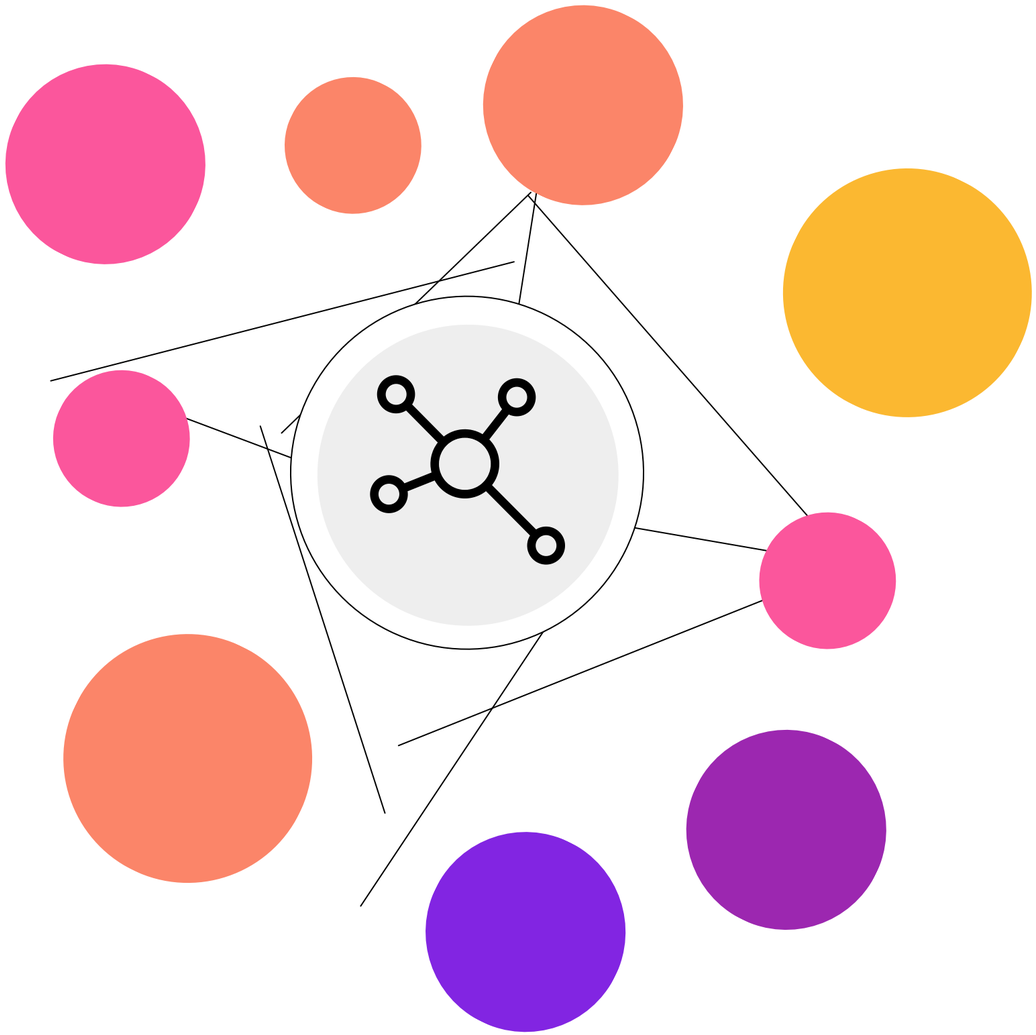
CS328



Sparsifying Graphs While Preserving Communities

Guntas Singh Saran (22110089)
Hrriday V. Ruparel (22110099)
Yajurvedh Bodala (19110077)

Context and Problem Statement

Data

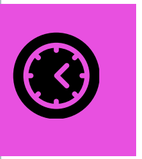
Modelling of large data as graphs and networks is a common technique. Doing so is beneficial to capture the nature of connections between entities.

Communities

Among multiple regimes of graph analysis algorithms, community structure detection are much sought after. They extract the close relationships of social, biological and physical networks.

Relevance

These algorithms are used for targeted advertising, identifying influential nodes, protein-protein interactions, brain connectivity networks and recommender systems.



Complexity

However, as data size increases, the time taken to run community detection algorithms blows up. Not to mention, the storage costs also increases significantly.

Broad Problem Statement:

Given a graph $G(V,E)$, where V is the vertex set and E is the edge set, we wish to "sparsify" the graph in a "meaningful" manner and obtain a graph $G'(V,E')$, where V is the vertex set similar to original graph G and $E' \subset E$ is the edge set of sparsified graph G'.

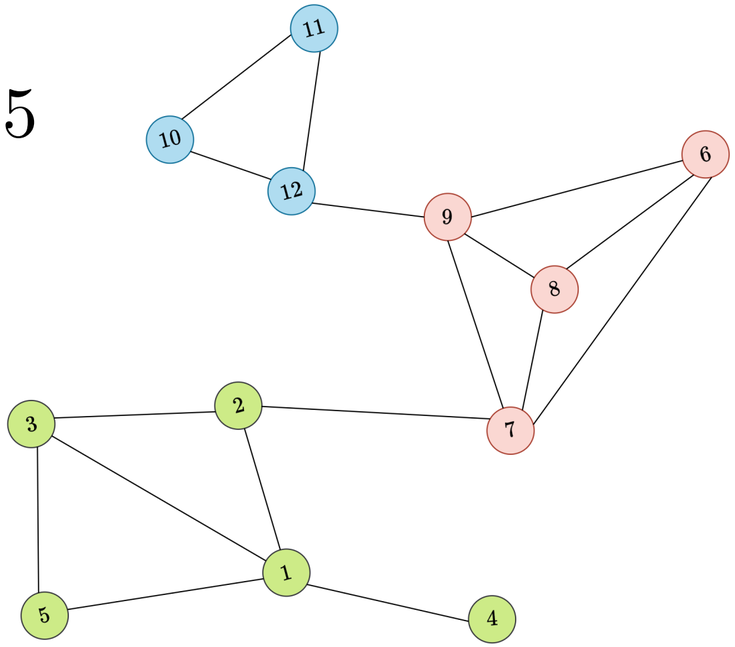
Context and Problem Statement

Broad Problem Statement:

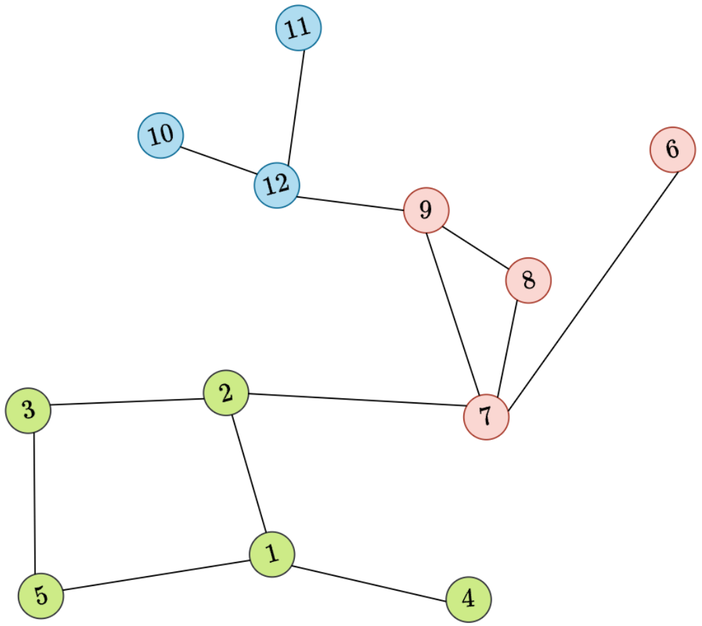
Given a graph $G(V,E)$, where V is the vertex set and E is the edge set, we wish to "sparsify" the graph in a "meaningful" manner and obtain a graph $G'(V,E')$, where V is the vertex set similar to original graph G and $E' \subset E$ is the edge set of sparsified graph G'.

By "sparsifying" a graph in a "meaningful manner", we intend to preserve the community structure of the original graph G by sampling a subset of edges from G so that community detection algorithms can be implemented on the "sparsified" graph G' with minimum loss of generality and accuracy.

$|E| = 15$

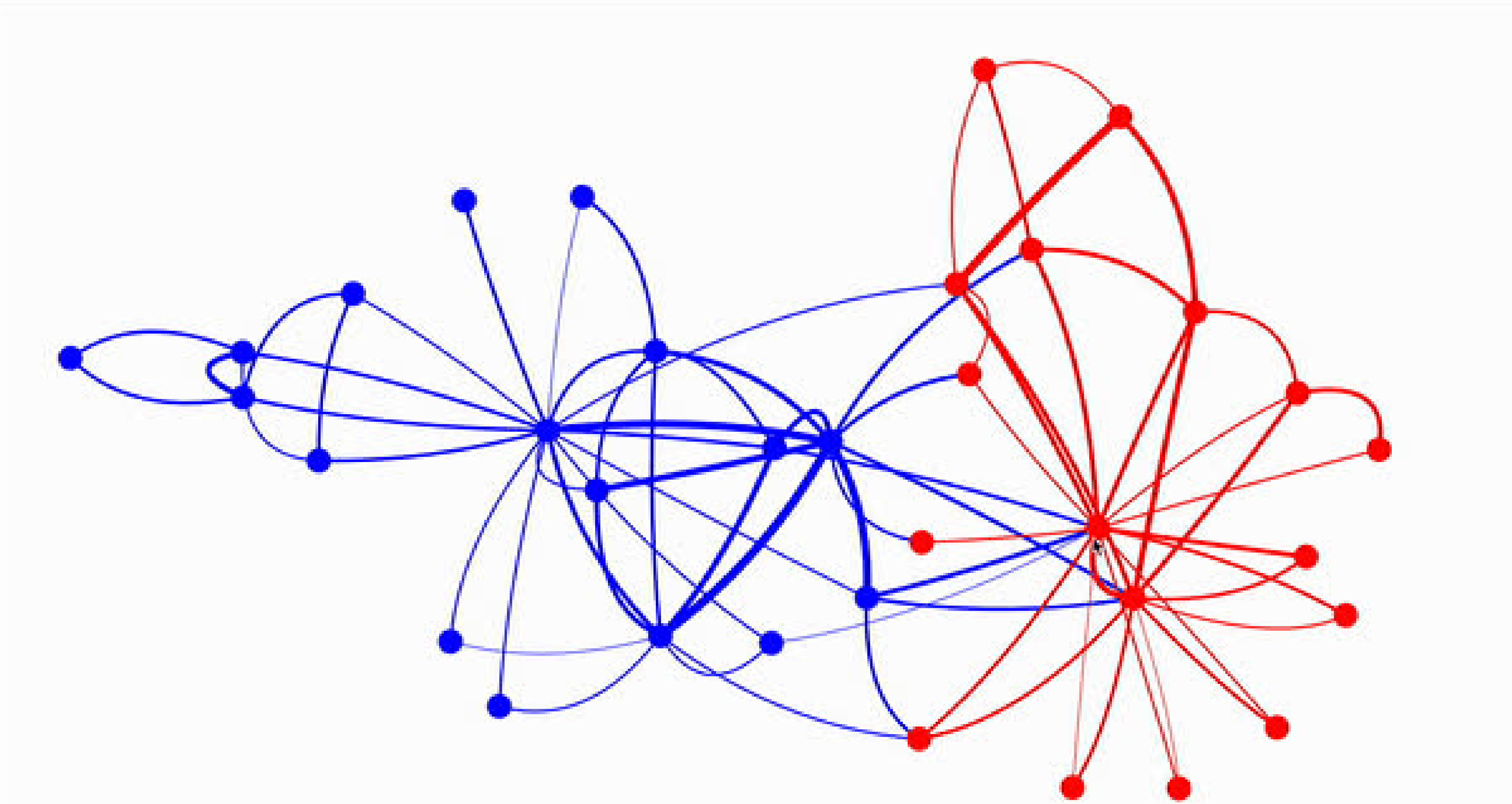


$|E'| = 11$



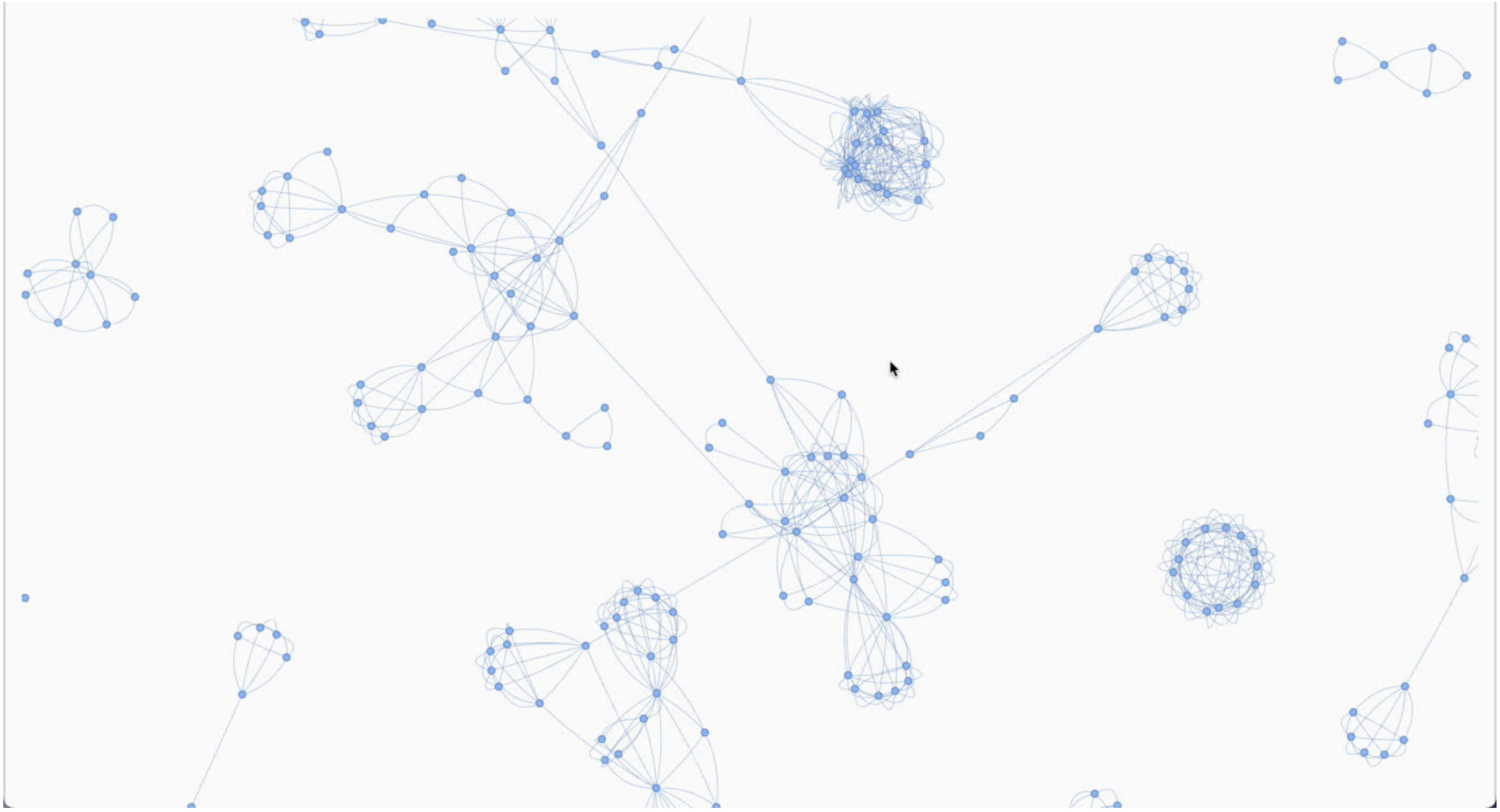
Graphs

Zachary’s Karate Club Network



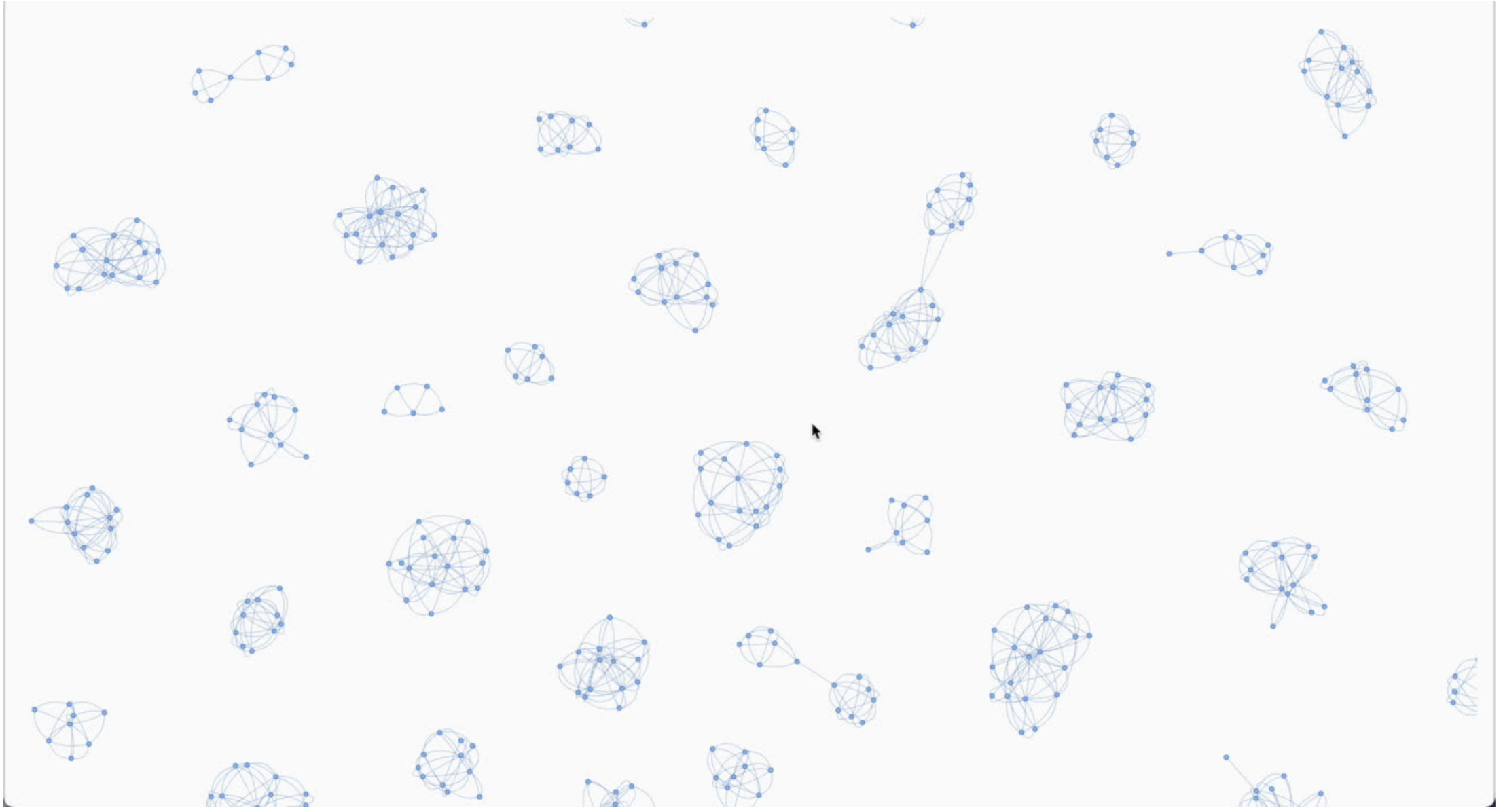
Communities in Networks

DBLP Network



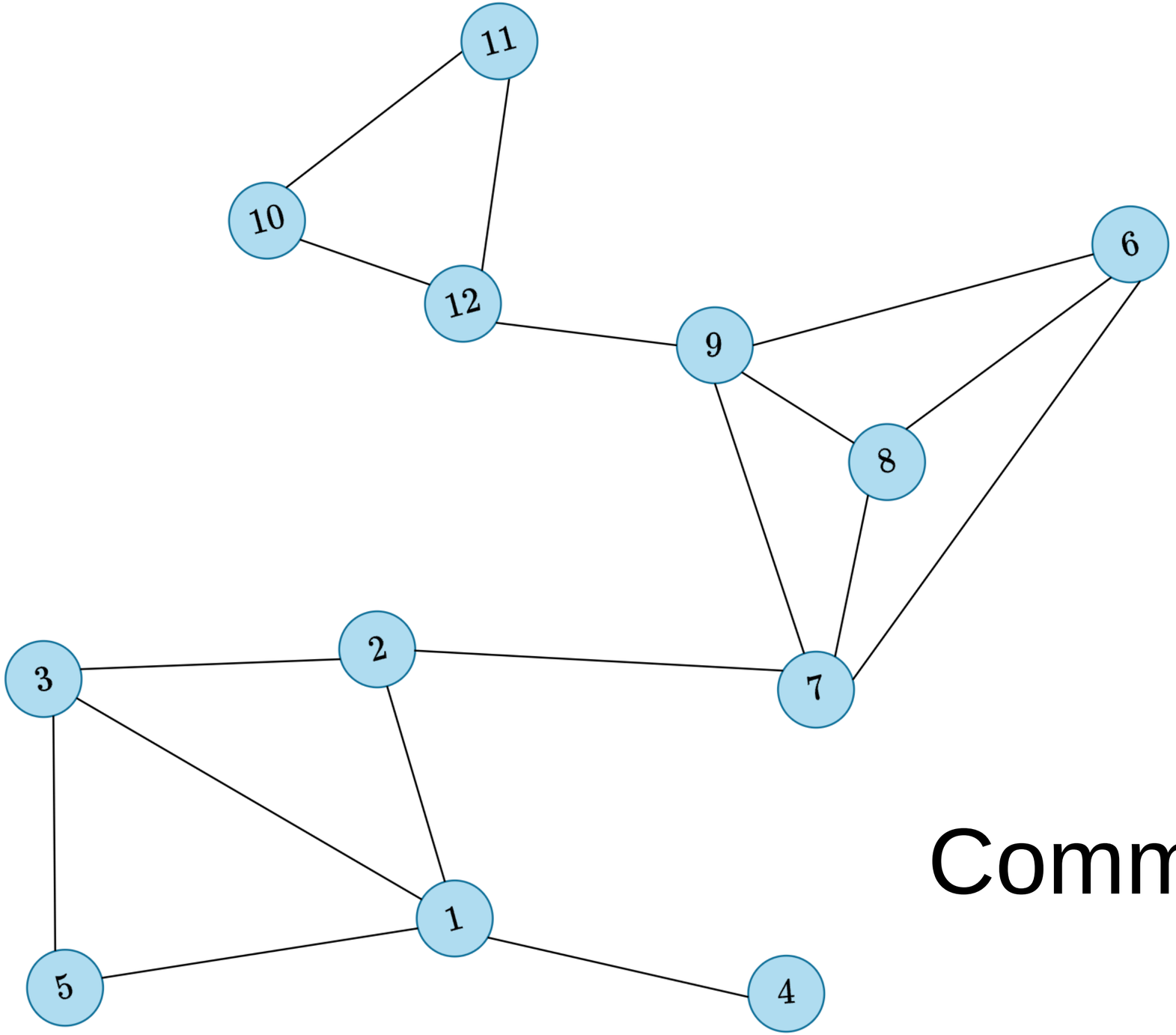
Communities in Networks

Amazon Co-Purchase



$$|V| = 12$$

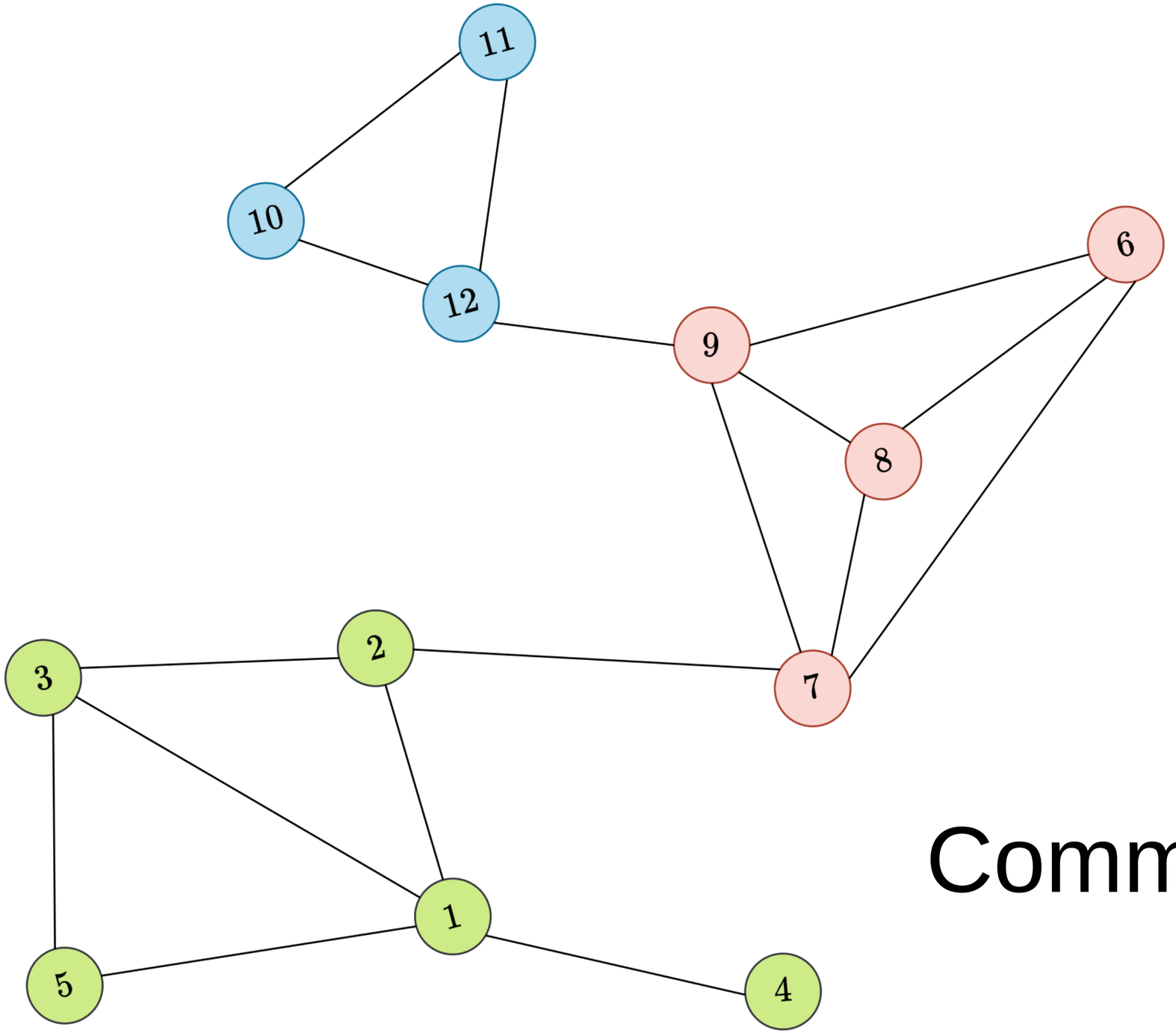
$$|E| = 15$$



Communities

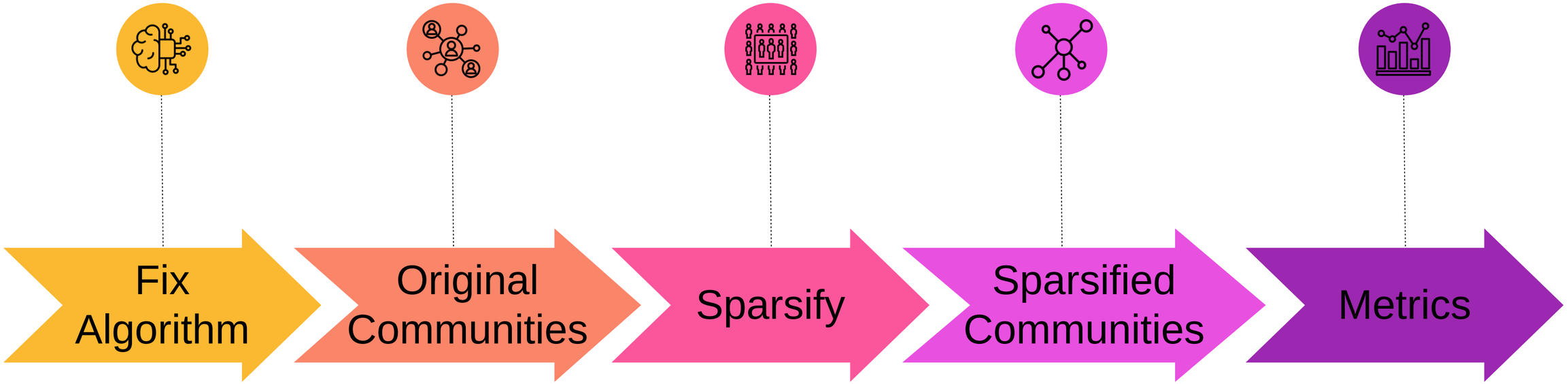
$$|V| = 12$$

$$|E| = 15$$



Communities

Evaluation Pipeline



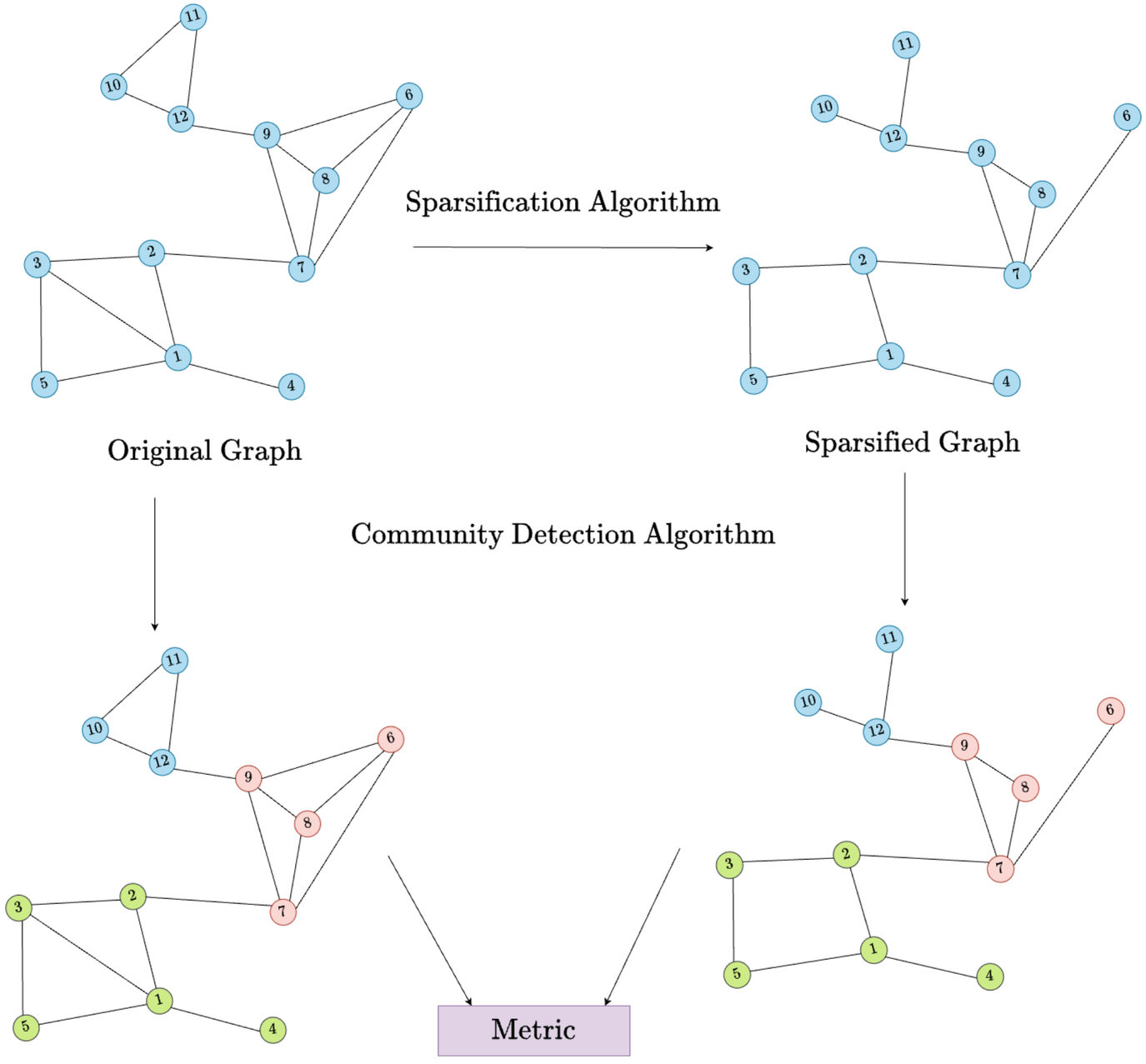
Select a Community Detection Algorithm

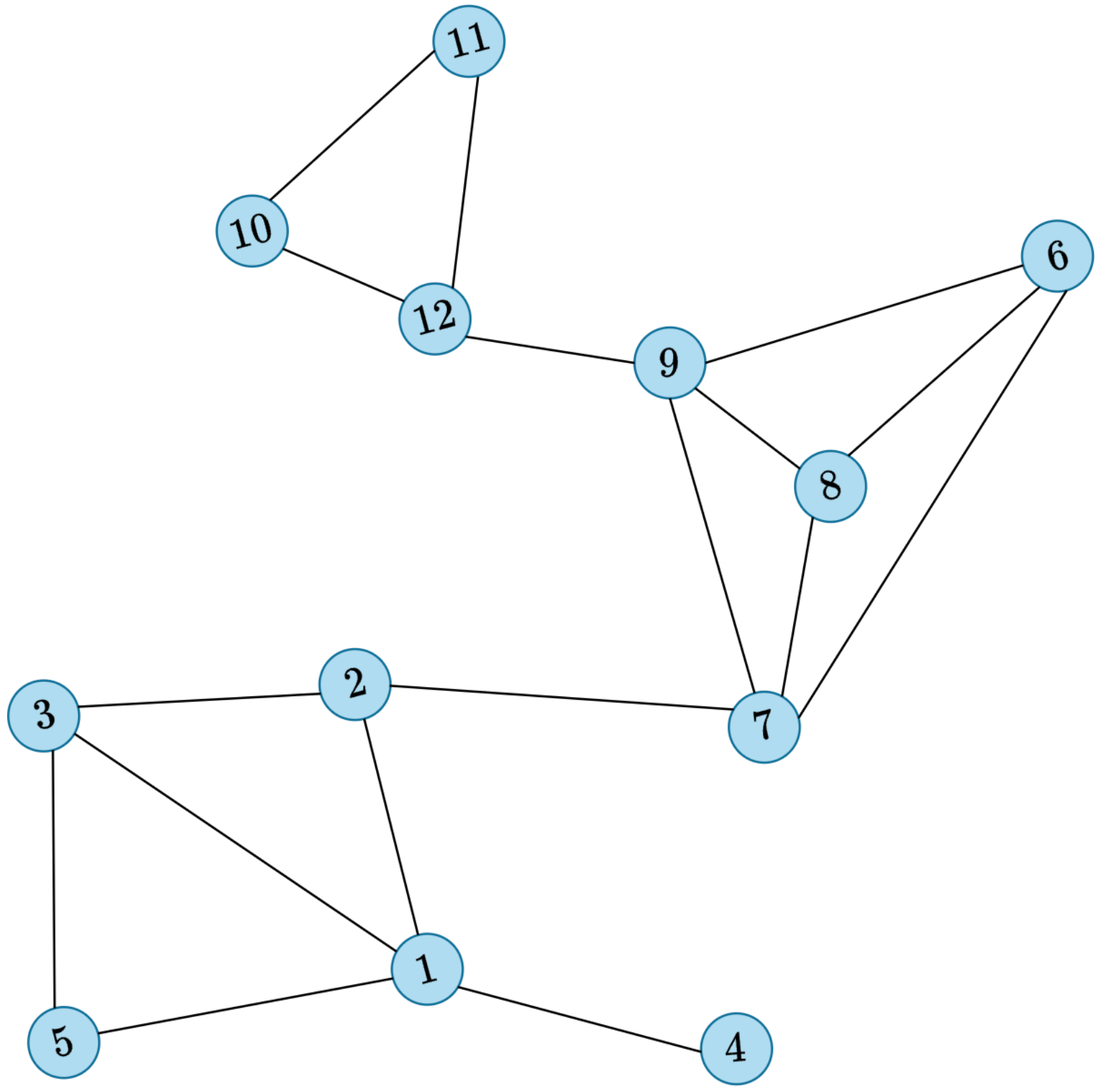
Find the communities in the original graph using the algorithm in hand

Sparsify the Network using various Sampling Methods

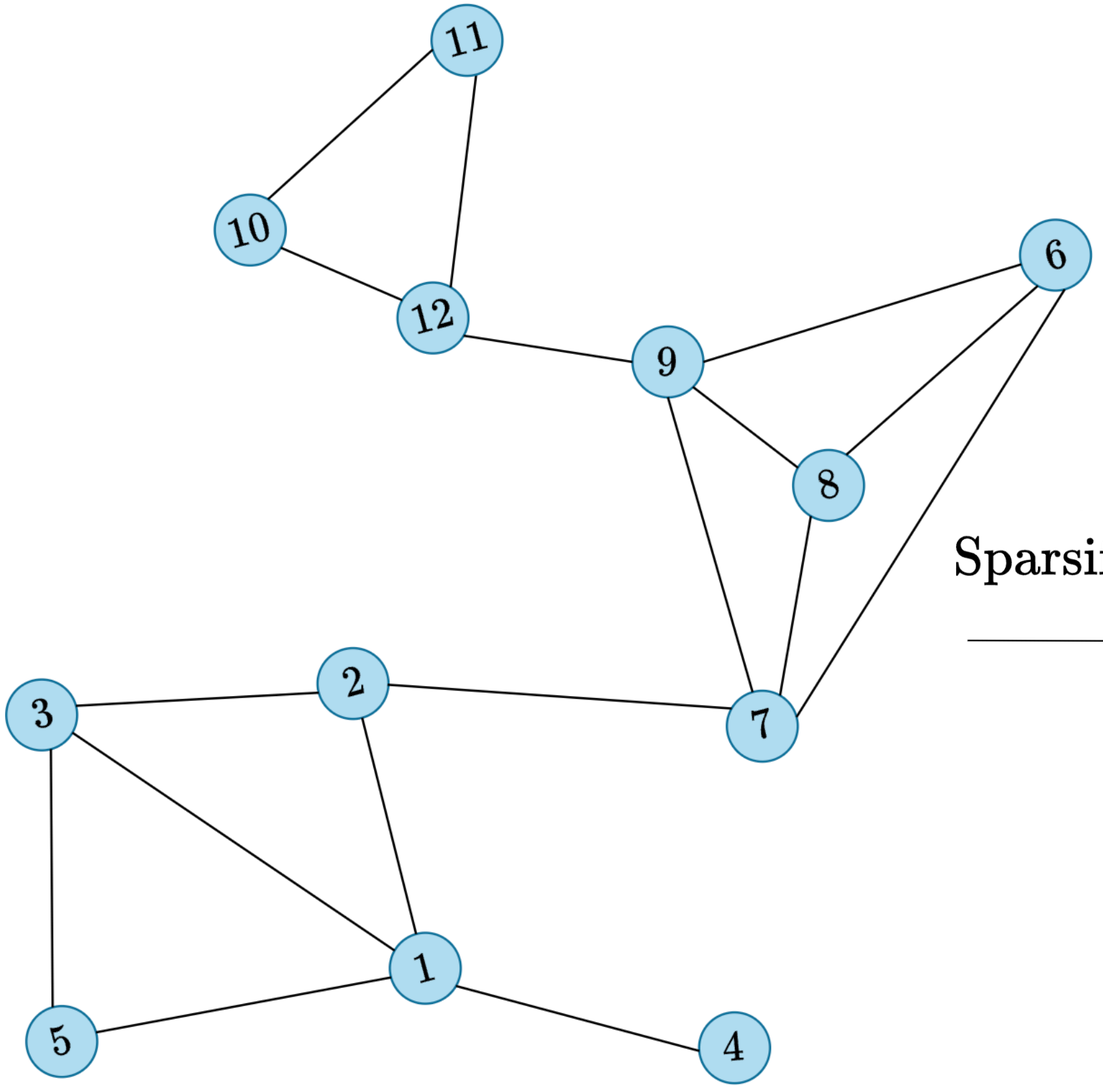
Find the communities in the sparsified network

Evaluate the quality of Communities using certain metrics


Sparsification Algorithm
Original Graph
Sparsified Graph
Community Detection Algorithm
Metric
Original Graph Communities
Sparsified Graph Communities

