SCAN: A Structural Clustering Algorithm for Networks

Xiaowei Xu
Nurcan Yuruk
Zhidan Feng
Thomas Schweiger

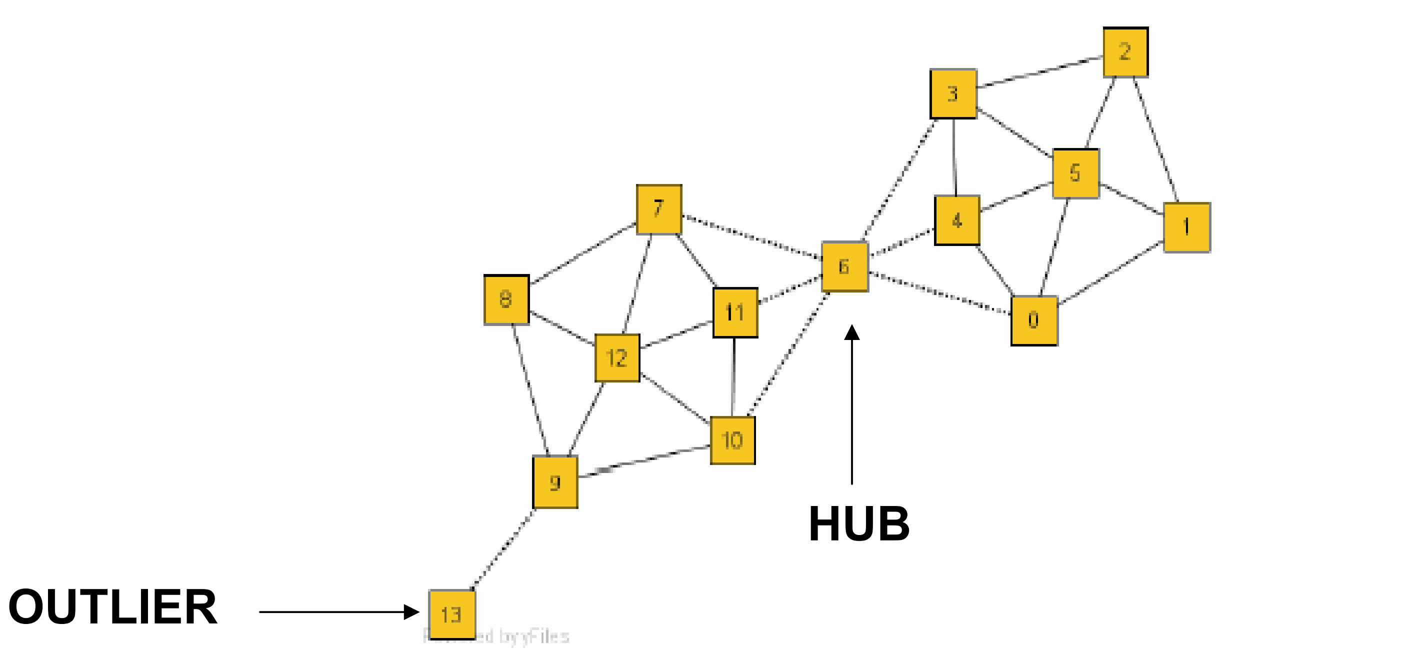
Presented By

Maen Hammad

Features

- Identify Clusters
- Fast algorithm O(m)
- Identify hubs and outliers nodes

Example



Related Work

1. Min-max cut method
   – Partition the graph into 2 clusters
   – Minimize edges between the clusters
   – Maximize edges within a cluster
   – Optimization → Find the best cut (difficult ?)
   – Some constraints to achieve optimality

Related work

2. Modularity clustering

$$Q = \sum_{s=1}^{k} \left[\frac{l_s}{L} - \left(\frac{d_s}{2L} \right)^2 \right]$$

L: number of edges
L_s number of edges in cluster S
d_s: sum of degrees in cluster s

- Optimality→ Maximize Q →NP Complete
- Greedy method is used
- Running time is O(md log n)

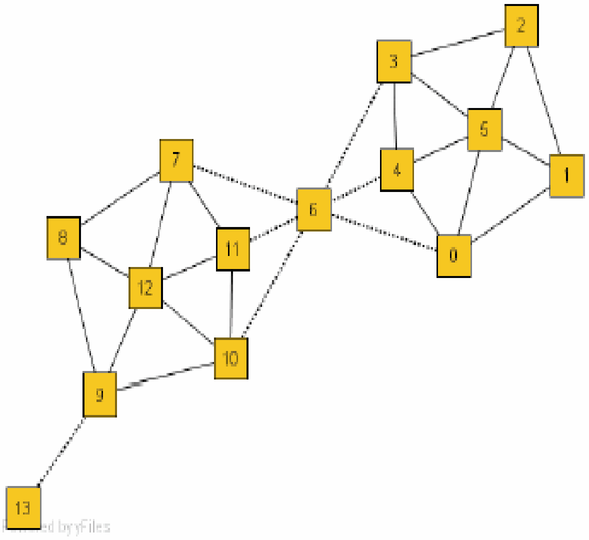
Structure-Connected Clusters

- Considering the neighborhood around two connected vertices
- Two vertices are assigned to a cluster according to how they share neighbors
- Some neighbors are outliers or hubs

Definitions

- VERTEX STRUCTURE

$$\Gamma(v) = \{w \in V \mid (v,w) \in E\} \cup \{v\}$$



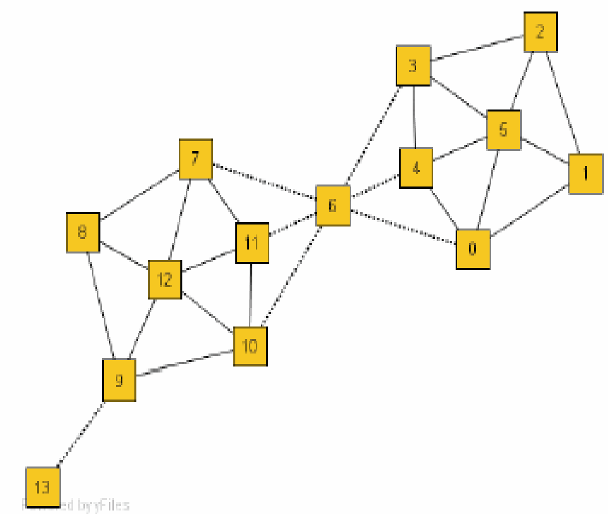
- STRUCTURAL SIMILARITY

$$\sigma(v,w) = \frac{|\Gamma(v) \cap \Gamma(w)|}{\sqrt{|\Gamma(v)||\Gamma(w)|}}$$

Definitions

- ε-NEIGHBORHOOD

$$N_\varepsilon(v) = \{w \in \Gamma(v) \mid \sigma(v,w) \geq \varepsilon\}$$



- CORE

$$CORE_{\varepsilon,\mu}(v) \Leftrightarrow |N_\varepsilon(v)| \geq \mu$$

- DIRECT STRUCTURE REACHABILITY

$$DirREACH_{\varepsilon,\mu}(v,w) \Leftrightarrow CORE_{\varepsilon,\mu}(v) \wedge w \in N_\varepsilon(v)$$

Definitions

- **HUB**

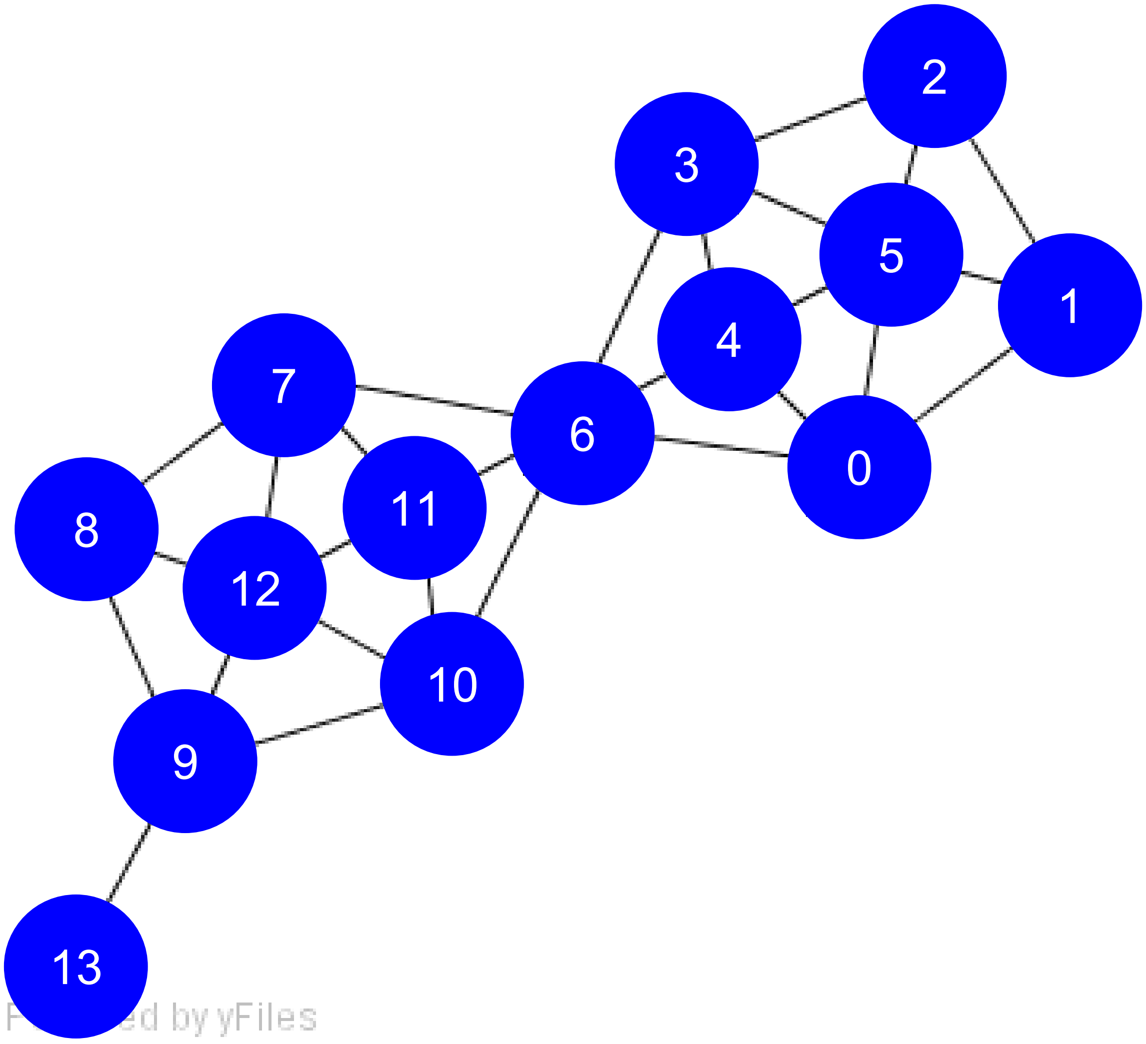
1. If *v* is not a member of any cluster
2. If *v* bridges different clusters

