UNIVERSITY OF
EASTERN FINLAND

Random Swap algorithm

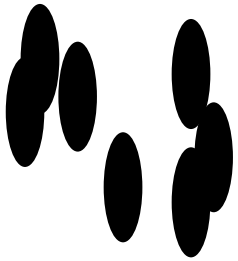
Pasi Fränti

24.4.2018

Definitions and data

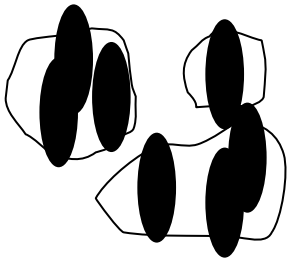
Set of *N* data points:

$X=\{x_1, x_2, \ldots, x_N\}$



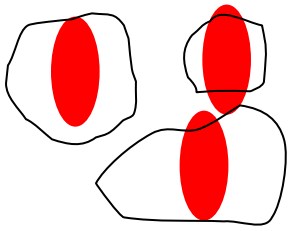
Partition of the data:

$P=\{p_1, p_2, \ldots, p_k\},$



Set of *k* cluster prototypes (centroids):

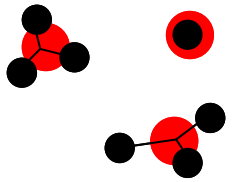
$C=\{c_1, c_2, \ldots, c_k\},$



Clustering problem

Objective function:

$$MSE = \frac{1}{dN} \cdot \sum_{i=1}^{N} d\left(x_i - C_{P_i}\right)^2$$



Optimality of partition:

$$P_i = \arg\min_{1 \le j \le k} \left\| x_i - C_j \right\|^2 \qquad \forall i \in [1, N]$$

Optimality of centroid:

$$C_j = \sum_{P_i = j} x_i \Big/ \sum_{P_i = j} 1 \qquad \forall j \in [1, k]$$

K-means algorithm

X = Data set
C = Cluster centroids
P = Partition

K-Means(X, C) $\rightarrow$ (C, P)

REPEAT

$C_{prev} \leftarrow C$;

FOR i=1 TO N DO
$p_i \leftarrow$ FindNearest(x_i, C);

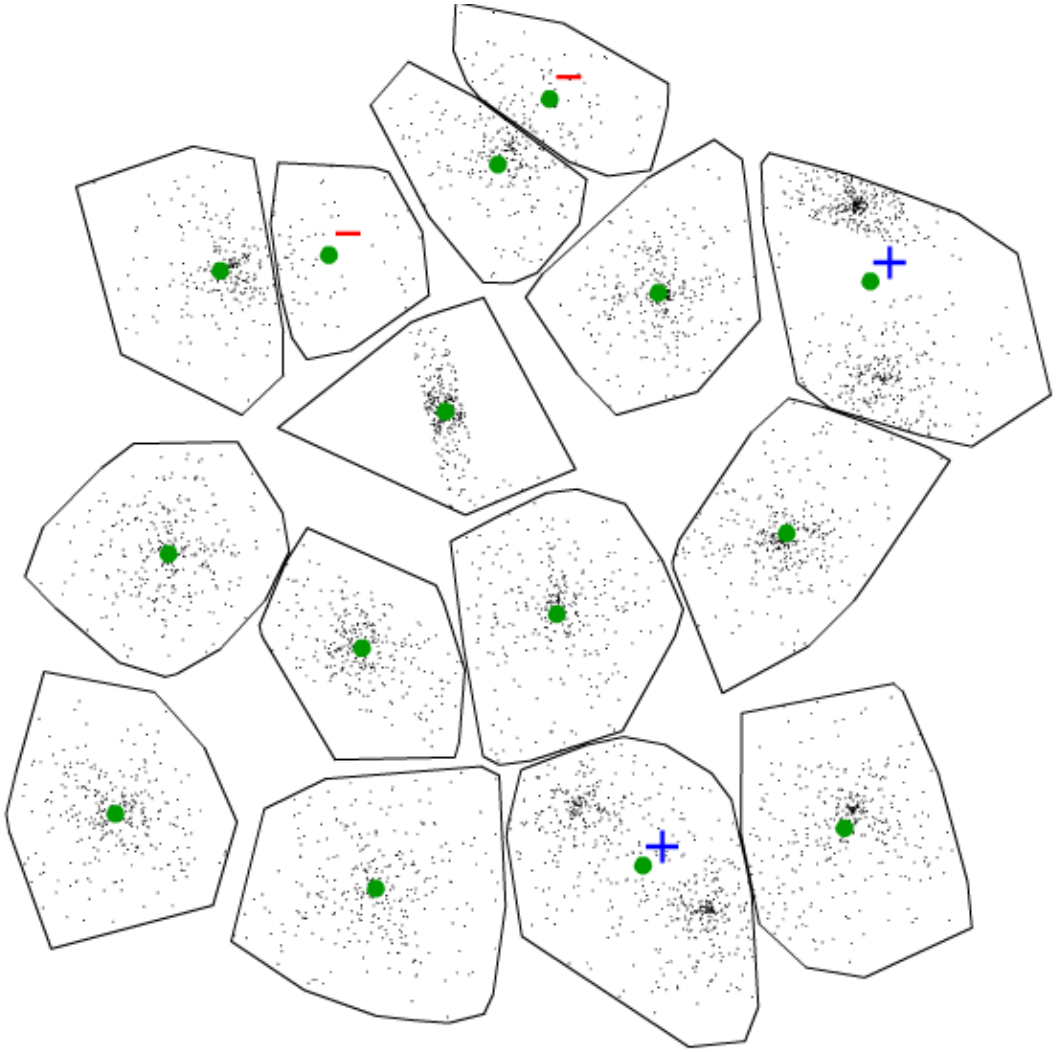
Optimal partition

FOR j=1 TO k DO
$c_j \leftarrow$ Average of $x_i \; \forall \; p_i = j$;

Optimal centoids

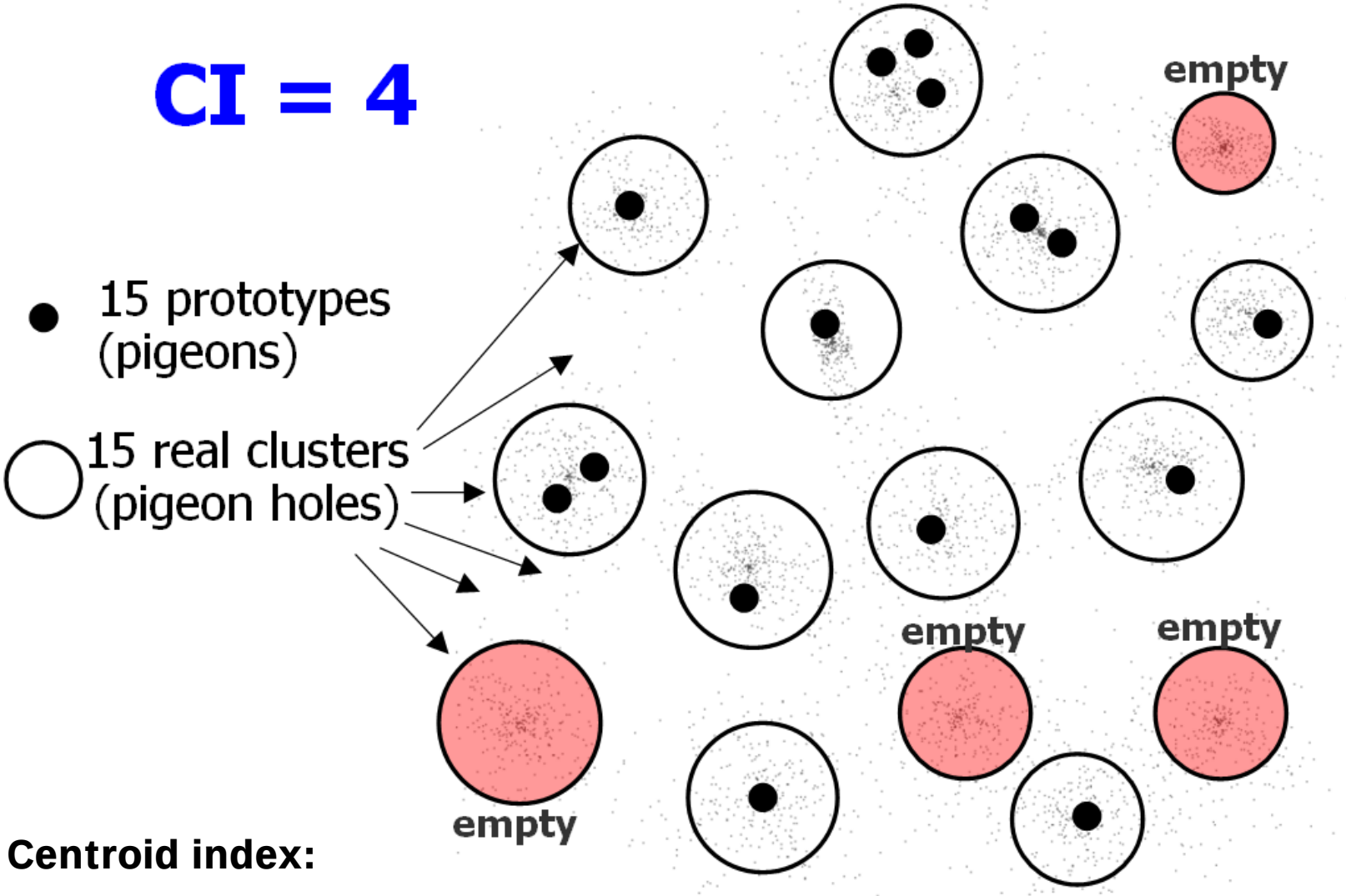
UNTIL $C = C_{prev}$

Problems of k-means



Swapping strategy

Pigeon hole principle



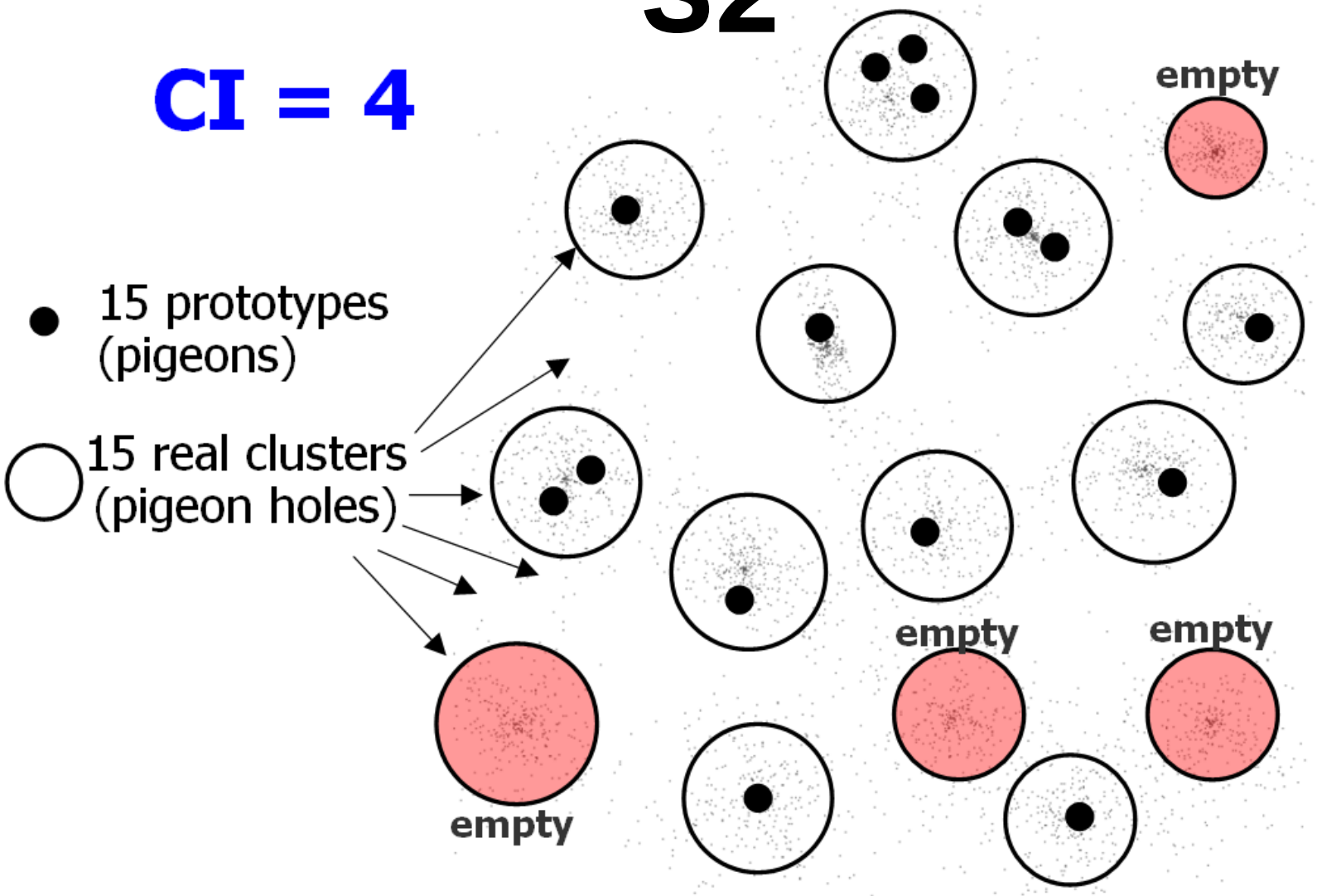
CI = Centroid index:

P. Fränti, M. Rezaei and Q. Zhao
"Centroid index: cluster level similarity measure"
Pattern Recognition, 47 (9), 3034-3045, September 2014, 2014.

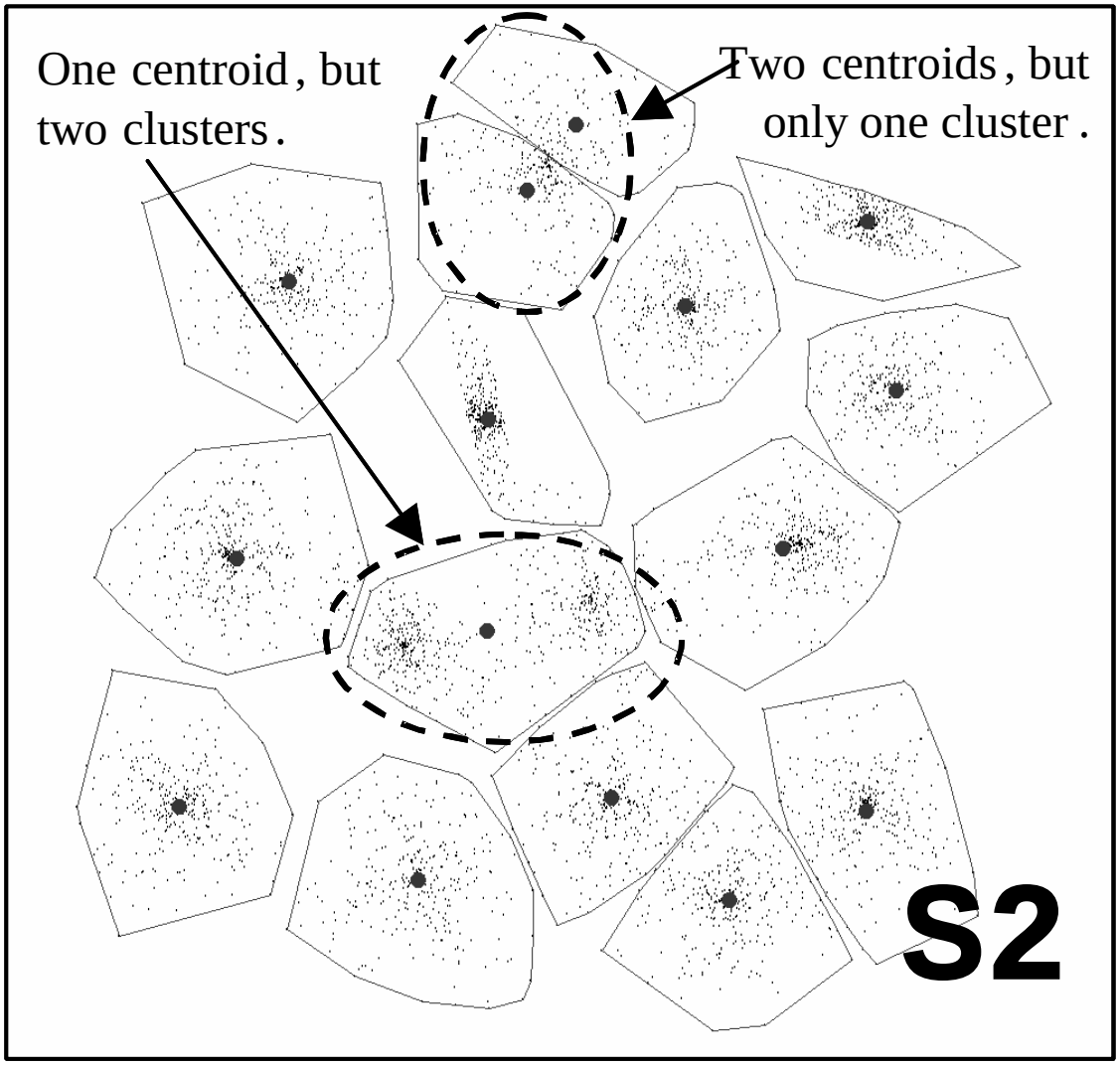
Pigeon hole principle

S2

CI = 4



Aim of the swap



Random Swap algorithm

Random Swap(X) $\rightarrow$ C, P

$C \leftarrow$ Select random representatives(X);
$P \leftarrow$ Optimal partition(X, C);
REPEAT T times
(C^{new},j) $\leftarrow$ Random swap(X, C);
$P^{new} \leftarrow$ Local repartition(X, C^{new}, P, j);
C^{new}, $P^{new} \leftarrow$ *Kmeans*(X, C^{new}, P^{new});
IF $f(C^{new}, P^{new}) < f(C, P)$ THEN
(C, P) $\leftarrow$ C^{new}, P^{new};
RETURN (C, P);

Steps of the swap

1. Random swap:

$$c_j \leftarrow x_i \,\big|\, j = random(1,k), i = random(1,N)$$

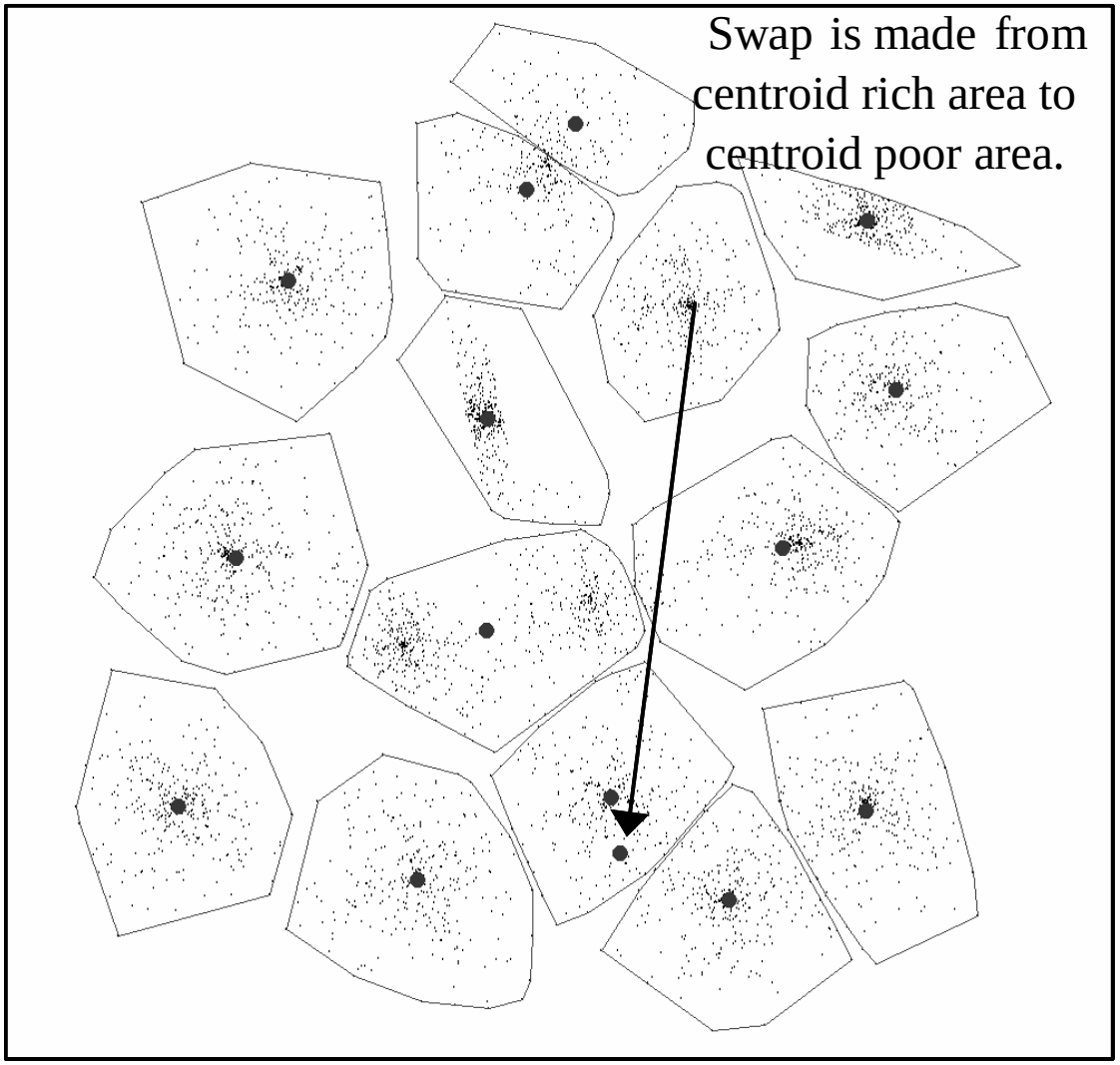
2. Re-allocate vectors from old cluster:

$$p_i \leftarrow \underset{1 \le j \le k}{\operatorname{argmin}} d\left(x_i, c_j\right)^2 \;\forall\; i \,\big|\, p_i = j$$

