NHH

INSTITUTT FOR FORETAKSØKONOMI

DEPARTMENT OF BUSINESS AND MANAGEMENT SCIENCE

FOR 40 2014

December 2014

Discussion paper

Centrality Computation in Weighted Networks Based on Edge-Splitting Procedure

BY

Ivan Belik AND **Kurt Jörnsten**

Norges Handelshøyskole

NORWEGIAN SCHOOL OF ECONOMICS

CENTRALITY COMPUTATION IN WEIGHTED NETWORKS BASED ON EDGE-SPLITTING PROCEDURE

Ivan Belik, Kurt Jörnsten

Norwegian School of Economics, Helleveien 30, 5045 Bergen, Norway
Email address: Ivan.Belik@nhh.no, Kurt.Jornsten@nhh.no

Abstract

The analysis of network's centralities has a high-level significance for many real-world applications. The variety of game and graph theoretical approaches has a paramount purpose to formalize a relative importance of nodes in networks. In this paper we represent an algorithm for the centrality calculation in the domain of weighted networks. The given algorithm calculates network centralities for weighted graphs based on the proposed procedure of edges' splitting. The approach is tested and illustrated based on different types of network topologies.

Keywords: network centrality, weighted graphs, edge splitting

1. INTRODUCTION

The variety of approaches for the analysis of network centralities has a purpose to understand and formalize a relative importance of nodes on graphs (i.e., networks). The classical approaches are based on the structural measurements that come from graph theory. Centralities based on degree (Freeman, 1979), closeness (Beauchamp, 1965; Sabidussi, 1966), betweenness (Anthonisse, 1971; Freeman, 1977), and information (Wasserman, 1994) are the most common measures. Also there exist centrality measures for the analysis of node's importance in complex networks: percolation centrality (Piraveenan, Prokopenko, & Hossain, 2013), cross-clique centrality (Faghani & Nguyen, 2013), etc. Details about each of these network centralities can be found in the corresponding literature. However, it is important to notice that the majority of the mentioned topological measures are based on two main concepts: (a) the analysis of different types of flows transferred across a network (Borgatti, 2005), and (b) the analysis of cohesiveness of a network (Borgatti & Everett, 2006).

There is a family of measures that are constructed based on the combination of graph and game theoretic concepts (Gómez, González-Arangüena, Manuel, Owen, del Pozo, & Tejada, 2003; Suri & Narahari, 2008). Specifically, Shapley Value (Shapley, 1952; Littlechild & Owen, 1973; Gul, 1989), which is considered as one of the most important concepts in coalition games, was employed by Aadithya, Ravindran, Michalak, & Jennings (2010) to measure an importance (i.e., centrality) of nodes in networks. According to Aadithya et al. (2010) the equation of Shapley value (SV) for player *i* in the coalition game with *n* players is the following:

$$\mathrm{SV_i(v)} = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(n-|S|-1)!}{n!} \big(v(S \cup \{i\}) - v(S)\big), \quad (1)$$

where:
N is the set of *n* players;
v is the characteristic function: $2^N \to \mathbb{R}$; $v(\emptyset) = 0$.

Aadithya et al., 2010 introduced the idea of SV application “in the domain of networks, where it is used to measure the importance of individual nodes, which is known as game theoretic network centrality”.
Consider graph *G(V,E)* and $v_i \in V$. Then $N_G(v_i)$ denotes the set of all neighboring nodes that are reachable from v_i in at most one hop within *G(V,E)*.
The degree of node v_i is defined by $deg_G(v_i)$. According to Aadithya et al. (2010) The SV interpretation for node v_i in *G(V,E)* is the following:

$$\mathrm{SV}(\mathrm{v_i}) = \sum_{\mathrm{v_j} \in \{\mathrm{v_i}\} \cup \mathrm{N_G}(\mathrm{v_i})} \frac{1}{1+\mathrm{deg_G}(\mathrm{v_j})}, \qquad (2)$$

The formulation represented in equation (2) “is an exact closed-form expression for computing the SV of each node on the network” (Aadithya et al., 2010).

2. CENTRALITY MEASURE IN WEIGHTED NETWORKS

Since weights play an important role in networks it is necessary to consider them as one of the key factors when measuring centralities. We assume that nodes in graphs with the same structure, but with different weights should have different centralities depending on the given weights. Inspired by ideas represented in Aadithya et al. (2010) we developed the algorithm to measure the importance of nodes in weighted graphs.

