Journal of King Saud University – Computer and Information Sciences 34 (2022) 4810–4820

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

Lexical sorting centrality to distinguish spreading abilities of nodes in complex networks under the Susceptible-Infectious-Recovered (SIR) model

Aybike Şimşek

Düzce University, Department of Computer Engineering, Faculty of Engineering, Düzce 81620, Turkey

ARTICLE INFO

Article history:
Received 31 March 2021
Revised 12 May 2021
Accepted 10 June 2021
Available online 24 June 2021

Keywords:
Complex networks
Social networks
Susceptible-Infectious-Recovered model
Centrality measure
Epidemic modeling
Super spreader

ABSTRACT

Epidemic modeling in complex networks is a hot research topic in recent years. The spreading of a virus (such as SARS-CoV-2) in a community, spreading computer viruses in communication networks, or spreading gossip on a social network is the subject of epidemic modeling. The Susceptible-Infectious-Recovered (SIR) is one of the most popular epidemic models. One crucial issue in epidemic modeling is the determination of the spreading ability of the nodes. Thus, for example, super spreaders can be detected in the early stages. However, the SIR is a stochastic model, and it needs heavy Monte-Carlo simulations. Hence, the researchers focused on combining several centrality measures to distinguish the spreading capabilities of nodes. In this study, we proposed a new method called Lexical Sorting Centrality (LSC), which combines multiple centrality measures. The LSC uses a sorting mechanism similar to lexical sorting to combine various centrality measures for ranking nodes. We conducted experiments on six datasets using SIR to evaluate the performance of LSC and compared LSC with degree centrality (DC), eigenvector centrality (EC), closeness centrality (CC), betweenness centrality (BC), and Gravitational Centrality (GC). Experimental results show that LSC distinguishes the spreading ability of nodes more accurately, more decisively, and faster.

1. Introduction

Complex networks are very convenient tools for modeling the real world. Many things in the real world connect to form a complex network (Barabási and Pósfai, 2016). Complex networks have applications in many areas, including biology (Hancock and Menche, 2020), social networks (Borgatti et al., 2009), ecology (Gao et al., Feb. 2016); cooperation dynamics in repeated games (Xu et al., 2019); (Liu, Dec. 2018), artificial intelligence (Xu et al., 2020) and more. In addition, there are many practical benefits to determining the spreading capacities of the nodes in complex networks under specific propagation models. Some of these include identification of nodes that can maximize the spread of information in a social network (Zareie and Sheikhahmadi, 2018) (Borgatti, 2006); detection of nodes that can minimize the spreading of a rumor (Yang et al., 2020); (Zhang et al., 2018), and revealing superspreaders that will play a role in the propagation of epidemics or computer viruses (Ali et al., 2020). For example, to maximize the spread of information, a small number of most influencer nodes on the network should be made active. This problem is an NP-Hard combinatorial optimization problem named Influence Maximization (IM) (Kempe et al., 2003). Therefore, greedy algorithms and optimization methods are used for IM (Kempe et al., 2003); (Zhang et al., 2017). However, both approaches need to distinguish the spreading abilities of the nodes under a given propagation model. For example, a greedy algorithm may select top-k the most influential nodes as the set of seed nodes (Kempe et al., 2003). In this context, determining the spreading abilities of nodes in complex networks has attracted the attention of researchers (Borgatti, 2006). In this way, the nodes can be ranked according to their spreading capabilities. For this, the nodes are selected as seed nodes one by one, and propagation is modeled. Propagation models such as Susceptible-Infectious-Recovered (SIR), however, require heavy Monte-Carlo simulations. Therefore, we need to less costly methods that can indirectly determine the spreading abilities of nodes under specific propagation models. On the other

E-mail address: aybikesimsek@duzce.edu.tr.
Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2021.06.010
1319-1578/

hand, graph-theoretical centrality measures are used to determine the importance of nodes on the network since a few decades. Researchers, who discovered that the centrality measures have correlation with the spreading abilities of the nodes under a specific propagation model, concentrated their research in this area. The main purpose here is to rank the nodes by means of a centrality measure, and these ranks are expected to be correlated with the ranks made according to the real spreading abilities of the nodes. Thus, nodes can be distinguished from each other without the need for heavy Monte-Carlo simulations. This is what the greedy algorithms or optimization methods we mentioned above need.

Many centrality measures have been developed and are still being designed for this purpose. The general problem with these measures is that they cannot provide a solution to the balance between speed and quality. Here, what we mean by quality is the correlation between the ranking created by the centrality measure and the ranking created according to the actual spreading abilities of the nodes. In this study, we developed a method we call Lexical Sorting Centrality (LSC). The LSC gives ranks to the nodes by using a combination of different basic centrality measures. Doing so does not use a predetermined closed-form solution, a factor that makes the LSC improvable. With this feature, LSC offers a new way to use different centrality measures in combination. According to the experimental results, the solution quality of LSC is better than the solution quality of the state-of-the-art centrality measures. At the same time, LSC is competitive in terms of speed.

2. Basic concept

Several centrality measures can be calculated for nodes in complex networks. Every measure indicates the importance (effect) of a node from its perspective. However, a centrality measure does not yield the same performance in all complex networks. Therefore, the idea has emerged recently of combining the centrality measures (Alshahrani et al., 2020; Li et al., 2018; Ma et al., 2016; Salavati et al., 2019; Şimşek and Kara, 2018; Yang et al., 2020; Şimşek et al., 2020). Almost all of the studies, which we will discuss in the next section, suggest combining the centrality measures with a closed formula. A merger in this way makes it difficult (if not impossible) to explain the logic of creating the formula when there is no natural phenomenon that inspired its creation. Creating close formulas also prevents the proposed methods from being improvable. This study suggests a way to use different graph-theoretical centrality measures together without using a closed formula. We termed this method Lexical Sorting Centrality (LSC). Using multiple centrality measures, LSC uses order-theoretical methods to sort nodes like reverse lexical (alphabetical) order. For example, LSC calculates three different centrality measures (C_1, C_2, and C_3) for a six-node graph. For convenience, we choose the precision of the centrality measures as one decimal place. Each centrality measure here is like a letter, and the "node-centrality measure array" can be thought of as a word. For example, in Fig. 1, the letters formed in node 0 are 0.2, 0.8, and 0.3. Fig. 1(a) shows the initial state. Here, the nodes are sorted according to their numbers. In situation Fig. 1(b), the nodes are sorted from large to small according to C_1 measure. Here, the C_1 values of nodes 4 and 5 and nodes 0 and 2 are the same. Therefore, it is not possible to say which of these nodes is more important than the other by looking only at the C_1 measure. Afterward, only the nodes with the same C_1 values (i.e., nodes 4 and 5 and nodes 0 and 2) are sorted from large to small according to C_2 values to arrive at the situation in Fig. 1(c). The C_1 and C_2 values of nodes 0 and 2 are the same. Finally, the nodes with the same C_1 and C_2 values (nodes 0 and 2) are ranked according to the C_3 value to reach the final state in Fig. 1(d). The final arrangement of the nodes is 5–4-1–2-0–3. If we consider each centrality measure as a letter here, this order is precisely that of a reverse lexical. If all centrality measures calculated for any two nodes are the same, the order of those nodes is not changed.

3. Motivation

Centrality measures represent the importance of nodes. Each centrality measure does this with its perspective, and the importance of a node depends on the given context (Saxena and Iyengar, 2020). Among the most basic centrality measures are the degree, closeness, betweenness, eigenvector, and Katz centrality measures. For example, the local influence of the high degree node can be high. However, it may not be effective globally. Likewise, a node with a high betweenness node can provide information flow between different communities. However, it may not be effective locally (Saxena and Iyengar, 2020). On the other hand, the spreading ability of a node depends on various parameters such as the number of neighbors, how influential their neighbors are, and their location on the network (Cherifi et al., 2019; Yan et al., 2020). If multiple centrality measures are used together, different perspectives are combined, and more accurate information about the spreading abilities of nodes can be obtained. Therefore, researchers are looking for a way to use various centrality measures together.