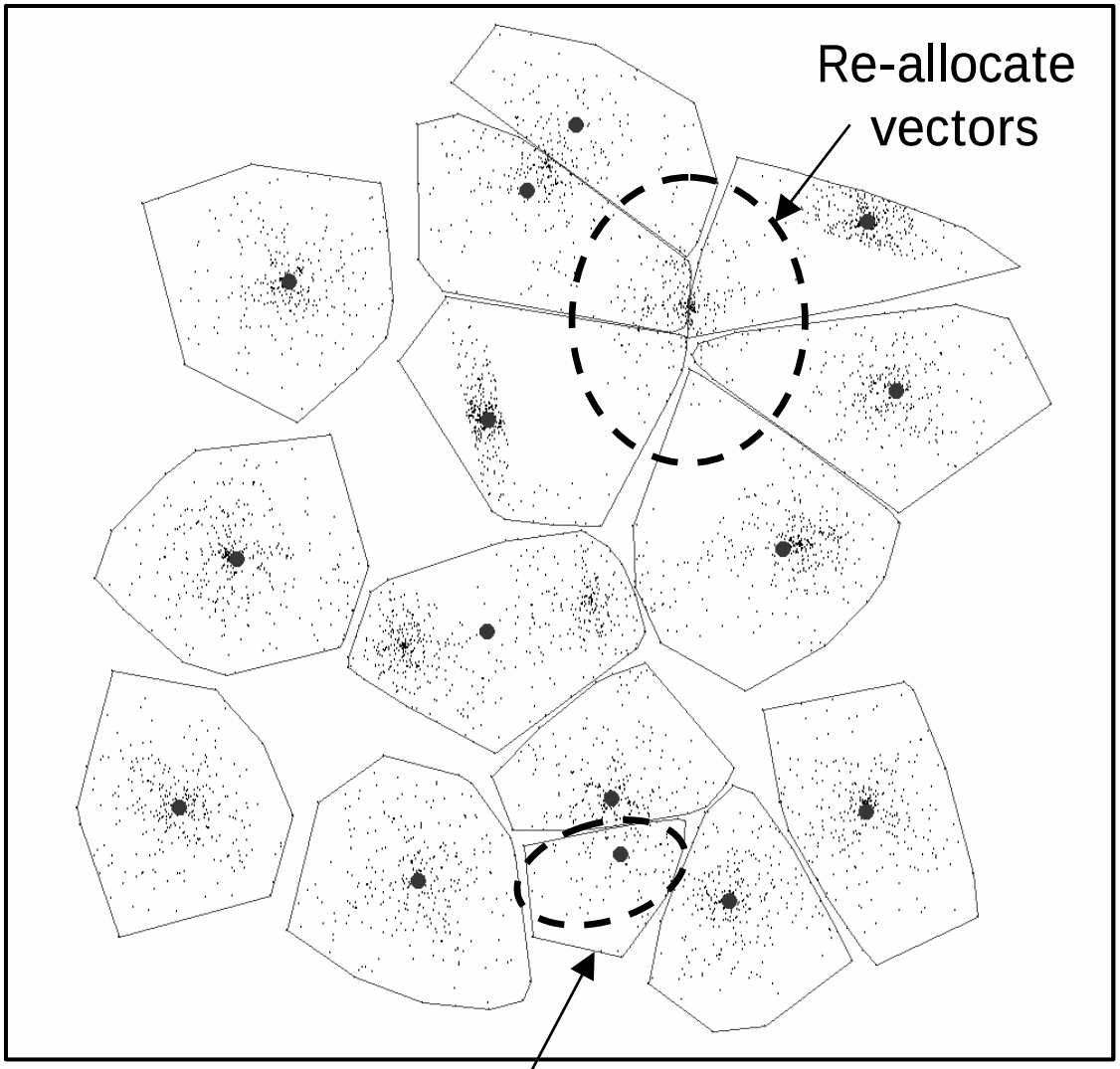
3. Create new cluster:

$$p_i \leftarrow \underset{k=j \,\vee\, k=p_i}{\arg\min} d\left(x_i, c_k\right)^2 \quad \forall \;\; i \in [1,N]$$

