Network Clustering

Part 2

Alexander Ponomarenko

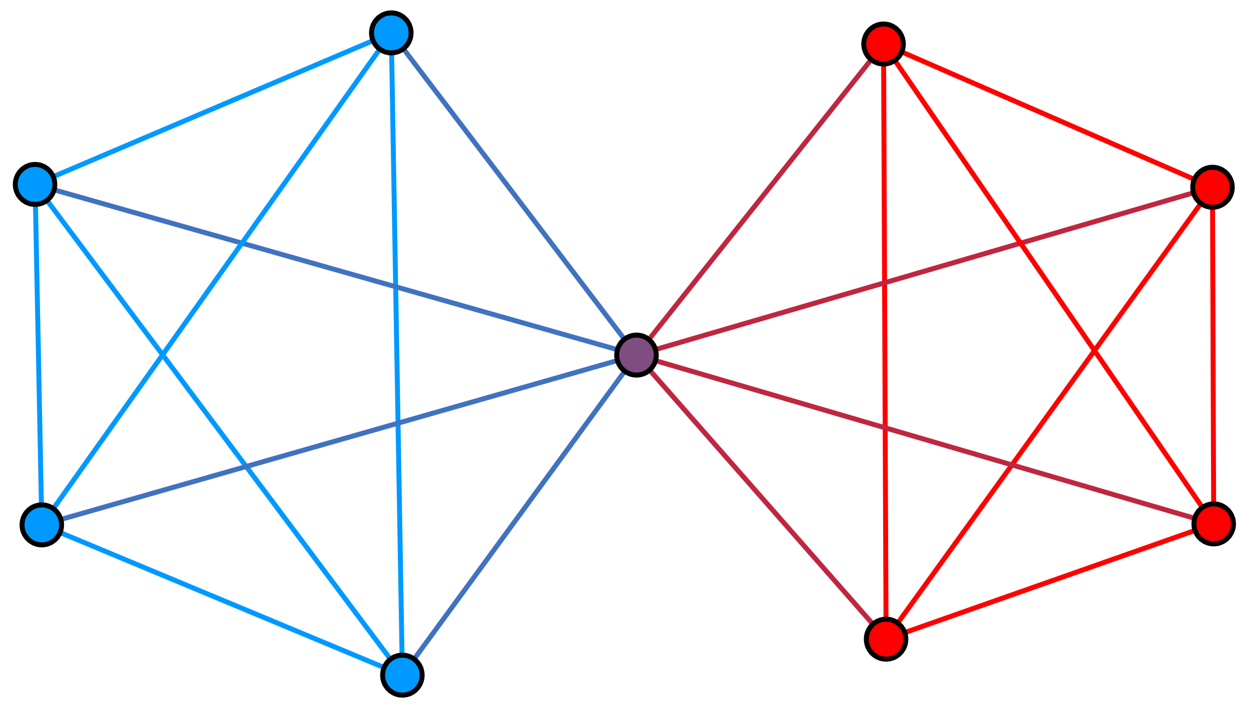
Overlapping Community Detection

Link partitioning approach

Let $D = (d_{ij})_{i=1,j=1}^{m,m}$ is a distance matrix defined on the set of edges

We calculate the belonging factor of node i to cluster c as

$$a_{ic} = \frac{\sum_{(i,j)\in E} x_{jc}}{|N_G(i)|}$$



Distance Based

$$b_{ij} = \sqrt{\sum_{k \neq i,j} \left(A_{ik} - A_{jk}\right)^2}\,.$$

Burt's distance

$$\zeta_{ij} = 1 - \underbrace{\frac{|c_i \cap c_j|}{|c_i \cup c_j|}}_{s_{ij}} \in [0,1]\,.$$

Jaccard Distance

$$o_{ij} = 1 - \frac{|c_i \cap c_j|}{\sqrt{|c_i| \times |c_j|}} \in [0,1]\,.$$

Otsuka-Ochiai Distance

Shortest path distance — well known

Partitioning around medoids

Let $D = (d_{ij})_{i=1,j=1}^{m,m}$ is a distance matrix defined on the set of edges

Centers of the clusters is a set of k vertices of line graph $L(G)$

$$S = \{s_1, s_2, ..., s_k\}$$

$$\sum_{c=1}^{k} d_{jc} x_{jc}, j \in E \to \min, \tag{3}$$

$$x_{jc} = \begin{cases} 1, & \text{if } d_{jc} \leq d_{js},\ s \in S, \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

[Kaufman, L., & Rousseeuw, P. (1987). *Clustering by means of medoids*. North-Holland.]

Partitioning around medoids

Let $D = (d_{ij})_{i=1,j=1}^{m,m}$ is a distance matrix defined on the set of edges

Centers of the clusters is a set of k vertices of line graph $L(G)$

$$S = \{s_1, s_2, ..., s_k\}$$

$$\sum_{c=1}^{k} d_{jc} x_{jc}, j \in E \to \min, \qquad (3)$$

$$x_{jc} = \begin{cases} 1, & \text{if } d_{jc} \leq d_{js},\ s \in S, \\ 0, & \text{otherwise} \end{cases} \qquad (4)$$

p-median problem also known as *facility location problem*

We solve *p*-median problem exactly with LP_solve by using efficient model of Goldengorin

[AlBdaiwi, B. F., Ghosh, D., & Goldengorin, B. (2011). Data aggregation for p-median problems. *Journal of Combinatorial Optimization*, *21*(3), 348-363.]

Функции растояния

- Shortest path distance
  [Floyd, R. W. (1962). Algorithm 97: shortest path. *Communications of the ACM*, *5*(6), 345]

- Commute distance
  [Yen, L., Vanvyve, D., Wouters, F., Fouss, F., Verleysen, M., & Saerens, M. (2005). clustering using a random walk based distance measure. In *ESANN* (pp. 317-324)]

- Amplified commute distance
  [Luxburg, U. V., Radl, A., & Hein, M. (2010). Getting lost in space: Large sample analysis of the resistance distance. In *Advances in Neural Information Processing Systems* (pp. 2622-2630)]

Community distance lost in space

Property (★): Vertices in the same cluster of the graph have a small commute distance, whereas two vertices in different clusters of the graph have a "large" commute distance.

$$\frac{1}{vol(g)} C_{ij} \approx \frac{1}{d_i} + \frac{1}{d_j}$$

The commute distance is not a useful distance function on large graphs

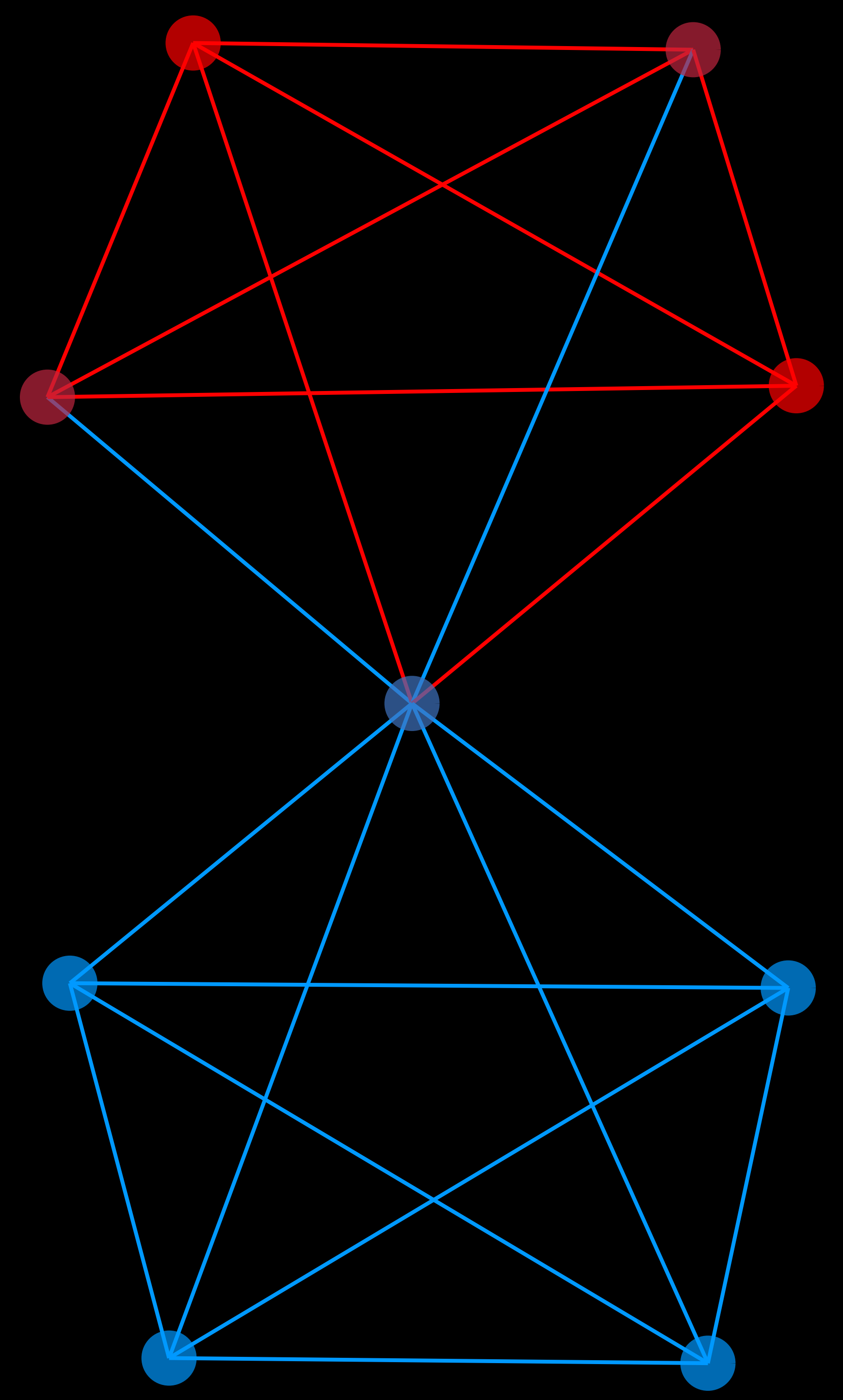
[Luxburg, U. V., Radl, A., & Hein, M. (2010). Getting lost in space: Large sample analysis of the resistance distance. In *Advances in Neural Information Processing Systems* (pp. 2622-2630)]

Amplified Commute distance

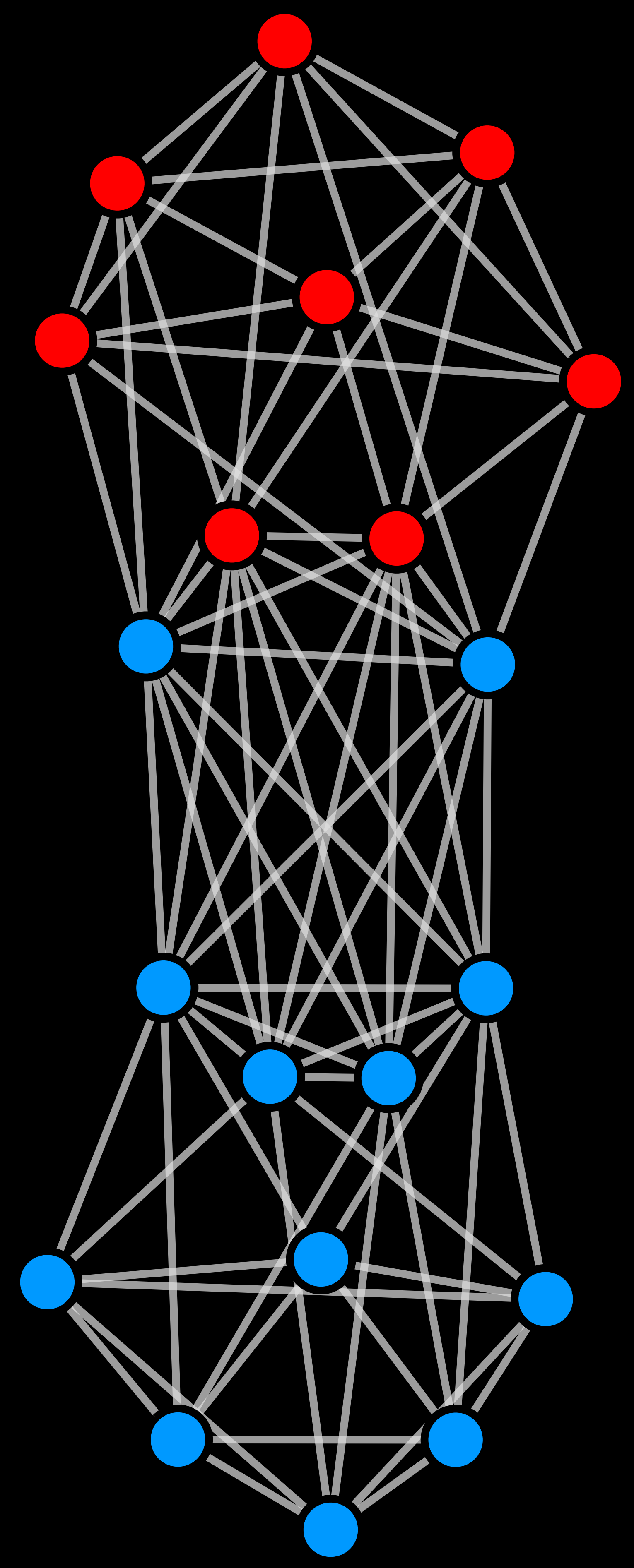
$$C_{amp}(i,j) = \frac{C_{i,j}}{vol(G)} - \frac{1}{d_i} - \frac{1}{d_j} + \frac{2w_{ij}}{d_i d_j} - \frac{w_{ii}}{d_i^2} - \frac{w_{jj}}{d_j^2}$$