Many studies in the literature are based on combining multiple measures with a closed formula and making the formula adjustable by assigning a coefficient to each sub-measure. The first problem here is to determine the values of the coefficients. Most of the studies empirically give the coefficients a value of "1". Another problem is the arithmetic operations. It is difficult to explain the purpose of multiplying or summing the two sub-measures with each other. On the other hand, each centrality measure has different correlations with the spreading abilities of nodes (Şimşek et al., 2020). For this reason, when deciding on the rank of a node, a highly correlated centrality measure should have a higher effect on this decision. LSC generates a new centrality measure by combining multiple centrality measures. In doing so, it doesn't use a closed formula. LSC assigns ranks to the nodes in a way similar to alphabetical order by using multiple centrality measures.

4. Main contributions

The main contributions of this paper are as follows:

1. It introduces a new method to combine multiple centrality measures in a fast and straightforward manner. Thus, different centrality measures can easily be used together, even for large networks.
2. LSC distinguishes the spreading abilities of nodes under the SIR propagation model better than recent and well-known centrality measures in the literature.

The rest of paper is organized as follows. Section 5 summarizes the related works in the literature. Section 6 gives the basic definitions and terminology used throughout this paper and then introduces LSC. Section 7 describes the details of the experiments and gives the experimental results. Finally, Section 8 presents the discussion and conclusions.

5. Related work

In this section, we reviewed the studies in the literature that combine different centrality measures. The following studies may be examined for Influence Maximization and Influential Spreader Detection: (Maji et al., 2020; Azaouzi et al., 2021).

$Node$	C_1	C_2	C_3
0	0.2	0.8	0.3
1	0.5	0.3	0.5
2	0.2	0.8	0.4
3	0.1	0.4	0.8
4	0.7	0.5	0.1
5	0.7	0.6	0.7

(a)

$Node$	C_1	C_2	C_3
4	**0.7**	0.5	0.1
5	**0.7**	0.6	0.7
1	**0.5**	0.3	0.5
0	**0.2**	0.8	0.3
2	**0.2**	0.8	0.4
3	**0.1**	0.4	0.8

(b)

$Node$	C_1	C_2	C_3
5	**0.7**	**0.6**	0.7
4	**0.7**	**0.5**	0.1
1	**0.5**	**0.3**	0.5
0	**0.2**	**0.8**	0.3
2	**0.2**	**0.8**	0.4
3	**0.1**	**0.4**	0.8

(c)

$Node$	C_1	C_2	C_3
5	**0.7**	**0.6**	**0.7**
4	**0.7**	**0.5**	**0.1**
1	**0.5**	**0.3**	**0.5**
2	**0.2**	**0.8**	**0.4**
0	**0.2**	**0.8**	**0.3**
3	**0.1**	**0.4**	**0.8**

(d)

Fig. 1. Sorting nodes like alphabetical order using multiple centrality measures.

Centrality measures such as degree (Newman, 2018), closeness (Sabidussi, 1966), betweenness (Freeman, 1977), eigenvector (Bonacich, Mar. 1987), Katz (Katz, 1953), and PageRank (Page et al., 1999) are well-known centrality measures used to determine the importance of nodes in complex networks. This section will discuss the more recent centrality measures and studies that use more than one measure together.

There are two general trends in the literature for using multiple centrality measures together: considering the network structure and using it for influential node detection, and combining different centrality measures into a unique centrality measure. Most of the studies considering the network structure have analyzed the community structures in the network. They have used basic centrality measures such as degree, closeness, betweenness, etc., together to determine the inter-community and intra-community influence capabilities of nodes (Ghalmane et al., 2019a, 2019b; Zhao et al., 2016, 2015). For more detailed classification of node ranking approaches for influential node detection, (Cherifi et al., 2019) could be examined. In this study, our primary focus is the combining of different centrality measures. Therefore, we have reviewed this type of studies in more detail.

Andrade and Rêgo developed a centrality measure called p-means (Andrade and Rêgo, 2019). The p-means combines degree, closeness, harmonic, and eccentricity centrality measures parametrically into one formula. By changing the p-means' formula's coefficient, p-means behave like one of the measures (or a combination of them). Zhao et al. developed the two sub-measure of self-importance and global importance and, by multiplying them, proposed a composite centrality measure demonstrating the importance of the nodes (Zhao et al., 2020). Using a coefficient for self-importance and global importance in their formula, they ensured that the balance could be changed towards one of these two sub-measures. They compared their proposed method with well-known centrality measures such as degree, closeness, betweenness, and PageRank on six real networks under the SIR propagation model and achieved competitive results. Alshahrani et al. developed two algorithms called MinCDegKatz d-hops and MaxCDegKatz d-hops (Alshahrani et al., 2020). These algorithms combine degree and Katz centrality measures, taking into account the local and general strength of the nodes. They compared their proposed algorithms to various competing algorithms under Independent Cascade (IC) and Linear Threshold (LT) propagation models on four real networks. Yang et al. developed a new centrality measure called DCC (Yang et al., 2020), which combines degree and clustering coefficient measure in a closed formula. Just like in the measure proposed by Zhao et al. (Zhao et al., 2020), by using a coefficient for the sub-components in their formula, they ensured that the balance could be changed towards one of these two sub- measures. They compared their proposed algorithms to different competing algorithms under the Susceptible – Infected (SI) propagation model on four real networks. Şimşek and Meyerhenke developed many new centrality measures using well-known centrality measures such as degree, closeness, and eigenvector (Şimşek et al., 2020). Their proposed combining method is to multiply the centrality measures with different coefficients and subject the results to some arithmetic operations. They compared their proposed new measures with competing algorithms under the IC propagation model on 50 real networks. Zhao et al. demonstrated that various centrality measures could be combined using Evidence Theory (Zhao et al., 2020). Their method is called Evidence Theory Centrality (ETC). They used DC, BC, and CC centrality measures. They have made experiments to demonstrate ETC's performance on 6 real network datasets under the SI propagation model. Xiao-Li et al. developed a new centrality measure by combining the topological characteristics of the nodes, their positions, propagation characteristics and the characteristics of their neighbors (Yan et al., 2020). They have done experiments on six real network datasets under the SIR propagation model. The proposed centrality measure has outperformed the basic centrality measures. Keng et al. proposed a new way of combining centrality measures (Keng et al., 2020). The method creates a convex combination of different centrality measures. First, they analyzed the correlation of centrality measures with each other. According to their analysis, they proposed coefficients for each centrality measure. They showed that the sum of the centrality measures multiplied by the coefficients could also be a new centrality measure.

In summary, many of the studies in the literature form closed formulas to combine several centrality measures and make the formula adjustable by assigning a coefficient to each sub-measure

(e.g.,$factor_1 \cdot CentralityMeasure_1 + factor_2 \cdot CentralityMeasure_2$). The first problem here is to determine the values of the coefficients. Most of the studies empirically give the coefficients a value of "1". Another problem is the arithmetic operations in the closed formulas. It is difficult to explain the purpose of multiplying or summing two centrality measures.

6. Distinguishing spreading abilities of nodes using LSC

6.1. Preliminary information

As mentioned in the introduction, LSC sorts nodes similar to reverse lexical order using multiple centrality measures. In this study, we selected the degree, eigenvector, and closeness measures. We also used Susceptible-Infectious-Recovered (SIR) as the propagation model. First, let us discuss these measures and the SIR model from a graph-theoretical perspective.

Let $G = (V, E)$ be an undirected unweighted graph (network). Here, V is the set of nodes (vertices), and E is the set of edges (links).

Definition 1 (*(Degree Centrality):*). *Degree centrality* (*DC*) *is calculated by dividing the degree of a node by the total number of nodes in the graph minus* 1.

$$DC(i) = \frac{degree(i)}{|V| - 1} \tag{1}$$

Here, $i \in V$.

Definition 2 (*(Eigenvector Centrality):*). *The eigenvector centrality* (*EC*) *of a node is calculated by dividing the sum of the EC values of neighbors by a constant.*

Let $A = (a_{ij})$ be the adjacency matrix of G; if i and j are neighbors, $a_{ij} = 1$; otherwise, $a_{ij} = 0$.

$$EC(i) = \frac{1}{\lambda} \sum_{j \in V} a_{ij} EC(j) \tag{2}$$

Here, λ is a constant.