- **OUTLIER**

1. If v is not a member of any cluster
2. If v does not bridge different clusters

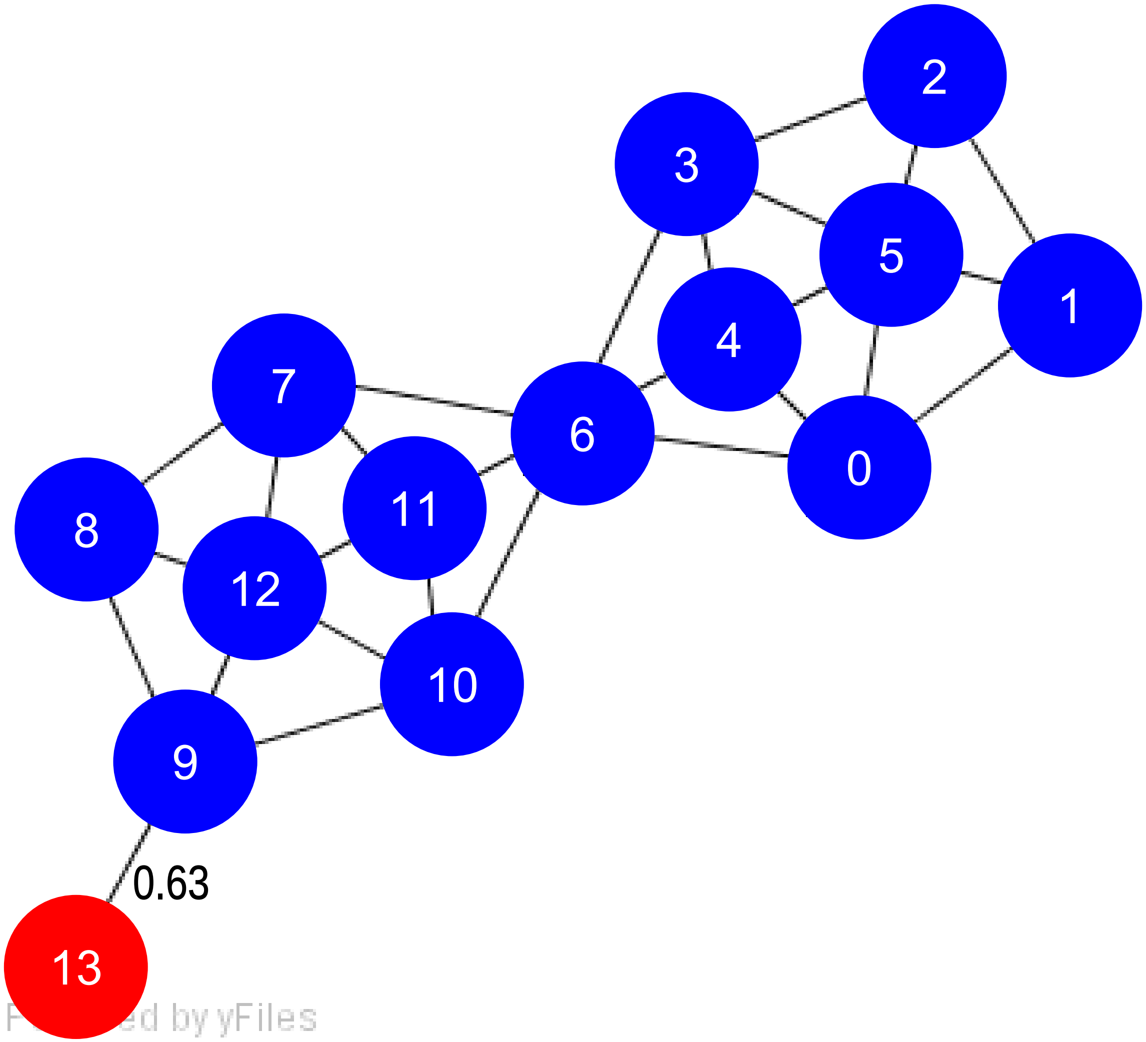
Algorithm

$\mu = 2$
$\varepsilon = 0.7$



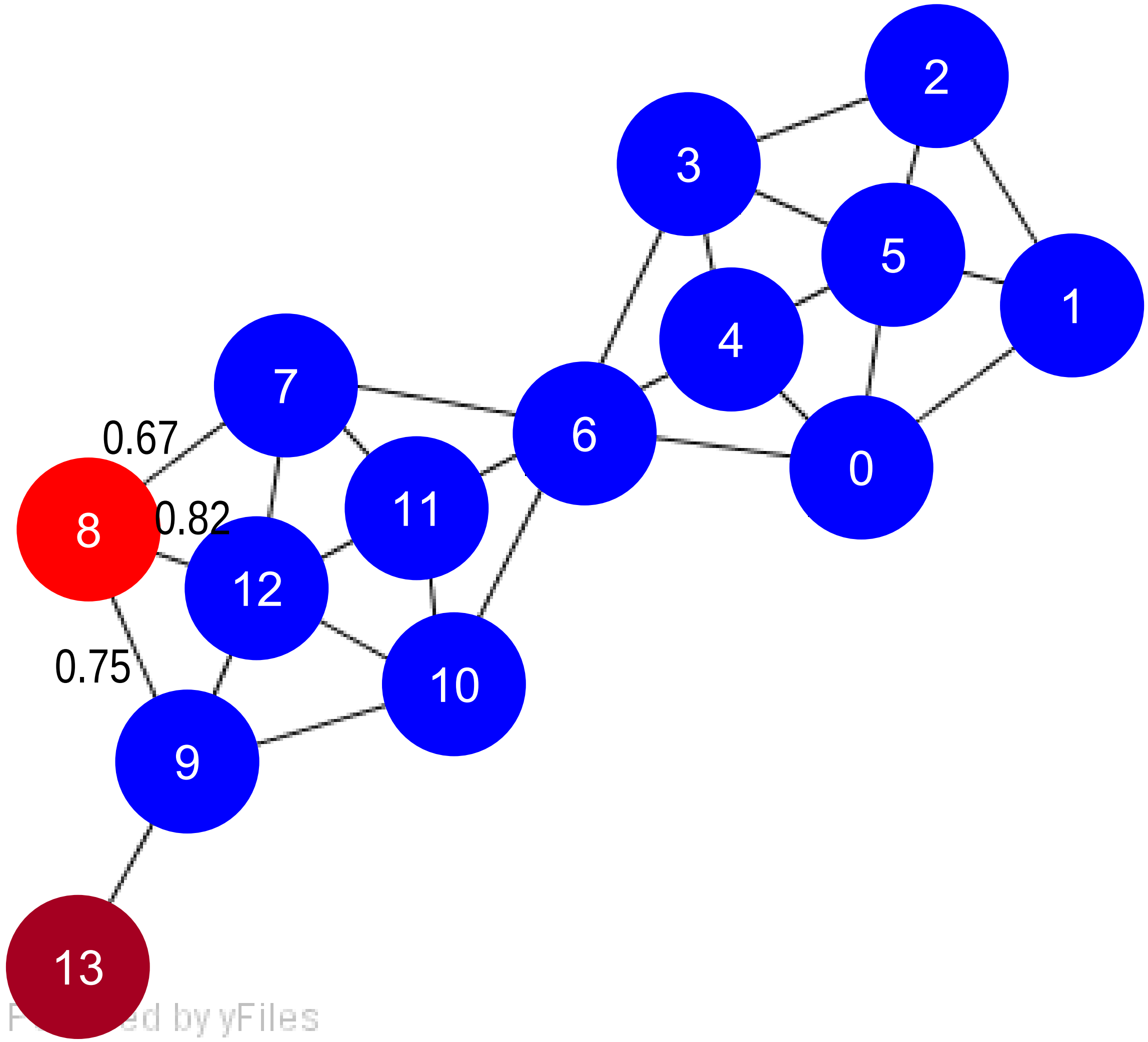
Algorithm

$\mu = 2$
$\varepsilon = 0.7$



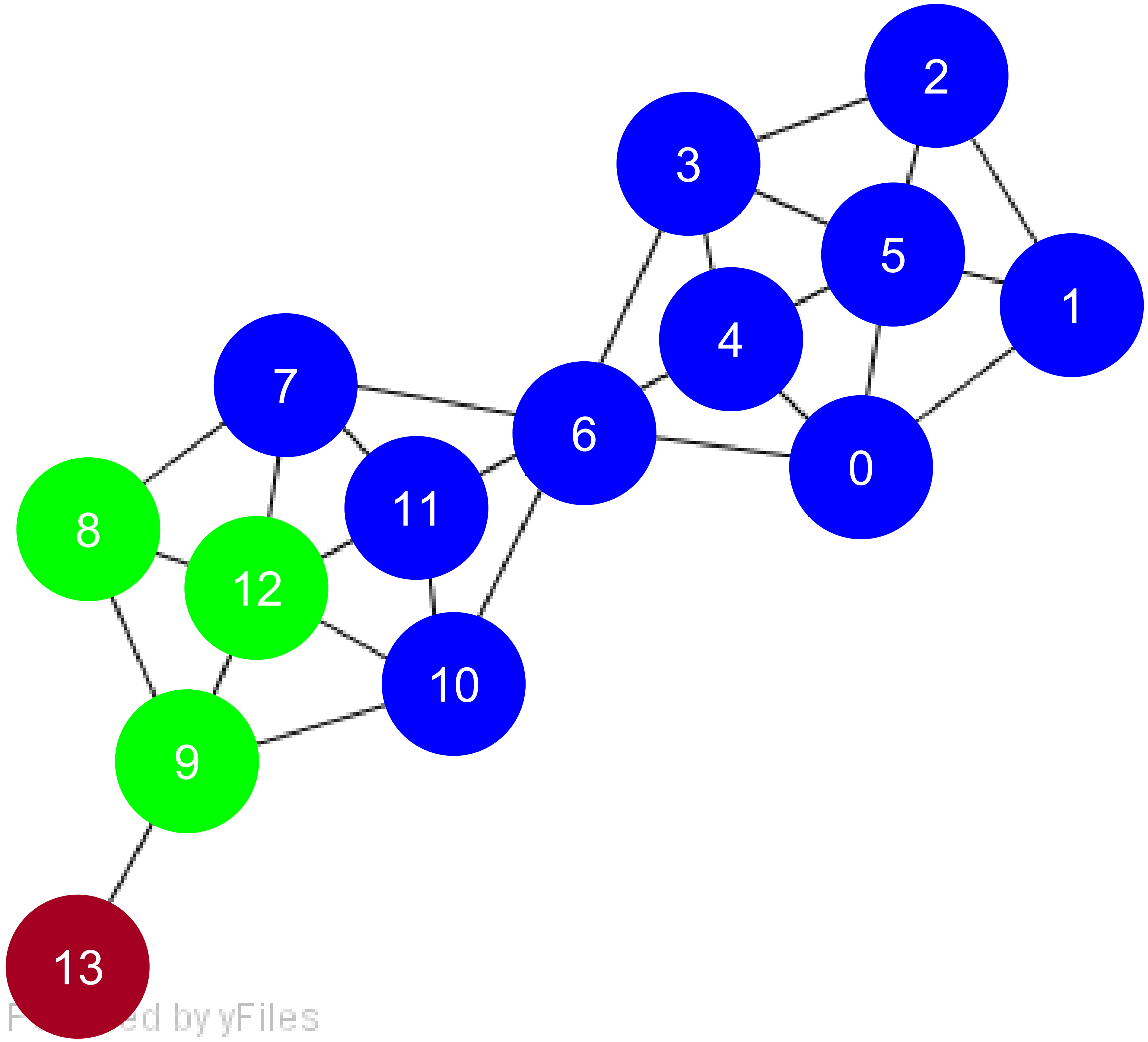
Algorithm

$\mu = 2$
$\varepsilon = 0.7$



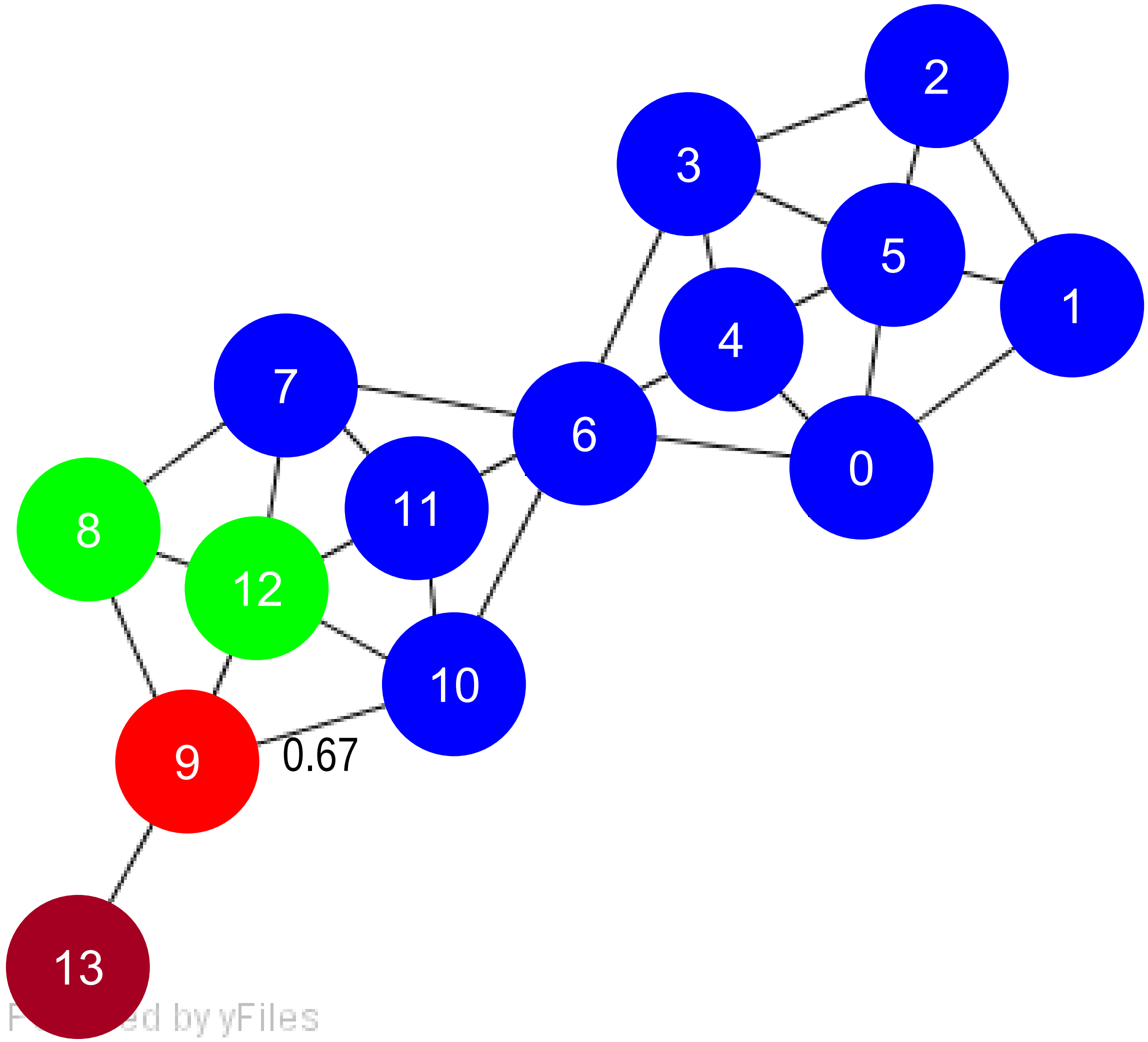
Algorithm

$\mu = 2$
$\varepsilon = 0.7$



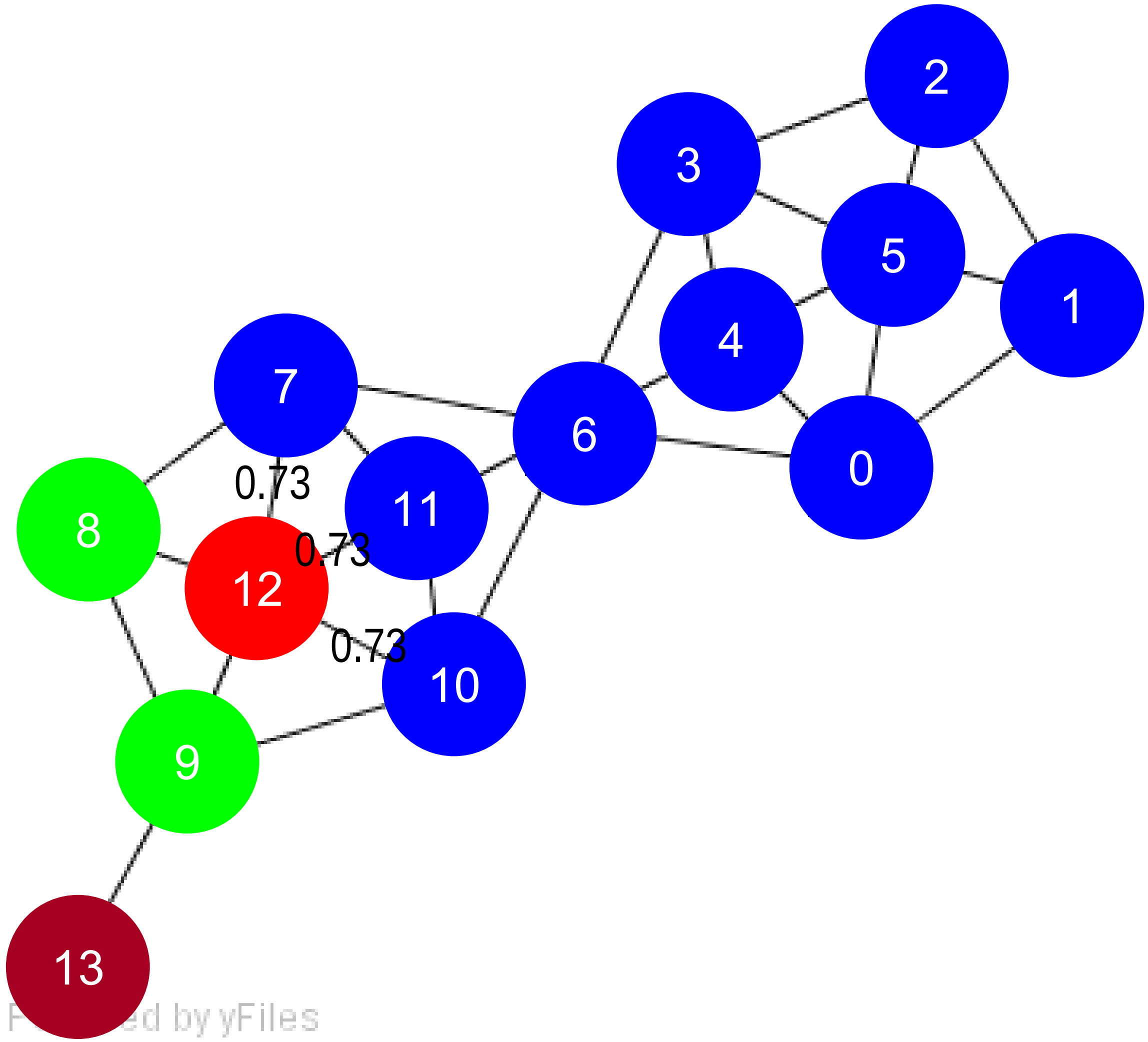
Algorithm

$\mu = 2$
$\varepsilon = 0.7$



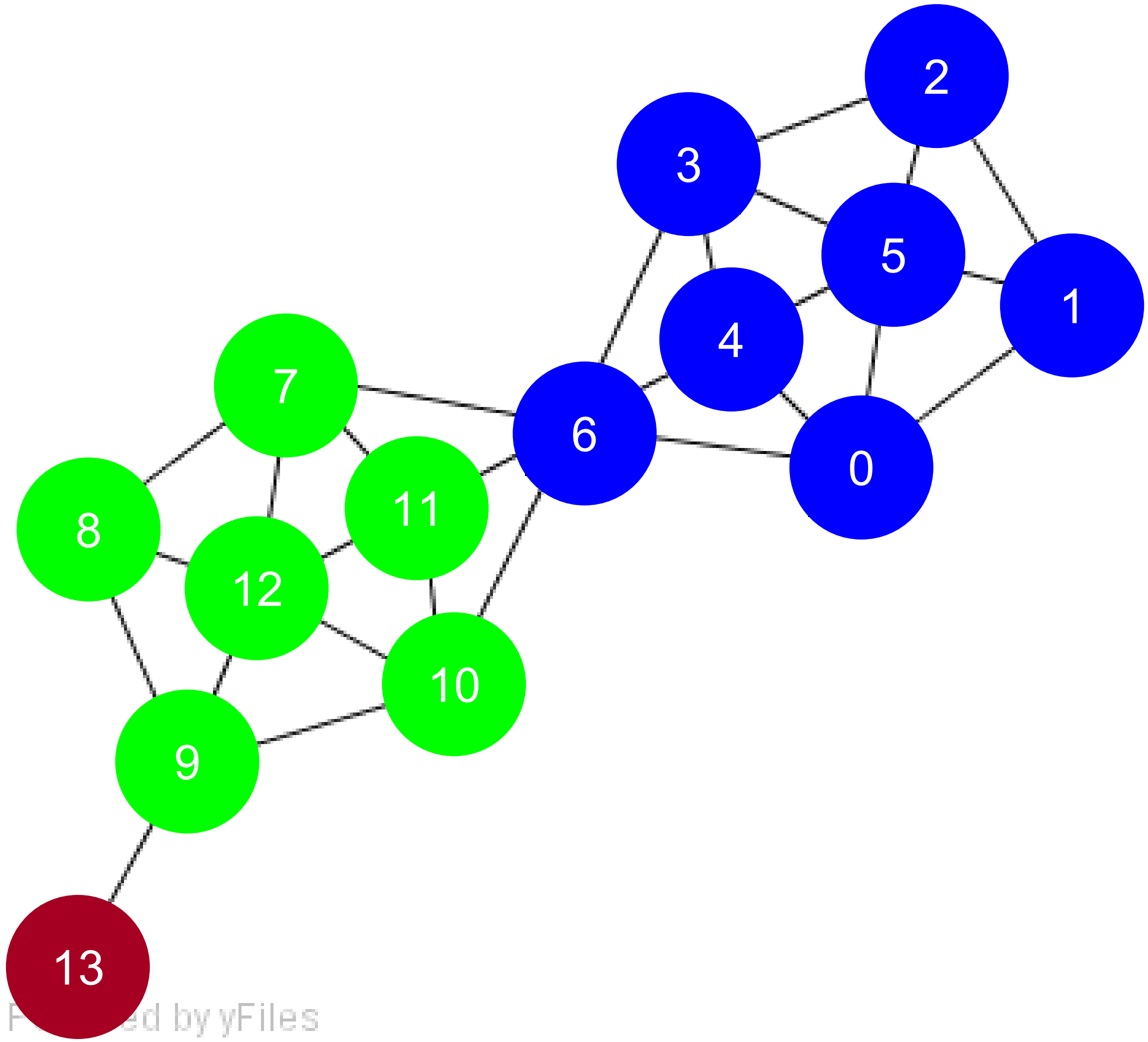
Algorithm

$\mu = 2$
