Journal of Computational Science 62 (2022) 101738

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

Comparative analysis of centrality measures for identifying critical nodes in complex networks

Onur Ugurlu

Faculty of Engineering and Architecture, Izmir Bakircay University, Izmir, Turkey

ARTICLE INFO

Keywords:
Complex networks
Robustness
Centrality measures
Critical nodes

ABSTRACT

One of the fundamental tasks in complex networks is detecting critical nodes whose removal significantly disrupts network connectivity. Identifying critical nodes can help analyze the topological characteristics of the network, such as vulnerability and robustness. This work considers a well-known critical node detection problem variant, Maximize the Number of Connected Components Problem, which aims to find a set of nodes whose removal maximizes the number of connected components and compares the centrality measures for detecting these nodes. While the existing literature focused only on small datasets, this work analyzes the widely used topology-based centrality measures on several synthetic and real-world networks. Our findings show that degree-like centralities are more relevant measures than path-like centralities for disconnecting networks into several connected components. However, our results also indicate that the traditional centrality measures cannot detect the most vital critical nodes. To overcome this drawback, a new centrality measure, namely Isolating Centrality, that aims to identify the nodes that significantly impact network connectedness is presented. The comprehensive computational study demonstrates that the proposed measure outperforms traditional measures in identifying critical nodes.

1. Introduction

Computer networks, social networks, road networks, and communication networks are examples of complex networks playing essential roles in daily life. Most real-life networks display two important complex network models: Scale-free and Small-world [1]. A network is scale-free [2] if the distribution of the number of nodes with a particular degree decays like a power law which means that most vertices have a small number of connections, while a few nodes have massive connections. US airports network can be given as an example of scale-free networks. Some airports, such as New York and Chicago, have connections with almost all other airports in this network. However, most airports have few connections with the other airports [3]. On the other hand, small-world networks [4] have a low diameter and most nodes are reachable from any other nodes in a relatively small number of links. Furthermore, these networks can be highly clustered, like regular lattices. The most important example of small-world networks is social networks. For example, people are friends with neighbors on the same street. Besides, some people have a few friends in other countries that provide connections between the communities. Different nodes have distinct impacts on network robustness and vulnerability against failures in complex networks. Therefore, finding the vital nodes, particularly for network connectivity, plays a significant role in examining many basic structural features of networks. Such nodes are known as "Critical Nodes" in the literature. The primary purpose of detecting critical nodes is to identify the groups that will negatively impact the network reliability in their absence and ensure that this group is protected or strengthened.

Critical node detection problems are optimization problems that aim to find the set of nodes whose deletion causes the most damage to the network connectivity, according to some predetermined measures. Thus, a node that is critical for a particular measure may not be critical for other measures. The most studied connectivity measures in the literature are the number of connected components, the largest connected component size, and pairwise connectivity, i.e., the number of pairs of nodes connected. This paper focuses on the critical nodes whose removal maximizes the number of connected components in the residual graph, known as the Maximize the Number of Connected Components (MaxNum) problem. The provided definition for the critical nodes has several applications in real-life networks, such as rumor spreading interception, traffic congestion control, virus and infectious disease prevention and the community detection [5]. Hence, finding accurately and effectively critical nodes in complex networks is an urgent and challenging research issue.

Another fundamental concept in networks is centrality. Centrality measures aim to determine the relative importance of the nodes. Considering the above definitions, the correlation between the critical and

E-mail address: onur.ugurlu@bakircay.edu.tr.

https://doi.org/10.1016/j.jocs.2022.101738
Received 26 February 2022; Received in revised form 23 April 2022; Accepted 24 May 2022
Available online 14 June 2022

central nodes is worth studying. Thus, many researchers investigate the efficiency of using central nodes for finding the critical nodes. The most used measures in this field are degree centrality, closeness centrality, and betweenness centrality. However, these measures have advantages and disadvantages. For example, degree centrality has low computing complexity, but it only uses local information of the graphs. On the other hand, closeness centrality and betweenness centrality have high computational complexity while using the global graph structure. In addition, traditional centrality measures could fail to detect some critical nodes in complex networks [6]. Since computing betweenness or closeness centrality of large-scale networks is very time-consuming, a new method called k-shell/k-core centrality was proposed. K-shell ranks the nodes based on the network coreness of the nodes by dividing the network into layers or shells. Nodes with the highest k-shell value, known as the core nodes, are the most central nodes. The most crucial drawback of k-shell is assigning many nodes to the same rank. Moreover, some studies show that some nodes with the small k-shell value are more important than the core nodes with the highest k-shell value [7]. Many researchers have studied the k-shell method in recent years and have developed hybrid centrality methods by combining k-shell with other centrality measures such as degree and closeness [8, 9].

Intending to overcome these drawback, this work presents a new centrality measure, namely Isolating Centrality, that aims to identify the nodes that significantly impact network connectedness. The core idea of isolating centrality is defining the values of nodes by integrating local and global information. A high isolating centrality value for a node indicates that the node is between the internal (inner) nodes and the terminal (outer) nodes. We compare the performance of the topology-based measures and the proposed measure in detecting critical nodes on several synthetic and real-world networks. Moreover, we include leverage centrality in the analysis since it is closely related to network connectivity. The main contributions of this paper can be summarized as follows:

- We introduce a novel centrality measure that reveals the important nodes for the connectedness of a network.
- The time complexity of the proposed centrality is $O(n^2)$, where n is the number of the nodes.
- The correlation between leverage centrality and critical nodes is investigated for the first time.
- Through extensive computational experiments, the performances of several centrality measures are evaluated. Our findings on the synthetic networks and real-world networks clearly demonstrate that the proposed centrality measure outperforms its counterparts. Moreover, especially for small-world networks, which are the most challenging instances for critical node problems, the proposed measure could find up to 4 times better solutions than the traditional ones.

The rest of the paper is organized as follows: Section 2 provides a brief survey of the existing research. Section 3 introduces the basic notations and definitions, and Section 4 presents the proposed centrality measure. The solution framework and the comprehensive computational results are given in Section 5. Finally, Section 6 closes the paper with some concluding remarks and ideas for future research.

2. Literature review

The most studied variants of critical node detection problems are the Critical Node Problem which aims to minimize pairwise connectivity (CNP), Minimize the Largest Component Size Problem (MinMaxC), and the MaxNum. Although the MaxNum are interesting problem and has several real-life applications, they have not attracted much attention as the CNP and MinMaxC. The MaxNum problem, which seeks to maximize the total number of connected components in graphs, was introduced by Shen et al. [10] in 2012. If the size of the connected components is limited to one in the residual graph, the MaxNum problem transforms into the Maximum Independent Set problem. By this transformation, Shen et al. [11] proved that the MaxNum is NP-Hard on general graphs. Berger et al. [12] also showed the NP-Completeness of the MaxNum through a reduction from the Minimum k-cut Problem. Note that the Minimum k-cut problem is the edge version of the MaxNum.

