

Identifying multiple influential spreaders in complex networks based on spectral graph theory

Dong-Xu Cui(崔东旭), Jia-Lin He(何嘉林), Zi-Fei Xiao(肖子飞), and Wei-Ping Ren(任卫平)
Citation: Chin. Phys. B, 2023, 32 (**9**): 098904. DOI: 10.1088/1674-1056/acac0b
Journal homepage: http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Self-similarity of complex networks under centrality-based node removal strategy

Dan Chen(陈单), Defu Cai(蔡德福), and Housheng Su(苏厚胜)
Chin. Phys. B, 2023, 32 (**9**): 098903. DOI: 10.1088/1674-1056/acd3e2

Identification of key recovering node for spatial networks

Zijian Yan(严子健), Yongxiang Xia(夏永祥), Lijun Guo(郭丽君), Lingzhe Zhu(祝令哲), Yuanyuan Liang(梁圆圆), and Haicheng Tu(涂海程)
Chin. Phys. B, 2023, 32 (**6**): 068901. DOI: 10.1088/1674-1056/acb75f

AIGCrank: A new adaptive algorithm for identifying a set of influential spreaders in complex networks based on gravity centrality

Ping-Le Yang(杨平乐), Lai-Jun Zhao(赵来军), Chen Dong(董晨),Gui-Qiong Xu(徐桂琼), and Li-Xin Zhou(周立欣)
Chin. Phys. B, 2023, 32 (**5**): 058901. DOI: 10.1088/1674-1056/ac8e56

Topological phase transition in network spreading

Fuzhong Nian(年福忠) and Xia Zhang(张霞)
Chin. Phys. B, 2023, 32 (**3**): 038901. DOI: 10.1088/1674-1056/ac7bfa

Analysis of cut vertex in the control of complex networks

Jie Zhou(周洁), Cheng Yuan(袁诚), Zu-Yu Qian(钱祖燏), Bing-Hong Wang(汪秉宏), and Sen Nie(聂森)
Chin. Phys. B, 2023, 32 (**2**): 028902. DOI: 10.1088/1674-1056/aca208

Identifying multiple influential spreaders in complex networks based on spectral graph theory

Dong-Xu Cui(崔东旭), Jia-Lin He(何嘉林)[†], Zi-Fei Xiao(肖子飞), and Wei-Ping Ren(任卫平)

School of Computer Science and Engineering, China West Normal University, Nanchong 637009, *China*

(Received 13 September 2022; revised manuscript received 11 December 2022; accepted manuscript online 16 December 2022)

One of the hot research topics in propagation dynamics is identifying a set of critical nodes that can influence maximization in a complex network. The importance and dispersion of critical nodes among them are both vital factors that can influence maximization. We therefore propose a multiple influential spreaders identification algorithm based on spectral graph theory. This algorithm first quantifies the role played by the local structure of nodes in the propagation process, then classifies the nodes based on the eigenvectors of the Laplace matrix, and finally selects a set of critical nodes by the constraint that nodes in the same class are not adjacent to each other while different classes of nodes can be adjacent to each other. Experimental results on real and synthetic networks show that our algorithm outperforms the state-of-the-art and classical algorithms in the SIR model.

Keywords: spectral graph theory, Laplace matrix, influence maximization, multiple influential spreaders

PACS: 89.75.Fb **DOI:** 10.1088/1674-1056/acac0b

1. Introduction

The influence maximization issue was first introduced by Domingos and Richardson in 2001.[1] One of the main problems of influence maximization is to find k critical nodes in a complex network, so that the number of nodes being influenced in the network is maximized. The selection of critical nodes in the influence maximization problem is very significant for controlling the spread of epidemics,[2] controlling the spread of rumors, apprehending criminal gangs,[3] and tapping potential customers.[4] Many scholars have already made studies on related aspects.[5–10]