Swap

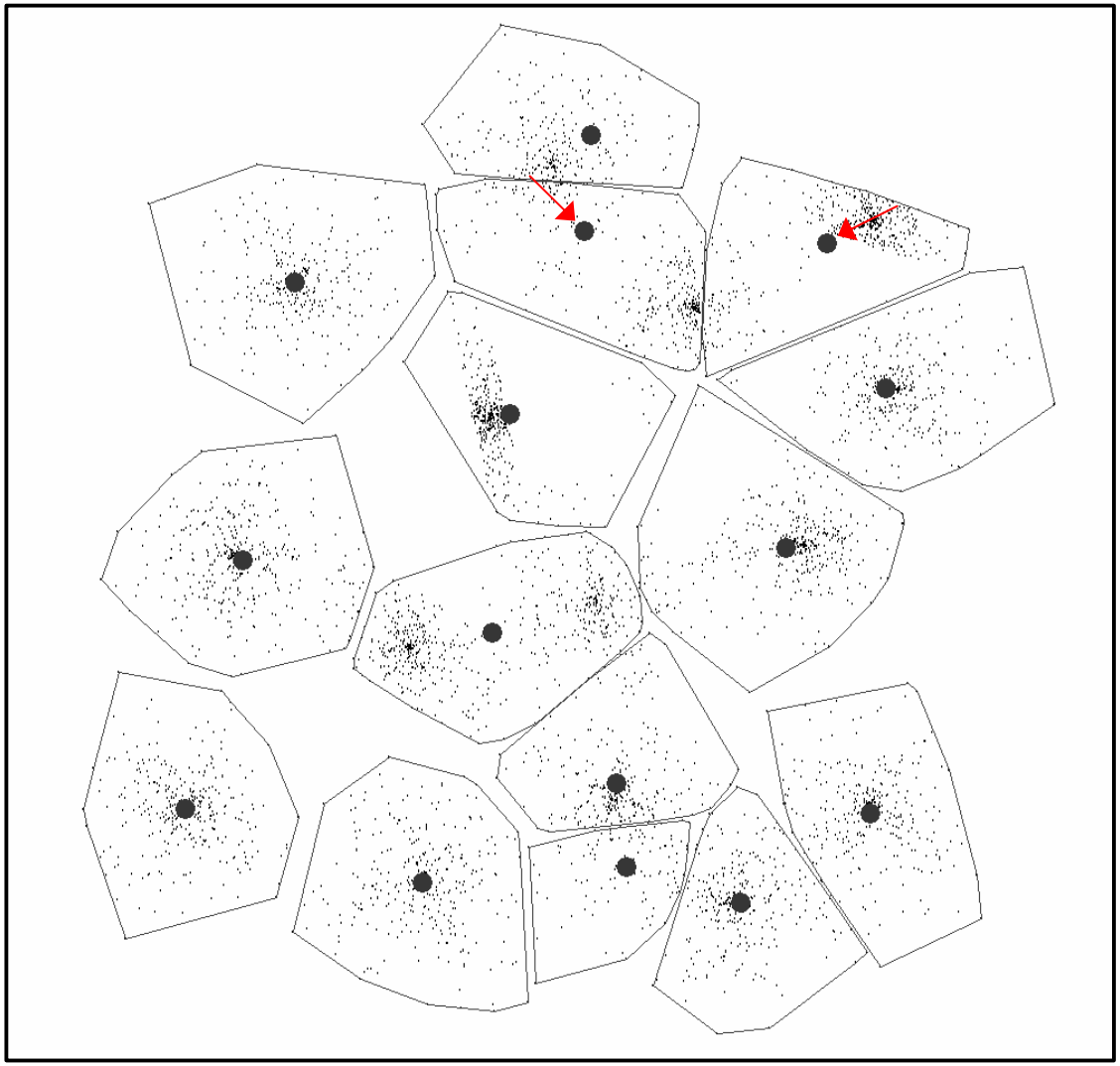


Local re-partition



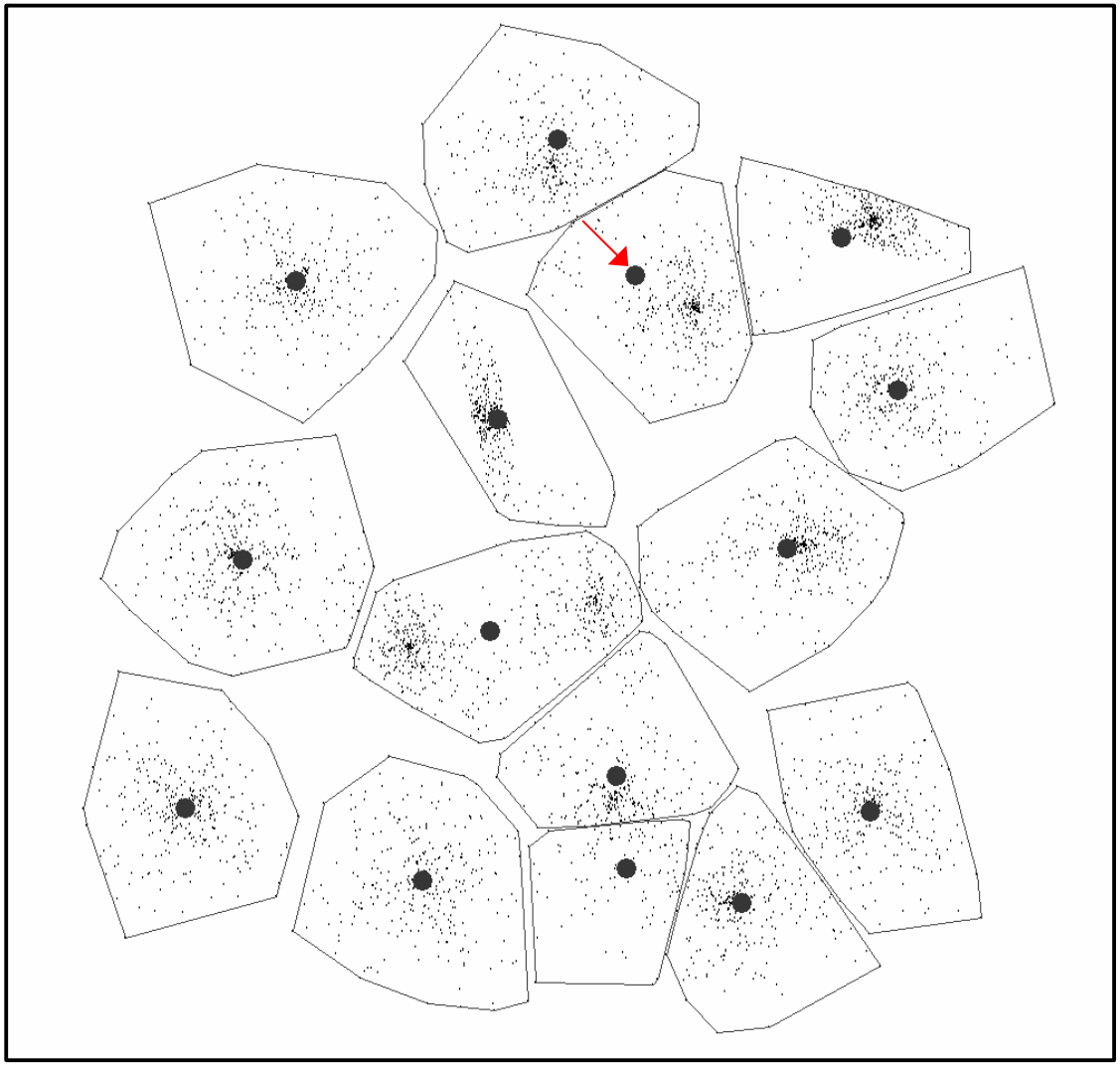
Iterate by k-means

1st iteration



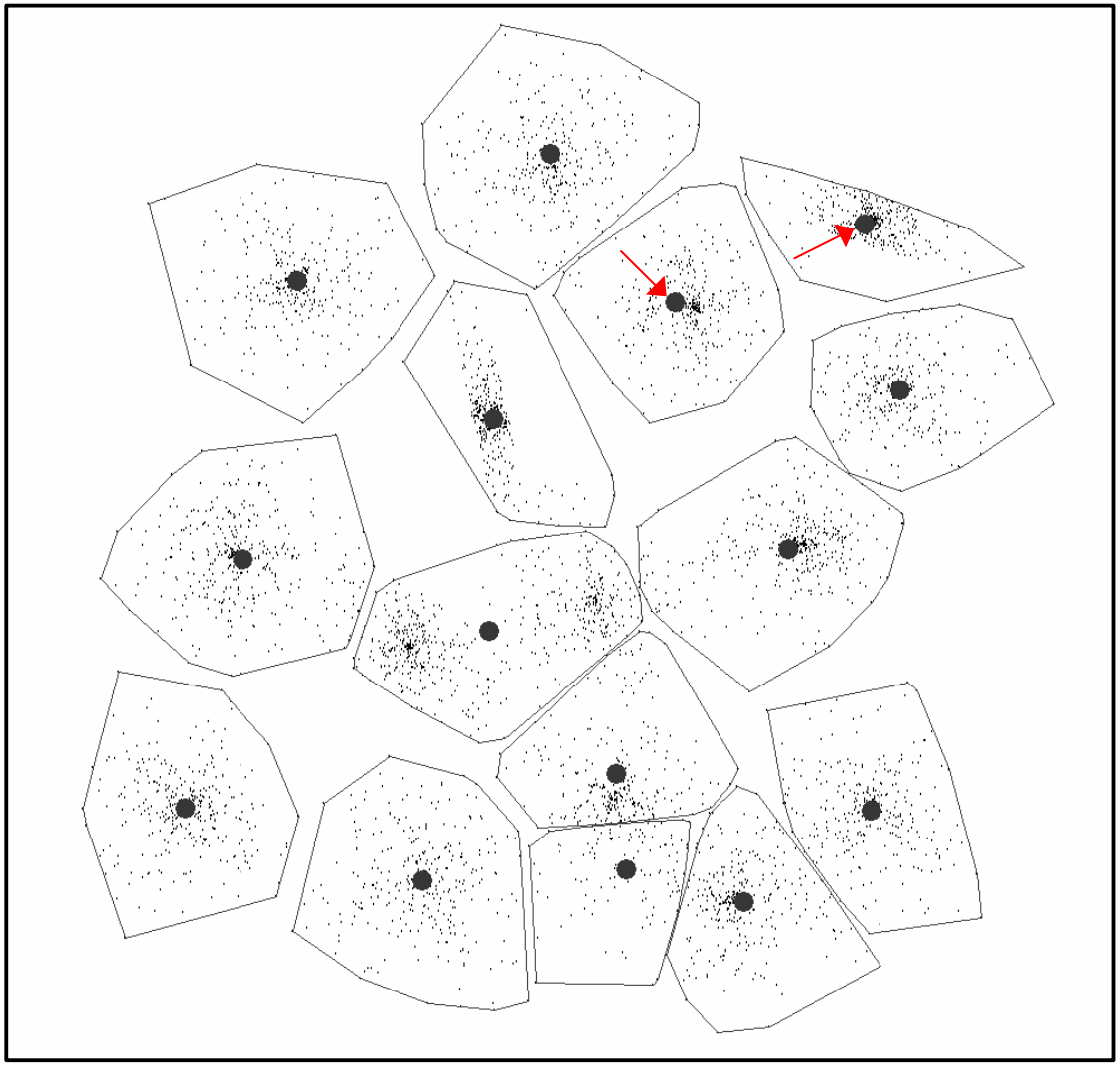
Iterate by k-means

2nd iteration



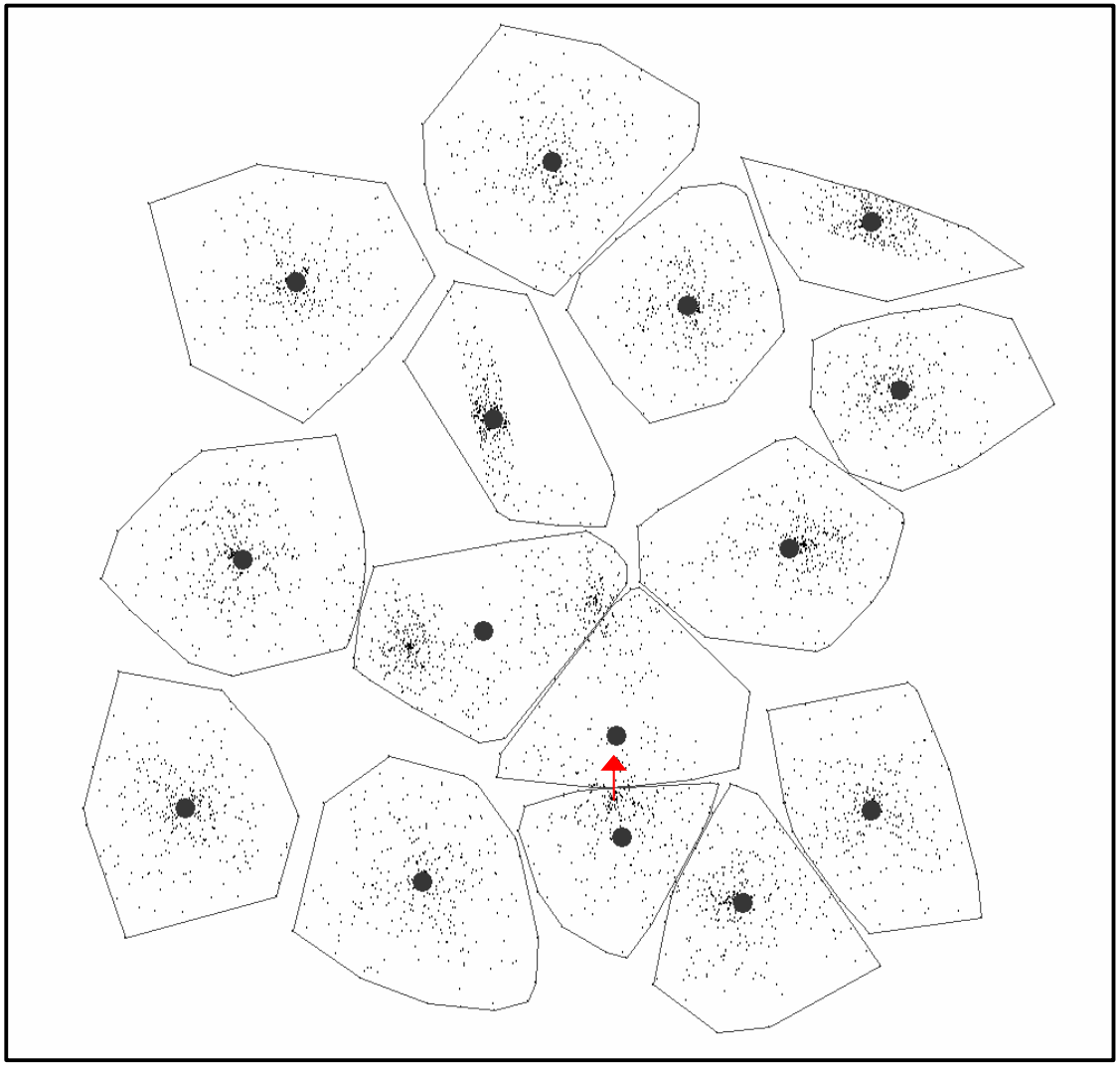
Iterate by k-means

3rd iteration



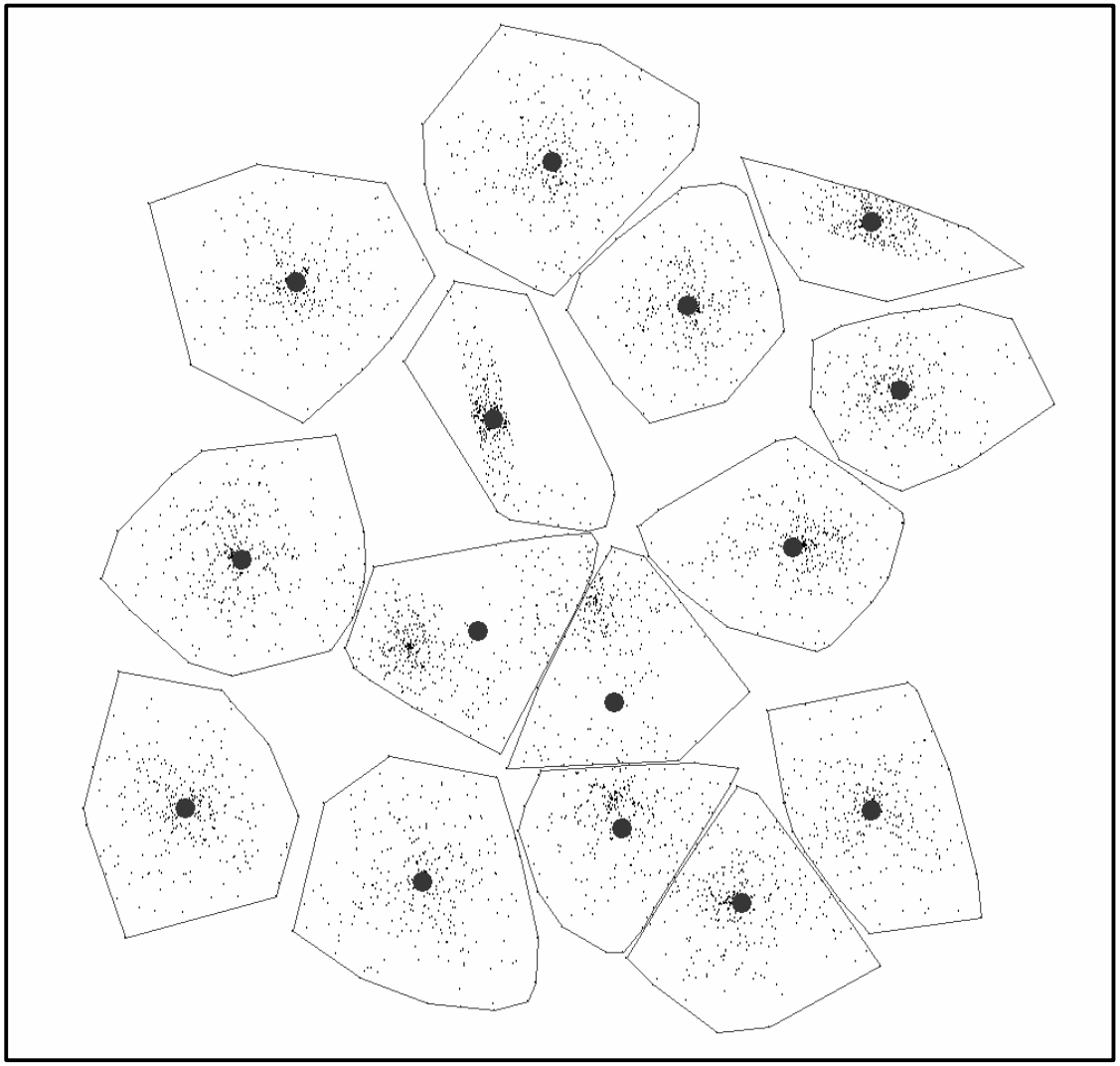
Iterate by k-means

16th iteration



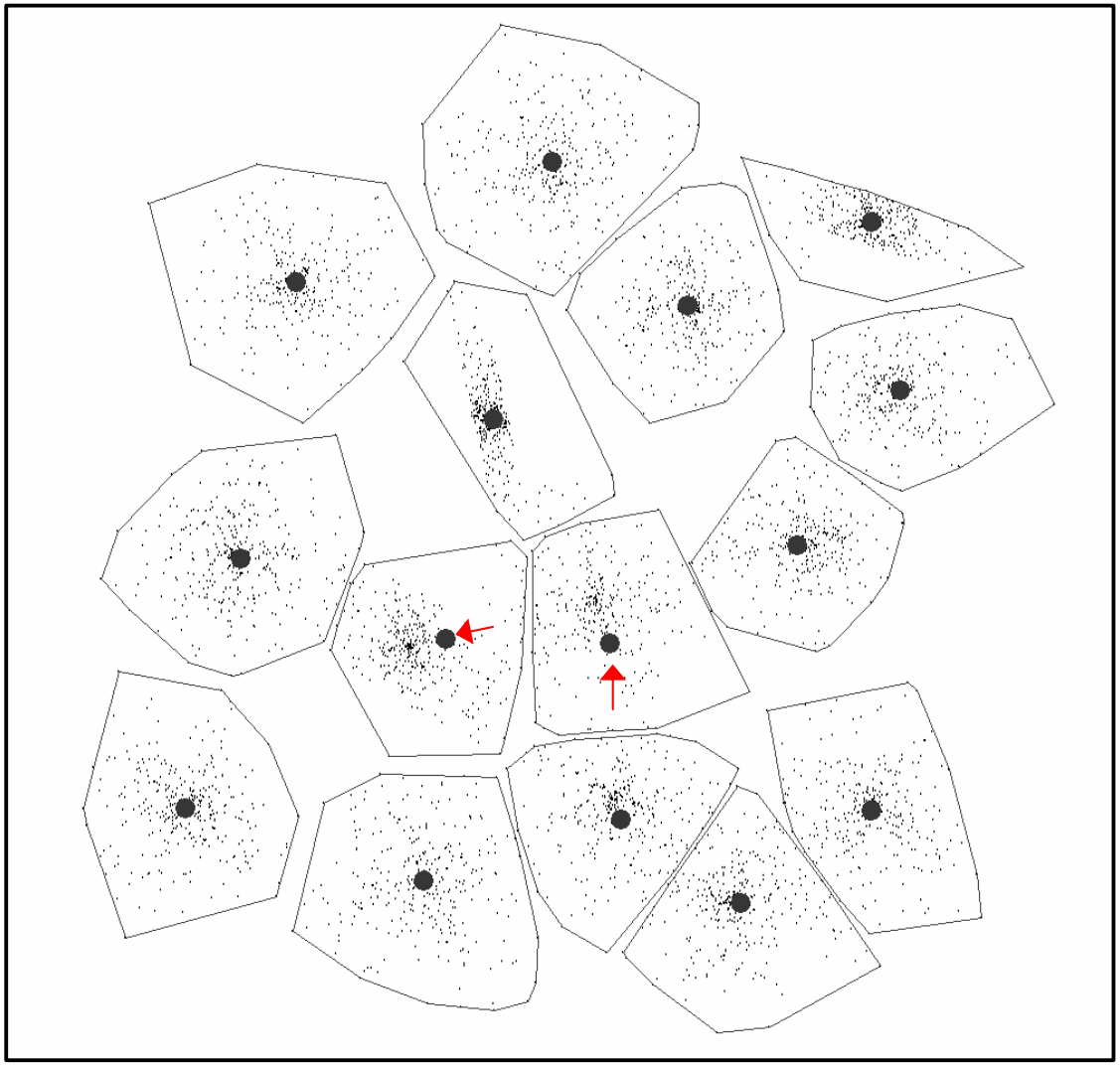
Iterate by k-means

17th iteration



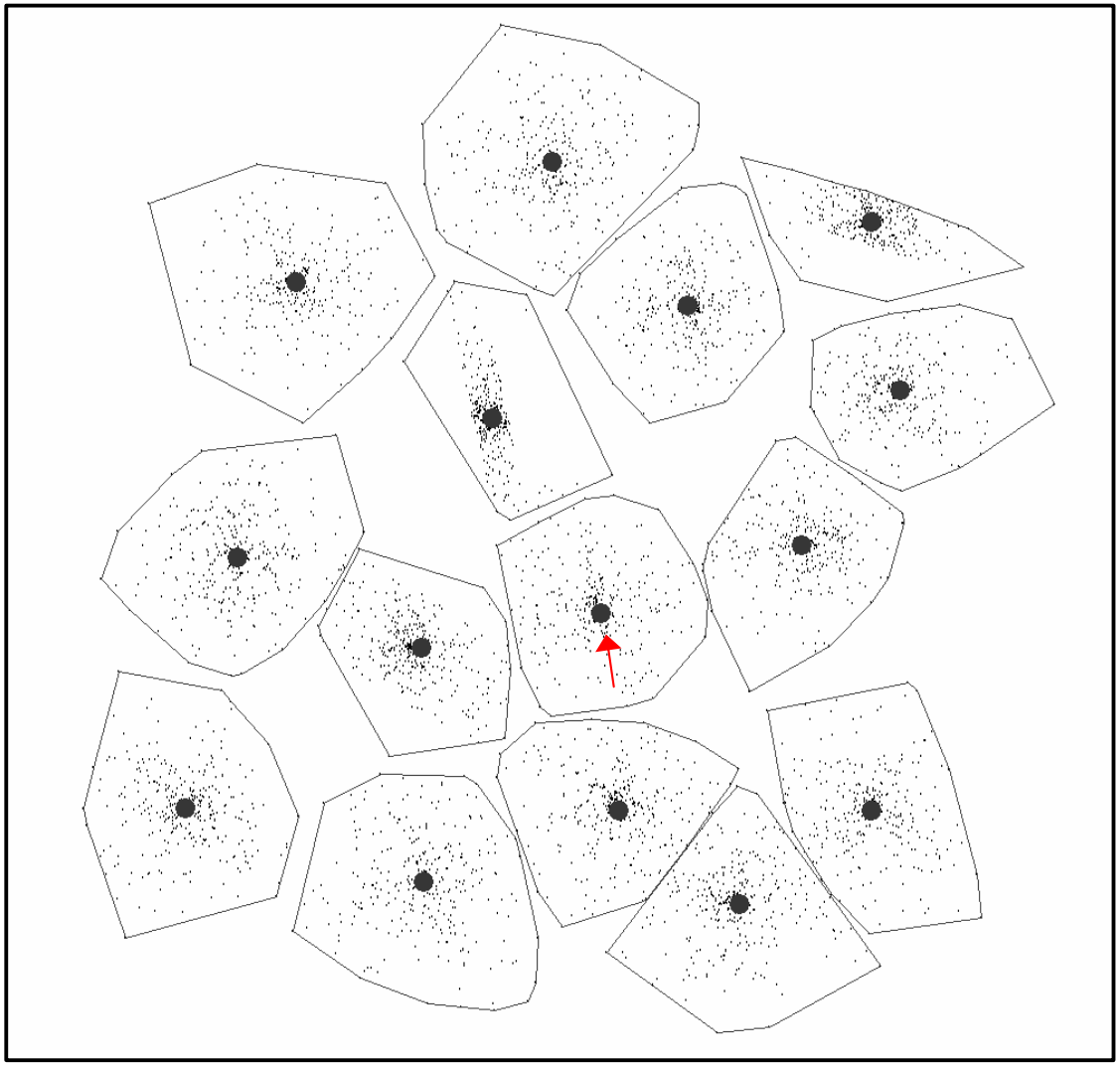
Iterate by k-means

18th iteration



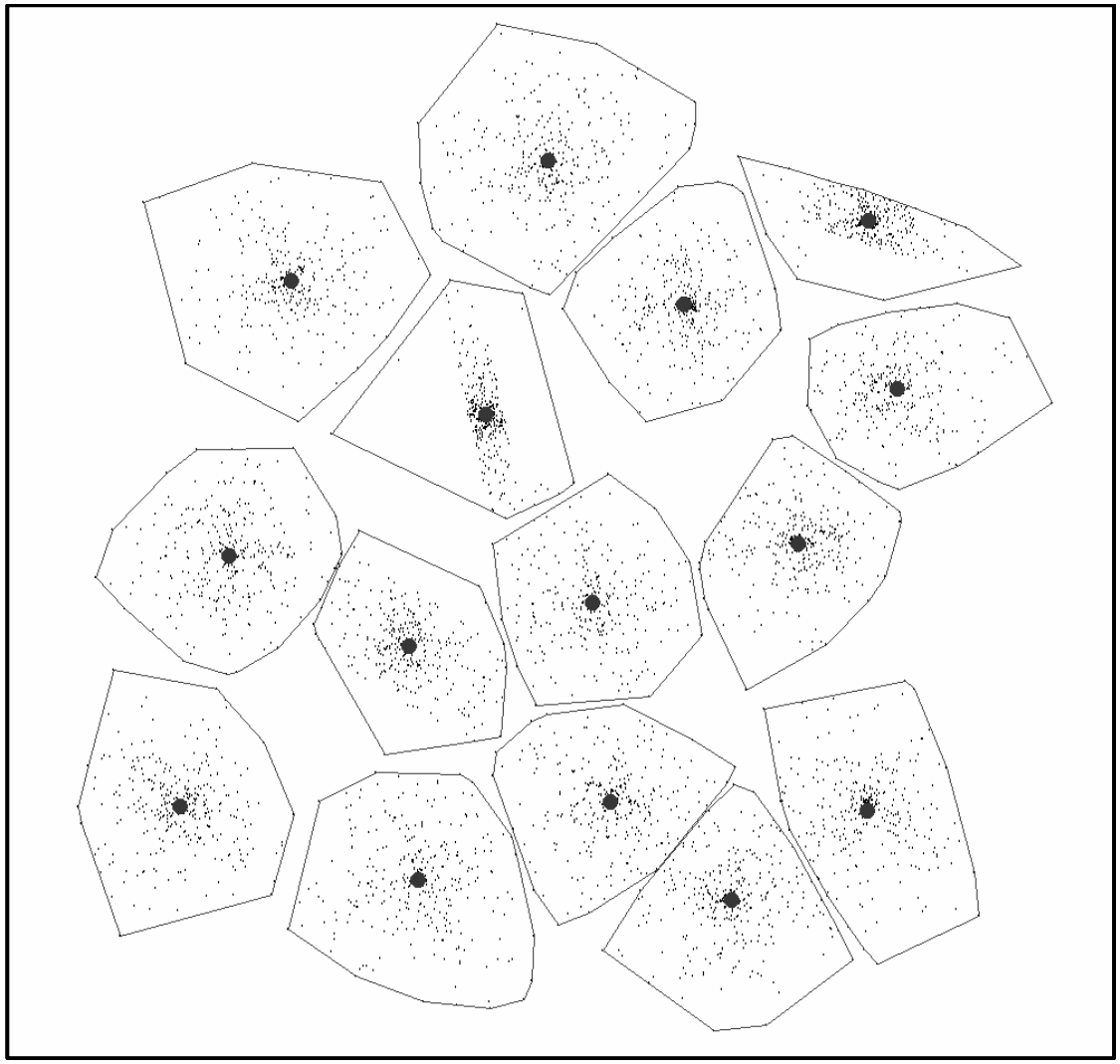
Iterate by k-means

19th iteration

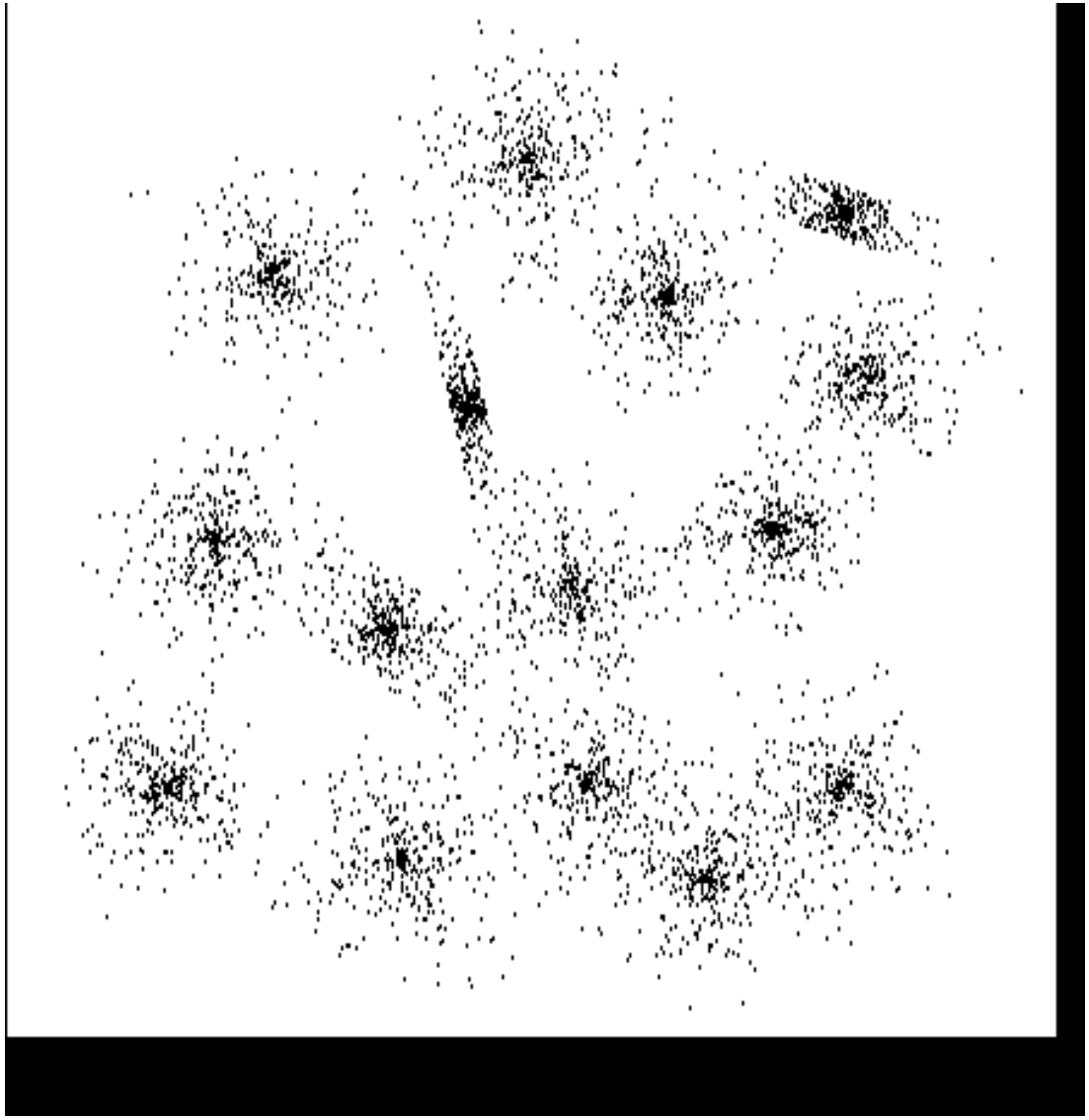


Final result

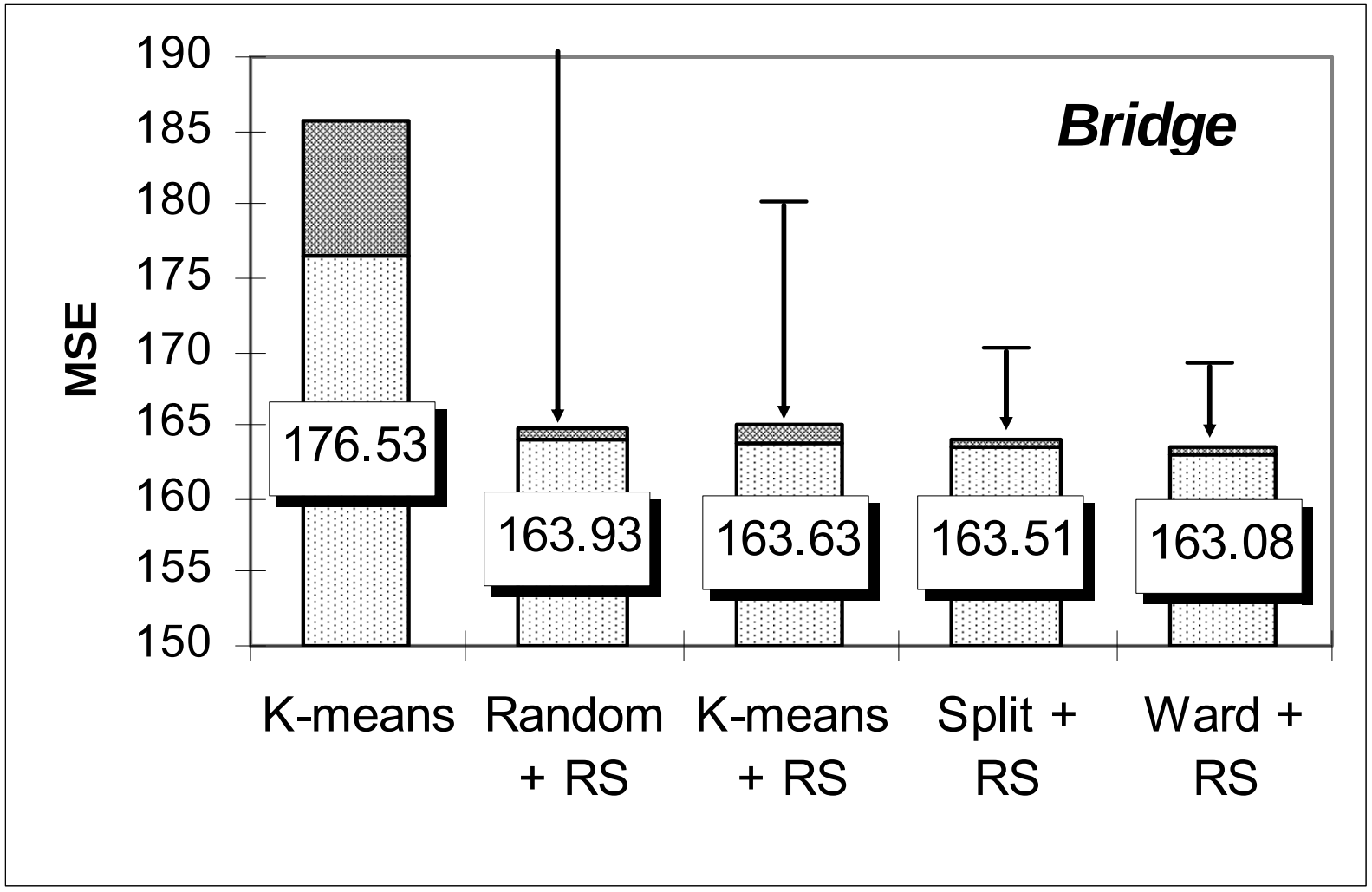
25 iterations



Extreme example



Dependency on initial solution



Data sets

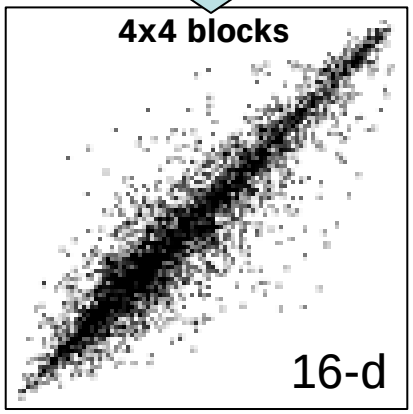
Data sets

Data set	Ref.	Type of data	Vectors (N)	Clusters (k)	Vectors per cluster	Dimension (d)
Bridge	[33]	Gray-scale image	4096	256	16	16
*House**	[33]	*RGB* image	34112	256	133	3
Miss America	[33]	Residual vectors	6480	256	25	16
Europe		Diff. coordinates	169673	256	663	2
$BIRCH_1$- $BIRCH_3$	[29]	Artificial	100000	100	1000	2
$S_1 - S_4$	[3]	Artificial	5000	15	333	2
Unbalance	[39]	Artificial	6500	8	821	2
Dim16 - Dim1024	[20]	Artificial	1024	16	64	16-1024
KDD04-Bio	[30]	DNA sequences	145751	2000	73	74