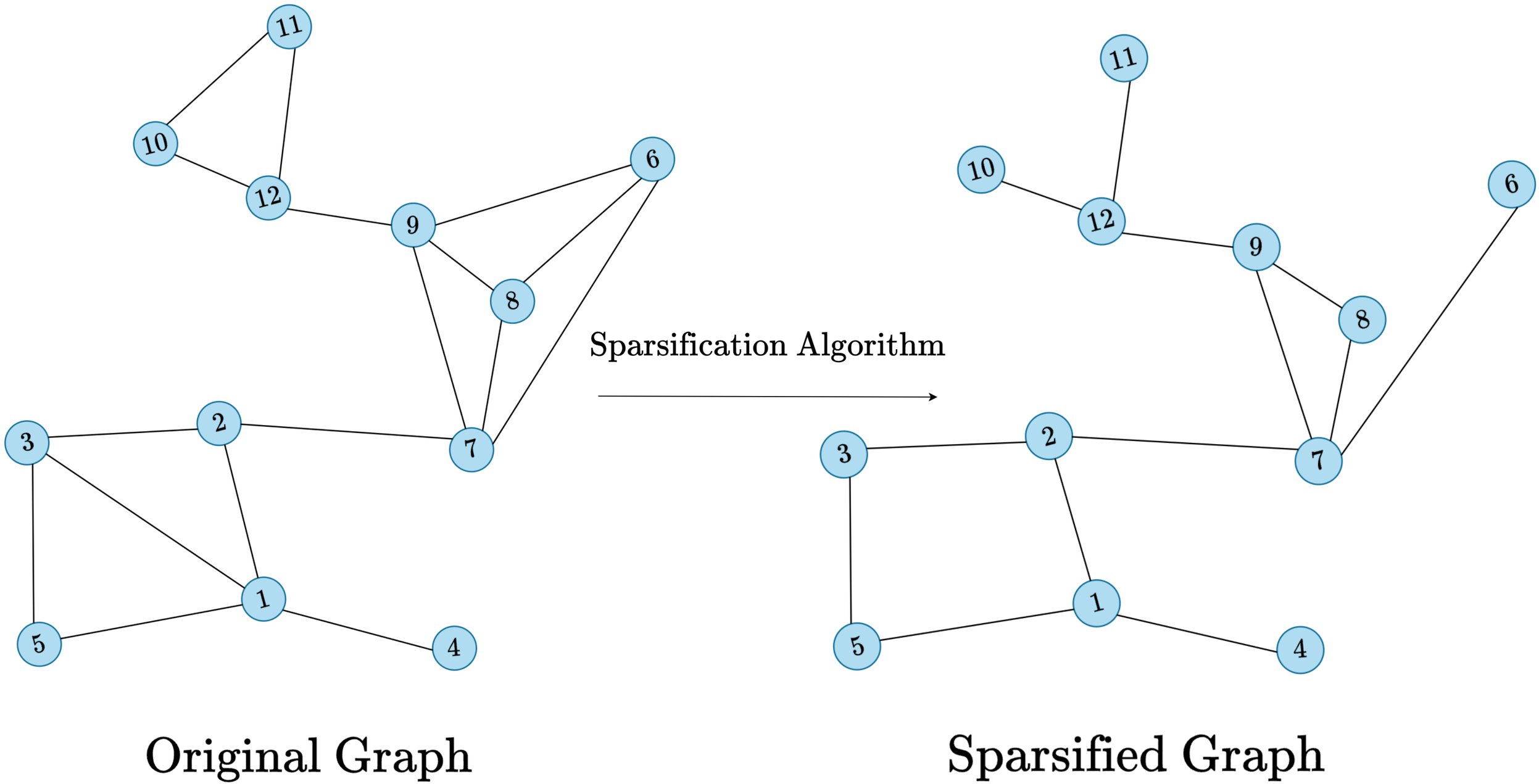


Original Graph



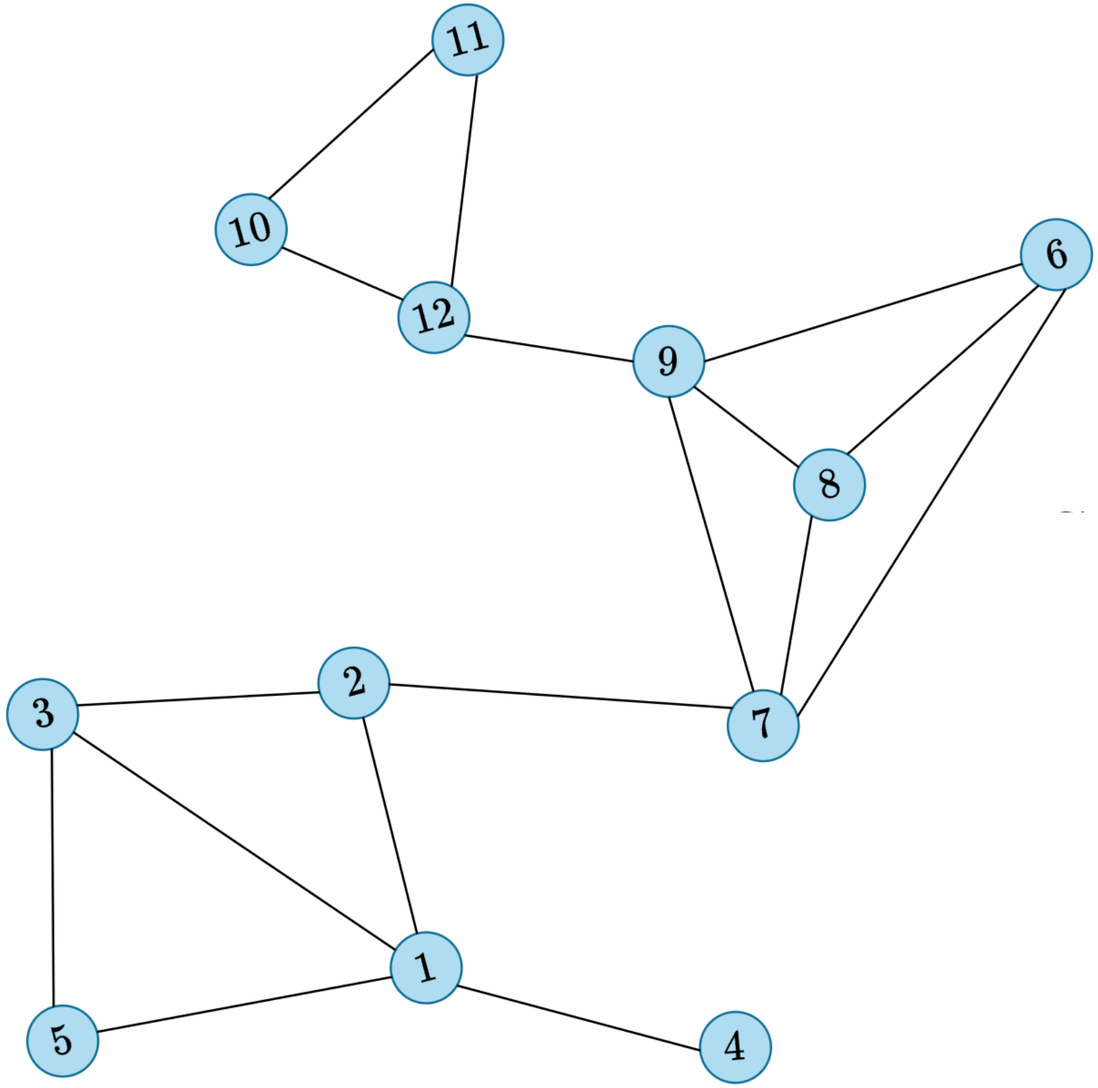
Sparsification Algorithm

Original Graph

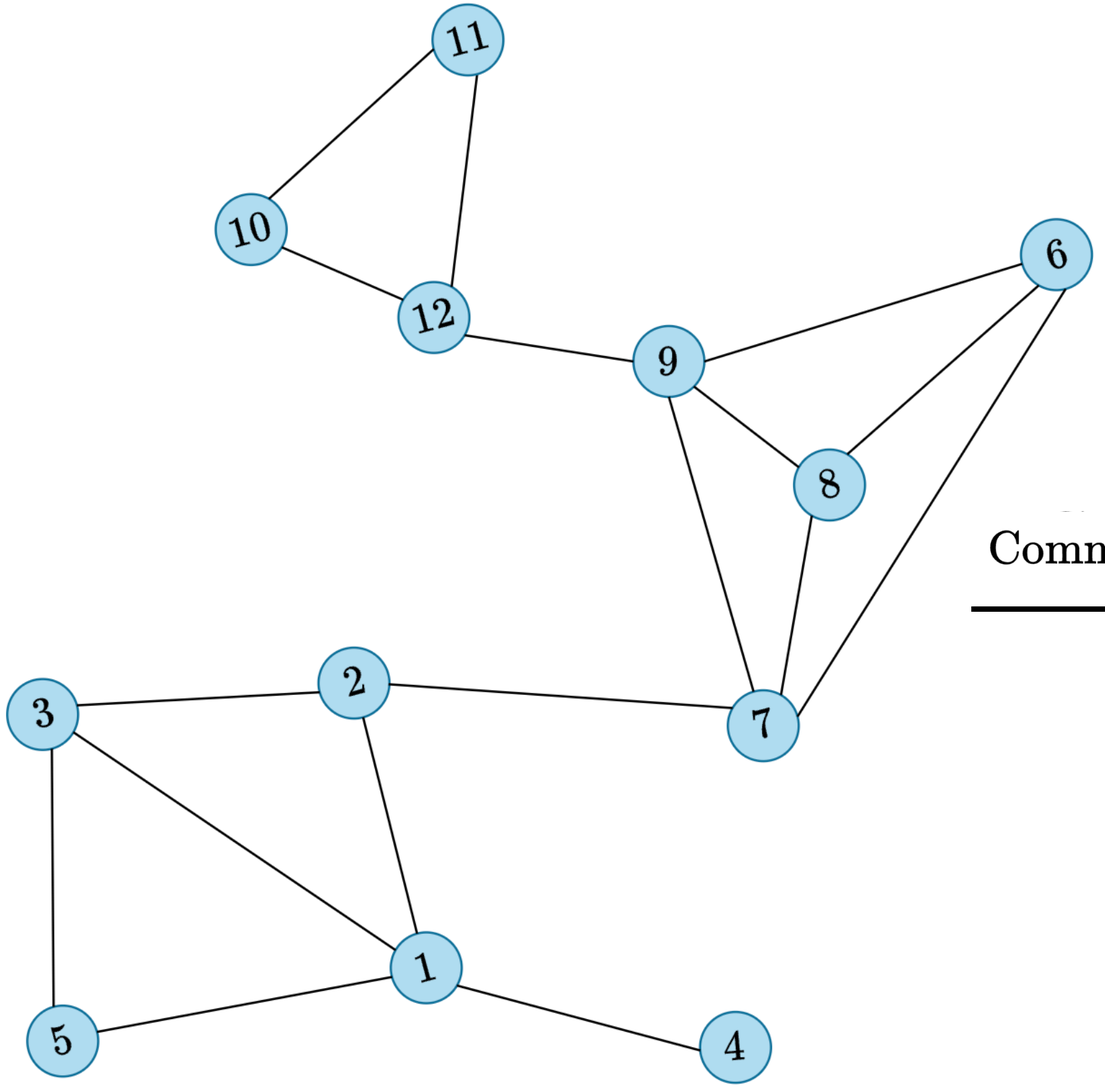


Original Graph

Sparsified Graph

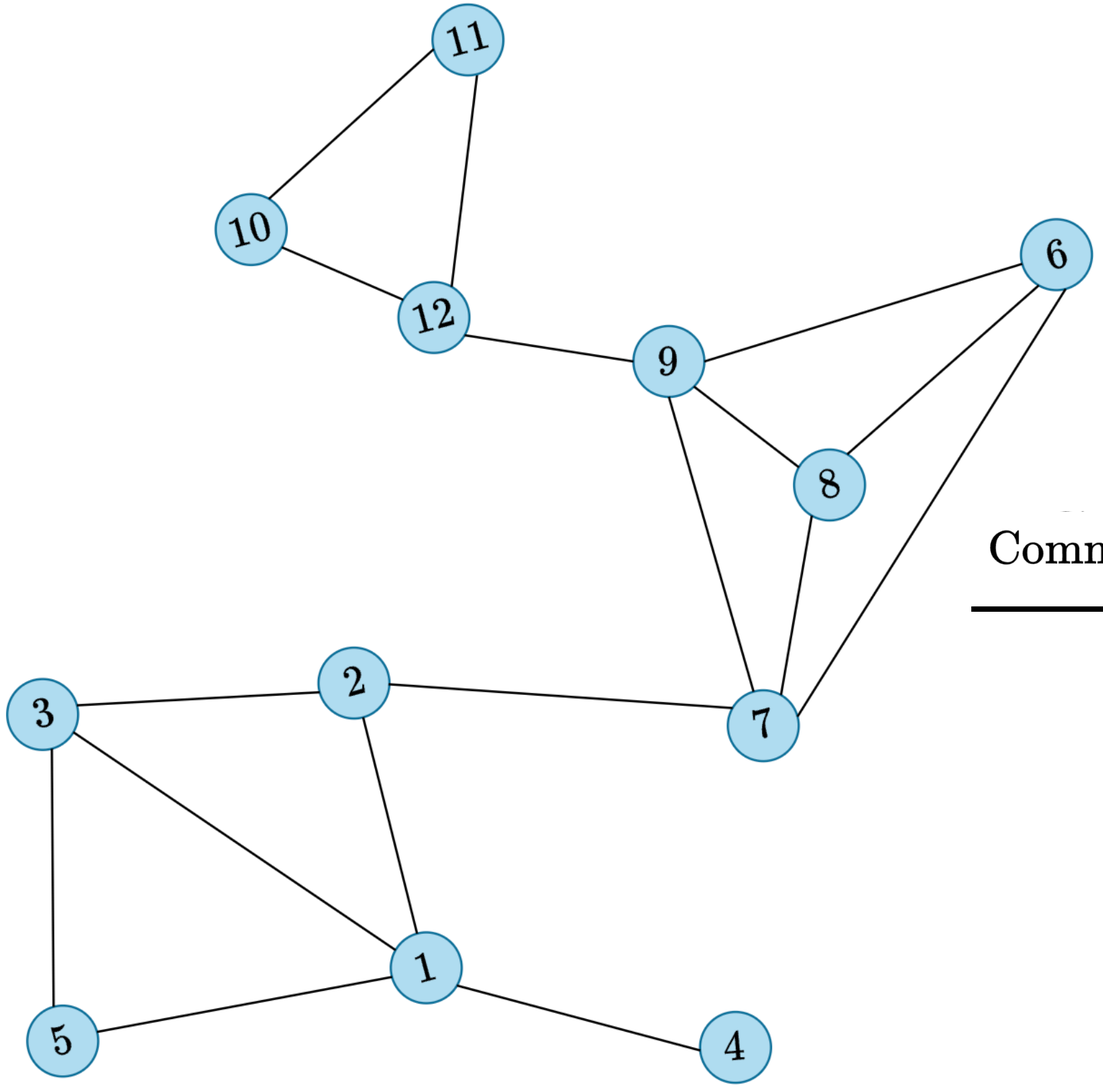


Original Graph

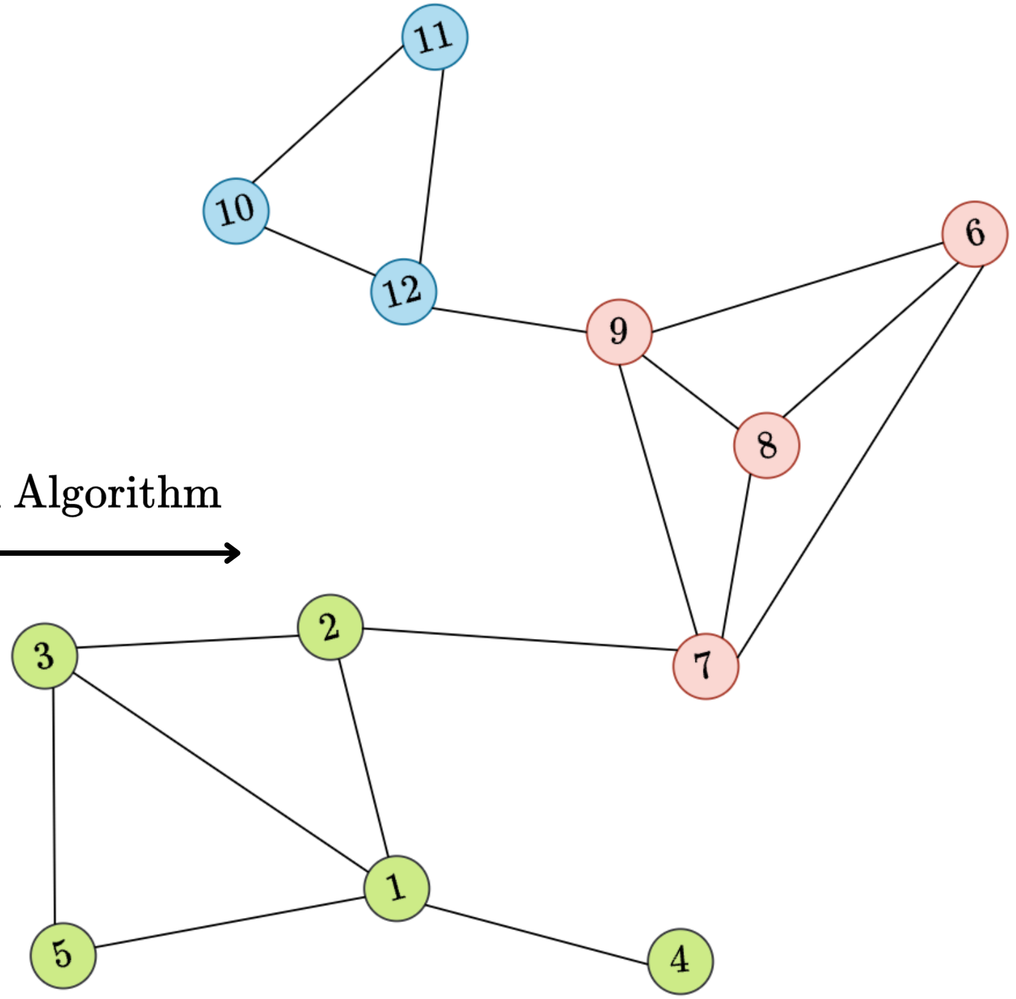


Community Detection Algorithm

Original Graph

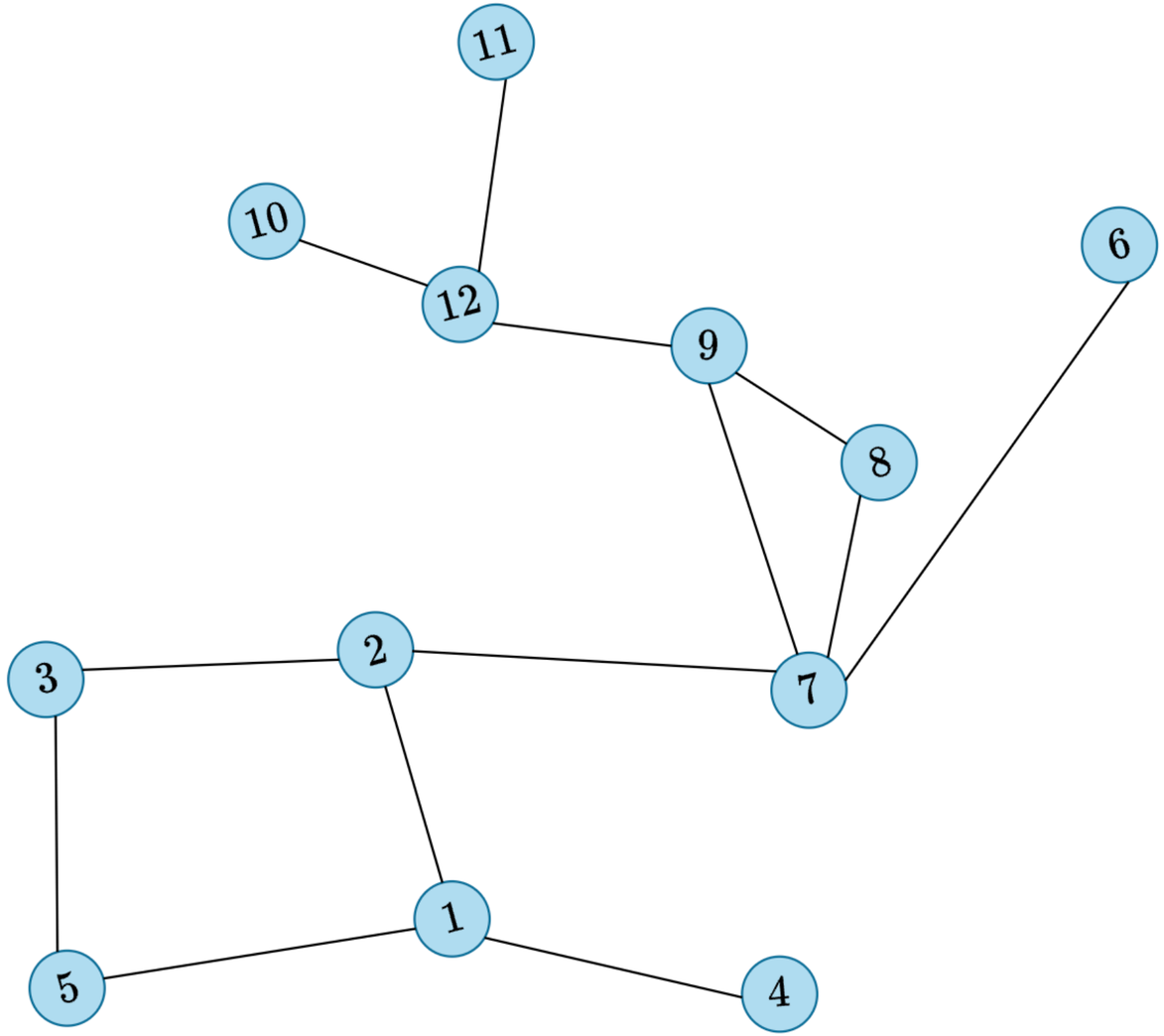


Community Detection Algorithm

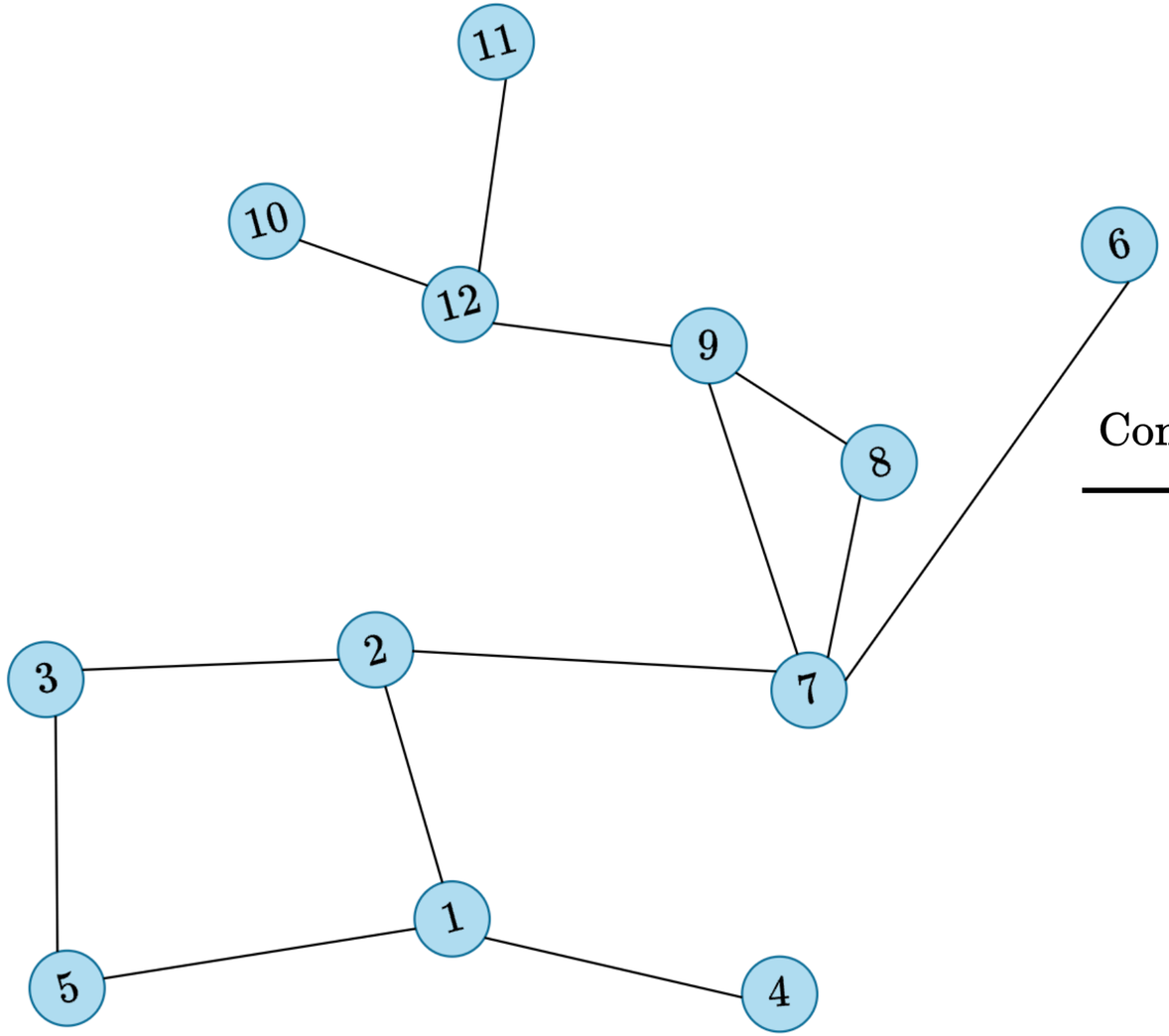


Original Graph

Original Graph Communities

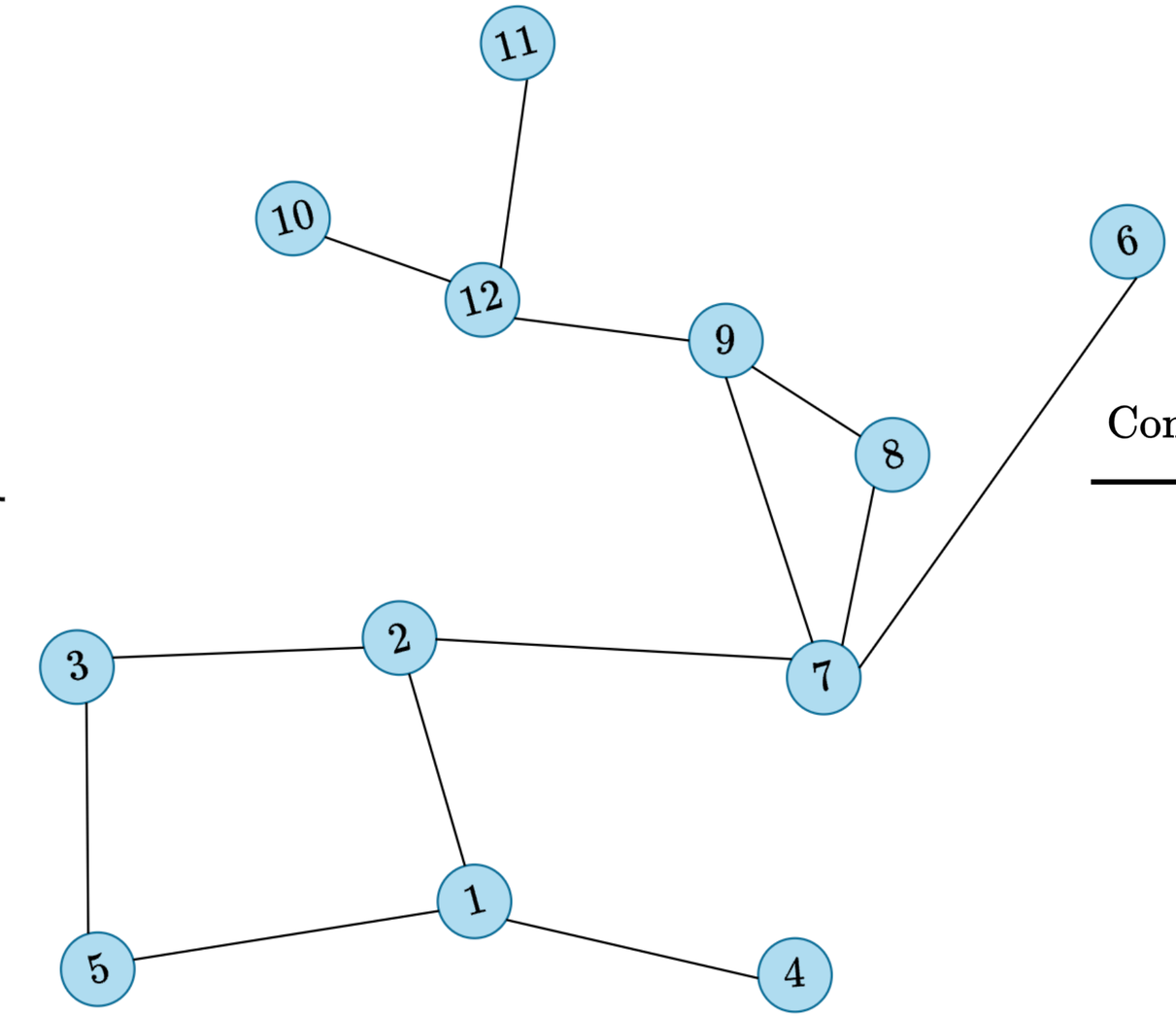


Sparsified Graph



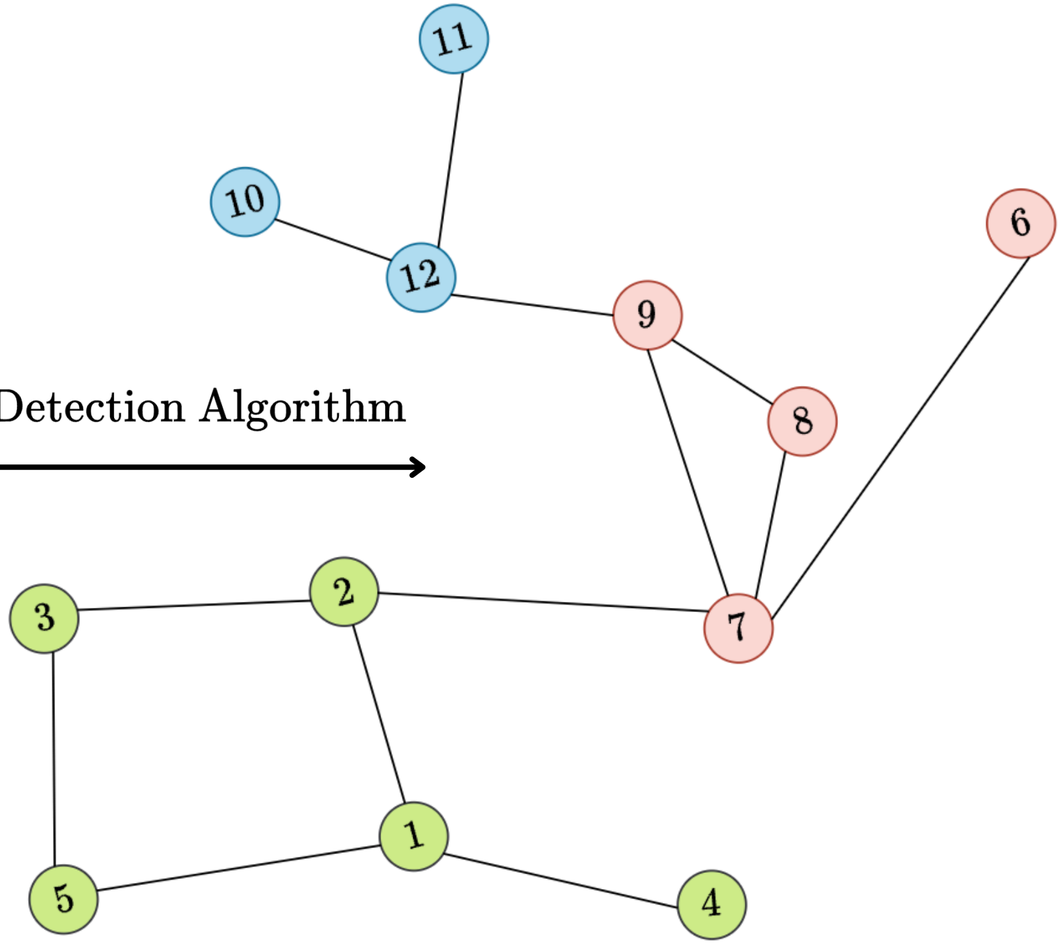
Community Detection Algorithm

Sparsified Graph

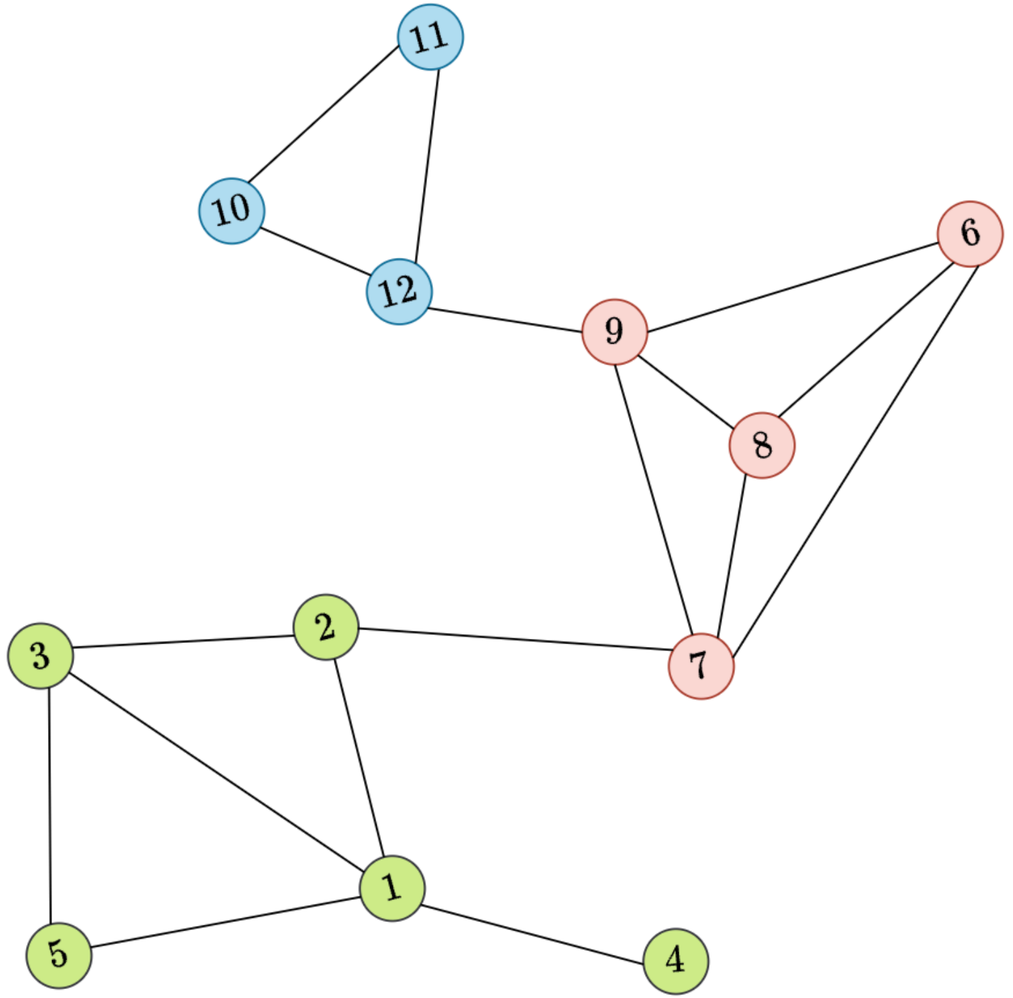


Sparsified Graph

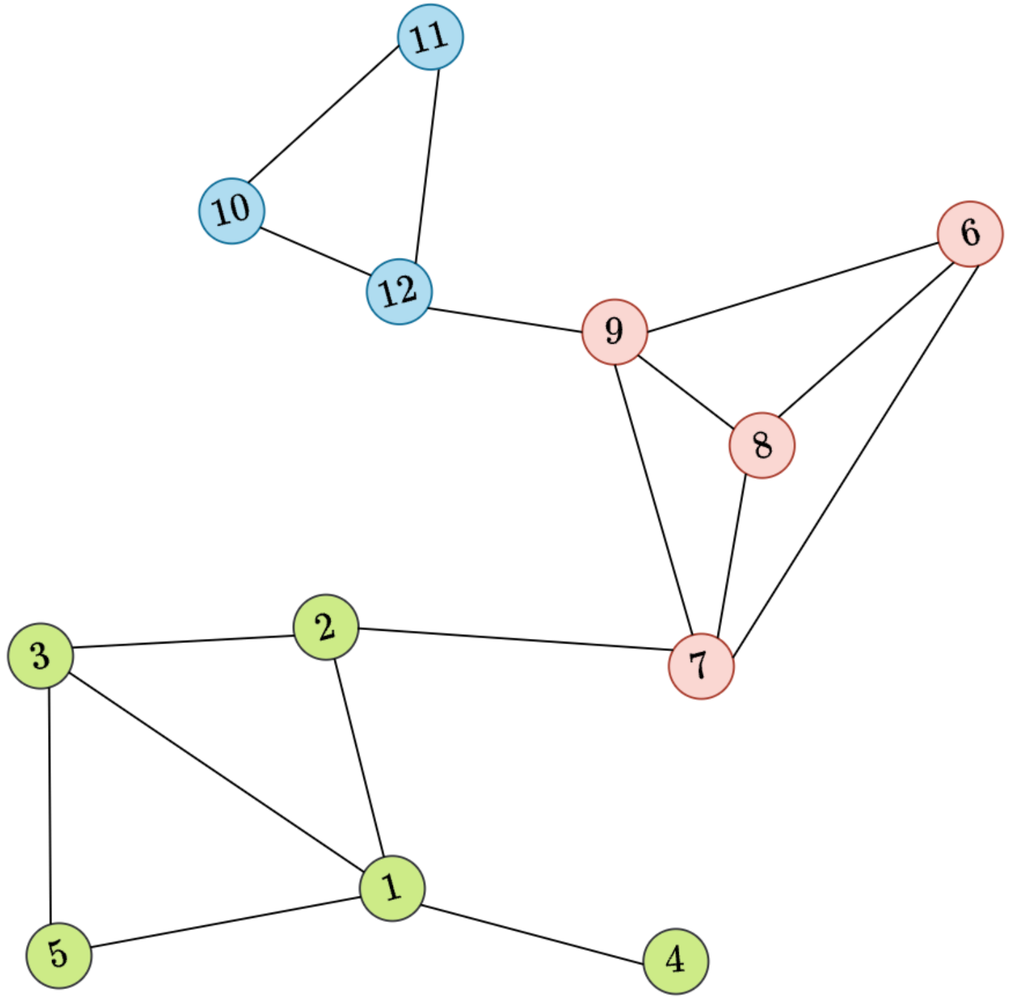
Community Detection Algorithm



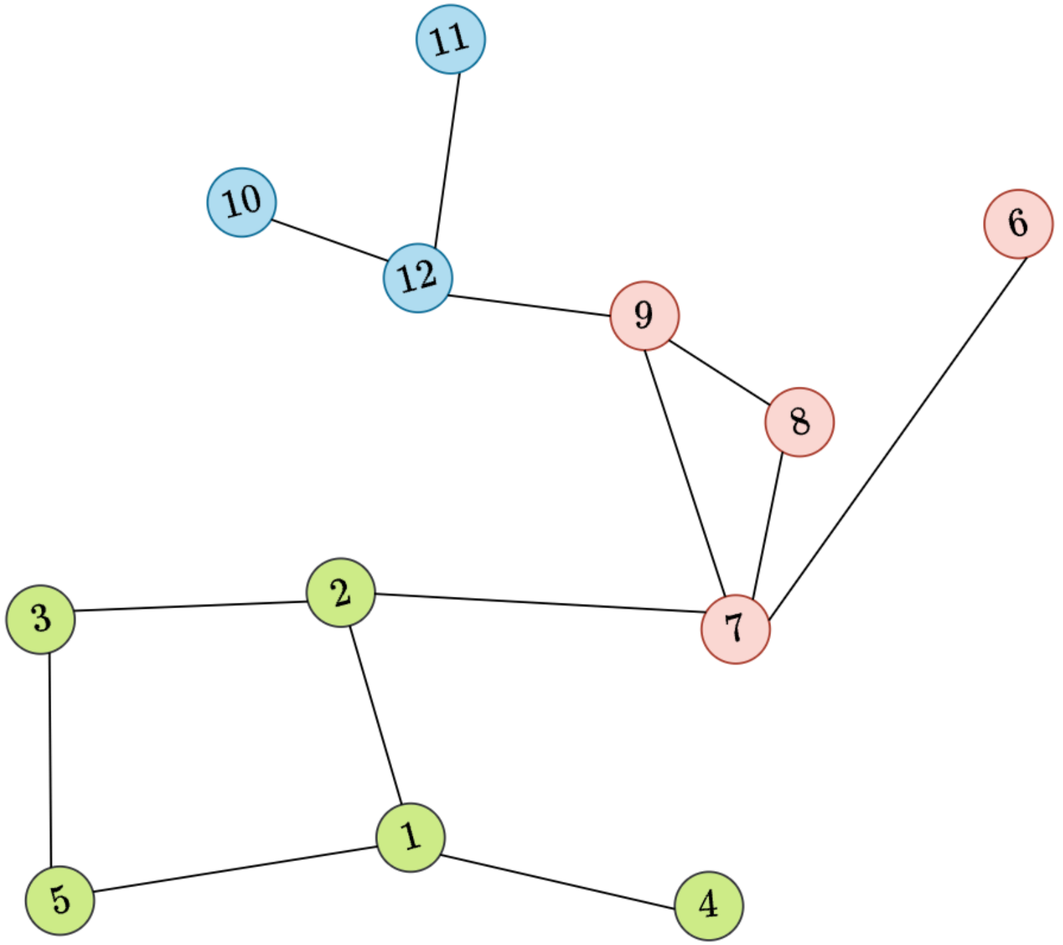
Sparsified Graph Communities



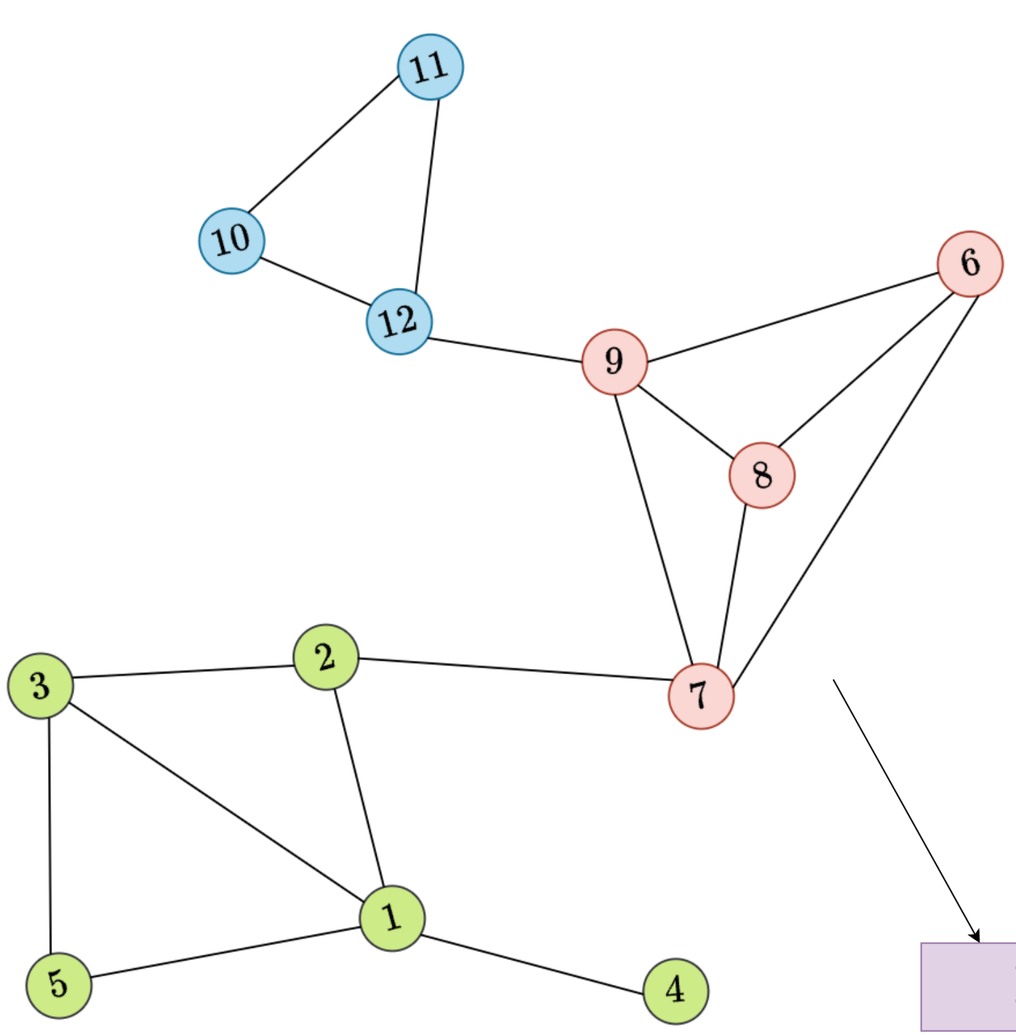
Original Graph Communities



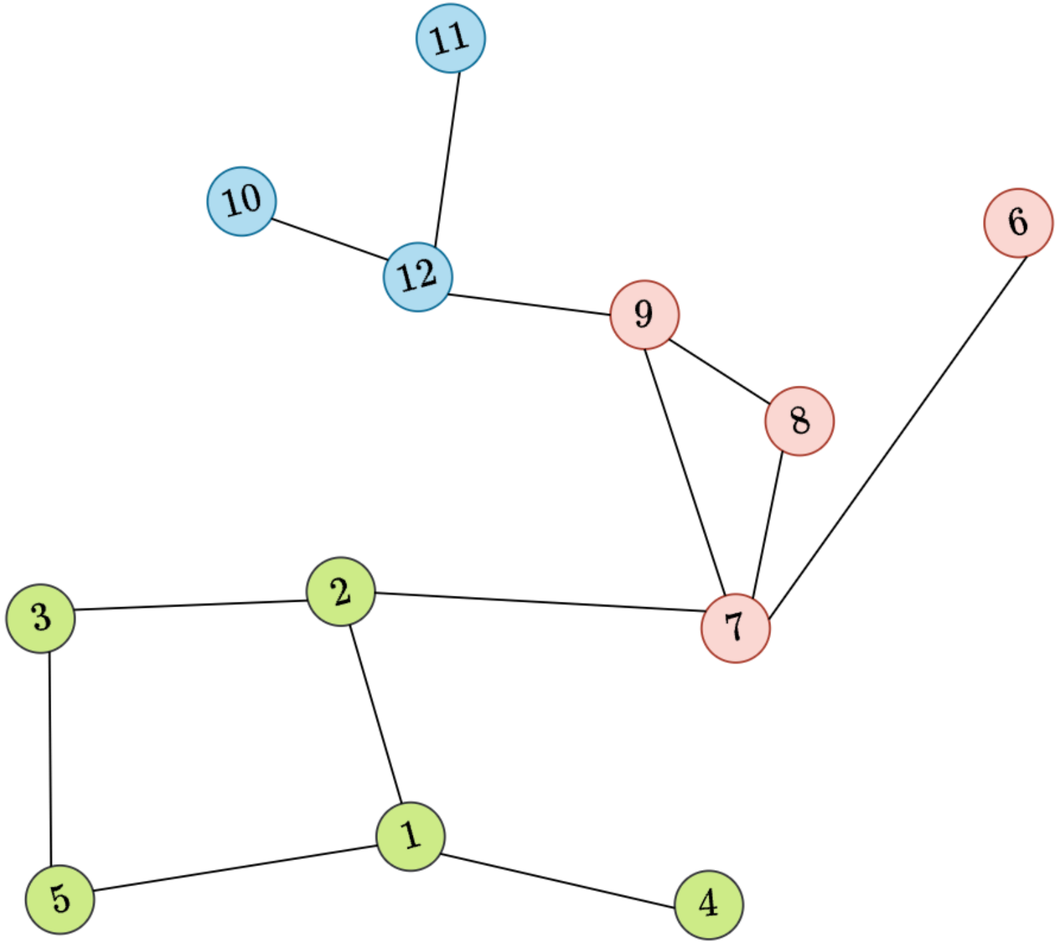
Original Graph Communities



Sparsified Graph Communities



Metric

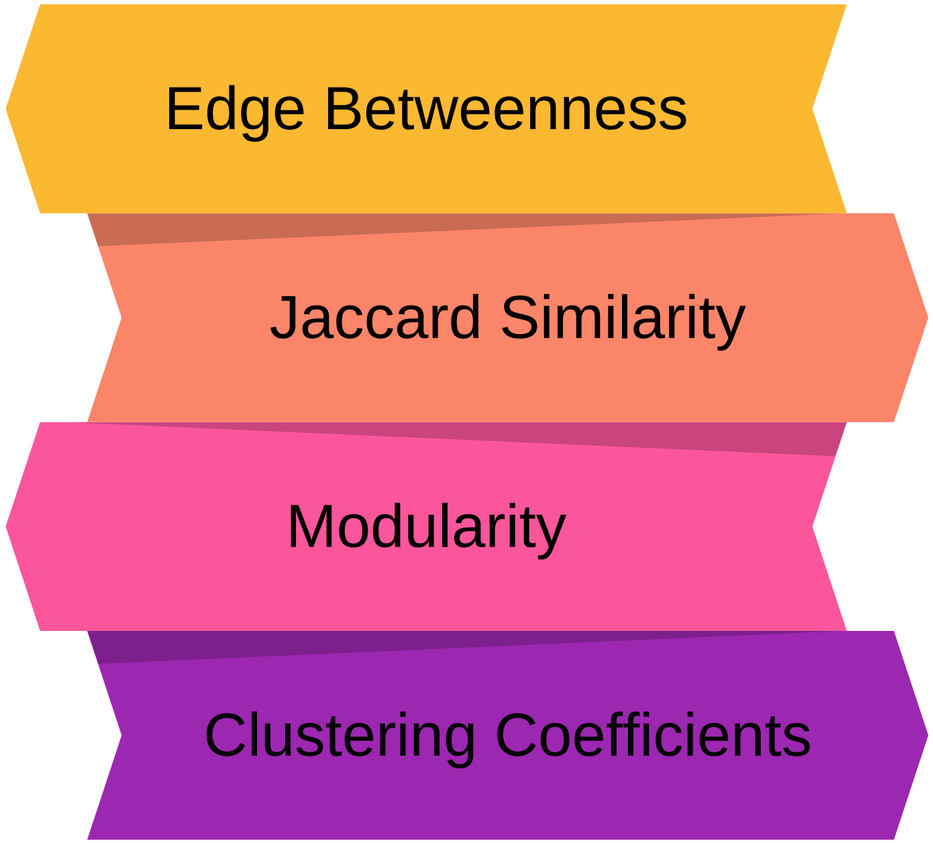


Original Graph Communities

Sparsified Graph Communities

Salient Community Structure Properties