The core idea of the algorithm is to split edges into sub-edges based on their weights. Specifically, we split an edge with weight *w* into *w*-number sub-edges, where the weight of each sub-edge is equal to “1”.
Consider the weighted graph G(V,E) where for each node (u,v) the specific weight *w* is assigned. The algorithm of edges’ split into single-weighted (i.e., *w*=1) sub-edges is the following:

WEIGHT-SPLIT G(V,E)
1 **FOR** each edge $(\mathrm{u},\mathrm{v}) \in \mathrm{G.E}$ **do**:
2 Split edge (u,v) into w-number of sub-edges $\{(\mathrm{u},\mathrm{v})'_1, \dots, (\mathrm{u},\mathrm{v})'_w\}$;
3 **FOR** i=1 to w **do**:
4 $w'(\mathrm{u},\mathrm{v})'_\mathrm{i}=1$;
5 **end**
6 **end**
7 return WEIGHT-SPLITTED-GRAPH $\mathrm{G}'(\mathrm{V}',\mathrm{E}')$

Note: $\forall(\mathrm{u},\mathrm{v}) \in \mathrm{G.E}$: $\sum_{\mathrm{i}=1}^{w} w'(\mathrm{u},\mathrm{v})'_\mathrm{i} = w(\mathrm{u},\mathrm{v})$; i={1,…, w };

According to lines 2-4 of WEIGHT-SPLIT G(V,E) each edge of graph G(V,E) has to be split into *w* sub-edges, where each sub-edge has a weight of “1”.
The trivial example of how WEIGHT-SPLIT G(V,E) works is represented in Figure 1.

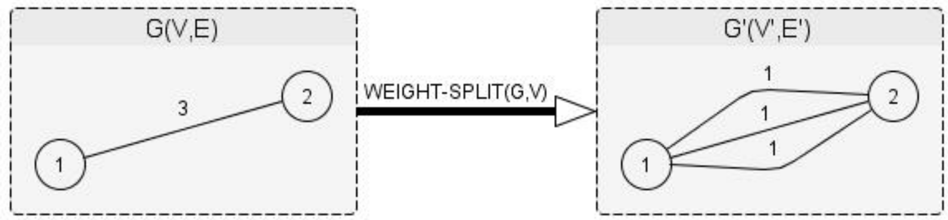

Figure 1. Edge-splitting procedure

According to Figure 1 there is a graph G(V,E) that consists of two nodes and the edge between them with the weight of “3”. Applying WEIGHT-SPLIT G(V,E) we get a new graph $G'(V', E')$ with three sub-edges, where each sub-edge has a weight equal to 1.

The WEIGHT-SPLIT G(V,E) procedure is the initial stage of the algorithm that calculates centrality measures for each node of a graph.
We consider the centrality measure of a node as its individual level of importance on the network. Specifically, we introduce the algorithm for the centrality measurement in weighted networks. Since the given algorithm is based on the idea of splitting weighted edges into sub-edges, we call the proposed centrality measure as Split-based Centrality (SC). The algorithm for SC-calculation is the following:

CENTRALITY-COMPUTING G(V,E)

```
WEIGHT-SPLIT G(V,E)
Total=0;
FOR each vertex u ∈ G.V do:
    V[u]=0;
    FOR each edge (u,v) ∈ G.E do:
        FOR each sub-edge (u,v)' ∈ G'.E' do:
            V[u]+= 1/(1+Neig(u)) + 1/(1+Neig(v));
        end
    end
    Total=Total + V[u];
end
FOR each vertex u ∈ G.V do:
    SC[u]= V[u]*|G.V| / Total ;
end
return all SCs
```

<u>Notation:</u>

V(node) is an intermediate non-normalized value for Split-based Centrality calculation;
Neig(node) is the number of neighboring vertices directly connected to the current node;
SC(node) is the Split-based Centrality measure;
|G.V| is the cardinality of the set of vertices G.V

To show how the algorithm works we consider a trivial example represented in Figure 2.