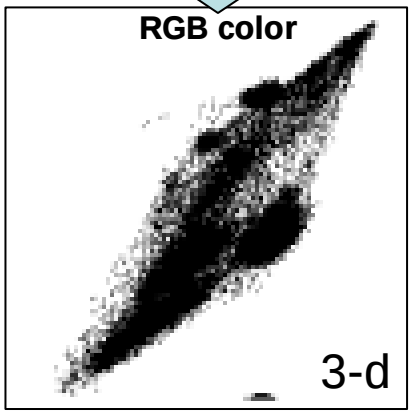
Data sets visualized

Images

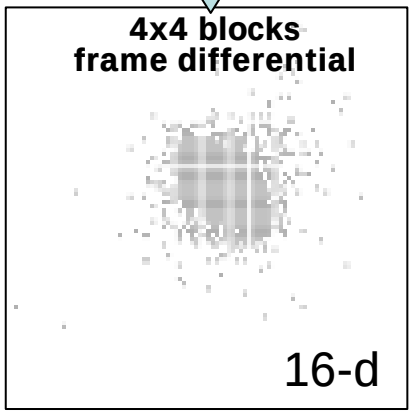
Bridge



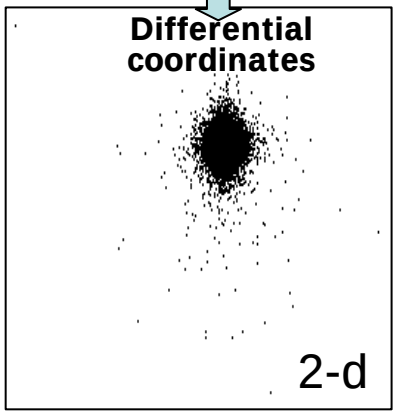
House



Miss America

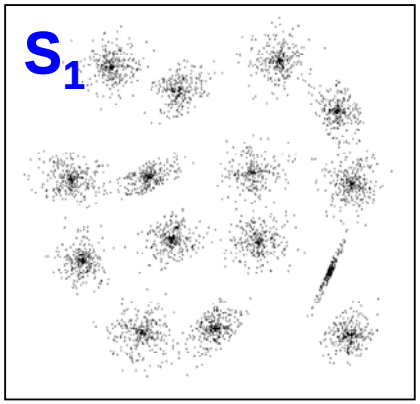


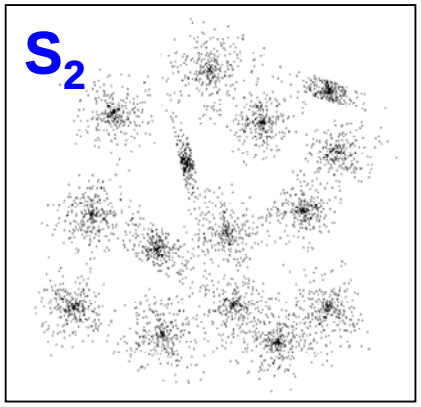
Europe

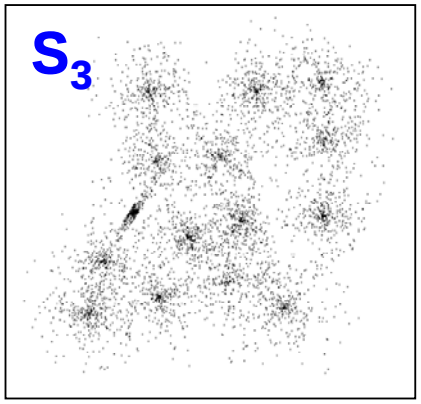


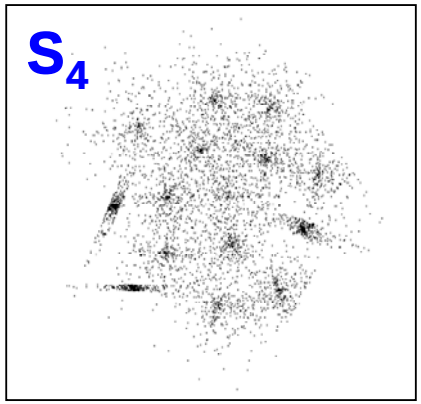
Data sets visualized

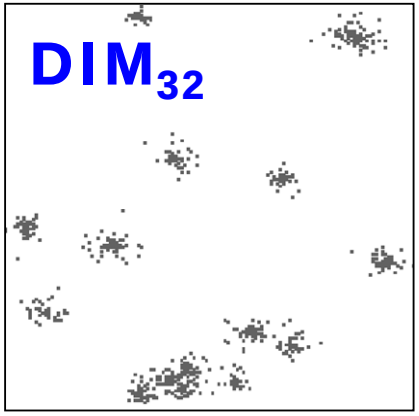
Artificial





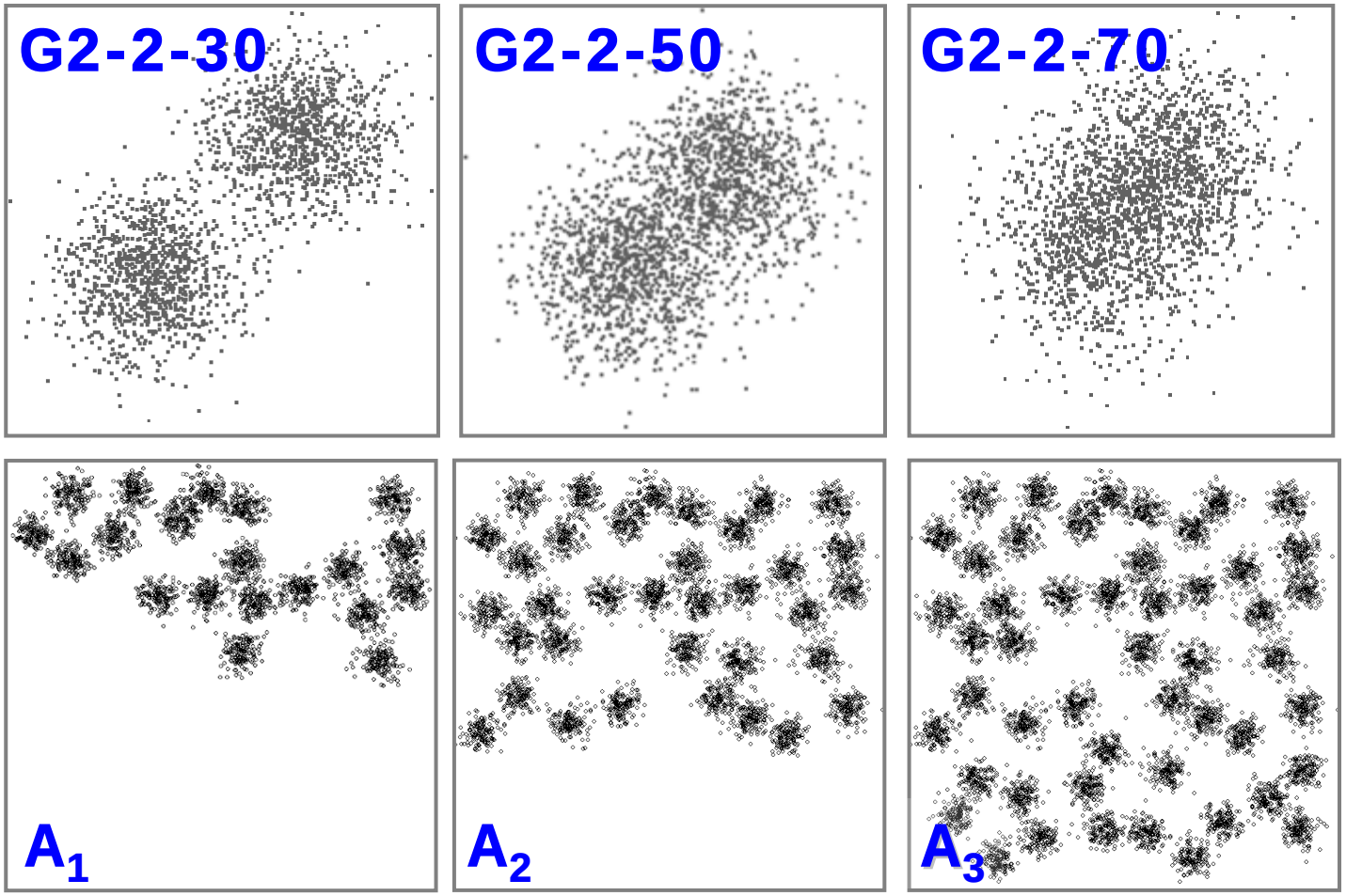






Data sets visualized

Artificial



Time complexity

Efficiency of the random swap

Total time to find correct clustering:

- Time per iteration $\times$ Number of iterations

Time complexity of single iteration:

- Swap: $O(1)$
- Remove cluster: $2k \cdot N/k = O(N)$
- Add cluster: $2N = O(N)$
- Centroids: $2N/k + 2N/k + 2\alpha = O(N/k)$
- K-means: $I \cdot k \cdot N = O(IkN)$

Bottleneck!

Efficiency of the random swap

Total time to find correct clustering:

- Time per iteration × Number of iterations

Time complexity of single iteration:

- Swap: $O(1)$
- Remove cluster: $2k \cdot N/k = O(N)$
- Add cluster: $2N = O(N)$
- Centroids: $2N/k + 2N/k + 2\alpha = O(N/k)$
- (Fast) K-means: $4\alpha \cdot N = O(\alpha N)$

2 iterations only!

T. Kaukoranta, P. Fränti and O. Nevalainen
"A fast exact GLA based on code vector activity detection"
IEEE Trans. on Image Processing, 9 (8), 1337-1342, August 2000.

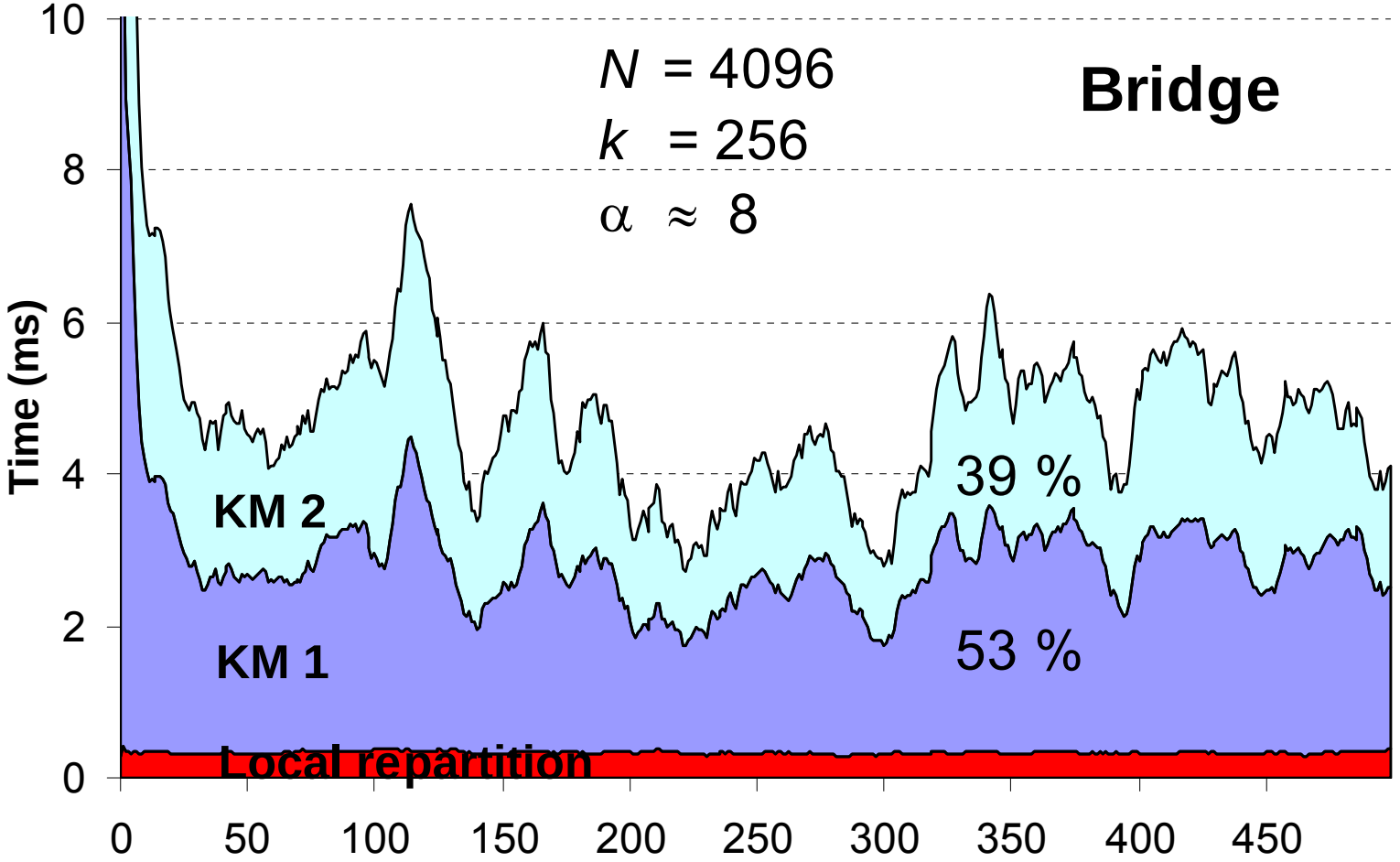
Estimated and observed steps

Bridge

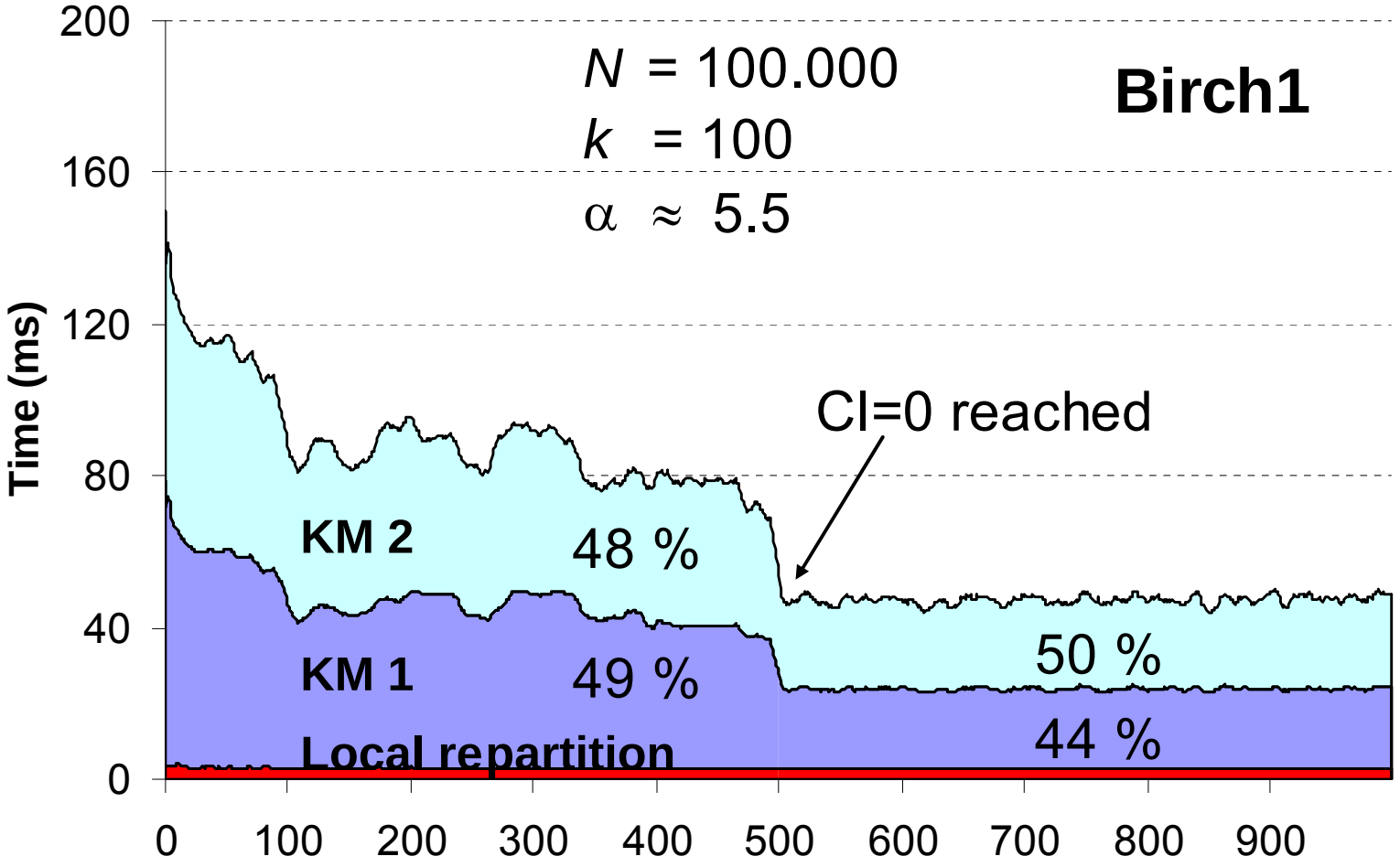
N=4096, k=256, N/k=16, $\alpha \approx 8$

Step:	Time comp.	Number of steps observed		
		50	100	500
Prototype swap	2	2	2	2
Cluster removal	$2N$	7526	8448	10137
Cluster addition	$2N$	8192	8192	8192
Prototype update	$4N/k + 2\alpha$	53	61	60
K-means iterations	$\leq 4\alpha N$	300901	285555	197327
Total	$O(\alpha N)$	316674	302258	215718

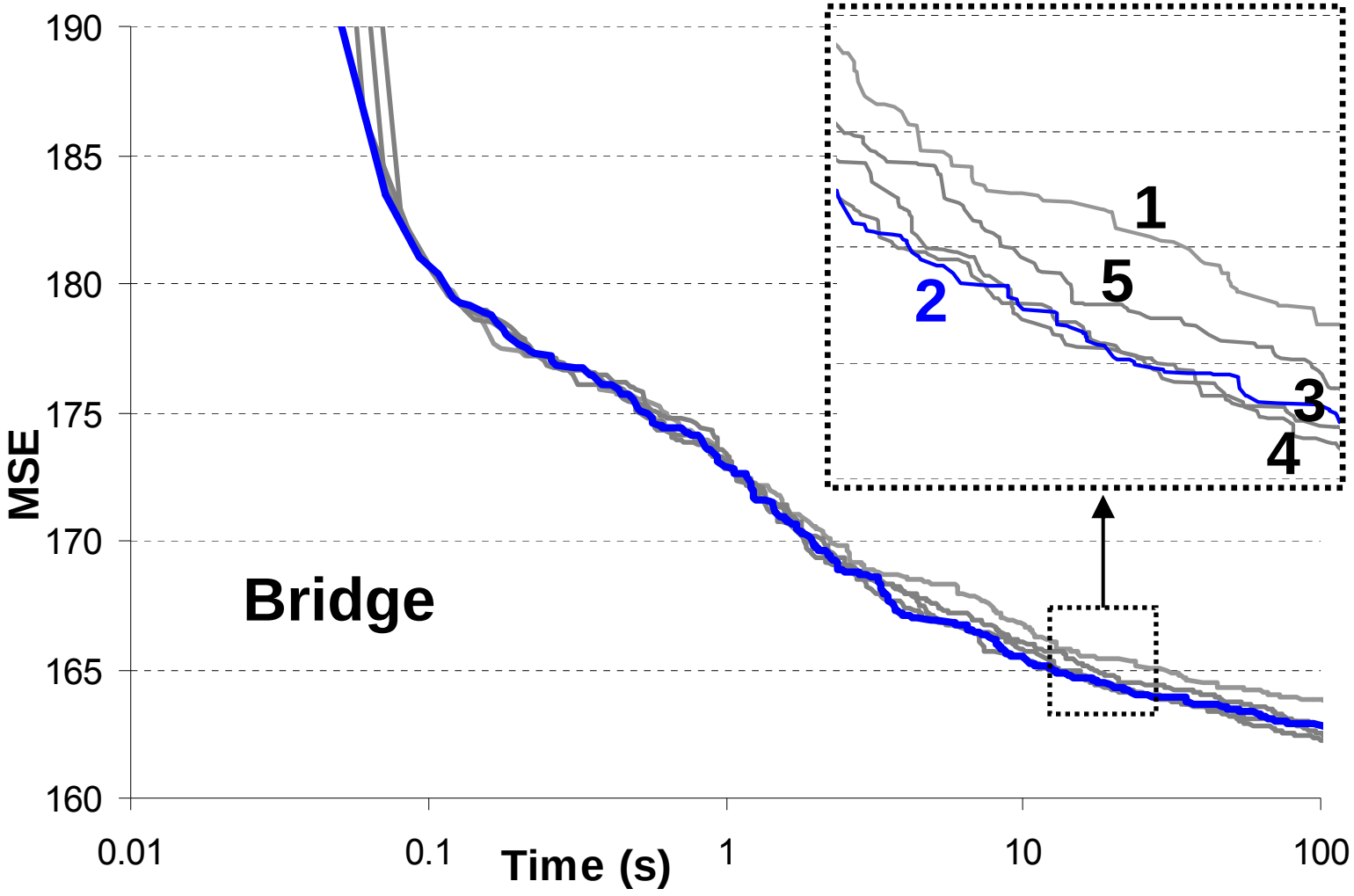
Processing time profile



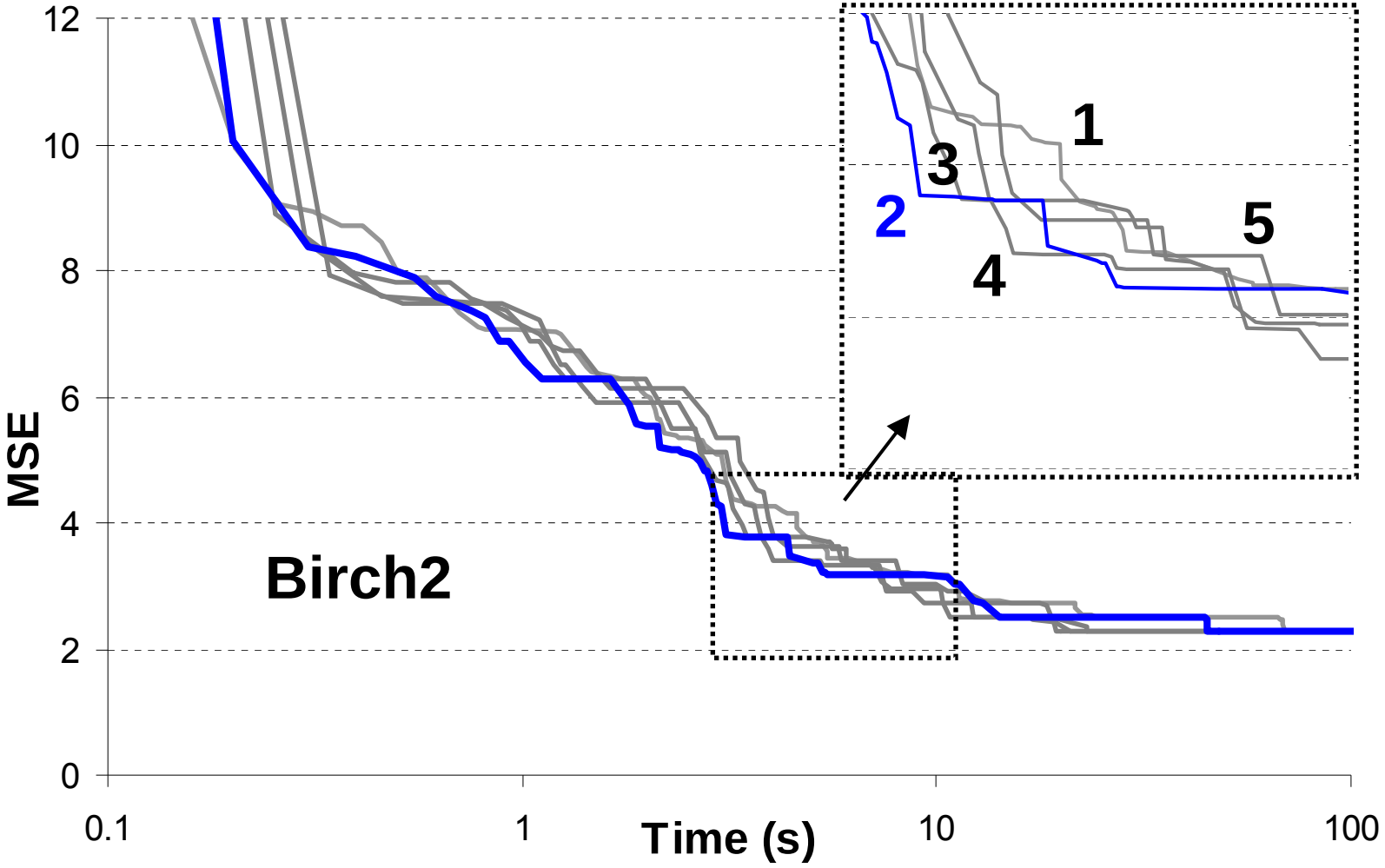
Processing time profile



Effect of K-means iterations



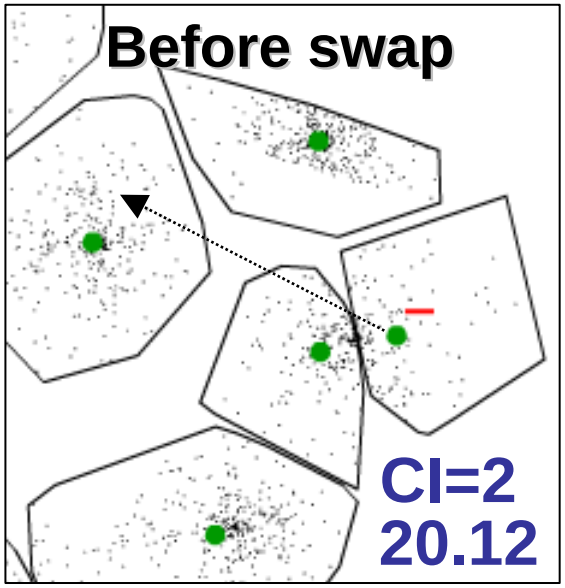
Effect of K-means iterations

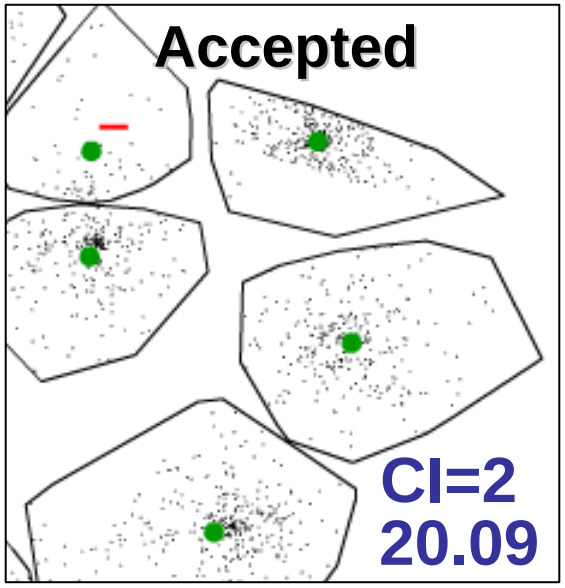


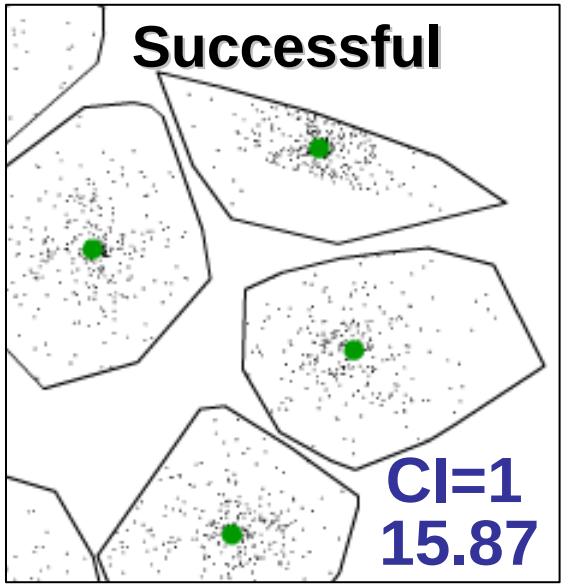
How many swaps?