The number of shortest paths between pairs of nodes in a graph that pass through a particular edge. It quantifies the importance of an edge in connecting different parts of the network.

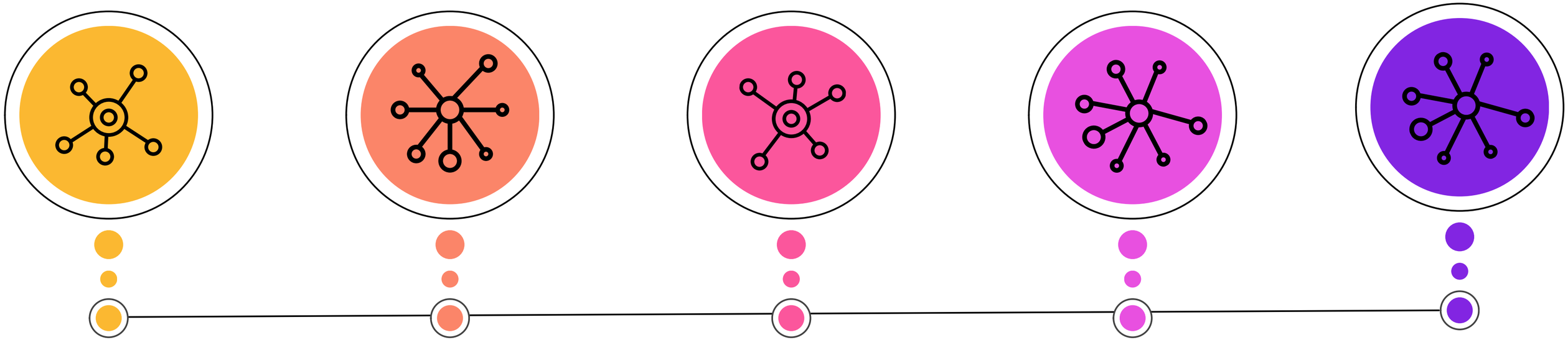


Measures similarity between two sets by comparing their intersection to their union. It is often used to measure the similarity between the sets of neighbors of two nodes.

Quantifies the degree to which a network is partitioned into communities or modules. Compares the number of edges within communities to the number of edges expected in a random network with the same degree distribution, indicating the presence of densely connected communities.

Measures the degree to which nodes in a graph tend to cluster together. It quantifies the likelihood that two neighbors of a node are connected, providing insights into the local clustering structure of the network.

Sparsification Methods



Random Sampling

Edge Betweenness Based

Global Similarity Based

Local Similarity Based

Clustering Coefficient Based

+ some more !

Edge Betweenness

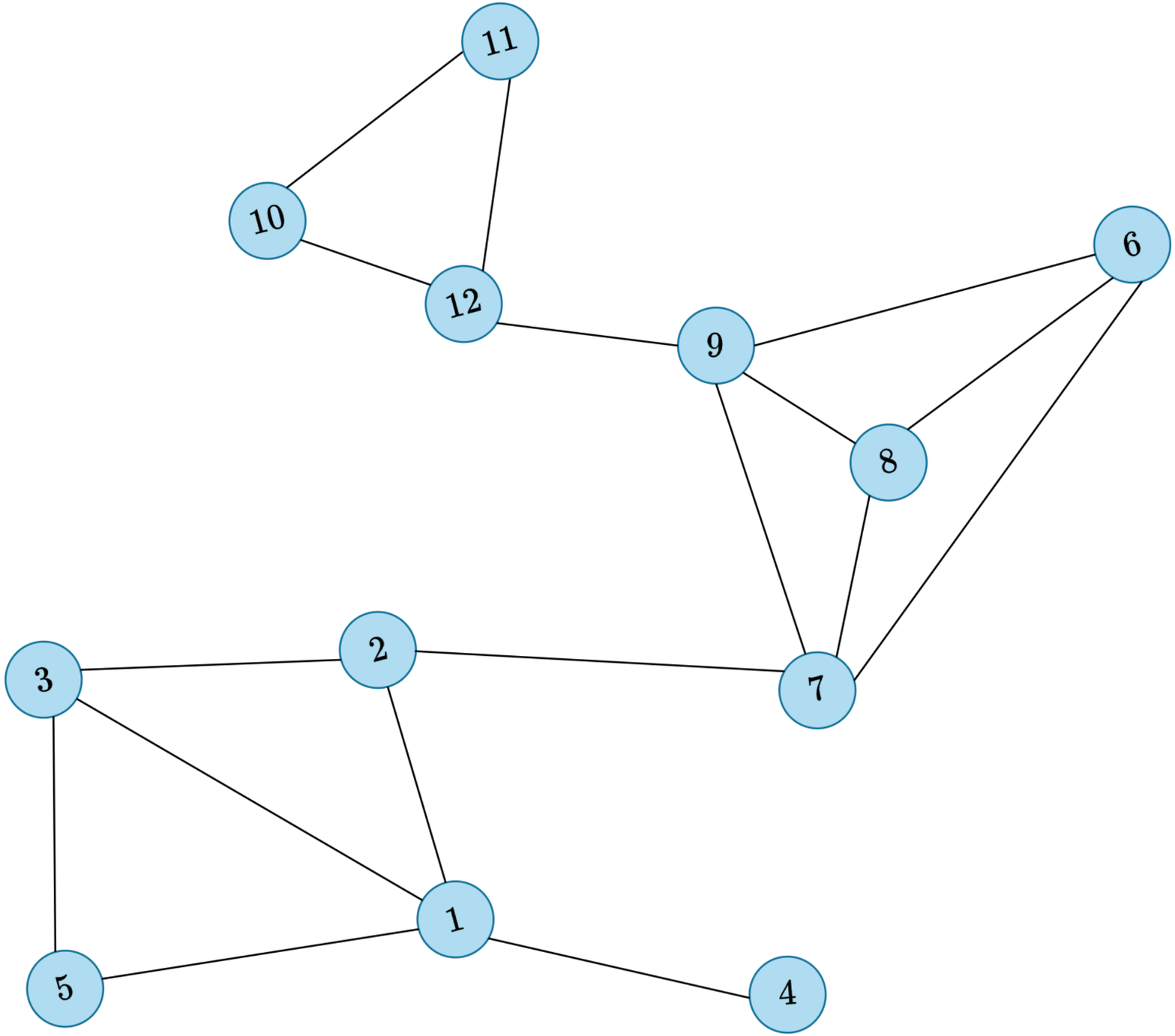
$$c_B(e) = \sum_{s,t \in V} \frac{\sigma(s,t|e)}{\sigma(s,t)}$$

$\longrightarrow$ No. of shortest (s,t) -paths crossing e

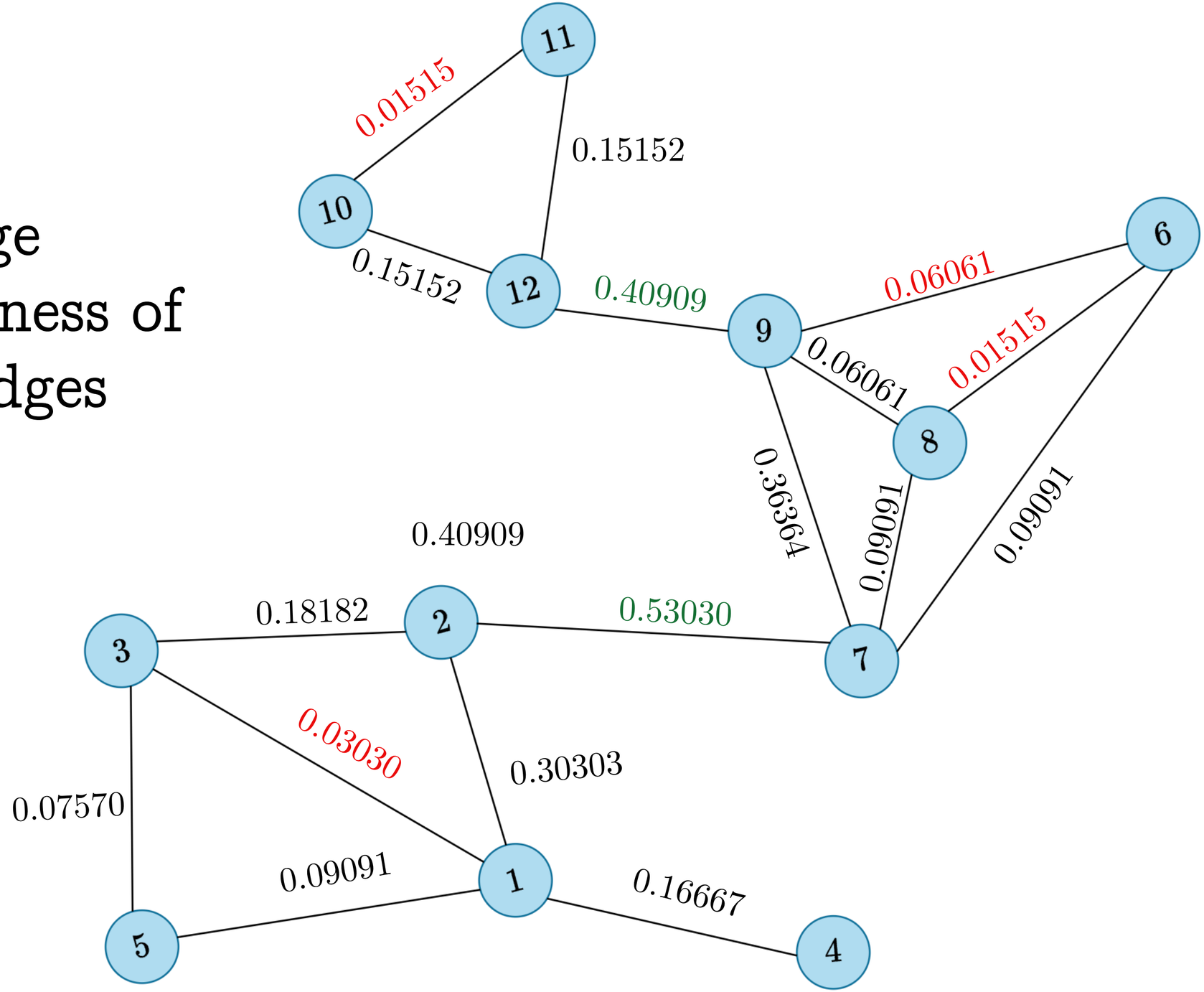
$\longrightarrow$ Total No. of shortest (s,t) -paths