From the solution perspective, early studies were proposed by [10, 11]. They developed a mixed-integer program formulation that can find solutions for small networks of at most 50 nodes. The same authors proposed polynomial-time algorithms based on dynamic programming for particular graph classes. Veremyev et al. [13] presented another mixed-integer program formulation that can find solutions for relatively large real-life networks up to 120 nodes. Lastly, Aringhieri et al. [14] considered the CNP, MinMaxC, and MaxNum and developed a general Evolutionary Framework for solving these problems. They also proposed greedy rules for adding and deleting procedures to construct feasible solutions. To reduce a solution set with high cardinality, the authors remove a node from the solution set when the node connects a minimum collection of connected components. For adding a node to a solution set, they select the node whose removal maximized the number of connected components.

Another main aspect of this work is centrality which is one of the most popular research areas of network analysis. Although the early studies of centrality had applications in social networks, researchers have begun to use centrality measures to analyze the vulnerability of networks over the past two decades. The first studies in this context focused on identifying and removing the nodes that play a significant role in connecting the network. It has been shown in [15] that removing the nodes with high centrality is an effective method for dividing the network into independent components. Holme et al. [16] investigated the performance of degree and between centralities on dividing complex networks with regard to average inverse path length and the largest component size. They reported that considering the recalculated degrees and betweenness centralities is a more effective approach than removing the nodes based on the initial centrality values.

Nasiruzzaman et al. [17] modified the degree, closeness and betweenness centralities for electrical topology rather than physical topology. They tested these measures on IEEE test systems for finding critical nodes. They concluded that the degree centrality is the most efficient measure. Ghanbari et al. [18] examined the correlation between cascade failures and centrality measures in complex networks. They found that nodes with high degree centrality values have small cascade depths, whereas nodes with high betweenness centrality values have large cascade depths. Since degree centrality is based on local information, these results can be considered a natural consequence. Veremyev et al. [19] studied the distance-based CNP, which considers the actual pairwise distance between nodes. They developed linear integer programming formulations to obtain the optimal solution for the problem and compare the results of different centrality measures with optimal solutions. They conducted detailed computational analysis on several real-life and randomly generated networks and reported that centrality measures could not detect some critical nodes for distance-based CNP. Gillen et al. [20] consider the same problem with imperfect data where there is some fake or misinformation about nodes or edges. Again, they compared different centrality measures with exact methods. Their results indicated that degree and betweenness have the best performance among the centrality measures, yet exact methods are still the best options for imperfect data.

Nomikos et al. [21] consider the three different critical nodes measures on the ISP (Internet Service Provider) network: average path length, size of the largest component, and the number of connected components. They tested several centrality measures for these problems by removing the nodes simultaneously after being ranked centrality values in decreasing order. For the MaxNum problem, degree centrality had the best solutions. Shen Yilen et al. [22] proposed an algorithm for the CNP and compared their algorithm with the degree and betweenness centrality and optimal solutions on Waxman, Power-law and terrorist networks. For these networks, the degree and betweenness centrality found similar solutions. Same authors [23] studied the MinMaxC on Interdependent Power Networks and a greedy framework with centrality functions. Again, they compare their algorithms with the centrality measures. For the MinMaxC, closeness centrality had the best solutions for interdependent power networks. Lastly, another study comparing several centrality measures for the MinMaxC on complex networks was presented by Iyer et al. [24].

Although several centrality measures exist in the literature, any centrality measure is not relevant for all network problems and applications. Thus, as new problems arise in network theory, researchers continue to propose new centrality measures or modify existing ones relevant for the nature of these problems [25–27]. Besides its relevance to the problem and application, another significant parameter for a centrality measure is its computational complexity. Nowadays, the sizes of real-life networks such as social networks and traffic networks have reached massive numbers. Hence, finding the central elements in large-sized networks in reasonable times is crucial for applicability. Thus, this work proposes a new centrality measure that reveals the important nodes for the connectedness of networks with reasonable computation times. Moreover, only a few studies examine the correlation between centrality measures and the MaxNum problem in the literature. Hence, we believe that our results have a valuable contribution to the critical nodes literature.

3. Basic notations and definitions

In this section, we briefly recall some definitions and introduce the existing centrality measures. Let $G = (V, E)$ be an undirected graph with node set V and edge set $E \subseteq V \times V$, where n denotes the number of nodes and m denotes the number of edges in G. The density of a graph is the number of edges over the number of all possible edges, and the average degree is the product of the density and the number of nodes. An articulation point (or cut vertex) is a node whose removal disconnects the graph. The neighborhood of node i consists of its directly connected nodes, represented by $N(i)$, and the degree of node i, denoted by $d(i)$, is the size of $N(i)$. $\delta = min\{d(i) : \forall i \in V\}$ and $\Delta = max\{d(i) : \forall i \in V\}$ are the minimum and maximum degree of nodes in a graph, respectively. Finally, let Deg_i be the set nodes having degree i.

Given a graph $G = (V, E)$ and an integer k, the Maximize the Number of Connected Components problem seeks to find a set of at most k nodes $S \subseteq V$ whose removal maximizes the number of connected components in the residual graph.

Fig. 1 shows a sample graph with 12 nodes and 16 edges. By removing nodes 3 and 8, the graph can be separated into four connected components, which is an optimal solution of the MaxNum for $k = 2$. Note that $S = \{6, 8\}$ is the other optimal solution.

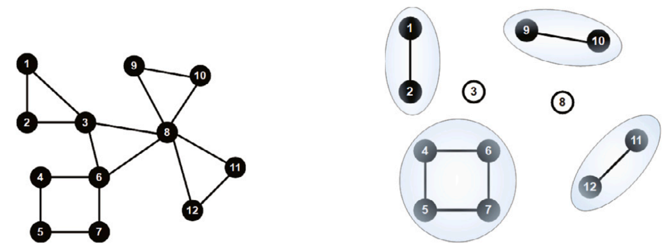
In this study, we consider four different centrality measures. These measures can be categorized as degree-like centralities (degree and leverage centrality) and path-like centralities (betweenness and closeness centrality).