Three types of swaps

- Trial swap
- Accepted swap ← MSE improves
- Successful swap ← CI improves







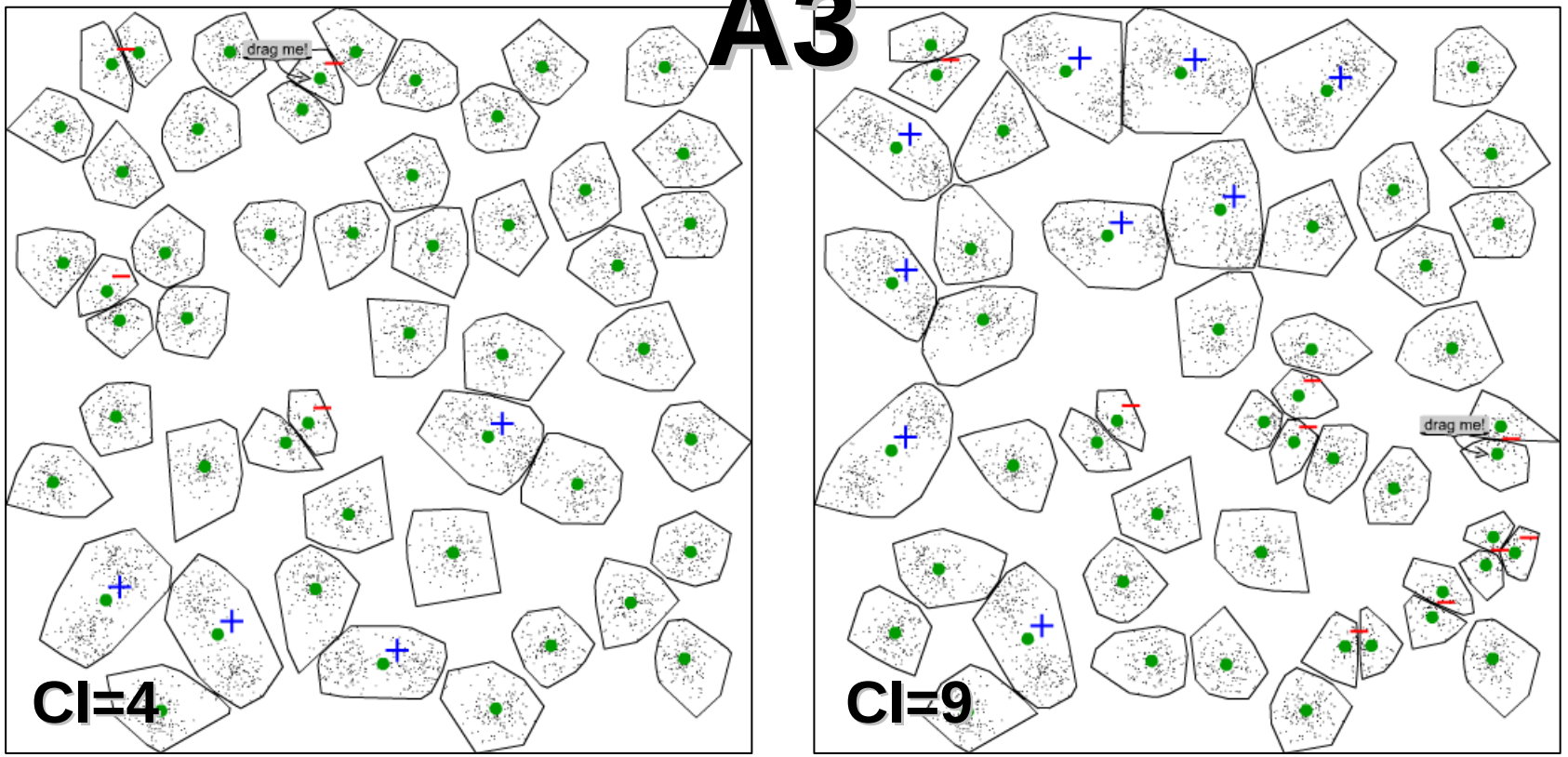
Accepted and successful swaps

Iteration	MSE	Time	
0	5312689297	0.006186	
1	2687180682	0.017132	CI=2
2	2275082565	0.027188	
3	1704970391	0.037289	CI=1
9	1700185570	0.097908	
16	1328087775	0.161352	CI=0
28	1327946264	0.265631	
58	1327923352	0.516983	
121	1327910949	1.027286	

Number of swaps needed

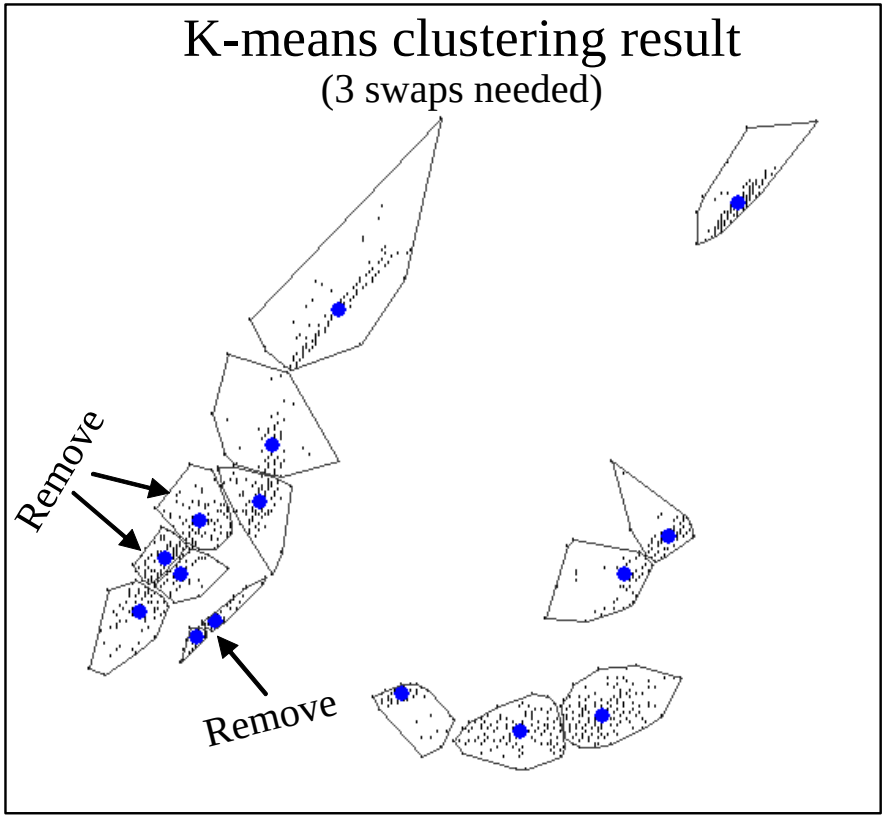
Example with 35 clusters

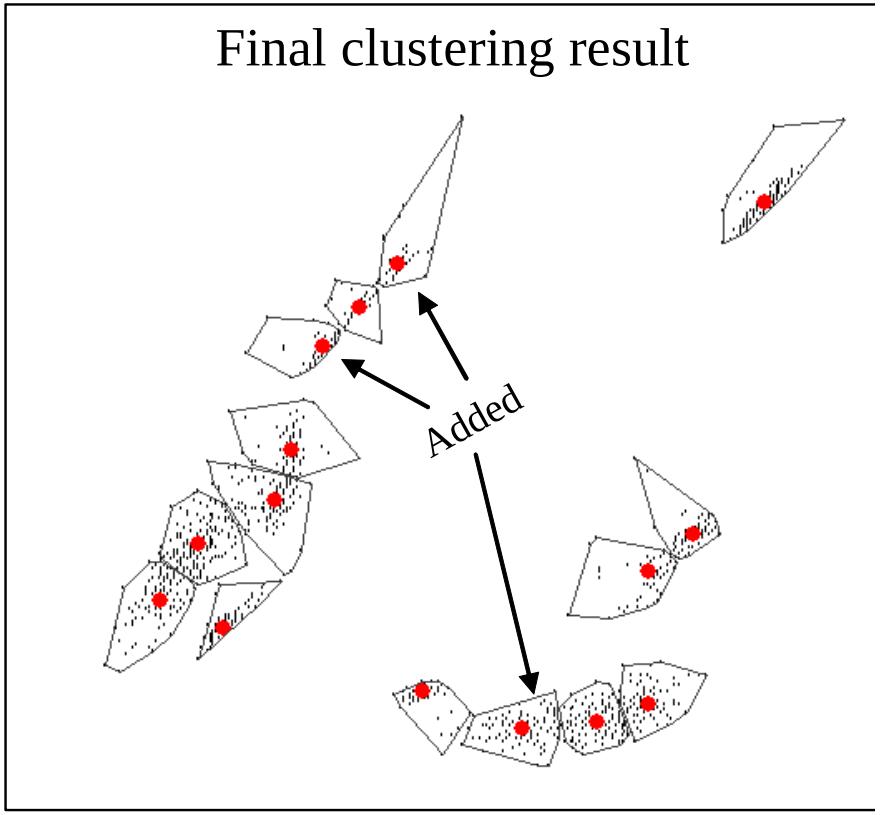
A3



Number of swaps needed

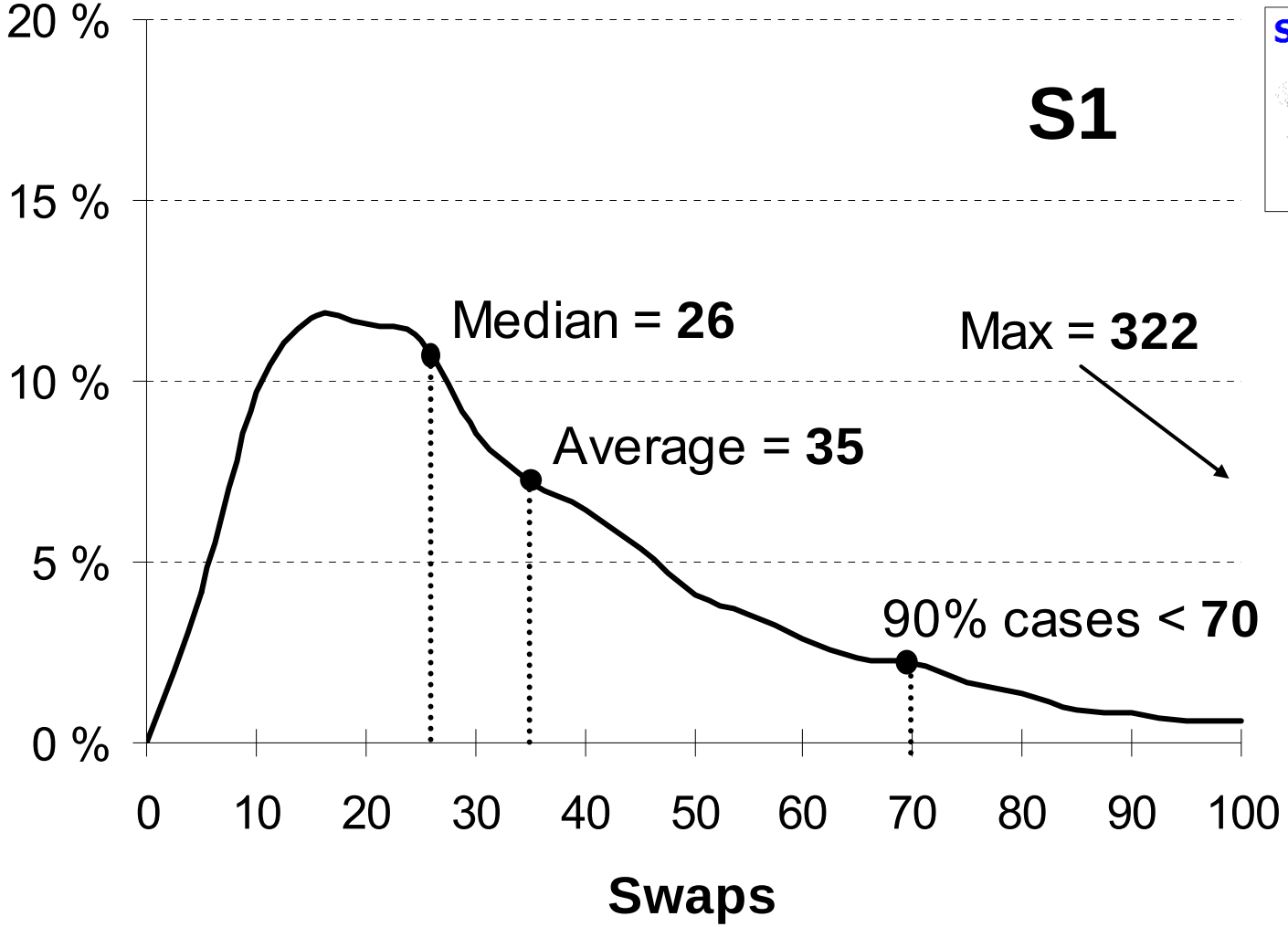
Example from image quantization

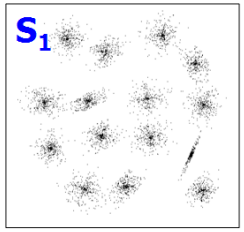




Statistical observations

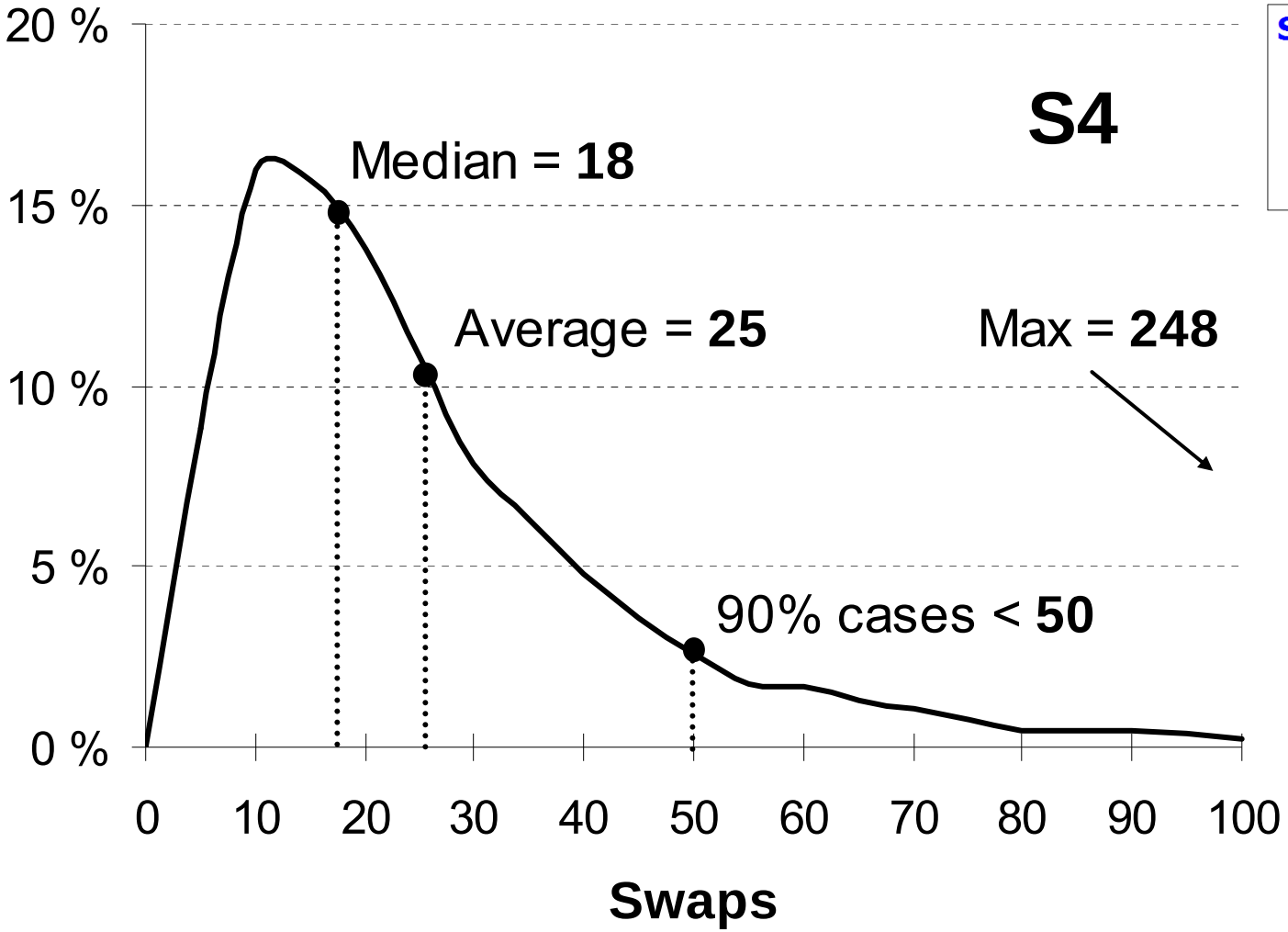
N=5000, k=15, d=2, N/k=333, $\alpha \approx 4.1$

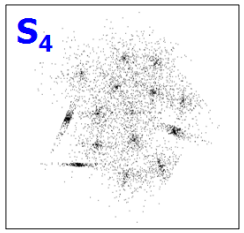




Statistical observations

N=5000, k=15, d=2, N/k=333, $\alpha \approx 4.8$





Theoretical estimation

Probability of good swap

- Select a proper prototype to remove:
  - There are k clusters in total: $p_{removal}=1/k$
- Select a proper new location:
  - There are N choices: $p_{add}=1/N$
  - Only k are significantly different: $p_{add}=1/k$
- Both happens same time:
  - k^2 significantly different swaps.
  - Probability of each different swap is $p_{swap}=1/k^2$
  - Open question: how many of these are good?