Definition 3 (*(Closeness Centrality):*). *The closeness centrality* (*CC*) *of a node is calculated using the average distance* (*over the shortest paths*) *of the node to all other nodes.*

$$CC(i) = \frac{|V|}{\sum_{j \in V - \{i\}} sp(j, i)} \tag{3}$$

Here, $sp(\cdot)$ is the shortest path between nodes i and j.

Definition 4 (*(Susceptible-Infectious-Recovered Model):*). *Susceptible-Infectious-Recovered* (*SIR*) *is a well-known epidemic model. Although it is a population-based model, it is beginning to be applied to network structures* (Ullman et al., 1974) *because of its popularity in recent years. The SIR model is defined by two parameters:* β; *the rate at which the sensitive nodes are infected by their already infected neighbors*; *and* γ; *the recovery rate of infected nodes. Initially, all nodes are susceptible. One or more nodes on the network are first infected* (*the nodes that bring the disease to the network*). *These are often referred to as seed nodes. Starting from these nodes, in step t* (*time frame*), *the infection spreads over the network and becomes an epidemic, or otherwise, it is finished before it becomes an epidemic. When the disease becomes an epidemic, the number of nodes infected first and their location* (*significance*) *on the network depends on the network topology* (*i.e., density*), β *and* γ.

6.2. LSC

The LSC uses order-theoretical methods to sort nodes similar to reverse lexical order, using multiple centrality measures, as explained in the Introduction section. In this study, we used DC, EC, and CC measures. First, DC, EC, and CC are calculated for all nodes. Thus, a node and its calculated measures can be written as $4 - tuple$:

$$T = (node\ DC\ EC\ CC) \tag{4}$$

After that, LSC creates a matrix as shown in Equation (5). We named this the Ranking Matrix (RM).

$$\mathbf{RM} = \begin{bmatrix} node_0 & DC_0 & EC_0 & CC_0 \\ node_1 & DC_1 & EC_1 & CC_1 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ node_{n-1} & DC_{n-1} & EC_{n-1} & CC_{n-1} \end{bmatrix}_{n \times 4} \tag{5}$$

Here, $n = |V|$.

Then, LSC performs reverse lexical sorting as follows:

First, it sorts the RM by *DC* column. Thus, the rows with the same *DC* (if any) will come one after another. Then, LSC sorts only these rows according to *EC*. Rows with the same *EC* (if any) will come one after another. Lastly, LSC sorts these rows among themselves according to *CC*. Thus, LSC completes the ranking. If we consider each centrality measure as a letter, each node can be thought of as a three-letter word. Therefore, the method is called Lexical Sorting. The LSC can be extended for more than one desired centrality measure.

Formally, X be a sequence $X_1, X_2, \cdots, X_n$, where X_i is a $k - tuple$. LSC creates a sequence $Y_1, Y_2, \cdots, Y_n$, which is a permutation of X, where $Y_i \geq Y_{i+1}$ for $1 \leq i \leq n$. For this, $k - tuples$ should be sorted into reverse lexicographical order. Let $>$ be a linear order on set S. If we extend $>$ to tuples from S; it will be lexicographical order if $(s_1, s_2, \cdots, s_n) > (t_1, t_2, \cdots, t_n)$ when there exists an $j \in \mathbb{Z}^+ - \{1\}$, and such that $s_j > t_j$ and for all $i < j$, $s_i = t_i$ (Ullman et al., 1974). In (4), the first element of the tuple is node number. In order not to use the node number in the sorting process, j should not be 1.

The generalized algorithm of LSC is shown in Algorithm 1. First, LSC computes the centralities for all nodes. Then it calls the Sort function. In practice, starting the process of sorting the RM from the last column makes it easy. Therefore, we set up a descending loop in the Sort function's main block. Another sorting algorithm sorts the RM according to each column. Here we have chosen the Bubble Sort algorithm. The point to note is that the sorting algorithm used must be a stable sort algorithm. Otherwise, the RM will not be sorted correctly. This algorithm can be considered a Radix sort implementation (in descending order) to tuples that values are real numbers.

```
Algorithm 1. LSC's algorithm
LSC(G: Graph, n: |V|)
Begin
RMnxr = ComputeCentralities(G: Graph, n: |V|)
Sort(RMnxr: Ranking Matrix, n: |V|, c: r-1) //r-1 is the number of //centrality measures
Sort(RM: Ranking Matrix, n: |V|, c)
Begin
for x = c to 1 do
  BubbleSort(RM, n, x)
End

BubbleSort(RM: Ranking Matrix, n: |V|, x: concerned column number in RM)
Begin
for i = 0 to n-1 do
  for j = 0 to n-i-2 do
    if RM[j + 1][x] > RM[j][x] then
      Swap(RM[j + 1],RM[j],c) // a function that swaps the rows.
End

Swap(A,B: two rows from RM, c: number of centrality measures)
Begin
for i = 0 to c-1 do
  temp = RM[j + 1][i]
  RM[j + 1][i] = RM[j][i]
  RM[j][i] = temp
End

ComputeCentralities(G: Graph, n: |V|)
Begin
Create an empty RMnxr matrix
for each node ∈ V do
  dc = calculate DC by using (1)
  ec = calculate EC by using (2)
  cc = calculate CC by using (3)
  add (node, dc, ec, cc) tuple to RMnxr as a new row
return RMnxr
End
```

LSC calls the Sort function one time, and the Sort function calls the Bubble Sort function one time for each column in the RM; Bubble Sort function calls Swap function if necessary. Thus, the time complexity of LSC is $O\left(c^2|V|^2\right)$; briefly $O\left(|V|^2\right)$. Here, c is the number of centrality measures used by LSC.

The time complexity of the centrality measures used by the LSC is another issue. Let's make calculations for DC, EC, and CC that we employ in this study. The time complexity of DC is $O\left(|V|^2\right)$. Time complexities of EC and CC are $O(|V|+|E|)$ and $O(|V||E|)$, respectively (Wandelt et al., 2020). Thus, the time complexity for LSC with DC, EC, and CC is $O\left(|V|^2+|V|^2+|V||E|+|V|+|E|\right)$. If we consider only leading terms, the time complexity will be $O\left(|V|^2+|V||E|\right)$.

The order of use of centrality measures is another critical issue. The centrality measure at the forefront in the **RM** has priority, just like starting from the first letters of the words in alphabetical order. The DC is an essential local centrality measure. If all nodes have the same probability of infection (β), a high-degree node has the chance to infect more nodes at once. The degree may be a useful first approximation of node centrality on unweighted networks (Oldham et al., 2019). We used the EC as a secondary measure as it incorporates the powers of the neighbors into the account. If the probability of a node infecting a neighbor is β, the probability of infecting its neighbor's neighbor is β^2 (usually, β is chosen as much smaller than 1). Since the probability will be $\beta \gg \beta^2$, we gave DC priority over EC. Finally, we used CC because it provides information about the location of the node on the network. In summary, DC represents the local power of a node, EC represents the power of its immediate neighbors, and CC indicates the strength of its location on the network. Also, Using highly correlated centrality measures together is practically redundant in most real-world scenarios. DC, EC, and CC are not highly correlated with each other (Oldham et al., 2019). Because of all this, we used DC, EC, and CC together and in this order.

Another critical point is the decimal precision of the measure. The selected decimal precision may change the order. Let us consider the nodes in Equation (6). Since the DC of $node_0$ is greater than the DC of $node_1$, the LSC positions $node_0$ first and $node_1$ second. However, if we take the precision of the digits to only two places after the decimal point, the DC of $node_0$ and $node_1$ will be the same, as seen in Equation (7). In this case, the LSC positions $node_1$ first and $node_0$ second because LSC sorts the nodes by EC (and since here, the EC of the two nodes are different). The order will be as in Equation (7). In this study, we choose the decimal precision to five places. Thus, we preserved the ranking of the prioritized measures as much as possible.

$$\mathbf{RM} = \begin{bmatrix} node_0 & 0.76525 & 0.05963 & 0.15423 \\ node_1 & 0.76234 & 0.06421 & 0.24563 \end{bmatrix} \tag{6}$$

$$\mathbf{RM} = \begin{bmatrix} node_1 & 0.76 & 0.06 & 0.24 \\ node_0 & 0.76 & 0.05 & 0.15 \end{bmatrix} \tag{7}$$

7. Experiments

To evaluate the performance of the LSC, we selected five competing centrality measures and conducted experiments on 1 synthetic and 5 real-world network datasets. First, let us discuss the competitive measures and datasets.