Degree Centrality (DC) is the most straightforward centrality measure. It is based on the idea that the most connected node is the most central node. The degree centrality of a node is defined as follows:

$$C_D(i) = d(i)$$

Leverage Centrality (LC) considers the extent of connectivity of a node relative to the connectivity of its neighbors [28]. If a node has a higher degree among its neighbors, it is likely to have high leverage centrality. The leverage centrality of a node is defined as follows:

$$C_L(i) = \frac{1}{d(i)} \sum_{j \in N_i} \frac{d(i) - d(j)}{d(i) + d(j)}$$



(a) (b)

Fig. 1. (a) A sample graph with 12 nodes (b) An optimal solution of the MaxNum problem for $k = 2$.

Betweenness Centrality (BC) is one of the most popular centrality measures in network literature [29]. It is based on the shortest paths in networks. The nodes among the shortest paths are more central. The betweenness centrality of a node is defined as follows:

$$C_B(i) = \sum_{j,k \in \mathbf{V}, j \neq k \neq i} \frac{\sigma_{jk}(i)}{\sigma_{jk}}$$

where σ_{jk} is the number of shortest paths between nodes j and k and $\sigma_{jk}(i)$ is the number of how many times node i used in these paths. Note that the given definition of BC holds only for connected graphs. For disconnected graphs, BC is calculated separately for each connected component.

Closeness Centrality (CC) is based on the closeness between nodes [30]. A node is central when it can reach any node in the network in a few steps. The closeness centrality of a node is defined as follows:

$$C_C(i) = \frac{n - 1}{\sum_{j \in \mathbf{V}, j \neq i} dis(i, j)}$$

where $dis(i, j)$ represents the number of the edges on the shortest paths between nodes i and j. Similar to BC, the given definition of CC holds only for connected graphs since two nodes that belong to different components do not have a finite distance between them. To overcome this limitation, Wasserman and Faust [31] proposed an improved definition for disconnected graphs.

$$C_{C-WF}(i) = \frac{|H_i| - 1}{n - 1} \frac{|H_i| - 1}{\sum_{j \in \mathbf{H}_i, j \neq i} dis(i, j)}$$

where H_i represents the component of nodes i.

The time complexity of DC and LC is O(n^2), whereas the time complexity of BC and CC is O(nm) [32]. Naturally, since BC and CC need to calculate the shortest paths between all node pairs, they require more time complexity than DC and LC. Lastly, it is necessary to mention that DC, BC and CC have several normalized versions in the literature. However, since we consider the ranking of the nodes, normalized versions of the centrality measures do not affect our results.

4. Proposed centrality measure

Two different approaches can be followed to make any network disconnected by deleting a small number of nodes. The first approach is to find the minimum number of nodes (minimum node cut set) required to separate the network into two or more components and the second approach is to create isolated nodes that are not connected to any nodes. To generate isolated nodes, at least δ nodes must be deleted from the network. The first approach may need fewer nodes to be deleted to disconnect the network. However, it needs the global topology information, which increases the computation time. Whereas, in the second approach, the local topology information is sufficient to determine the nodes that need to be deleted. Note that the δ value sets

Fig. 2. (a) A sample graph with 9 nodes (b) Optimal solution of MaxNum problem for $k = 2$.

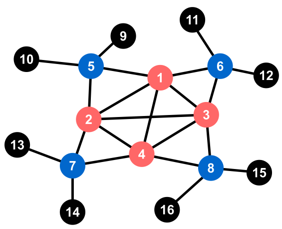


Fig. 3. Comparing degree centrality and Isolating centrality on the sample graph: The most central nodes for degree and isolating centrality are colored with red and blue, respectively.

an upper limit for the minimum node cut set size, and in some cases, these two values can be equal.

If we want to solve the MaxNum problem for $k = 2$ on the sample graph in Fig. 2(a), the optimal solution is 3. To disconnect the sample graph, the minimum number of nodes that must be deleted is 1 (node 5), and the δ value is 2 ($d(8) = d(9) = 2$). If we delete node 5 from the graph, the graph splits into two components. No matter which node is deleted after this step, the number of components in the residual graph will not change. However, if we want to disconnect the graph by creating isolated nodes, the neighbors of node 8 (or node 9), nodes 6 and 7 must be deleted. By deleting these nodes, the graph is divided into three components (Fig. 2(b)). Although at first glance it seems that the minimum node cut set is closely related to the MaxNum, Fig. 2 shows that it does not always give the optimal result. Creating isolated nodes may require more nodes than the minimum node cut set size, yet deleting the neighbors of the nodes with the minimum degree is a more relevant approach to maximize the number of connected components in the residual network. Besides being suitable for the MaxNum, creating isolated nodes requires much less time complexity than finding the minimum node cut set.

The biggest motivation of the proposed centrality measure is the detection of the nodes that will separate the network into several components with reasonable computation times. In this context, we introduce a new definition for nodes, namely Isolating Nodes, which is based on the degree of the node and its neighbor nodes with the minimum degree. An isolating node lies between the internal (inner) nodes and the terminal (outer) nodes (considering the highly connected nodes as inner nodes and low connected nodes as outer nodes). We define the isolating coefficient of a node by the number of its neighbor nodes with a degree of δ, which assesses the nodes' influence on disconnecting the network. Isolating Centrality (ISC) of a node is the product of its degree and its isolated coefficient. More specifically, the isolating centrality of a node is defined by:

$$C_{ISC}(i) = |N(i) \cap Deg_{\delta}| \times d(i)$$

It is worth mentioning that although isolating centrality uses degree centrality as a factor, there is a distinct difference between them. A high degree node is not highly central according to isolating centrality if it has no neighbor node with δ degree. For example, for the sample graph in figure Fig. 3, although nodes 1, 2, 3, and 4 (red nodes) have the highest degree, their isolating centrality values equal 0. If we remove these nodes, the graph will be separated into four components. Whereas the nodes with the highest isolating centrality values ($4 \times 2 = 8$) are nodes 5, 6, 7, 8 (blue nodes) and by deleting these nodes, the graph will be separated into nine components.

Fig. 4. A path graph with 5 nodes (b) Optimal solution of the MaxNum problem for $k = 2$.

Table 1
Comparison of the centrality values of the nodes for the path graph with 5 nodes.

Centrality measures	Node 1	Node 2	Node 3	Node 4	Node 5
Betweenness Centrality	4	3	3	0	0
Closeness Centrality	0.17	0.14	0.14	0.1	0.1
Degree Centrality	2	2	2	1	1
Leverage Centrality	0	0.33	0.33	−0.33	−0.33
Isolating Centrality	0	2	2	0	0

The biggest disadvantage of DC, BC and CC in detecting critical nodes for the network connectedness is that they do not consider the neighbor of nodes. Hence, they can fail to find the optimal solution, even on simple graphs.