$$p = (\alpha/k)\cdot(\alpha/k) = O(\alpha/k)^2$$

Expected number of iterations

- Probability of **not** finding good swap:

$$q = \left(1 - \frac{\alpha^2}{k^2}\right)^T$$

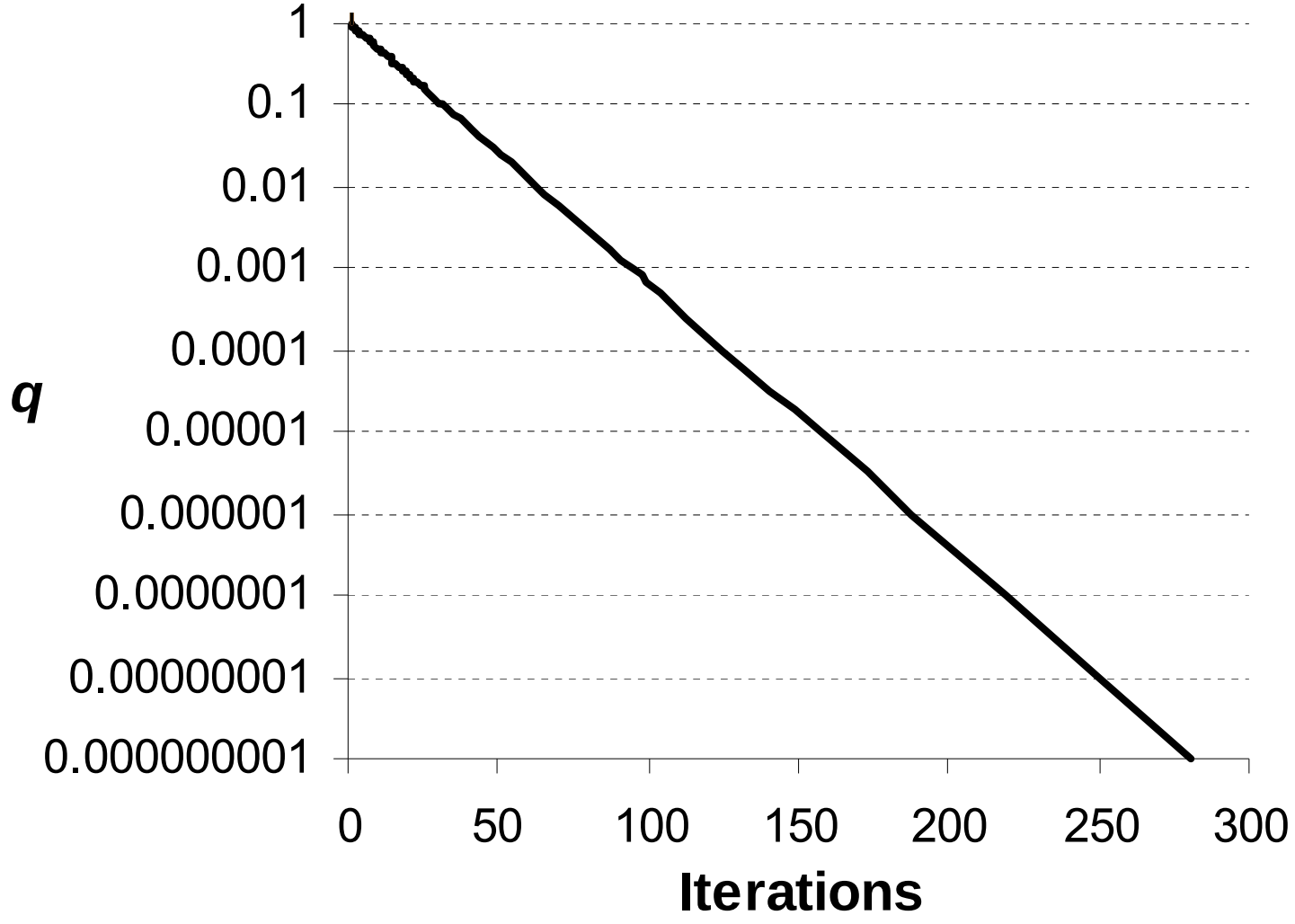
- Estimated number of iterations:

$$\log q = T \cdot \log\left(1 - \frac{\alpha^2}{k^2}\right)$$

$$\Leftrightarrow T = \frac{\log q}{\log\left(1 - \frac{\alpha^2}{k^2}\right)}$$

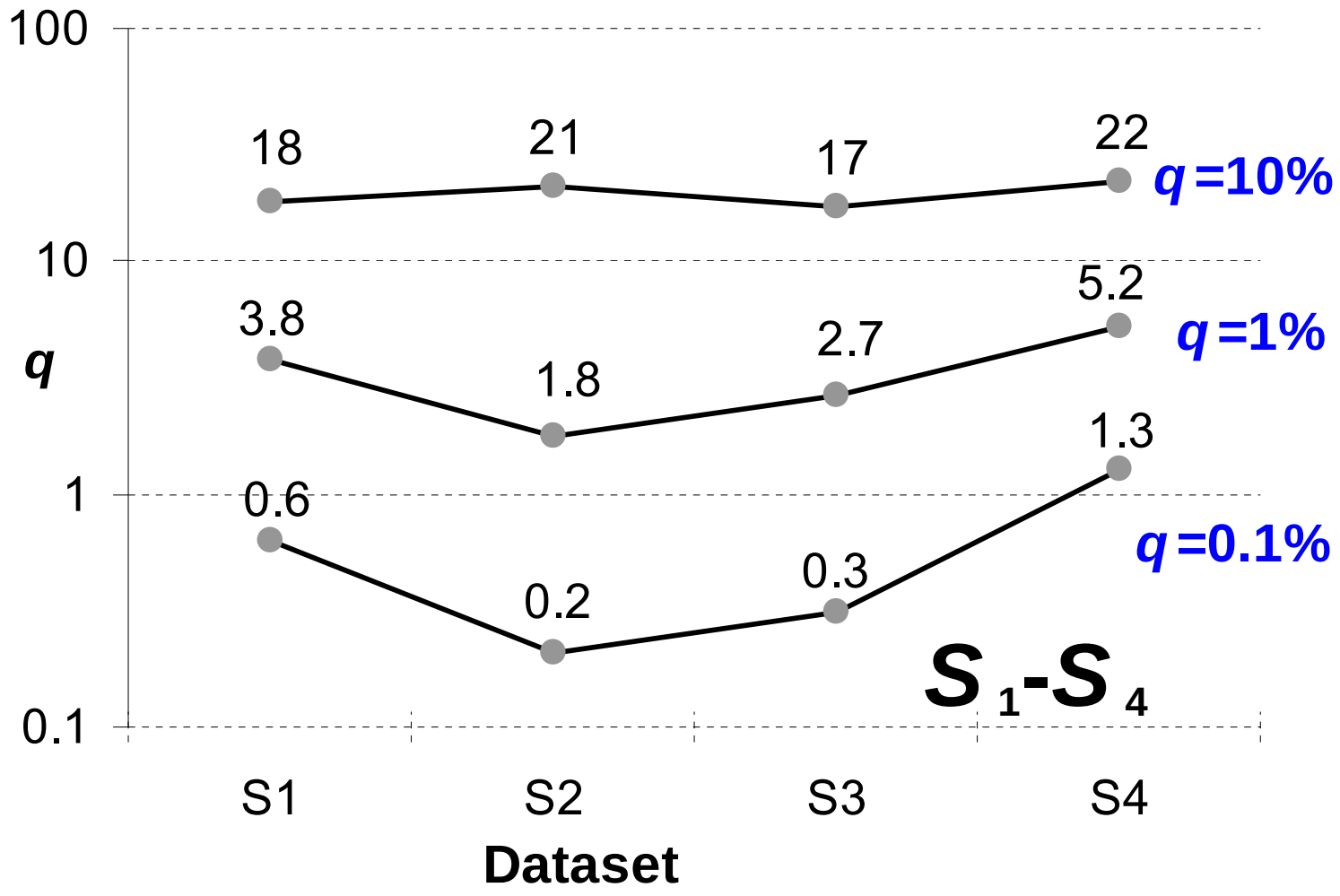
Probability of failure (*q*)

depending on *T*



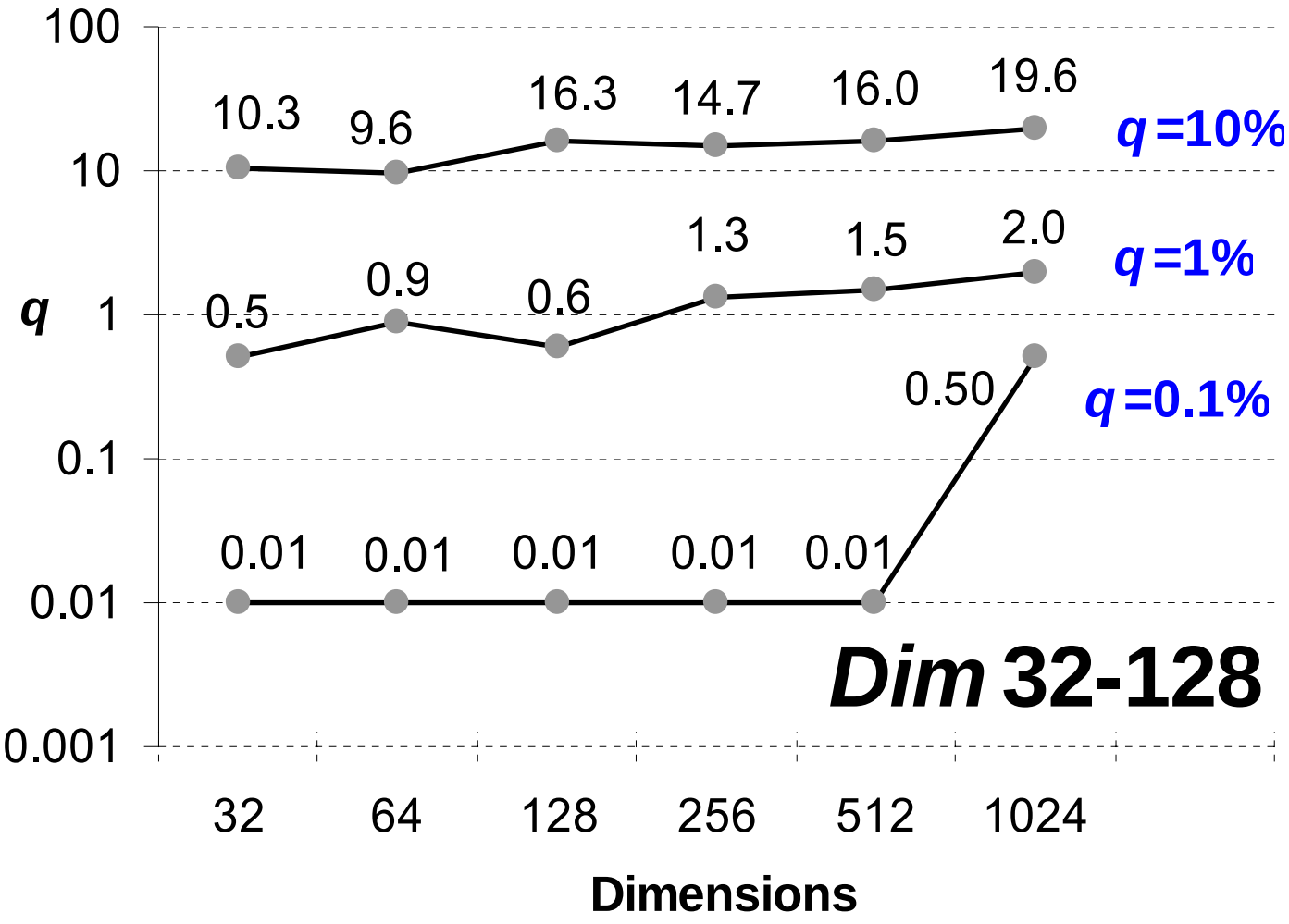
Observed probability (%) of fail

N=5000, k=15, d=2, N/k=333, $\alpha \approx 4.5$



Observed probability (%) of fail

N=1024, k=16, d=32-128, N/k=64, $\alpha \approx 1.1$



Bounds for the iterations

Upper limit:

$$T = \frac{\ln q}{\ln\left(1-\alpha^2/k^2\right)} \leq \frac{-\ln q}{\alpha^2/k^2} = -\ln q \cdot \frac{k^2}{\alpha^2}$$

Lower limit similarly; resulting in:

$$T = \Theta\left(-\ln q \cdot \frac{k^2}{\alpha^2}\right)$$

Multiple swaps (*w*)

Probability for performing less than *w* swaps:

$$q \leq \sum_{i=0}^{w-1} \binom{T}{i} \cdot \left(\frac{\alpha^2}{k^2}\right)^i \cdot \left(1-\frac{\alpha^2}{k^2}\right)^{T-i}$$

Expected number of iterations:

$$\hat{t} = \left(\sum_{i=1}^{w} \frac{1}{i}\right) \cdot \frac{k^2}{\alpha^2} = \mathrm{O}\left(\log w \cdot \frac{k^2}{\alpha^2}\right)$$

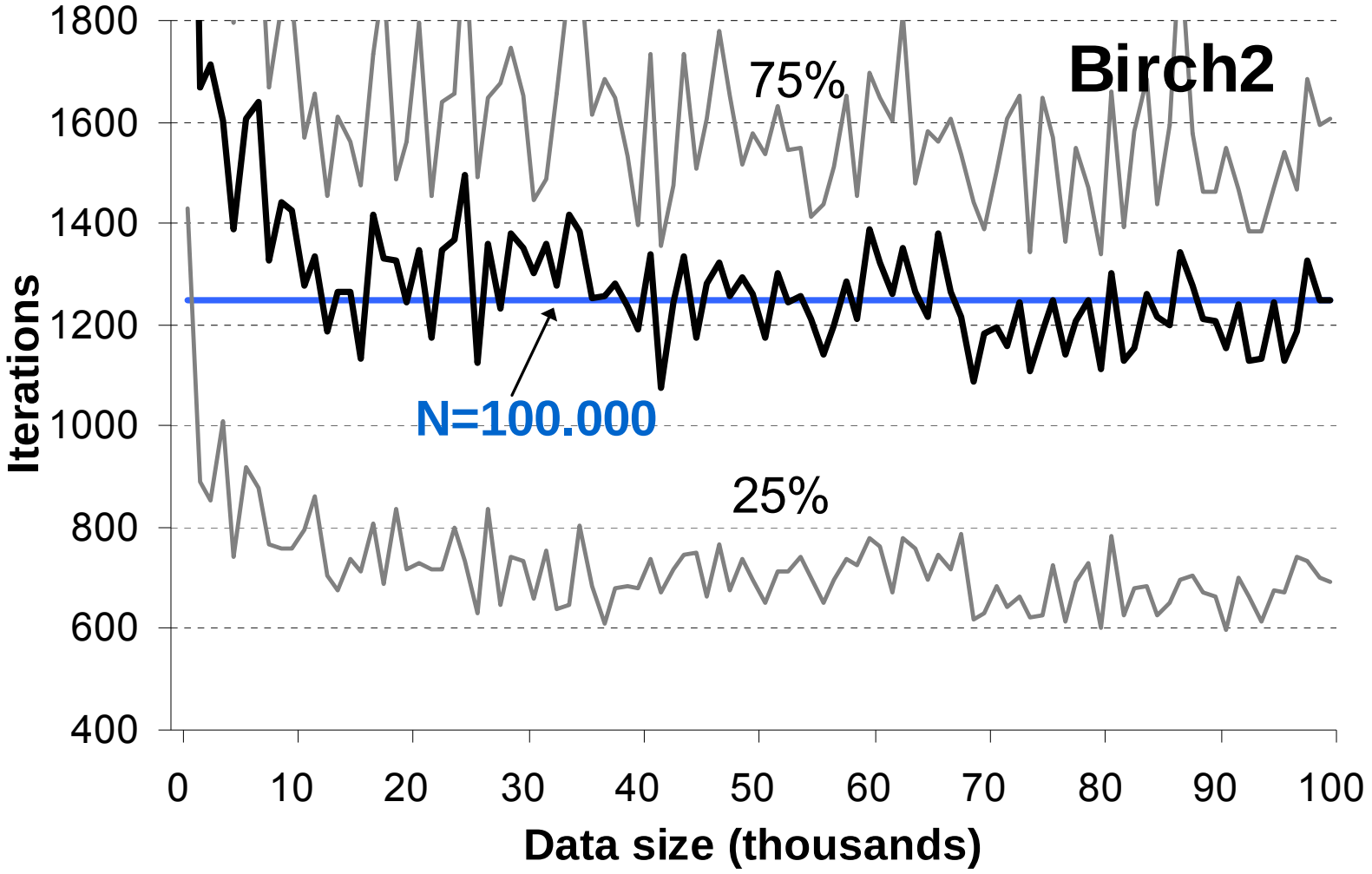
Expected time complexity

$$\hat{t}(N,k) \leq \log w \cdot \frac{k^2}{\alpha^2} \cdot \alpha N = \mathrm{O}\left(\frac{-\log(w) \cdot Nk^2}{\alpha} \right)$$

1. Linear dependency on N
2. Quadratic dependency on k
   (With large number of clusters, it can be too slow)
3. Logarithmic dependency on w
   (Close to constant)
4. Inverse dependency on α
   (Higher the dimensionality, faster the method)

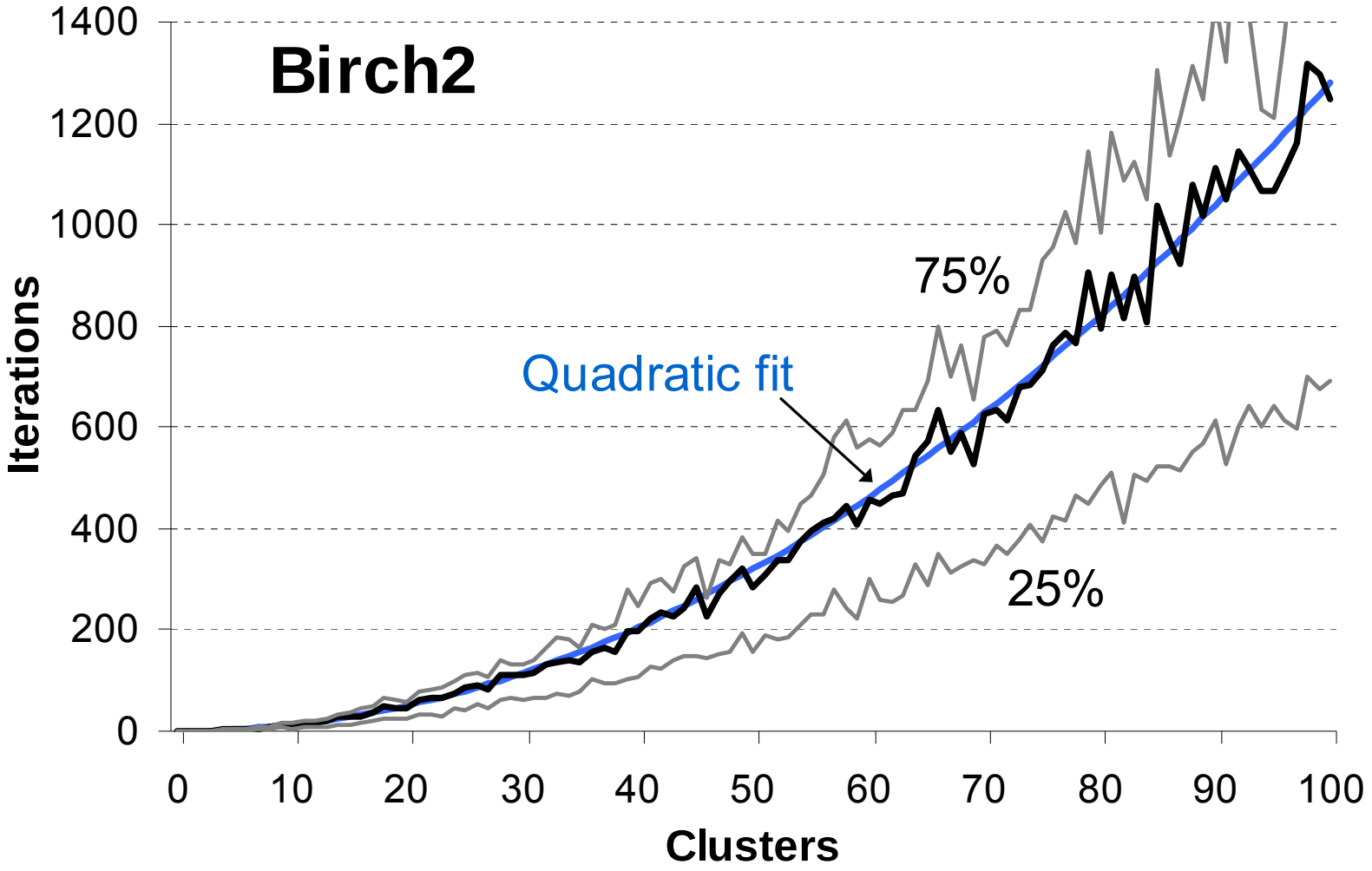
Linear dependency on N

N<100.000, k=100, d=2, N/k=1000, $\alpha \approx 3.1$



Quadratic dependency on k

N<100.000, k=100, d=2, N/k=1000, $\alpha \approx 3.1$



Logarithmic dependency on w

Data set	Iterations to reach CI-value						Factor	log w
	Total	0	1	2	3	4		
BIRCH$_1$	637	440	78	44	24	14	1.45	4.2
BIRCH$_2$	1246	761	191	84	51	33	1.64	4.4
S_1	33	26	7	2	1	1	1.34	1.5
S_2	25	19	4	1	1	1	1.30	1.4
S_3	22	17	4	1	1	1	1.34	1.4
S_4	25	19	3	1	1	1	1.27	1.4
Unbalance	122	58	29	23	13	1	2.09	2.0
Dim-32	73	52	13	6	3	2	1.42	1.9
Dim-64	83	59	13	7	4	2	1.41	1.9
Dim-128	91	56	20	9	4	3	1.61	1.9

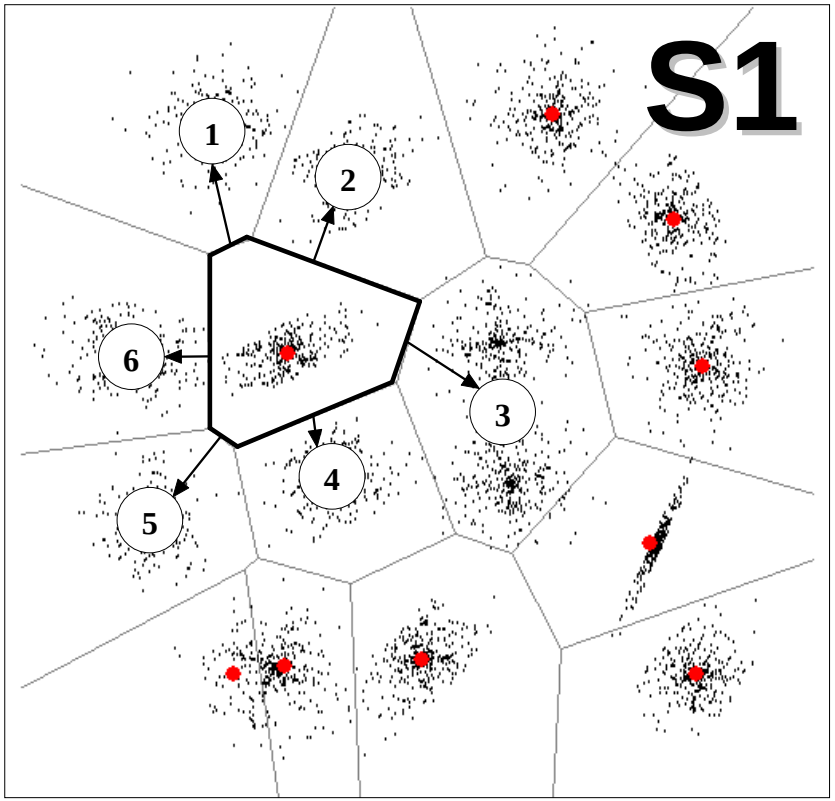
Theory vs. reality

Dataset	α	w	Last swap			All swaps		
			Exp	Real[2]	Real[10]	Exp	Real[2]	Real[10]
Bridge	5.4	90	2247	-	-	14590	-	-
House	8.3	69	951	-	-	5811	-	-
Miss America	17.1	116	224	-	-	1537	-	-
Europe	6.3	130	1651	-	-	11595	-	-
BIRCH$_1$	5.8	19	297	440	121	1263	637	197
BIRCH$_2$	3.1	21	1041	761	548	4571	1246	924
BIRCH$_3$	4.9	26	416	-	-	1958	-	-
S_1	4.1	2.8	13	26	18	20	33	23
S_2	4.5	2.7	11	19	9	16	25	12
S_3	4.4	2.7	12	17	7	17	22	10
S_4	4.8	2.7	10	19	9	14	25	11
Unbalance	2.3	4.0	12	58	54	24	122	110
Dim-32	1.1	3.7	212	52	52	399	73	76
Dim-64	1.1	3.7	212	59	64	399	83	88
Dim-128	1.0	3.8	256	56	65	493	91	98
KDD04-Bio	33.3	435	3607	-	-	31617	-	-

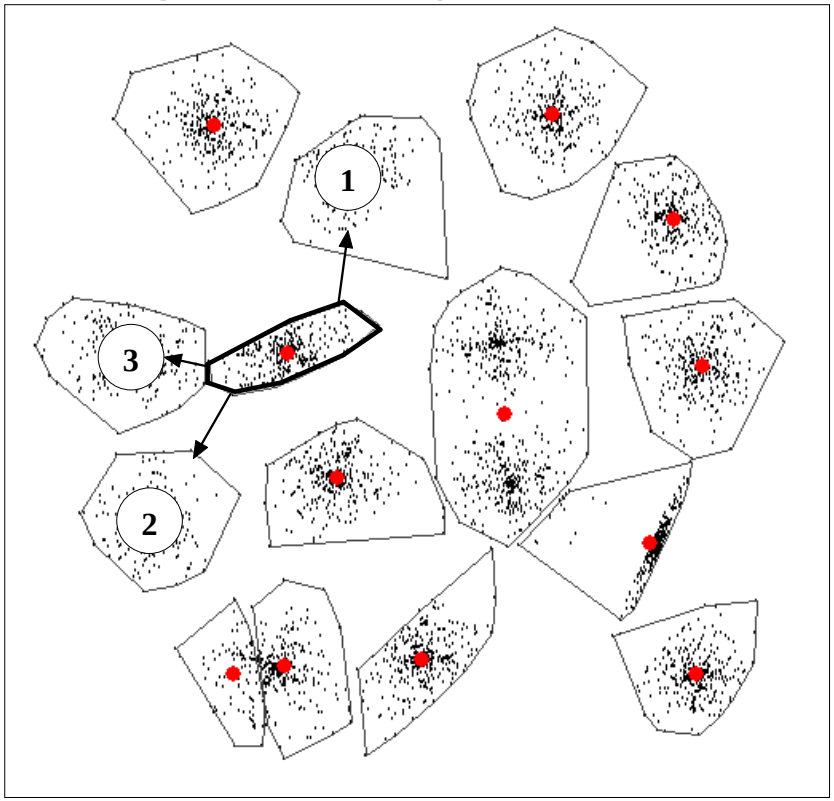
Neighborhood size

How much is α ?

