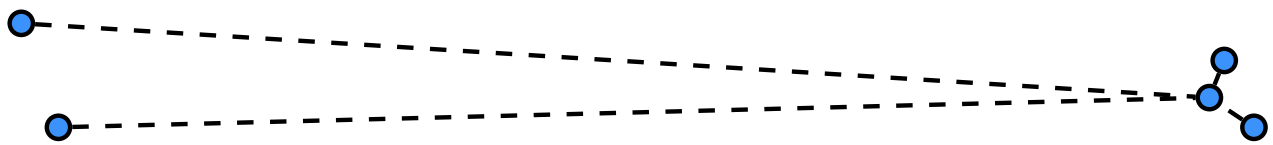
Constraints and Composability

For k-means, coresets are composable because the cost is linear:

$$\text{cost}(P_1 \cup P_2, C) = \text{cost}(P_1, C) + \text{cost}(P_2, C) \quad \forall P_1, P_2, C$$

For problems with constraints, cost can increase or decrease!

lower-bounded k-means with $k = 2$ and $L = 5$



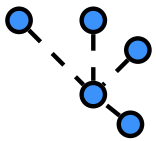
Constraints and Composability

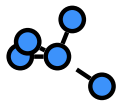
For k-means, coresets are composable because the cost is linear:

$$\text{cost}(P_1 \cup P_2, C) = \text{cost}(P_1, C) + \text{cost}(P_2, C) \quad \forall P_1, P_2, C$$

For problems with constraints, cost can increase or decrease!

lower-bounded k-means with $k = 2$ and $L = 5$





S., Schwiegelssohn, Sohler, 2018

S., Schwiegelssohn, Sohler, 2018

- Composable coresets for fair k-means

S., Schwiegelssohn, Sohler, 2018

- Composable coresets for fair k-means
- Size is $\mathcal{O}(k\varepsilon^{-d}\log n)$

S., Schwiegelssohn, Sohler, 2018

- Composable coresets for fair k-means
- Size is $\mathcal{O}(k\varepsilon^{-d}\log n)$
- Based on geometric coreset construction

S., Schwiegelssohn, Sohler, 2018

- Composable coresets for fair k-means
- Size is $\mathcal{O}(k\varepsilon^{-d}\log n)$
- Based on geometric coreset construction

Final open questions

S., Schwiegelssohn, Sohler, 2018

- Composable coresets for fair k-means
- Size is $\mathcal{O}(k\varepsilon^{-d}\log n)$
- Based on geometric coreset construction

Final open questions

- Composable coresets for other constraints?

S., Schwiegelssohn, Sohler, 2018

- Composable coresets for fair k-means
- Size is $\mathcal{O}(k\varepsilon^{-d}\log n)$
- Based on geometric coreset construction

Final open questions

- Composable coresets for other constraints?
- Smaller size by using sampling?

S., Schwiegelssohn, Sohler, 2018

- Composable coresets for fair k-means
- Size is $\mathcal{O}(k\varepsilon^{-d}\log n)$
- Based on geometric coreset construction

Final open questions

- Composable coresets for other constraints?
- Smaller size by using sampling?

Thank you for your attention!