When we try to solve the Maxnum problem for $k = 2$ on the path graph with five nodes in Fig. 4(a), the most central node for DC, BC and CC (considering the tie between nodes 1, 2, and 3 breaks by selecting smaller ID) is node 1. Table 1 gives the centrality values of the nodes. If DC, BC and CC are considered, node 1 must be deleted. After deleting this node, all nodes' DC, BC and CC values are equal and whichever node is deleted, the residual graph will contain two separate components. Whereas deleting the most central nodes for LC and ISC (nodes 2 and 3) will create three separate components.

Based on the definition of isolating centrality, removing the central nodes weakens or completely terminates the communication paths between some other groups of active nodes. Therefore, identifying the central nodes may help to increase the fault tolerance of the networks. For example, in wireless multi-hop networks, detecting the central nodes may reveal the potential bottlenecks or bridges that transfer most of the network traffic. After finding the bottlenecks, we may put the redundant nodes to create alternate paths between other nodes and resolve the bottlenecks. Another example, in a road network, we may build new roads that directly connect the neighbors of central nodes.

4.1. Complexity analysis

The time complexity of ISC can be determined straightforwardly: First, the degree of the nodes should be calculated, which needs $O(m)$ running time. Then, finding the minimum degree of the nodes requires $O(n)$ running time. Finally, calculating the isolating coefficient needs $O(\Delta n)$ running time, where Δ is the maximum degree of nodes. Therefore, the time complexity of ISC is $O(m+n+\Delta n) = O(n^2)$ asymptotically.

5. Computational study

To evaluate the performance of centrality measures in identifying critical nodes, we used two well-known benchmark sets. Moreover, we used random graphs with various densities to investigate the effect of the network intensity.

5.1. Solution framework

To test the correlation between critical nodes and central nodes, we employed a traditional greedy algorithm.

The greedy algorithm iteratively adds the nodes to set S. In each iteration, the node with the highest centrality value in the current graph is added to S and removed from the graph. The pseudocode of the greedy algorithm is given in Algorithm 1. In lines 3–24, isolating

Algorithm 1: Greedy Algorithm for solving the MaxNum Problem

```
Input : Graph G(V; E) and an integer k
Output: Set S and Number of the Connected Components
S ← { }
while |S| < k do
    δ ← |V|; maxscore ← 0
    for each vertex v_i in V do
        update N(v_i) and d(v_i)
        C_ISC(v_i) ← 0; IS(v_i) ← 0
        if d(v_i) < δ then
            δ ← d(v_i)
        end if
    end for
    for each vertex v_i in V do
        if d(v_i) == δ then
            for each vertex v_j in N(i) do
                IS(v_j) ← IS(v_j) + 1
            end for
        end if
    end for
    for each vertex v_i in V do
        C_ISC(v_i) ← d(v_i) * IS(v_i)
        if C_ISC(v_i) > maxscore then
            maxscore ← C_ISC(v_i)
            v* ← v_i
        end if
    end for
    S ← S ∪ {v*}
    V ← V \ {v*}
end while
Calculate the number of the connected components in G.
```

centrality values of the nodes are calculated and the node with the highest centrality value is selected. This part of the algorithm can be modified according to the chosen centrality measure. After kth iteration, the algorithm returns S and the number of the connected components in the residual graph. For experimental simulations, the algorithm was implemented in C.

5.2. Dataset

There are two popular benchmark sets used in critical node studies in the literature. The first benchmark (*Benchmark1*) consists of 4 different synthetic graph types: Barabasi–Albert (BA), Forest Fire (FF), Erdos–Renyi (ER), Watts–Strogatz (WS). Each graph type has four different instances. BA graphs are scale-free networks. FF graphs are another type of scale-free network similar to the BA but denser than BA graphs. ER graphs are random graphs modeled with a Poisson distribution. Finally, WS graphs imitate the small-world networks. Although none of these graph types represents a real-life network exactly, real-life networks can be considered a mixture of these graph types. Hence, they constitute an interesting benchmark for critical nodes. Among these graphs, BA has the easiest instances, whereas WS has the most challenging instances for detecting critical nodes.

The second benchmark (*Benchmark2*) includes real-life networks. Circuit represents an electronic circuit [33]. USAir97 and EU_flights are the networks of flight connections between the airports in the European Union and between the major US airports [34] respectively. OClinks graph represents the interactions inside a social network [35]. The Facebook graph is created from the relations on the Facebook website [36]. e_Mail graph is a friendship network, consisting of 1133 students from ten different colleagues [37]. Jazz [38] is another social network consisting of Jazz musicians' connections. Finally, Ham1000 is a graph with hamiltonian cycles in TSPLIB [39]. Although it is not a real-life network, it helps diversify the benchmark set. All the graphs of *Benchmark1* and *Benchmark2* (except for Jazz and e-Mail) are available at the following address: http://www.di.unito.it/~aringhie/cnp.html.

Table 2 gives some basic features of the benchmark instances. The first three columns represent the instances' names, the number of the nodes, and the edges, respectively. The density (D) is given in the following column. The fifth column shows the number of articulation points (AP), and the sixth column represents the number of nodes having degree 1 ($|Deg_1|$). These two parameters can give some insight into the hardness of the instances. For example, a graph with many articulation points and nodes with degree 1 can easily be separated into disconnected components. Finally, in the last column, we provide the ratio of the number of nodes having degree 1 and the number of the nodes ($|Deg_1|/n$), indicating the actual difficulty of the instance.

In addition to *Benchmark1* and *Benchmark2*, we use random graphs with various densities. We created graphs of 100 nodes with different densities (0.05, 0.06, 0.07, 0.08, 0.09, 0.1).

Table 2
Main characteristics of benchmark instances.

Graph	n	m	D	AP	$\|Deg_1\|$	$\|Deg_1\|/n$
BA500	500	499	0.004	164	336	0.672
BA1000	1000	999	0.002	324	676	0.676
BA2500	2500	2499	0.0008	825	1675	0.67
BA5000	5000	4999	0.0004	1672	3328	0.666
FF250	250	514	0.017	83	57	0.228
FF500	500	828	0.007	195	160	0.32
FF1000	1000	1817	0.004	362	280	0.28
FF2000	2000	3413	0.002	725	552	0.276
ER235	235	350	0.013	48	39	0.166
ER466	466	700	0.006	84	69	0.149
ER941	941	1400	0.003	177	147	0.156
ER2344	2344	3500	0.001	419	396	0.169
WS250	250	1246	0.040	0	0	0
WS500	500	1496	0.012	0	0	0
WS1000	1000	4996	0.010	0	0	0
WS1500	1500	4498	0.004	0	0	0
Jazz	198	2742	0.281	5	5	0.025
Circuit	252	399	0.013	25	17	0.067
USAir97	332	2126	0.039	27	55	0.166
Ham1000	1000	1998	0.004	0	0	0
e_Mail	1133	5451	0.017	132	151	0.133
EU_flights	1191	31610	0.045	109	178	0.15
OClinks	1899	13838	0.008	220	388	0.204
facebook	4039	88234	0.011	11	75	0.019

Table 3
Results of the MaxNum problem on Benchmark1.