[Luxburg, U. V., Radl, A., & Hein, M. (2010). Getting lost in space: Large sample analysis of the resistance distance. In *Advances in Neural Information Processing Systems* (pp. 2622-2630)]

Distance: Shortest path
Number of Clusters: 2

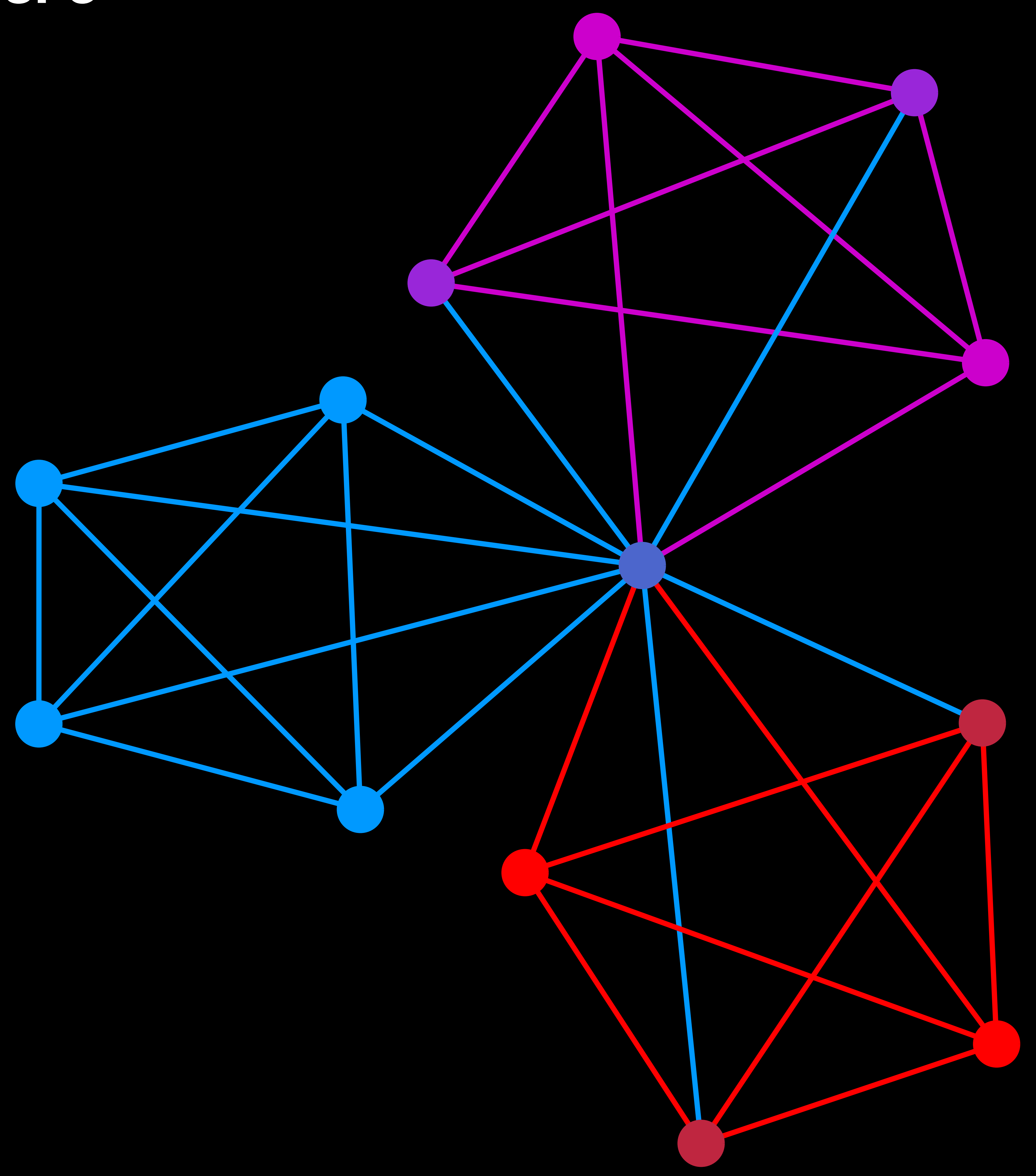


Original Graph



Line Graph

Distance: Shortest path
Number of Clusters: 3



Commute distance

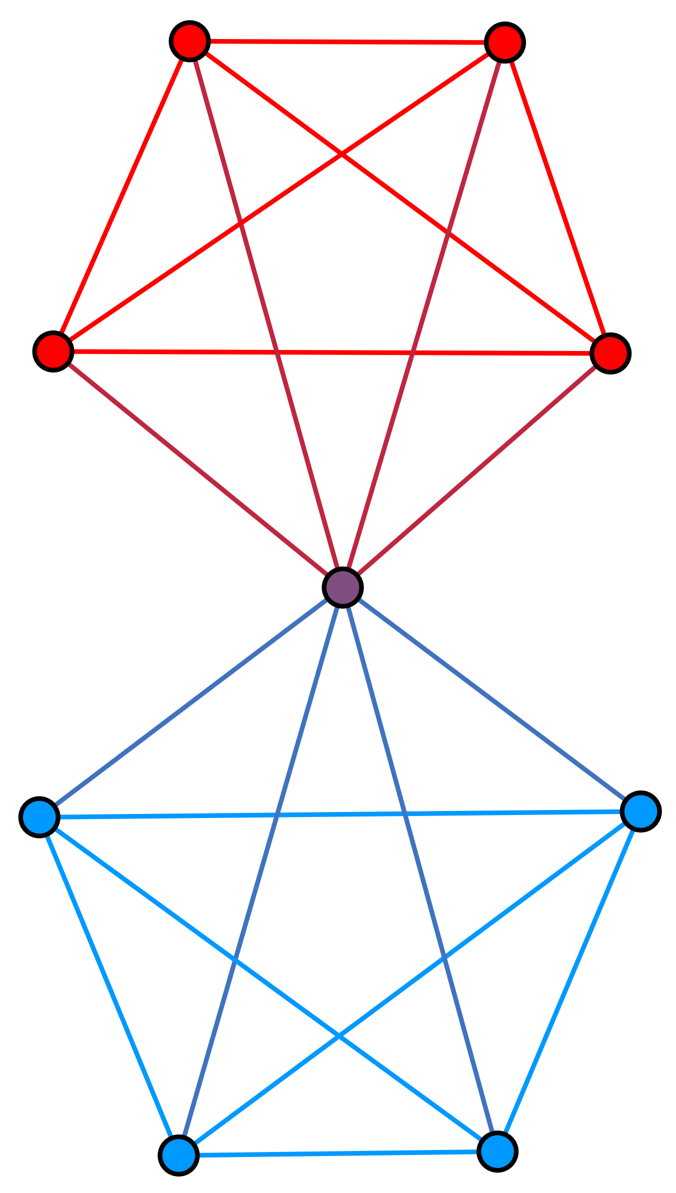
$$\text{Commute distance is } C_{ij} := H_{ij} + H_{ji}$$

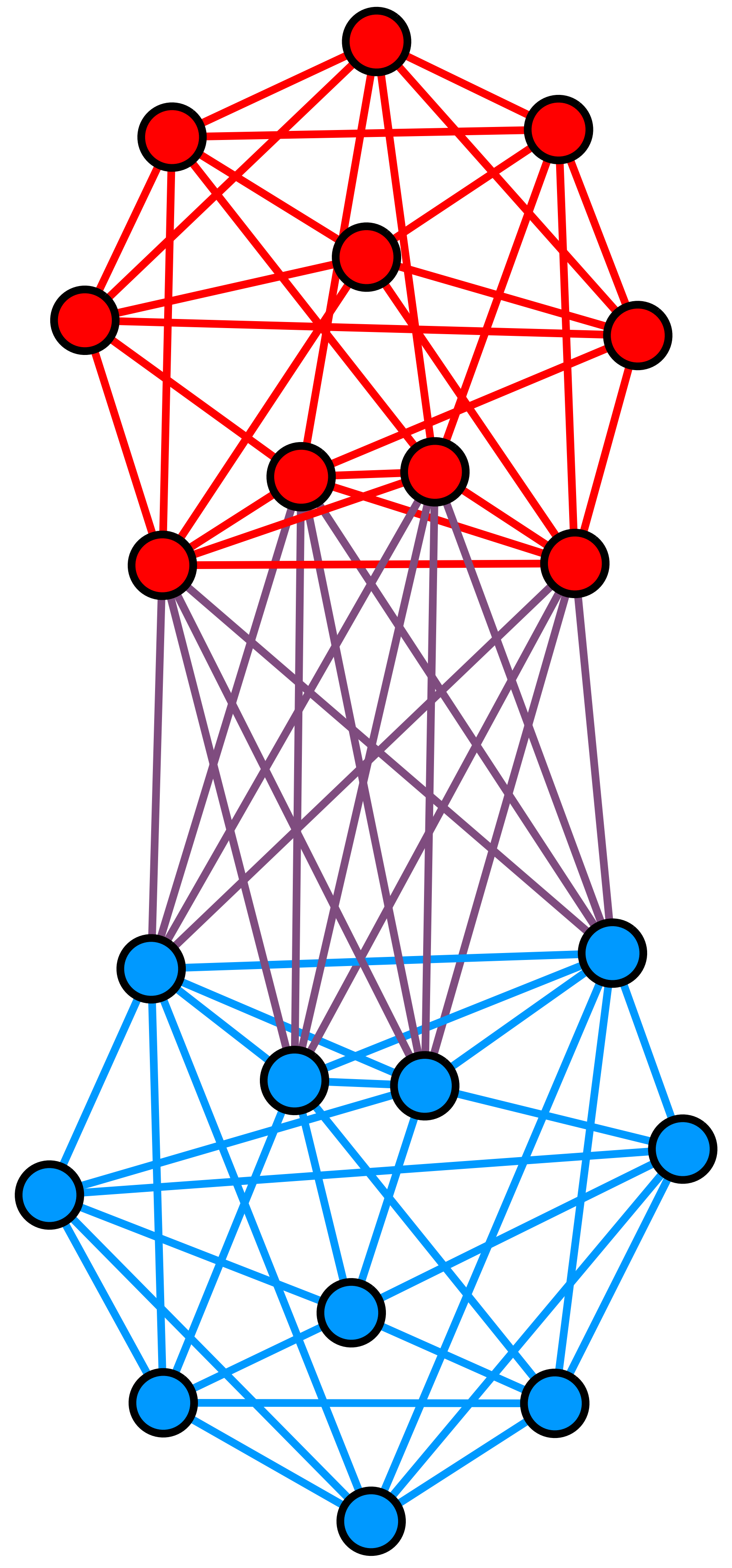
where H_{ij} is a hitting time, defined as the expected time for a random walk starting in vertex v_i to travel to vertex to v_j