Time Complexity : $O(|V|.|E|)$

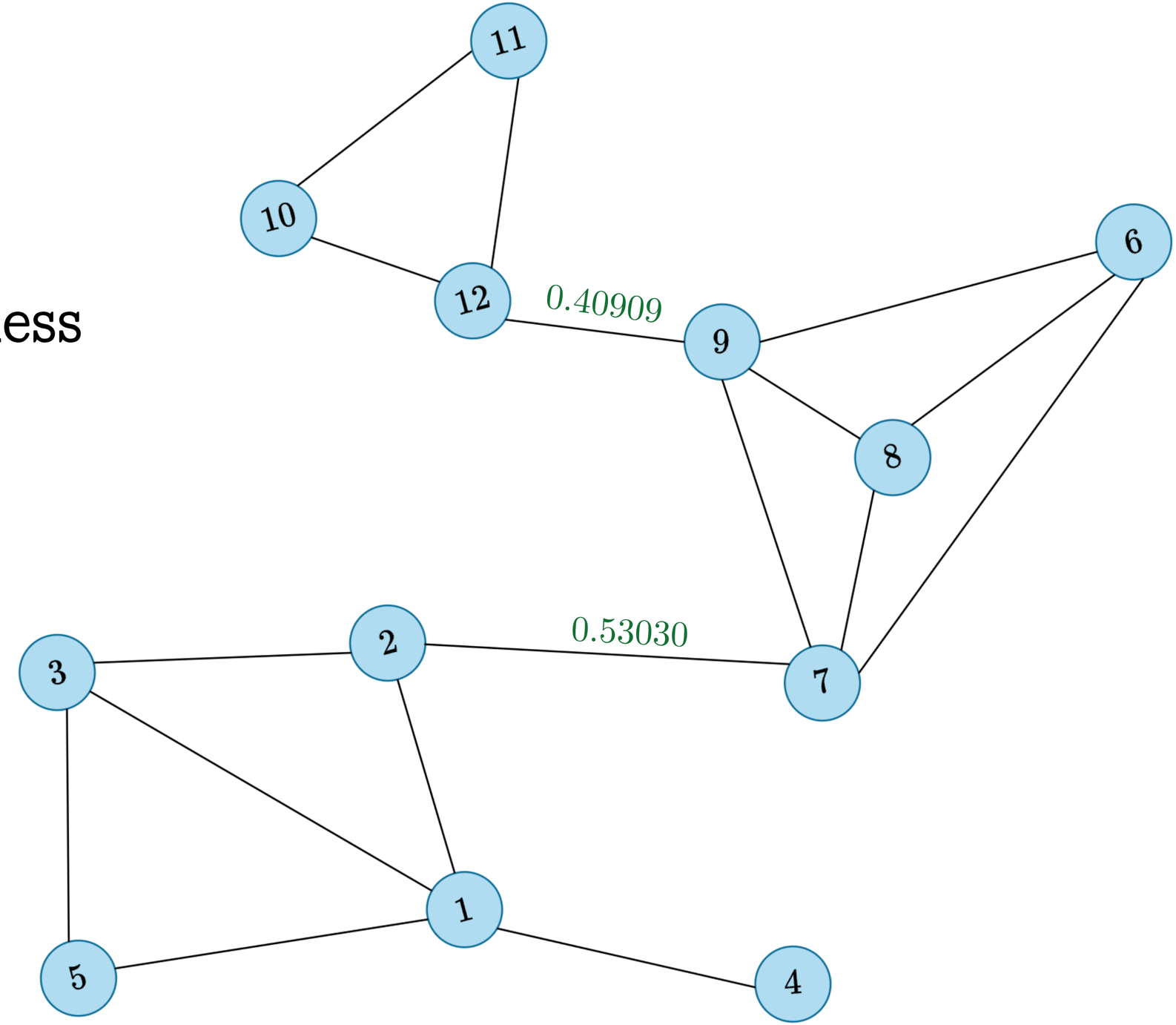
Space Complexity : $O(|V|+|E|)$

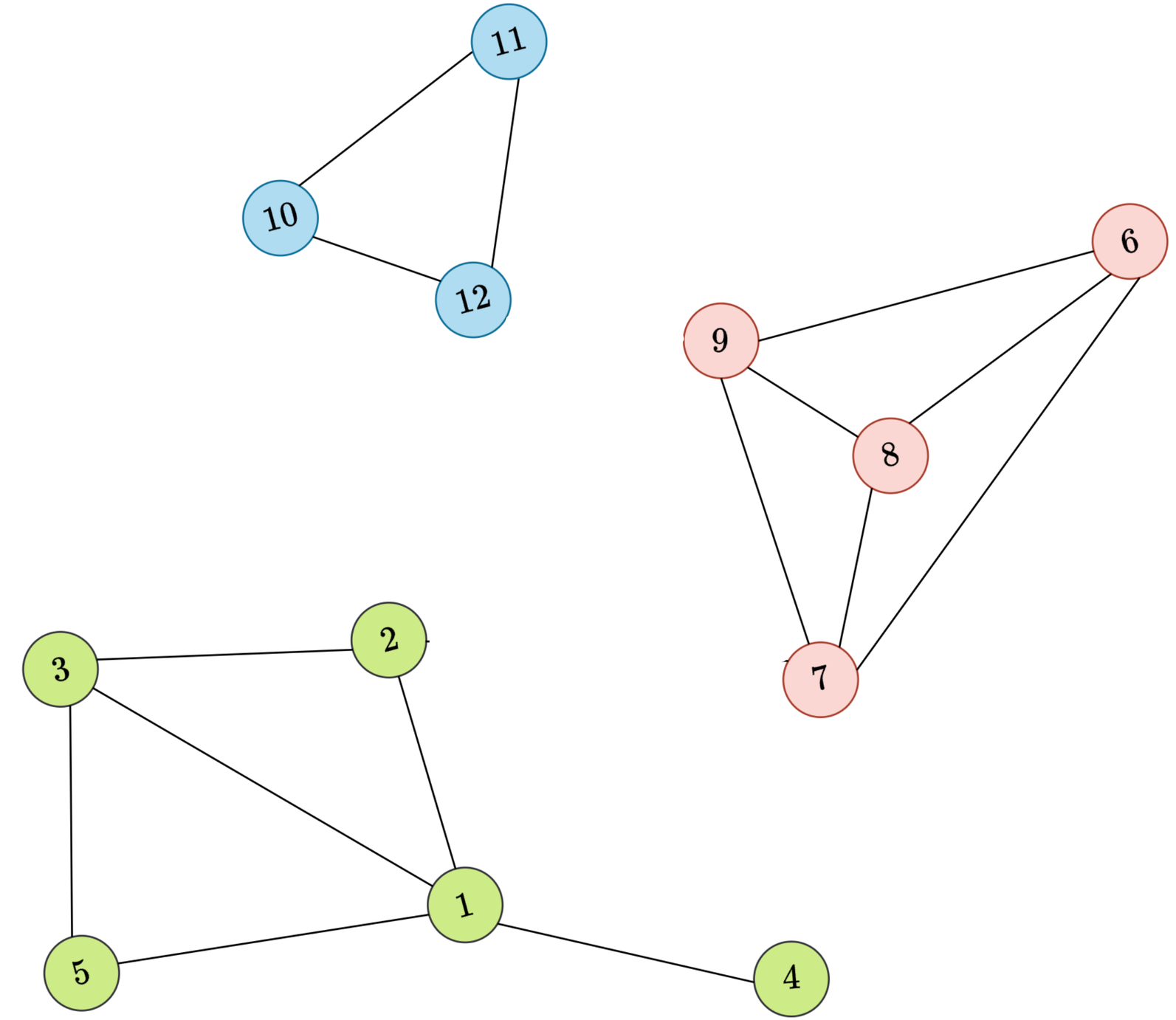

11
10
12
6
9
8
2
3
7
1
5
4

Edge Betweenness of the Edges

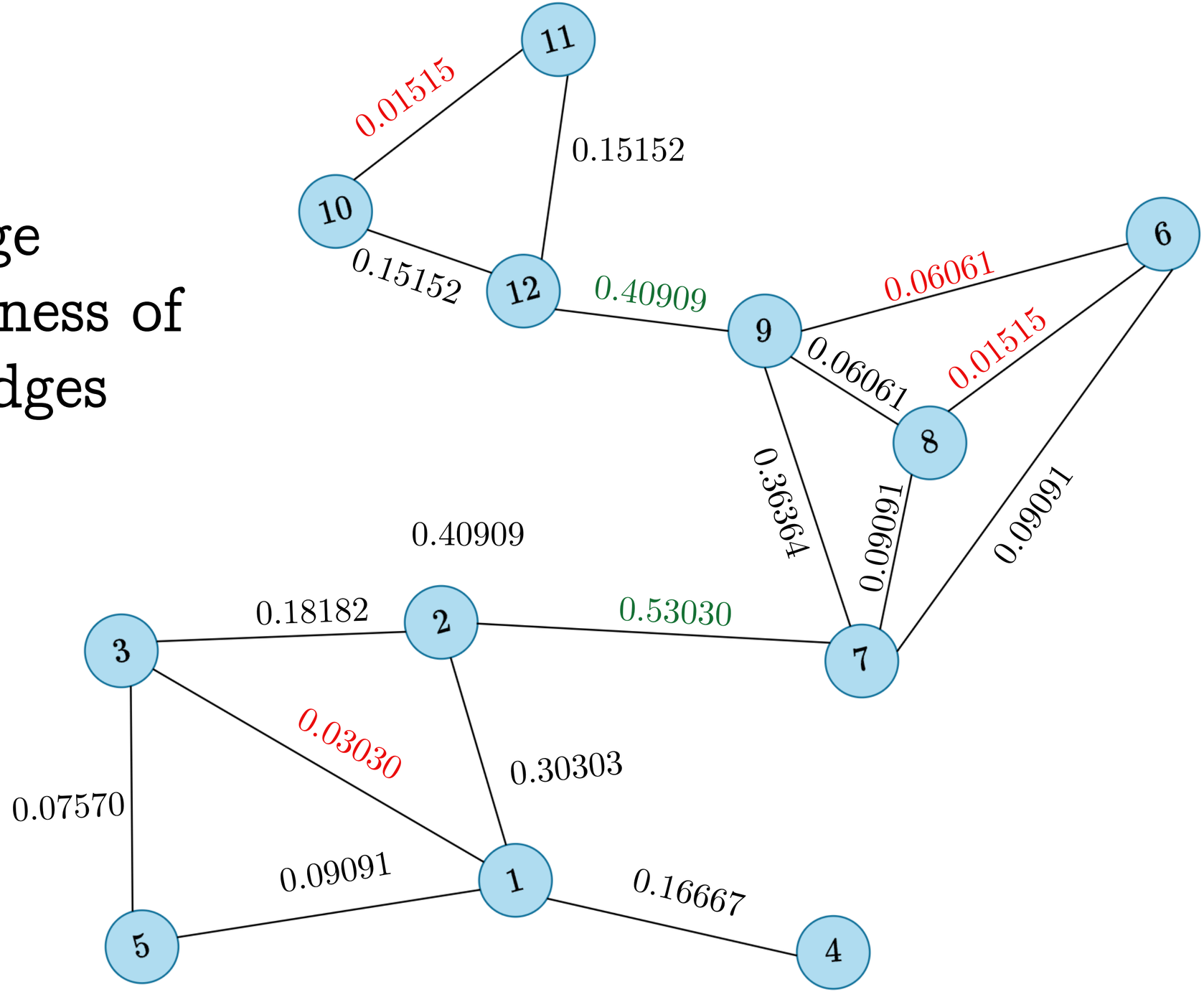


High Betweenness Edges

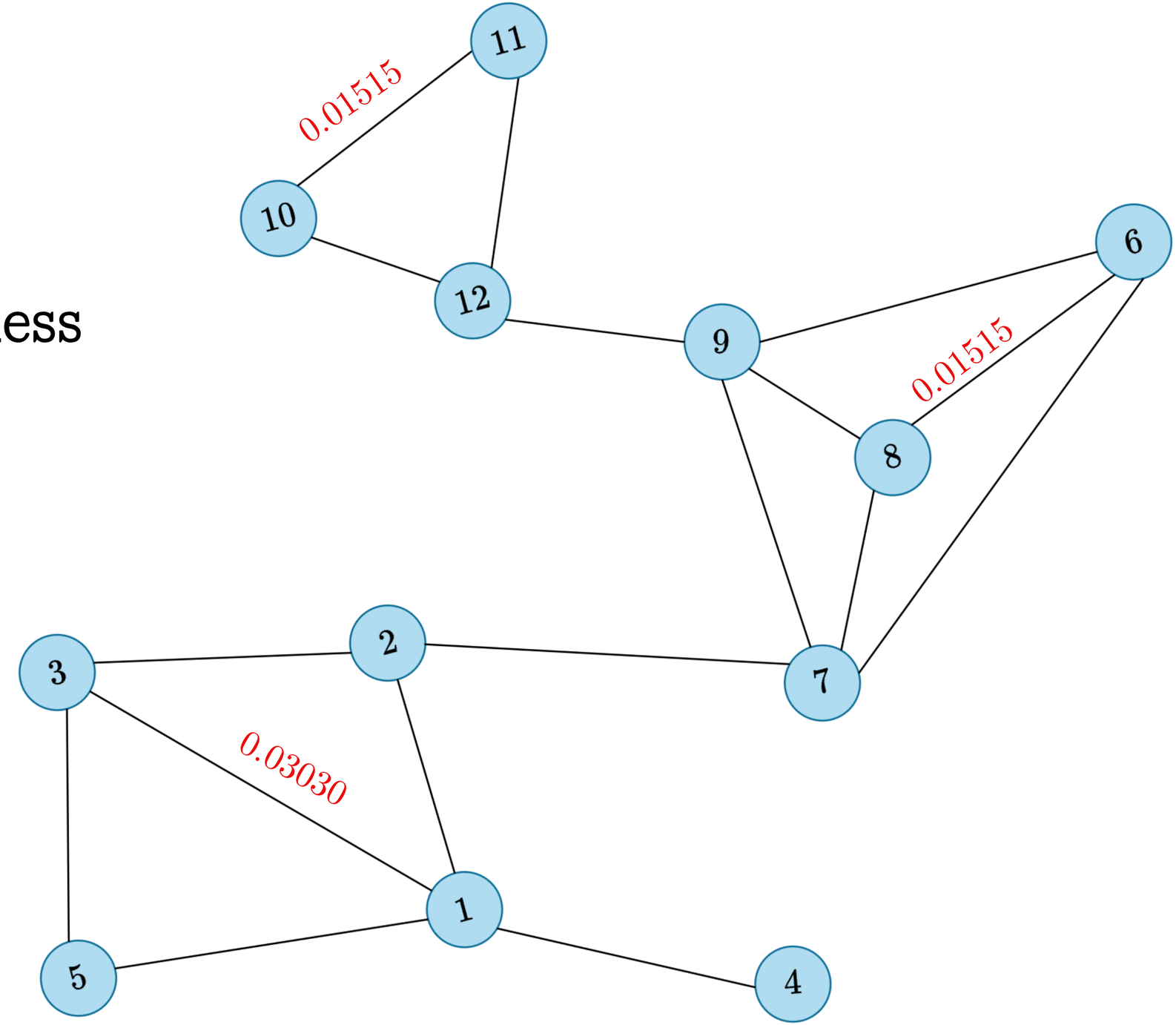



11
10
12
6
9
8
7
2
3
1
5
4

Edge Betweenness of the Edges

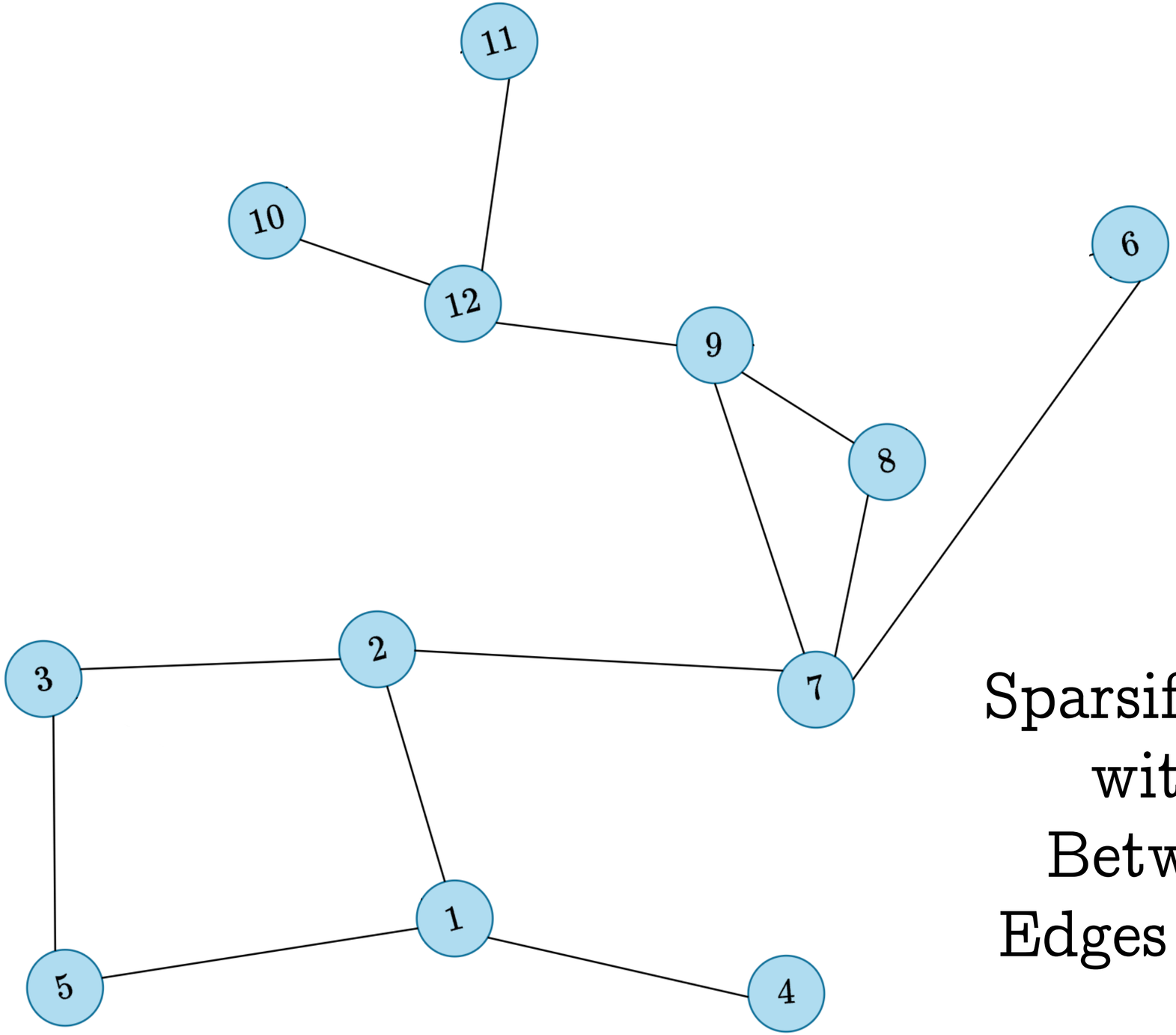


Low Betweenness Edges

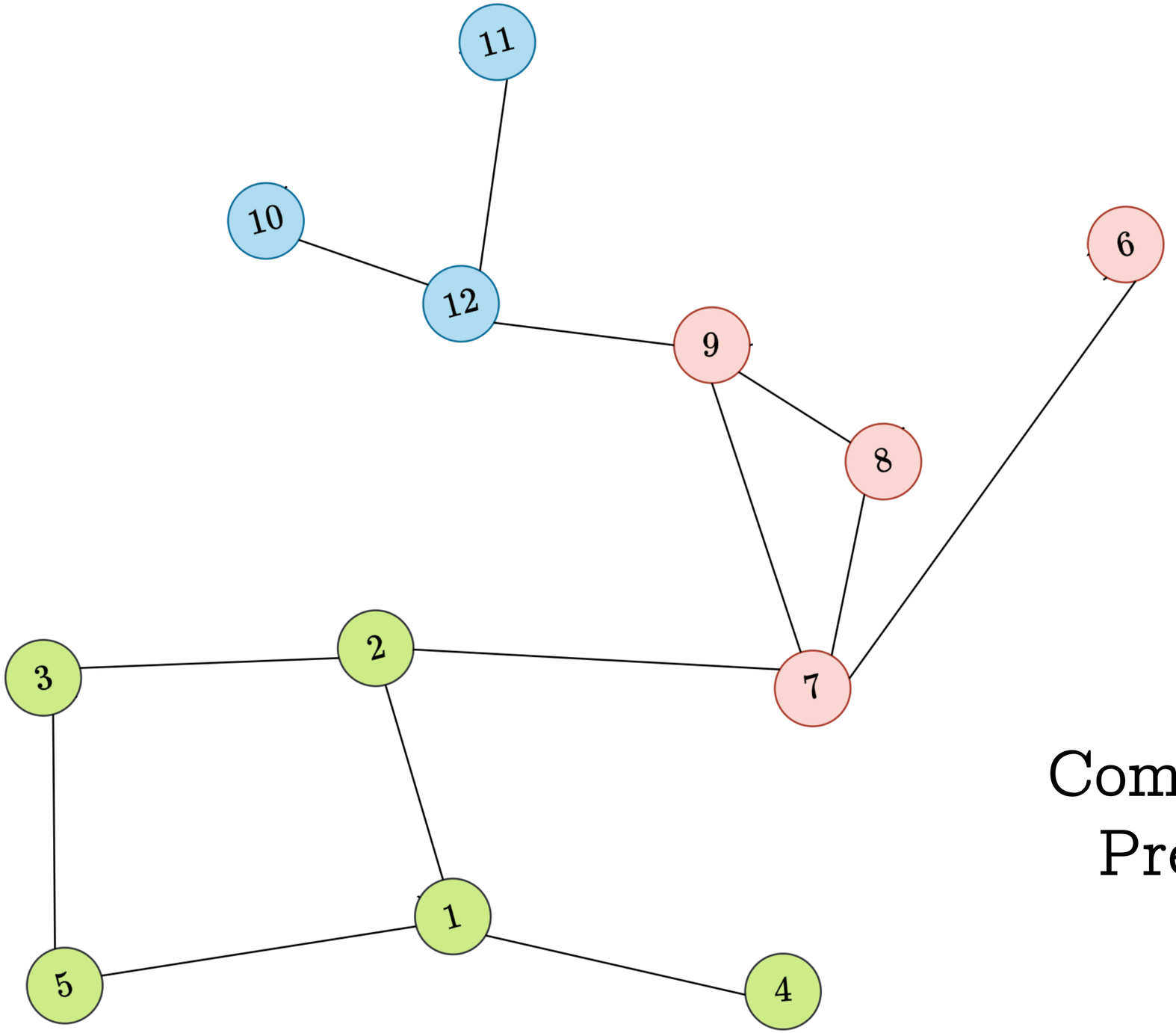


$$|V| = 12$$

$$|E| = 11$$

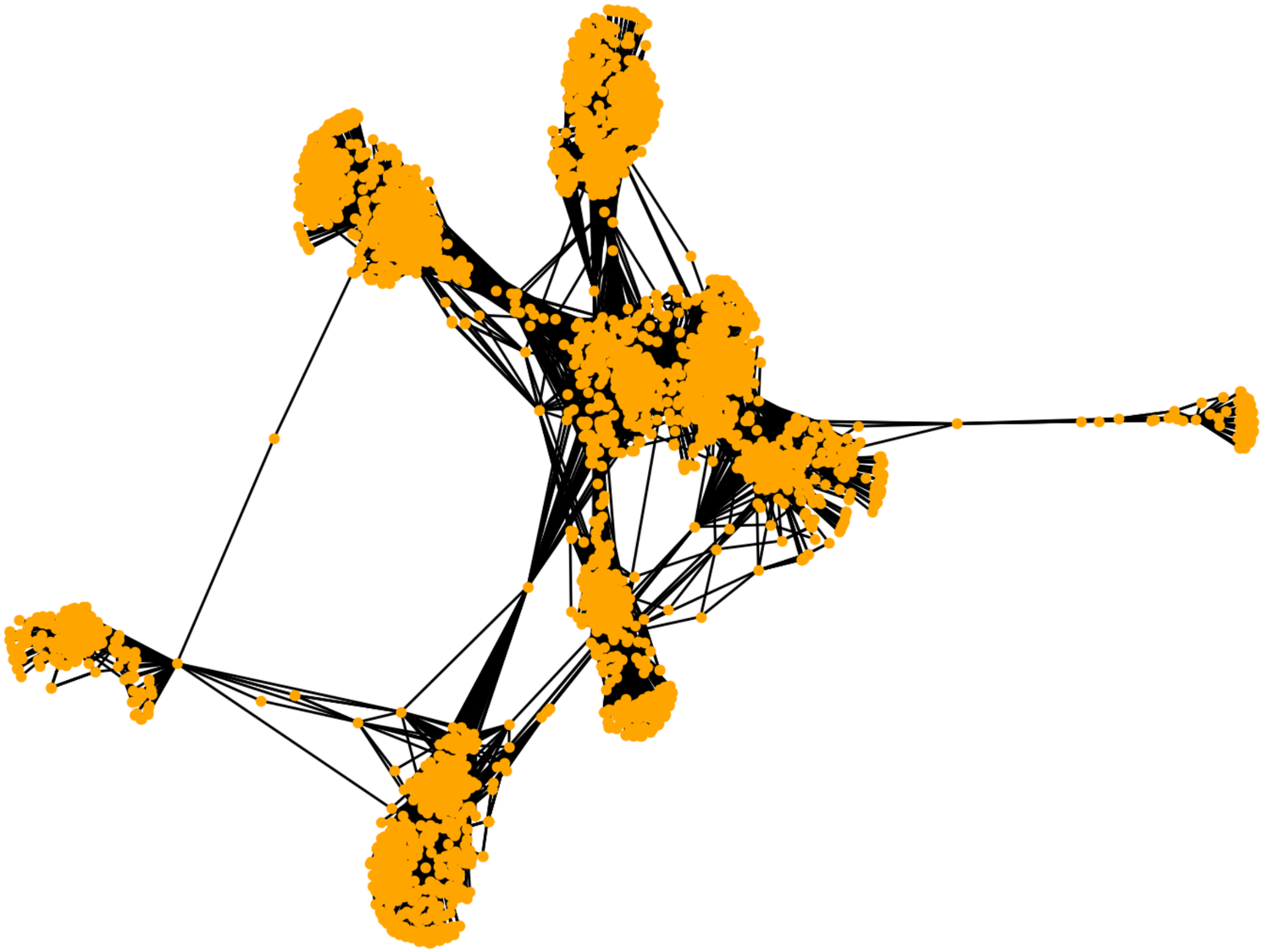


Sparsified Graph with Low Betweenness Edges Removed



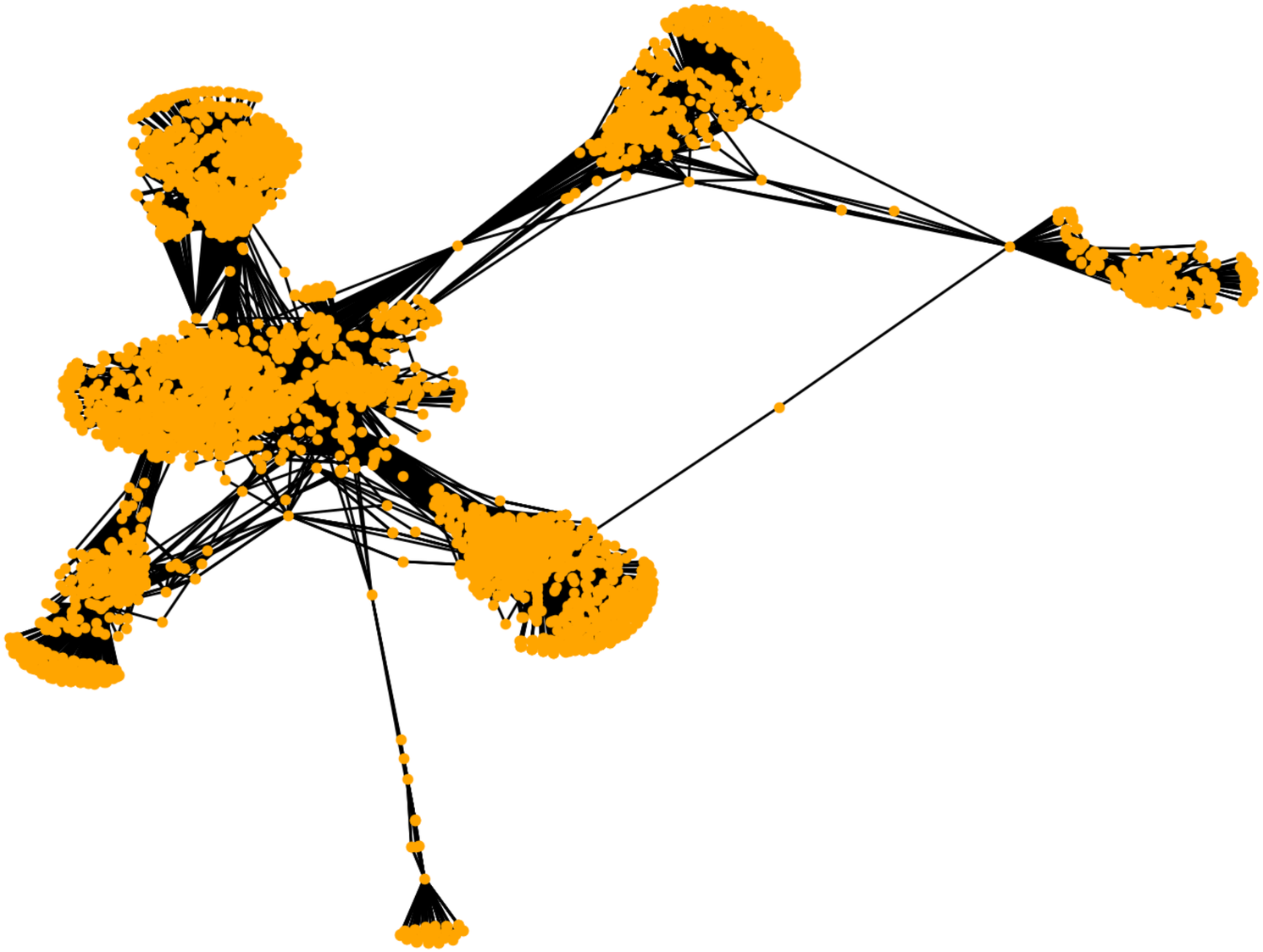
Communities Preserved

Graph with Top 50.0 % Highest Betweenness Edges Retained
Number of edges: 44117



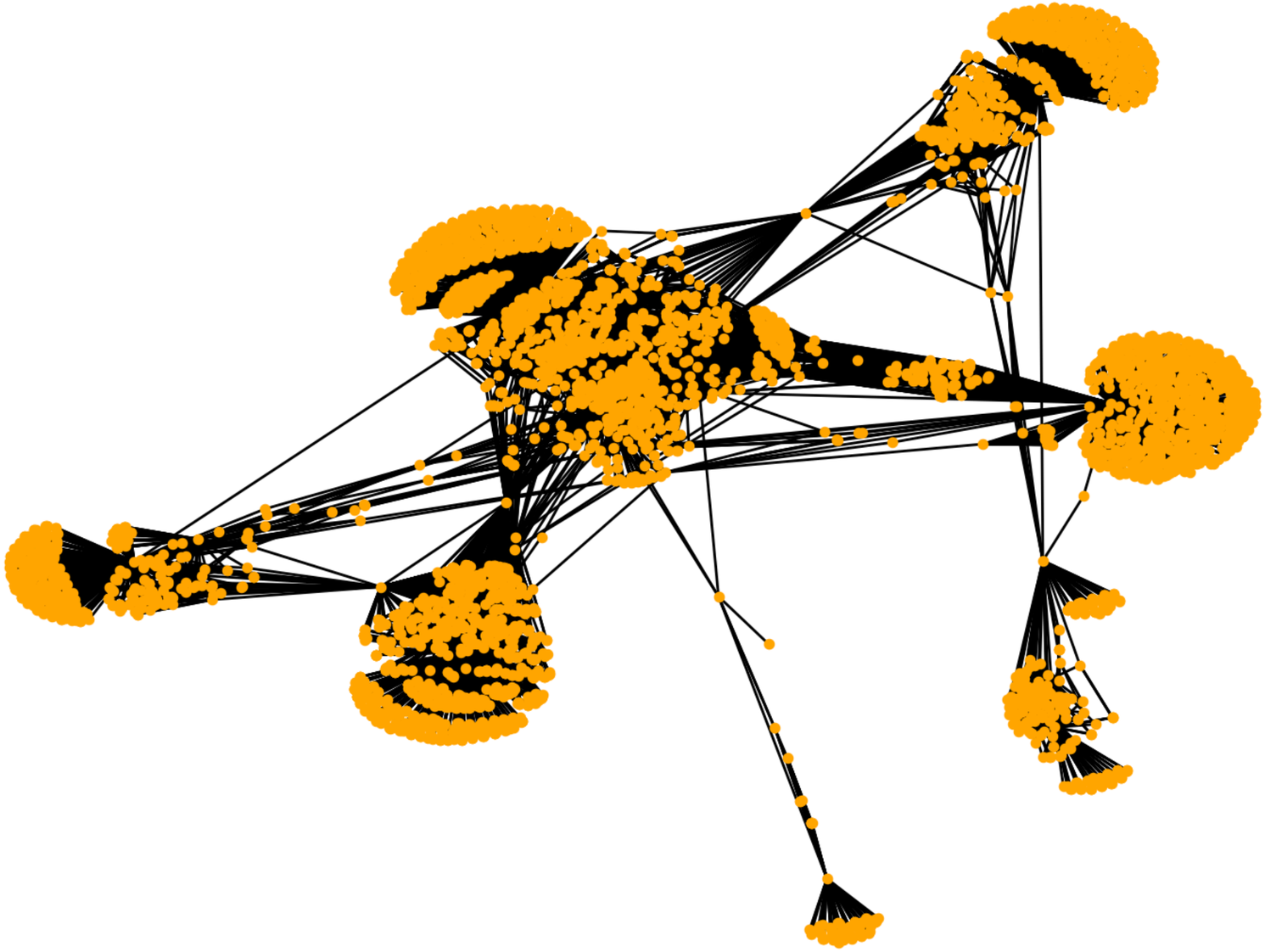
FaceBook

Graph with Top 30.0 % Highest Betweenness Edges Retained
Number of edges: 26471



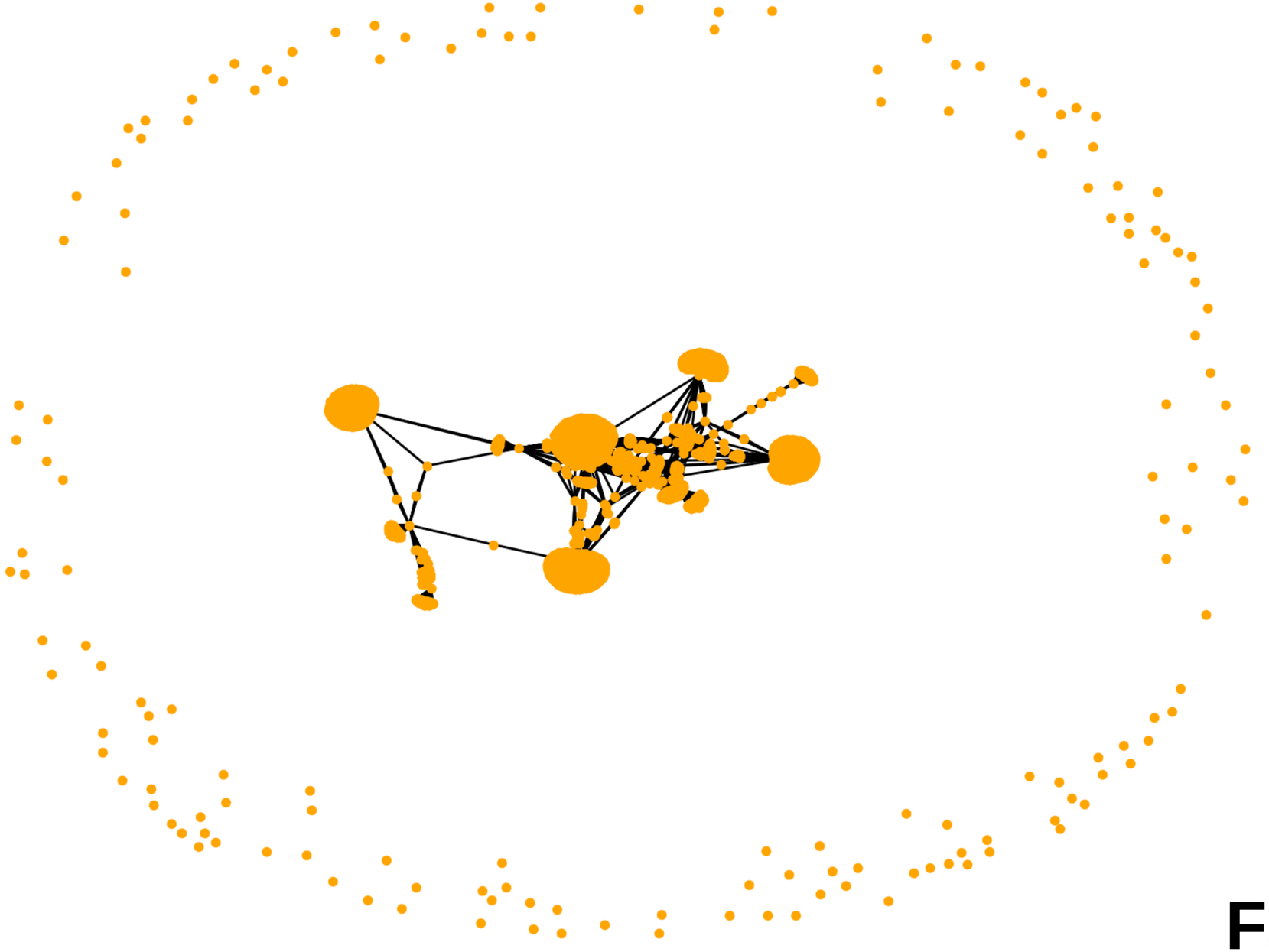
FaceBook

Graph with Top 10.0 % Highest Betweenness Edges Retained
Number of edges: 8824



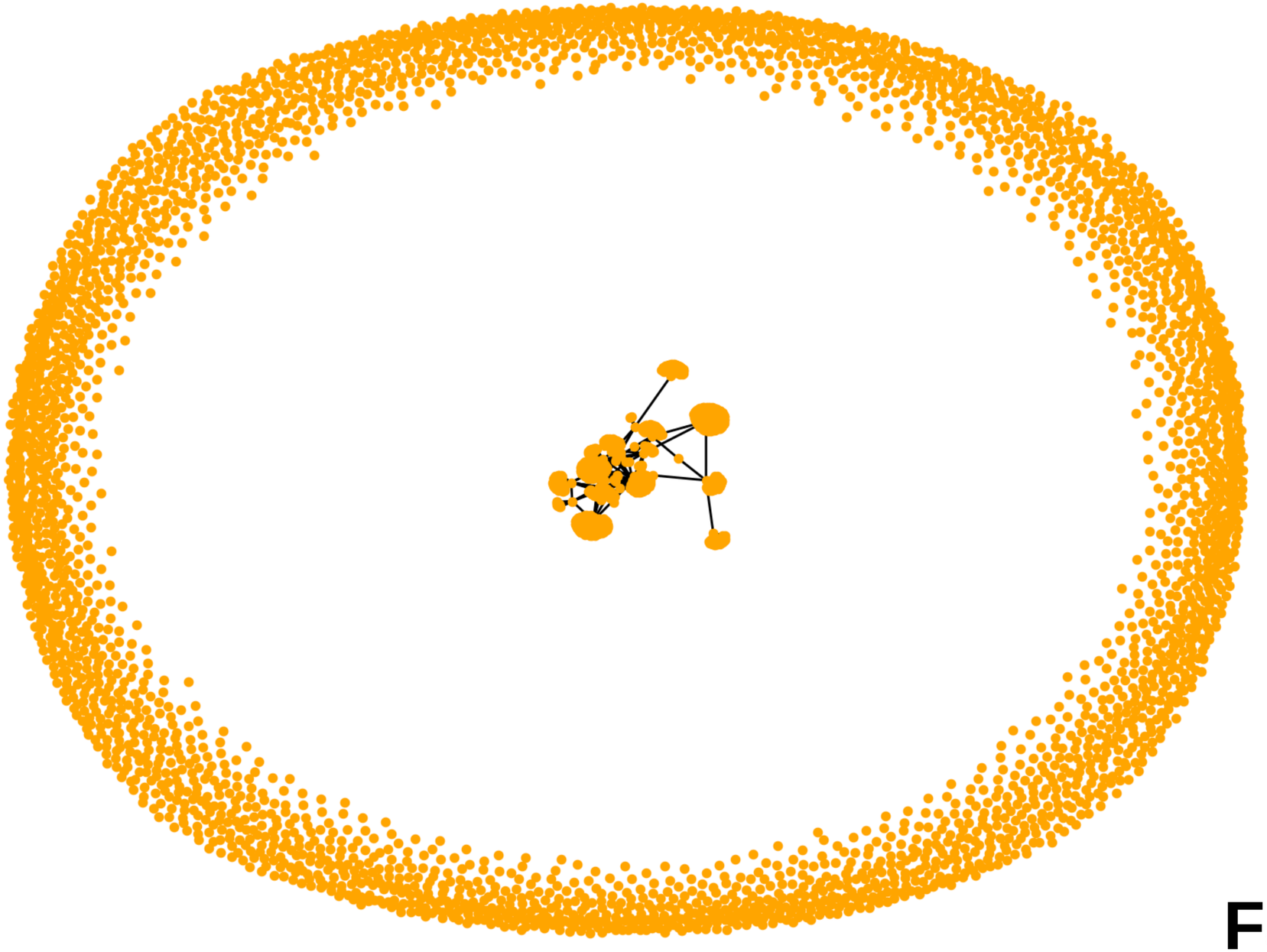
FaceBook

Graph with Top 5.0 % Highest Betweenness Edges Retained
Number of edges: 4412



FaceBook

Graph with Top 1.0 % Highest Betweenness Edges Retained
Number of edges: 883



FaceBook

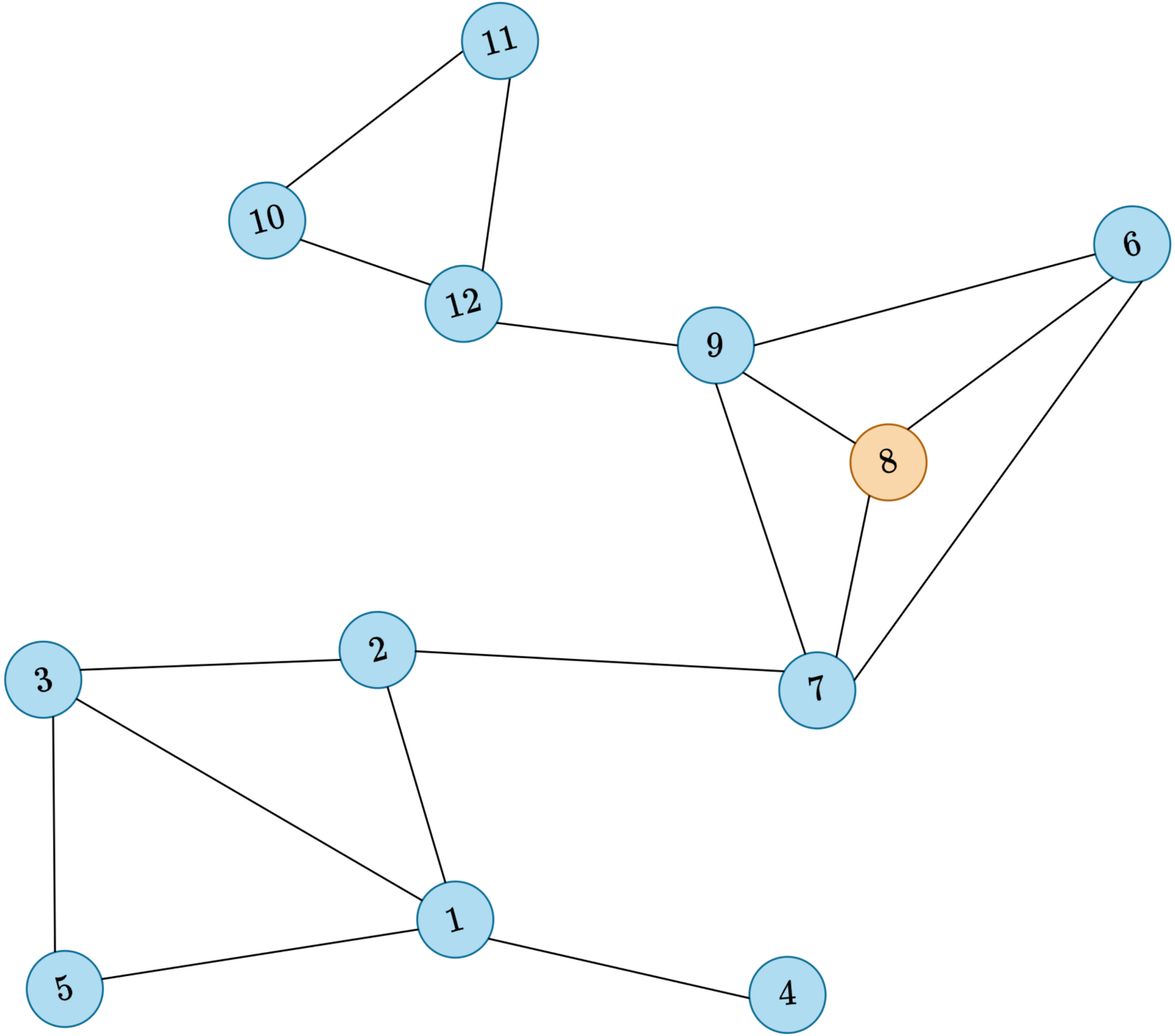
Jaccard Similarity

An edge (**i**, **j**) is *likely* to lie within a cluster if the vertices **i** and **j** have adjacency lists with *high overlap*.

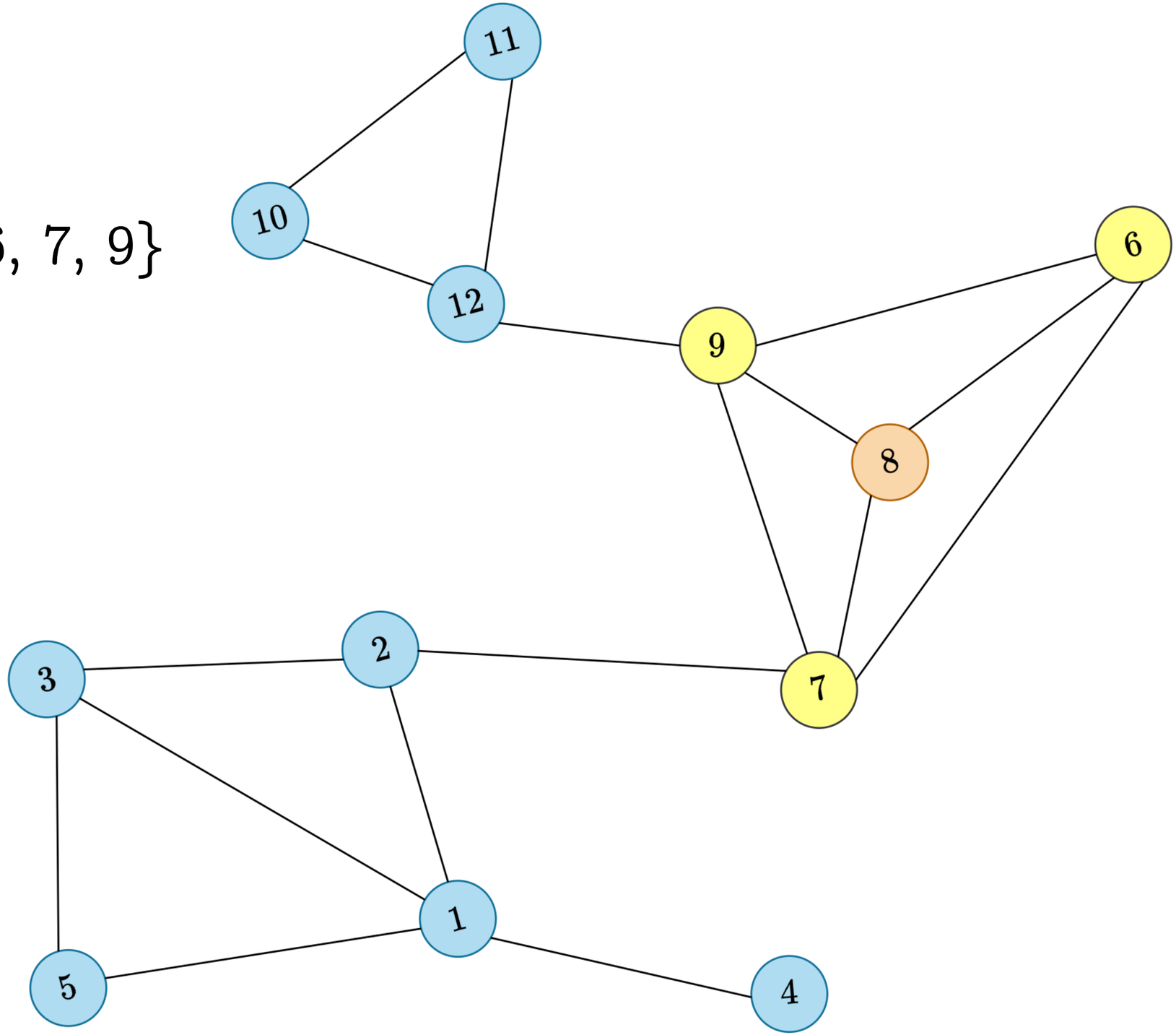
$$Sim(i,j) = \frac{|Adj(i) \cap Adj(j)|}{|Adj(i) \cup Adj(j)|}$$

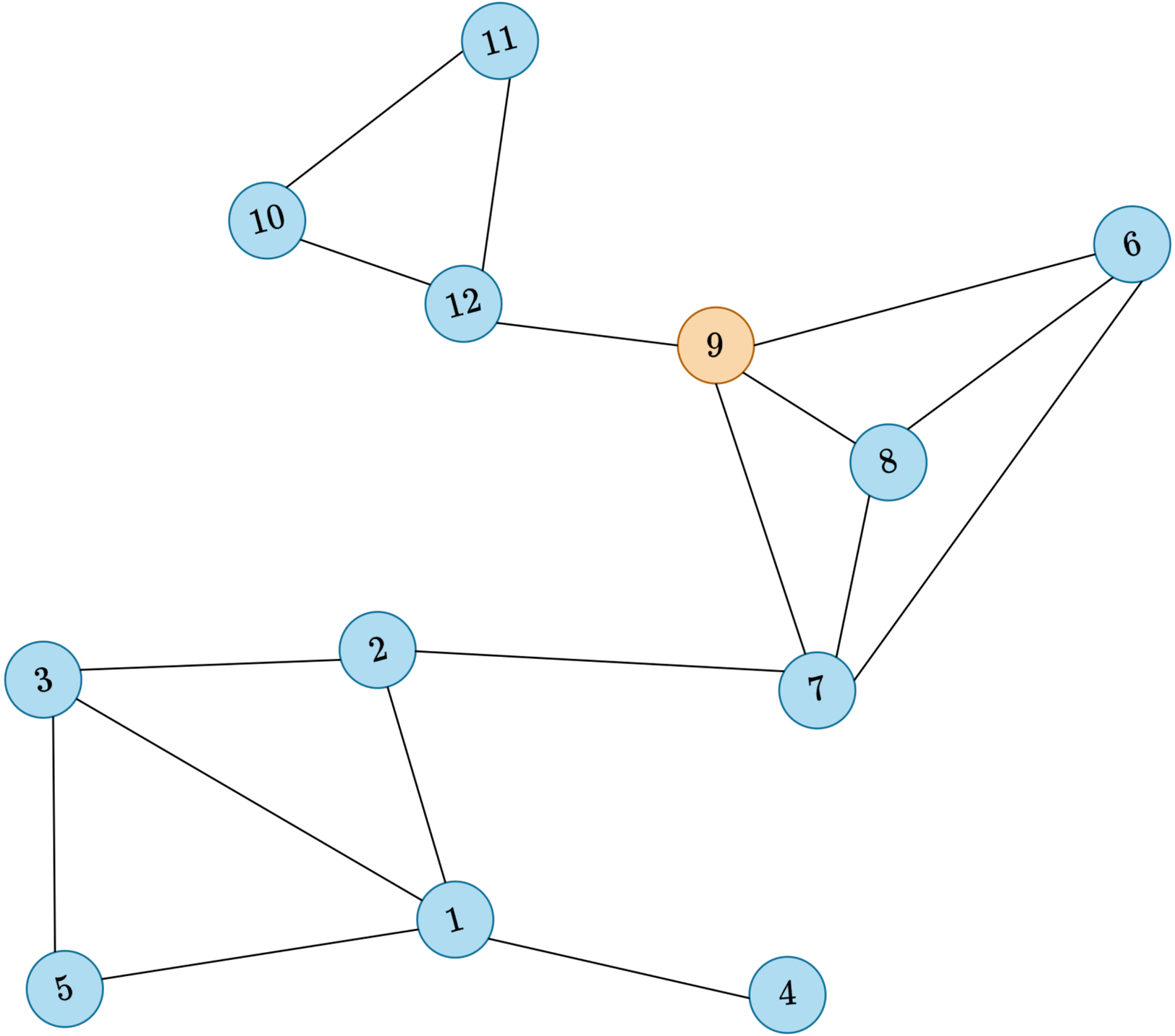
↑

Adjacency List of Node i

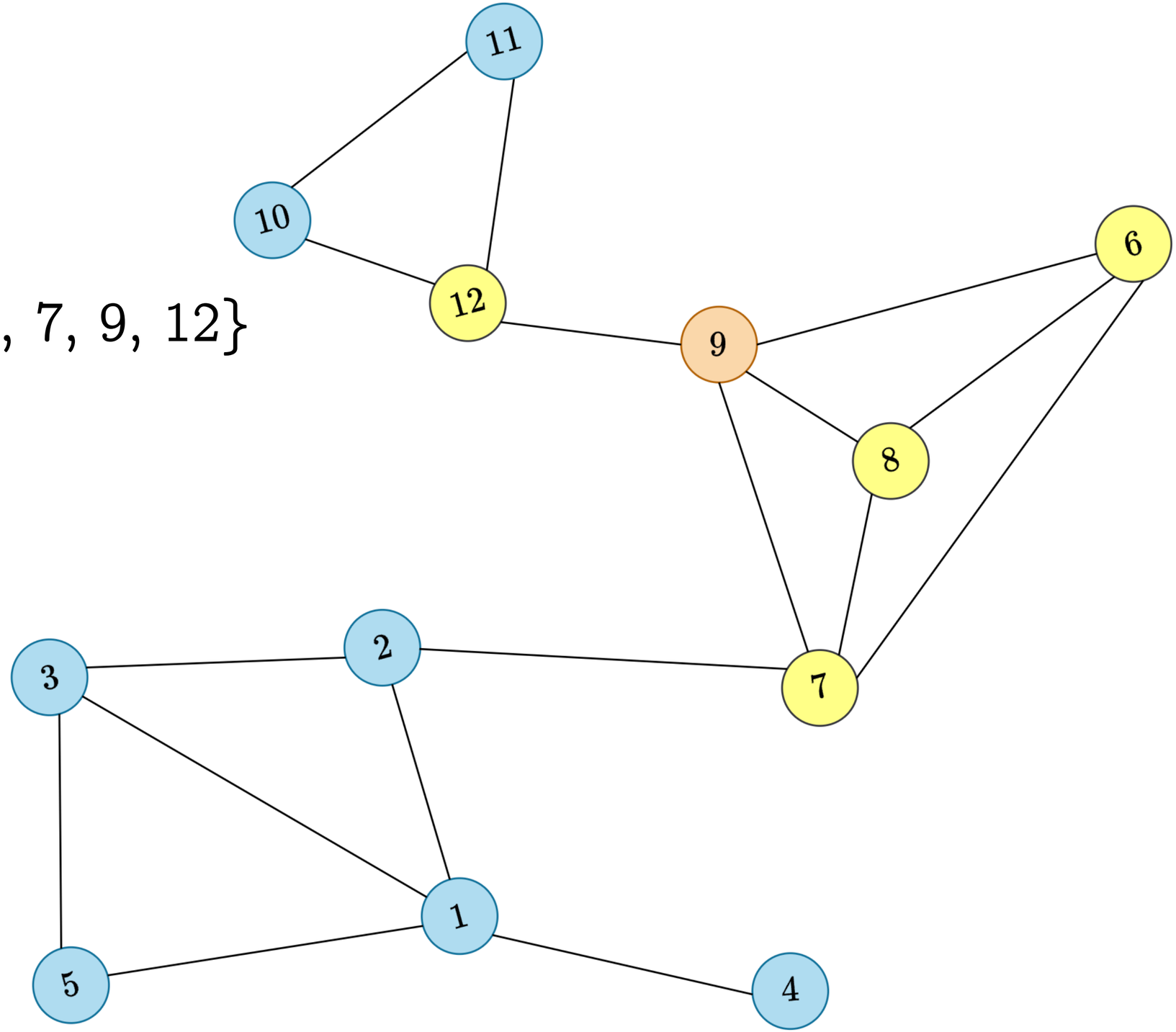

11
10
12
6
9
8
2
3
7
1
5
4

$\text{Adj}(8) = \{6, 7, 9\}$




11
10
12
6
9
8
2
3
7
1
5
4

$\text{Adj}(9) = \{6, 7, 9, 12\}$



$$\mathrm{Adj}(8) = \{6, 7, 9\}$$

$$\mathrm{Adj}(9) = \{6, 7, 9, 12\}$$

$$\text{Adj}(8) = \{6, 7, 9\}$$

$$\text{Adj}(9) = \{6, 7, 9, 12\}$$

$$\text{Sim}(8, 9) = \frac{|\ \{6, 7, 9\} \cap \{6, 7, 9, 12\}\ |}{|\ \{6, 7, 9\} \cup \{6, 7, 9, 12\}\ |}$$

$$\mathrm{Adj}(8) = \{6, 7, 9\}$$

$$\mathrm{Adj}(9) = \{6, 7, 9, 12\}$$

$$\mathrm{Sim}(8, 9) = \frac{3}{4}$$

Algorithm 1 Global Sparsification Algorithm

Input: Graph $G = (V, E)$, Sparsification ratio s

$G_{sparse} \leftarrow \emptyset$
for each edge $e=(i,j)$ in E **do**
    $e.sim = Sim(i, j)$ according to Eqn 1
end for

Sort all edges in E by $e.sim$
Add the top $s\%$ edges to G_{sparse}

return G_{sparse}

Time Complexity : $O(|E|(m+log|E|))$
m is size of largest adjacency list.

Space Complexity : $O(|V|+|E|)$

Algorithm 2 Local Sparsification Algorithm

Input: Graph $G = (V, E)$, Local Sparsification exponent e
Output: Sparsified graph G_{sparse}

$G_{sparse} \leftarrow \emptyset$
for each node i in V **do**
  Let d_i be the degree of i
  Let E_i be the set of edges incident to i
  for each edge $e=(i,j)$ in E_i **do**
    $e.sim = Sim(i, j)$ according to Eqn. 1
  end for
  Sort all edges in E_i by $e.sim$
  Add top d_i^e edges to G_{sparse}
end for

return G_{sparse}

Time Complexity : $O(|E|(m+log|E|))$
m is size of largest adjacency list.