Graph	n	k	RND	CC	BC	DC	LC	ISC
BA500	500	50	99	310	311	313	312	312
BA1000	1000	75	95	586	590	590	589	589
BA2500	2500	100	96	1111	1100	1129	1097	1117
BA5000	5000	150	398	1940	1932	1999	1952	1982
FF250	250	50	26	84	87	86	86	79
FF500	500	110	54	204	211	207	212	205
FF1000	1000	150	63	276	307	298	302	295
FF2000	2000	200	109	426	446	430	461	443
ER235	235	50	19	48	49	62	61	60
ER466	466	80	14	52	64	81	92	98
ER941	941	140	50	74	94	150	169	183
ER2344	2344	200	52	66	71	154	261	294
WS250	250	70	1	3	3	1	5	15
WS500	500	125	1	13	28	1	30	48
WS1000	1000	200	1	1	7	1	16	46
WS1500	1500	265	1	19	44	1	61	95

5.3. Results

Table 3 gives the results of the centrality measures on *Benchmark1*. Moreover, we also reported the result of the randomly deleted k nodes in the third column.

Since all graph type has 4 different instances, we averaged the results of each graph type for a more precise analysis. With the highest $|Deg_1|/n$ values, BA graphs are the easiest instances for critical nodes problems. The average results for BA instances are 986.75, 983.25, 1007.75, 987.5, and 1000 for CC, BC, DC, LC, and ISC, respectively. Although DC seems to have the best results, all centrality measures could find comparable solutions. FF graphs are another easy graph

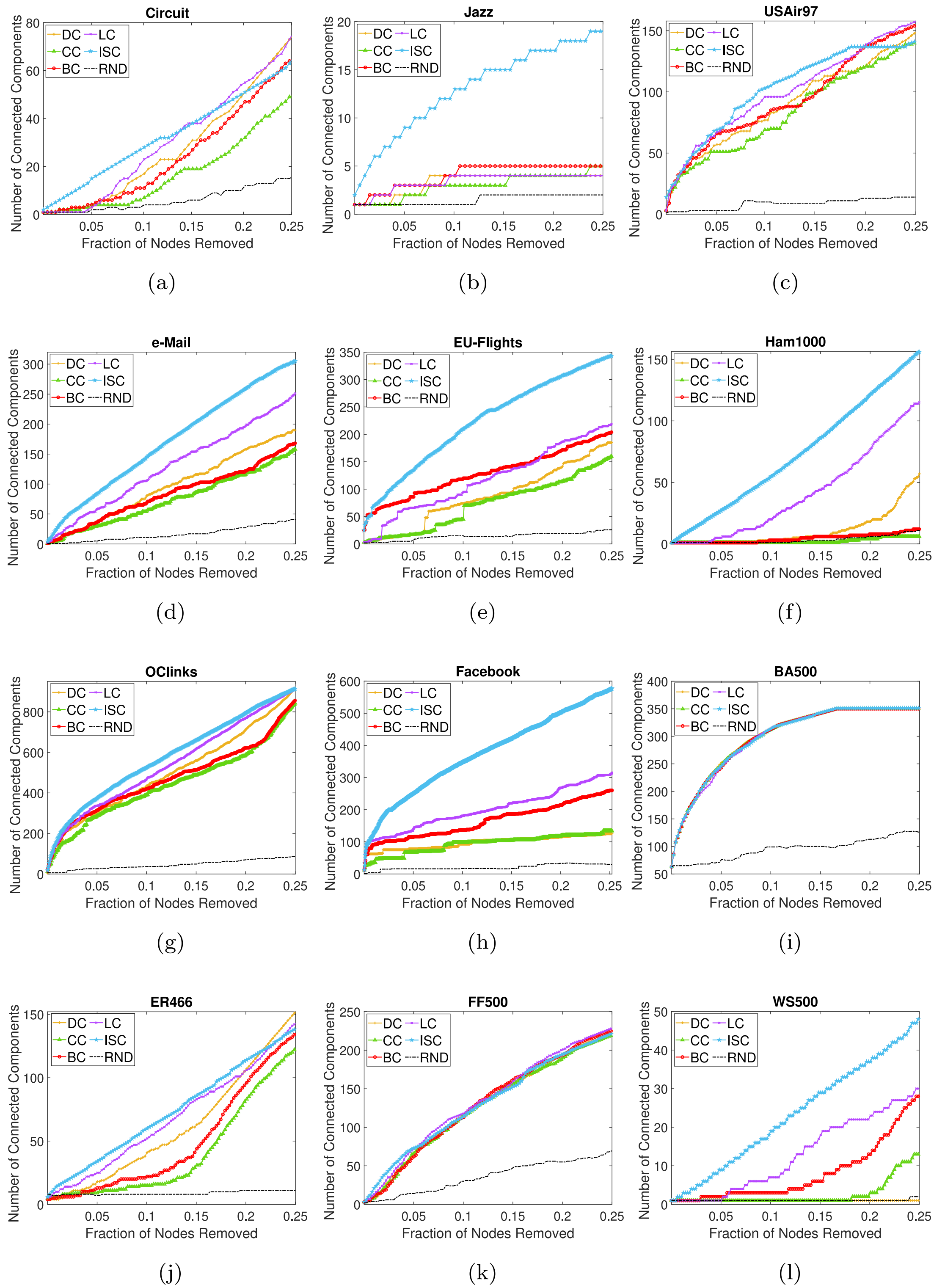


Fig. 5. Number of the connected components found by the centrality measures against the fraction of nodes removed on the benchmark instances.

family with a high number of AP and high $|Deg_1|/n$ values. Again, all centrality measures have similar results for FF graphs. The average results for FF instances are 247.5, 262.75, 255.25, 265.25, and 255.5 for CC, BC, DC, LC, and ISC, respectively.