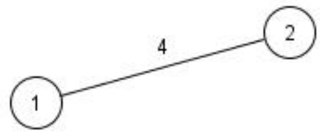

Figure 2. Illustrative example

We apply CENTRALITY-COMPUTING G(V,E) to calculate SCs in the graph represented in Figure 2. According to line 1 of the algorithm, we split the only edge (1,2) with *w*= 4 into four sub-edges. The weight of "1" is assigned to each sub-edge based on the WEIGHT-SPLIT algorithm (see Figure 3).

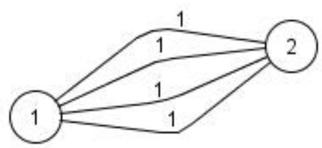

Figure 3. Illustrative example with split weight

Next, following lines 2-15 we calculate SCs for all vertices in the graph. Line 3 initiates the cycle to go through the list of all nodes in G(V,E). In line 4 we assign zero-value to V of the current vertex. Every time, when the algorithm starts to calculate V for the next vertex of the graph, line 4 resets V to zero-value. Line 5 initiates the process of going through the list of all edges of the current vertex. Line 6 extracts all sub-edges $(u, v)' \in G'.E'$ that are encapsulated in the currently processed edge $(u, v) \in G.E$. Line 7 accumulates all V sub-values going through all edges and sub-edges of the currently analysed vertex $u \in G.V$. It is important to notice that the idea of V-calculation (as the main component of SC) is interrelated with the formalization of SV represented in equations (1)-(2). Next, we tot up Vs of all vertices $u \in G.V$ getting Total-value in line 10. In lines 12-14 we normalize the results calculating SCs for each vertex $u \in G.V$. Finally, in line 15 we get SCs of all vertices in G(V,E).

For our trivial example represented in Figures 2-3 we get SC[1]=SC[2]=1.

In section 3 we represent SC-calculation results applying the algorithm in different network topologies.

3. APPLICATION IN DIFFERENT NETWORK TOPOLOGIES

3.1 "Point-to-point" or "Line" topology

The initial structure of the weighted "point-to-point" graph is represented in Figure 4. It is characterized by two nodes with the following weights: $w(1,2)=4$ and $w(2,3)=5$. Applying the CENTRALITY-COMPUTING algorithm we get the graph with split weights represented in Figure 5. The results of SC calculations are represented in Table 1. Since node 1 has the only neighbor node 2 (i.e., Neig(1)=1) and $w(1,2)=4$ we get four sub-edges connecting node 1 and node 2. Similarly, node 3 has the only neighbor node 2, but $w(2,3)=5$. In this case we get five sub-edges connecting node 3 and node 2. Based on the CENTRALITY-COMPUTING algorithm we have SC(1)=0.667 and SC(3)=0.833. Both nodes are connected to node 2 only, but the weights of initial links (i.e., edges) are different. As the result SC(1)<SC(3). Node 2 connected to both mentioned nodes, and it has nine sub-edges in total. Playing the role of a hub it gets the highest SC=1.5.

Table 1. SCs calculation for the "point-to- point" topology

#	node	Neig	SC
1	1	1	0.667
2	1	1	
3	1	1	
4	1	1	
5	2	2	1.500
6	2	2	
7	2	2	
8	2	2	
9	2	2	
10	2	2	
11	2	2	
12	2	2	
13	2	2	
14	3	1	0.833
15	3	1	
16	3	1	
17	3	1	
18	3	1	

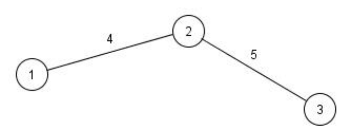

Figure 4. Weighted "Point-to-point" graph

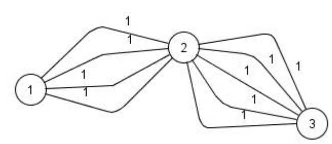

Figure 5. "Point-to-point" graph with split weights

3.2 "Star" topology

"Star" topology is characterized by the existence of a centric node, which is connected to all other nodes on the network. All nodes, excluding the centric one, form an independent set (Robson, 1986; Boppana & Halldórsson, 1992; Merris, 2003). The "star" graph with established weights is represented in Figure 6. It is characterized by three links with the following weights: $w(1,2)=3$, $w(1,3)=2$, $w(1,4)=4$. Applying CENTRALITY-COMPUTING to the given graph we get the graph with split weights (see Figure 7) and SCs computational results (see Table 2). Based on the corresponding weights, edge (1,2) is split into three sub-edges, edge (1,3) – into two and edge (1-4) – into four. Nodes 2, 3 and 4 are connected to node 1 only, but weights of their initial links are different. The computational results are the following: SC(2) = 0.667; SC(3)=0.444; SC(4)=0.889. Node 1 is connected to all others, and it is characterized by 18 sub-edges in total. Playing the role of a hub it gets the highest SC=2.0.