A nice property: it becomes smaller when the number of path are increasing

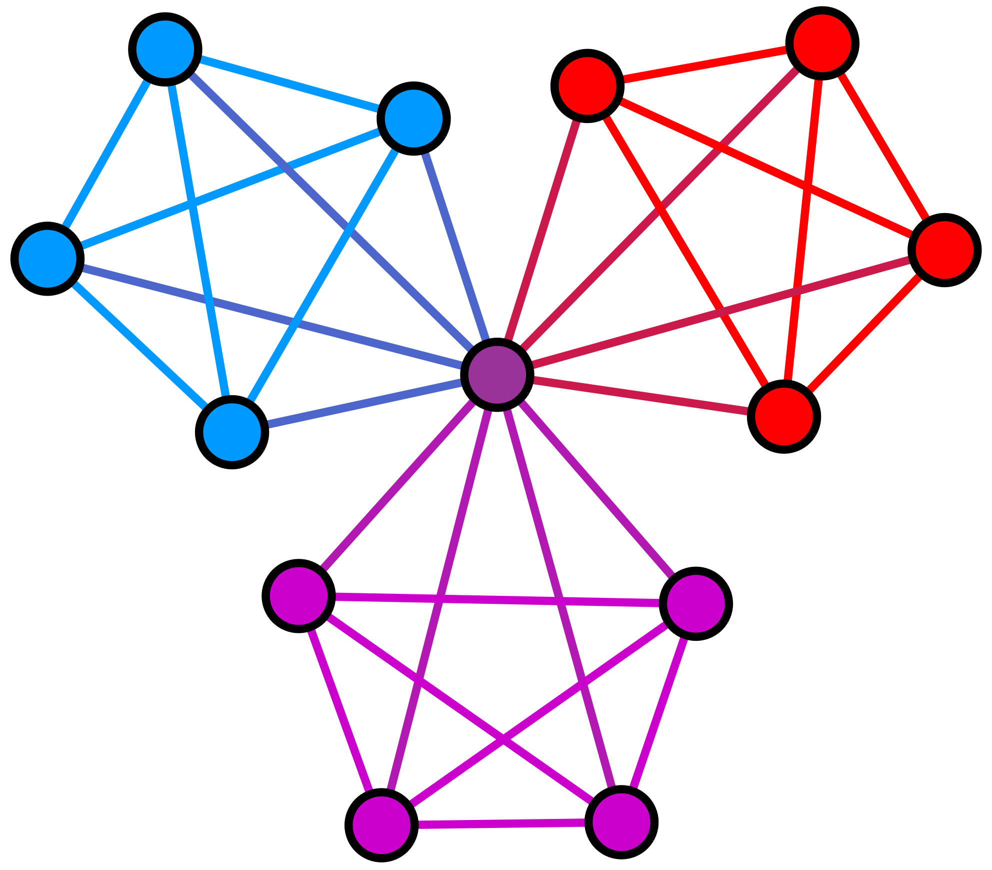
[Yen, L., Vanvyve, D., Wouters, F., Fouss, F., Verleysen, M., & Saerens, M. (2005). clustering using a random walk based distance measure. In *ESANN* (pp. 317-324)]