Although ER graph family and FF graph family have close average n (937.5 for FF family and 996.5 for ER family) and k (127.5 for FF family and 117.5 for ER family) values, centrality measures find smaller solutions for ER graphs. However, ISC has the best average result of 158.75, which is two times better than BC (69.5) and almost three times better than CC (60). On the other hand, LC result (145.75) is competitive with ISC, and DC result (111.75) is worse than LC and ISC. WS graphs are the most challenging instances; hence DC failed to disconnect these instances. ISC outperforms its counterparts with an

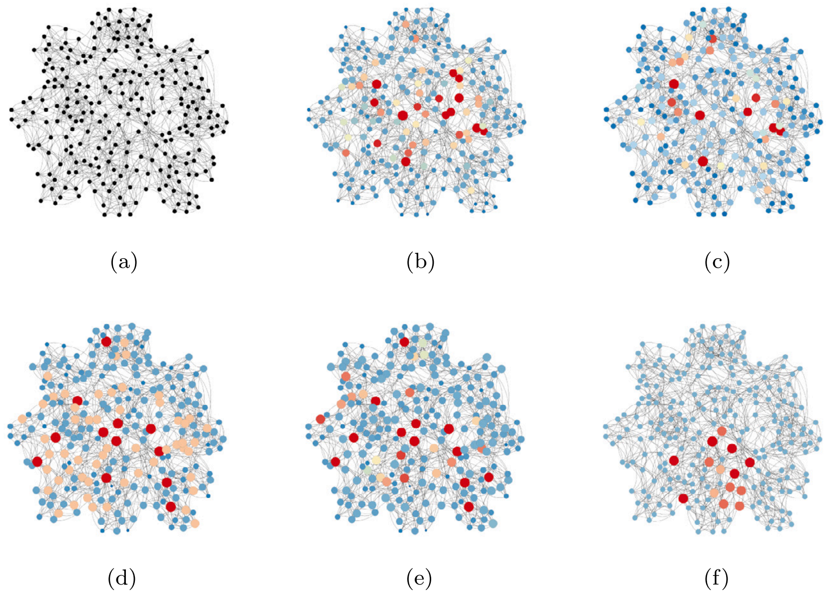


Fig. 6. The most central nodes of WS250 according to different centrality measures: (a) Original graph (b) Closeness Centrality (c) Betweenness Centrality (d) Degree Centrality (e) Leverage Centrality (f) Isolating Centrality.

Table 4
Results of the MaxNum problem on Benchmark2.

Graph	n	k	RND	CC	BC	DC	LC	ISC
Jazz	198	19	1	3	4	4	3	12
Circuit	252	25	3	6	11	16	21	27
USAir97	332	33	10	66	80	76	93	102
Ham1000	1000	100	1	1	3	3	20	54
email	1133	113	11	57	71	82	109	148
EU_flights	1191	119	15	45	118	74	90	208
OClinks	1899	189	36	389	418	426	463	523
facebook	4039	404	1	98	137	93	181	349

average result of 51. In comparison, these values are 8.75, 20.5, and 28 for CC, BC, and LC, respectively.

Table 4 gives the results of the centrality measures on *Benchmark2*. For real-life networks, k is set as 10 percent of n. It can be seen from the table that ISC can find better solutions for all instances. For Circuit, USAir97 and OClinks networks, LC is competitive with the ISC. However, ISC dominates all other measures for the remaining networks by finding at least two times better solutions.

For a more detailed analysis, we remove 25% of nodes from the real-life networks and report the number of connected components found by the centrality measures. Moreover, we add one instance from each graph type of *Benchmark1*. Fig. 5 shows the fractional of nodes removed against the number of connected components. For the easiest graphs, BA500 and FF500, all measures can find similar solutions for all k values. BC, DC, LC, and ISC have competitive performance for ER466, circuit, USAir97, and OClinks graphs. For the first 15 percent, ISC still has the best results; however, after 20 percent, BC, CC, and LC can find better results than ISC on these graphs. On the other hand, ISC dominates all other measures on the remaining networks. Especially for WS500, Jazz, and Facebook graphs, ISC finds almost 3–4 times better solutions than its counterparts.

Taking everything into account, ISC has the best performance and CC has the worst performance on benchmark instances. For small $|Deg_1|/n$ (relatively hard instances) and large $AvDeg$ values, ISC outperforms the other centrality measures for disconnecting the networks into several separate components. Since the primary aim of ISC is creating isolated nodes, at first glance, one can think that it needs several nodes with degree 1 to produce quality solutions. However, ISC dominates its counterparts even if the network has a small number of nodes with degree of 1 (Jazz, Facebook, Ham1000, and WS graphs). Thus, it can be concluded that ISC is a more efficient measure than the existing ones for separating complex networks.

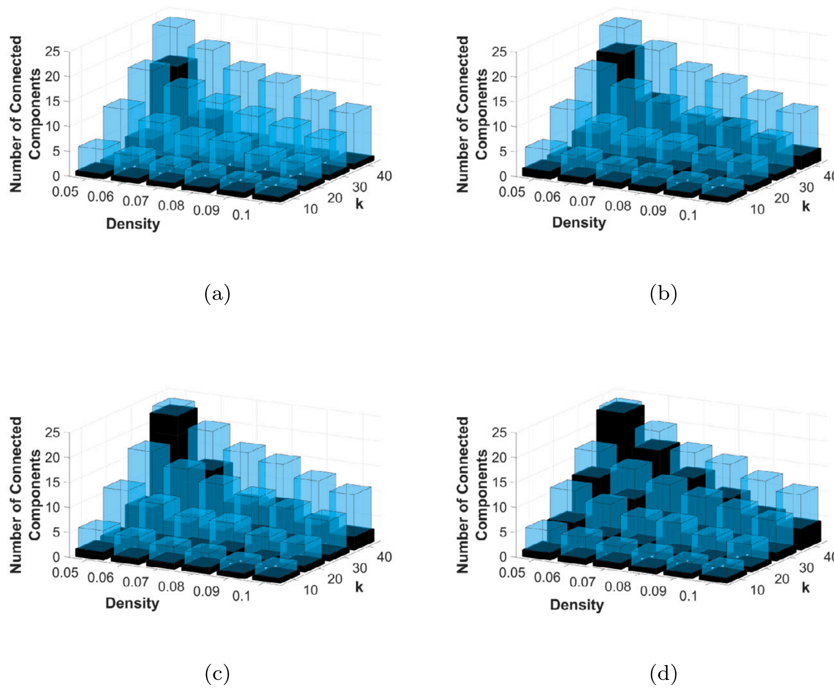
Furthermore, it is worth noting that ISC finds the best solutions in the literature (comparing with the results of [14]) for Ham1000 and WS graphs. It is interesting to visualize the central nodes according to different centrality measures for these graphs. Hence, we illustrated WS250 graphs in Fig. 6. In this figure, red nodes indicate the nodes with high centrality, and blue nodes indicate the nodes with low centrality. In addition, the size of nodes is scaled according to their centrality values.