Space Complexity : $O(|V|+|E|)$

Modularity

$$Q(\mathcal{C}) = \frac{1}{2m} \sum_{C \in \mathcal{C}} \sum_{u \in C, v \in C} \left(A_{u,v} - \frac{d_u d_v}{2m} \right)$$

Expected Edges b/w u, v

Edges b/w u, v

Clustering Coefficients

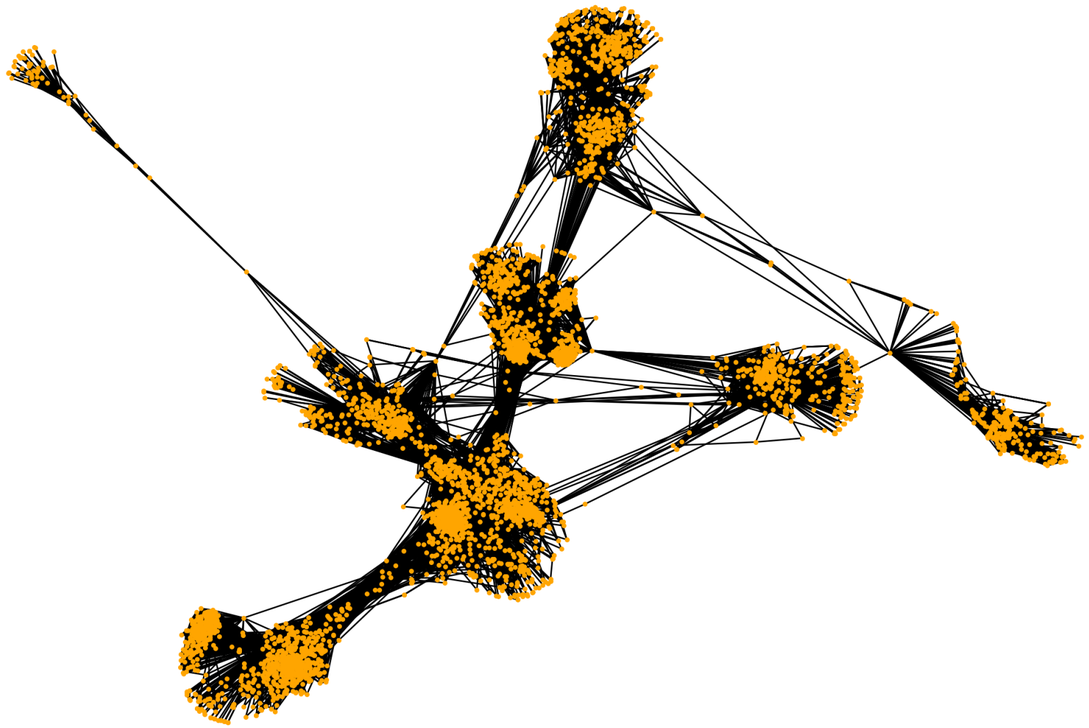
$$c_u = \frac{2T(u)}{deg(u)(deg(u)) - 1}$$

$2T(u)$: # triangles through node u

c_u: $u \in E$

Networks Used

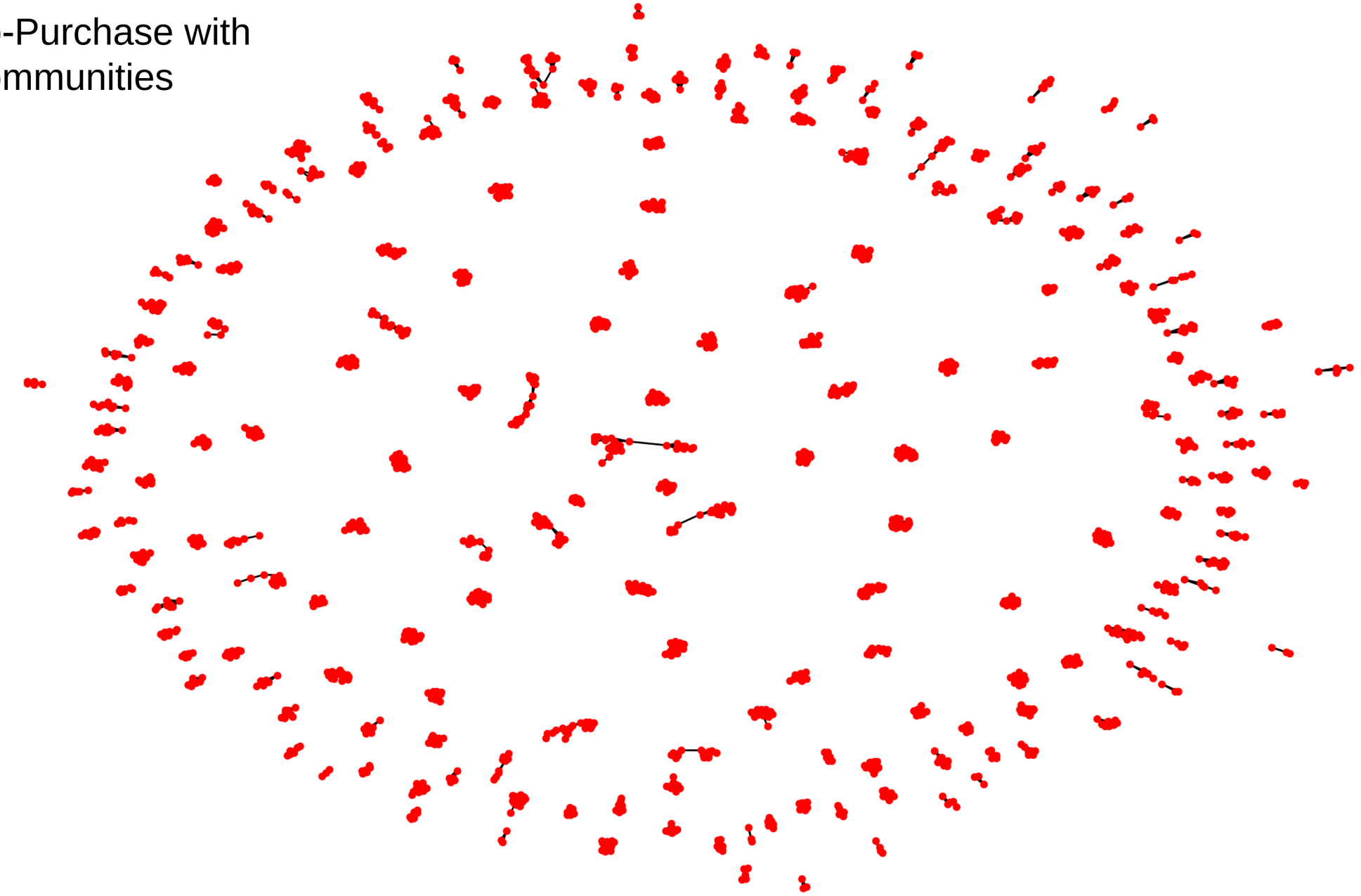
1. Amazon Co-Purchase
2. DBLP
3. Facebook Social
4. Email EU Core



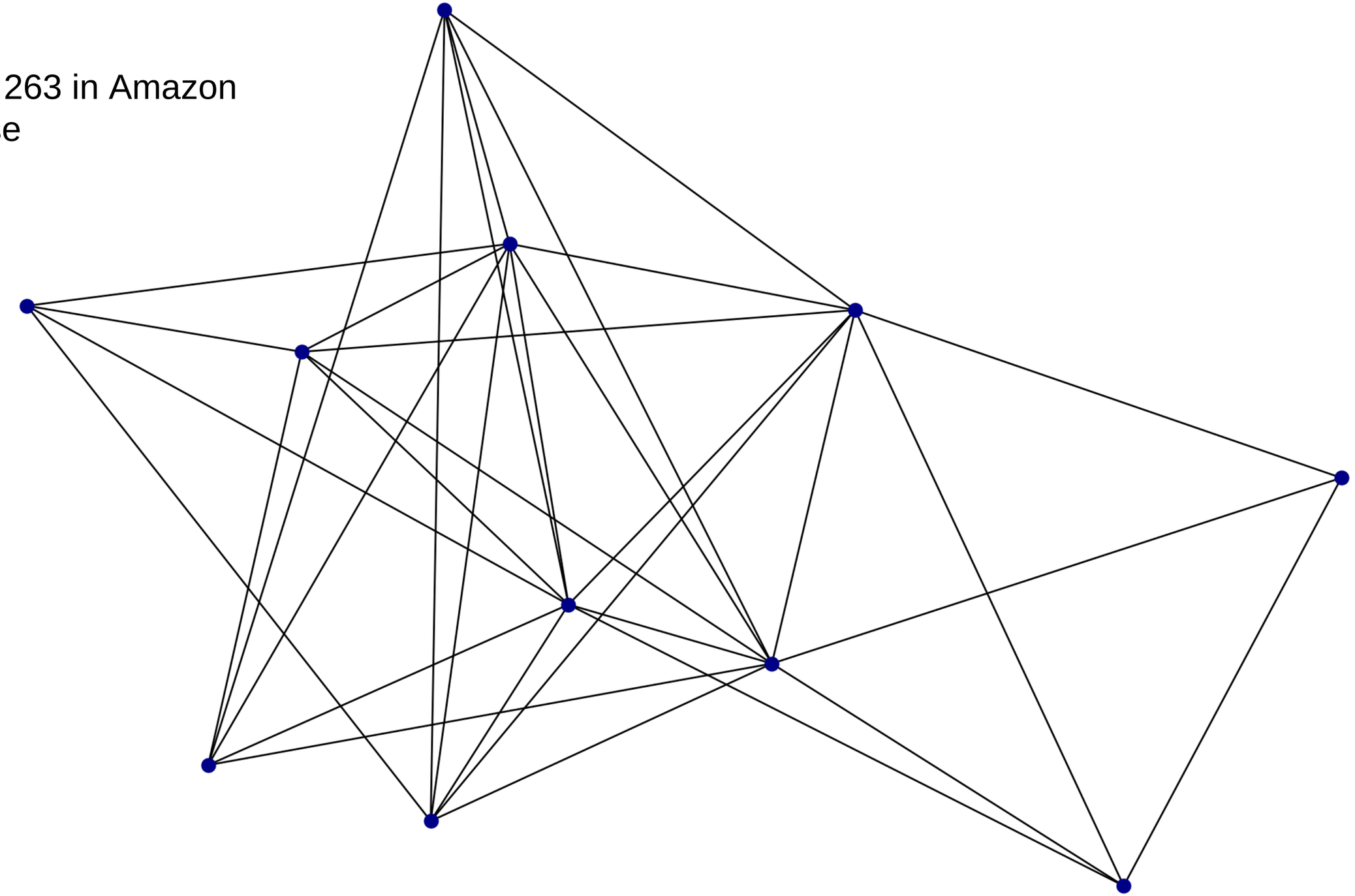
FaceBook Network

Network	$\|\mathbf{V}\|$	$\|\mathbf{E}\|$	$\|C\|$	$\|\mathbf{V}\|_{\mathbf{ind}}$	$\|\mathbf{E}\|_{\mathbf{ind}}$
DBLP	317080	1049866	150	1420	4609
Amazon	334863	925872	300	2008	5960
EU	1005	16064	42	1005	16064
Facebook	4039	88234	NA	4039	88234

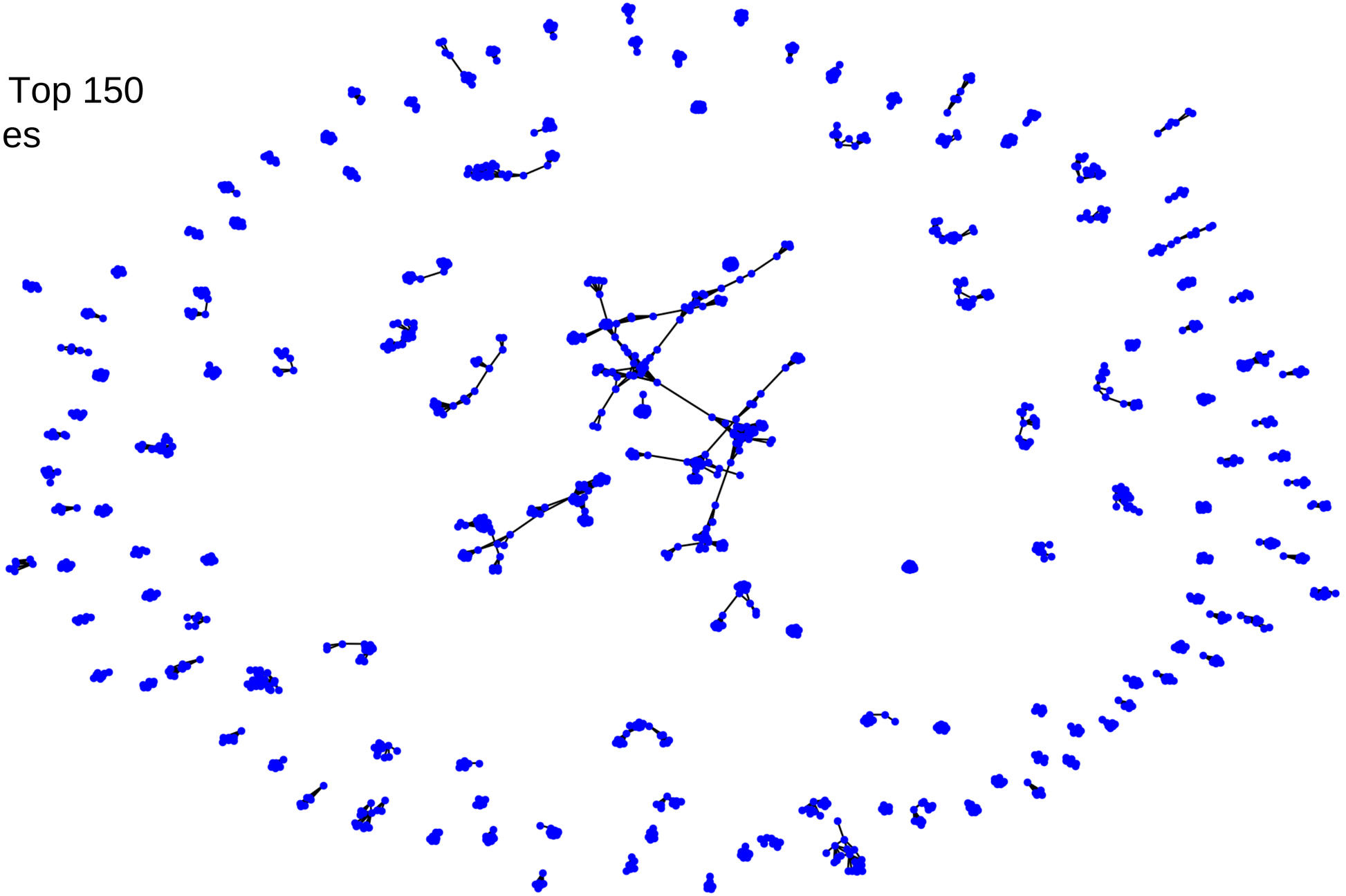
Amazon Co-Purchase with
Top 300 Communities



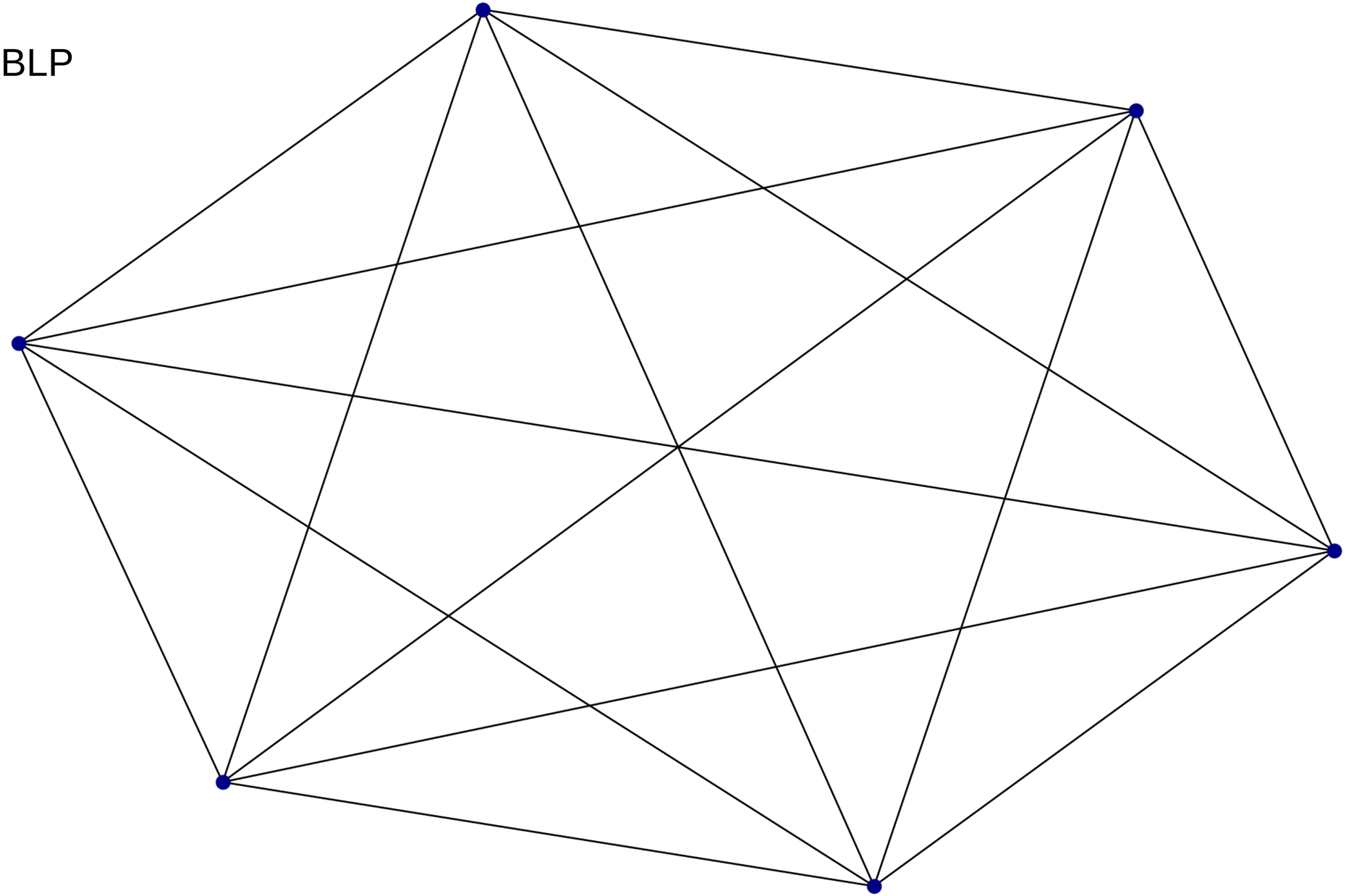
Community 263 in Amazon Co-Purchase



DBLP with Top 150 Communities



Community 4 in DBLP



Email EU Core Network

Community Detection Algorithms Used

1. Louvain Algorithm

A greedy algorithm to extract the community structure of a network based on modularity optimization.

2. Label Propogation Algorithm

The algorithm detects communities using network structure alone. At every iteration of propagation, each node updates its label to the one that the maximum numbers of its neighbours belongs to

3. InfoMap Algorithm

The algorithm uses the probability flow of random walks on a network as a proxy for information flows in the real system and decomposes the network into modules by compressing a description of the probability flow

Metrics for comparing Community Quality

1. Adjusted Rand Index

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}]/\binom{n}{2}}$$

2. Normalized Mutual Information

$$NMI(X,Y) = \frac{-2\sum_{ij} p_{ij}\log(p_{ij}/p_{i+}p_{+j})}{\sum_i p_{i+}\log(p_{i+}) + \sum_j p_{+j}\log(p_{+j})}$$

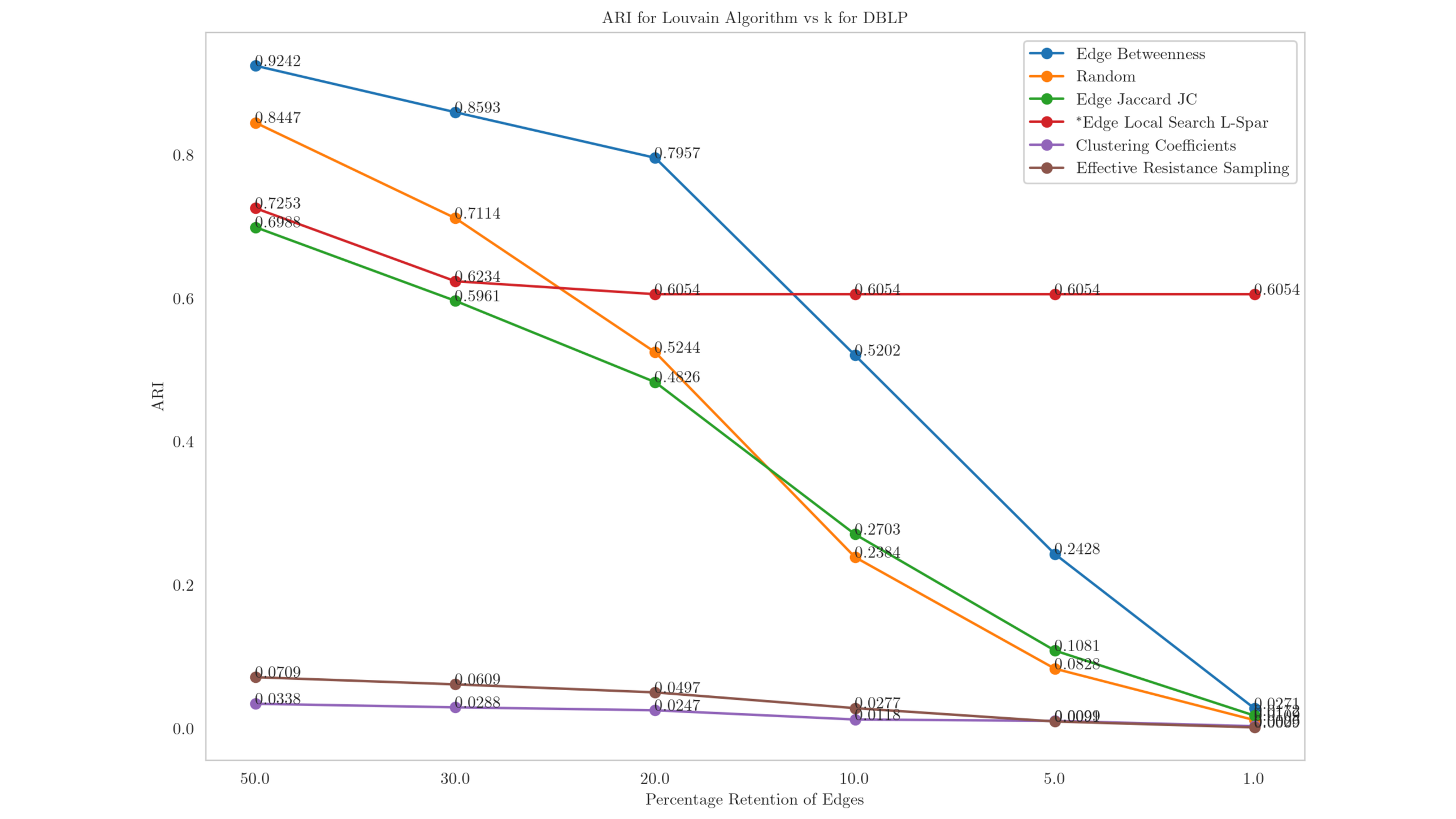
3. Modularity

$$Q(\mathcal{C}) = \frac{1}{2m}\sum_{C\in\mathcal{C}}\sum_{u\in C, v\in C}\left(A_{u,v} - \frac{d_u d_v}{2m}\right)$$

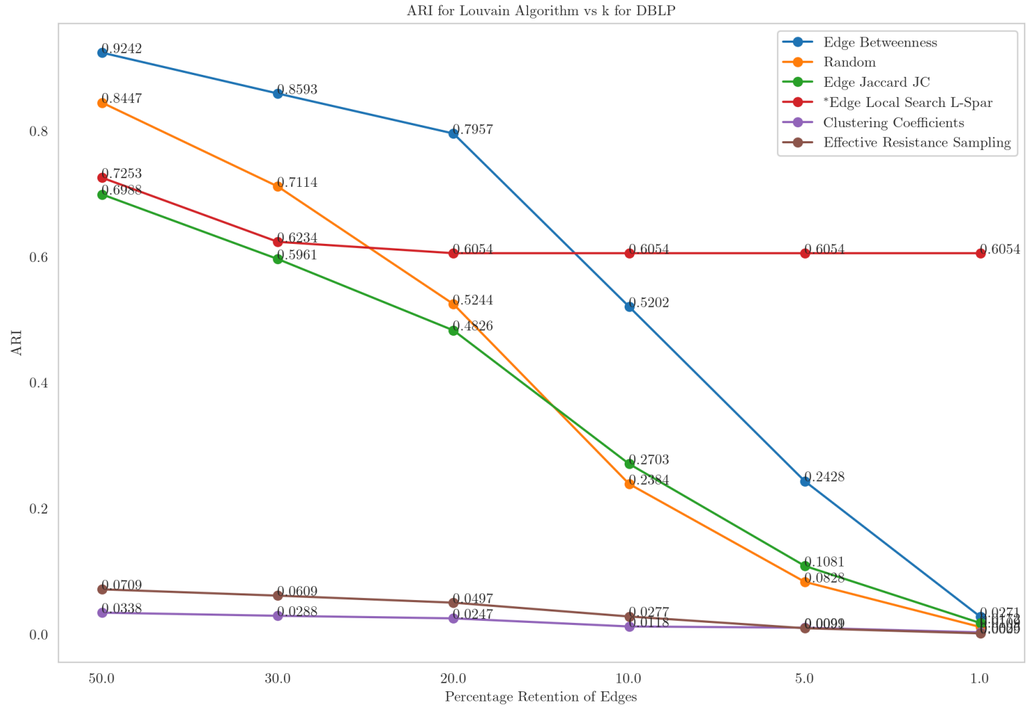
4. Clustering Coefficient

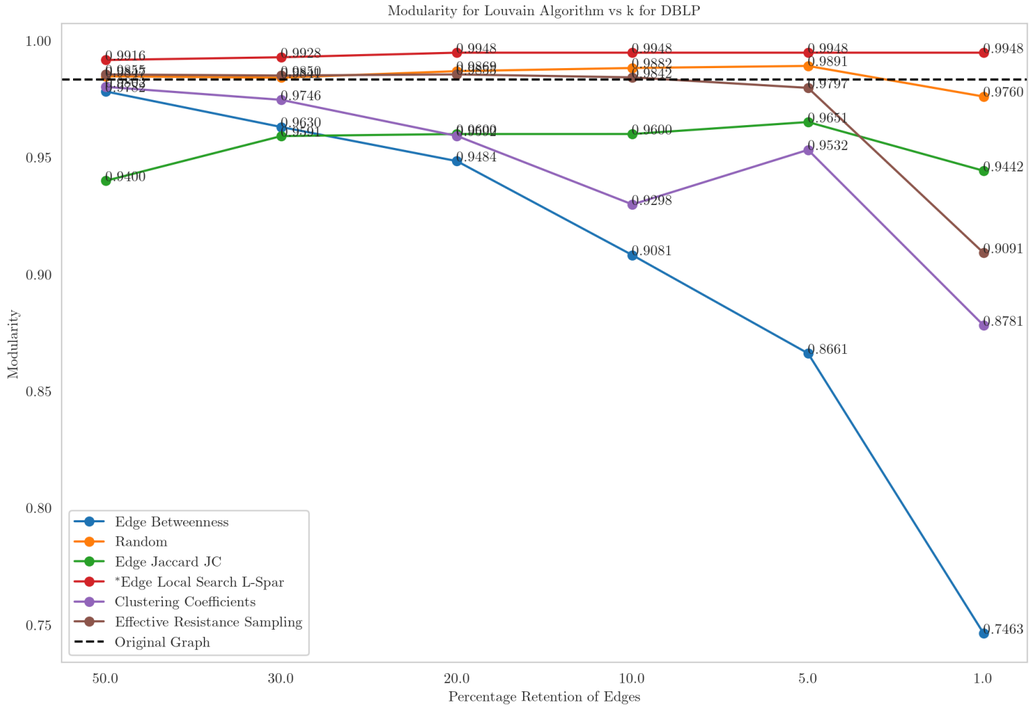
$$c_u = \frac{2T(u)}{deg(u)(deg(u)) - 1}$$

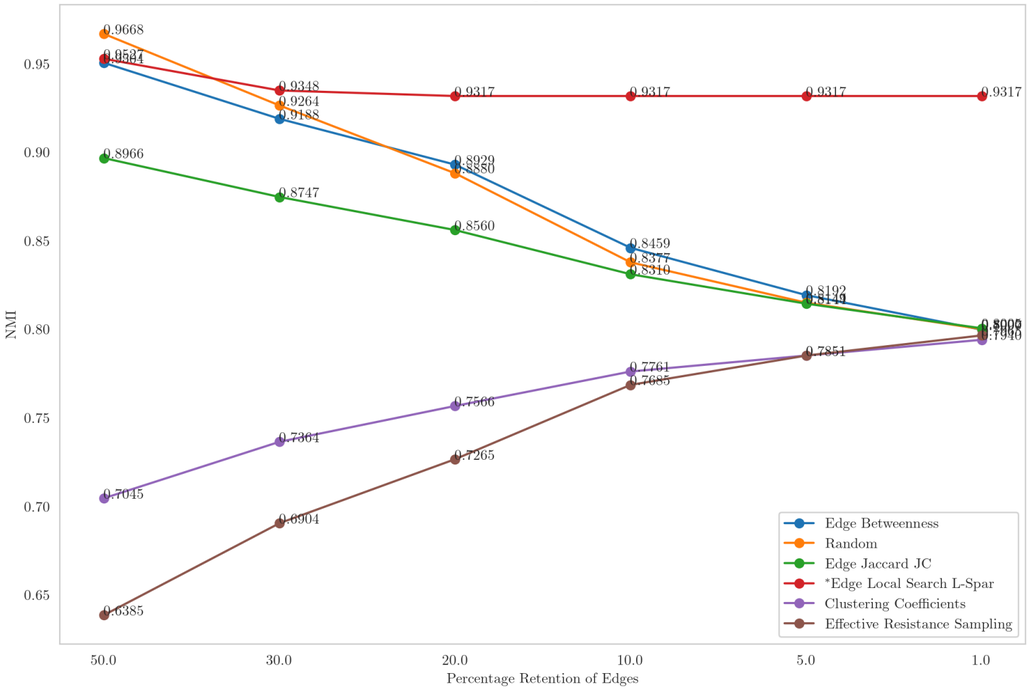
Results for DBLP Network

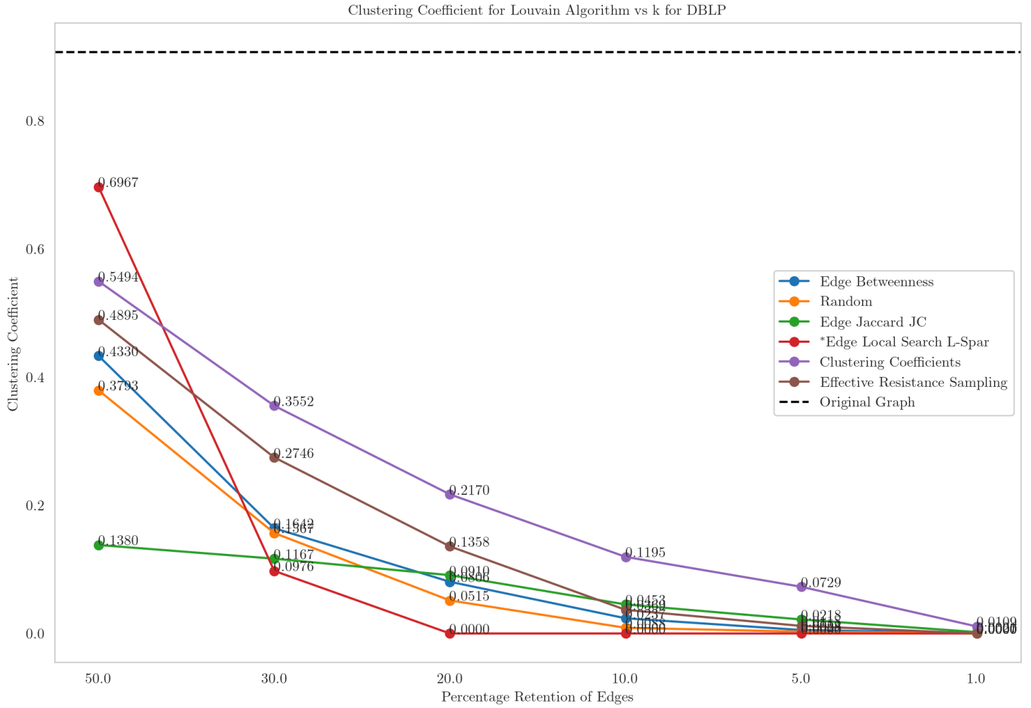

ARI for Louvain Algorithm vs k for DBLP
Edge Betweenness
Random
Edge Jaccard JC
*Edge Local Search L-Spar
Clustering Coefficients
Effective Resistance Sampling
ARI
Percentage Retention of Edges
0.0
0.2
0.4
0.6
0.8
50.0
30.0
20.0
10.0
5.0
1.0
0.9242
0.8593
0.7957
0.5202
0.2428
0.0271
0.8447
0.7114
0.5244
0.2384
0.0828
0.7253
0.6988
0.6234
0.6054
0.6054
0.6054
0.6054
0.5961
0.4826
0.2703
0.1081
0.0709
0.0609
0.0497
0.0277
0.0338
0.0288
0.0247
0.0118

DBLP Louvain

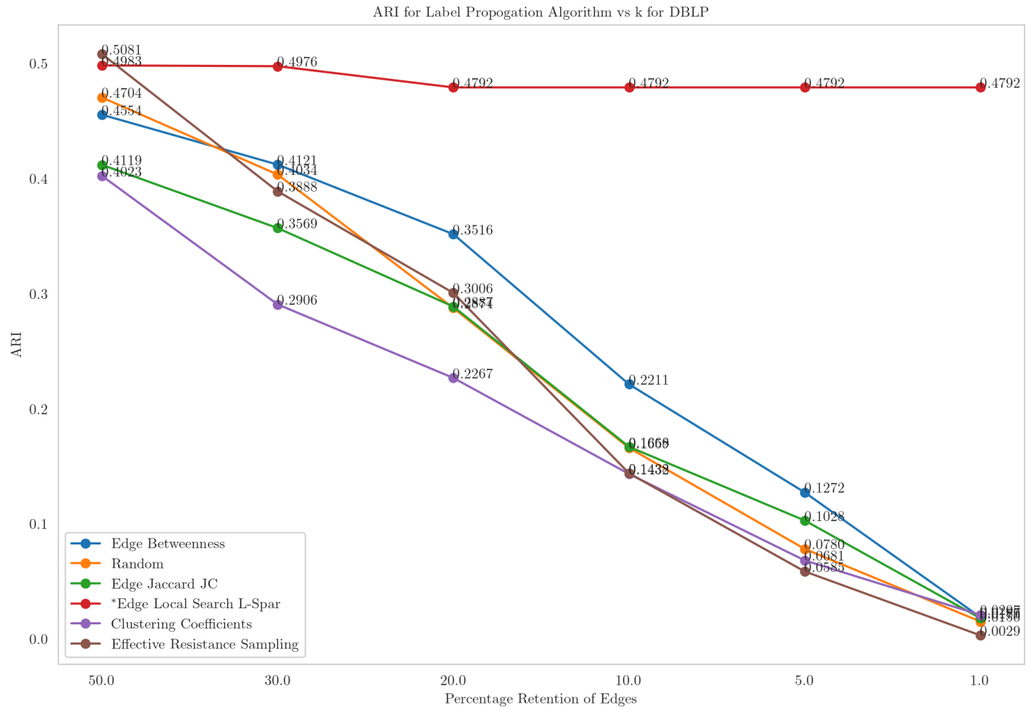


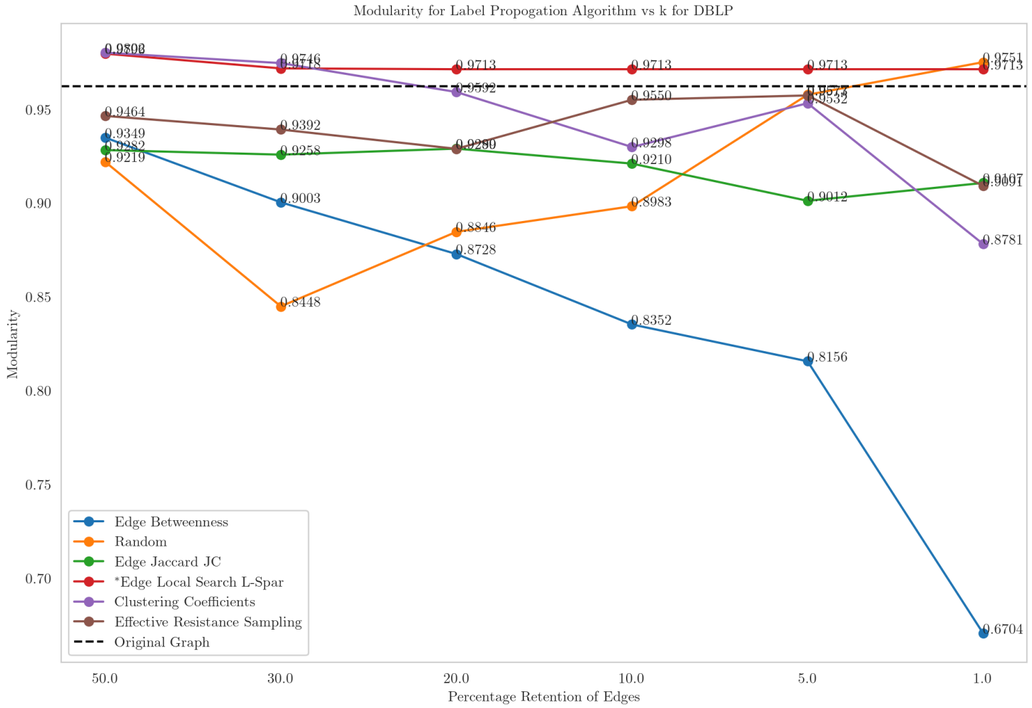


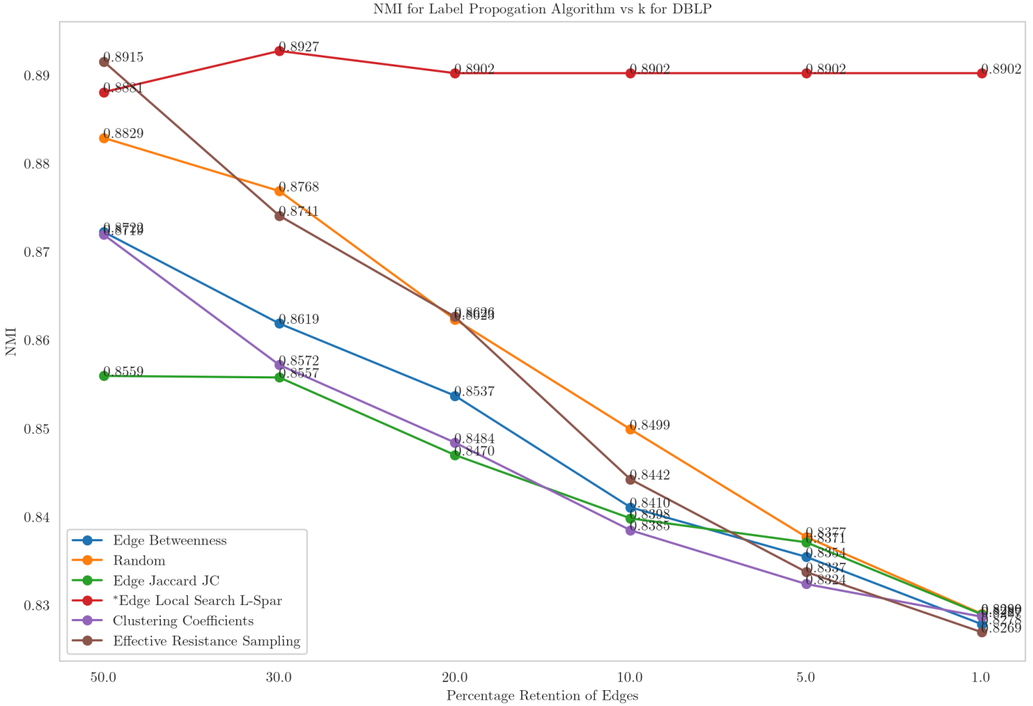


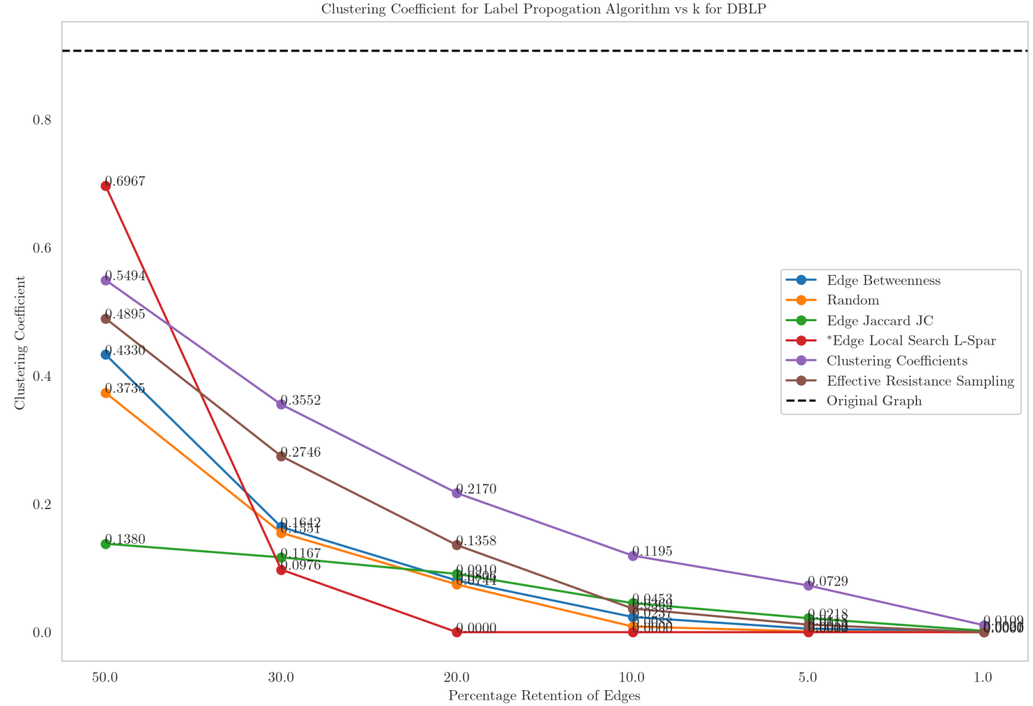


DBLP LPA

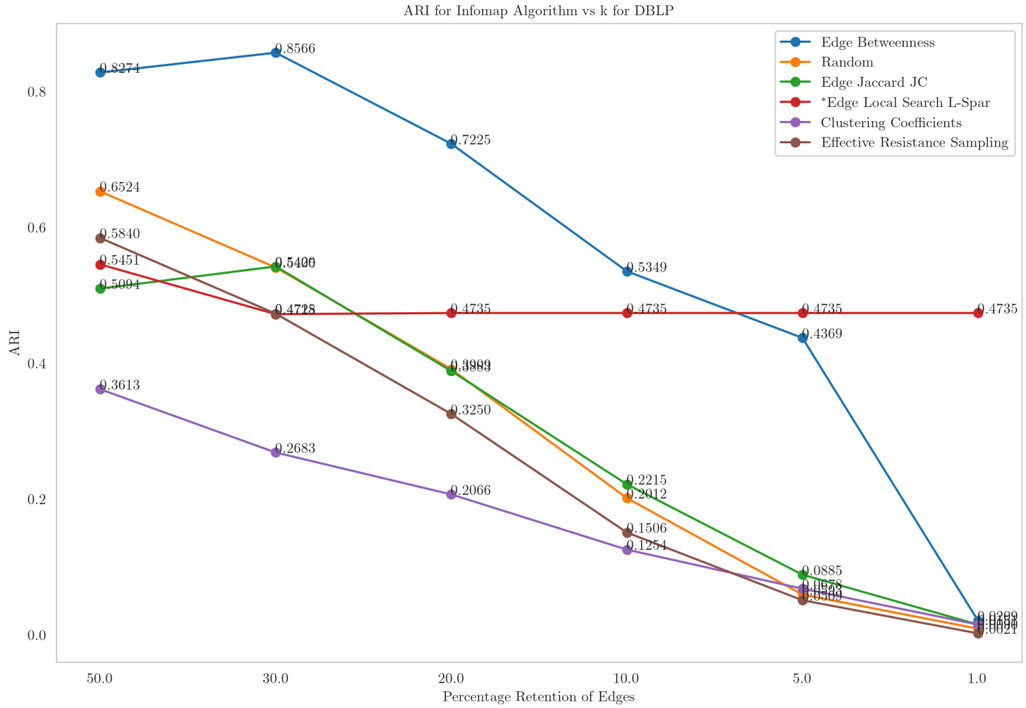

ARI for Label Propogation Algorithm vs k for DBLP
ARI
Percentage Retention of Edges
50.0
30.0
20.0
10.0
5.0
1.0
Edge Betweenness
Random
Edge Jaccard JC
*Edge Local Search L-Spar
Clustering Coefficients
Effective Resistance Sampling

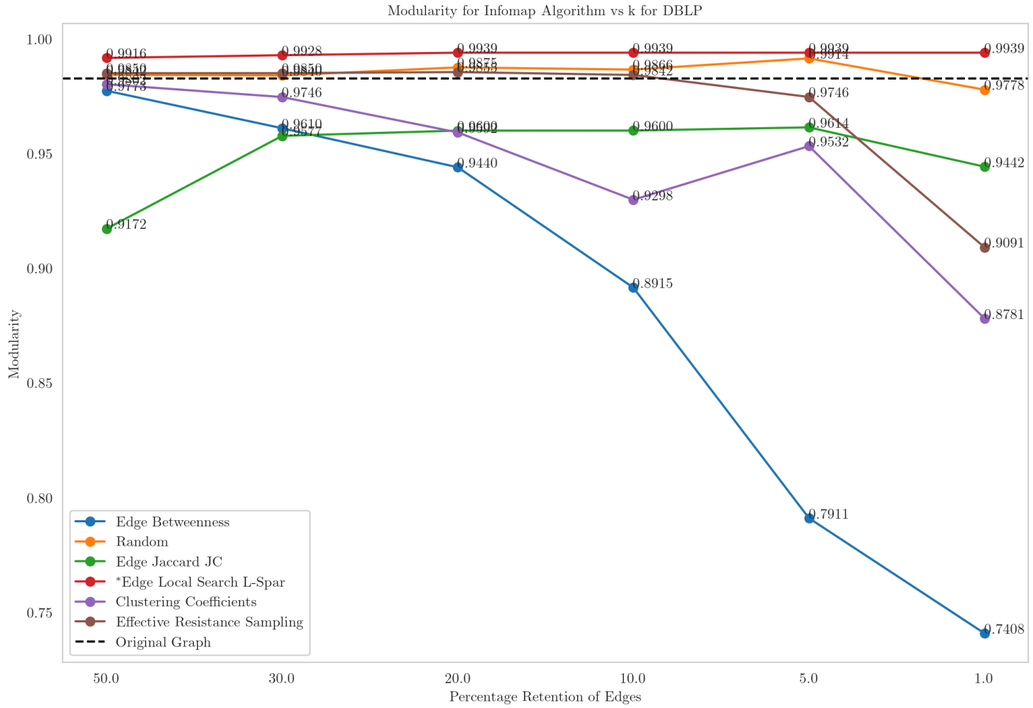

Modularity for Label Propogation Algorithm vs k for DBLP
Modularity
Percentage Retention of Edges
50.0
30.0
20.0
10.0
5.0
1.0
Edge Betweenness
Random
Edge Jaccard JC
*Edge Local Search L-Spar
Clustering Coefficients
Effective Resistance Sampling
Original Graph