Distance: Commute Distance
Number of clusters: 2 Clusters

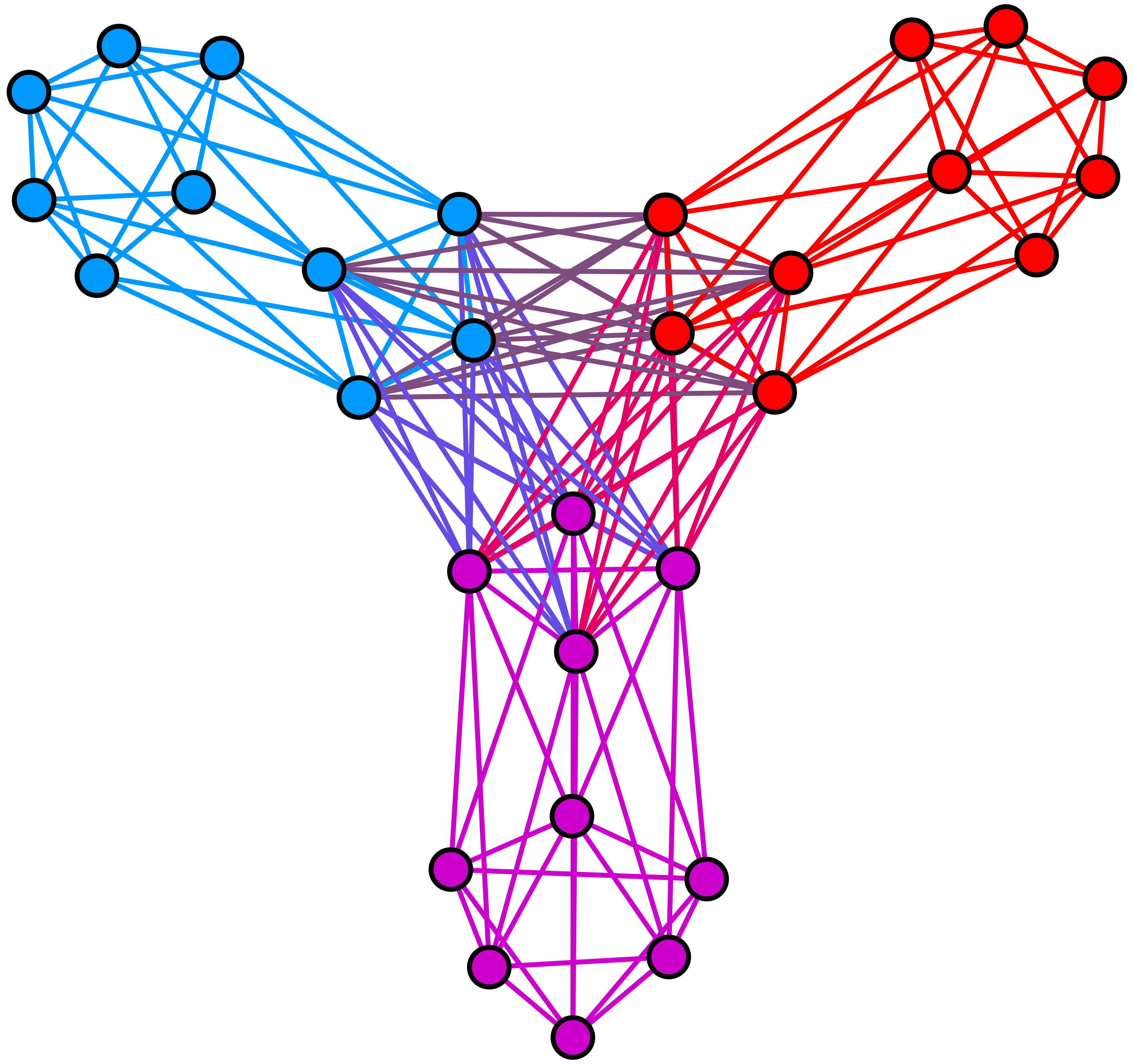




Distance: Commute Distance
Number of clusters: 3 Clusters

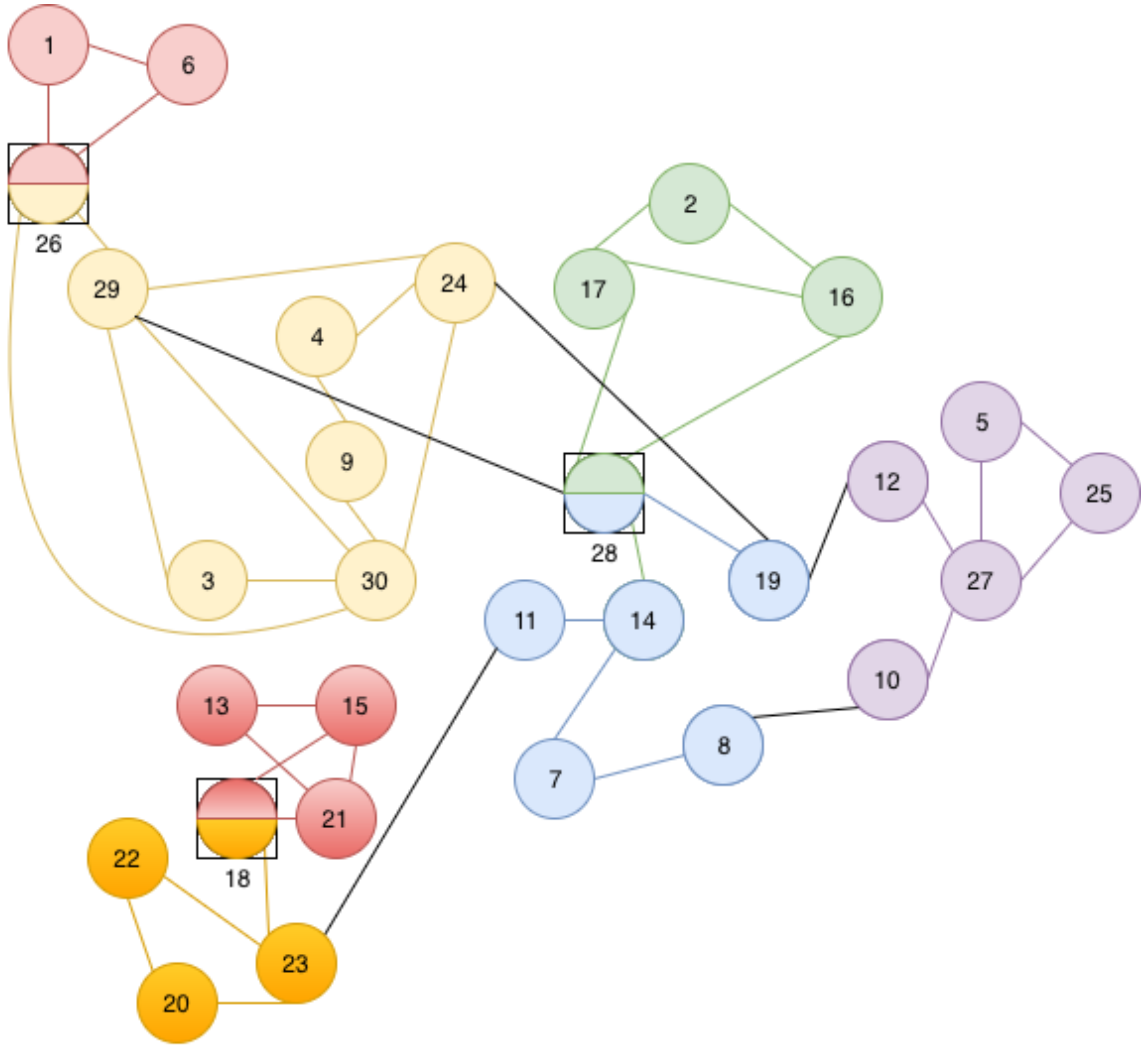


Original graph

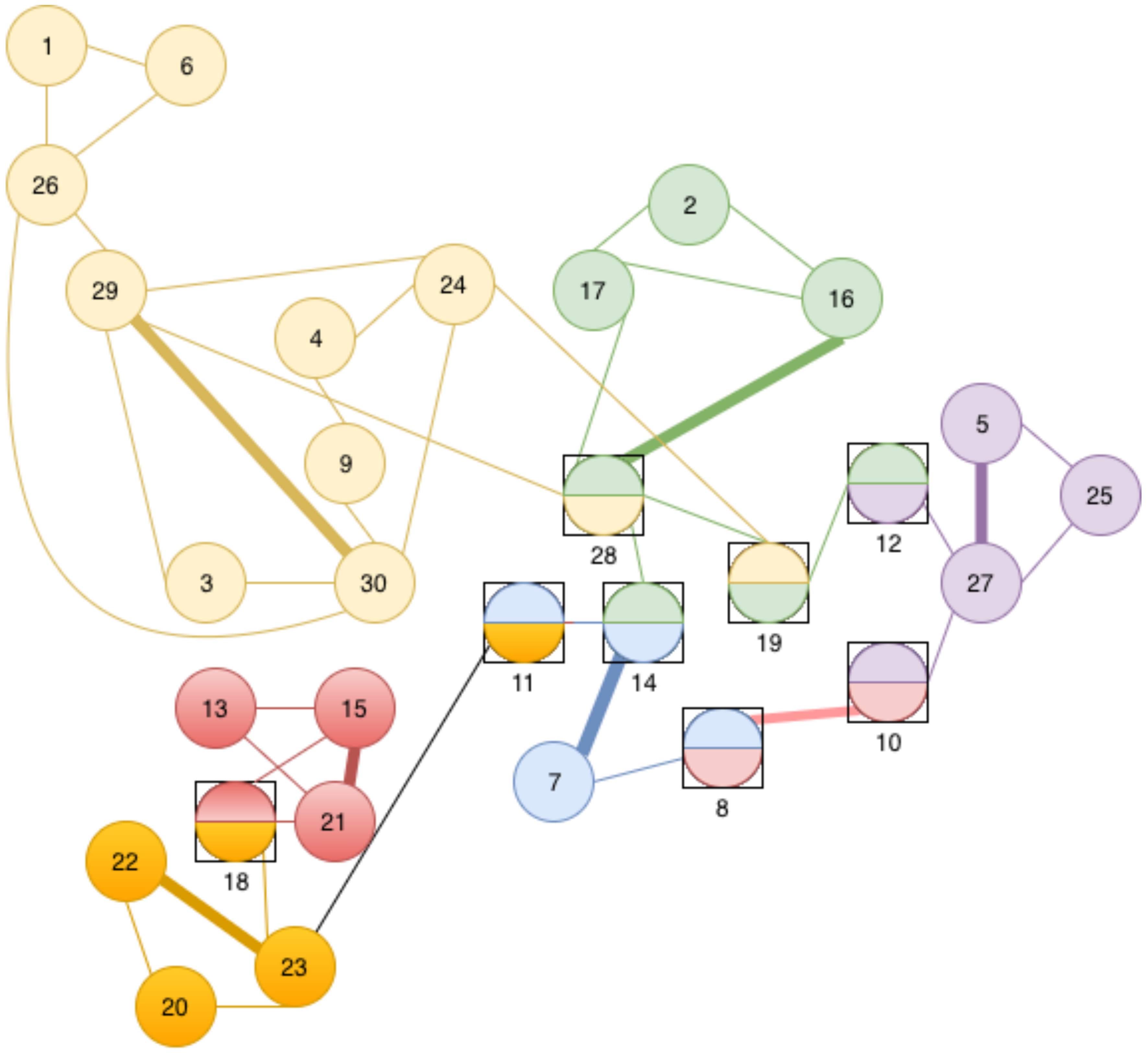


Line graph

Distance: Commute Distance
Number of clusters: 6 Clusters