Finally, we tested the performance of the centrality measures on random graphs (n = 100) with various densities. For each density, we generated five different graphs and took the average of the results. Fig. 7 presents the numbers of connected components and provides a comparison of ISC and existing measures. For k = 10, k = 20, and k = 30, CC, BC and DC could not produce quality solutions. Increasing the density reduces the size of the solutions and as the density increases, the difference between ISC and existing measures increases.

In summary, the results indicate that the degree-like centralities (DC, LC, and ISC) are more successful than the path-like centralities (CC and BC) for disconnecting networks into several independent components by deleting a small number of nodes. This implies that the network connectivity mainly relies on strategic high connected nodes rather than bridge nodes (cut nodes). Nevertheless, among the path-like centralities, BC is a better option for disconnecting networks. For scale-free networks (BA and FF graph family) and the networks with high $|Deg_1|/n$ values, which are relatively easy instances for critical node problems, BC, DC, LC, and ISC have comparable performance. However, for the hard instances (small number of AP and small $|Deg_1|/n$ ratio), ISC is dominant in terms of effectively separating the network. Especially for social and transportation networks, ISC finds three or four times better solutions than its counterparts. In this context, the proposed measure can be seen as much more proper for real-life applications.

6. Conclusion

This work compares and analysis the centrality measures for detecting critical nodes, which separate networks into several disconnecting



(a) (b)

(c) (d)

Fig. 7. Comparing the results of existing centrality measure with the proposed measure on random graphs: (a) Closeness Centrality vs Isolating Centrality (b) Betweenness Centrality vs Isolating Centrality (c) Degree Cen Isolating Centrality (d) Leverage Centrality vs Isolating Centrality.

components. Moreover, a new centrality measure that reveals important nodes for network connectedness is presented. The proposed centrality has low computational complexity, allowing it to be applied to large-scale networks. Computational experiments with several synthetic and real-world network instances demonstrate the effectiveness of the proposed measure. Besides, our results reveal interesting insights. In particular, our results support earlier observations from the literature that degree-like centralities are more relevant measures than the path-like centralities for disconnecting networks into several components. More importantly, our findings indicate that the proposed measure can detect the most vital critical nodes, which the traditional centrality measures cannot, in hard instances of critical node problems. Furthermore, the proposed method can be used for detecting the most critical node of a network if its removal disconnects the network by creating isolated nodes. On the other hand, if the most critical node separates the network into connected components, detecting this node depends on the structure of the nodes' neighborhood. The proposed method can still find the most critical node as long as one of its neighbors has the minimum degree of the network. Consequently, isolating centrality provides a new perspective for evaluating the importance of nodes, and it has several applications in real-world networks.

In future works, we plan to investigate the proposed measure on different classes of critical node detection problems. Also, comparing the centrality measures on unit disk graphs for the MaxNum can give insights into vulnerability analysis of distributed systems. Finally, extremely large networks with millions of nodes will be considered.

CRediT authorship contribution statement

Onur Ugurlu: Conceptualization, Methodology, Software, Formal analysis, Visualization, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research was supported by the TUBITAK (Scientific and Technological Research Council of Turkey) [Project number 121F092].

References

[1] Y.-C. Lai, A.E. Motter, T. Nishikawa, Attacks and cascades in complex networks, Complex Netw. (2004) 299–310, http://dx.doi.org/10.1007/978-3-540-44485-5_14.

[2] A.-L. Barabási, R. Albert, Emergence of scaling in random networks, Science 286 (5439) (1999) 509–512, http://dx.doi.org/10.1126/science.286.5439.509.

[3] X.F. Wang, G. Chen, Complex networks: small-world, scale-free and beyond, IEEE Circuits Syst. Mag. 3 (1) (2003) 6–20, http://dx.doi.org/10.1109/MCAS.2003.1228503.

[4] D.J. Watts, S.H. Strogatz, Collective dynamics of 'small-world'networks, Nature 393 (6684) (1998) 440–442, http://dx.doi.org/10.1038/30918.

[5] M. Curado, L. Tortosa, J.F. Vicent, G. Yeghikyan, Understanding mobility in Rome by means of a multiplex network with data, J. Comput. Sci. 51 (2021) 101305, http://dx.doi.org/10.1016/j.jocs.2021.101305.

[6] M. Lalou, M.A. Tahraoui, H. Kheddouci, The critical node detection problem in networks: A survey, Comp. Sci. Rev. 28 (2018) 92–117, http://dx.doi.org/10.1016/j.cosrev.2018.02.002.

[7] M. Kitsak, L.K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H.E. Stanley, H.A. Makse, Identification of influential spreaders in complex networks, Nat. Phys. 6 (11) (2010) 888–893, http://dx.doi.org/10.1038/nphys1746.

[8] G. Maji, S. Mandal, S. Sen, A systematic survey on influential spreaders identification in complex networks with a focus on K-shell based techniques, Expert Syst. Appl. 161 (2020) 113681, http://dx.doi.org/10.1016/j.eswa.2020.113681.

[9] G. Maji, A. Dutta, M.C. Malta, S. Sen, Identifying and ranking super spreaders in real world complex networks without influence overlap, Expert Syst. Appl. 179 (2021) 115061, http://dx.doi.org/10.1016/j.eswa.2021.115061.

[10] S. Shen, J.C. Smith, Polynomial-time algorithms for solving a class of critical node problems on trees and series-parallel graphs, Networks 60 (2) (2012) 103–119, http://dx.doi.org/10.1002/net.20464.

[11] S. Shen, J.C. Smith, R. Goli, Exact interdiction models and algorithms for disconnecting networks via node deletions, Discrete Optim. 9 (3) (2012) 172–188, http://dx.doi.org/10.1016/j.disopt.2012.07.001.

[12] A. Berger, A. Grigoriev, R. van der Zwaan, Complexity and approximability of the k-way vertex cut, Networks 63 (2) (2014) 170–178, http://dx.doi.org/10.1002/net.21534.

[13] A. Veremyev, O.A. Prokopyev, E.L. Pasiliao, An integer programming framework for critical elements detection in graphs, J. Comb. Optim. 28 (1) (2014) 233–273, http://dx.doi.org/10.1007/s10878-014-9730-4.

[14] R. Aringhieri, A. Grosso, P. Hosteins, R. Scatamacchia, A general evolutionary framework for different classes of critical node problems, Eng. Appl. Artif. Intell. 55 (2016) 128–145, http://dx.doi.org/10.1016/j.engappai.2016.06.010.