Among the algorithms to select out multiple influential spreaders in complex networks, the most commonly used basic idea is the top-k strategy, which selects the top k nodes with the largest values as the initial spreaders. For example, degree centrality (DC)[11] measures the importance of a node by the number of nodes that are directly connected to it. The greater the DC, the greater impact on the propagation, so the top k nodes with the largest degree values are selected as the initial spreaders in the propagation process. However, DC only considers the first-order neighbors and fails to consider the connections of other nodes (non-neighboring nodes) with it in the algorithm. In view of this, Chen *et al.* proposed the LocalRank[12] algorithm, which considers fourth-order neighbors and improves on the DC. Betweenness centrality (BC)[13] and closeness centrality (CC)[14,15] consider global network information. BC measures how important a node is based on how many shortest paths pass through it. CC evaluates the importance of a node by the sum of its distances to other nodes in the network, with the closer being more important. In addition, there is ClusterRank,[16] which considers both the degree of the node and the clustering coefficient, and PageRank[17] and LeaderRank,[18] proposed as algorithms to calculate the importance of Internet pages. Fan *et al.*[19] proposed the cycle ratio (CR) algorithm, which proposes and defines the cycle number matrix and cycle ratio by analyzing the information contained in cycles in a network structure; the cycle number matrix contains the cycle of nodes in the network information, and the cycle ratio measures the importance of nodes. Algorithms such as these all select the top k nodes with the largest values as the initial spreaders in the propagation process, i.e., using the top-k strategy.

Although the nodes selected by the top-k strategy can improve the final affected scale, it causes local propagation if the selected nodes are clustered, because the top-ranked nodes selected by these algorithms might be clustered as a rich-club.[20] Initially, some nodes that contribute to effective information propagation are selected. As more nodes are added to the rich-club, their contributions to effective information propagation become smaller. To avoid the occurrence of a rich-club, some scholars have started to study the dispersion of selected nodes, such as excluding the selected nodes from being neighboring nodes, etc. Kempe *et al.*[21] proposed a "mountain climbing" greedy strategy to select k critical nodes, and the performance of this greedy strategy is close to 63% of the optimal performance. Chen *et al.*[22] proposed a "degree discount" heuristic algorithm to select critical nodes, which has a wider propagation range and lower time complexity. Zhao *et al.*[23] applied the "coloring problem" in graph theory to find multiple influential spreaders in complex networks, by dividing nodes with the same color into same independent sets, and selecting a certain percentage of important nodes from each

[†]Corresponding author. E-mail: hejialin32@126.com

© 2023 Chinese Physical Society and IOP Publishing Ltd

independent set; the larger the independent set is, the more nodes are selected, and this algorithm accelerates the propagation process. Guo *et al.*[24] proposed distance-based coloring algorithms, which significantly improve the final affected scale. Flaviano Morone *et al.*[25] addressed the problem of quantifying nodes' influence by finding the optimal (that is, minimal) set of structural influencers, and proposed collective influence (CI), a scalable algorithm to solve the optimization problem in large-scale real data sets. Zhang *et al.*[26] proposed the VoteRank algorithm, which selects multiple influential spreaders through each round of voting iterations and weakens the voting power of their neighbors; Sun *et al.*[27] proposed WVoteRank (WVR) based on the VoteRank algorithm, which can be used not only on unweighted networks but also on weighted networks. Fan *et al.*[28] proposed an algorithm to find critical nodes in complex networks by introducing a deep reinforcement learning framework.

However, although the dispersion strategy improves the final affected scale by increasing the dispersion of the selected nodes, the disadvantage is that the local structure of many critical nodes is not conducive to disease propagation. Based on this, this paper proposes a compromise strategy to balance the top-k strategy and the dispersion strategy. In the experimental study of seven real networks and four synthetic networks generated by the LFR[29] model that is a community generation model proposed by Lancichinetti *et al.*, the algorithm in this paper has a clear advantage in the final affected scale, compared with BC, CC, DC, CI, WVR, CR and FINDER.

The major contributions of this paper are shown as follows.