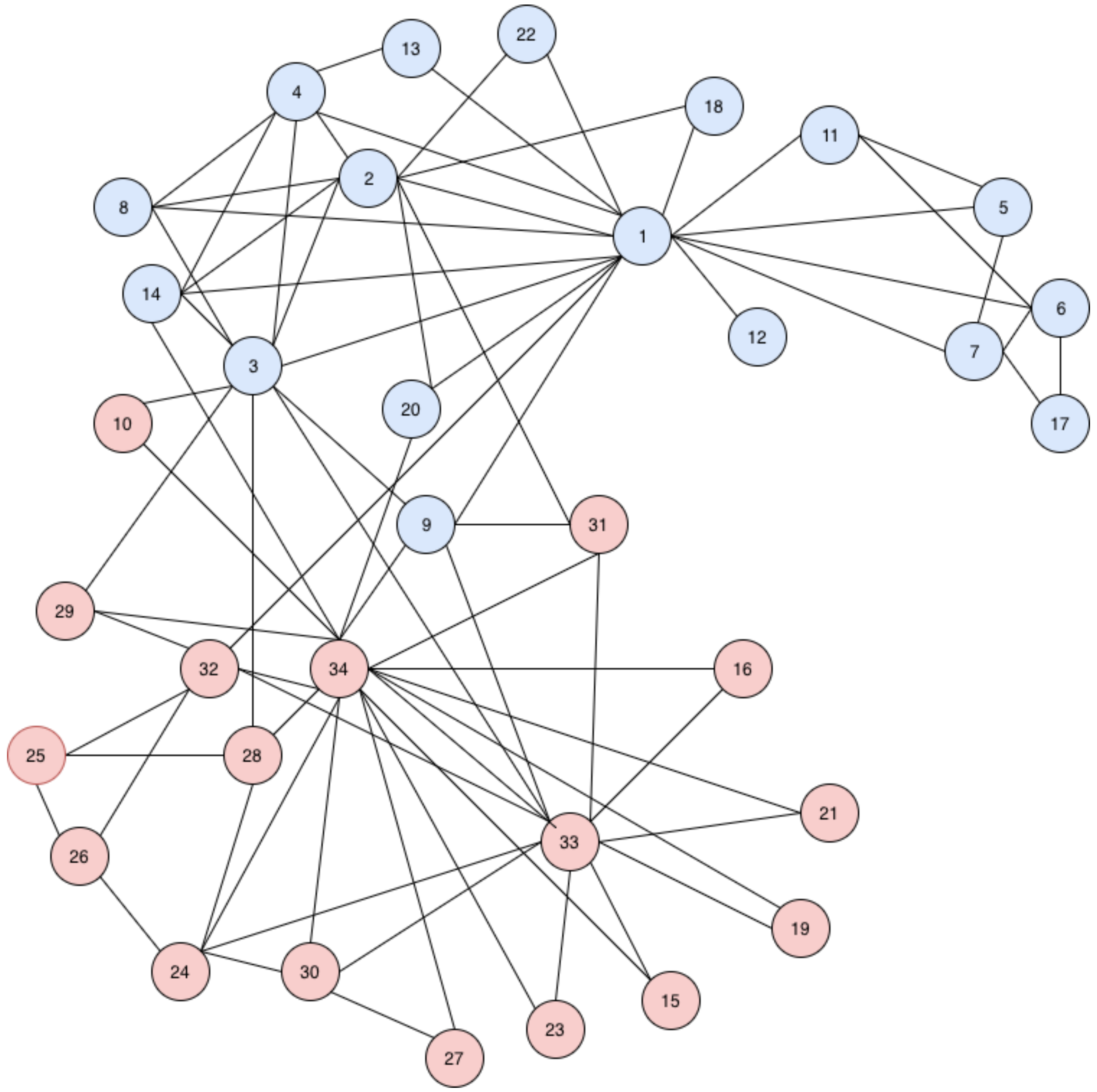
ground truth



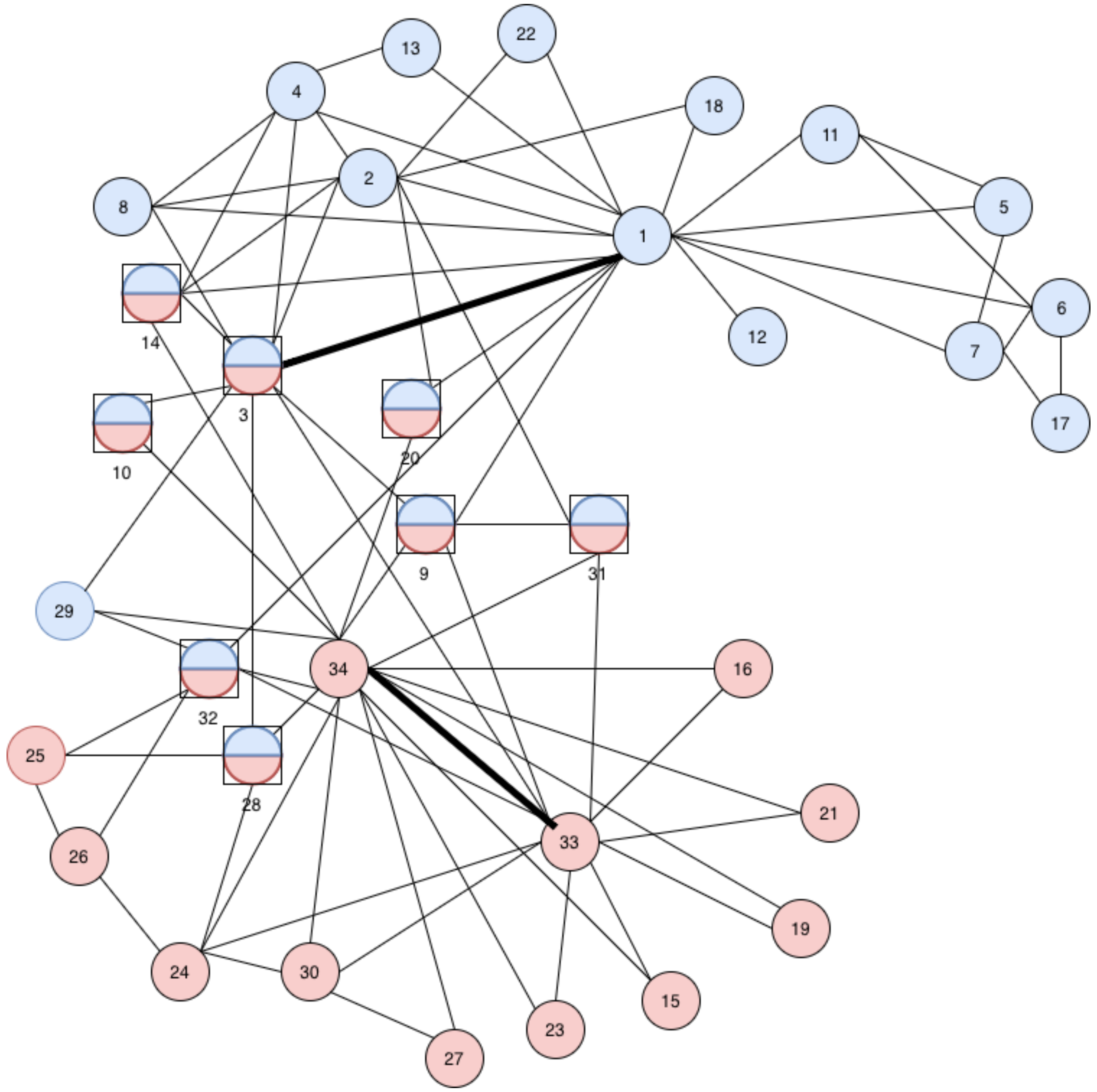
method output

Distance: Commute Distance
Number of clusters: 6 Clusters

Zachary Karate Club



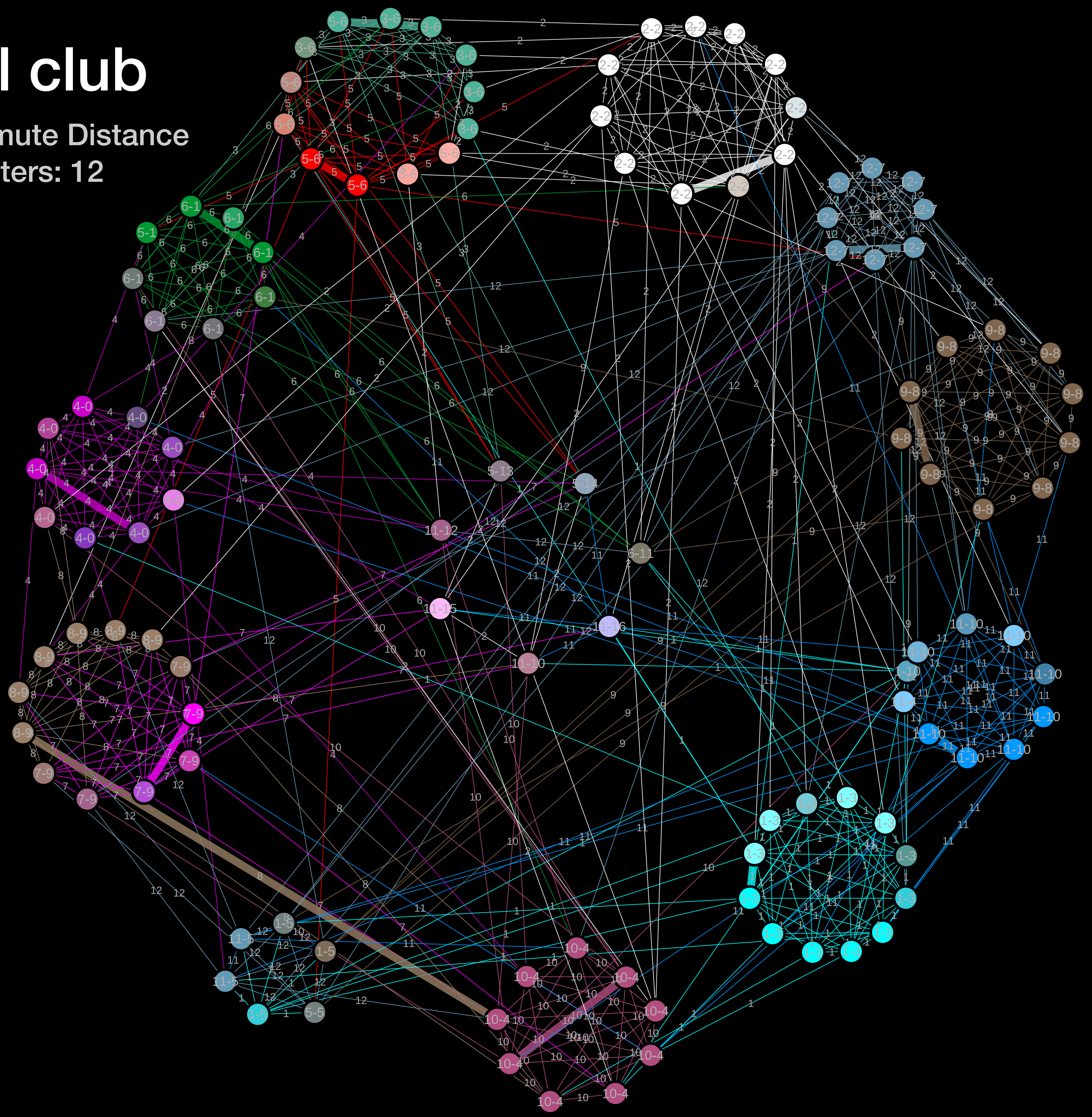
ground truth



method output

Football club

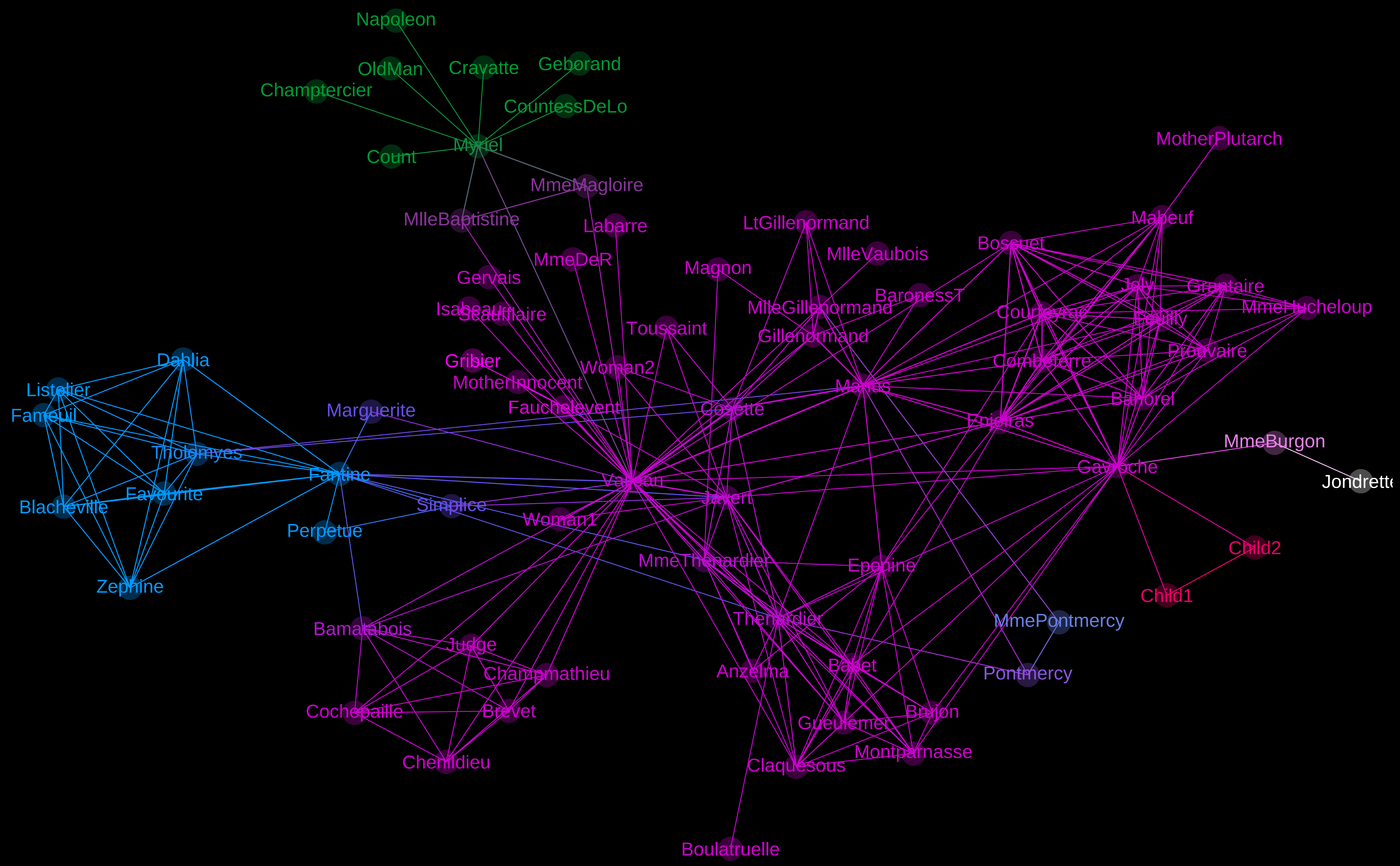
Distance: Commute Distance
Number of Clusters: 12