[15] R. Albert, H. Jeong, A.-L. Barabási, Error and attack tolerance of complex networks, Nature 406 (6794) (2000) 378–382, http://dx.doi.org/10.1038/35019019.

[16] P. Holme, B.J. Kim, C.N. Yoon, S.K. Han, Attack vulnerability of complex networks, Phys. Rev. E 65 (5) (2002) 056109, http://dx.doi.org/10.1103/PhysRevE.65.056109.

[17] A. Nasiruzzaman, H. Pota, M. Mahmud, Application of centrality measures of complex network framework in power grid, in: IECON 2011-37th Annual Conference of the IEEE Industrial Electronics Society, IEEE, 2011, pp. 4660–4665, http://dx.doi.org/10.1109/IECON.2011.6120079.

[18] R. Ghanbari, M. Jalili, X. Yu, Correlation of cascade failures and centrality measures in complex networks, Future Gener. Comput. Syst. 83 (2018) 390–400, http://dx.doi.org/10.1016/j.future.2017.09.007.

[19] A. Veremyev, O.A. Prokopyev, E.L. Pasiliao, Critical nodes for distance-based connectivity and related problems in graphs, Networks 66 (3) (2015) 170–195, http://dx.doi.org/10.1002/net.21622.

[20] C.P. Gillen, A. Veremyev, O.A. Prokopyev, E.L. Pasiliao, Robustness of solutions to critical node detection problems with imperfect data: a computational study, Optim. Methods Softw. 32 (2) (2017) 250–273, http://dx.doi.org/10.1080/10556788.2016.1214958.

[21] G. Nomikos, P. Pantazopoulos, M. Karaliopoulos, I. Stavrakakis, The multiple instances of node centrality and their implications on the vulnerability of ISP networks, 2013, arXiv preprint arXiv:1312.4707.

[22] Y. Shen, N.P. Nguyen, Y. Xuan, M.T. Thai, On the discovery of critical links and nodes for assessing network vulnerability, IEEE/ACM Trans. Netw. 21 (3) (2012) 963–973, http://dx.doi.org/10.1109/TNET.2012.2215882.

[23] D.T. Nguyen, Y. Shen, M.T. Thai, Detecting critical nodes in interdependent power networks for vulnerability assessment, IEEE Trans. Smart Grid 4 (1) (2013) 151–159, http://dx.doi.org/10.1109/TSG.2012.2229398.

[24] S. Iyer, T. Killingback, B. Sundaram, Z. Wang, Attack robustness and centrality of complex networks, PLoS One 8 (4) (2013) e59613, http://dx.doi.org/10.1371/journal.pone.0059613.

[25] M. Kaur, S. Singh, Ranking based comparative analysis of graph centrality measures to detect negative nodes in online social networks, J. Comput. Sci. 23 (2017) 91–108, http://dx.doi.org/10.1016/j.jocs.2017.10.018.

[26] G. Maji, Influential spreaders identification in complex networks with potential edge weight based k-shell degree neighborhood method, J. Comput. Sci. 39 (2020) 101055, http://dx.doi.org/10.1016/j.jocs.2019.101055.

[27] L. Meng, G. Xu, P. Yang, D. Tu, A novel potential edge weight method for identifying influential nodes in complex networks based on neighborhood and position, J. Comput. Sci. (2022) 101591, http://dx.doi.org/10.1016/j.jocs.2022.101591.

[28] K.E. Joyce, P.J. Laurienti, J.H. Burdette, S. Hayasaka, A new measure of centrality for brain networks, PLoS One 5 (8) (2010) e12200, http://dx.doi.org/10.1371/journal.pone.0012200.

[29] L.C. Freeman, A set of measures of centrality based on betweenness, Sociometry (1977) 35–41, http://dx.doi.org/10.2307/3033543.

[30] L.C. Freeman, Centrality in social networks conceptual clarification, Social Networks 1 (3) (1978) 215–239, http://dx.doi.org/10.1016/0378-8733(78)90021-7.

[31] S. Wasserman, K. Faust, et al., Social Network Analysis: Methods and Applications, Cambridge university press, 1994, http://dx.doi.org/10.1017/CBO9780511815478.

[32] Z. Wan, Y. Mahajan, B.W. Kang, T.J. Moore, J.-H. Cho, A survey on centrality metrics and their network resilience analysis, IEEE Access 9 (2021) 104773–104819, http://dx.doi.org/10.1109/ACCESS.2021.3094196.

[33] R. Milo, S. Itzkovitz, N. Kashtan, R. Levitt, S. Shen-Orr, I. Ayzenshtat, M. Sheffer, U. Alon, Superfamilies of evolved and designed networks, Science 303 (5663) (2004) 1538–1542, http://dx.doi.org/10.1126/science.1089167.

[34] USAir, Computational analysis of social and organizational systems, in: the USAir97 Network, 1997, Freely Available At URL http://www.casos.cs.cmu.edu/computational_tools/datasets/external/USAir97/index11.php.

[35] T. Opsahl, P. Panzarasa, Clustering in weighted networks, Social Networks 31 (2) (2009) 155–163, http://dx.doi.org/10.1016/j.socnet.2009.02.002.

[36] J. Leskovec, A. Krevl, SNAP Datasets: Stanford large network dataset collection, Ann Arbor, MI, USA, 2014, URL https://snap.stanford.edu/data/.

[37] P. Moradi, M.E. Shiri, A.A. Rad, A. Khadivi, M. Hasler, Automatic skill acquisition in reinforcement learning using graph centrality measures, Intell. Data Anal. 16 (1) (2012) 113–135, http://dx.doi.org/10.3233/IDA-2011-0513.

[38] P.M. Gleiser, L. Danon, Community structure in jazz, Adv. Complex Syst. 6 (04) (2003) 565–573, http://dx.doi.org/10.1142/S0219525903001067.

[39] G. Reinelt, TSPLIB — A traveling salesman problem library, ORSA J. Comput. 3 (4) (1991) 376–384, http://dx.doi.org/10.1287/ijoc.3.4.376.

Onur Ugurlu is an assistant professor of Computer Science at the Department of Fundamental Sciences, Izmir Bakircay University, Izmir, Turkey. He completed his undergraduate studies at Ege University with a major in computer science and a minor in mathematics. He has a M.S. in Computer Science from Ege University, Izmir, in 2013. He obtained his Ph.D. degree in Computer Science from Ege University University, Izmir, Turkey, in 2018. He was a visiting scholar in the Theoretical Computer Science Group at Northwestern University in the U.S. in 2017. He is mainly interested in design and analysis of algorithms, computational theory and graph theory.