(i) From analysis of two network topologies, it is found that the local tree structure is more conducive to the spread of disease than the local cluster structure.

(ii) Using the eigenvectors of the Laplace matrix to classify structural importance of nodes, a balancing strategy is proposed in which nodes of the same class are not adjacent, while nodes of different classes are adjacent.

2. Preliminaries

2.1. Graph Laplacian quadratic form

Given a graph $G=(V,E)$, V denotes the set of nodes in the graph, which is assumed to be of length N. The graph signal is a mapping describing $V \to R$, expressed in the form of a unit vector: $\boldsymbol{x}=[x_1,x_2,\ldots,x_N]^{\mathrm{T}}$, where x_i denotes the signal strength on node v_i. When studying the properties of the graph signal, the topology of the graph should be considered in addition to the strength of the graph signal. The Laplace matrix is the core object used to study the structural properties of the graph and is defined as follows:

$$\boldsymbol{L}=\boldsymbol{D}-\boldsymbol{A}, \tag{1}$$

where $\boldsymbol{A}$ and $\boldsymbol{D}$ denote the adjacency matrix and the diagonal matrix of the degree of the graph respectively, and $D_{ii}=\sum_j A_{ij}$ denotes the degree of node v_i. Based on the Laplace matrix, we can define a very important quadratic form as follows:

$$TV(\boldsymbol{x})=\boldsymbol{x}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{x}=\sum\nolimits_{e_{ij}\in E}(x_i-x_j)^2, \tag{2}$$

where $TV(\boldsymbol{x})$ is the total variance of the graph signal, which sums the differences of the signals on each edge and portrays the overall smoothness of the graph signal.

Assuming that the N eigenvalues of the Laplace matrix $\boldsymbol{L}$ are $\gamma_1 \geq \gamma_2 \geq \cdots \geq \gamma_N$ and the corresponding unit eigenvectors are $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_N$, respectively, then equation (2) can be rewritten as

$$TV(\boldsymbol{x})=\boldsymbol{x}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{x}=\boldsymbol{x}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{x}=\tilde{\boldsymbol{x}}^{\mathrm{T}}\boldsymbol{\Lambda}\tilde{\boldsymbol{x}}=\sum\nolimits_{i=1}^{N}\gamma_i\tilde{x}_i^2, \tag{3}$$

where $\tilde{x}_i$ denotes the coordinates of signal $\boldsymbol{x}$ on the unit eigenvector $\boldsymbol{u}_i$ corresponding to γ_i, Λ denotes the diagonal matrix consisting of eigenvalues and $\boldsymbol{U}$ denotes the orthogonal eigenmatrix consisting of eigenvectors. From Eq. (3), it can be seen that when the graph signal $\boldsymbol{x}=\boldsymbol{u}_i$, then $TV(u_i)=\gamma_i$. Hence, for $\gamma_1 \geq \gamma_2 \geq \cdots \geq \gamma_N$, the inequality $TV(u_1) \geq TV(u_2) \geq \cdots \geq TV(u_N)$ holds.

2.2. SIR model

The SIR model[30,31] and the SIS model are two relatively well-known infectious disease models, and in this experiment the SIR model is used as an evaluation metric for the performance of the algorithm. In the SIR model, each node in the network can only be in one of the following three states at each time step: susceptible (S), infected (I), and recovered (R). At time step t, infected node u randomly selects the neighboring node v. If node v belongs to susceptible nodes, node u infects it with probability α. Meanwhile, infected node u becomes a recovered node with probability β. The disease propagation process ends when all infected nodes in the network have become recovered nodes. At this point, the final scale of affected nodes $F(n)$ is used as a measure of the performances of different algorithms, defined as follows:

$$F(n)=\frac{N_{\mathrm{R}}}{N'}, \tag{4}$$

where N' and N_{R} denote the number of all nodes in the network and the number of nodes eventually recovered, respectively. In addition, in the SIR model, the basic reproduction number $\lambda=\alpha/\beta$ reflects the disease propagation trend; for example, when $\lambda>1$, the disease is able to spread throughout the population.