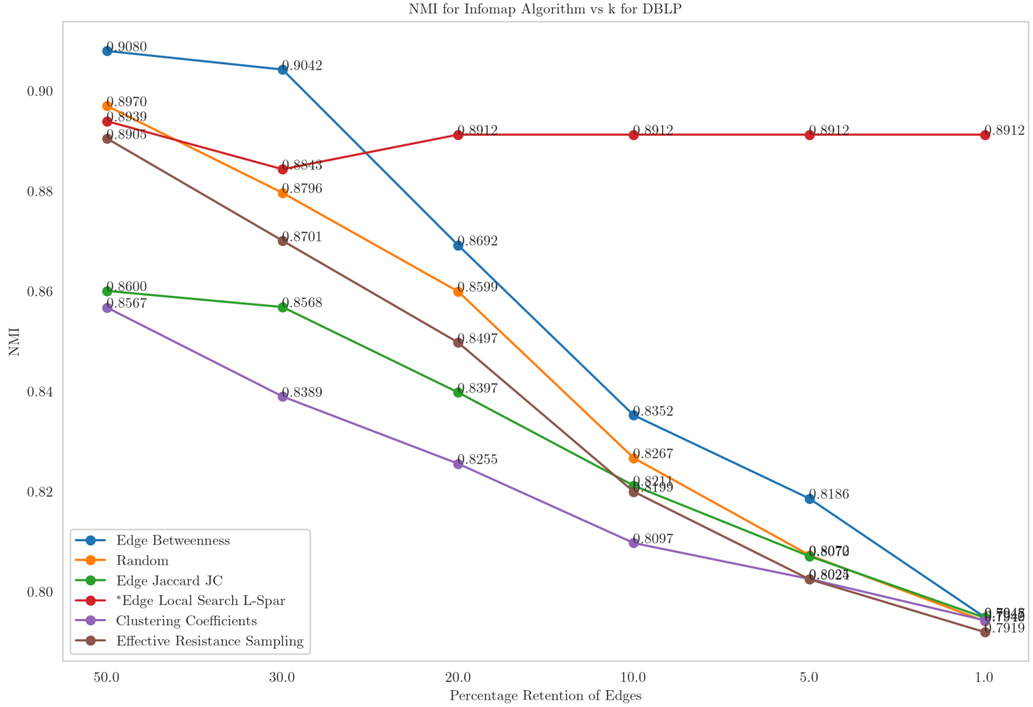

NMI for Label Propogation Algorithm vs k for DBLP
NMI
Percentage Retention of Edges
50.0
30.0
20.0
10.0
5.0
1.0
Edge Betweenness
Random
Edge Jaccard JC
*Edge Local Search L-Spar
Clustering Coefficients
Effective Resistance Sampling

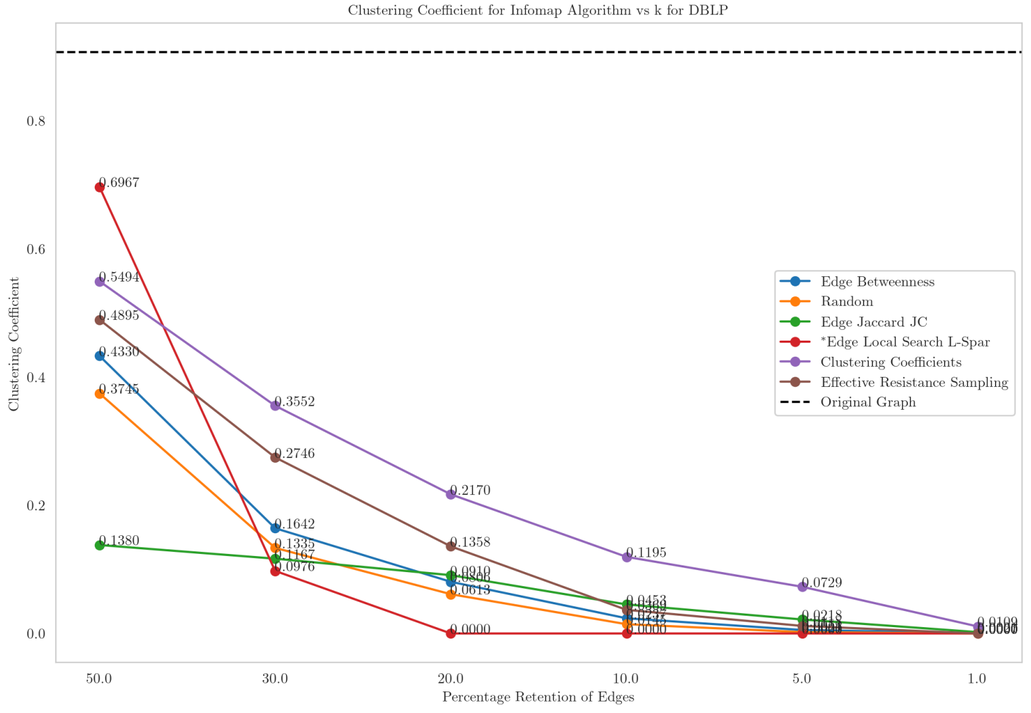

Clustering Coefficient for Label Propogation Algorithm vs k for DBLP
Clustering Coefficient
Percentage Retention of Edges
50.0
30.0
20.0
10.0
5.0
1.0
Edge Betweenness
Random
Edge Jaccard JC
*Edge Local Search L-Spar
Clustering Coefficients
Effective Resistance Sampling
Original Graph

DBLP InfoMap

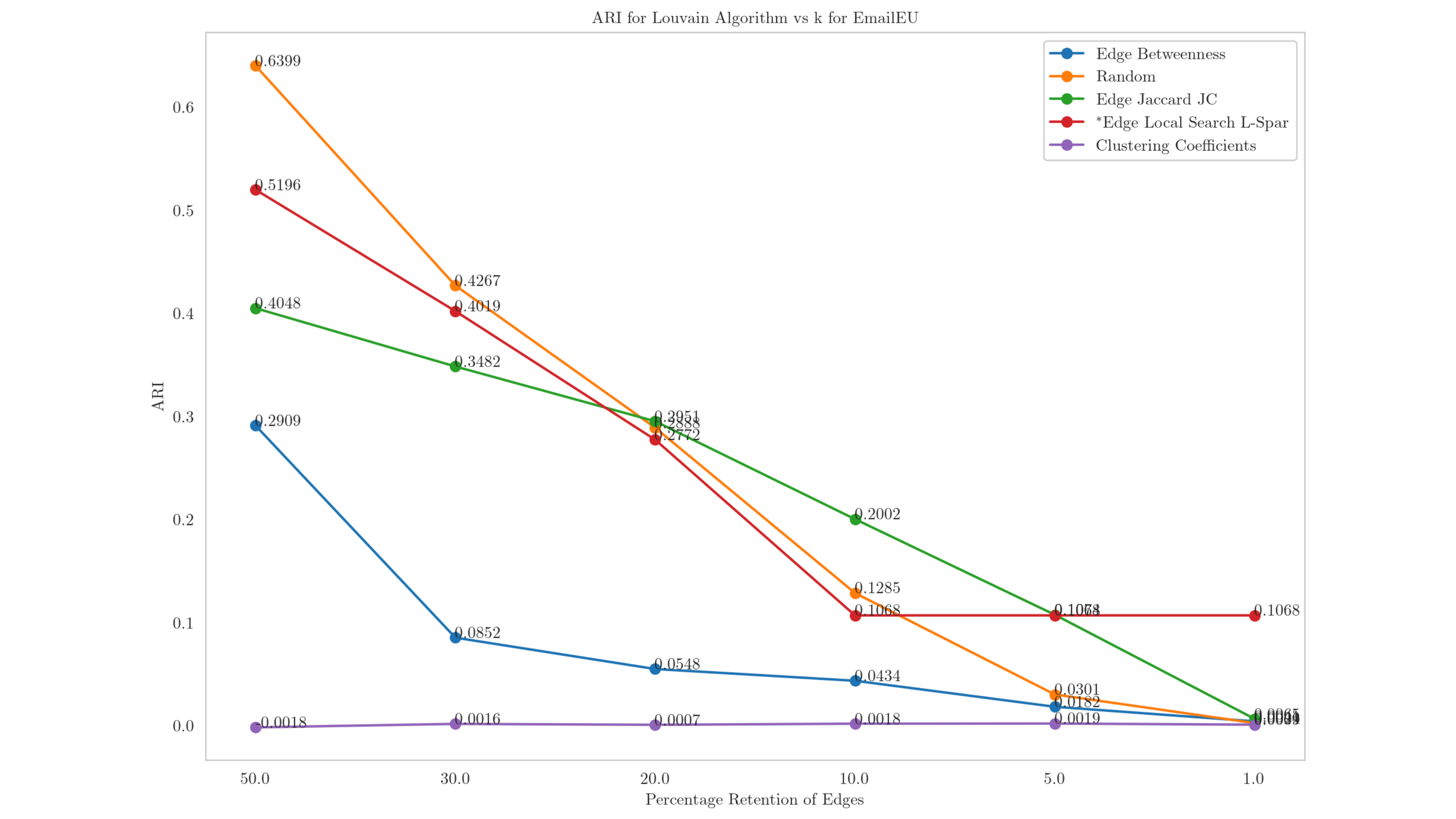




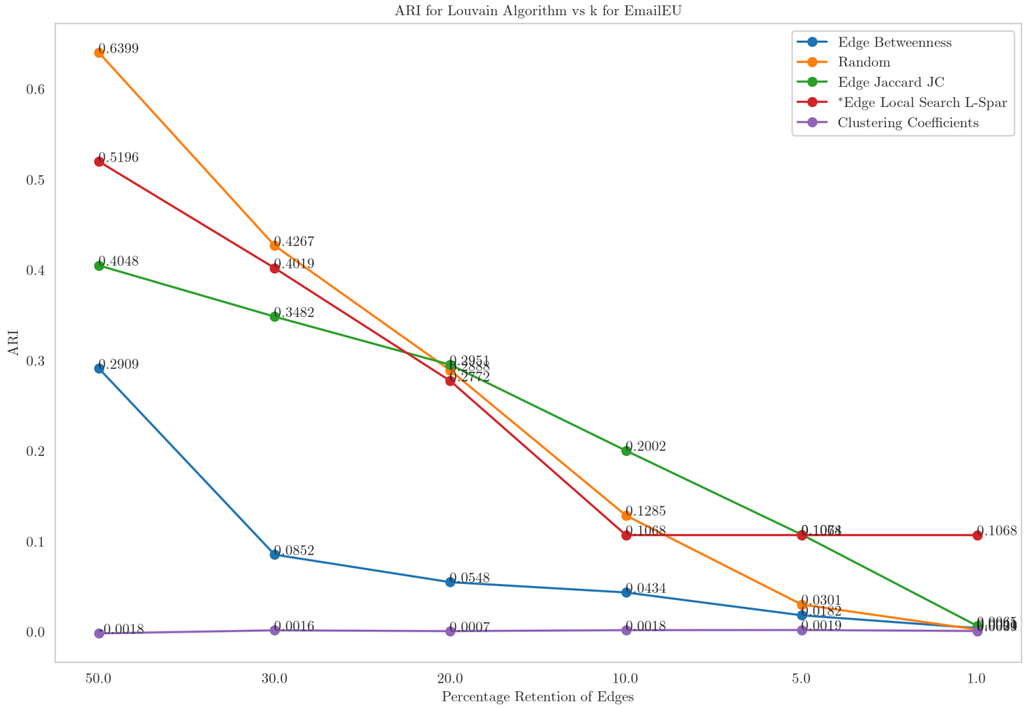


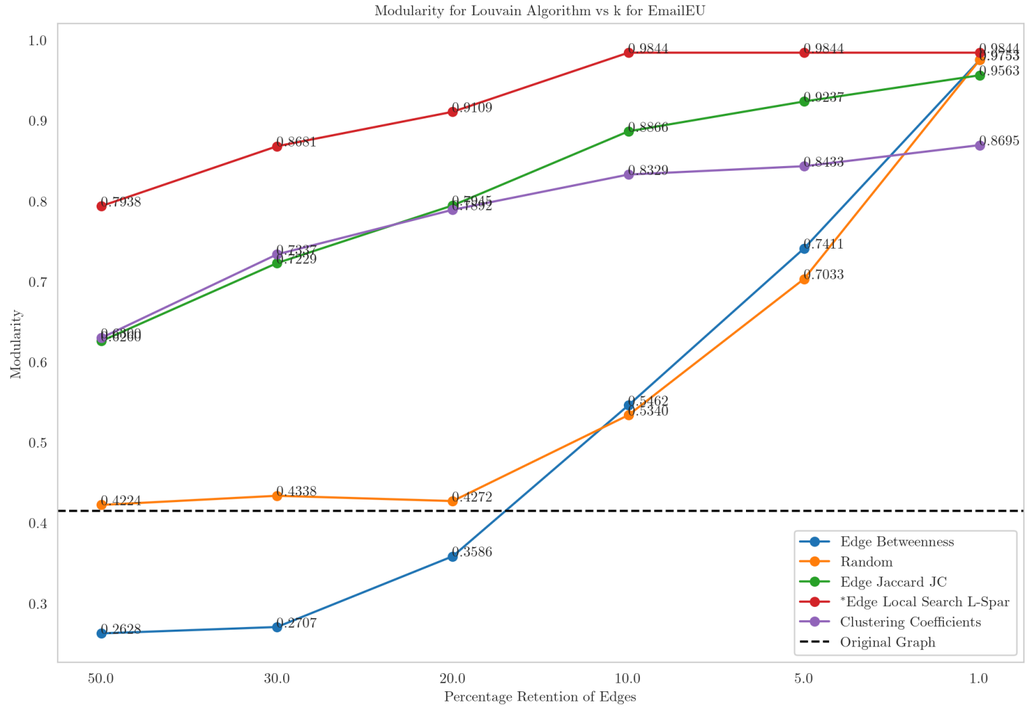


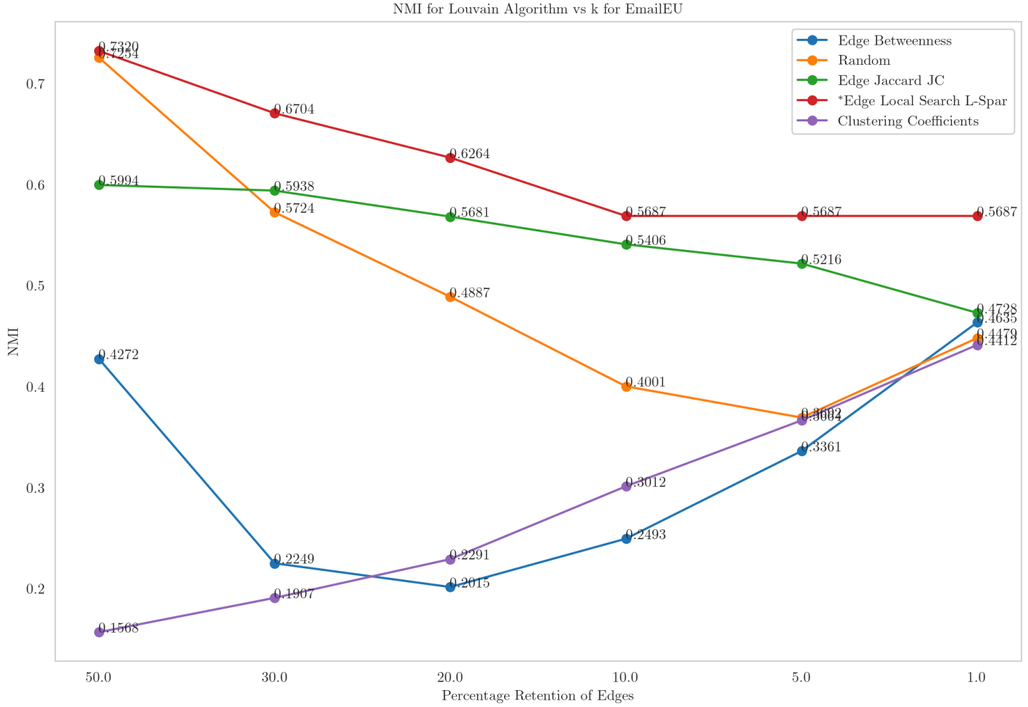
Results for
Email EU Core Network

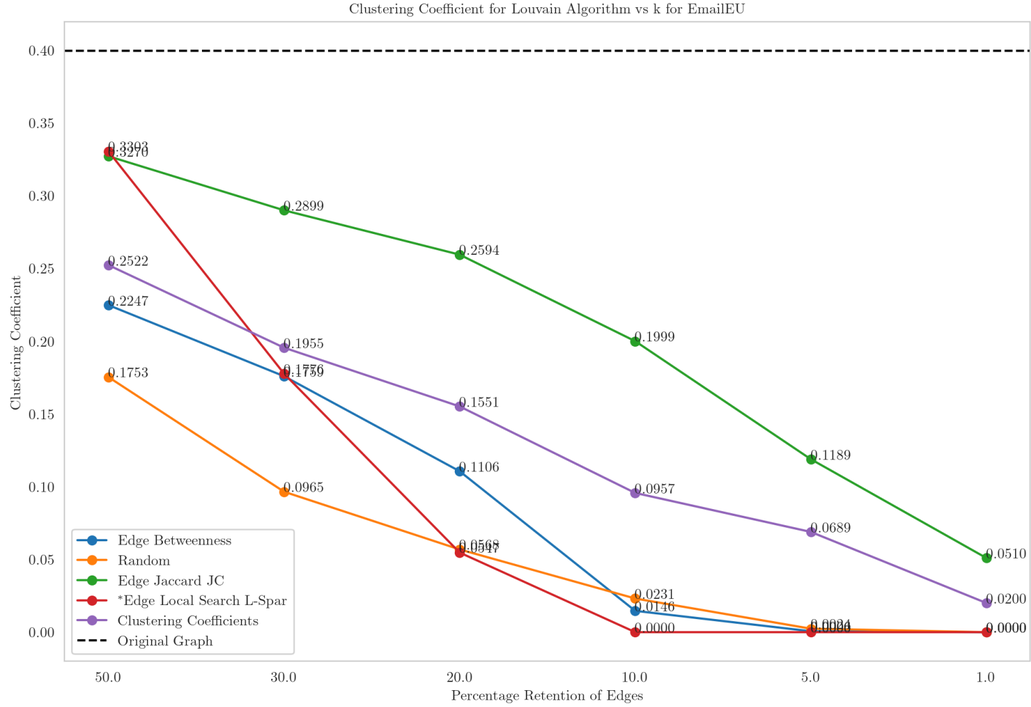

ARI for Louvain Algorithm vs k for EmailEU
Edge Betweenness
Random
Edge Jaccard JC
*Edge Local Search L-Spar
Clustering Coefficients
ARI
Percentage Retention of Edges
0.0
0.1
0.2
0.3
0.4
0.5
0.6
50.0
30.0
20.0
10.0
5.0
1.0
0.6399
0.4267
0.1285
0.0301
0.5196
0.4019
0.2772
0.1068
0.1068
0.4048
0.3482
0.2002
0.2909
0.0852
0.0548
0.0434
0.0182
-0.0018
0.0016
0.0007
0.0018
0.0019

Email Louvain

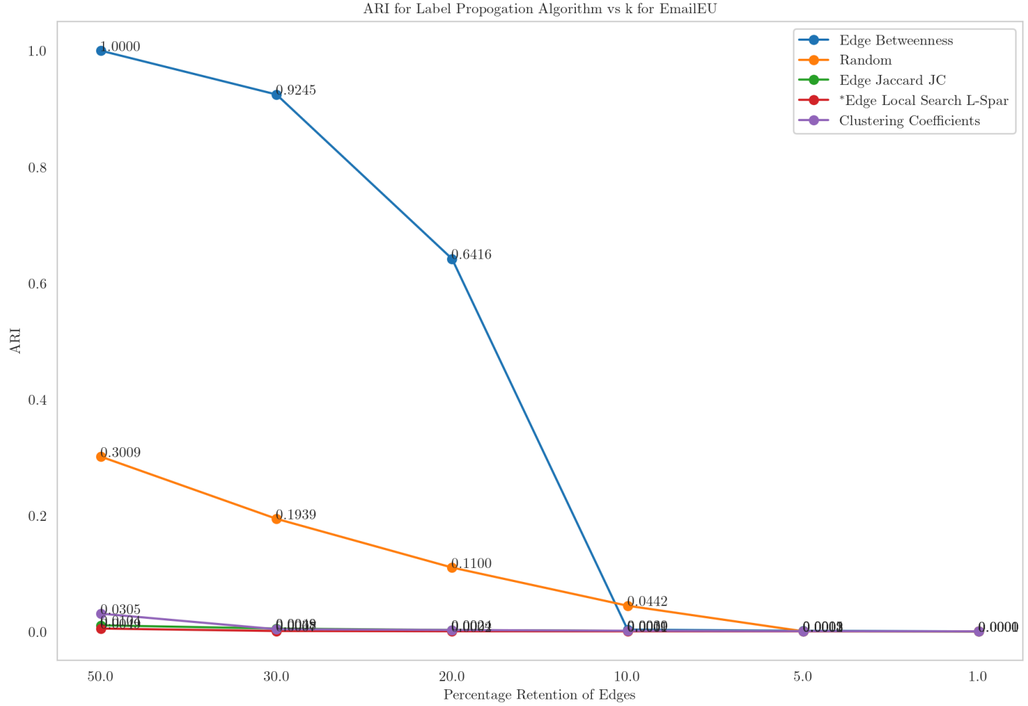


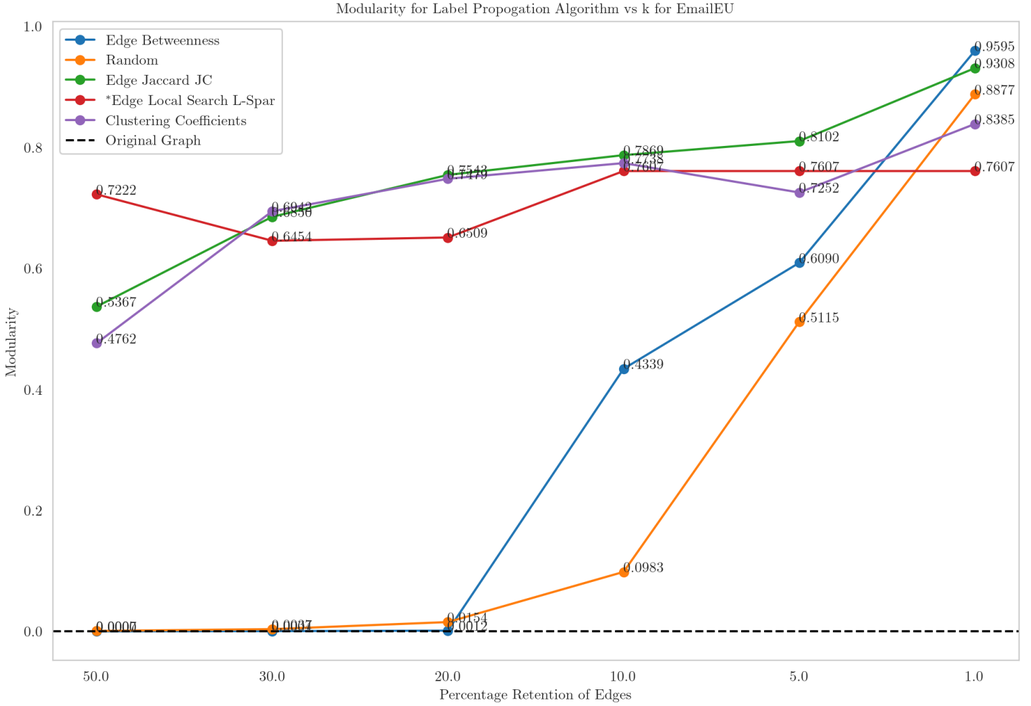


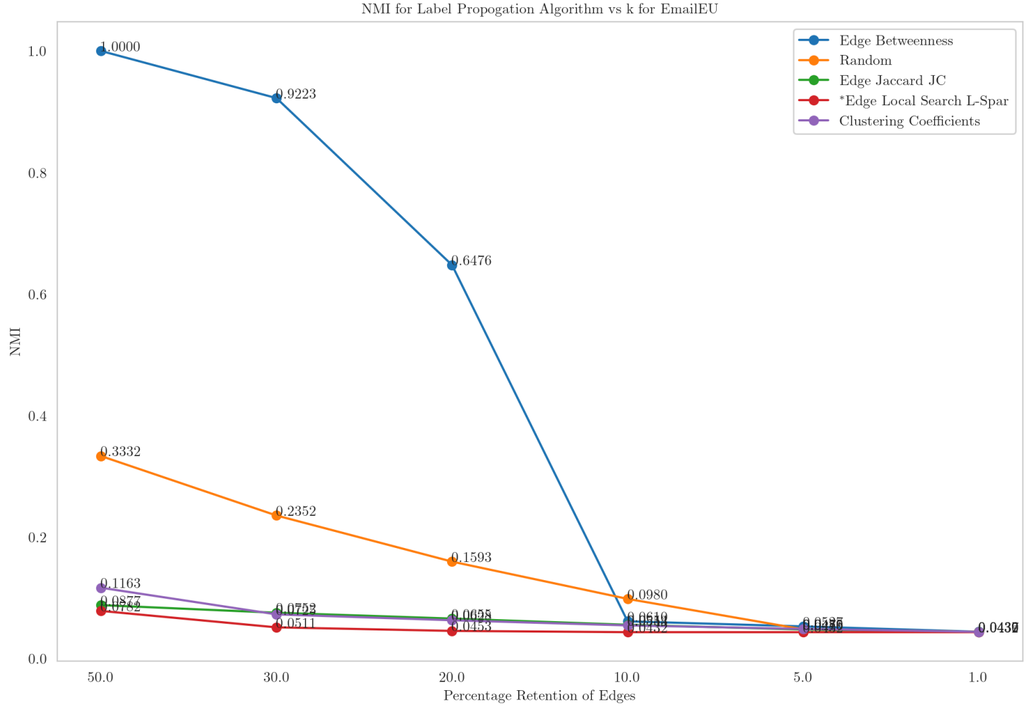


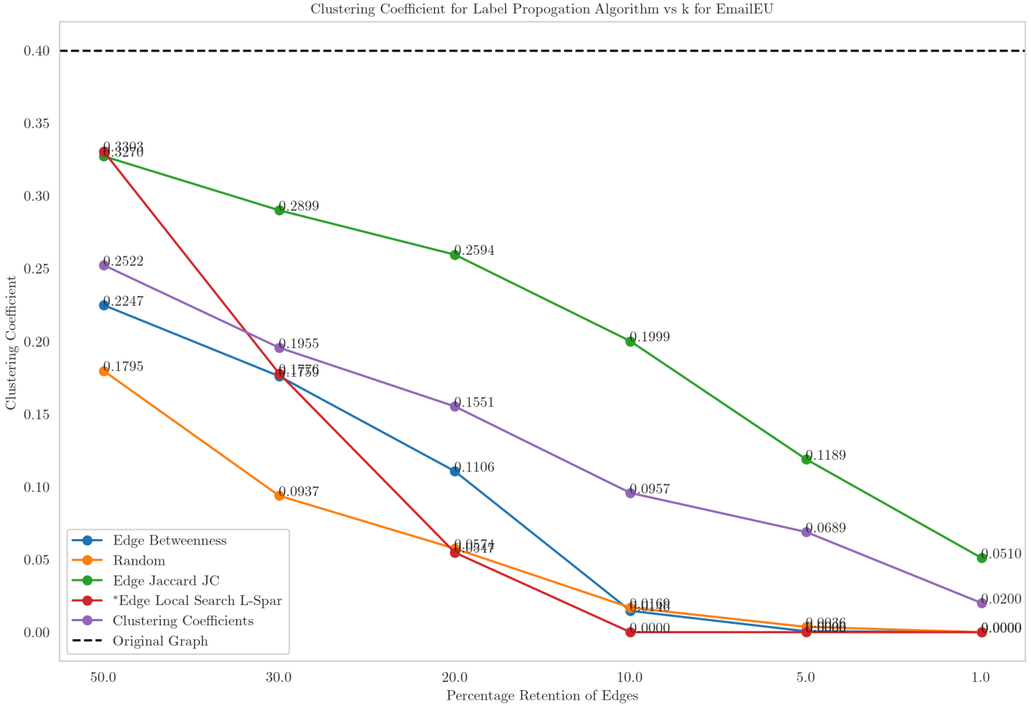


Email LPA

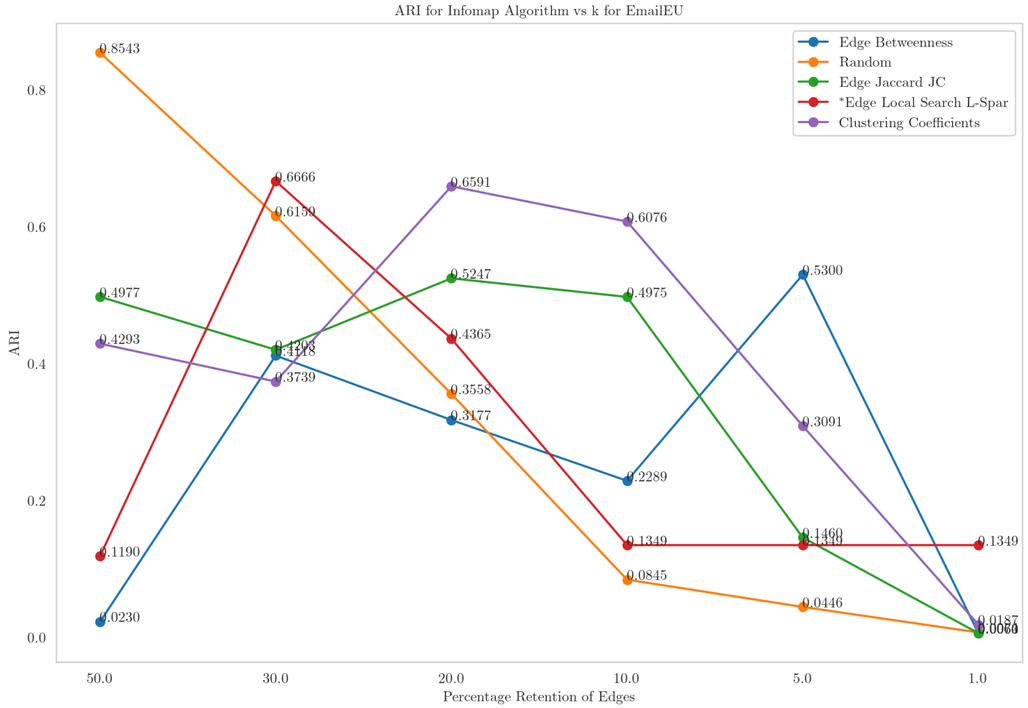

ARI for Label Propogation Algorithm vs k for EmailEU
ARI
Percentage Retention of Edges
Edge Betweenness
Random
Edge Jaccard JC
*Edge Local Search L-Spar
Clustering Coefficients
1.0000
0.9245
0.6416
0.3009
0.1939
0.1100
0.0442
0.0305

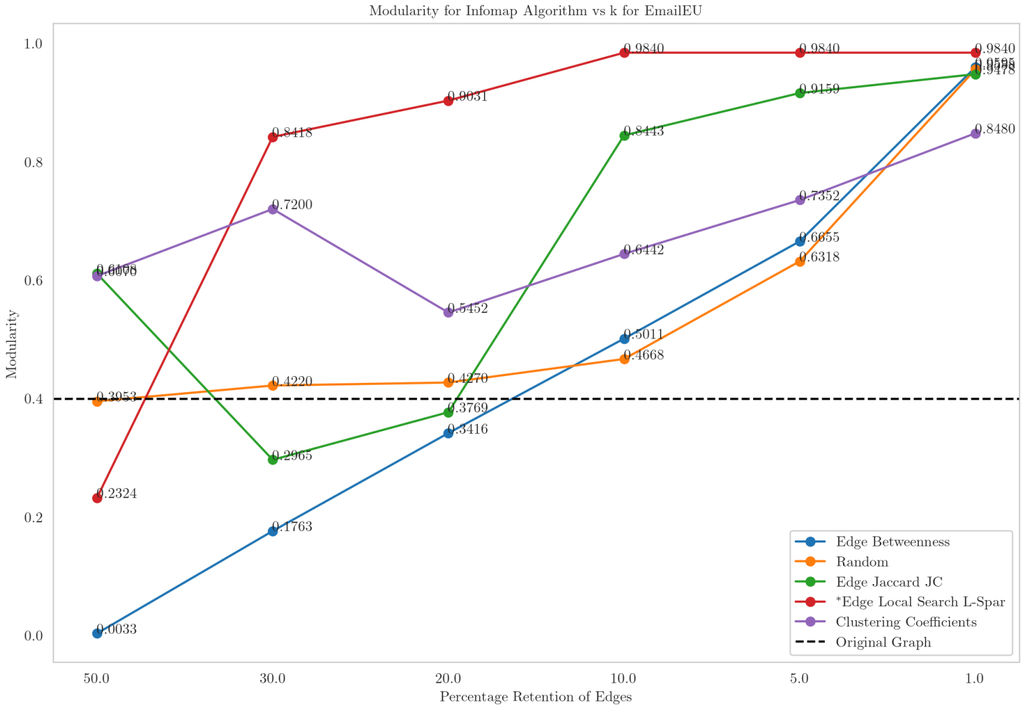

Modularity for Label Propogation Algorithm vs k for EmailEU
Modularity
Percentage Retention of Edges
Edge Betweenness
Random
Edge Jaccard JC
*Edge Local Search L-Spar
Clustering Coefficients
Original Graph
0.9595
0.9308
0.8877
0.8385
0.8102
0.7869
0.7607
0.7252
0.7222
0.6454
0.6509
0.6090
0.5367
0.5115
0.4762
0.4339
0.0983

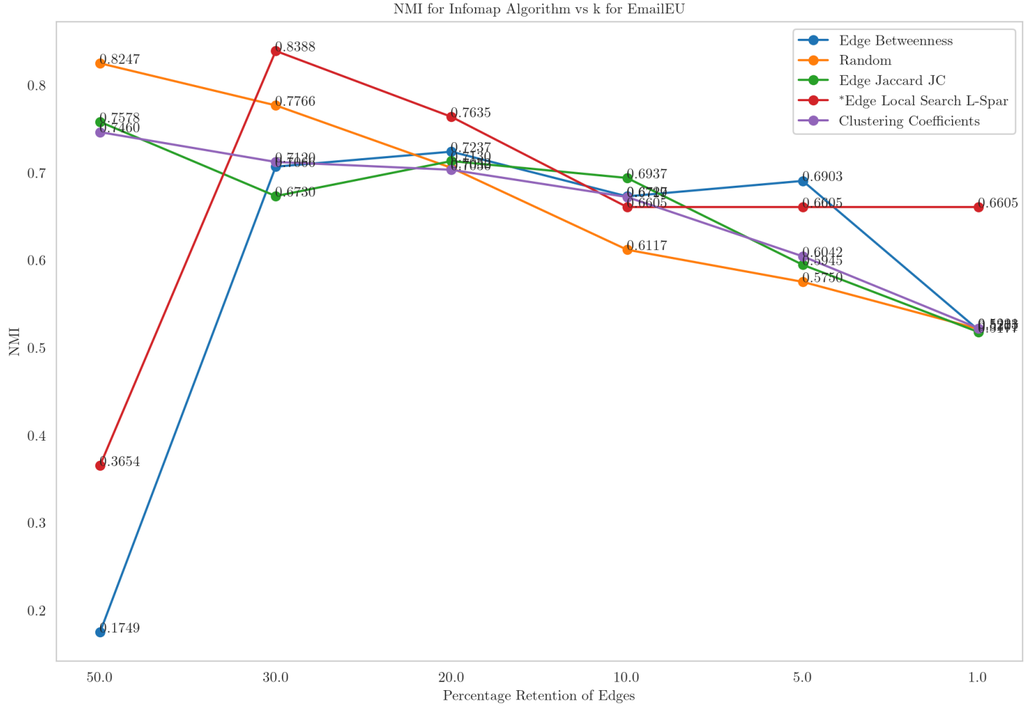

NMI for Label Propogation Algorithm vs k for EmailEU
NMI
Percentage Retention of Edges
Edge Betweenness
Random
Edge Jaccard JC
*Edge Local Search L-Spar
Clustering Coefficients
1.0000
0.9223
0.6476
0.3332
0.2352
0.1593
0.0980
0.1163

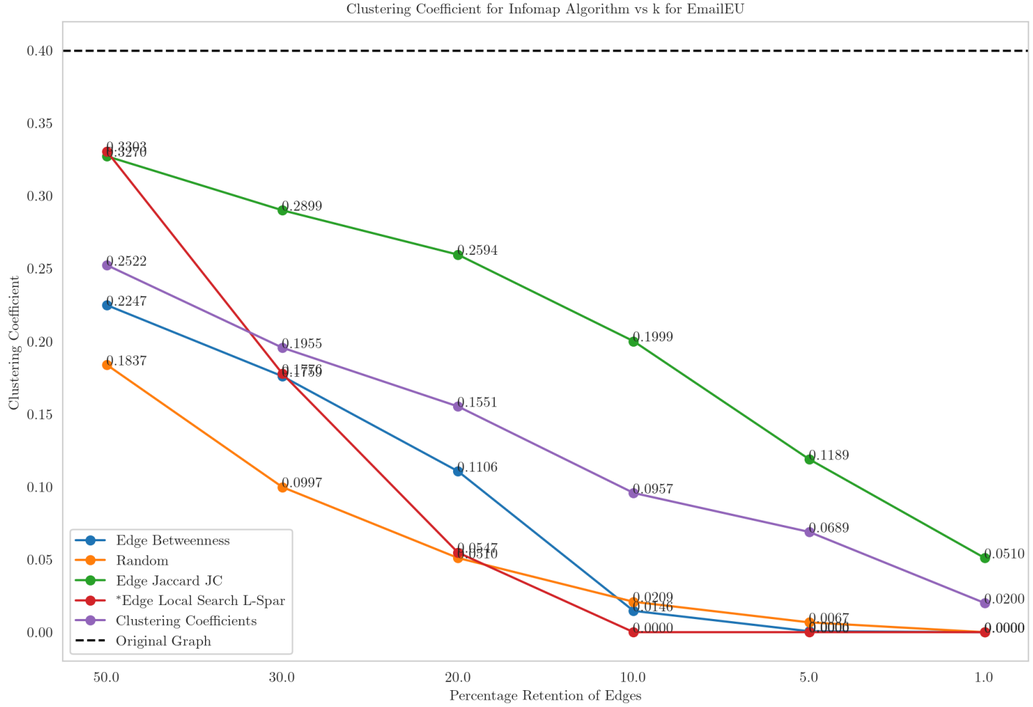

Clustering Coefficient for Label Propogation Algorithm vs k for EmailEU
Clustering Coefficient
Percentage Retention of Edges
Edge Betweenness
Random
Edge Jaccard JC
*Edge Local Search L-Spar
Clustering Coefficients
Original Graph
0.3393
0.3270
0.2899
0.2594
0.1999
0.2522
0.2247
0.1955
0.1795
0.1551
0.1189
0.1106
0.0957
0.0937
0.0689
0.0510
0.0200
0.0000