7.1. Centrality measures

DC (*Degree Centrality*) (Newman, 2018): The DC is calculated by dividing the degree of the node by the total number of nodes in the graph minus one. The DC, one of the main centrality measures, is a local measure.

EC (*Eigenvector Centrality*) (Bonacich, Mar. 1987): The EC is obtained by dividing the sum of EC values of the nodes to which the EC node is directly connected (i.e., neighbors) by one constant. Each node is initially assigned a default EC.

CC (*Closeness Centrality*) (Sabidussi, 1966): The CC is calculated using the average distance (over the shortest paths) of the CC node to all other nodes. The CC of the node that is closest to all other nodes is the highest.

BC (*Betweenness Centrality*) (Freeman, 1977): The BC provides information on the number of times a node can intersect the shortest paths among all other node pairs.

GC (*Gravitational Centrality*) (Ma et al., 2016): The GC is a recent centrality measure whose development was inspired by Newton's gravitational formula. In the GC, the k-shell values of the nodes replace the mass in Newton's formula. It uses the length of the shortest path between nodes rather than the distance between masses. Its formula is as follows:

$$GC_i = \frac{ks_i \times ks_j}{\sum_{j \in N} sp(j,i)} \tag{8}$$

Here, $sp(\cdot)$ is the shortest path between nodes i and j; N is the set of 3-hop neighbors of node i. The GC was chosen as a competitor because it can use various centrality measures (e.g., k-shell).

7.2. Datasets

We used one synthetic and five real-world public networks for the experimental studies. Experiments involving much more and larger data sets will provide more precise results. However, a very high processing capacity is required for modeling the propagation over large data sets. For this reason, most of the similar studies in the literature have used fewer and smaller data sets. In addition, the data sets we use are frequently used data sets in the literature. Therefore, the experiments give acceptable results of the performance of LSC and the other competitor centrality measures. The properties of the networks are shown in Table 1.

Barabasi-Albert: This synthetically-created scale-free network includes 1000 nodes and 9900 edges (Barabási and Albert, 1999).

Karate: This network consists of 34 nodes and 78 edges. The nodes denote members of the club, and the edges represent the friendship between members (Zachary, 1977). This dataset is taken from http://konect.uni-koblenz.de/publications.

Email-Enron: This Email network consists of 143 nodes and 623 edges (Rossi and Ahmed, 2015). This dataset is taken from http://networkrepository.com.

Email-Univ: This network consists of 1133 nodes and 5452 edges (Guimerà et al., 2003). This dataset is taken from http://konect.uni-koblenz.de/publications.

CS-PhD: This network consists of 1882 nodes and 1740 edges (De Nooy et al., 2011). This dataset is taken from http://networkrepository.com.

Ia-reality: This network consists of 6809 nodes and 7680 edges (Eagle and (Sandy) Pentland, 2006). This dataset is taken from http://networkrepository.com.

7.3. Evaluation of the ranking performance of centrality measures

First, we evaluated the ranking performance of the centrality measures under the SIR model. For this, we used the Kendall *tau* correlation coefficient commonly used in the literature (Kendall, 1938). Let (a_i, b_i) and (a_j, b_j) be tuples of joint A and B ranking lists. If $a_i > a_j$ and $b_i > b_j$ or $a_i < a_j$ and $b_i < b_j$, then the tuples are concordant. If $a_i > a_j$ and $b_i < b_j$ or $a_i < a_j$ and $b_i > b_j$, then the tuples are discordant. If $a_i = a_j$ or $b_i = b_j$, then the tuples are neither concordant nor discordant. Finally, *tau* is defined as in Equation (9).

$$tau = \frac{N_c - N_d}{0.5N(N-1)} \tag{9}$$

Here, N_c is the number of concordant pairs, N_d is the number of discordant pairs, and$N$ is the number of all combinations. Positive *tau* values indicate a positive correlation, and negative *tau* values indicate a negative correlation. The ranking performances of LSC and other centrality measures under the SIR model are shown in Fig. 2. In the simulations, the infection rate for small or less dense networks was $\beta = 0.1$, and for more extensive or more dense networks, $\beta = 0.01$. The recovery rate for all simulations was taken as $\gamma = 1$. Since it is difficult to differentiate the spreading abilities of nodes for large β values, the β value was chosen according to the scale of the network (Sheng, 2020). The simulations continued until there were no infected nodes on the network. The SIR score of a node is the total number of recovered nodes at the end of the simulation when that node is selected as the sole seed. All SIR simulations in this study were repeated 1000 times, and their averages were used. NetworkX was used for the network operations (Hagberg et al., 2008).

The LSC performed the best in three datasets. Its performance in other datasets is very close to the competitors' performance. Thus, the LSC was shown to yield excellent and stable results.

In addition, the graphics of the ranking lists created by the centrality measures vs. the SIR scores are shown in Fig. 3. A node with a lower index value is expected to have a higher SIR score. Therefore, a decrease in the SIR score as the index increases indicates the success of the centrality measure. In Fig. 3, the graphics created by the LSC are, for the most part smoother compared to those produced by the other measures.

Additionally, when evaluating the performances of the centrality measures, the spreading abilities of the nodes determined as $top - x$ by the centrality measures were assessed. To this purpose, firstly, in Table 2, we have shown the ranking lists of the centrality measures and the number of matching nodes in the top 5% of the ranking list of the SIR simulation. For example, the table shows that 5% of the number of nodes of the Barabasi-Albert network is 50. According to the SIR scores, 42 of the nodes that fall in the top-50 when ranked from large to small were also in the top-50 of the LSC's ranking list. LSC yielded the best results on all networks for both.

Finally, we examined how the SIR scores (created according to the core of the $top - 10$ nodes determined by the centrality measures) changed over time. Trials were conducted for different β values; however, only the result of $\beta = 0.05$ is given because other results were similar, and we wanted to save space. The trials yielded $t = 25$. The trends are shown in Fig. 4. The rapid increase of the curves in the graphs indicates that the nodes selected by the relevant centrality measure maximized the spread in a short time. According to the graphics, the nodes chosen by the LSC increased the spread to the highest level in a short time. In addition, it can be seen in the enlarged inserts in the graphics that in four datasets, the nodes selected by the LSC have a higher spreading capacity.

7.4. Evaluation of the influence of the decimal precision in LSC calculation

We mentioned under 3.2 LSC that the decimal precision of centrality measure could change the rank that LSC will produce. Another critical is the decimal precision of the centrality measures used by LSC. For this purpose, we have rounded the values of centrality measures used in the LSC by selecting different decimal

Table 1
Network dataset features.

Dataset	$\|V\|$	$\|E\|$	$\langle K \rangle$	K_{max}	Density
Barabasi-Albert	1000	9900	19.8	198	0.0198198
Karate	34	78	4.588	17	0.1390374
Email-Enron	143	623	8	42	0.0613612
Email-Univ	1133	5452	9.62	71	0.0085002
CS-PhD	1882	1740	1.849	46	0.0009830
Ia-reality	6809	7680	2.256	261	0.0009830