Voronoi neighbors



Neighbors by distance



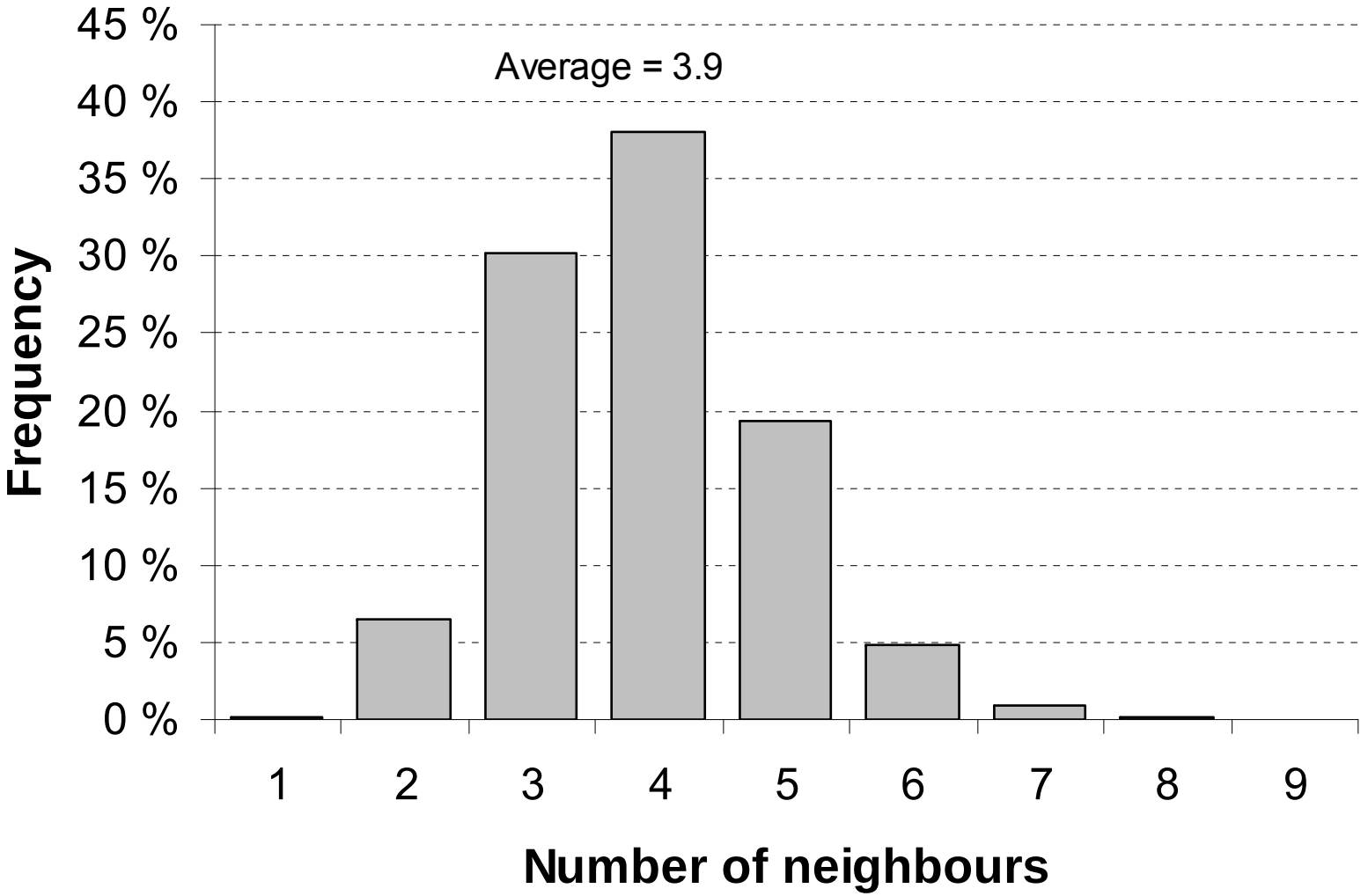
2-dim: $2\cdot(3k-6)/k = 6 - 12/k$

D-dim: $2\cdot k\lceil D/2\rceil/k = O(2\cdot k\lceil D/2\rceil-1)$

Upper limit: $\alpha \leq k$

Observed number of neighbors

Data set S_2



Estimate α

- Five iterations of random swap clustering
- Each pair of prototypes A and B:
    1. Calculate the half point HP = (A+B)/2
    2. Find the nearest prototype C for HP
    3. If C=A or C=B they are potential neighbors.
- Analyze potential neighbors:
    1. Calculate all vector distances across A and B
    2. Select the nearest pair (a,b)
    3. If d(a,b) < min(d(a,C(a), d(b,C(b)) then Accept
- α = Number of pairs found / k

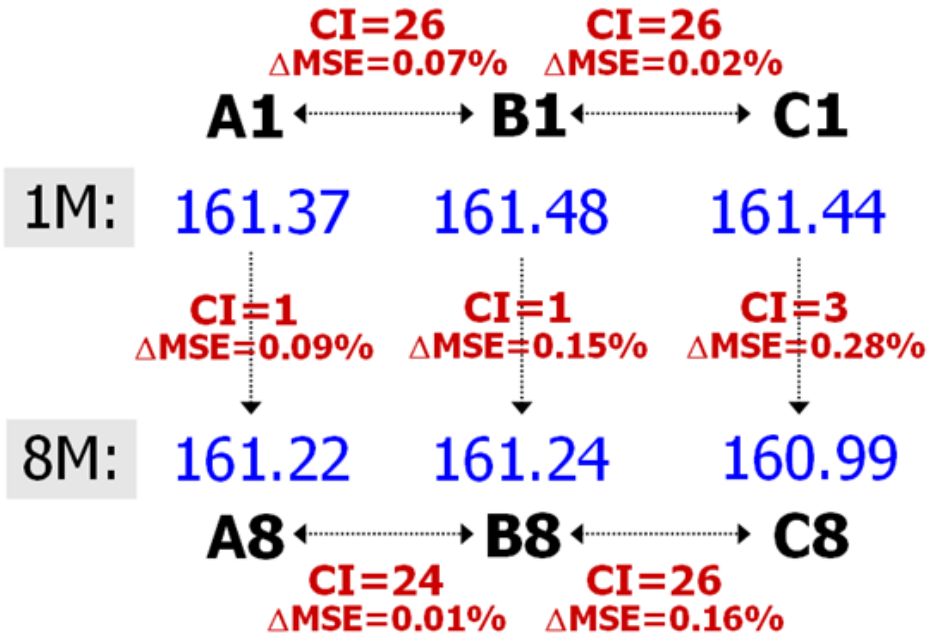
Observed values of α

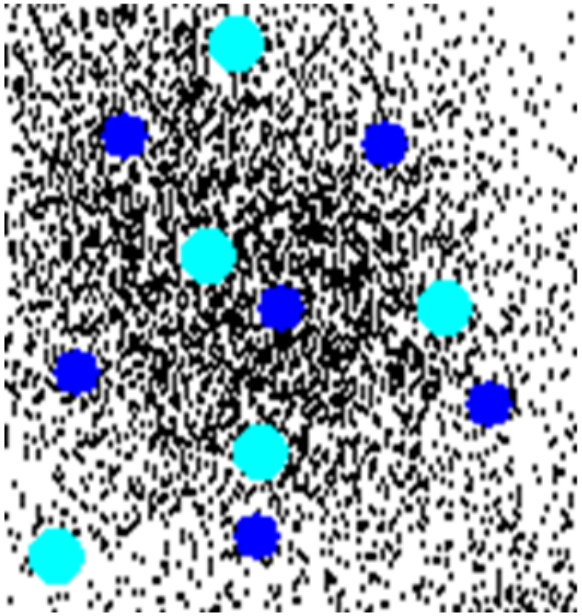
Dataset	α from data	α from clustering			Estimated (T)		
		Initial T=0	Early T=5	Final T=5000	q=10%	q=1%	q=0.1%
Bridge	69.8	8.7	5.4	4.6	33595	67910	100785
House	15.4	6.7	8.3	8.2	13381	26761	40142
Miss America	346	34.2	17.1	11.9	3593	7078	10617
Europe	(5.0)	4.8	6.3	6.3	26699	53398	80098
BIRCH$_1$	5.0	4.5	5.8	5.6	2908	5815	8723
BIRCH$_2$	(4.7)	3.1	3.1	2.9	10524	21048	31572
BIRCH$_3$	(4.9)	4.1	4.9	5.0	4508	9016	13523
S_1	4.8	3.7	4.1	4.2	46	92	137
S_2	4.9	3.7	4.5	4.7	37	73	110
S_3	4.9	3.9	4.4	4.3	38	77	115
S_4	4.9	3.9	4.8	5.0	32	64	97
Unbalance	3.4	2.3	2.3	2.0	56	111	167
Dim-32	26.8	1.5	1.1	1.0	920	1839	2759
Dim-64	37.1	1.9	1.1	1.0	920	1839	2759
Dim-128	47.3	1.4	1.0	1.0	1135	2271	3406
KDD04-Bio	---	286.2	33.3	30.4	72800	145600	218401

Optimality

Multiple optima (=plateaus)

- Very similar result (<0.3% diff. in MSE)
- CI-values significantly high (≈9%)
- Finds one of the near-optimal solutions