Les Miserables

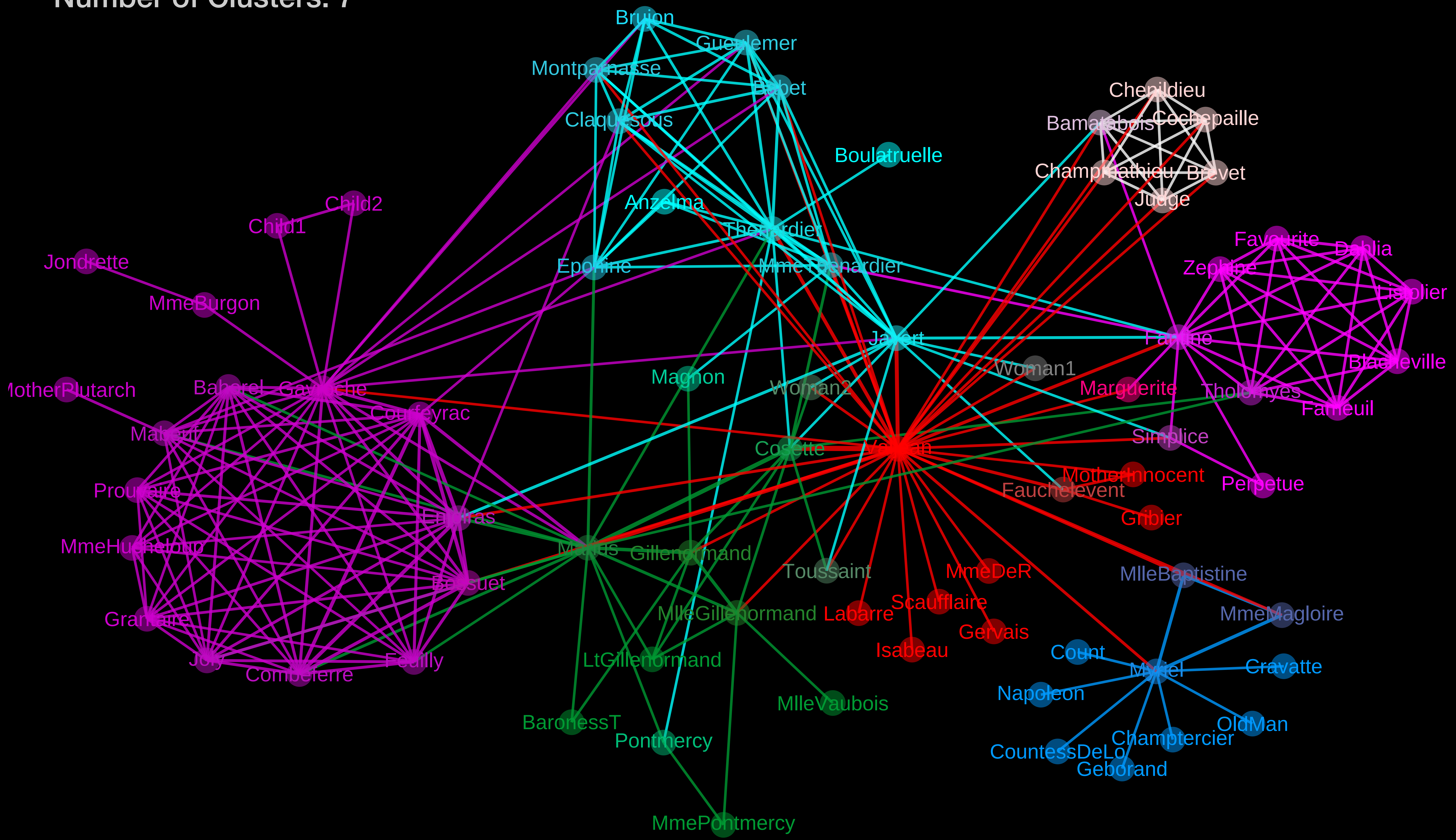
Distance: Commute Distance

Number of Clusters: 7



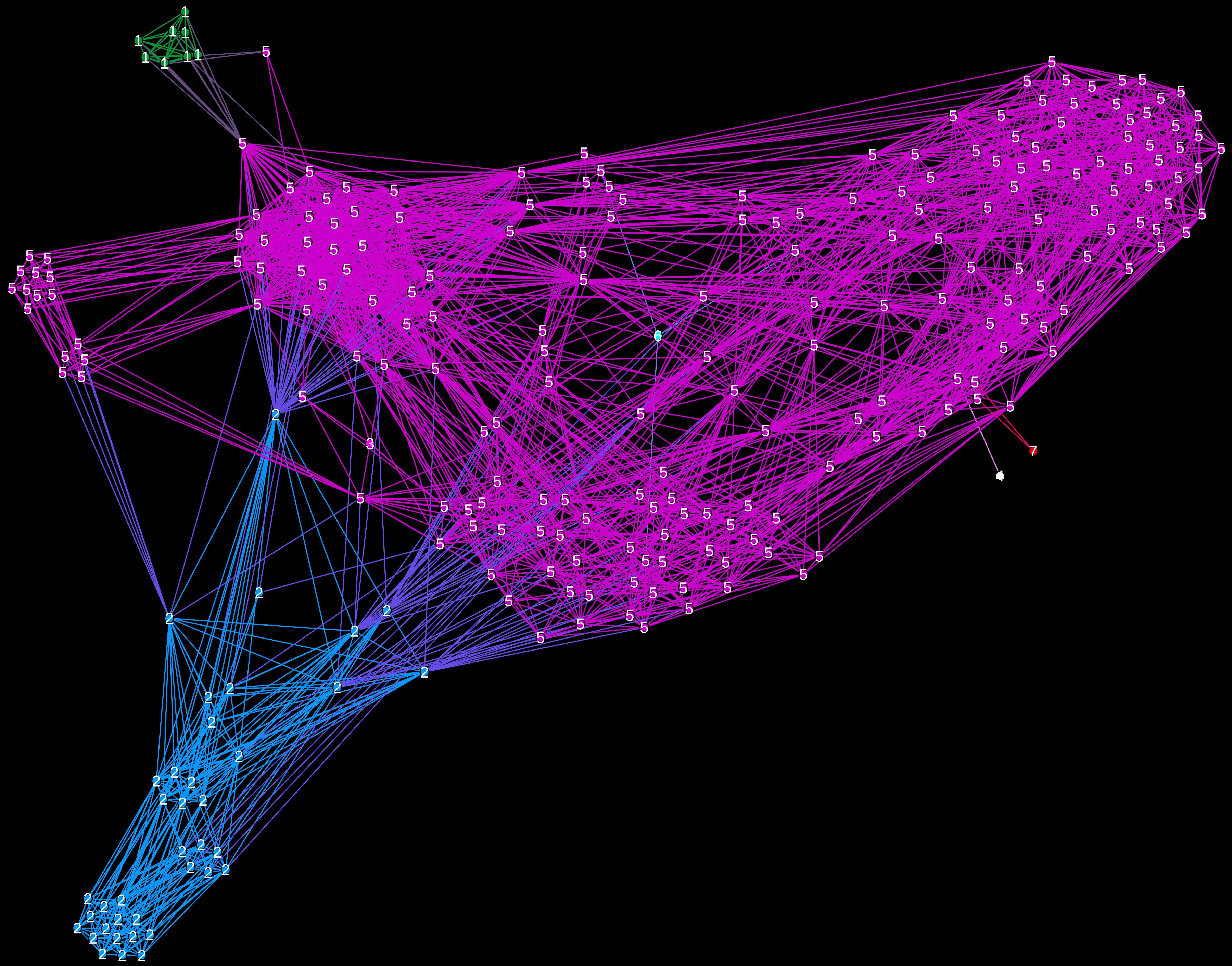
Les Miserables

Distance: Amplified Commute Distance
Number of Clusters: 7



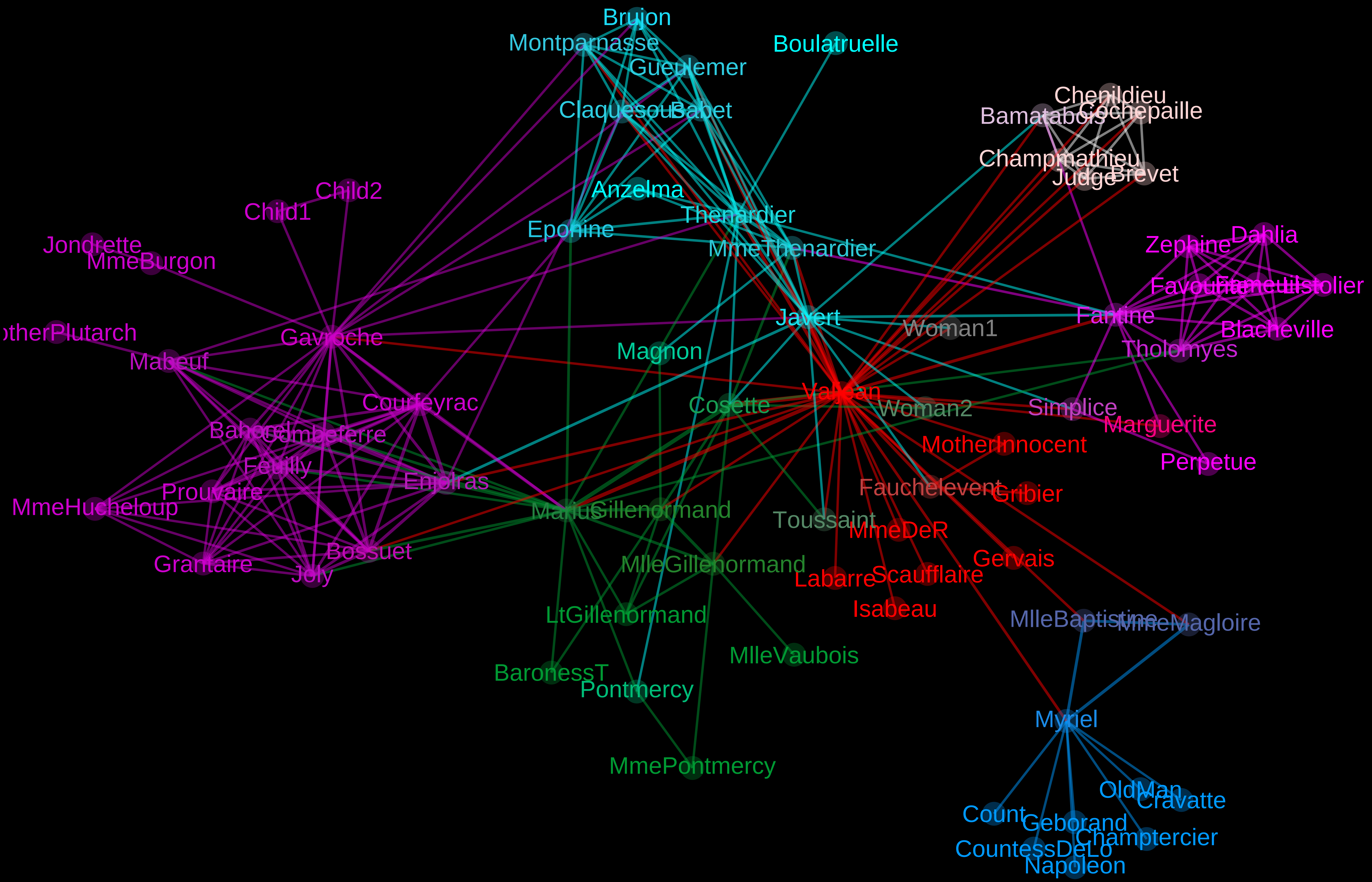
Les Miserables – line graph

Distance: Commute Distance
Number of Clusters: 7



Les Miserables

Distance: Amplified Commute Distance
Number of Clusters: 7



Les Miserables – line graph

Distance: Amplified Commute Distance
Number of Clusters: 7

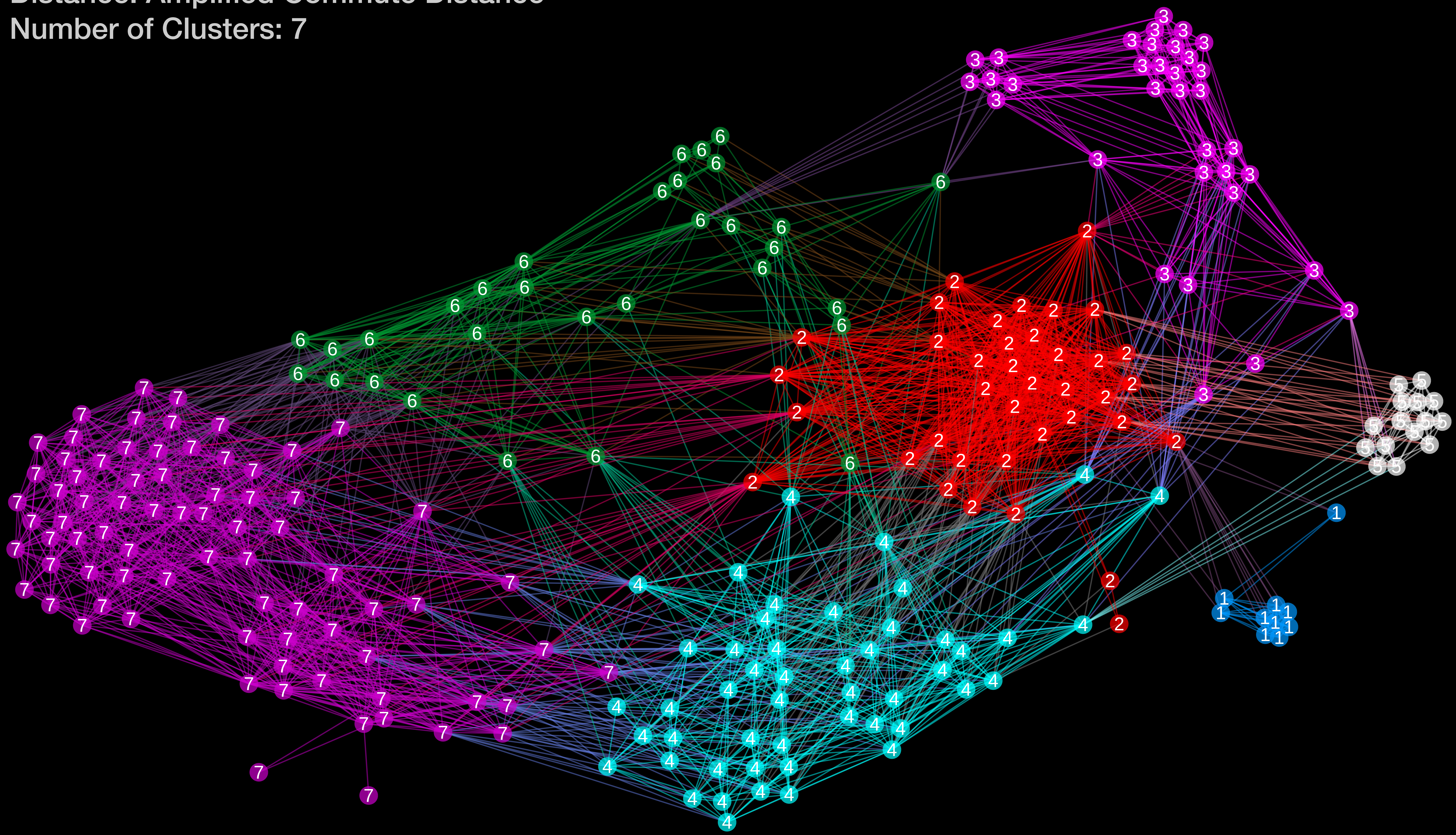
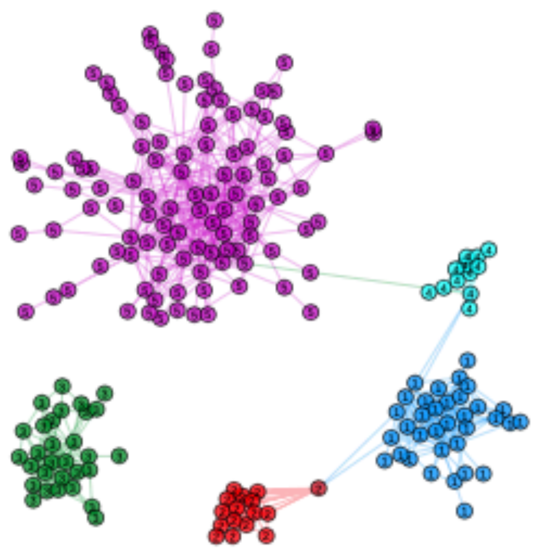
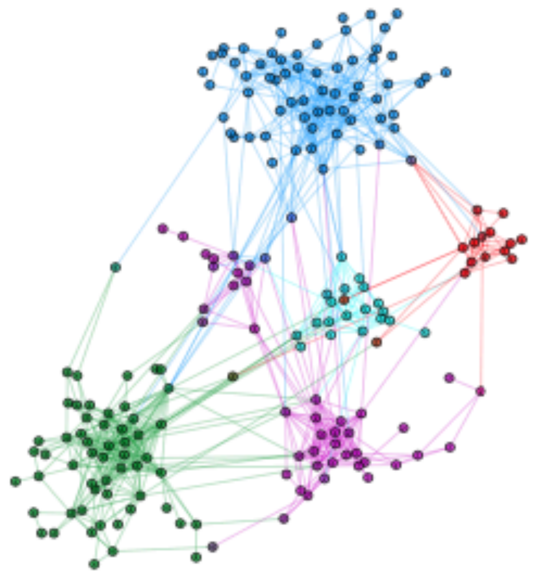


Table 6. Clustering results of heuristic version of LPAM method with Amplified Commute Distance for FARZ networks with 200 nodes and 20 communities