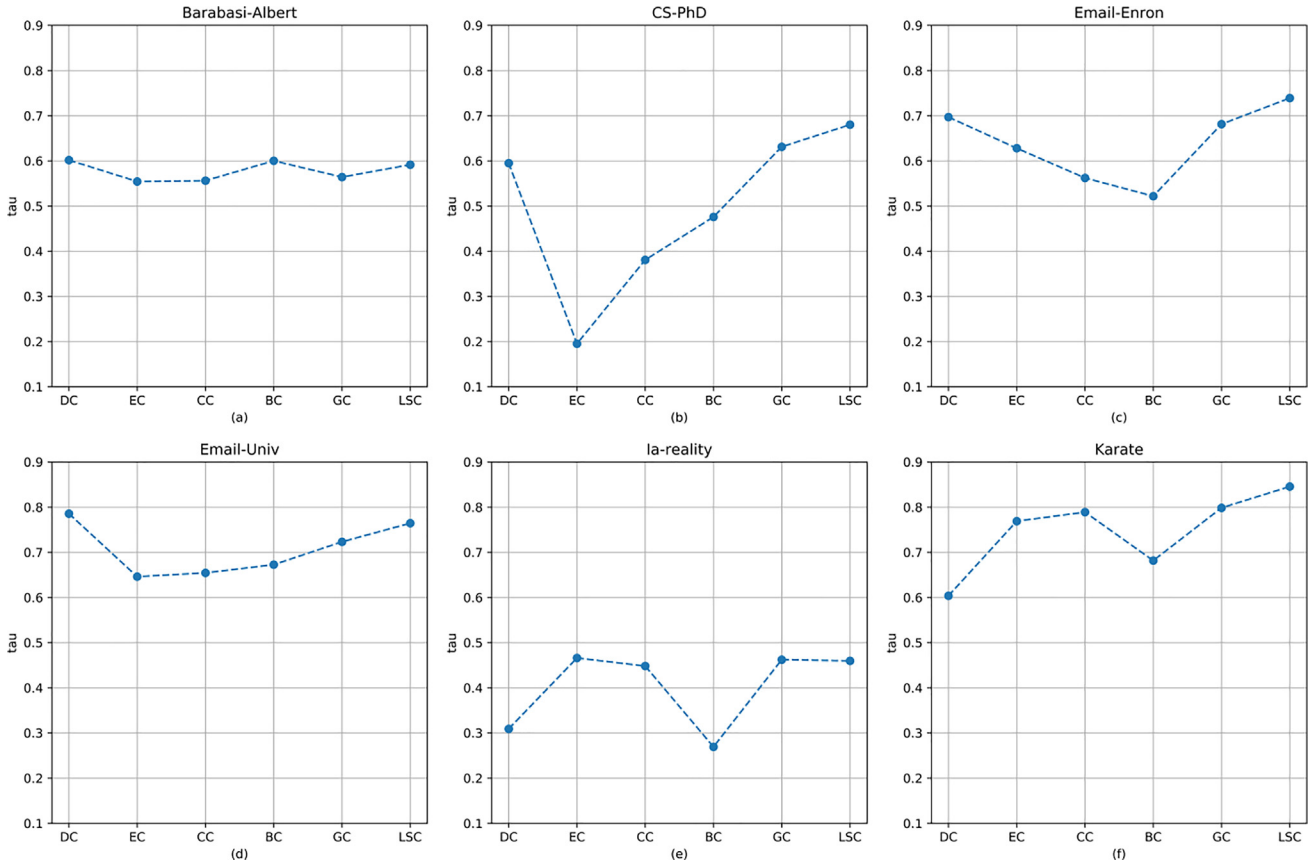

Fig. 2. Kendall ***tau*** correlation coefficient values of different centrality measures. Infection rate: $\beta = 0.1$ for (b), (d), and (f); $\beta = 0.01$ for (a), (c), and (e). Recovery rate: $\gamma = 1$ for all experiments.

precision from 1 to 6. We calculated the Kendall tau correlation coefficient for the LSC ranks created with each decimal precision. The ranking performances of LSC for different decimal precisions are shown in Fig. 5.

In general, better results have been obtained for values of decimal precision greater than two. Again, very close tau values were obtained for decimal precision values greater than two. Choosing a higher decimal precision preserves the ranks made by the preceding centrality measure does. Thus, if the order of centrality measures is determined well, LSC gives better results with high decimal precision. In addition, it is worth noting that the datasets we use in this study have at most several thousand nodes and edges. In small networks, the centrality measures of the nodes differ from each other with less decimal precision. To observe this, we produced 2 Erdös-Renyi graphs with 1000 and 10,000 nodes. The density of both graphs is the same. While the DC of a node in 1000 node graph starts to repeat after the 3rd digit; The DC of a node in the 10,000 node graph starts repeating after the 4th digit (such as 0.100100100100 and 0.099409940994, respectively). From here, we can conclude the following: As the number of nodes in the network increases, if we want to preserve the ranks made by the preceding centrality measures, we should increase the decimal precision of the centrality measures. Conducting experiments for larger real network datasets and making deeper analyses on decimal precision can be considered future work.

7.5. Combining different numbers of centrality measures

In the experimental studies, we conducted experiments where LSC combines three centrality measures (DC, EC, and CC). Additionally, in this section, we will present the experimental results where LSC combines two and four centrality measures. The two centrality measures are DC and EC, respectively. The [DC, EC] combination will be referred to as LSC2. The four centrality measures are DC, EC, CC, and BC, respectively. The combination [DC, EC, CC, BC] will be referred to as LSC4. We calculated the Kendall tau correlation coefficient for the ranking list created by LSC2 and LSC4. The ranking performances are shown in Fig. 6.

The original LSC (namely, [DC, EC, CC] combination) and LSC4 give very close results. LSC2 gives the best result in only one dataset. Increasing the number of combined centrality measures from 2 to 3 increased the performance. There is no significant difference between using 3 and 4 centrality measures. This result can be interpreted as follows. If the used centrality measures assign different ranks to all nodes, adding one more centrality measure will not change the rank. Therefore, we can say that the original LSC gives different ranks to the vast majority of nodes, and the addition of a fourth measure (i.e., the addition of BC) does not make a meaningful change. Nevertheless, as a future study, it would be helpful to investigate how more and different centrality measures work in more extensive networks.

7.6. Operating speed of LSC and GC

In addition to the performance of a centrality measure, its speed is also essential. We compared the calculation times of the LSC and GC on 6 datasets (Table 3). We ran the calculations 1000 times on a computer with an Intel i7 2.8 GHz processor and 16 GB of RAM and averaged the runtimes.

In four of the six datasets, LSC is faster than the GC. The crucial time-consuming factor for the GC is the calculation of each node for its 3-hop neighbors. In dense networks, 3-hop neighbors make up a large part of the network. On the other hand, the chief time-