$\varepsilon = 0.7$



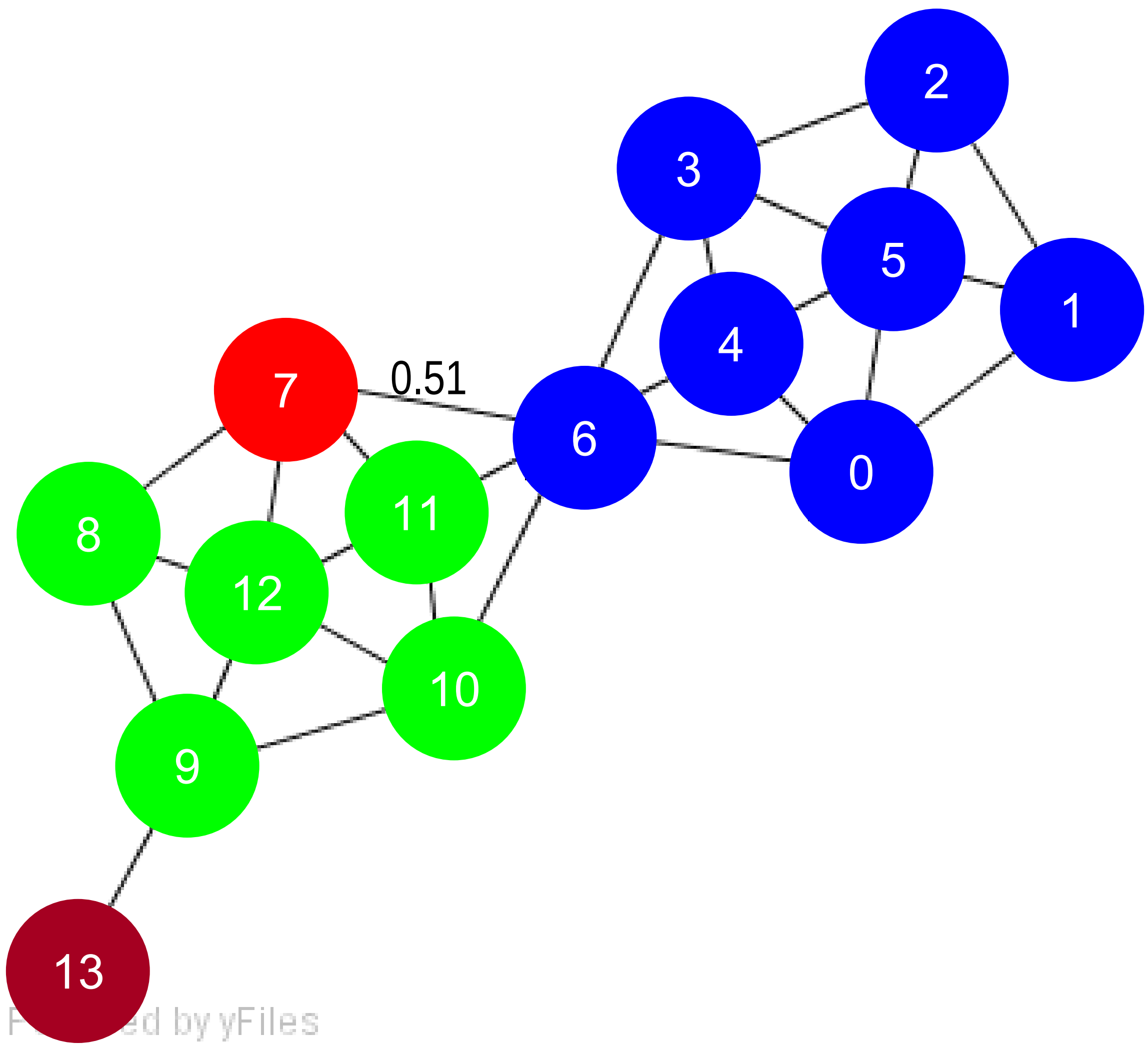
Algorithm

$\mu = 2$
$\varepsilon = 0.7$



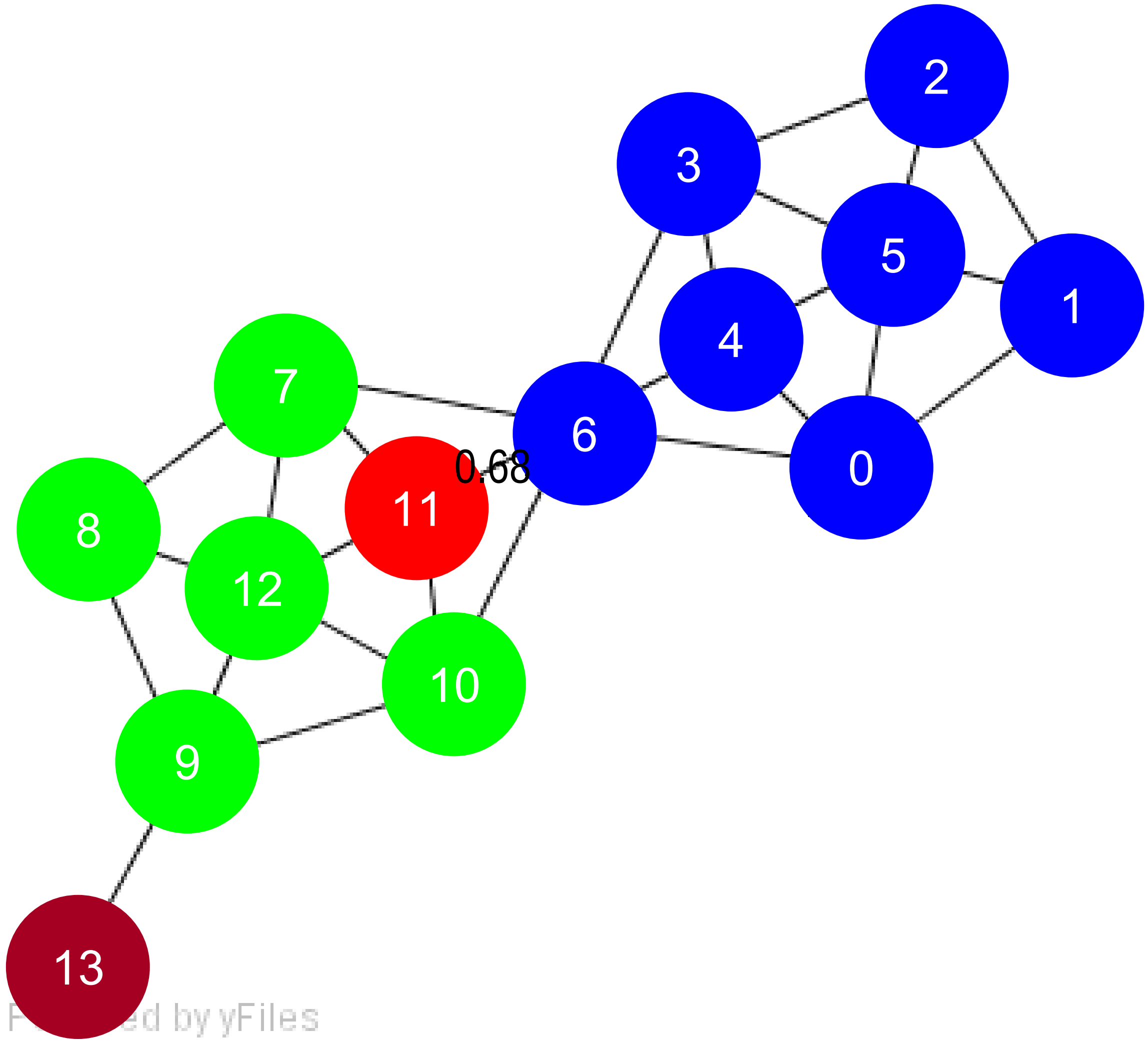
Algorithm

$\mu = 2$
$\varepsilon = 0.7$



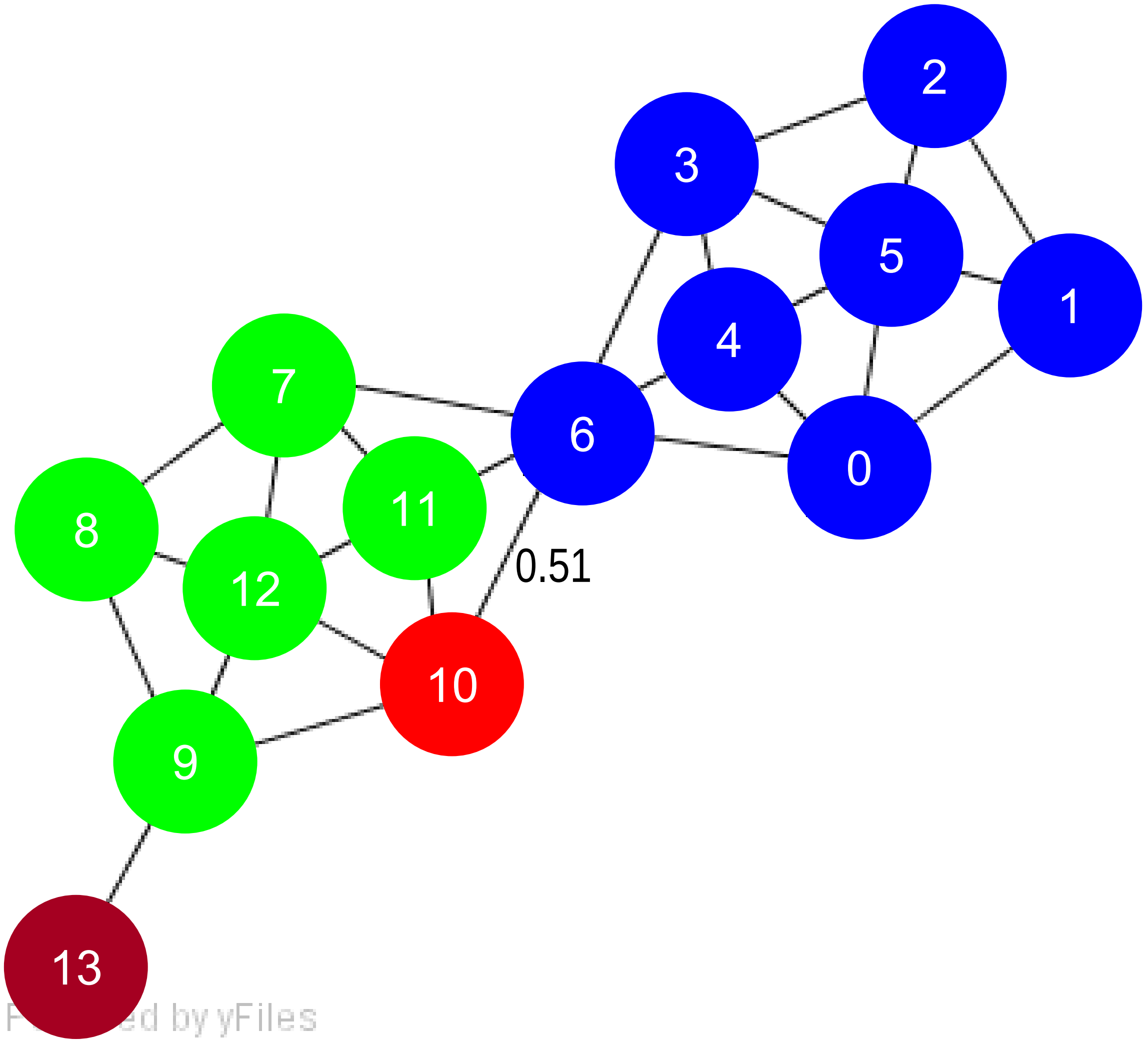
Algorithm

$\mu = 2$
$\varepsilon = 0.7$



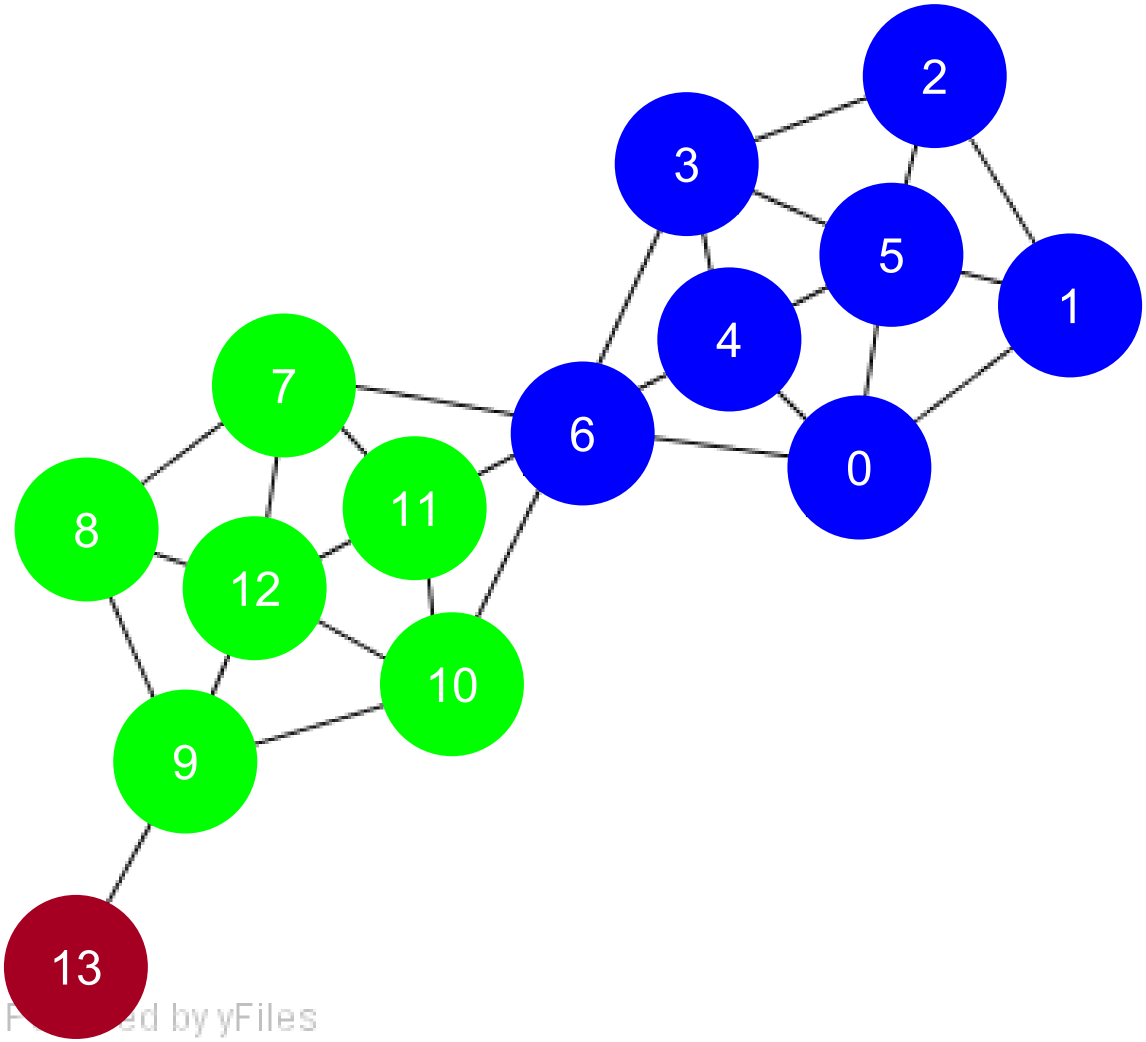
Algorithm

$\mu = 2$
$\varepsilon = 0.7$



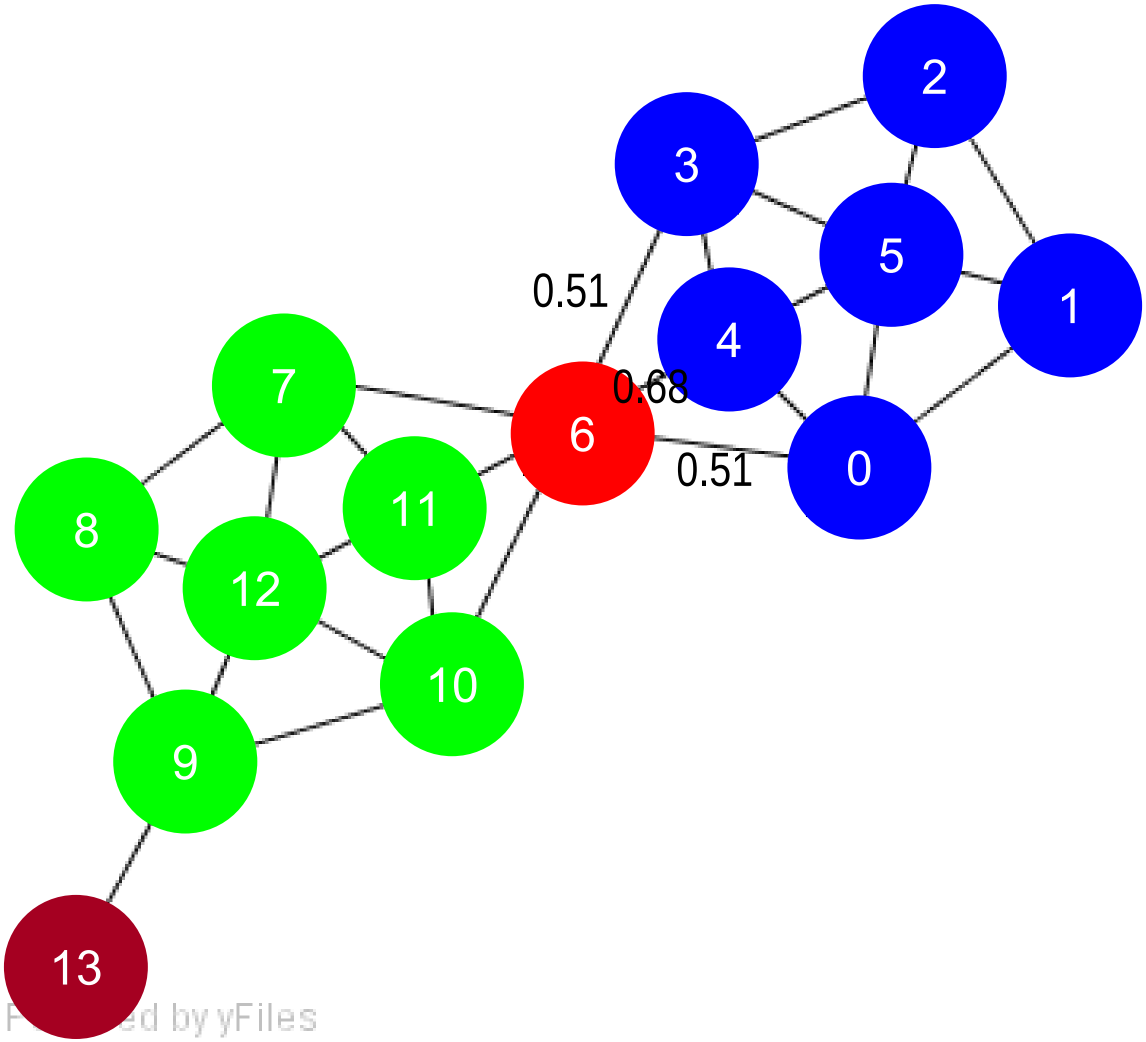
Algorithm

$\mu = 2$
$\varepsilon = 0.7$



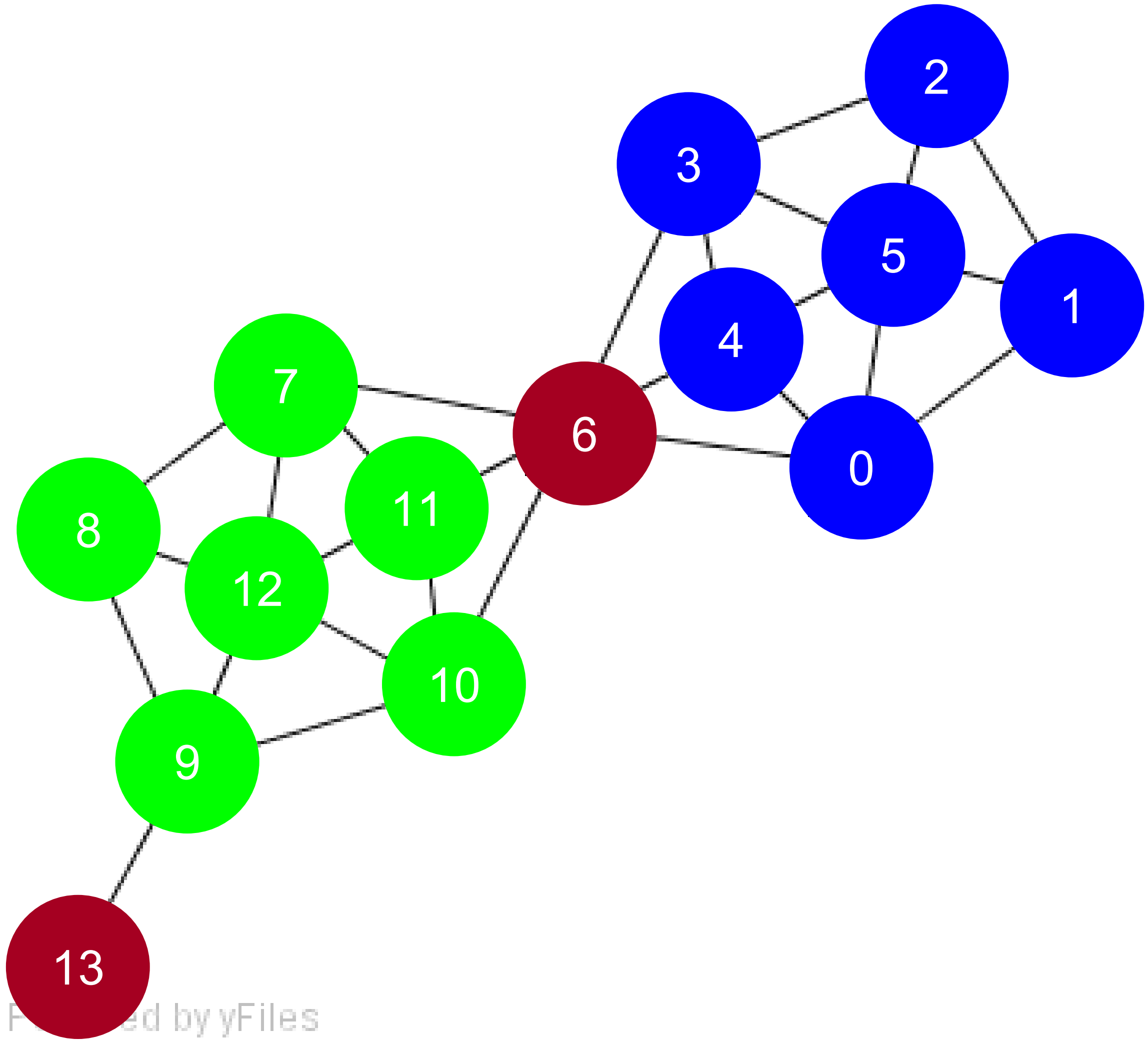
Algorithm

$\mu = 2$
$\varepsilon = 0.7$



Algorithm

$\mu = 2$
$\varepsilon = 0.7$



```
ALGORITHM SCAN(G=<V, E>, ε, μ)
// all vertices in V are labeled as unclassified;
for each unclassified vertex v ∈ V do
// STEP 1. check whether v is a core;
    if CORE_ε,μ(v) then
// STEP 2.1. if v is a core, a new cluster is expanded;
        generate new clusterID;
        insert all x ∈ N_ε (v) into queue Q;
        while Q ≠ 0 do
            y = first vertex in Q;
            R = {x ∈ V | DirREACH_ε,μ(y, x)};
            for each x ∈ R do
                if x is unclassified or non-member then
                    assign current clusterID to x;
                if x is unclassified then
                    insert x into queue Q;
            remove y from Q;
    else
// STEP 2.2. if v is not a core, it is labeled as non-member
        label v as non-member;
end for.
// STEP 3. further classifies non-members
for each non-member vertex v do
    if ( ∃ x, y ∈ Γ(v) ( x.clusterID ≠ y.clusterID) then
      label v as hub
    else
      label v as outlier;
end for.
end SCAN.
```

Complexity Analysis

- Given a graph with m edges and n vertices
- Check the core:
  - Compute structural similarity
  - Examine all the vertex's neighbors
  - $O(\deg(V1)+\deg(V2)+\ldots\deg(Vn)) = O(2m)$
  - $O(m)$