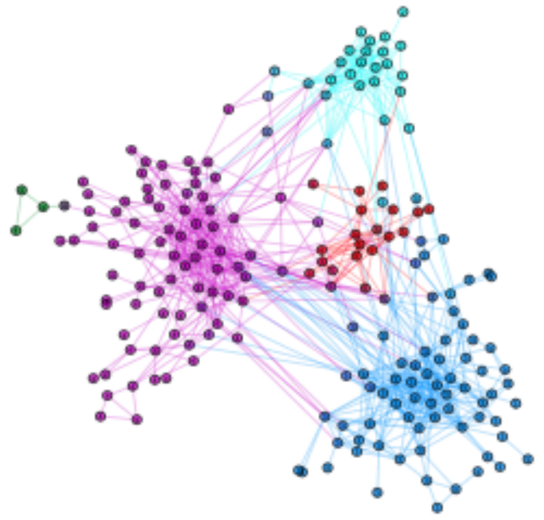


(a) $\beta = 1$

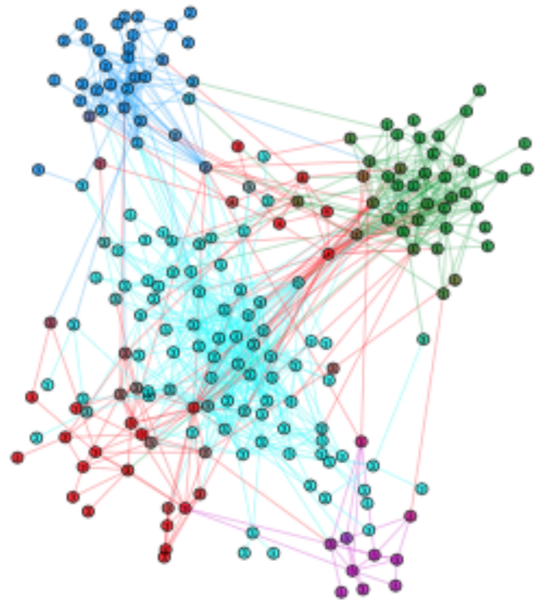
(b) $\beta = 0.95$



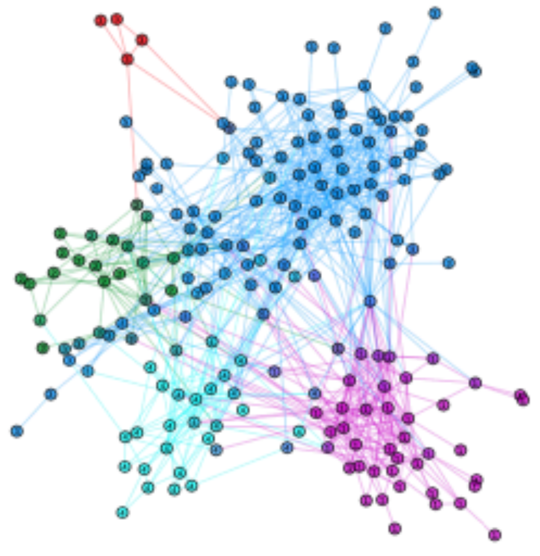
(c) $\beta = 0.9$



(d) $\beta = 0.85$



(e) $\beta = 0.8$



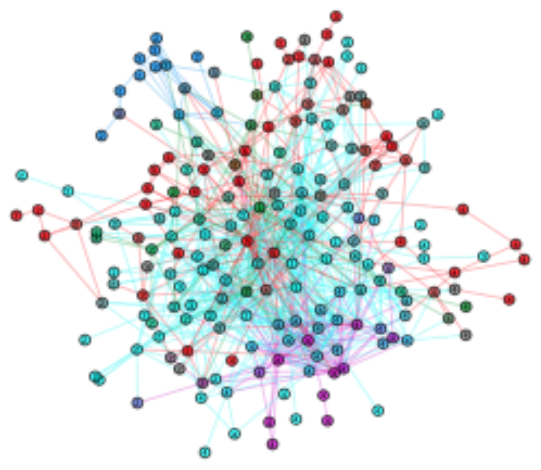
(f) $\beta = 0.75$

(g) $\beta = 0.7$

(h) $\beta = 0.65$

(j) $\beta = 0.6$

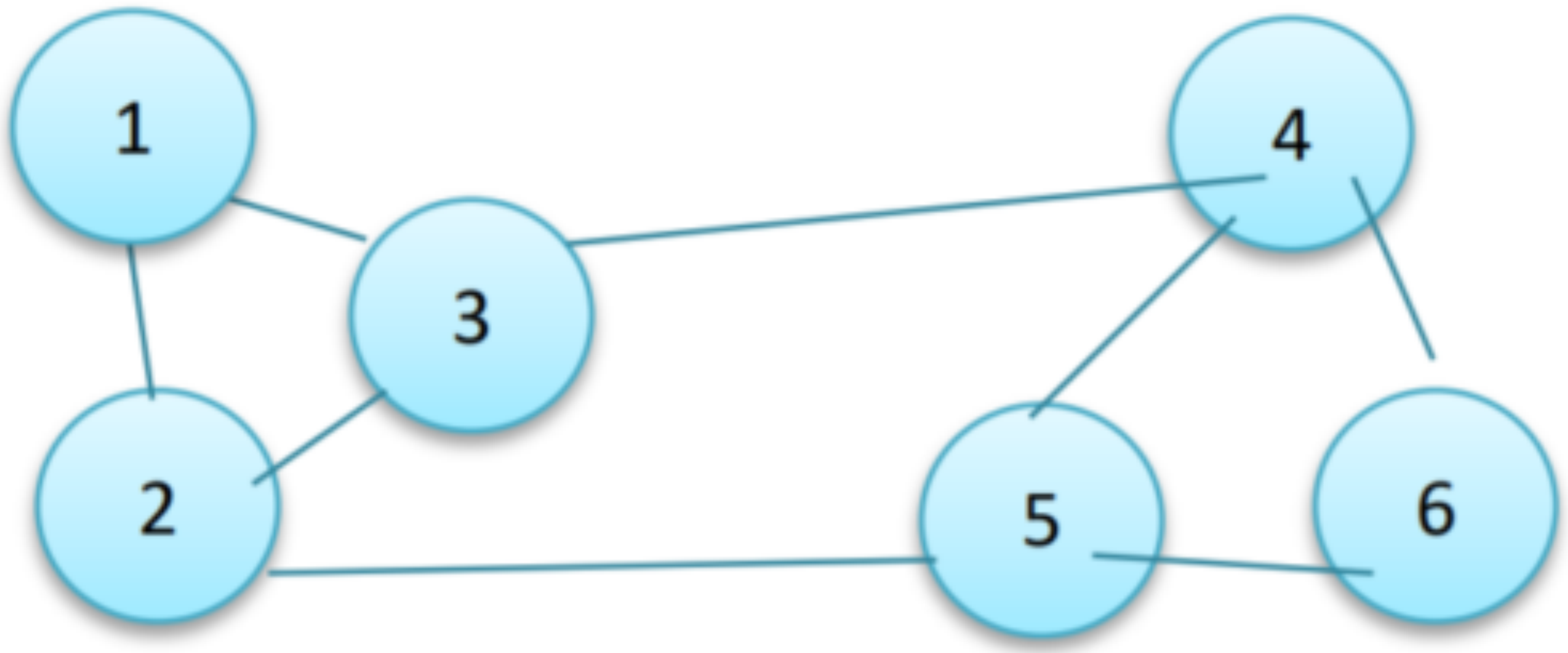
(k) $\beta = 0.55$



(l) $\beta = 0.5$

Spectral Clustering

n × n symmetric matrix



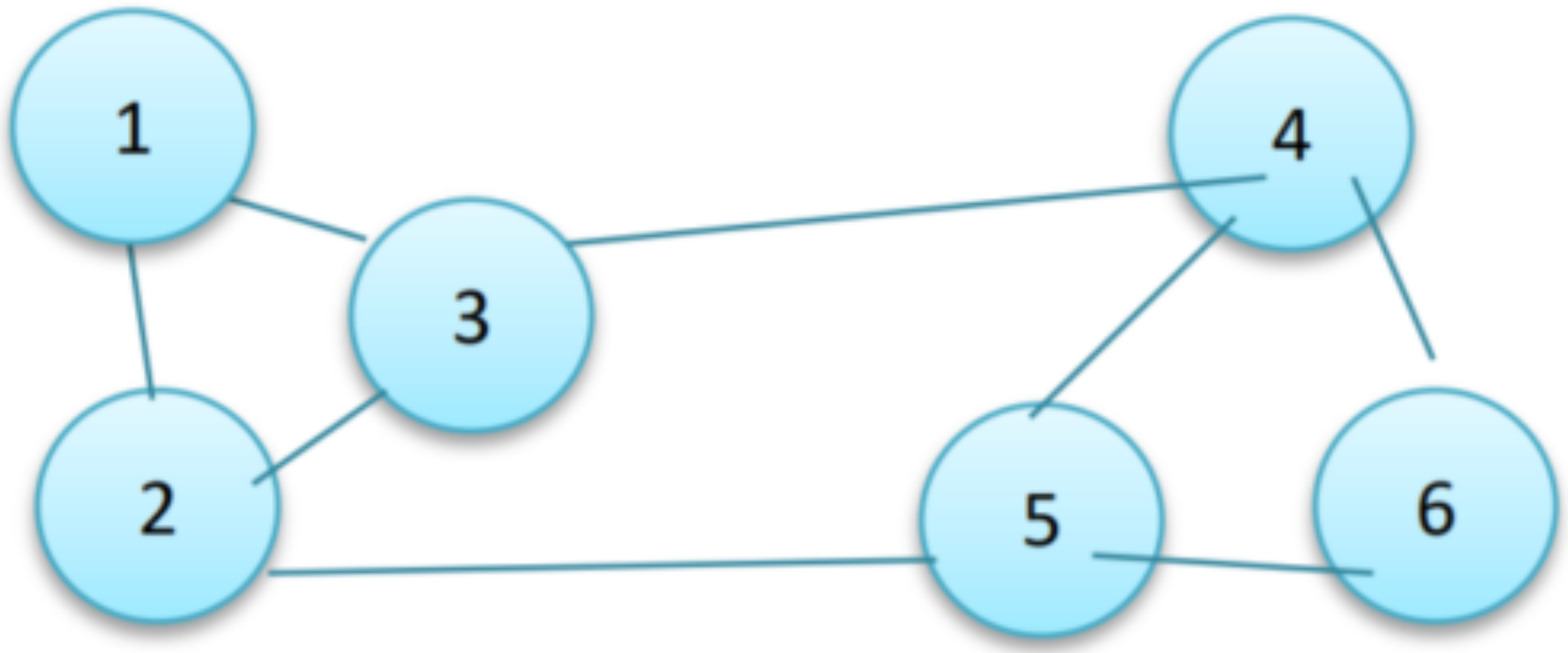
	1	2	3	4	5	6
1	3	-1	0	-1	-1	0
2	0	2	-1	0	-1	0
3	-1	0	3	0	-1	-1
4	0	0	-1	3	0	-1
5	-1	-1	0	0	2	0
6	0	-1	0	-1	0	3

To gain insights and perform clustering, the eigenvalues of **L** are used.

1. собственные значения не отрицательные
2. собственные вектора вещественные (и всегда ортогональные)

какой есть тривиальный собственный вектор?

n × n symmetric matrix



	1	2	3	4	5	6
1	3	-1	0	-1	-1	0
2	0	2	-1	0	-1	0
3	-1	0	3	0	-1	-1
4	0	0	-1	3	0	-1
5	-1	-1	0	0	2	0
6	0	-1	0	-1	0	3

To gain insights and perform clustering, the eigenvalues of **L** are used.

1. собственные значения не отрицательные
2. собственные вектора вещественные (и всегда ортогональные)

какой есть тривиальный собственный вектор?

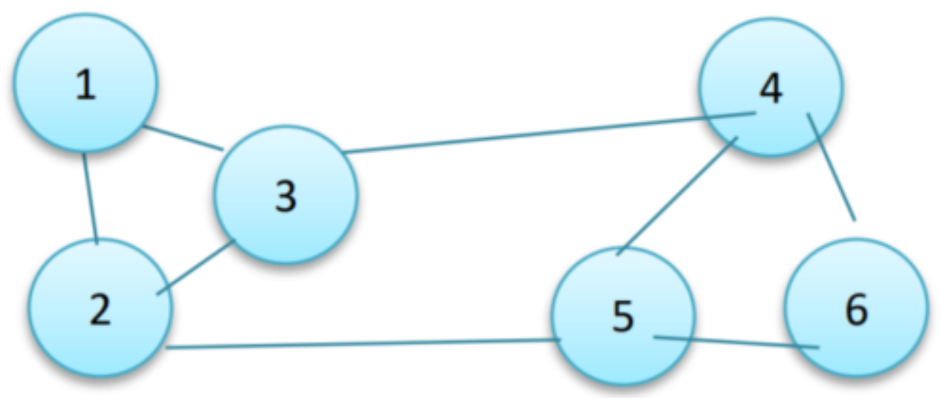
$x = (1, ..., 1)$ $L \cdot x = 0$ $\lambda = \lambda_1 = 0$

Оптимизационная задача $$\lambda_2 = \min_{x^T w_1 = 0} \frac{x^T M x}{x^T x}$$ **Для симметричной матрицы *M***

$$x^T L x = \sum_{i,j=1}^{n} L_{ij} x_i x_j = \sum_{i,j=1}^{n} (D_{ij} - A_{ij}) x_i x_j = \sum_i D_{ii} x_i^2 - \sum_{(i,j) \in E} 2 x_i x_j$$

$$= \sum_{(i,j) \in E} (x_i^2 + x_j^2 - 2 x_i x_j) = \sum_{(i,j) \in E} (x_i - x_j)^2$$

n × n symmetric matrix



	1	2	3	4	5	6
1	3	-1	0	-1	-1	0
2	0	2	-1	0	-1	0
3	-1	0	3	0	-1	-1
4	0	0	-1	3	0	-1
5	-1	-1	0	0	2	0
6	0	-1	0	-1	0	3

To gain insights and perform clustering, the eigenvalues of **L** are used.