Email InfoMap

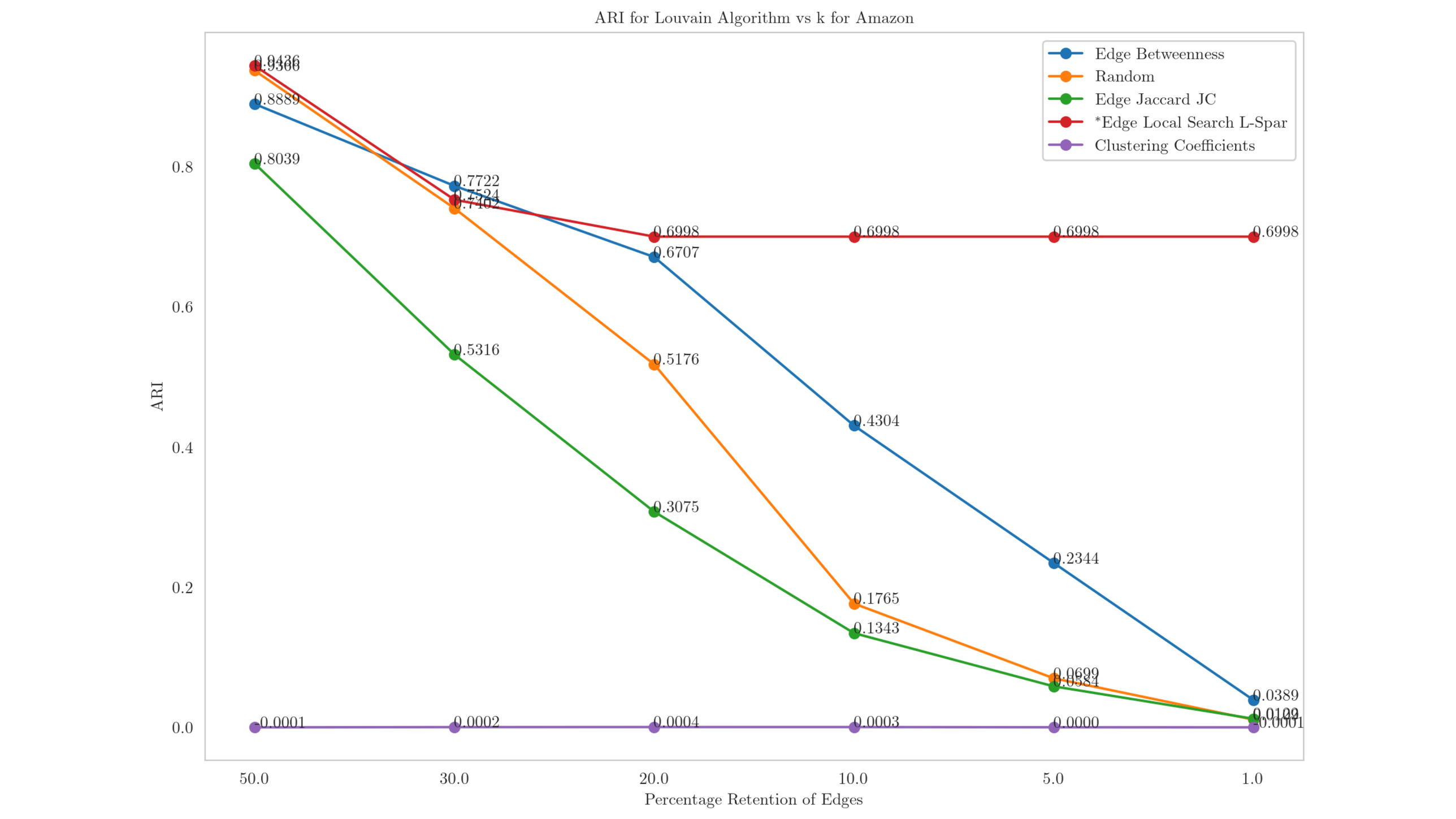




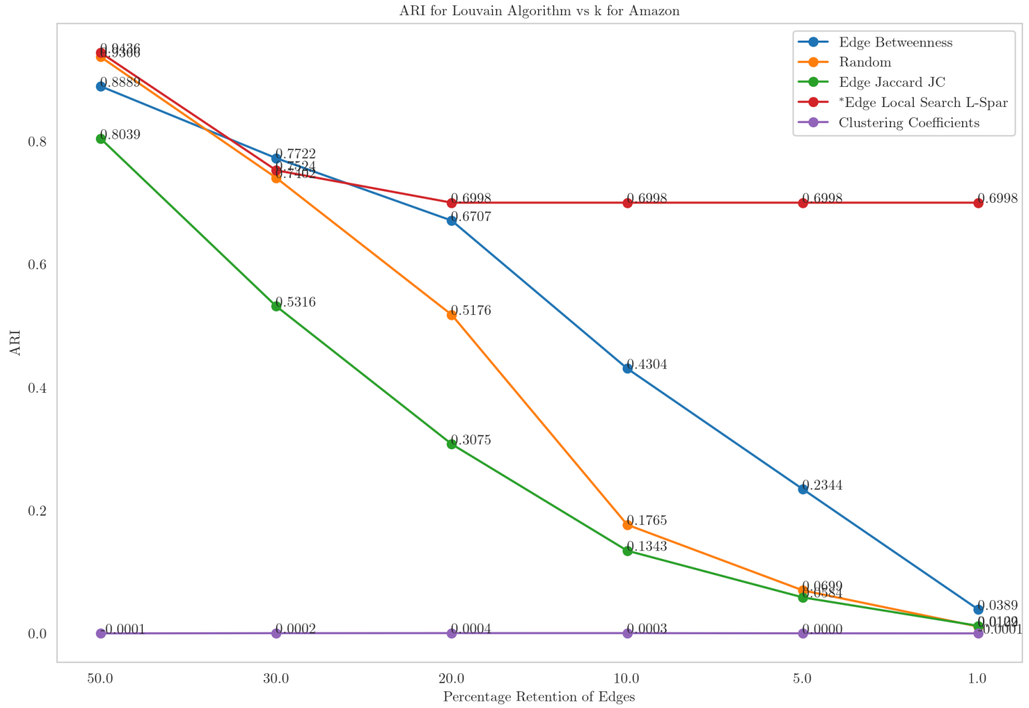


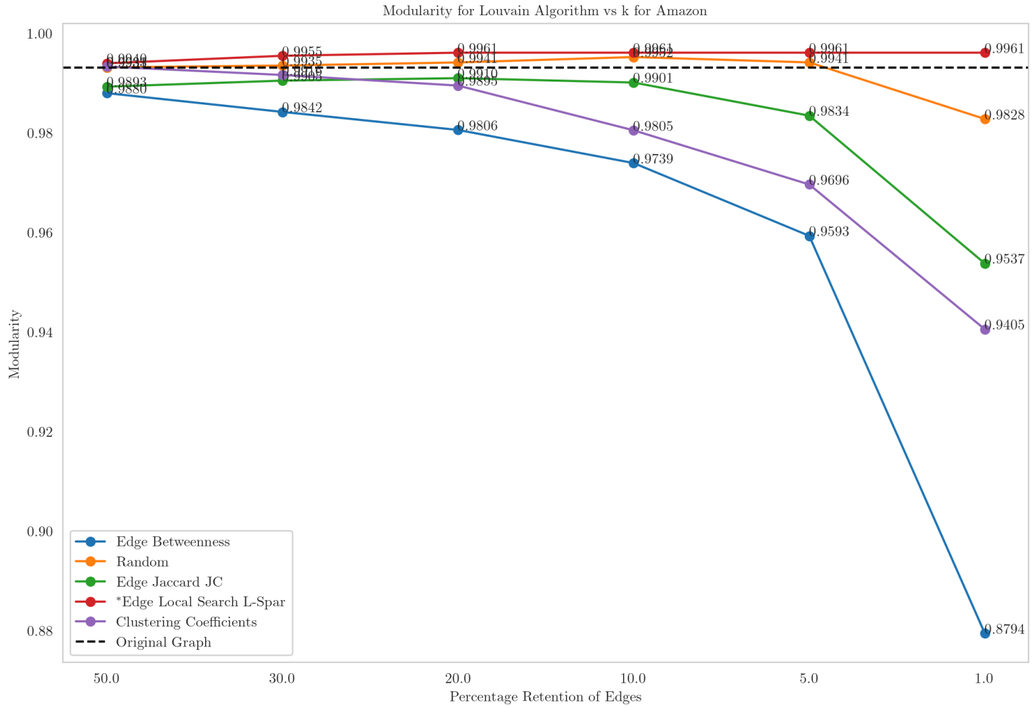


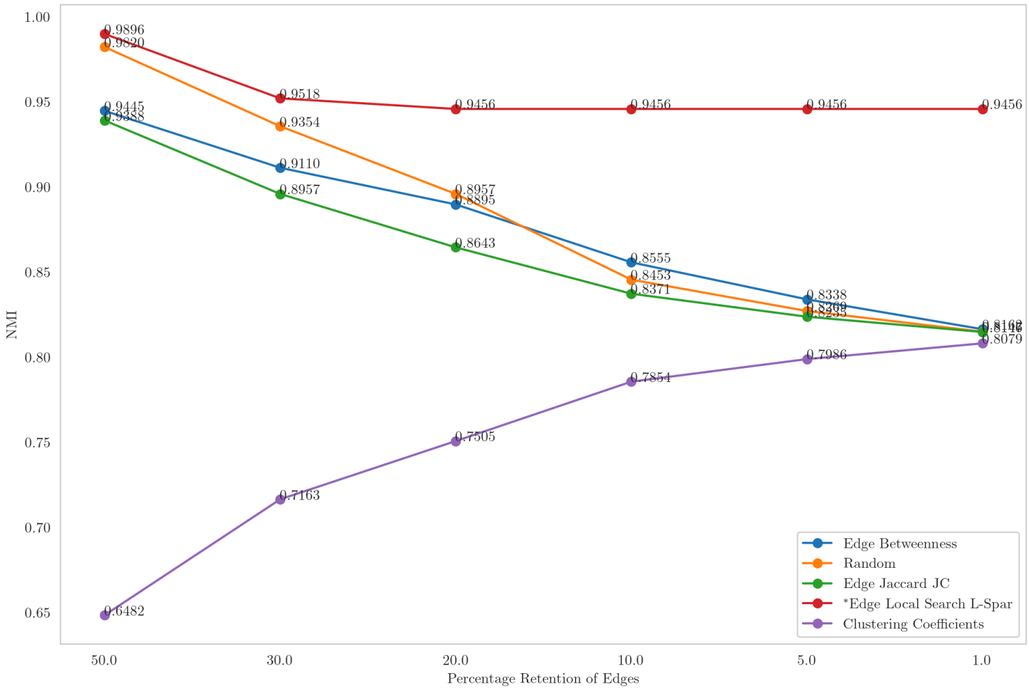
Results for
Amazon Co-Purchase Network

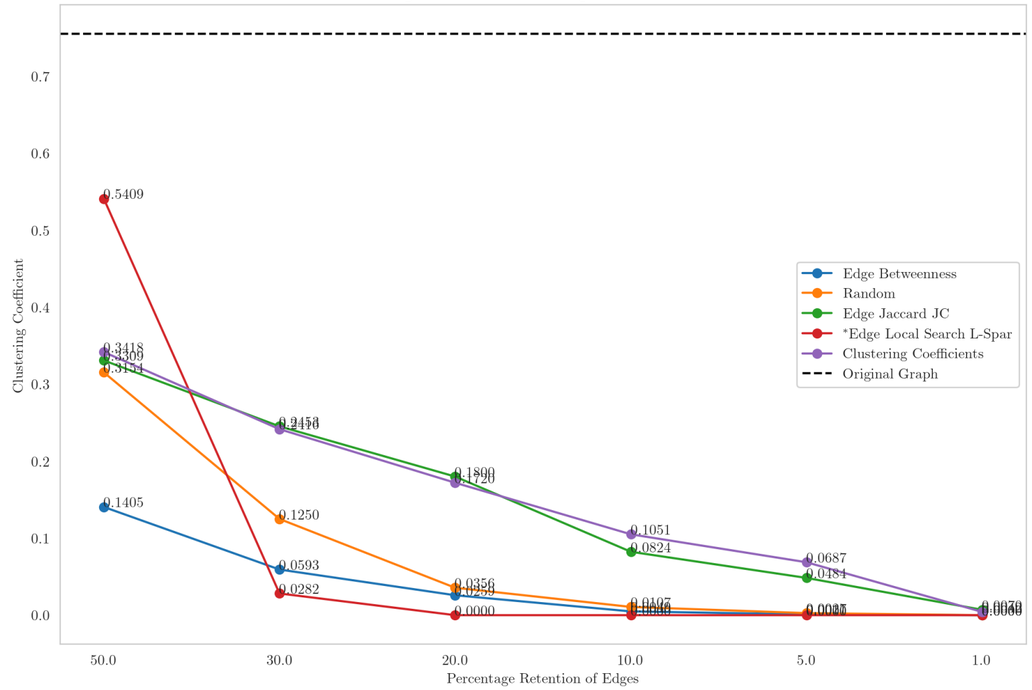

ARI for Louvain Algorithm vs k for Amazon
Edge Betweenness
Random
Edge Jaccard JC
*Edge Local Search L-Spar
Clustering Coefficients
ARI
Percentage Retention of Edges
0.0
0.2
0.4
0.6
0.8
50.0
30.0
20.0
10.0
5.0
1.0
0.9436
0.9300
0.8889
0.8039
0.7722
0.7524
0.7402
0.6998
0.6998
0.6998
0.6998
0.6707
0.5316
0.5176
0.4304
0.3075
0.2344
0.1765
0.1343
0.0699
0.0584
0.0389
-0.0001
0.0002
0.0004
0.0003
0.0000
-0.0001

Amazon Louvain

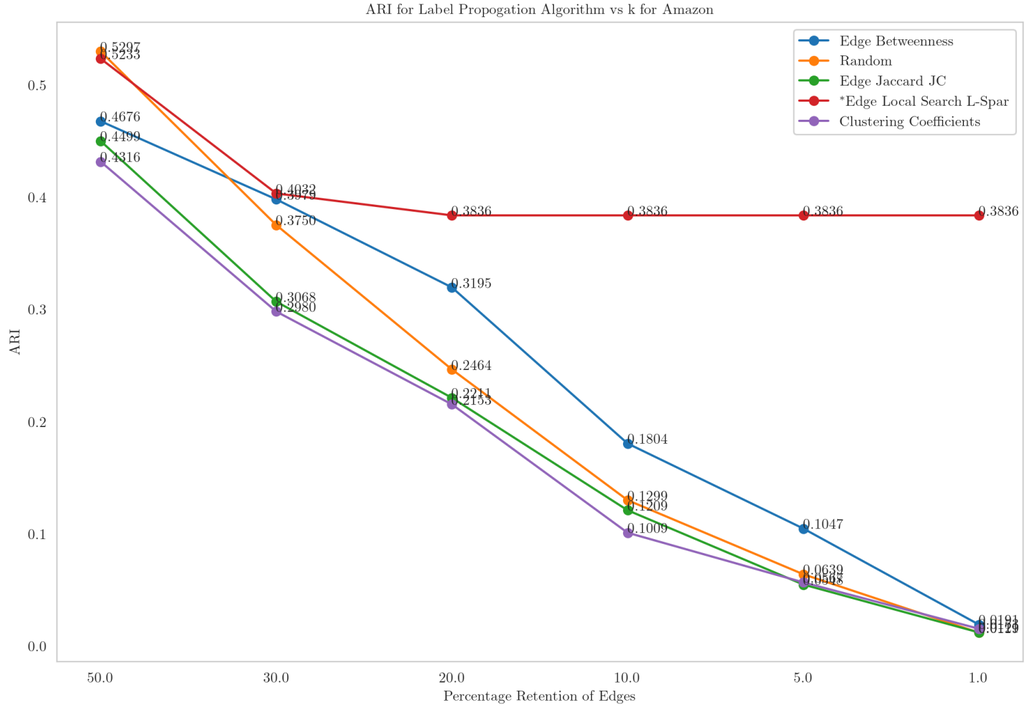


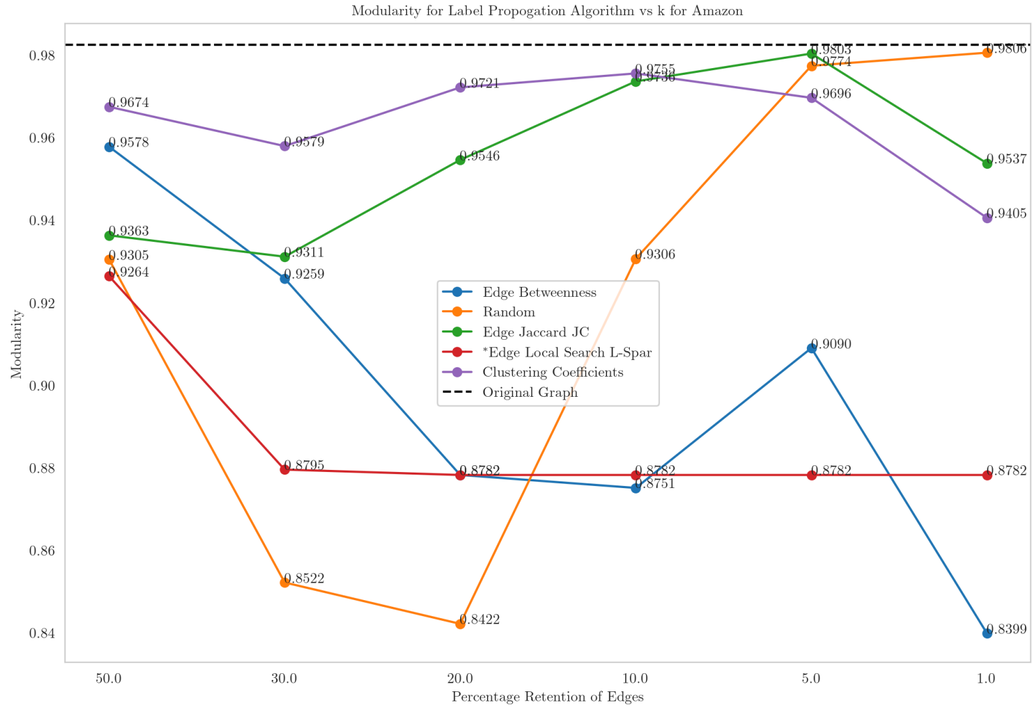


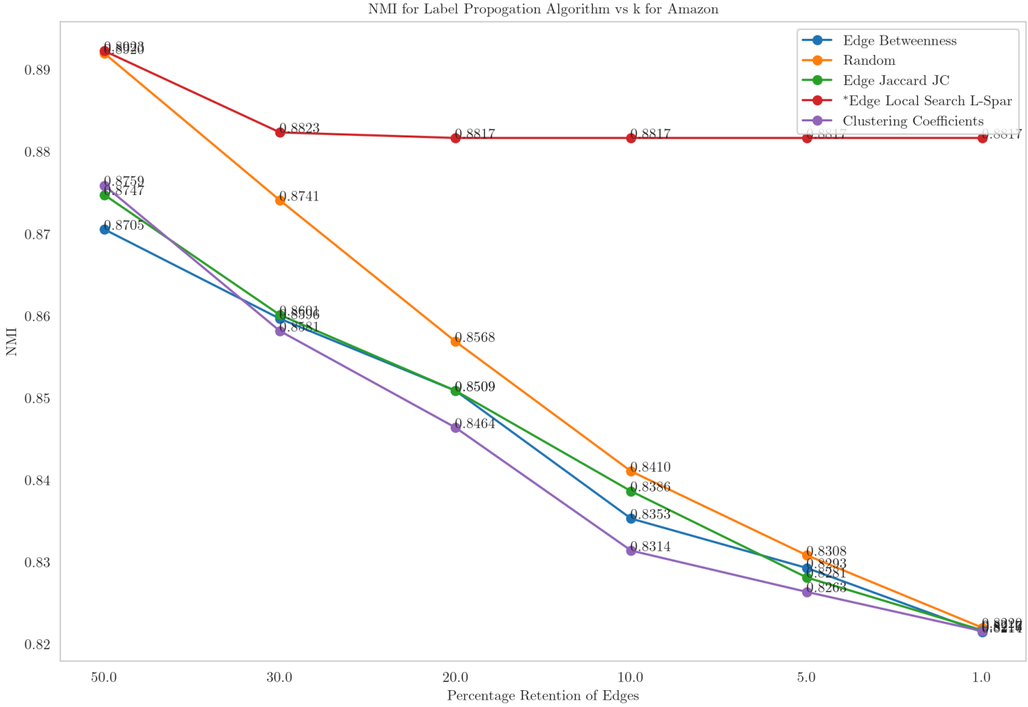


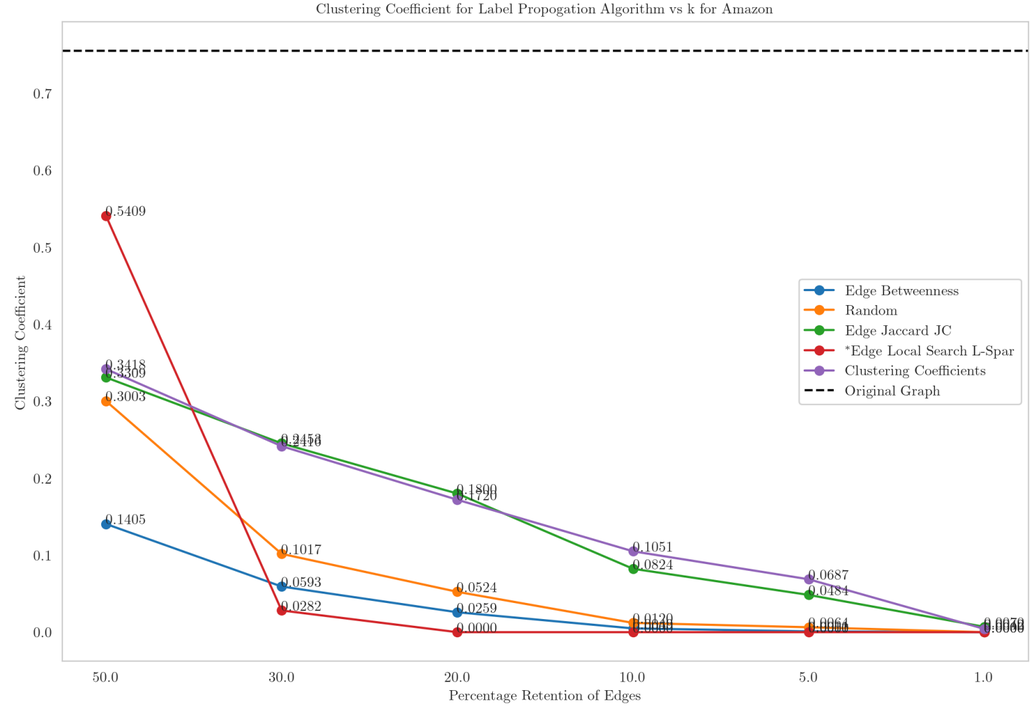


Amazon LPA

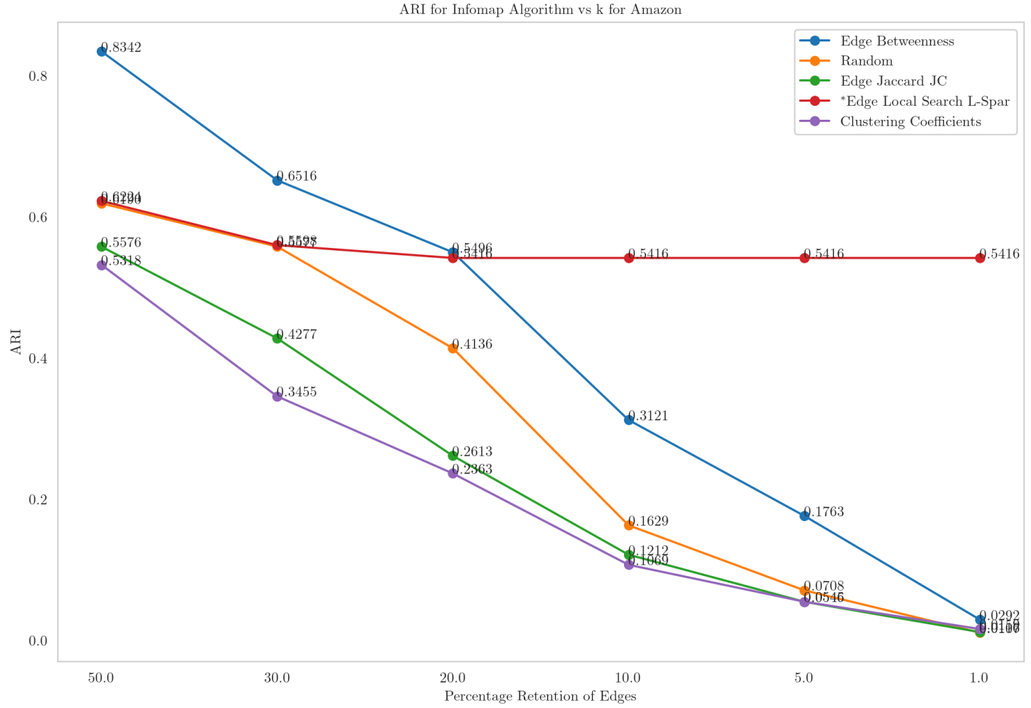

ARI for Label Propogation Algorithm vs k for Amazon
Edge Betweenness
Random
Edge Jaccard JC
*Edge Local Search L-Spar
Clustering Coefficients
ARI
Percentage Retention of Edges
50.0
30.0
20.0
10.0
5.0
1.0
0.0
0.1
0.2
0.3
0.4
0.5
0.5297
0.5235
0.4676
0.4499
0.4316
0.4022
0.3750
0.3836
0.3836
0.3836
0.3836
0.3195
0.3068
0.2980
0.2464
0.2211
0.2153
0.1804
0.1299
0.1209
0.1009
0.1047
0.0639
0.0191

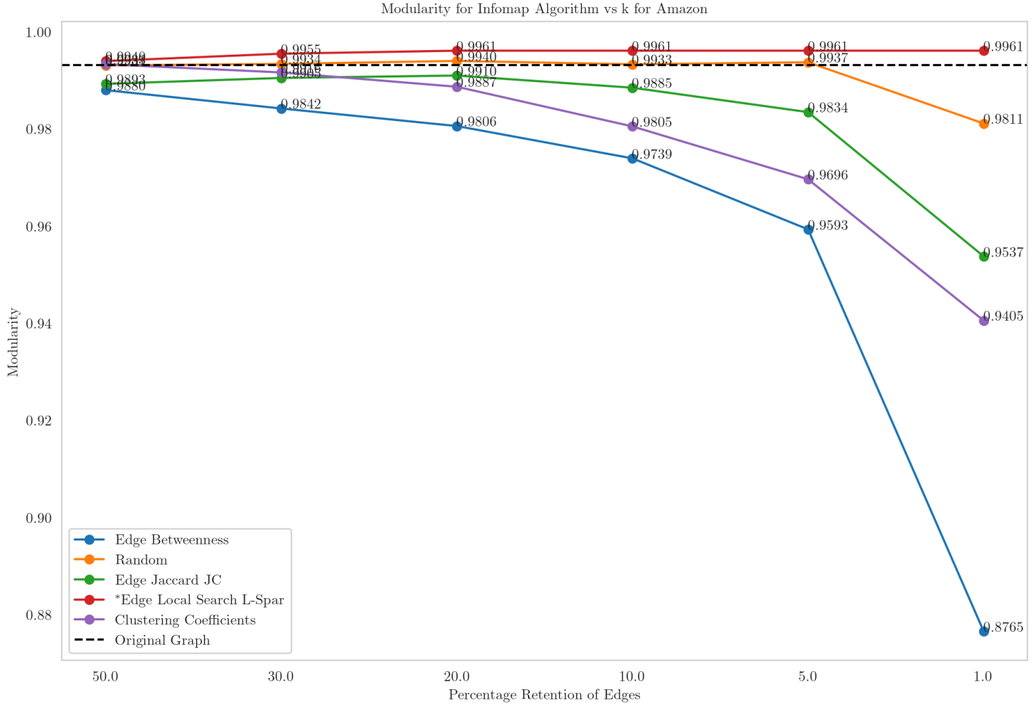

Modularity for Label Propogation Algorithm vs k for Amazon
Edge Betweenness
Random
Edge Jaccard JC
*Edge Local Search L-Spar
Clustering Coefficients
Original Graph
Modularity
Percentage Retention of Edges
50.0
30.0
20.0
10.0
5.0
1.0
0.84
0.86
0.88
0.90
0.92
0.94
0.96
0.98
0.9674
0.9578
0.9579
0.9721
0.9755
0.9696
0.9803
0.9774
0.9806
0.9537
0.9546
0.9363
0.9311
0.9305
0.9264
0.9259
0.9306
0.9405
0.8795
0.8782
0.8751
0.8782
0.8782
0.8782
0.9090
0.8522
0.8422
0.8399

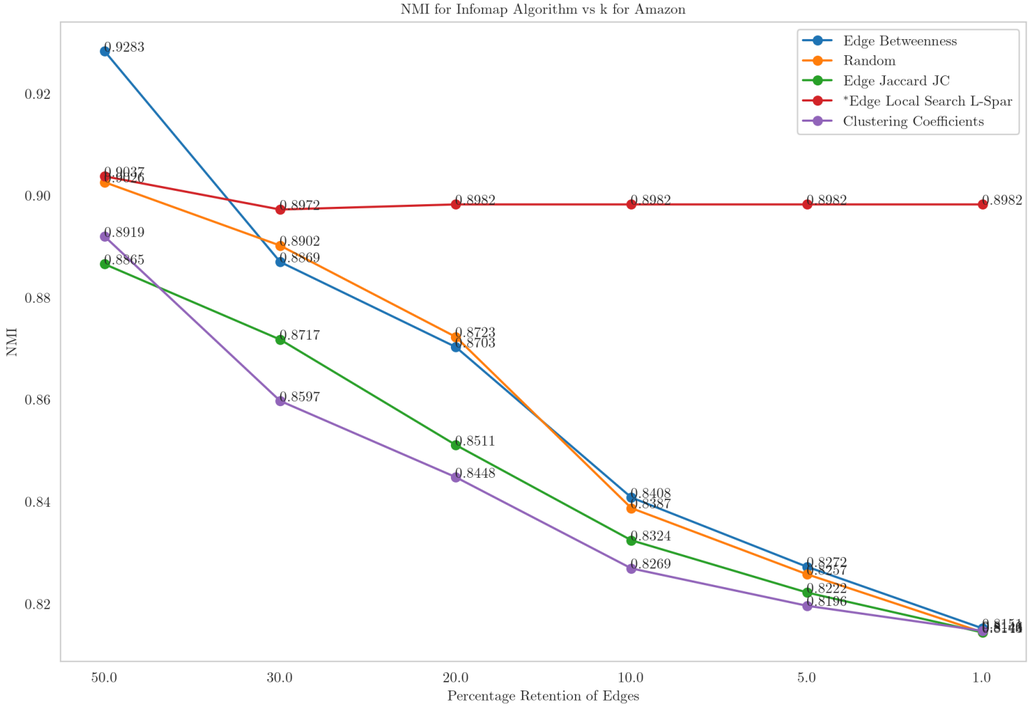

NMI for Label Propogation Algorithm vs k for Amazon
Edge Betweenness
Random
Edge Jaccard JC
*Edge Local Search L-Spar
Clustering Coefficients
NMI
Percentage Retention of Edges
50.0
30.0
20.0
10.0
5.0
1.0
0.82
0.83
0.84
0.85
0.86
0.87
0.88
0.89
0.8759
0.8747
0.8705
0.8823
0.8817
0.8817
0.8741
0.8568
0.8509
0.8464
0.8410
0.8336
0.8353
0.8314
0.8308
0.8263

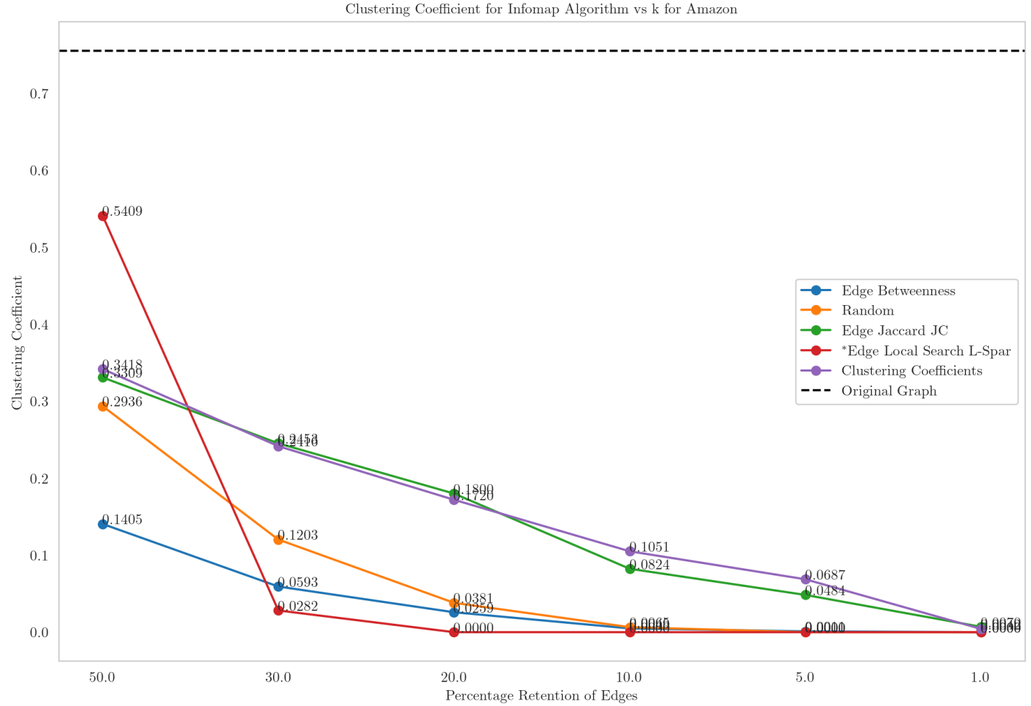

Clustering Coefficient for Label Propogation Algorithm vs k for Amazon
Edge Betweenness
Random
Edge Jaccard JC
*Edge Local Search L-Spar
Clustering Coefficients
Original Graph
Clustering Coefficient
Percentage Retention of Edges
50.0
30.0
20.0
10.0
5.0
1.0
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.5409
0.3418
0.3309
0.3003
0.1405
0.1017
0.0593
0.0524
0.0282
0.0259
0.0000
0.1051
0.0824
0.0687
0.0484

Amazon InfoMap

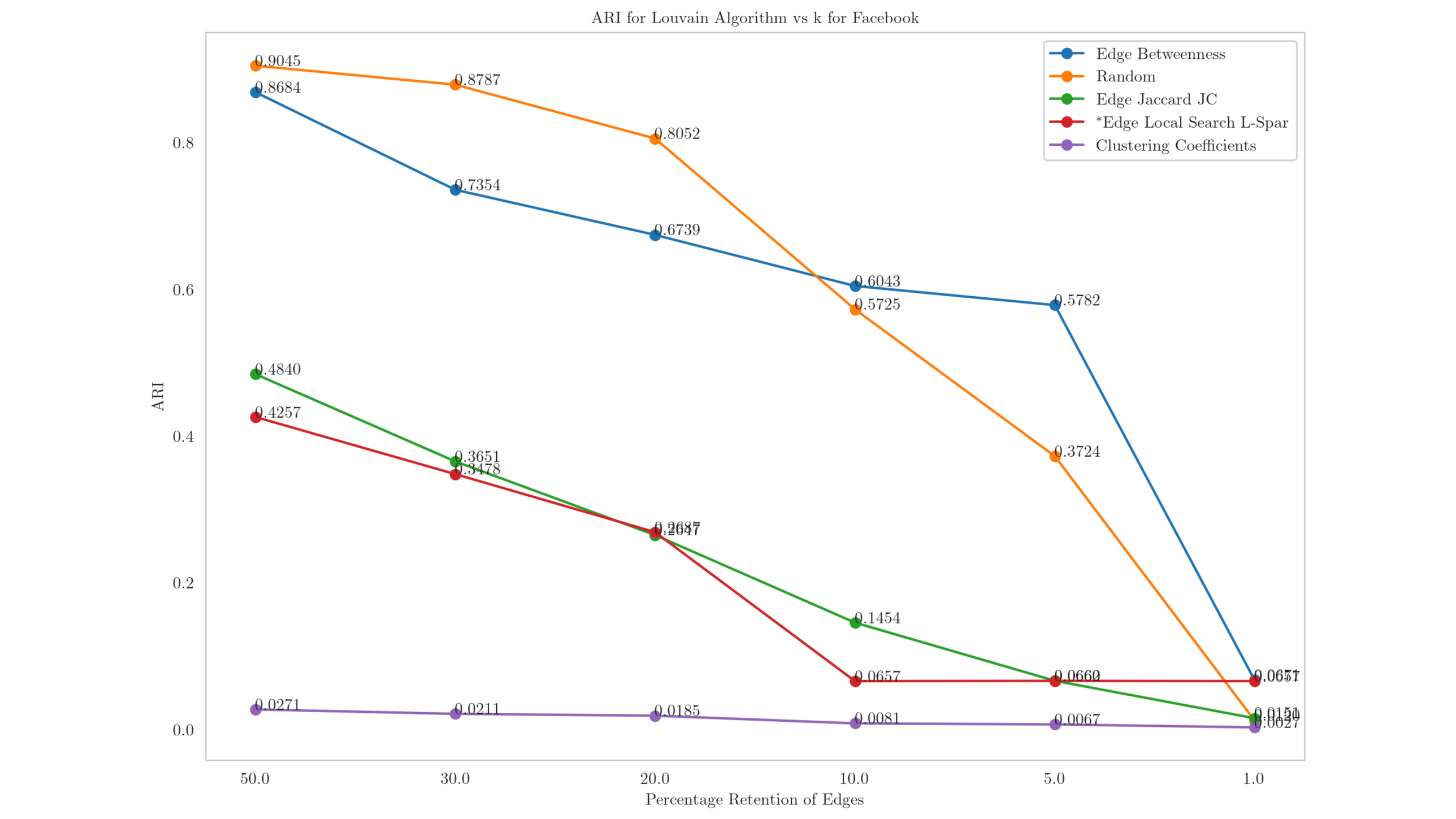




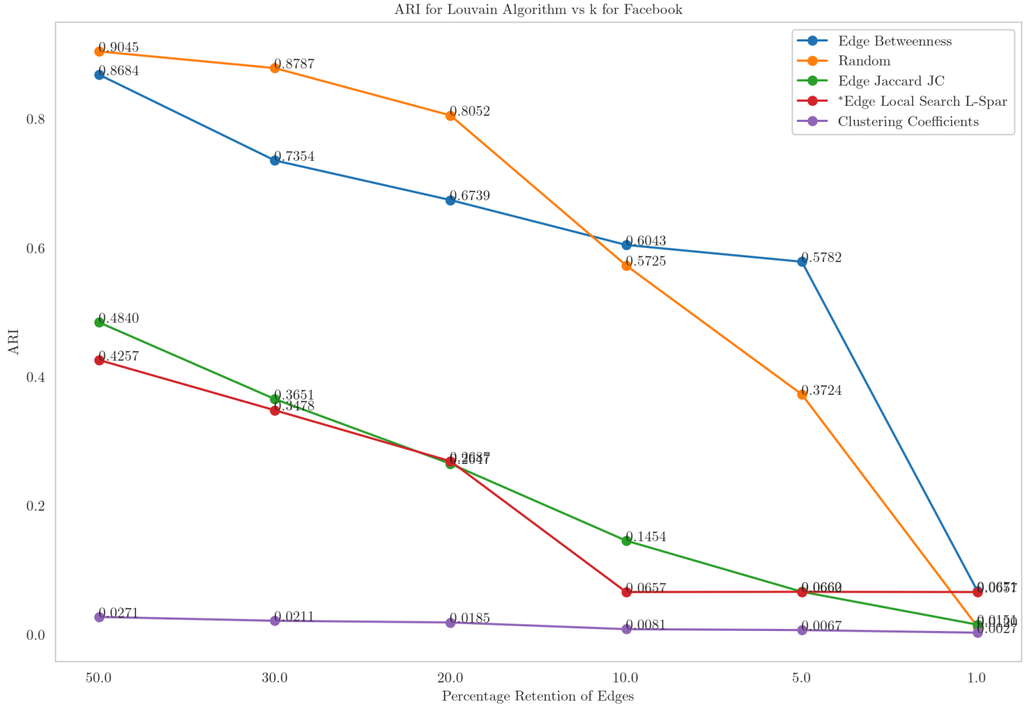


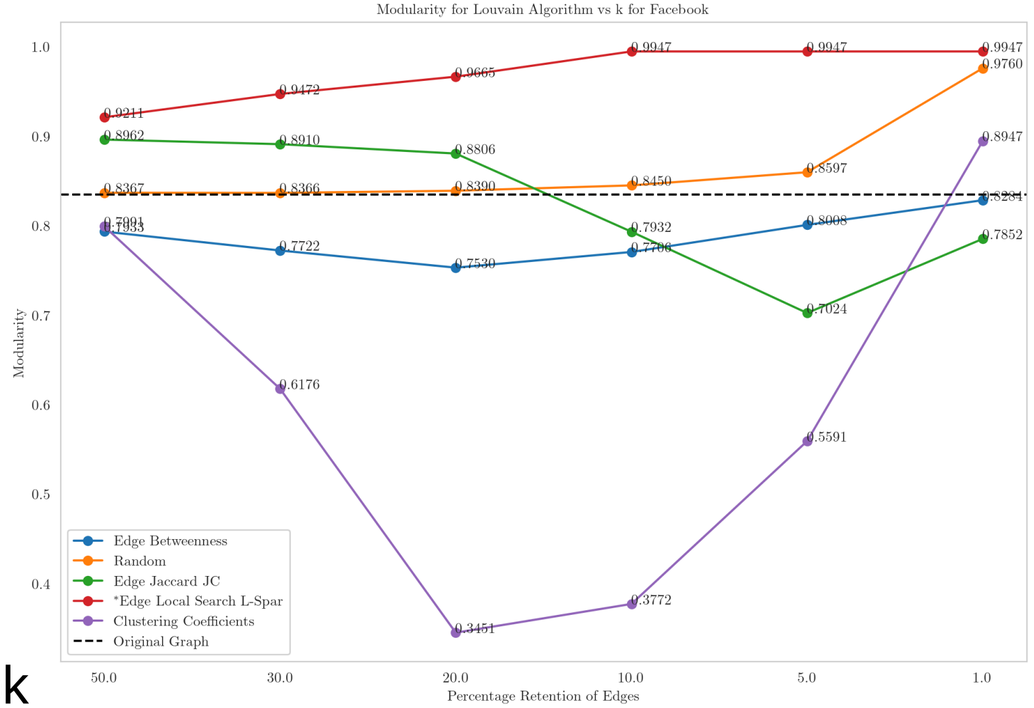


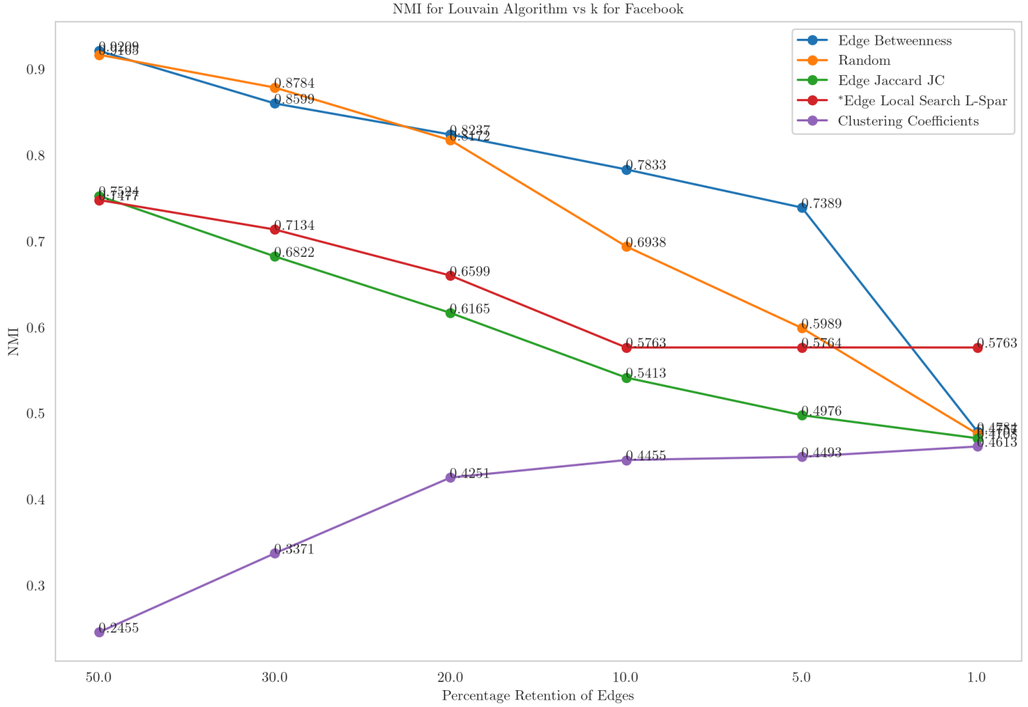
Results for
Facebook Social Network

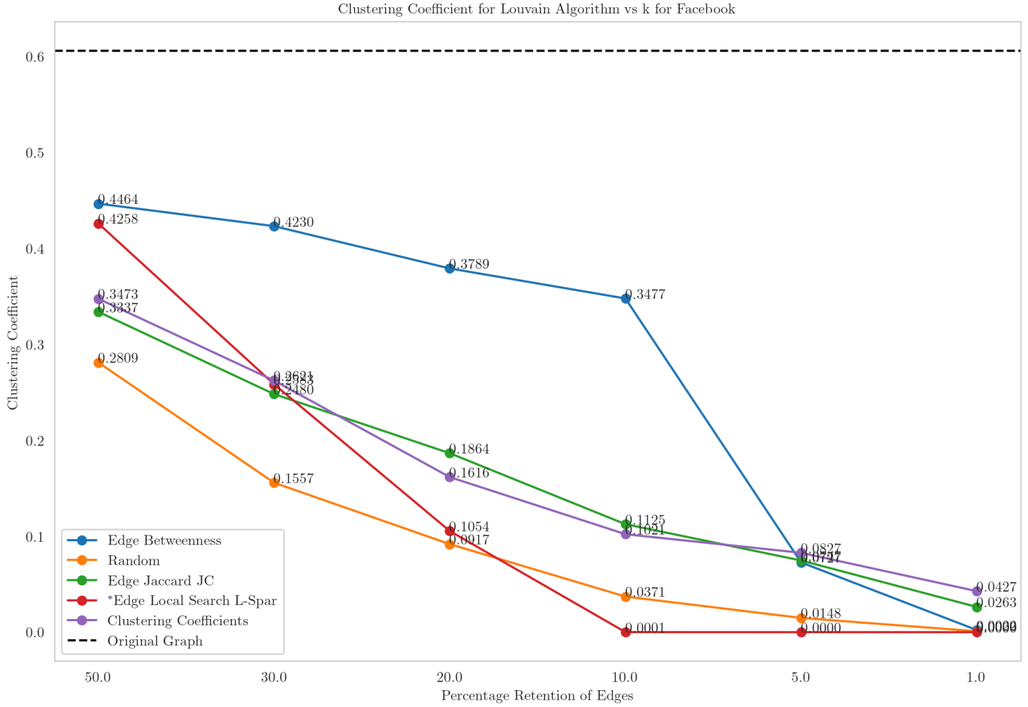

ARI for Louvain Algorithm vs k for Facebook
Edge Betweenness
Random
Edge Jaccard JC
*Edge Local Search L-Spar
Clustering Coefficients
ARI
Percentage Retention of Edges
0.0
0.2
0.4
0.6
0.8
50.0
30.0
20.0
10.0
5.0
1.0
0.9045
0.8787
0.8052
0.5725
0.3724
0.8684
0.7354
0.6739
0.6043
0.5782
0.4840
0.3651
0.1454
0.4257
0.3478
0.0657
0.0660
0.0271
0.0211
0.0185
0.0081
0.0067