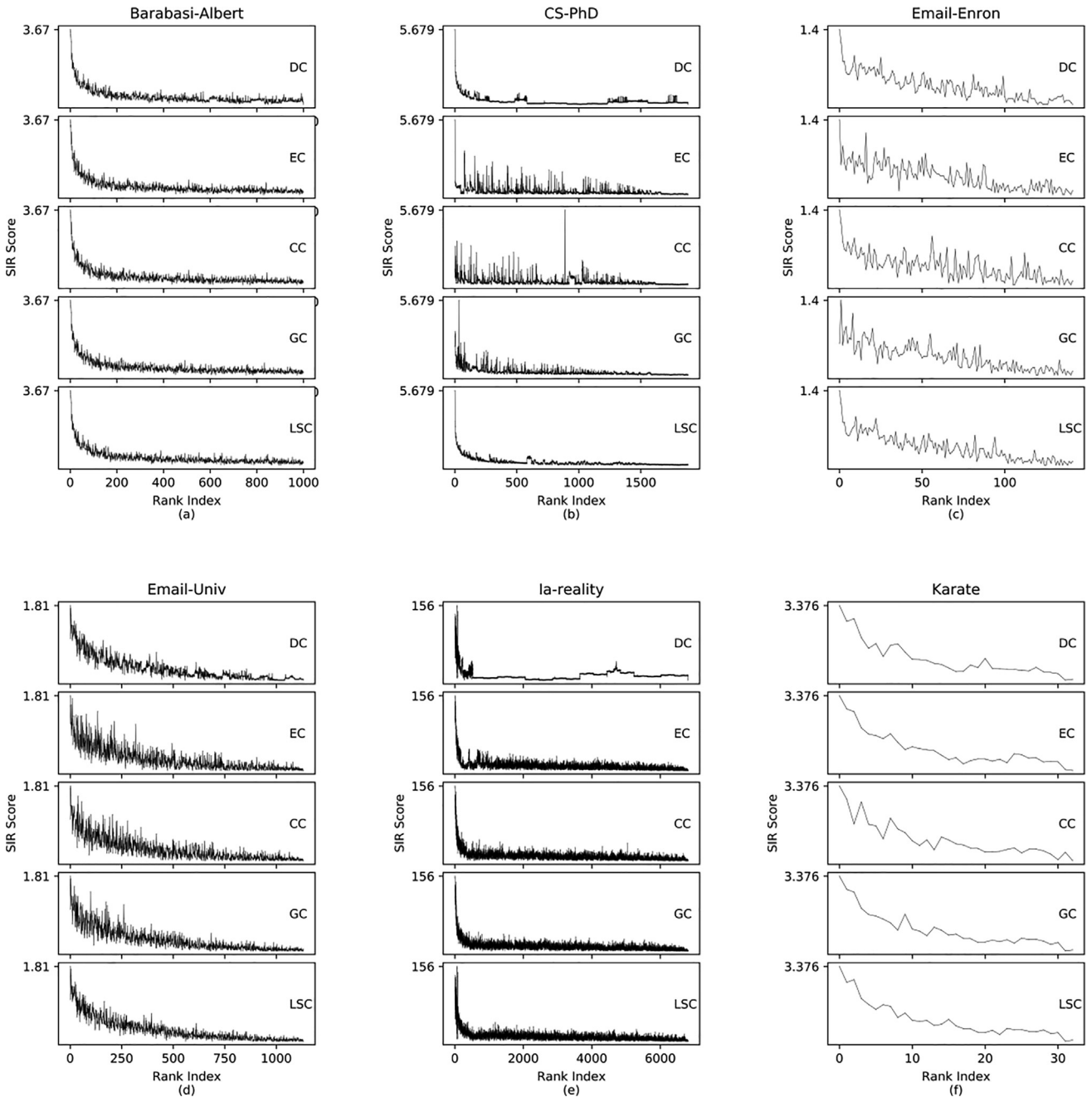

Fig. 3. SIR score trends of nodes sorted from large to small according to different centrality measures.

Table 2
The number of matching nodes in the top 5% of the ranking list of the centrality measures and the ranking list of the SIR simulation.

	DC	EC	CC	BC	GC	LSC
Barabasi-Albert	40	42	42	41	42	42
CS-PhD	77	15	27	46	53	80
Email-Enron	4	2	4	4	3	4
Email-Univ	41	29	35	36	36	41
Ia-reality	201	166	245	199	245	247
Karate	1	1	1	1	1	1

consuming element of the LSC is the calculation of the centrality measures it uses. Once the centrality measures are calculated, LSC performs only the sorting process. Also, several studies have been carried out to achieve a faster calculation of centrality measures (van der Grinten et al., May 2020). By using these methods, the operating time of the LSC can also be reduced.

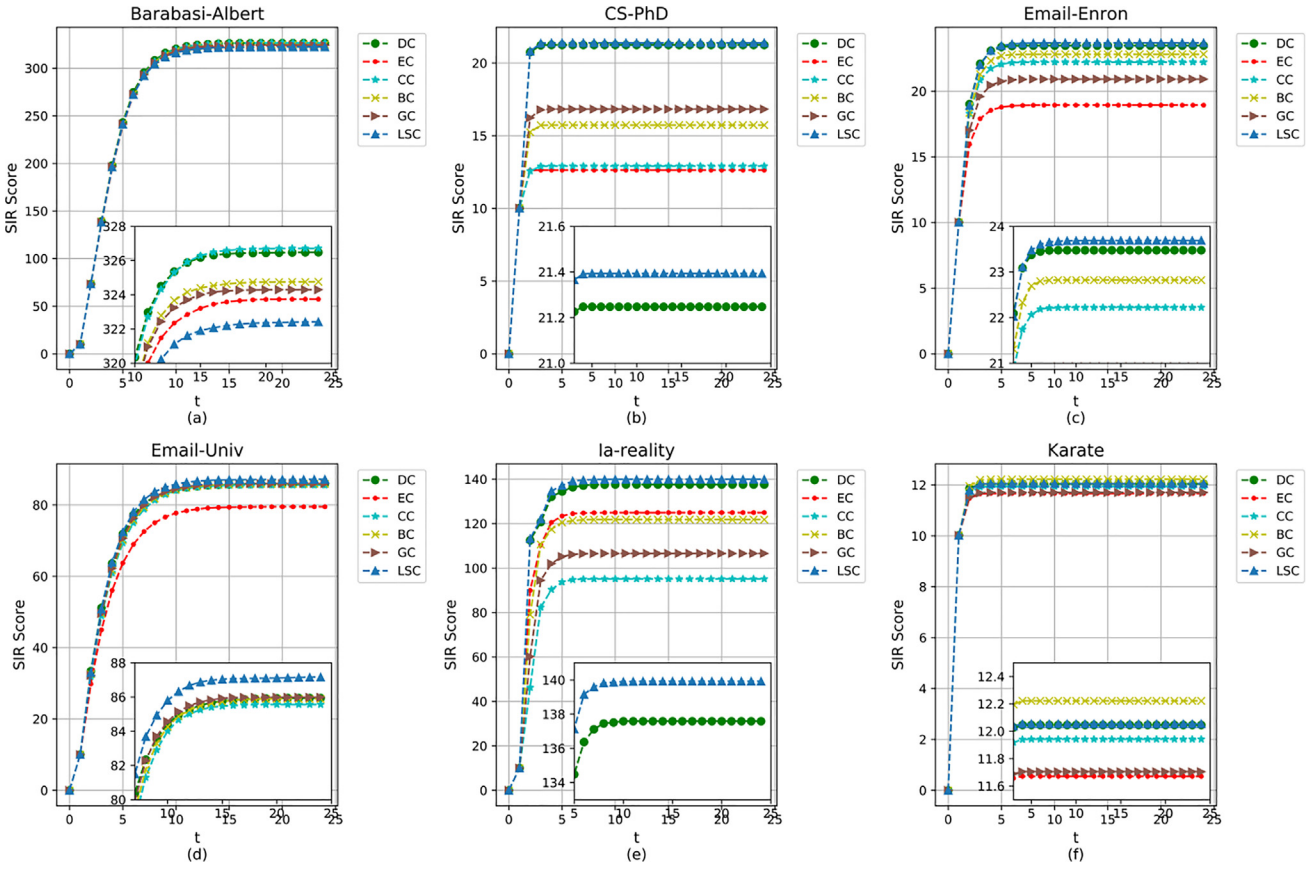

Fig. 4. The SIR scores of nodes in the top-10 are determined by different centrality measures. ($\boldsymbol{t} = 25, \boldsymbol{\beta} = 0.05$).