Table 2. SCs calculation for the “star” topology

#	node	Neig	SC
1	1	3	2.000
2	1	3	
3	1	3	
4	1	3	
5	1	3	
6	1	3	
7	1	3	
8	1	3	
9	1	3	
10	2	1	0.667
11	2	1	
12	2	1	
13	3	1	0.444
14	3	1	
15	4	1	0.889
16	4	1	
17	4	1	
18	4	1	

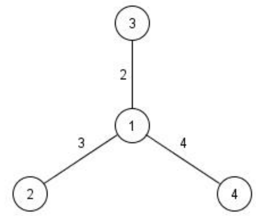

Figure 6. Weighted “star” graph

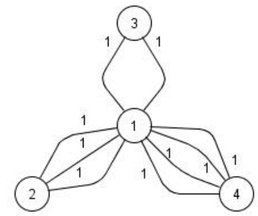

Figure 7. “Star” graph with split weights

3.3 “Ring” topology

“Ring” topology is characterized by sequential connection of each node to exactly two other nodes forming a cycle. There is no leading node in a “ring”-based structure, and the deletion of at least one edge may cause the destructive effect of the overall network. We consider two types of “ring” topologies: (a) graph with even number of nodes (b) graph with odd number of nodes.

3.3.1 “Ring” topology with even number of nodes

The graph with assigned weights is represented in Figure 8. It consists of four nodes connected by edges with the following weights: $w(1,2)=2$, $w(2,3)=3$, $w(3,4)=4$, $w(4,1)=5$. Applying the CENTRALITY-COMPUTING algorithm we get a modified graph based on weights’ split (see Figure 9). The resulting SCs are represented in Table 3. Node 2 gets the smallest SC=0.714 based on five sub-edges created within two links: (1,2) and (2,3). Node 4 has the highest SC=1.286. It is characterized by two “heaviest” links (i.e., $w(3,4)=4$ and $w(4,1)=5$) that split into nine sub-edges.

Table 3. SCs calculation for the "ring" topology with even number of nodes

#	node	Neig	SC
1	1	2	1.000
2	1	2	
3	1	2	
4	1	2	
5	1	2	
6	1	2	
7	1	2	
8	2	2	0.714
9	2	2	
10	2	2	
11	2	2	
12	2	2	
13	3	2	1.000
14	3	2	
15	3	2	
16	3	2	
17	3	2	
18	3	2	
19	3	2	
20	4	2	1.286
21	4	2	
22	4	2	
23	4	2	
24	4	2	
25	4	2	
26	4	2	
27	4	2	
28	4	2	

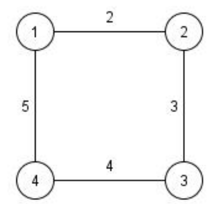

Figure 8. Weighted "ring" graph with even number of nodes

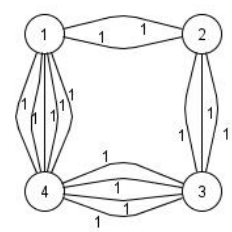

Figure 9. "Ring" graph (even number of nodes) with split weights

3.3.2 "Ring" topology with odd number of nodes

We calculate SCs for the "ring" graph with odd number of nodes. The initial graph structure is represented in Figure 10. It consists of five nodes connected by five edges with the following weights: $w(1,2)=2$, $w(2,3)=3$, $w(3,4)=4$, $w(4,5)=5$ and $w(5,1)=1$. Applying the CENTRALITY-COMPUTING algorithm we get the graph with split weights (see Figure 11) and the resulting SCs (see Table 4).