Facebook Louvain

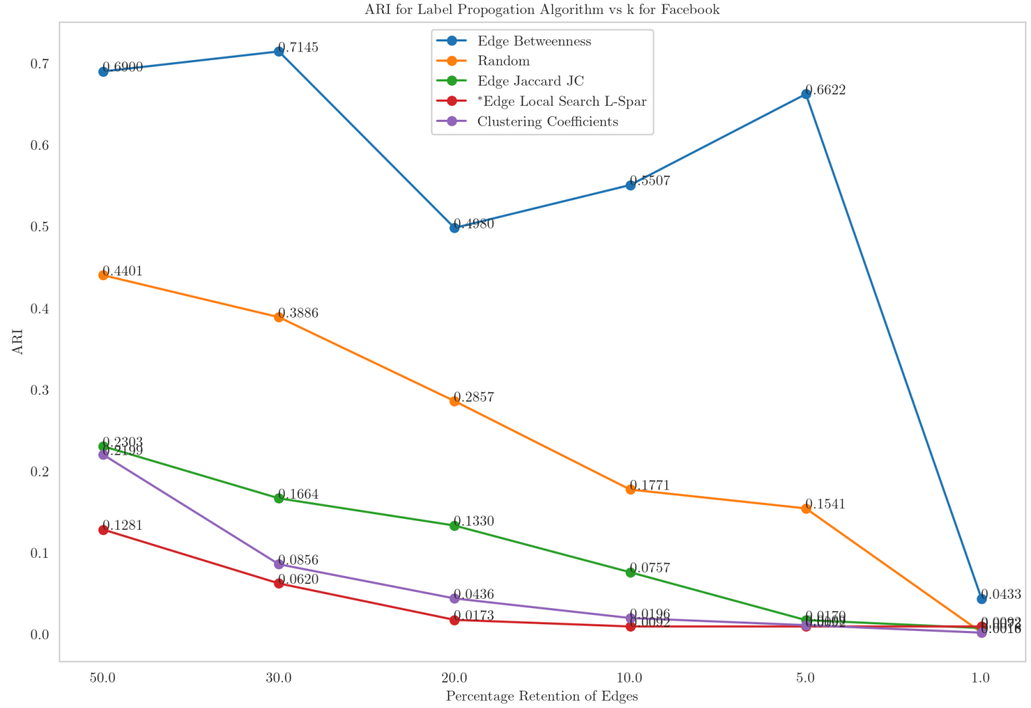


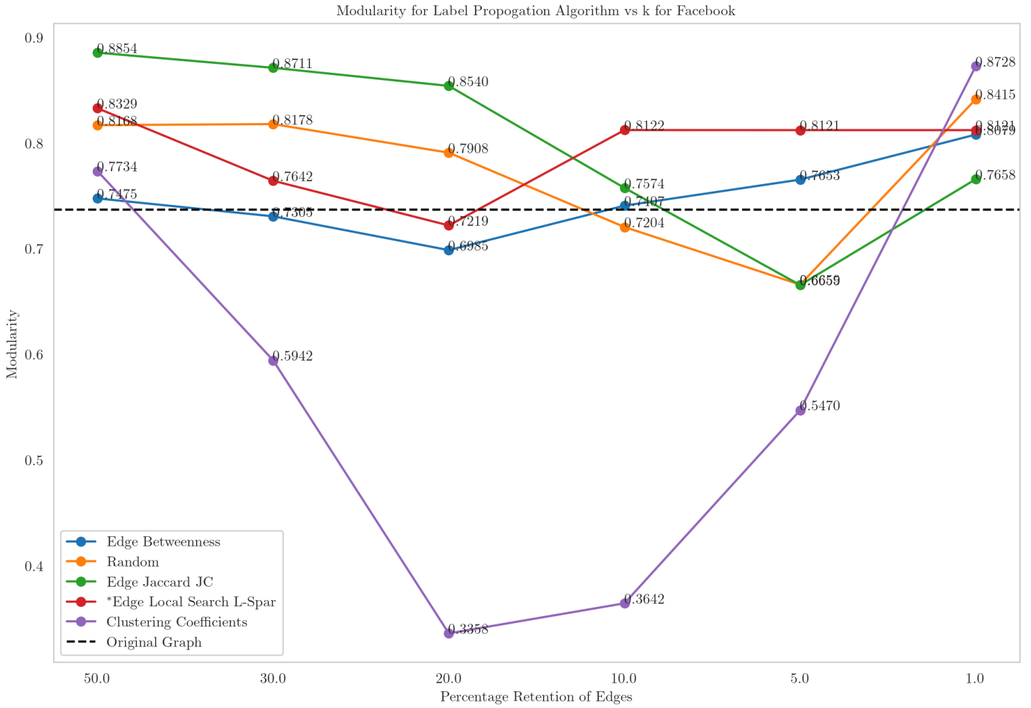


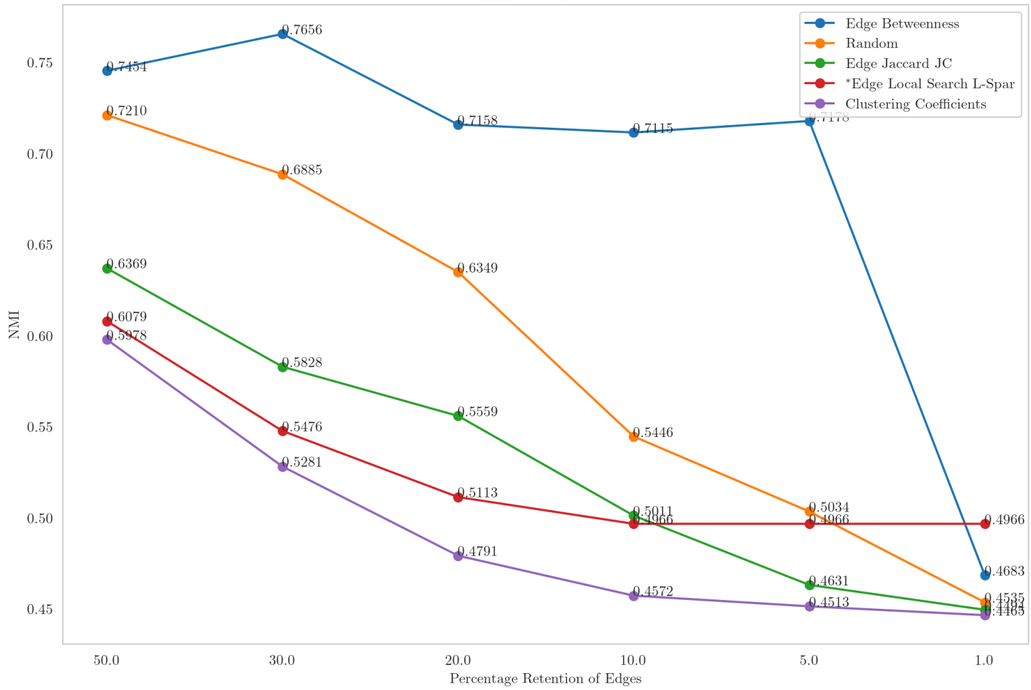


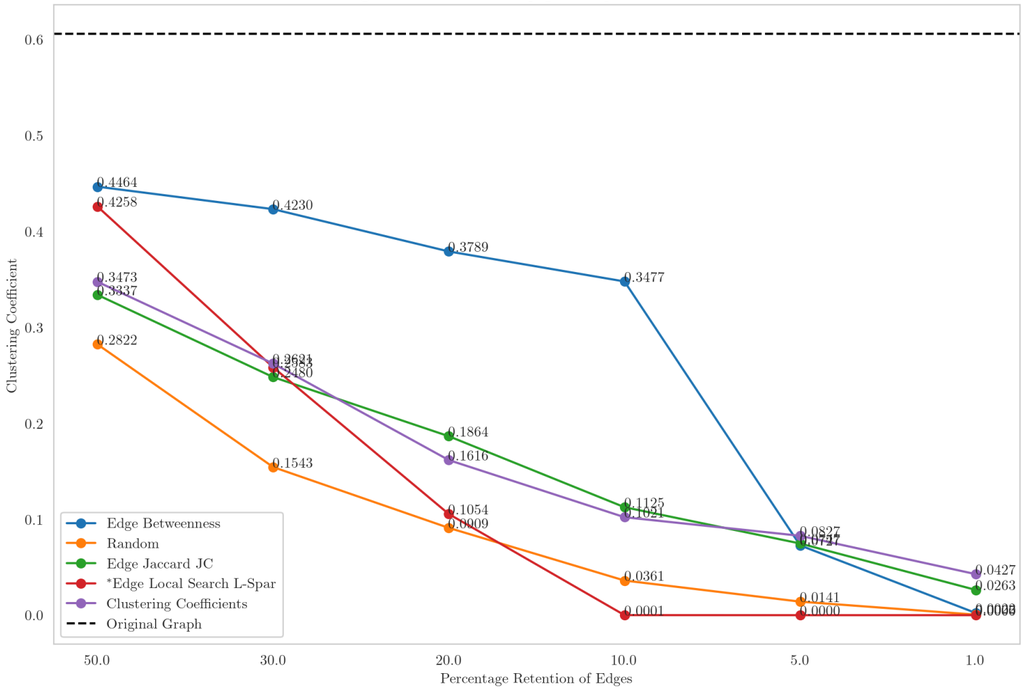


Facebook LPA

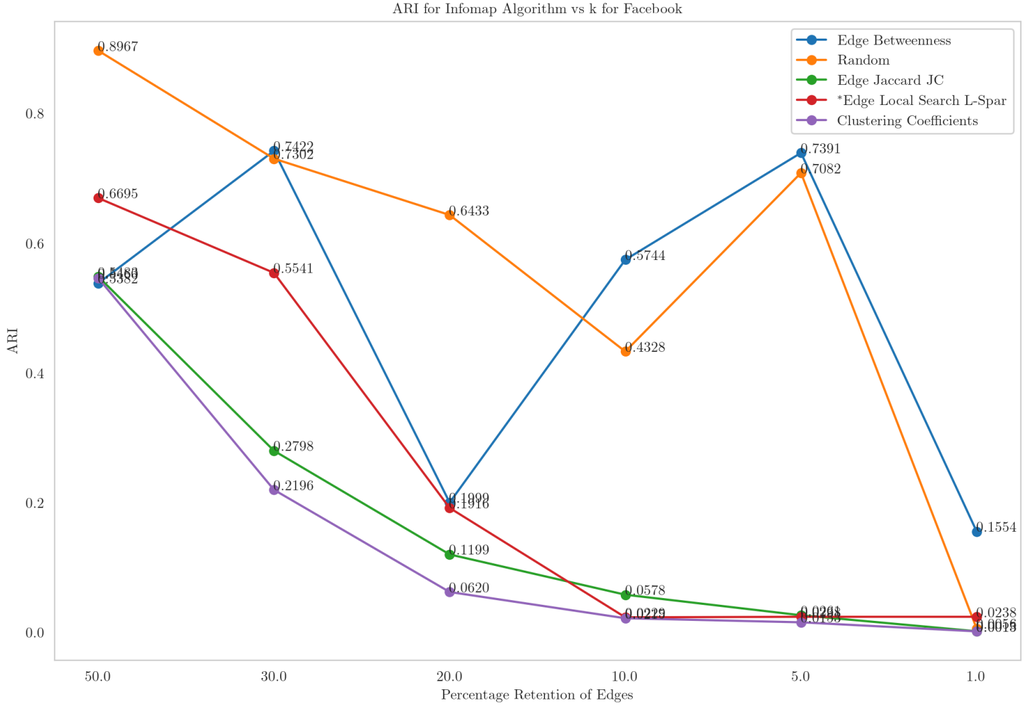


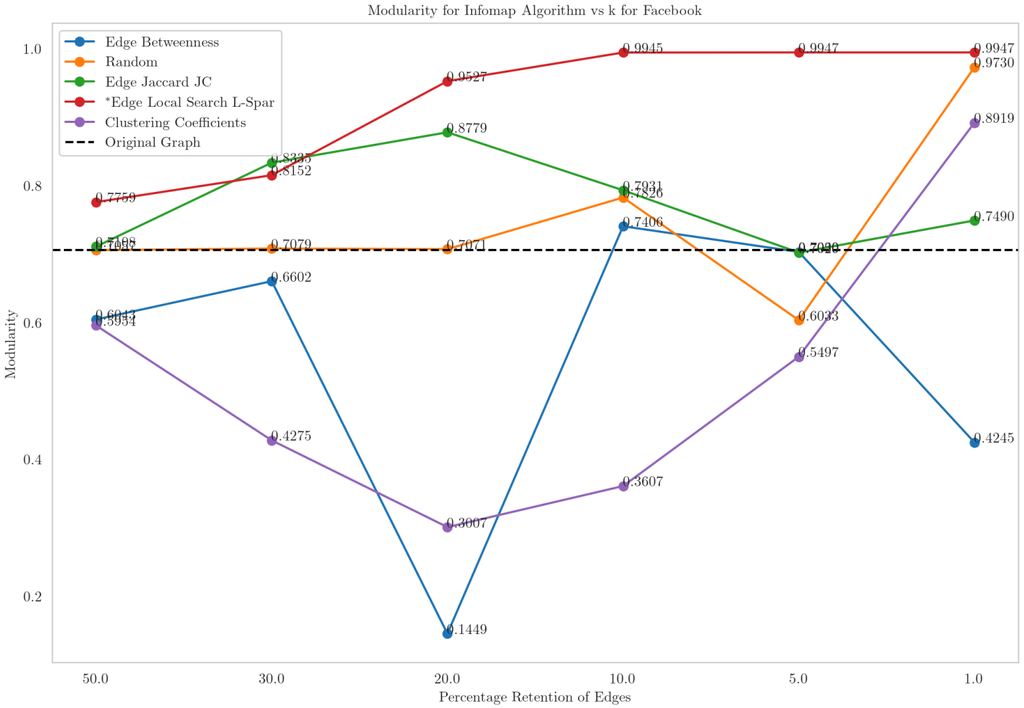


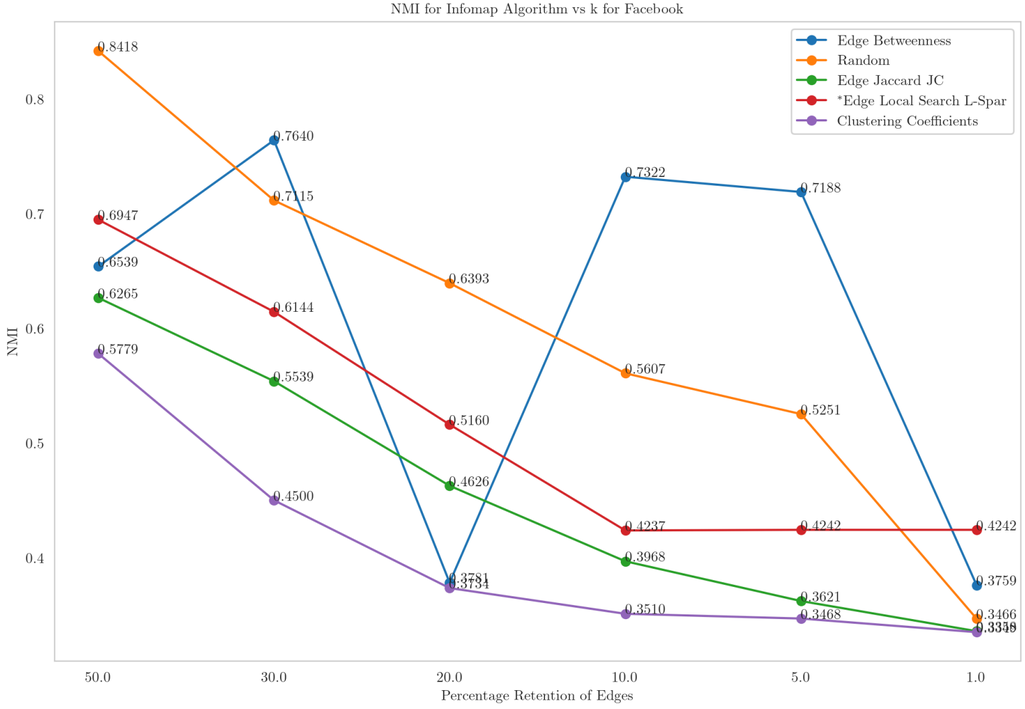


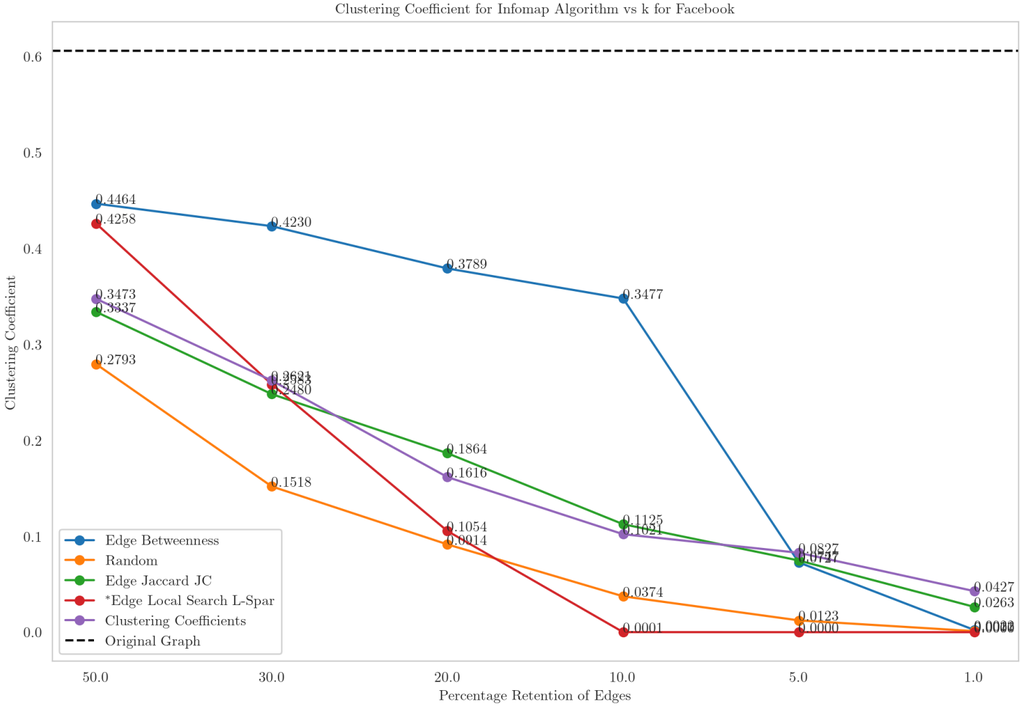


Facebook InfoMap

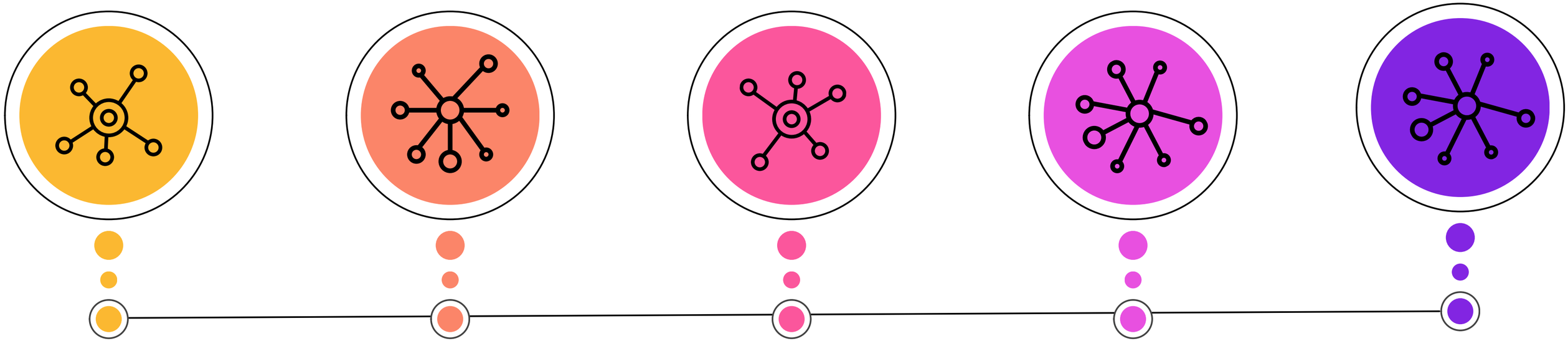








Some more sparsifying techniques



Degree Based Edge Sampling

Modified Metropolis Hastings

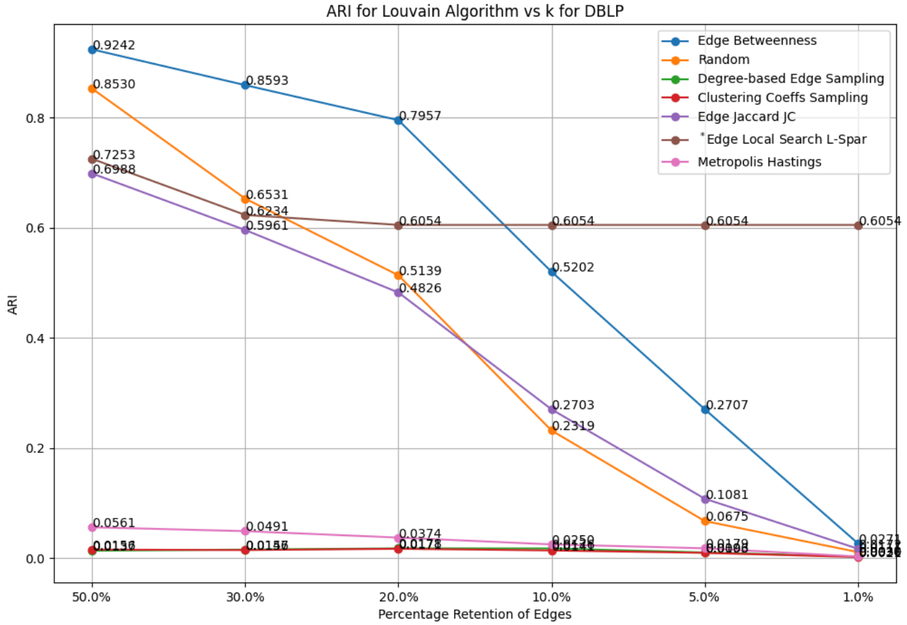
Forest Fire Sampling

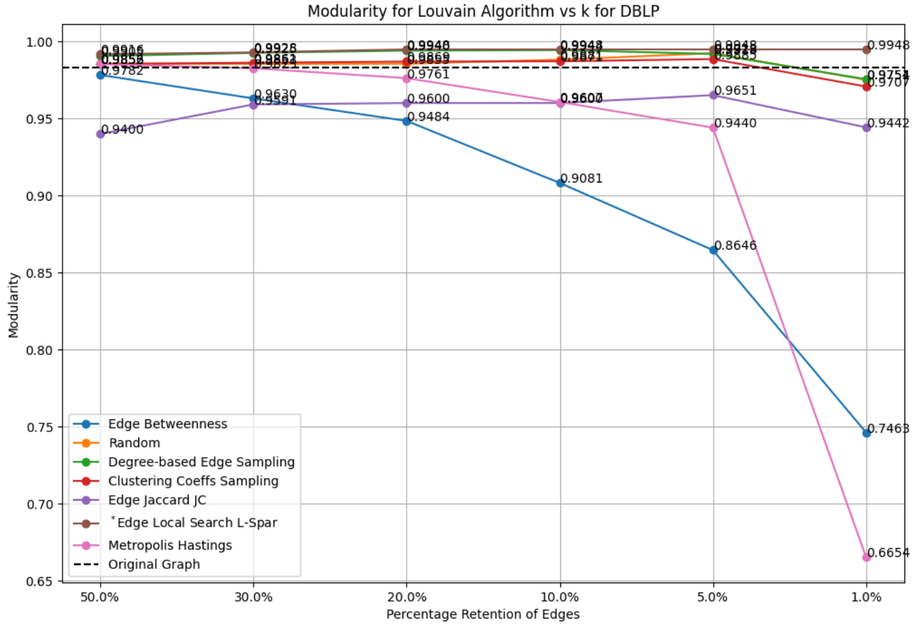
Triangle Count Based Sampling

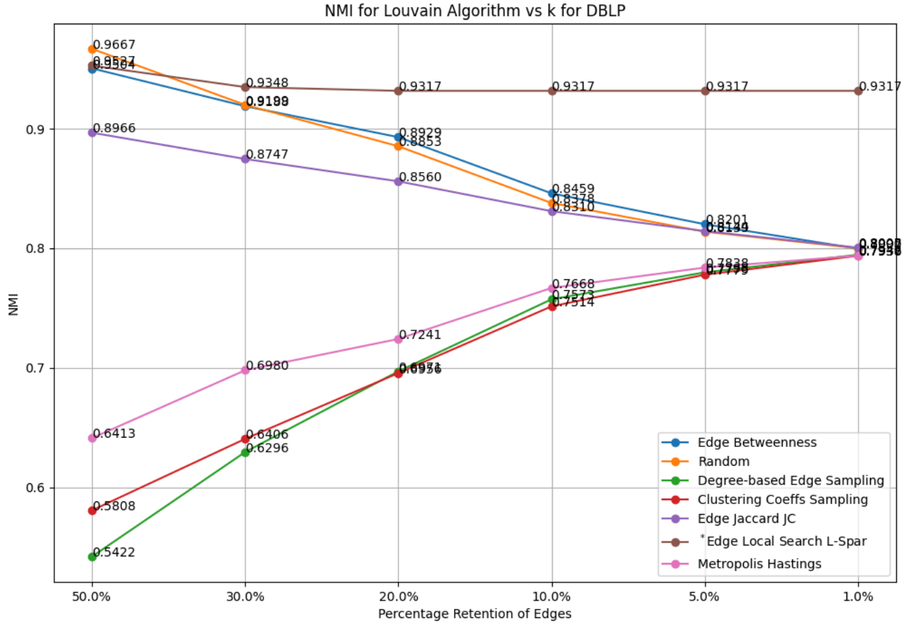
Effective Resistance

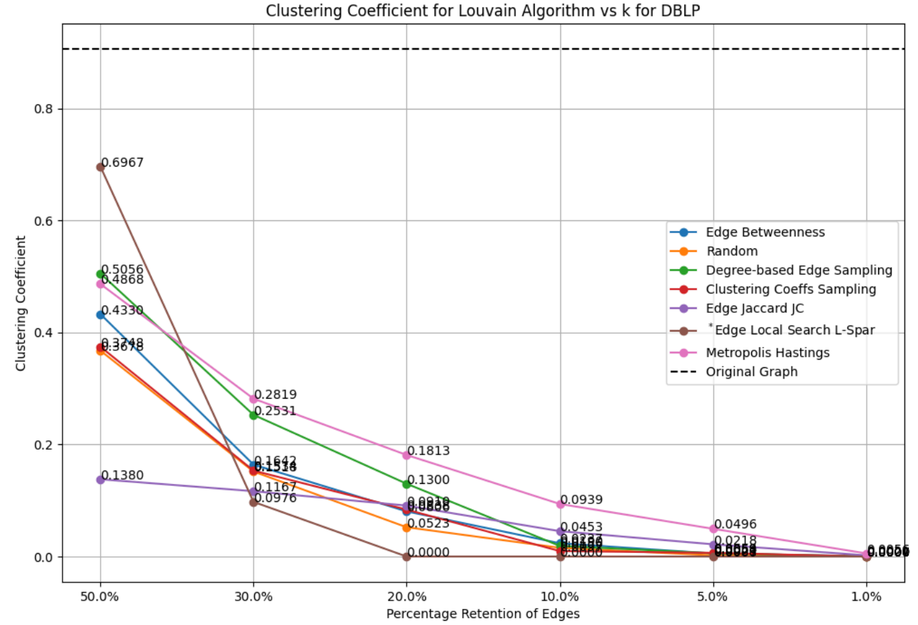
Glimpses...

DBLP
Louvain



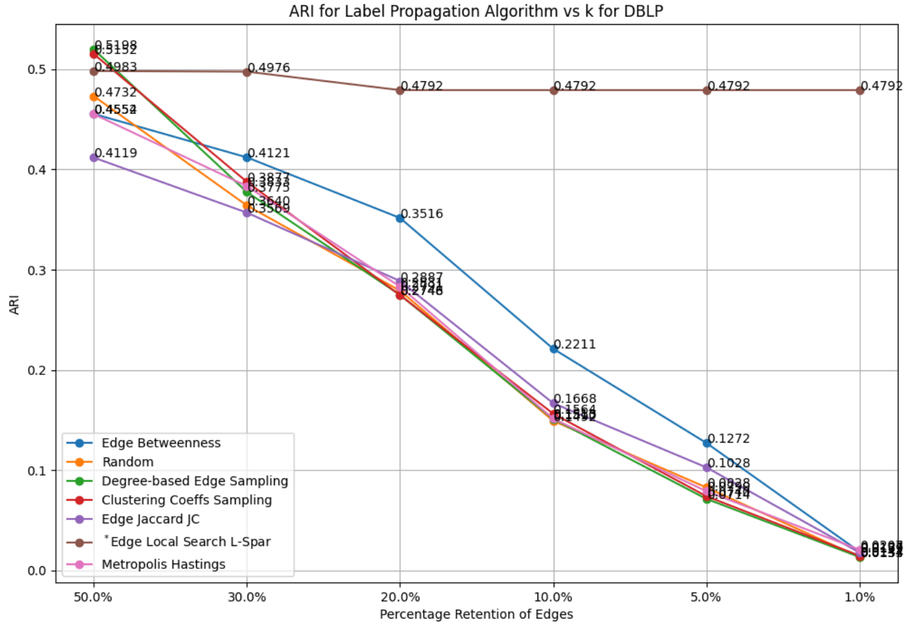


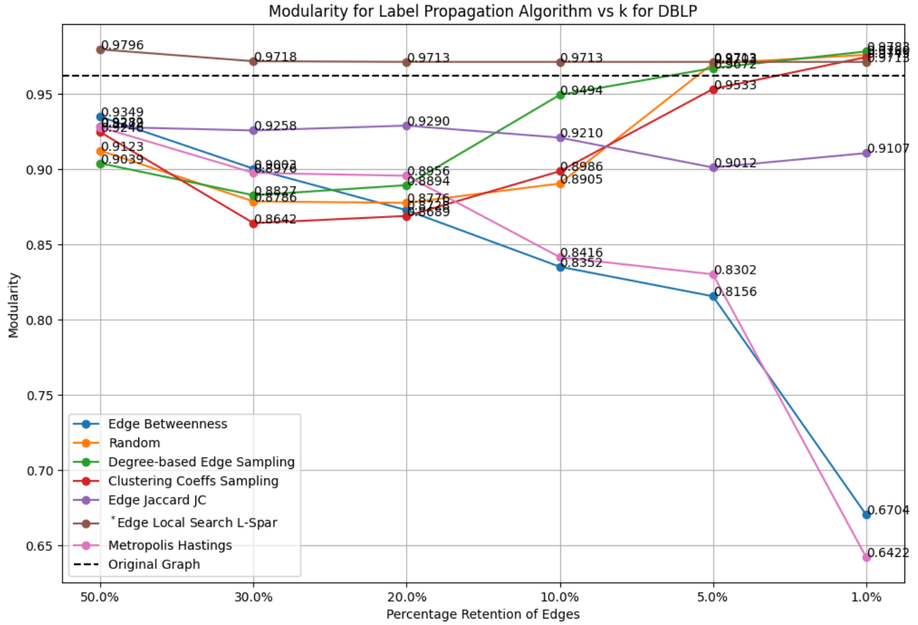


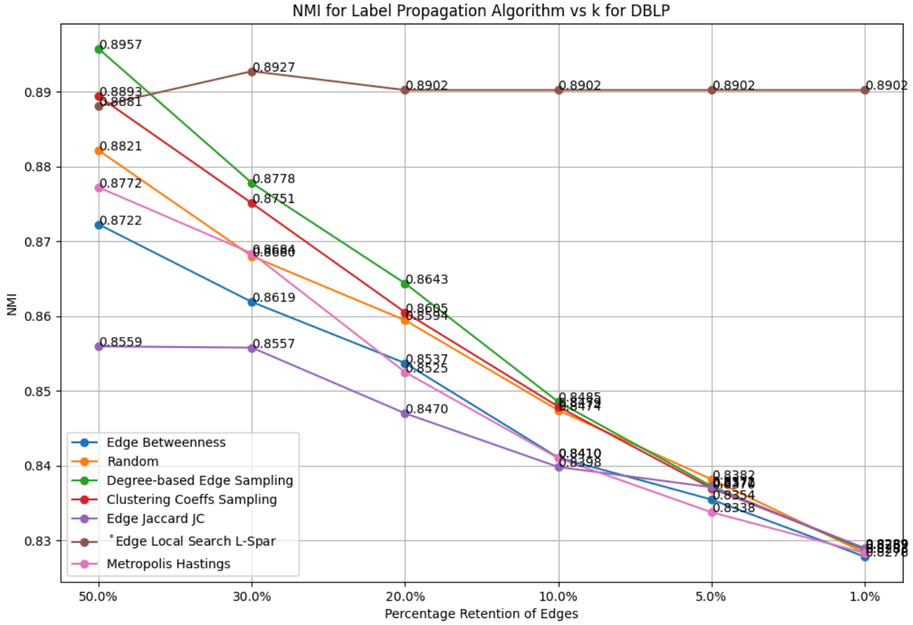


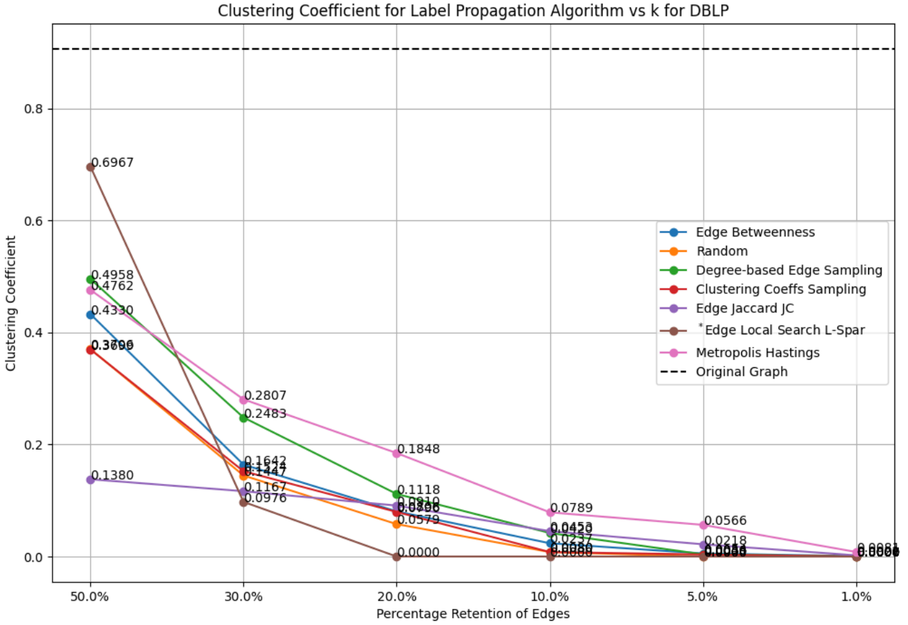
Glimpses...

DBLP
Label Propagation









References

[1.] Hu, P., & Lau, W. C. (2013). A survey and taxonomy of graph sampling. arXiv preprint arXiv:1308.5865.

[2.] Satuluri, V., Parthasarathy, S., & Ruan, Y. (2011, June). Local graph sparsification for scalable clustering. In Proceedings of the 2011 ACM SIGMOD International Conference on Management of data (pp. 721-732).

[3.] Wu, H. Y., & Chen, Y. L. (2020, November). Graph sparsification with generative adversarial network. In 2020 IEEE International Conference on Data Mining (ICDM) (pp. 1328-1333). IEEE.

[4.] Spielman, D. A., & Srivastava, N. (2008, May). Graph sparsification by effective resistances. In Proceedings of the fortieth annual ACM symposium on Theory of computing (pp. 563-568).

[5.] Wickman, R., Zhang, X., & Li, W. (2021). A Generic Graph Sparsification Framework using Deep Reinforcement Learning. arXiv preprint arXiv:2112.01565.

[6.] Hamann, M., Lindner, G., Meyerhenke, H., Staudt, C. L., & Wagner, D. (2016). Structure-preserving sparsification methods for social networks. Social Network Analysis and Mining, 6, 1-22.

[7.] Yang, J., & Leskovec, J. (2012, August). Defining and evaluating network communities based on ground-truth. In Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics (pp. 1-8).

[8.] Lindner, G., Staudt, C. L., Hamann, M., Meyerhenke, H., & Wagner, D. (2015, August). Structure-preserving sparsification of social networks. In Proceedings of the 2015 IEEE/ACM International conference on advances in social networks analysis and mining 2015 (pp. 448-454).

Graph Libraries Used

[1.] NetworKit

[2.] Networkx

[3.] CDlib

[4.] Pyvis

Internet Mathematics Vol. 6, No. 1: 29–123

Community Structure in Large Networks: Natural Cluster Sizes and the Absence of Large Well-Defined Clusters

Jure Leskovec, Kevin J. Lang, Anirban Dasgupta, and Michael W. Mahoney

Abstract. A large body of work has been devoted to defining and identifying clusters or communities in social and information networks, i.e., in graphs in which the nodes represent underlying social entities and the edges represent some sort of interaction between pairs of nodes. Most such research begins with the premise that a community or a cluster should be thought of as a set of nodes that has more and/or better connections between its members than to the remainder of the network. In this paper, we explore from a novel perspective several questions related to identifying meaningful communities in large social and information networks, and we come to several striking conclusions.

Rather than defining a procedure to extract sets of nodes from a graph and then attempting to interpret these sets as "real" communities, we employ approximation algorithms for the graph-partitioning problem to characterize as a function of size the statistical and structural properties of partitions of graphs that could plausibly be interpreted as communities. In particular, we define the *network community profile plot*, which characterizes the "best" possible community—according to the conductance measure—over a wide range of size scales. We study over one hundred large real-world networks, ranging from traditional and online social networks, to technological and information networks and web graphs, and ranging in size from thousands up to tens of millions of nodes.

Our results suggest a significantly more refined picture of community structure in large networks than has been appreciated previously. Our observations agree with previous work on small networks, but we show that large networks have a very different structure. In particular, we observe tight communities that are barely connected to the rest of the network at very small size scales (up to ≈ 100 nodes); and communities of size scale beyond ≈ 100 nodes gradually "blend into" the expander-like core of the network and thus become less "community-like," with a roughly inverse relationship between community size and optimal community quality. This observation agrees well

1542-7951/08 $0.50 per page