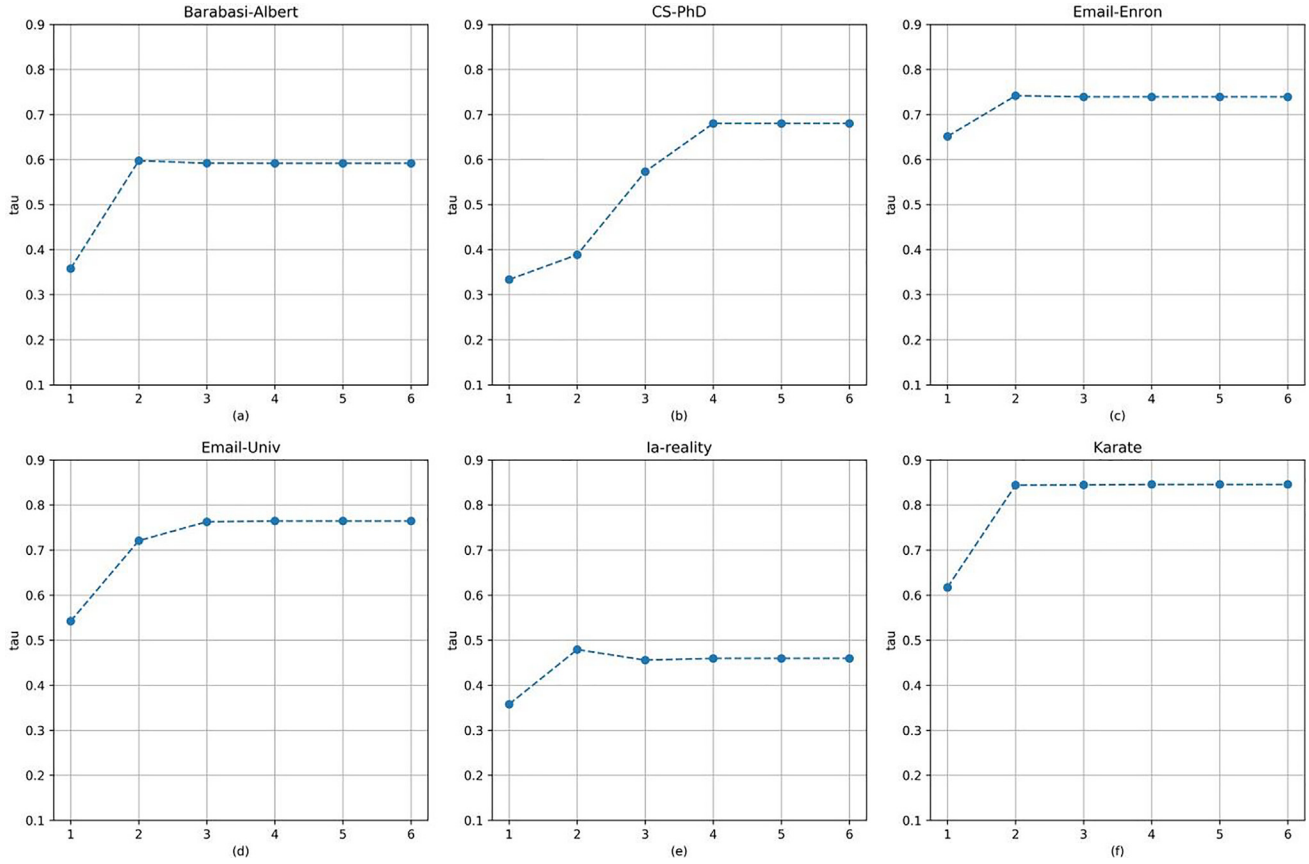

Fig. 5. Kendall ***tau*** correlation coefficient values of LSC for different decimal precisions. The x-axis of subfigures is the decimal precisions. Infection rate: $\boldsymbol{\beta} = 0.1$ for (b), (d), and (f)**;** $\boldsymbol{\beta} = 0.01$ for (a), (c), and (e). Recovery rate: $\boldsymbol{\gamma} = 1$ for all experiments.

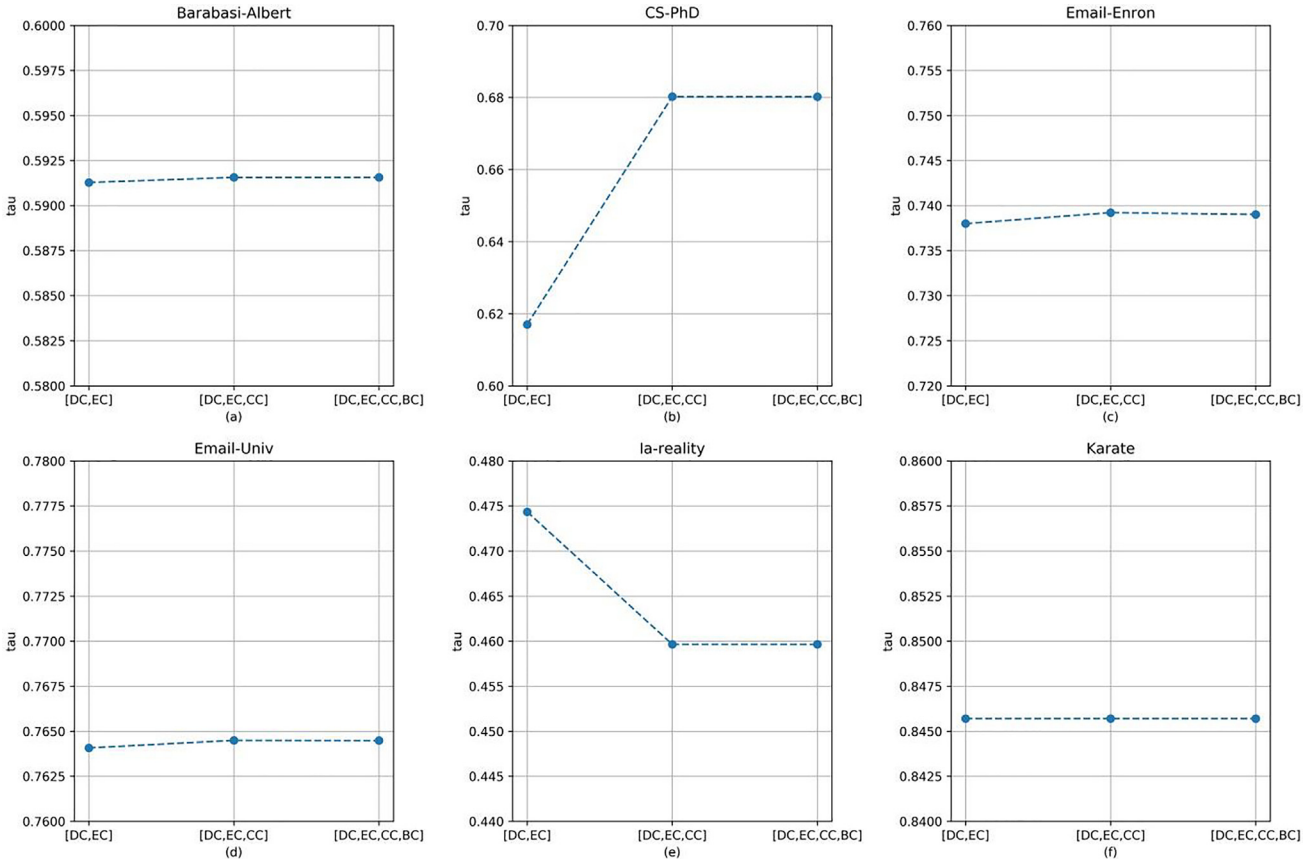

Fig. 6. Kendall ***tau*** correlation coefficient values of LSC with the different number of centrality measures. the x-axis of subfigures is the combined centrality measures by LSC. Infection rate: $\beta = 0.1$ for (b), (d), and (f); $\beta = 0.01$ for (a), (c), and (e). Recovery rate: $\gamma = 1$ for all experiments.

Table 3
Calculated times (s) of LSC and GC for six datasets.

	Barabasi-Albert	CS-PhD	Email-Enron	Email-Univ	Ia-reality	Karate
LSC	7.476	3.045	0.113	6.068	121.858	0.065
GC	51.467	6.011	0.367	11.351	101.591	0.005

8. Discussion and conclusions

This study proposed a new centrality measure for complex networks that ranks the nodes according to their spreading abilities under the SIR model. Like lexical sorting, LSC sorts nodes according to multiple centrality measures. So, LSC is a new centrality measure and a framework for creating new centrality measures. Our detailed simulations have demonstrated that LSC performs better than well-known centrality measures such as degree, closeness, eigenvector, and betweenness and the state-of-the-art GC measures. Future studies might consider using different centrality measures in different orders and different decimal precisions in LSC.

Author Biography

Aybike ŞİMŞEK received her BS degree from the Selçuk University Department of Computer Engineering in 2003, her MS degree from the Gazi University Department of Computer Engineering in 2010, and her Ph.D. from the Düzce University Department of Electrical-Electronics and Computer Engineering in 2018. She was a postdoctoral researcher in the Department of Computer Science at the Humboldt University of Berlin from August 2018 to July 2019. She was a lecturer in the Department of Computer Programming at Düzce University from January 2015 to December 2020. She has been working as an assistant professor in the Department of Computer Engineering at Düzce University since December 2020. Her current research interests include social network analysis, complex networks, and epidemic modeling.

Declaration of Competing Interest

The author declares that she has no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Barabási, A.-L., Pósfai, M., 2016. Network Science. Cambridge University Press, Cambridge.
Hancock, J., Menche, J. "Structure and Function in Complex Biological Networks," in Reference Module in Biomedical Sciences, Elsevier, 2020.
Borgatti, S.P., Mehra, A., Brass, D.J., Labianca, G., 2009. Network analysis in the social sciences. Science (80-.) 323 (5916), 892–895.