Table 4. SC calculations for the "ring" topology with odd number of nodes

#	node	Neig	SC
1	1	2	0.500
2	1	2	
3	1	2	
4	2	2	0.833
5	2	2	
6	2	2	
7	2	2	
8	2	2	
9	3	2	1.167
10	3	2	
11	3	2	
12	3	2	
13	3	2	
14	3	2	
15	3	2	
16	4	2	1.500
17	4	2	
18	4	2	
19	4	2	
20	4	2	
21	4	2	
22	4	2	
23	4	2	
24	4	2	
25	5	2	1.000
26	5	2	
27	5	2	
28	5	2	
29	5	2	
30	5	2	

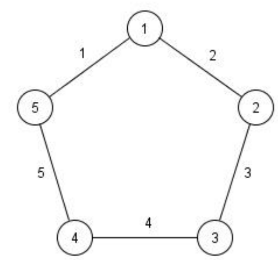

Figure 10. Weighted "ring" graph with odd number of nodes

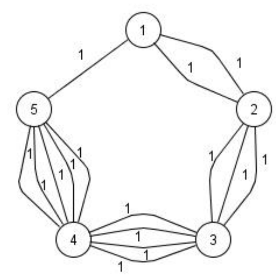

Figure 11. "Ring" graph (odd number of nodes) with split weights

3.4 Mixed topology

We consider a graph as a mixed structured if it is represented by different combinations of trivial topologies, such as "point-to-point", "ring" and "star". The initial graph structure is represented in Figure12. It includes seven nodes connected by the following links with assigned weights:

$w(1,2)=5$; $w(4,5)=4$;
$w(1,3)=2$; $w(5,6)=4$;
$w(2,3)=3$; $w(5,7)=2$;
$w(3,4)=2$; $w(6,7)=3$.

Based on the CENTRALITY-COMPUTING algorithm we get the graph with split weights (see Figure 13). SC computational results are represented in Table 5. Node 5 has got the highest SC=1.339 based on three links split into ten sub-edges. Node 7 is the node with the minimum SC-value: SC(7)=0.727. It is based on the split of two links ($w(5,7)=2$ and $w(6,7)=3$) into 5 sub-edges in total.

Table 5. SCs calculation for the mixed topology

#	node	Neig	SC
1	1	2	1.033
2	1	2	
3	1	2	
4	1	2	
5	1	2	
6	1	2	
7	1	2	
8	2	2	1.167
9	2	2	
10	2	2	
11	2	2	
12	2	2	
13	2	2	
14	2	2	
15	2	2	
16	3	3	0.937
17	3	3	
18	3	3	
19	3	3	
20	3	3	
21	3	3	
22	3	3	
23	4	2	0.803
24	4	2	
25	4	2	
26	4	2	
27	4	2	
28	4	2	
29	5	3	1.339
30	5	3	
31	5	3	
32	5	3	
33	5	3	
34	5	3	
35	5	3	
36	5	3	
37	5	3	
38	5	3	
39	6	2	0.995
40	6	2	
41	6	2	
42	6	2	
43	6	2	
44	6	2	
45	6	2	
46	7	2	0.727
47	7	2	
48	7	2	
49	7	2	
50	7	2	