Efficiency

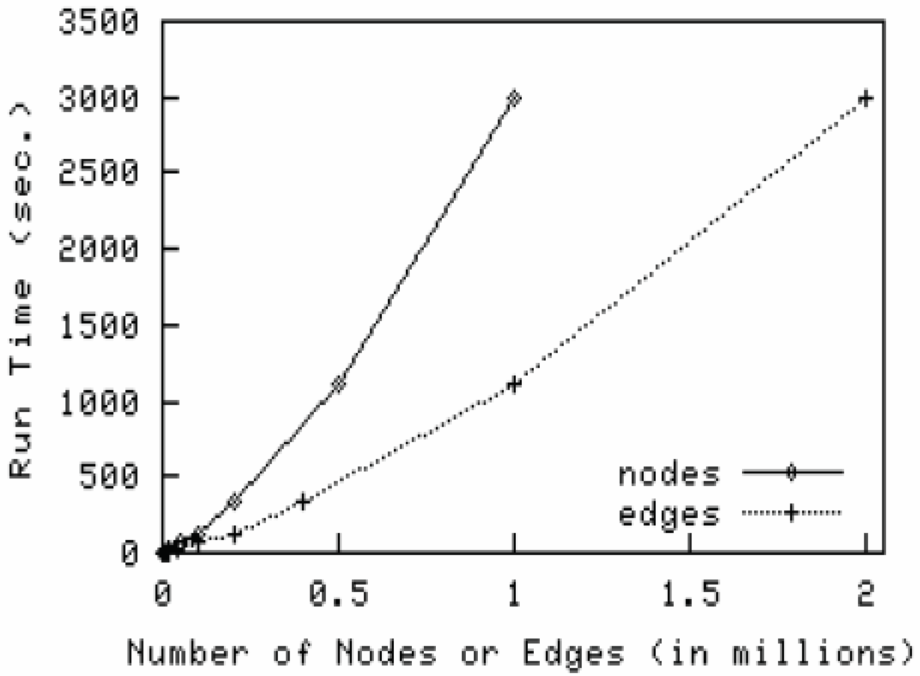


Figure 4. Running Time for FastModularity

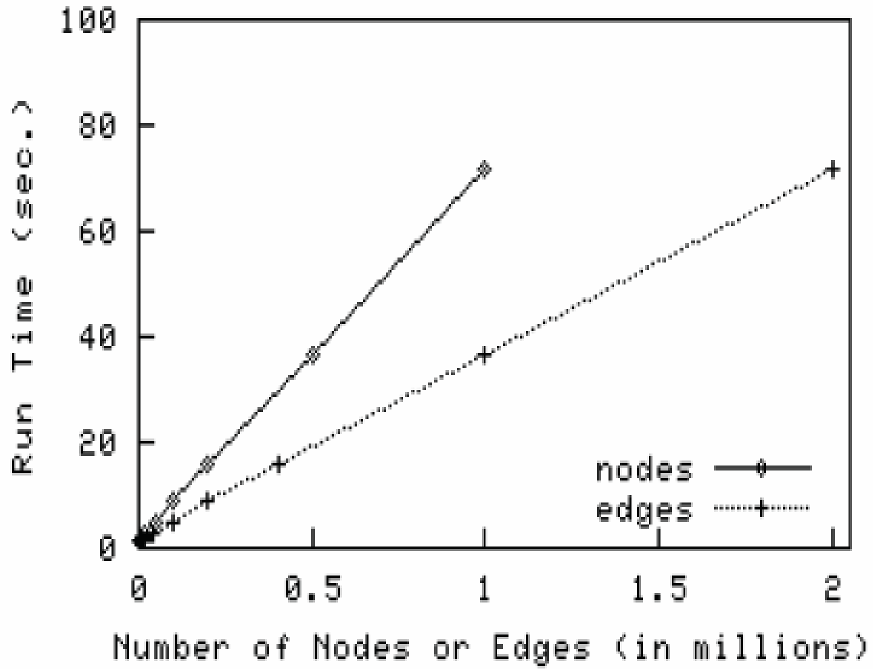


Figure 5. Running Time for SCAN

Effectiveness

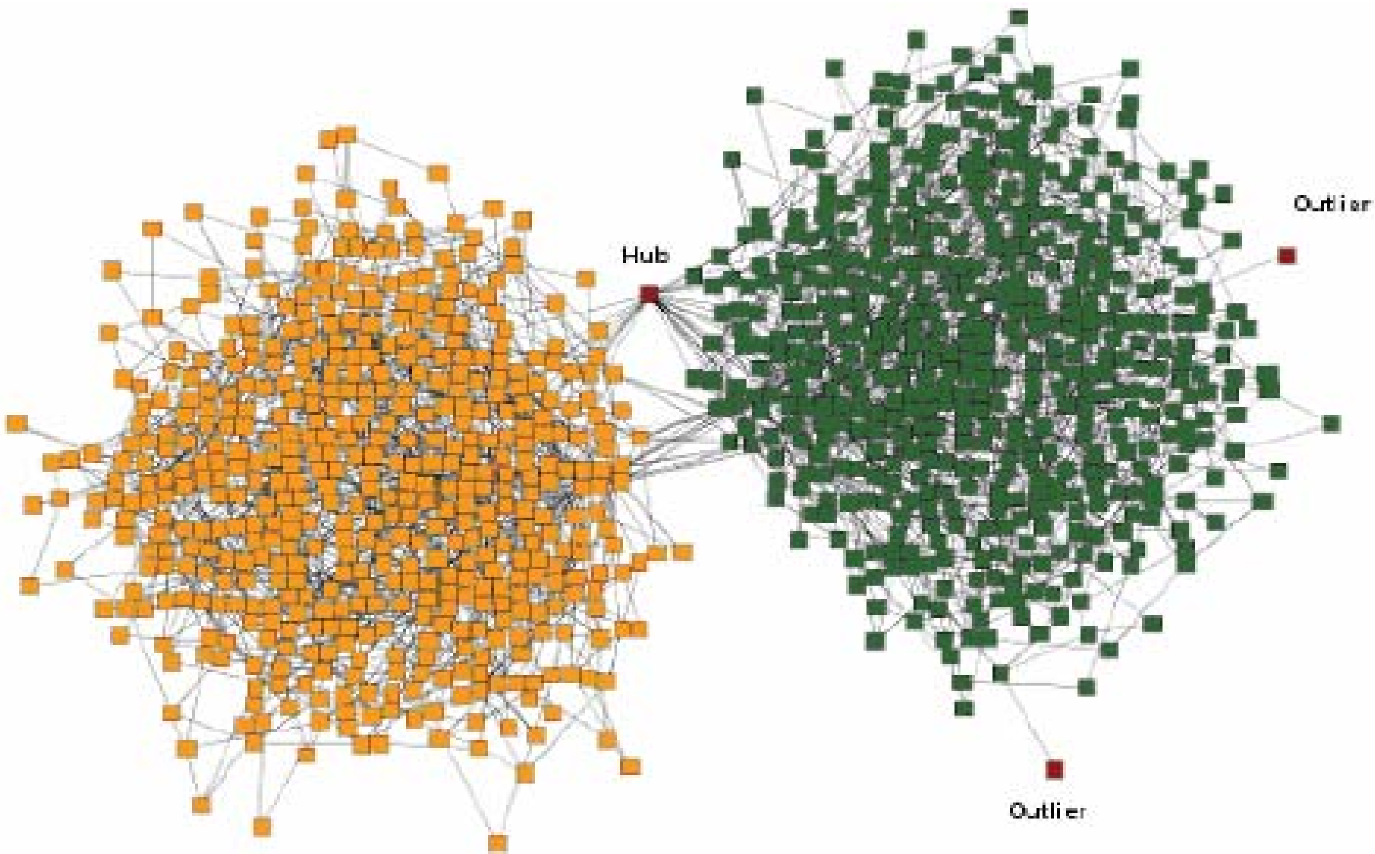


Figure 3. A Synthetic Graph with 1,000 Vertices

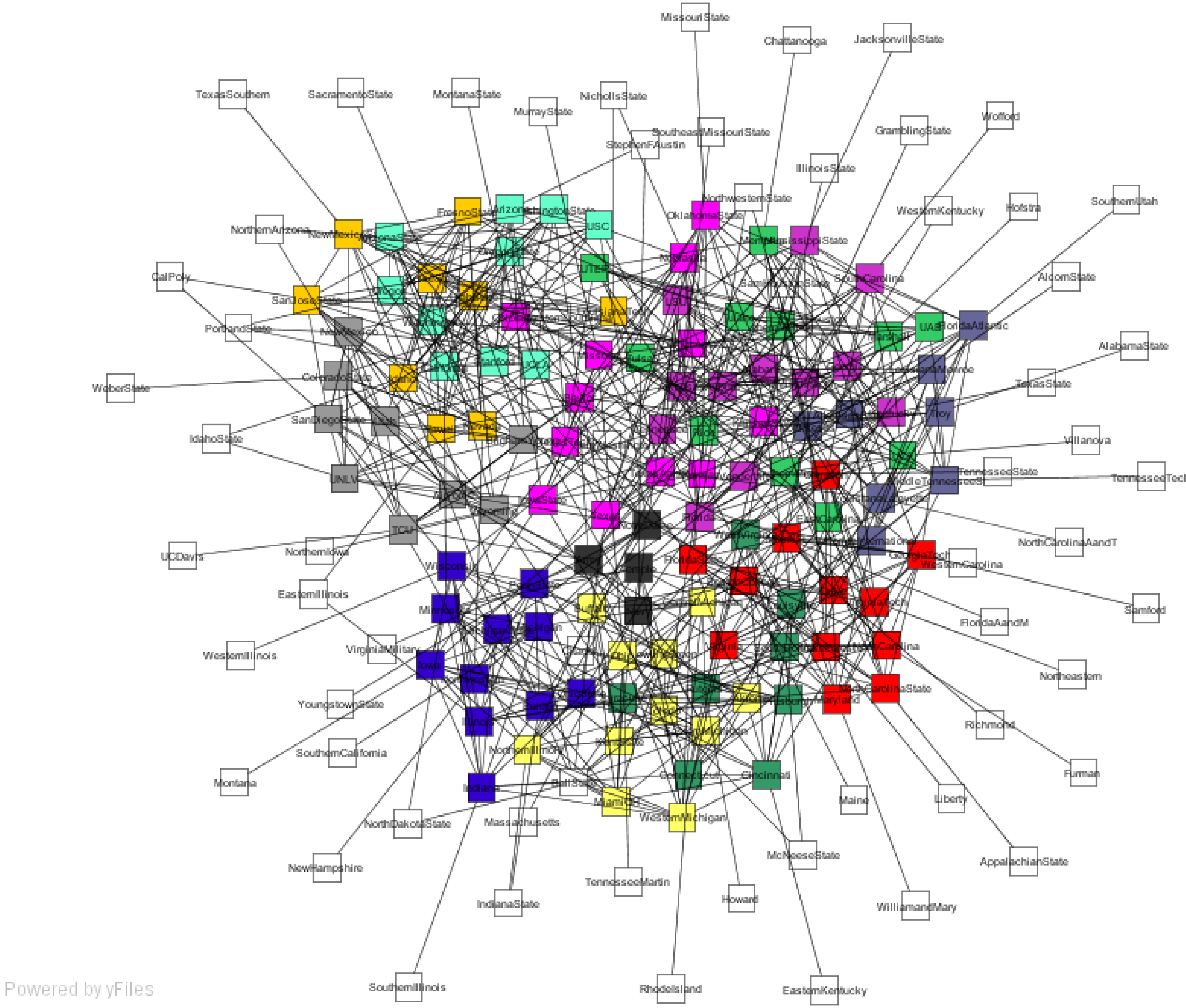