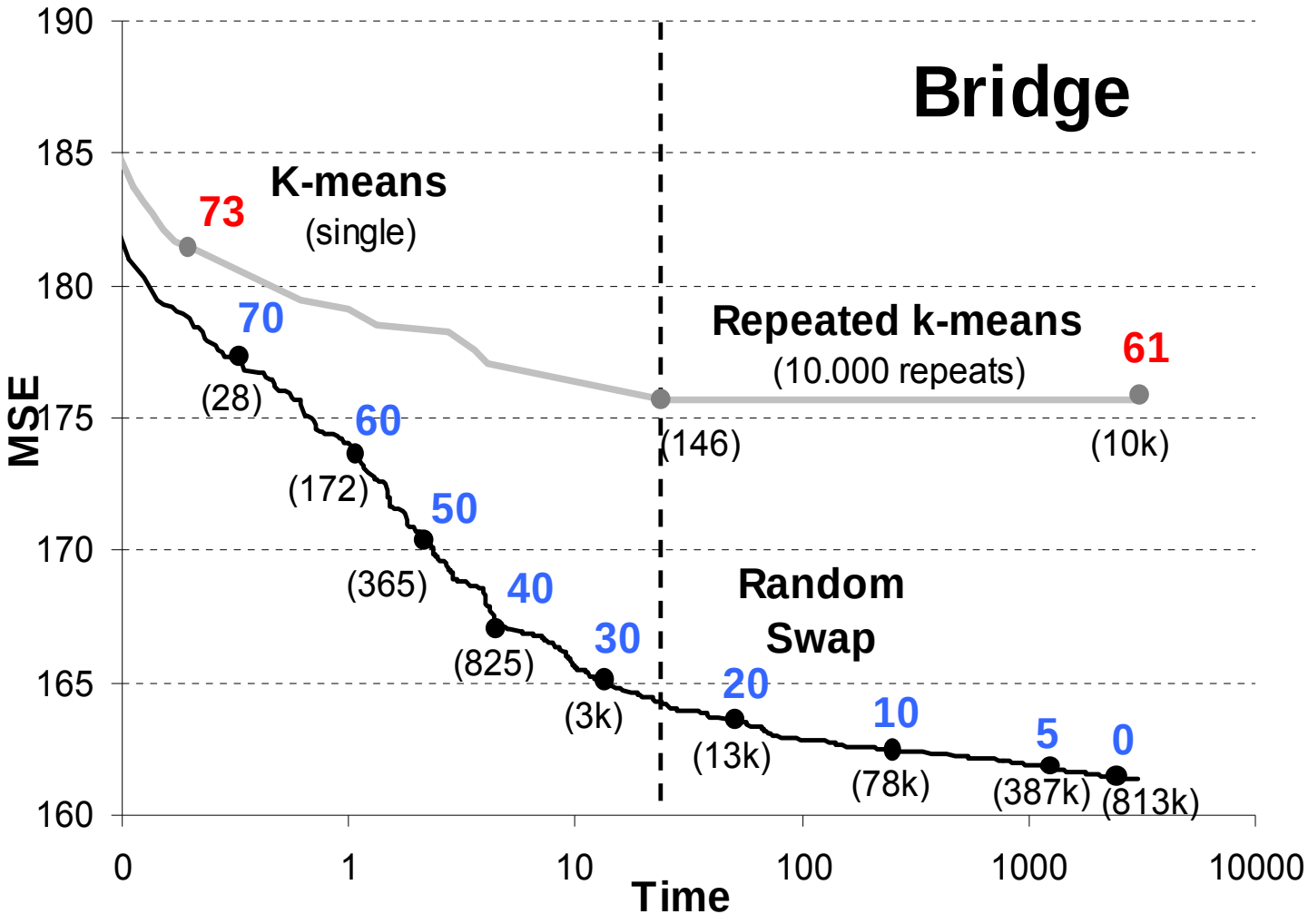




Experiments

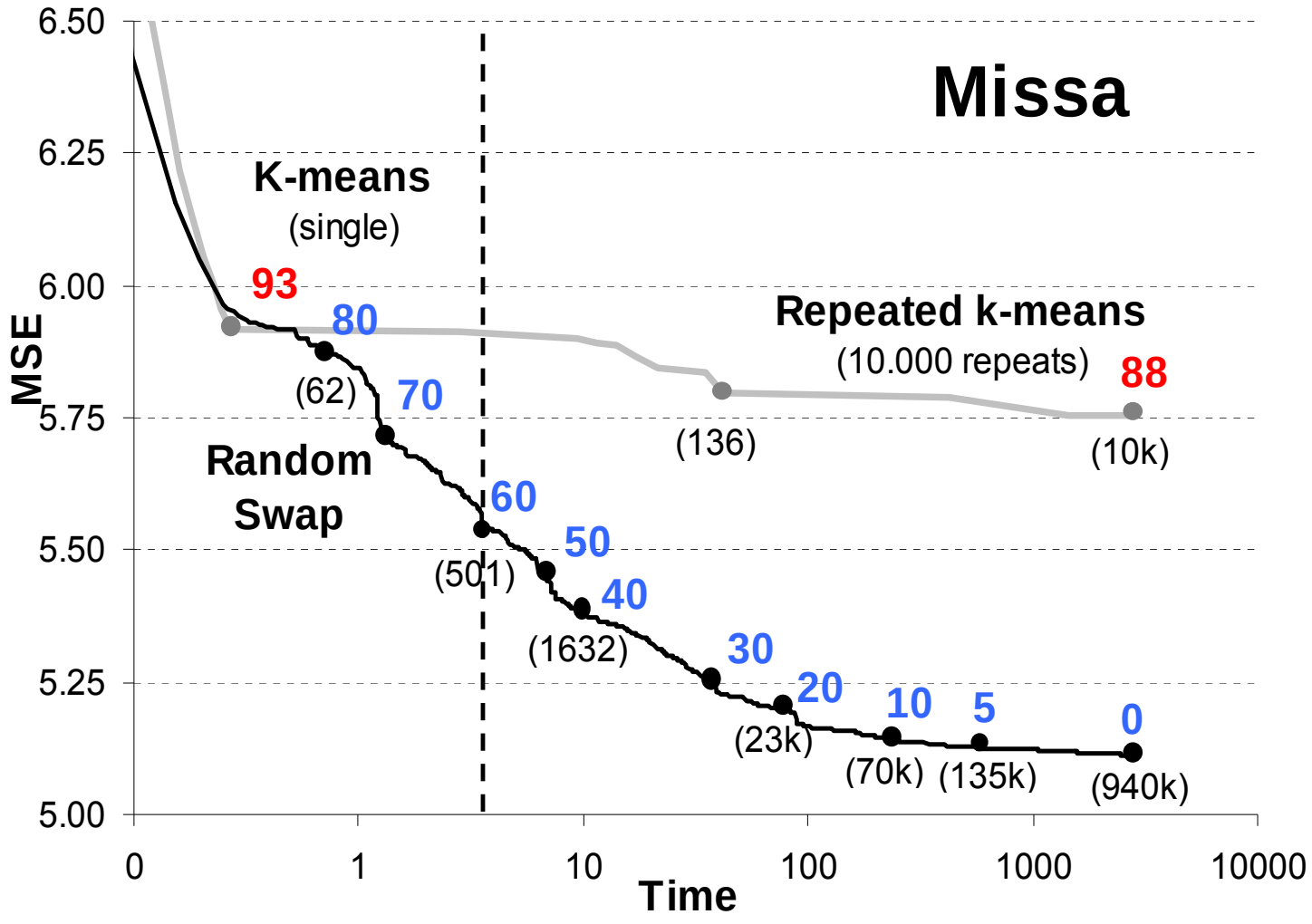
Time-versus-distortion

N=4096, k=256, d=16, N/k=16, $\alpha \approx 5.4$



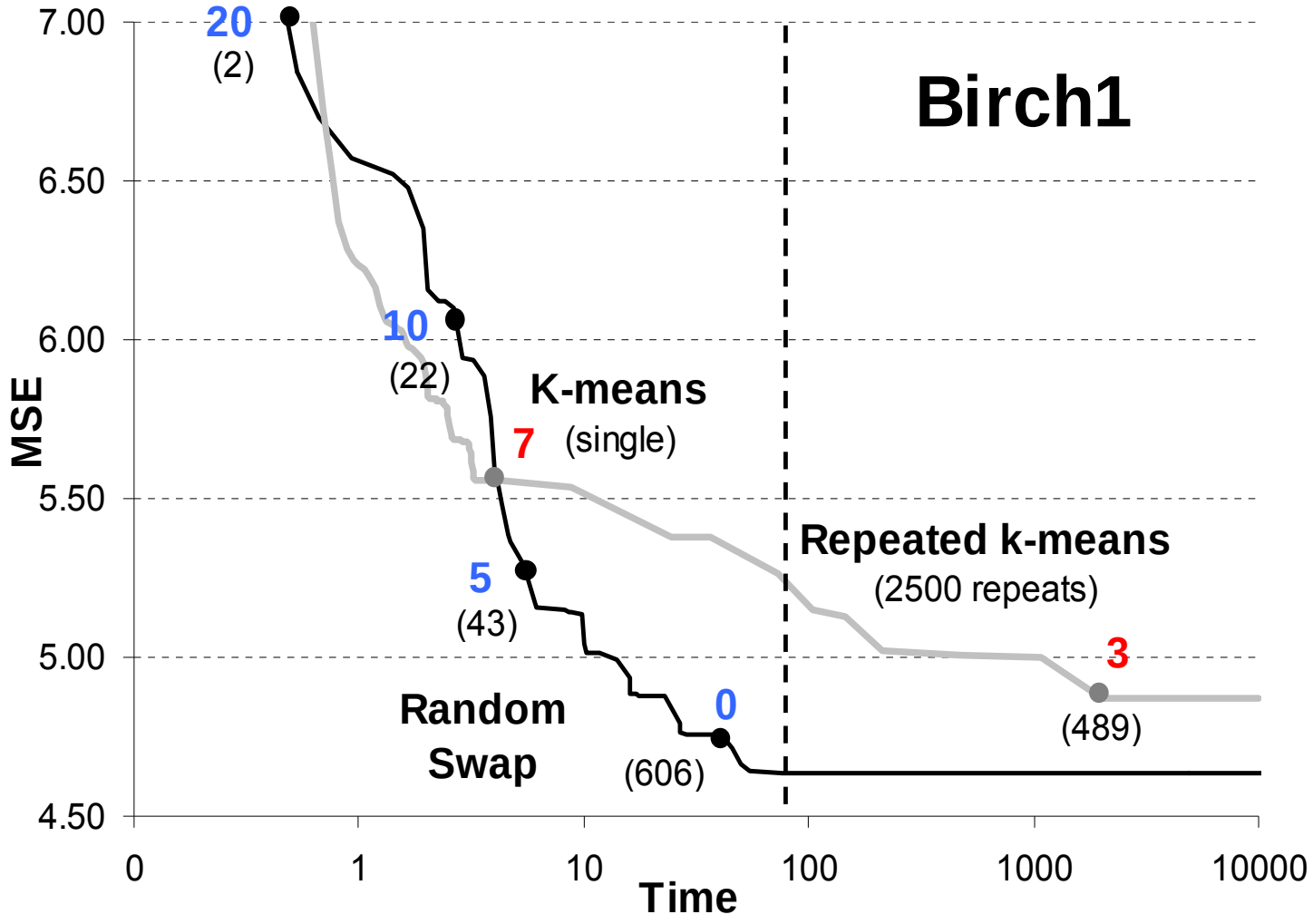
Time-versus-distortion

N=6480, k=256, d=16, N/k=25, $\alpha \approx 17.1$



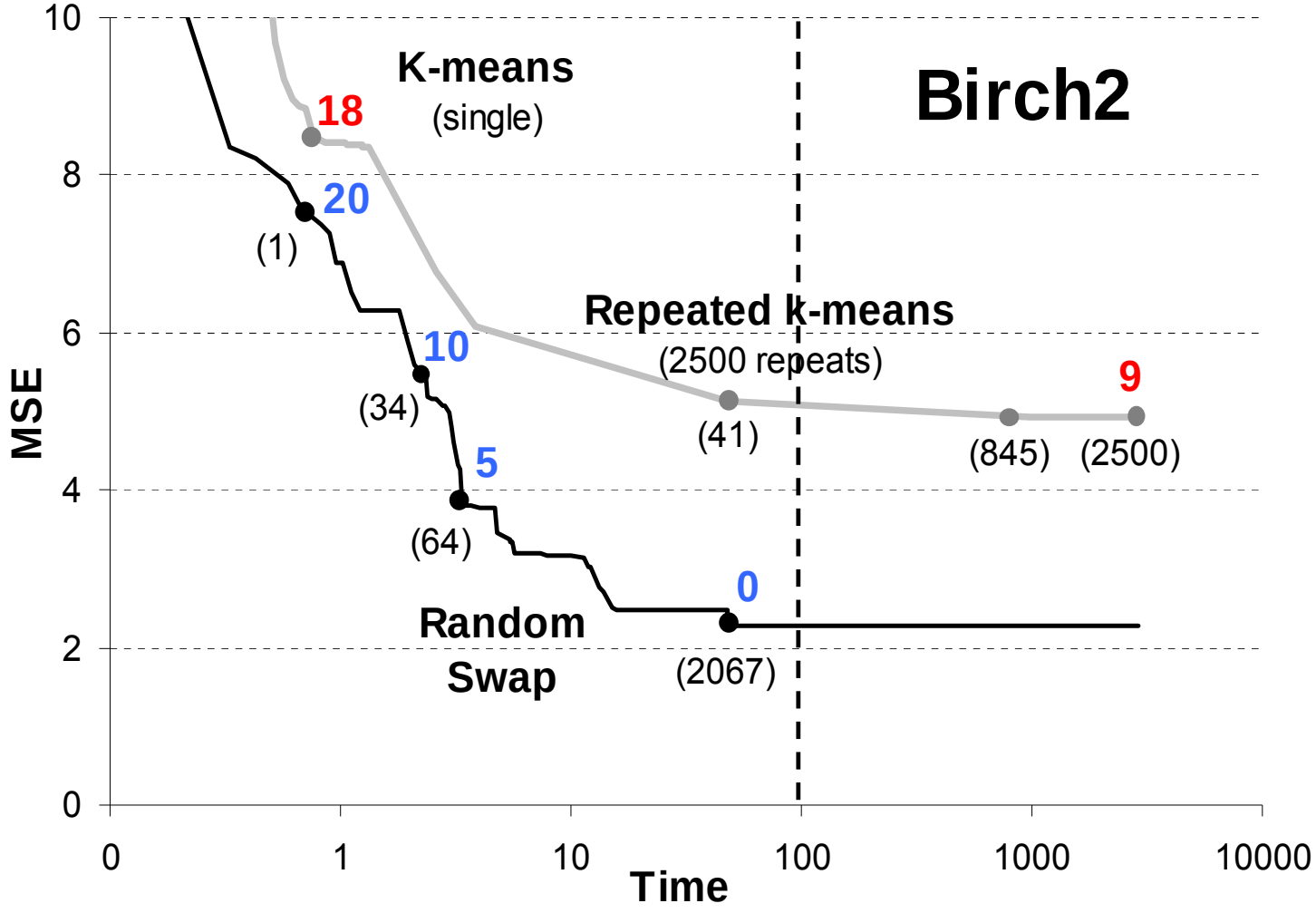
Time-versus-distortion

N=100.000, k=100, d=2, N/k=1000, $\alpha \approx 5.8$



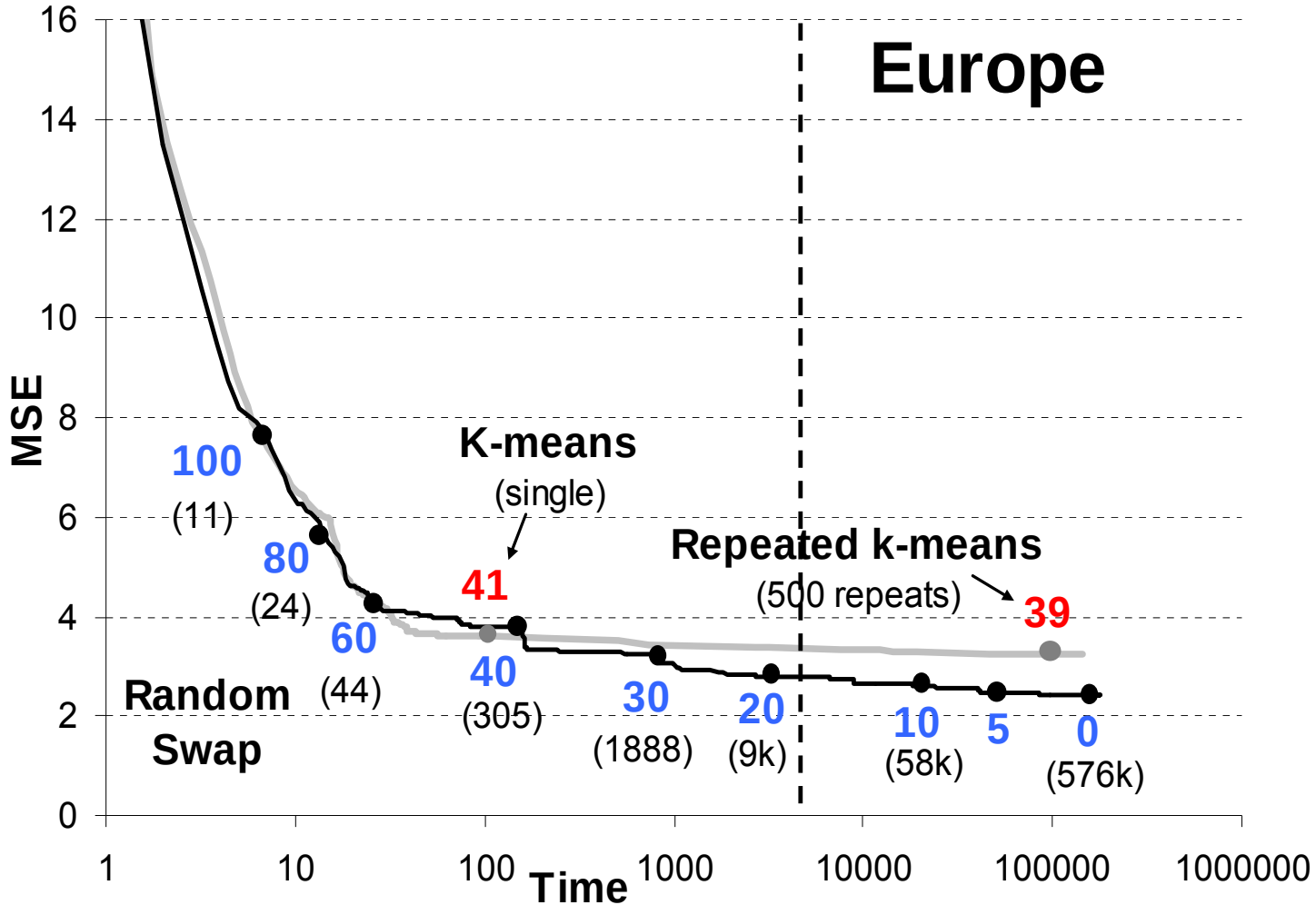
Time-versus-distortion

N=100.000, k=100, d=2, N/k=1000, $\alpha \approx 3.1$



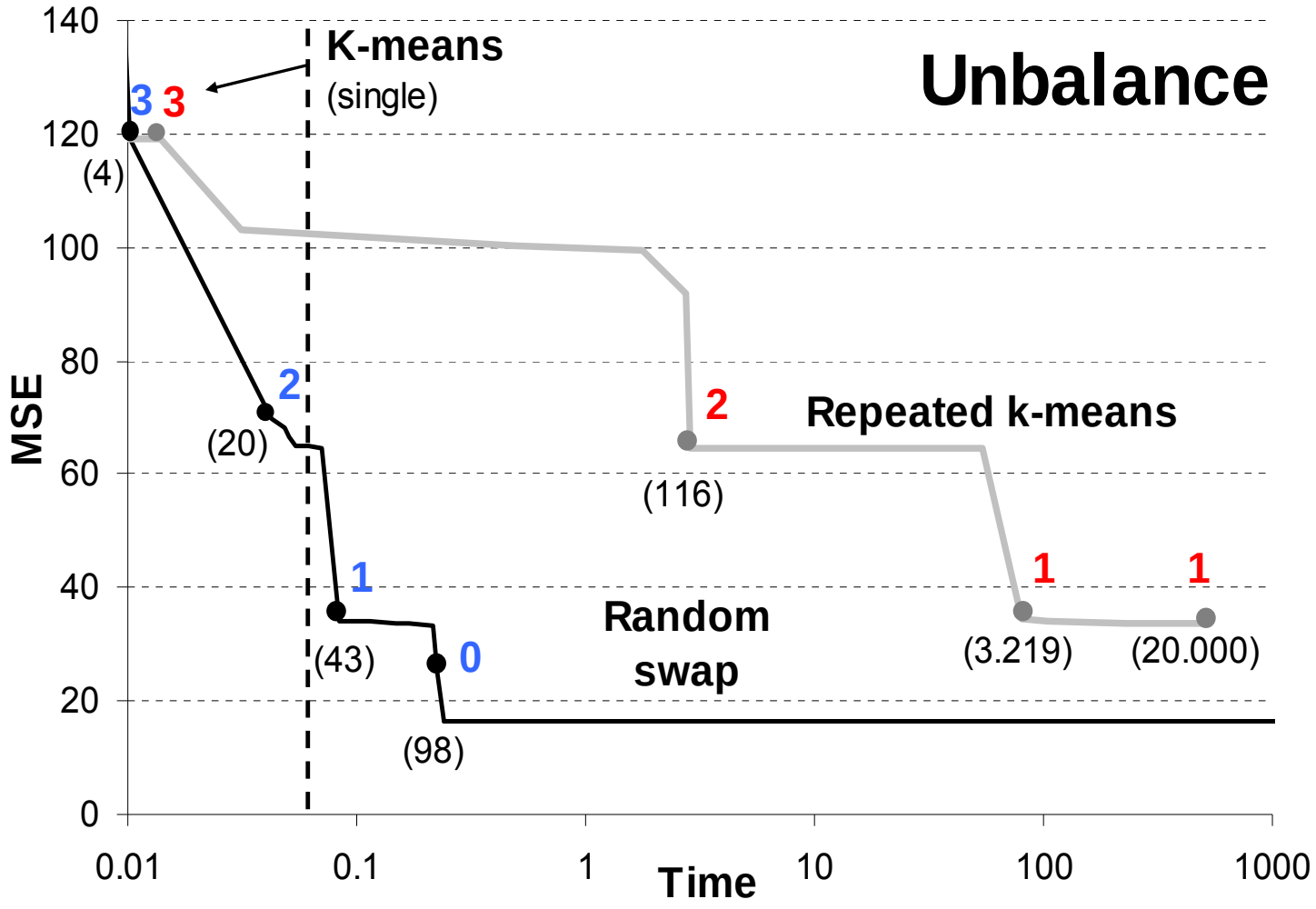
Time-versus-distortion

N=169.673, k=256, d=2, N/k=663, $\alpha \approx 6.3$

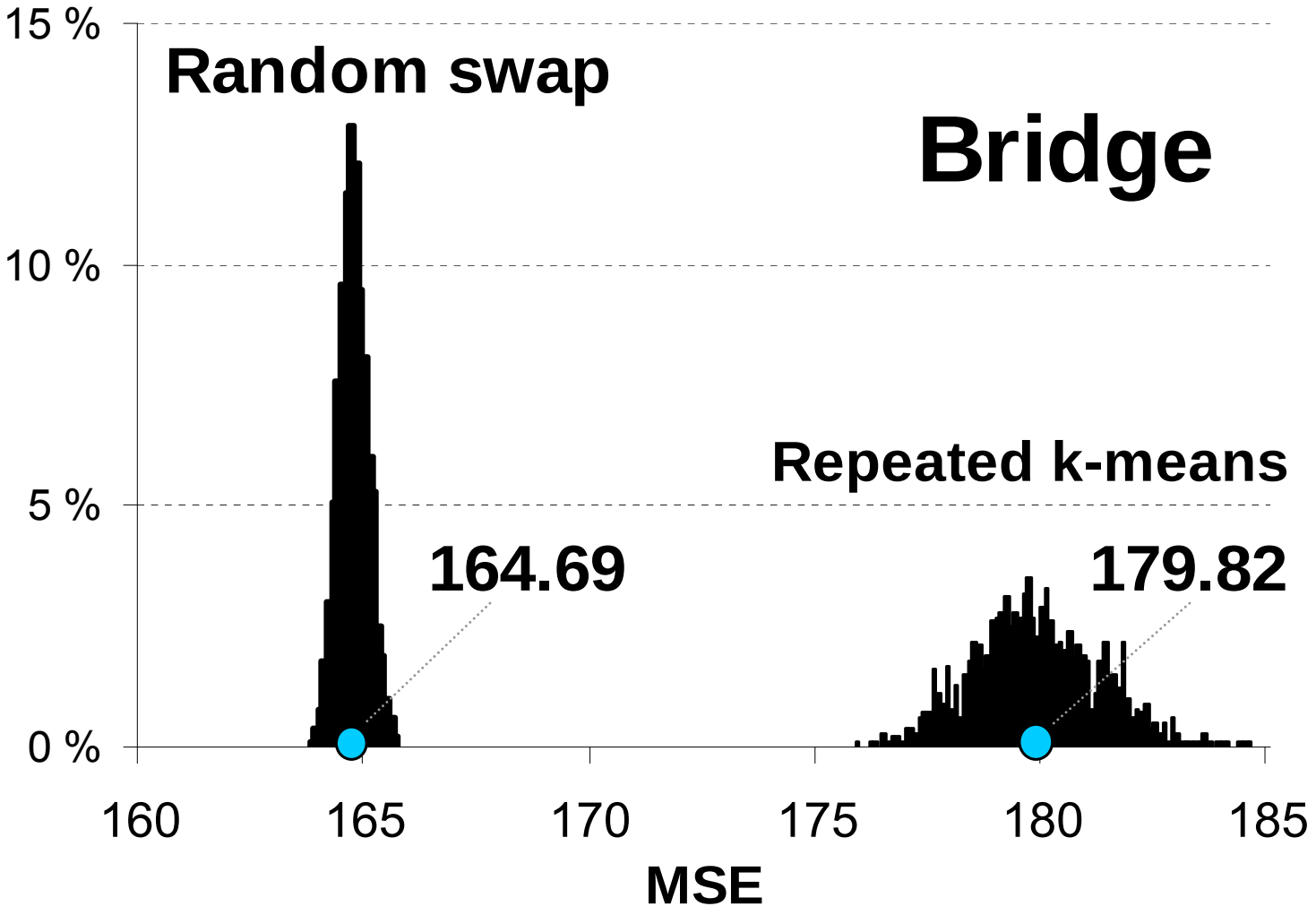


Time-versus-distortion

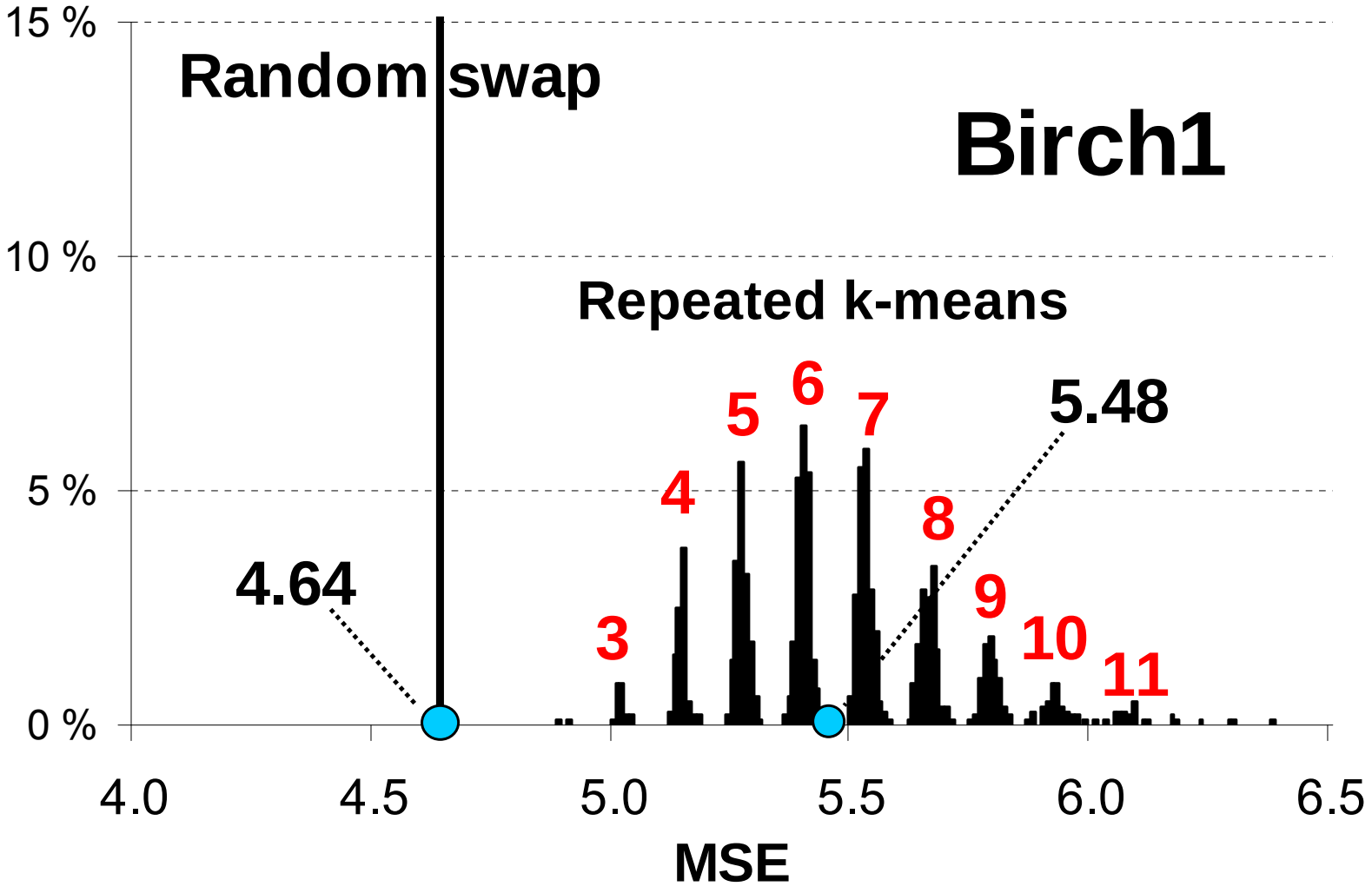
N=6500, k=8, d=2, N/k=821, $\alpha \approx 2.3$



Variation of results



Variation of results



Comparison of algorithms

- k-means (KM)
- k-means++
- repeated k-means (RKM)
- x-means
- agglomerative clustering (AC)
- random swap (RS)
- global k-means
- genetic algorithm

D. Arthur and S. Vassilvitskii, "K-means++: the advantages of careful seeding", *ACM-SIAM Symp. on Discrete Algorithms (SODA'07)*, New Orleans, LA, 1027-1035, January, 2007.

D. Pelleg, and A. Moore, "X-means: Extending k-means with efficient estimation of the number of clusters", *Int. Conf. on Machine Learning, (ICML'00)*, Stanford, CA, USA, June 2000.

P. Fränti, T. Kaukoranta, D.-F. Shen and K.-S. Chang, "Fast and memory efficient implementation of the exact PNN", *IEEE Trans. on Image Processing*, 9 (5), 773-777, May 2000.

A. Likas, N. Vlassis and J.J. Verbeek, "The global k-means clustering algorithm", *Pattern Recognition* 36, 451-461, 2003.

P. Fränti, "Genetic algorithm with deterministic crossover for vector quantization", *Pat. Rec. Let.*, 21 (1), 61-68, January 2000.

Processing time

Data set	Processing time (s)								
	KM	RKM	KM++	XM	AC	RS_{5000}	RS_x	GKM	GA
BIRCH$_1$	3.7	374	3.2	107	141	276	420	---	297
BIRCH$_2$	1.1	114	1.0	12	144	92	537	---	256
S_1	<1	1.1	<1	<1	<1	5	<1	74	3
S_2	<1	1.4	<1	<1	<1	6	<1	95	2
S_3	<1	1.8	<1	<1	<1	6	<1	109	3
S_4	<1	2.5	<1	<1	<1	7	<1	117	3
Unbalance	<1	2.6	<1	<1	<1	12	<1	152	2
Dim-32	<1	<1	<1	<1	<1	3	1.5	6	1
Dim-64	<1	<1	<1	<1	<1	5	3	11	2
Dim-128	<1	<1	<1	<1	<1	8	5	19	3

Clustering quality

Data set	Centroid Index (CI)								
	KM	RKM	KM++	XM	AC	RS_{5000}	RS_x	GKM	GA
BIRCH$_1$	6.6	2.9	4.0	1.6	0	0	0	---	0
BIRCH$_2$	16.9	10.5	7.6	1.7	0	0	0	---	0
S_1	1.8	0.0	1.1	0.3	0	0	0	0	0
S_2	1.5	0.0	1.0	0.2	0	0	0	0	0
S_3	1.1	0.0	0.9	0.3	0	0	0	0	0
S_4	0.8	0.0	0.9	0.4	1	0	0	0	0
Unbalance	3.9	2.0	0.5	1.7	0	0	0	0	0
Dim-32	3.8	1.1	0.5	2.7	0	0	0	0	0
Dim-64	3.7	1.1	0.0	4.0	0	0	0	0	0
Dim-128	4.0	1.4	0.0	4.2	0	0	0	0	0

Conclusions

What we learned?

1. Random swap is efficient algorithm
2. It does not converge to sub-optimal result
3. Expected processing has dependency:
   - Linear O(*N*) dependency on the size of data
   - Quadratic O(*k*2) on the number of clusters
   - Inverse O(1/α) on the neighborhood size
   - Logarithmic *O(log w*) on the number of swaps

References

- P. Fränti, "Efficiency of random swap clustering", *Journal of Big Data*, 5:13, 1-29, 2018.
- P. Fränti and J. Kivijärvi, "Randomised local search algorithm for the clustering problem", *Pattern Analysis and Applications*, 3 (4), 358-369, 2000.
- P. Fränti, J. Kivijärvi and O. Nevalainen, "Tabu search algorithm for codebook generation in VQ", *Pattern Recognition*, 31 (8), 1139-1148, August 1998.
- P. Fränti, O. Virmajoki and V. Hautamäki, “Efficiency of random swap based clustering", *IAPR Int. Conf. on Pattern Recognition (ICPR’08)*, Tampa, FL, Dec 2008.
- **Pseudo code:** http://cs.uef.fi/pages/franti/research/rs.txt

Supporting material

Implementations available:
(C, Matlab, Java, Javascript, R and Python)

http://www.uef.fi/web/machine-learning/software

Interactive animation:

http://cs.uef.fi/sipu/animator/

Clusterator:

http://cs.uef.fi/sipu/clusterator