Что мы знаем о *x?*

$$= \sum_{(i,j)\in E} (x_i^2 + x_j^2 - 2x_i x_j) = \sum_{(i,j)\in E} (x_i - x_j)^2$$

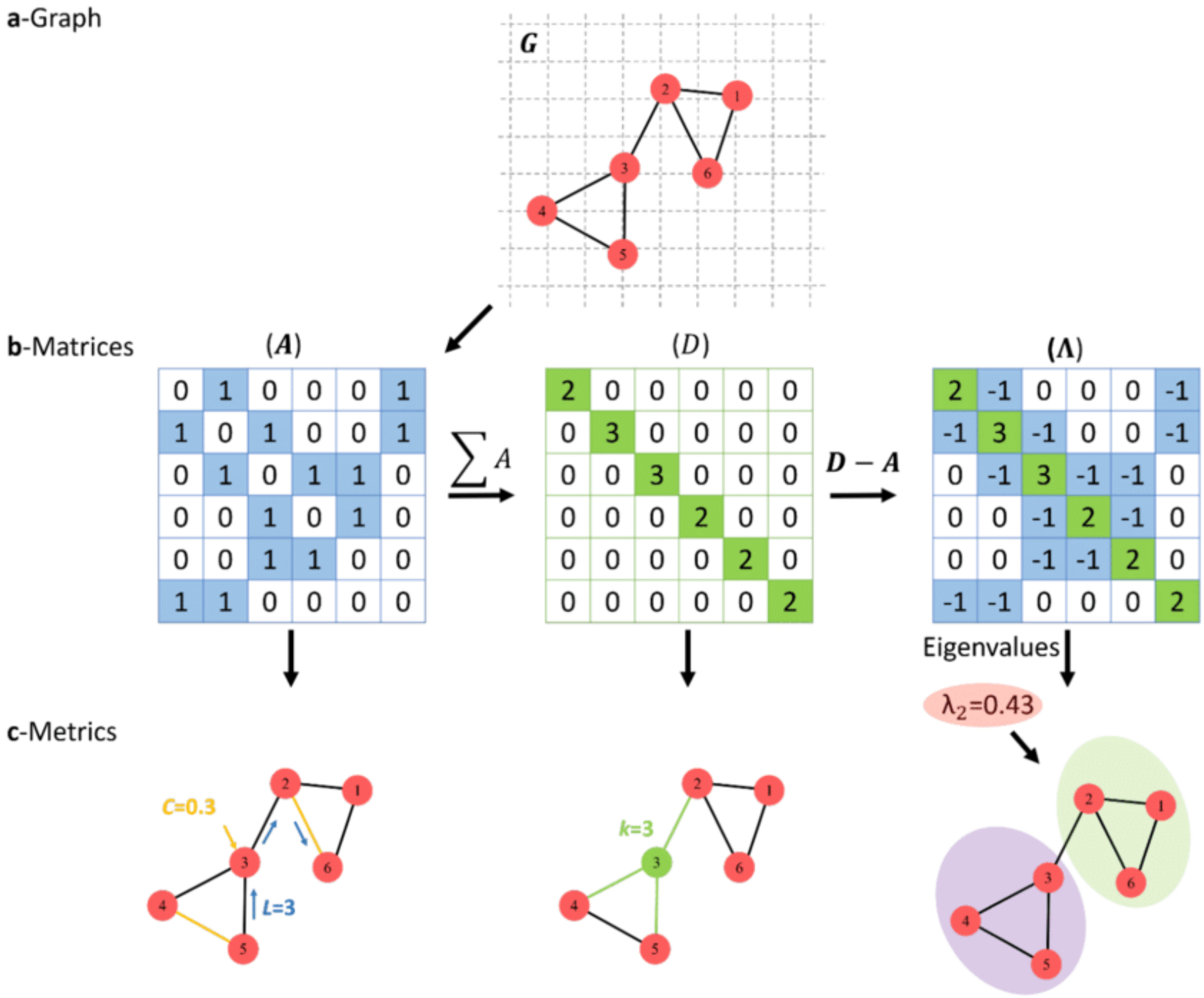
$$\sum_i x_i^2 = 1$$

$$w_1 = (1, ..., 1)$$

$$\sum_i x_i \cdot 1 = \sum_i x_i = 0$$

$$\sum_i x_i = 0$$

$$\boxed{\lambda_2 = \min_{x^T w_1 = 0} \frac{x^T M x}{x^T x}}$$



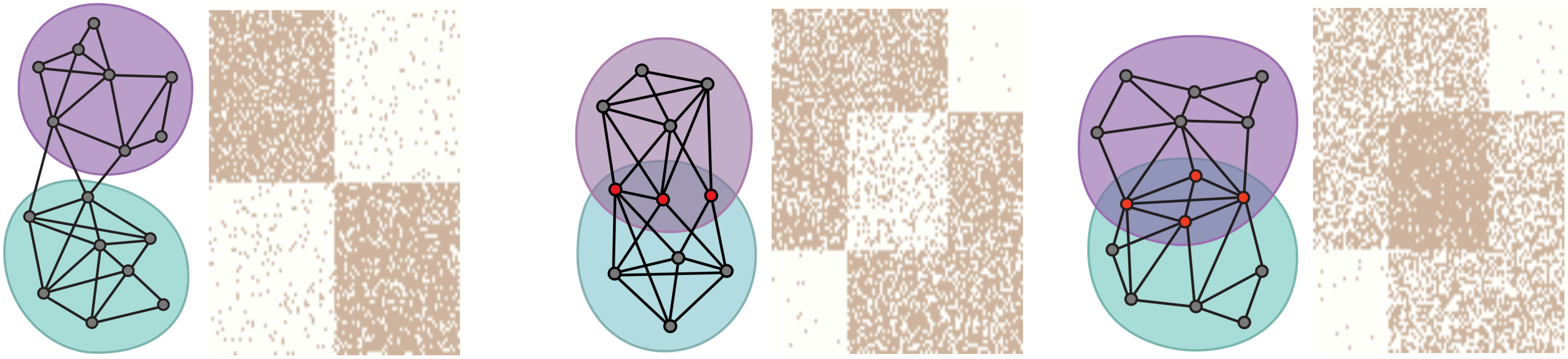
a-Graph
G
b-Matrices
(A)
(D)
(Λ)
∑A
D − A
Eigenvalues
λ2=0.43
c-Metrics
C=0.3
L=3
k=3

BIGCLAM

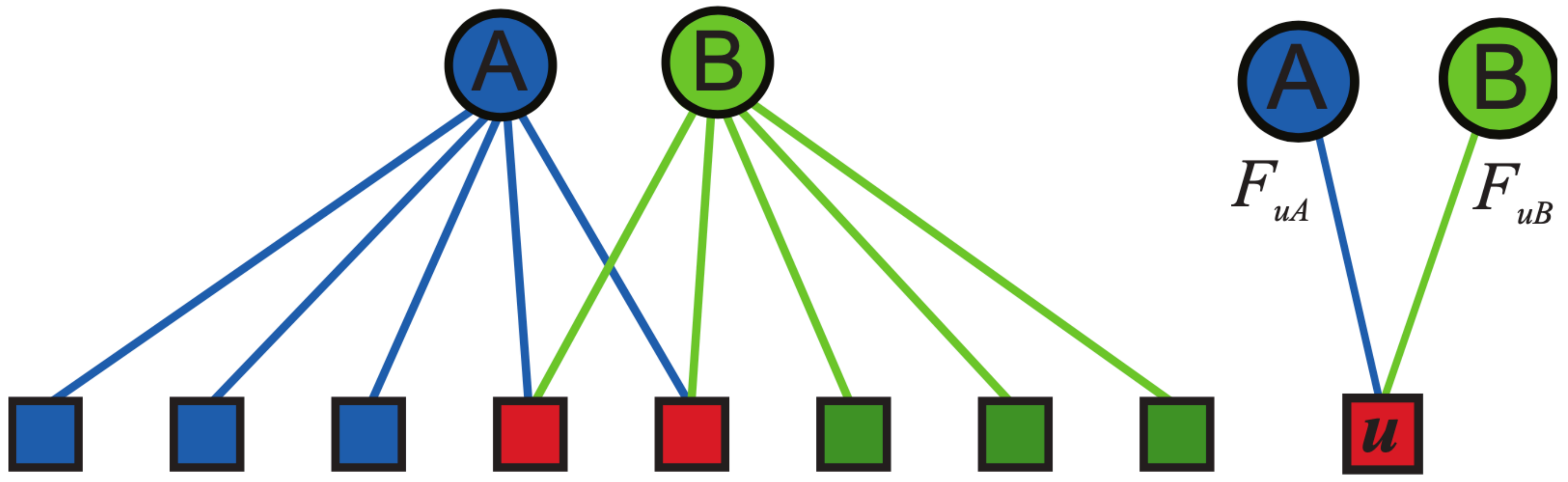
Cluster Affiliation Model for Big Networks

[Yang J., Leskovec J. Overlapping community detection at scale: a nonnegative matrix factorization approach //Proceedings of the sixth ACM international conference on Web search and data mining. – 2013. – C. 587-596]

No overlaps **Sparse overlaps** **Dense overlaps**



BIGCLAM bible to find densely connected community overlaps

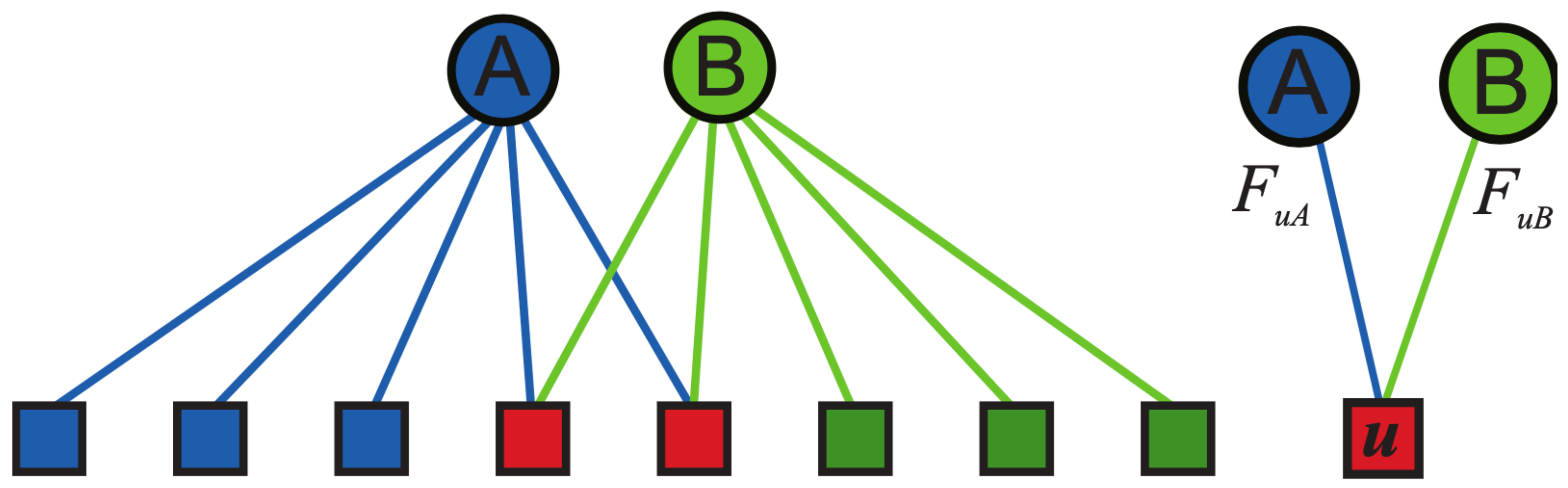


Bipartite community affiliation graph. Circles: Communities, Squares: Nodes of the underlying network. Edges indicate node community memberships. Edges with zero weight are not shown.

Each affiliation edge from node u to community c has strength $F_{uc} \geq 0$

$$p_c((u,v) \in E) = 1 - e^{-F_{uc} \cdot F_{vc}}$$

$$p((u,v) \in E) = 1 - e^{-\sum_c F_{uc} \cdot F_{vc}}$$



$$p_c((u,v) \in E) = 1 - e^{-F_{uc} \cdot F_{vc}}$$

$$p((u,v) \in E) = 1 - e^{-\sum_c F_{uc} \cdot F_{vc}}$$

likelihood $l(F) = \log P(G|F)$

$$\hat{F} = \underset{F \geq 0}{\operatorname{argmax}}\, l(F) \qquad \hat{F}, F \in \mathbb{R}^{N \times K}$$

$$l(F) = \sum_{(u,v) \in E} \log(1 - \exp(-F_u F_v^T)) - \sum_{(u,v) \notin E} F_u F_v^T$$