Figure 6. NCAA Football Bowl Subdivision schedule as a network, showing the 12 conferences in color, independent schools in black, and lower division schools in white.

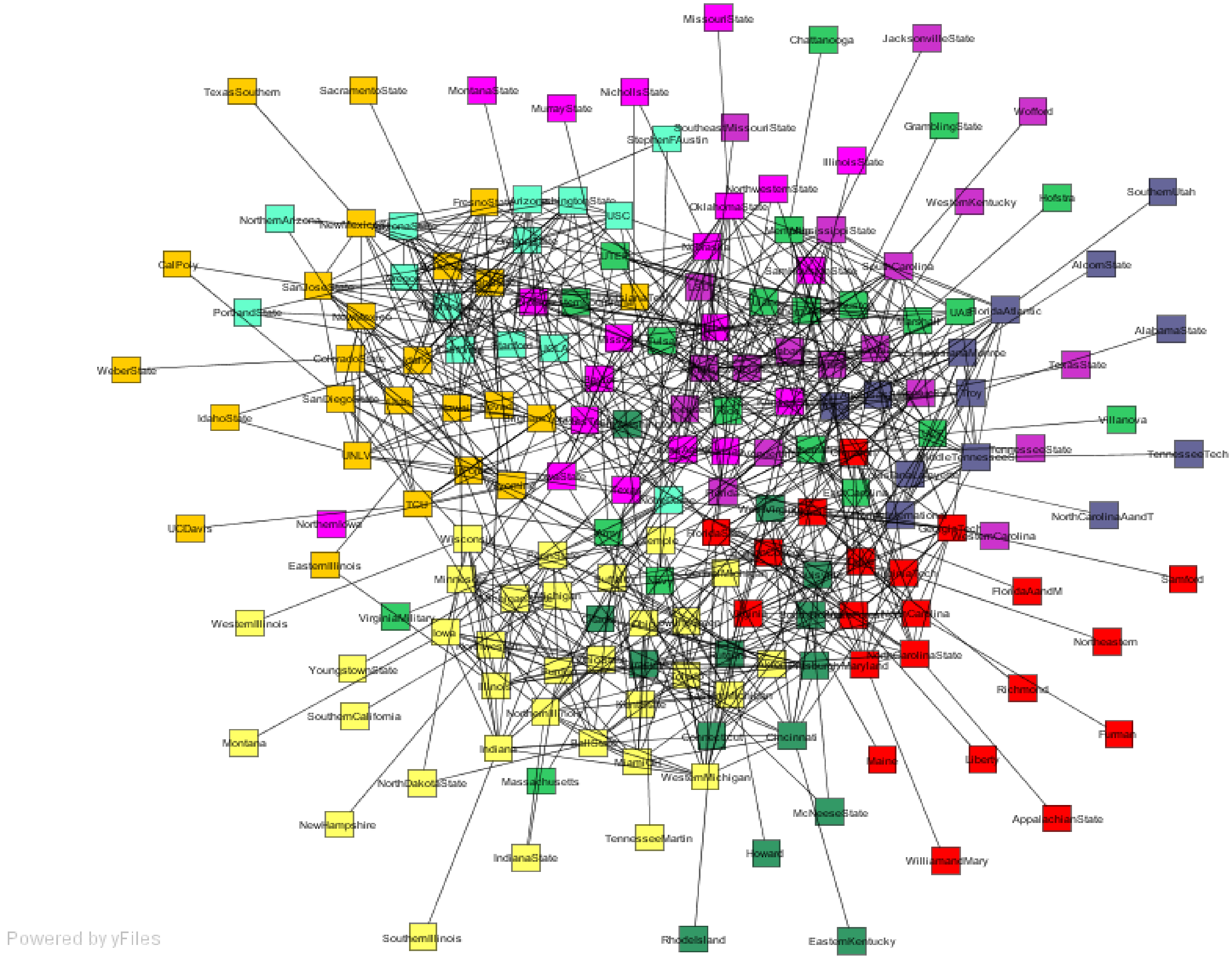


Figure 7. NCAA Football Bowl Subdivision schedule as clustered by FastModularity Algorithm.

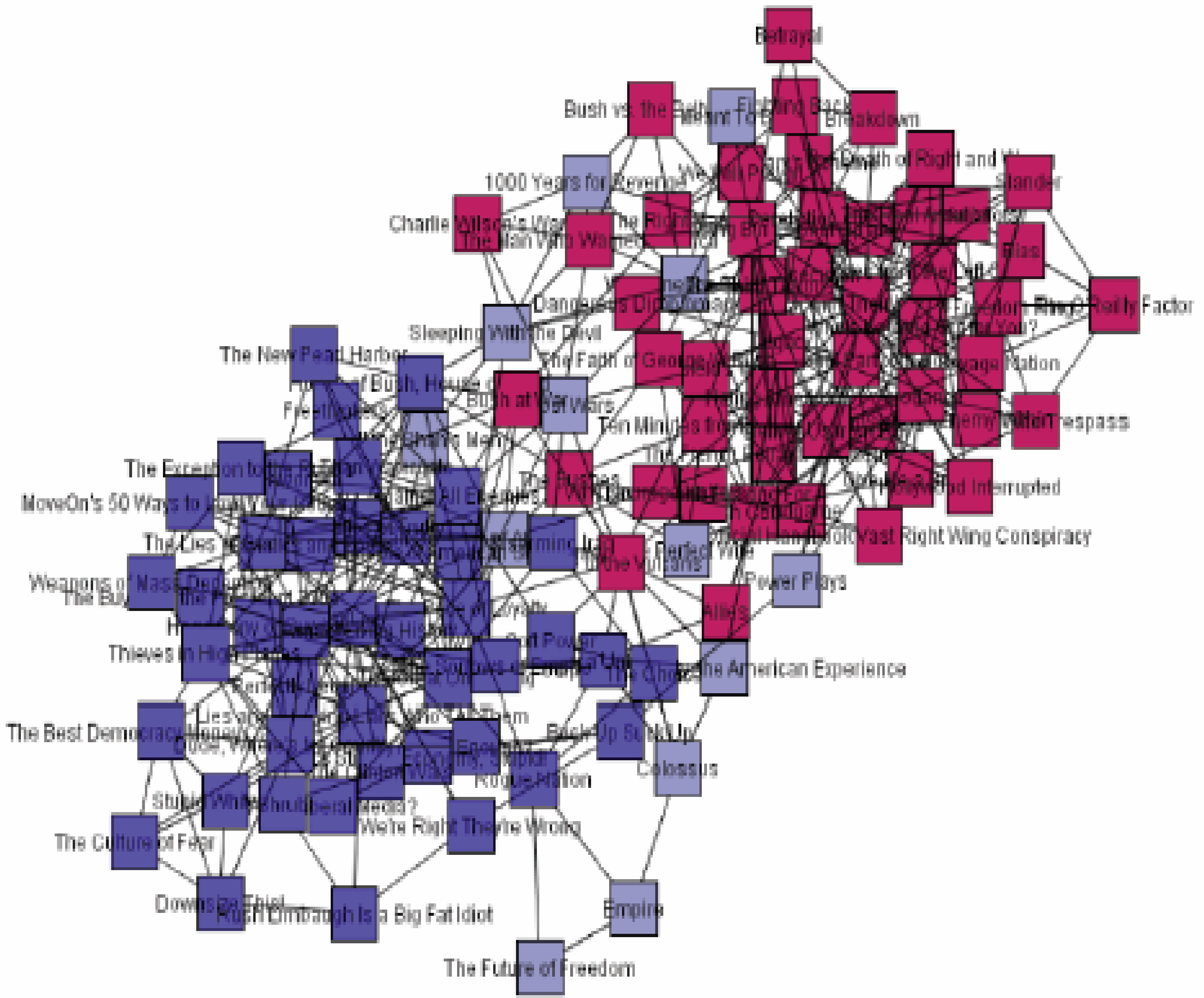


Figure 8. Political Book Graph.