Gao, J., Barzel, B., Barabási, A.-L., Feb. 2016. Universal resilience patterns in complex networks. Nature 530 (7590), 307–312.
Xu, X., Rong, Z., Tian, Z., Wu, Z.-X., 2019. Timescale diversity facilitates the emergence of cooperation-extortion alliances in networked systems. Neurocomputing 350, 195–201.
Liu, C. et al., Dec. 2018. Popularity enhances the interdependent network reciprocity. New J. Phys. 20, (12) 123012.
Xu, D., Tian, Z., Lai, R., Kong, X., Tan, Z., Shi, W. 2020. Deep learning based emotion analysis of microblog texts. Inf. Fusion 64 (September 2019),1–11.
Zareie, A., Sheikhahmadi, A., 2018. A hierarchical approach for influential node ranking in complex social networks. Expert Syst. Appl. 93, 200–211.
Kempe, D., Kleinberg, J., Tardos, É., 2003. Maximizing the spread of influence through a social network. In: in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* -, p. 137.
Borgatti, S.P., 2006. Identifying sets of key players in a social network. Comput. Math. Organ. Theory 12 (1), 21–34.
Yang, L., Li, Z., Giua, A., 2020. Containment of rumor spread in complex social networks. Inf. Sci. (Ny) 506, 113–130.
Zhang, Y., Su, Y., Weigang, L.i., Liu, H., 2018. Rumor and authoritative information propagation model considering super spreading in complex social networks. Phys. A Stat. Mech. its Appl. 506, 395–411.
Ali, S.S., Anwar, T., Rizvi, S.A.M. 2020. A revisit to the infection source identification problem under classical graph centrality measures. Online Soc. Networks Media 17 (xxxx),100061.
Zhang, K., Du, H., Feldman, M.W., 2017. Maximizing influence in a social network: Improved results using a genetic algorithm. Phys. A Stat. Mech. its Appl. 478, 20–30.
Salavati, C., Abdollahpouri, A., Manbari, Z., 2019. Ranking nodes in complex networks based on local structure and improving closeness centrality. Neurocomputing 336, 36–45.
Li, M., Zhang, Q.i., Deng, Y., 2018. Evidential identification of influential nodes in network of networks. Chaos, Solitons Fractals 117, 283–296.
Şimşek, M., Meyerhenke, H. 2020. Combined centrality measures for an improved characterization of influence spread in social networks. J. Complex Networks 8 (1).
Alshahrani, M., Fuxi, Z., Sameh, A., Mekouar, S., Huang, S., 2020. Efficient algorithms based on centrality measures for identification of top-K influential users in social networks. Inf. Sci. (Ny) 527, 88–107.
Ma, L.-L., Ma, C., Zhang, H.-F., Wang, B.-H., 2016. Identifying influential spreaders in complex networks based on gravity formula. Phys. A Stat. Mech. its Appl. 451, 205–212.
Yang, Y., Wang, X., Chen, Y., Hu, M., Ruan, C., 2020. A Novel Centrality of Influential Nodes Identification in Complex Networks. IEEE Access 8, 58742–58751.
Saxena, A., Iyengar, S. 2020 Centrality measures in complex networks: a survey.
Cherifi, H., Palla, G., Szymanski, B.K. Lu, X. 2019. On community structure in complex networks: challenges and opportunities. Appl. Netw. Sci. 4 (1).
Yan, X.-L., Cui, Y.-P., Ni, S.-J., 2020. Identifying influential spreaders in complex networks based on entropy weight method and gravity law. Chinese Phys. B 29, (4) 048902.
Maji, G., Mandal, S., Sen, S., 2020. A systematic survey on influential spreaders identification in complex networks with a focus on K-shell based techniques. Expert Syst. Appl. 161, 113681.
Azaouzi, M., Mnasri, W., Ben Romdhane, L., 2021. New trends in influence maximization models. Comput. Sci. Rev. 40, 100393.
Newman, M., 2018. *Networks*, Second. Oxford University Press, Oxford, UK.
Sabidussi, G., 1966. The centrality index of a graph. Psychometrika 31 (4), 581–603.
Freeman, L.C., 1977. A Set of Measures of Centrality Based on Betweenness. Sociometry 40 (1), 35. https://doi.org/10.2307/3033543.
Bonacich, P., Mar. 1987. Power and Centrality: A Family of Measures. Am. J. Sociol. 92 (5), 1170–1182.
Katz, L., 1953. A new status index derived from sociometric analysis. Psychometrika 18 (1), 39–43.
Page, L., Brin, S., Motwani, R., Winograd, T., 1999. The PageRank Citation Ranking: Bringing Order to the Web. Stanford InfoLab.
Ghalmane, Z., Cherifi, C., Cherifi, H., El Hassouni, M., 2019a. Centrality in Complex Networks with Overlapping Community Structure. Sci. Rep. 9 (1), 1–29.
Ghalmane, Z., El Hassouni, M., Cherifi, C., Cherifi, H. 2019. Centrality in modular networks. EPJ Data Sci. 8 (1).
Zhao, Y., Li, S., Jin, F., 2016. Identification of influential nodes in social networks with community structure based on label propagation. Neurocomputing 210, 34–44.
Zhao, Z., Wang, X., Zhang, W., Zhu, Z., 2015. A community-based approach to identifying influential spreaders. Entropy 17 (4), 2228–2252.
Andrade, R.L.d., Rêgo, L.C., 2019. p-means centrality. Commun. Nonlinear Sci. Numer. Simul. 68, 41–55.
Zhao, J., Wang, Y., Deng, Y., 2020. Identifying influential nodes in complex networks from global perspective. Chaos, Solitons Fractals 133, 109637.
Zhao, J., Song, Y., Deng, Y., 2020. A novel model to identify the influential nodes: Evidence theory centrality. IEEE Access 8, 46773–46780.
Keng, Y.Y., Kwa, K.H., McClain, C., 2020. Convex combinations of centrality measures. J. Math. Sociol.
Şimşek, A., Kara, R., 2018. Using swarm intelligence algorithms to detect influential individuals for influence maximization in social networks. Expert Syst. Appl., Dec. 114, 224–236.
Ullman, J.D., Hopcroft, J., Aho, A.V., 1974. The Design and Analysis of Computer Algorithms. Addison-Wesley.
Wandelt, S., Shi, X., Sun, X., 2020. Complex network metrics: can deep learning keep up with tailor-made reference algorithms? IEEE Access 8, 68114–68123.
Oldham, S., Fulcher, B., Parkes, L., Arnatkevičiūtė, A., Suo, C., Fornito, A. 2019. Consistency and differences between centrality measures across distinct classes of networks. PLoS One 14 (7) e0220061.
Barabási, A.-L., Albert, R. 1999. Emergence of scaling in random networks. Science (80-.) 286 (5439) 509–512.
Zachary, W.W., 1977. An information flow model for conflict and fission in small groups. J. Anthropol. Res. 33 (4), 452–473.
Rossi, R.A., Ahmed, N.K., 2015. The Network Data Repository with Interactive Graph Analytics and Visualization. in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*.
Guimerà, R., Danon, L., Díaz-Guilera, A., Giralt, F., Arenas, A., 2003. Self-similar Community Structure in a Network of Human Interactions. Phys. Rev. E 68 (6), 65103.
De Nooy, W., Mrvar, A., Batagelj, V., 2011. Exploratory social network analysis with Pajek, vol. 27. Cambridge University Press.
Eagle, N., (Sandy) Pentland, A., 2006. Reality mining: sensing complex social systems. Pers. Ubiquitous Comput. 10 (4), 255–268.
Kendall, M.G., 1938. A new measure of rank correlation. Biometrika 30 (1-2), 81–93.
Sheng, J. et al., 2020. Identifying influential nodes in complex networks based on global and local structure. Phys. A Stat. Mech. its Appl. 541, 123262.
Hagberg, A.A., Schult, D.A., Swart, P.J., 2008. Exploring Network Structure, Dynamics, and Function using NetworkX. In: in *Proceedings of the 7th Python in Science Conference*, pp. 11–15.
van der Grinten, A., Angriman, E., Meyerhenke, H., May 2020. Scaling up network centrality computations – A brief overview. it - Inf Technol. 62 (3–4), 189–204.