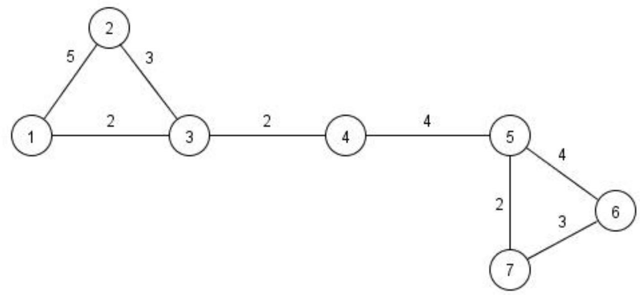

Figure 12. Weighted mixed graph

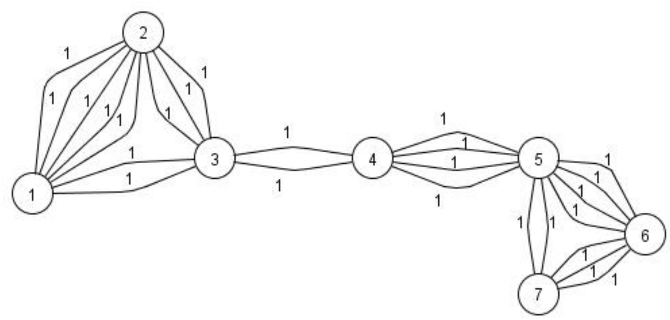

Figure 13. Mixed graph with split weights

4. CONCLUSION

In the given paper we represented an algorithm for centrality calculation in weighted networks. The algorithm consists of two sub-components. The first component (i.e., WEIGHT-SPLIT) processes edges' weights. Specifically, it splits network links into sub-links (i.e., sub-edges) based on the corresponding weights. This procedure is polynomially executable in terms of running time. As the result of the first component execution, we get a modified graph to be processed for the Split-based Centrality calculation. The second subcomponent of the CENTRALITY-COMPUTING algorithm is based on the weight-splitting concept. Going through all edges and through the corresponding sub-edges of each node in the network, we accumulatively calculate SC-values. Therefore, the procedure has an iterative nature running in polynomial time to calculate SC-values for every node on the network. Next, we apply the CENTRALITY-COMPUTING algorithm to different types of network topologies, such as "point-to-point", "star", "ring", and "mixed". The results are represented in tabular and graphical formats.

Analyzing network centralities it is necessary to consider edges' weights as an important factor as well as the structural factor. In this paper we maintain a statement that two networks with identical structures but with different weights are expected to have different levels of importance depending on the corresponding weights of links. The represented algorithm is based on this statement.

REFERENCES

Aadithya, K. V., Ravindran, B., Michalak, T. P., & Jennings, N. R. (2010). Efficient Computation of the Shapley Value for Centrality in Networks. In *Internet and Network Economics* (pp. 1-13). Springer Berlin Heidelberg.

Anthonisse, J. M. (1971). *The Rush in a Directed Graph.* Tech. Rep. BN 9/71, Stichting Mathematisch Centrum, 2e Boerhaavestraat 49 Amsterdam.

Beauchamp, M. A. (1965). An Improved Index of Centrality. *Behavioral Science, 10*, 161–163.

Borgatti, S. P. (2005). Centrality and network flow. *Social networks*, *27*(1), 55-71.

Borgatti, S. P., & Everett, M. G. (2006). A graph-theoretic perspective on centrality. *Social networks*, *28*(4), 466-484.

Boppana, R., & Halldórsson, M. M. (1992). Approximating maximum independent sets by excluding subgraphs. *BIT Numerical Mathematics*, *32*(2), 180-196.

Faghani, M.R.; Uyen Trang Nguyen, "A Study of XSS Worm Propagation and Detection Mechanisms in Online Social Networks," *Information Forensics and Security, IEEE Transactions on* , vol.8, no.11, pp.1815,1826, Nov. 2013

Freeman, L. C. (1977). A Set Of Measures of Centrality Based upon Betweenness. *Sociometr*y, *40*, 35–41.

Freeman, L. C. (1979). Centrality in Social Networks: Conceptual Clarification. *Social Networks, 1*, 241–256.

Gómez, D., González-Arangüena, E., Manuel, C., Owen, G., del Pozo, M., & Tejada, J. (2003). Centrality and power in social networks: a game theoretic approach. *Mathematical Social Sciences*, *46*(1), 27-54.

Gul, F. (1989). Bargaining foundations of Shapley value. *Econometrica: Journal of the Econometric Society*, 81-95.

Littlechild, S. C., & Owen, G. (1973). A simple expression for the Shapley value in a special case. *Management Science*, *20*(3), 370-372.

Merris, R. (2003). Split graphs. *European Journal of Combinatorics*, *24*(4), 413-430.

Piraveenan, M., Prokopenko, M., & Hossain, L. (2013). Percolation centrality: Quantifying graph-theoretic impact of nodes during percolation in networks. *PloS one*, *8*(1), e53095.

Robson, J. M. (1986). Algorithms for maximum independent sets. *Journal of Algorithms*, *7*(3), 425-440

Sabidussi, G. (1966). The Centrality Index of a Graph. *Psychometrika, 31*, 581–603.

Shapley, L. S. (1952). *A value for n-person games* (No. RAND-P-295). RAND CORP SANTA MONICA CA.

Suri, N. R., & Narahari, Y. (2008, May). Determining the top-k nodes in social networks using the shapley value. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 3* (pp. 1509-1512). International Foundation for Autonomous Agents and Multiagent Systems.

Wasserman, S. (1994). *Social Network Analysis: Methods and Applications*(Vol. 8). Cambridge University Press.