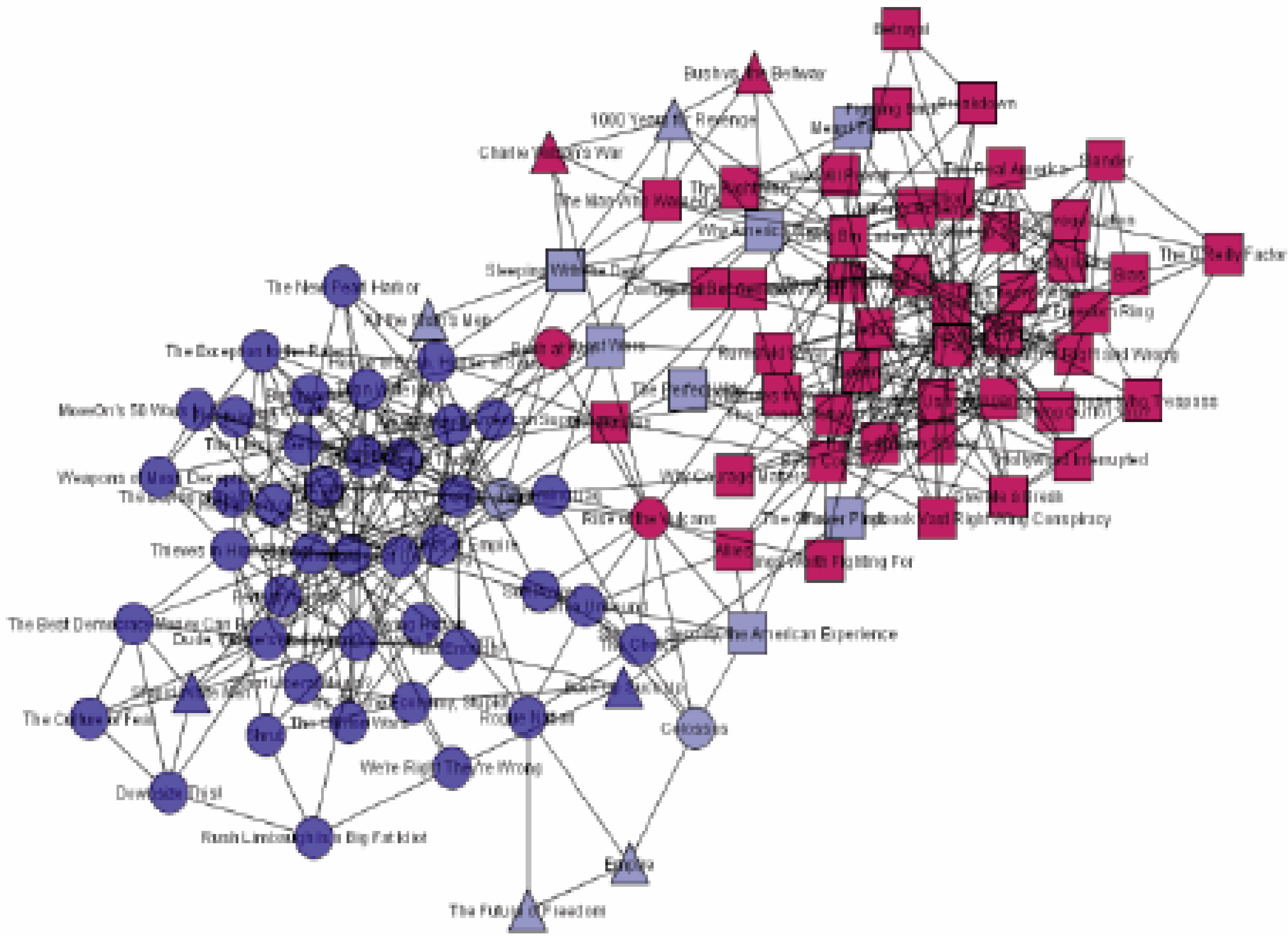


Figure 9. The Result of SCAN on Political Book Graph.

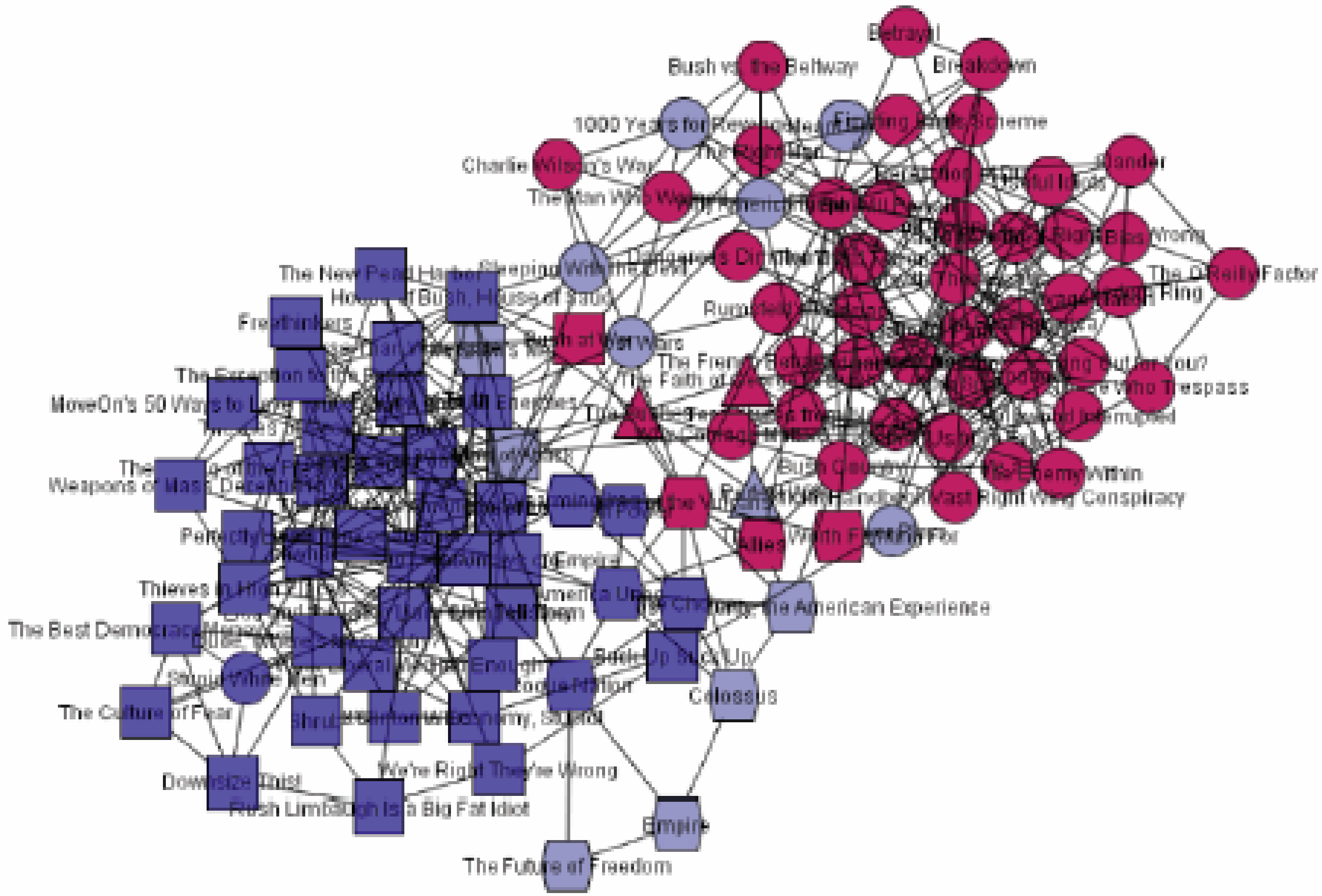


Figure 10. The Result of FastModularity on Political Book Graph.

Choosing ε

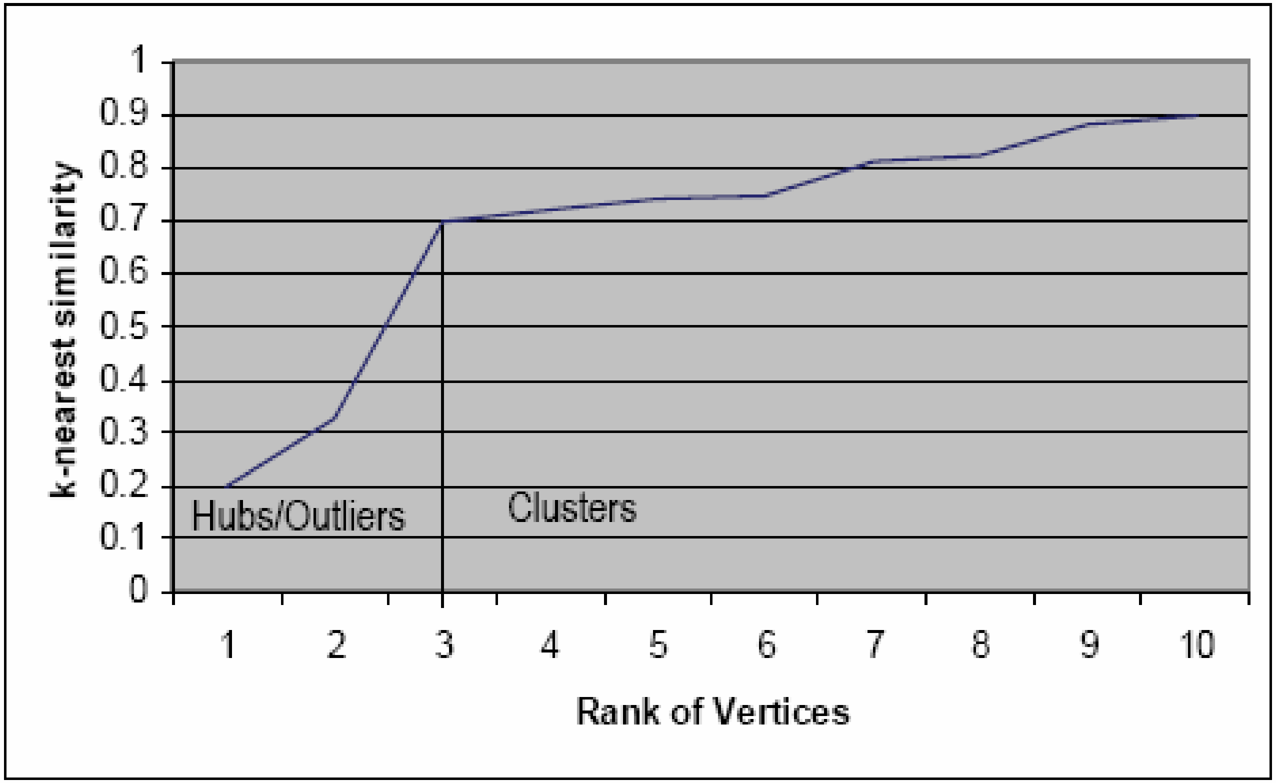


Figure 14. Sorted k-Nearest Structural Similarity.

Adjusted Rand Index (ARI)

$$\frac{\sum_{i,j}\binom{n_{ij}}{2} - \left[\sum_i \binom{n_{i\cdot}}{2} \sum_j \binom{n_{\cdot j}}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{n_{i\cdot}}{2} + \sum_j \binom{n_{\cdot j}}{2}\right] - \left[\sum_i \binom{n_{i\cdot}}{2} \sum_j \binom{n_{\cdot j}}{2}\right] / \binom{n}{2}}.$$

	SCAN	**FastModularity**
College football	**1**	0.24
Political books	**0.71**	0.64