2.3. The K-core algorithm

The K-core[32] algorithm is a subgraph mining algorithm to find the set of nodes in a graph that match a specified core degree, requiring each node to be connected to at least K other

nodes in that graph. In addition, K-core hierarchically divides the graph according to K values. For example, 2-core removes nodes with degree less than 2 each time, 3-core removes nodes with degree less than 3 each time, and each repeated removal of nodes as well as edges is specified by K values as shown in Fig. 1. Thus, it shows that nodes with larger core degrees tend to have larger degrees, i.e., more neighboring nodes, which can play a greater role to a certain extent.

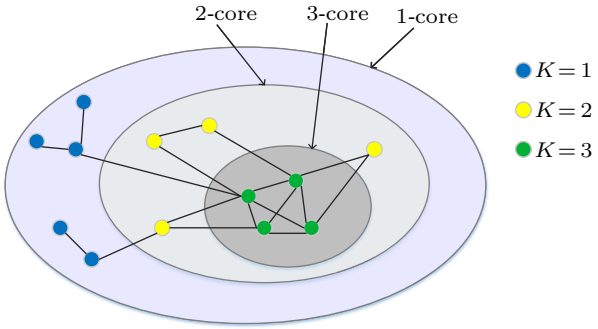


Fig. 1. Schematic of the 1-core, 2-core, and 3-core subgraphs.

3. Multiple influential spreaders identification algorithm

3.1. Node local structure analysis

From Subsection 2.3, it is clear that the outermost nodes of the network, i.e., those with $K = 1$, play a very limited role in the propagation process. Therefore, we only analyze the subgraph of the nodes with $K \geq 2$ in the network. First, all nodes with $K = 1$ are removed from the initial network G to obtain the new network G'. Next, the adjacency matrix $\boldsymbol{A}'$ of the new network G' is normalized, which can be obtained as follows:

$$\tilde{\boldsymbol{A}}' = \boldsymbol{D}'^{-1}\boldsymbol{A}'. \tag{5}$$

Assuming that $\tilde{\boldsymbol{a}}'_i$, the i-th row of $\tilde{\boldsymbol{A}}'$, denotes the graph signal of node v_i, and $\tilde{a}'_{ij}$ denotes the signal strength of the j-th element of $\tilde{\boldsymbol{a}}'_i$, the total variance of the graph signal of node v_i is:

$$TV\left(\tilde{\boldsymbol{a}}'_i\right) = \tilde{\boldsymbol{a}}'^{\mathrm{T}}_i \boldsymbol{L}\tilde{\boldsymbol{a}}'_i = \sum\nolimits_{e_{pq}\in E}\left(\tilde{a}'_{ip} - \tilde{a}'_{iq}\right)^2. \tag{6}$$

In $\tilde{\boldsymbol{a}}'_i$, $\tilde{a}'_{ij} = 1/d_i$, if $v_j \in N'(v_i)$, and $\tilde{a}'_{ij} = 0$ otherwise, where d_i and $N'(v_i)$ denote the degree of v_i and the set of neighbor nodes, respectively. Based on this, equation (6) can be further simplified as follows:

$$TV\left(\tilde{\boldsymbol{a}}'_i\right) = \frac{1}{d_i^2}\left[\left(\sum\nolimits_{v_j\in N(v_i)} d_j\right) - 2m_i\right] = p_i - g_i, \tag{7}$$

where $p_i = (1/d_i^2)\sum_{v_j\in N'(v_i)} d_j$, $g_i = (2m_i/d_i^2)$ and m_i denotes the number of edges between the neighboring nodes of node v_i. Let $d_j^{\max}$ and $d_j^{\min}$ denote the maximum degree and minimum degree among the neighboring nodes of node v_i, respectively; then $\frac{d_j\min}{d_i} \leq p_i \leq \frac{d_j^{\max}}{d_i}$. When d_i is certain, the larger p_i is, the more connected edges the neighboring node $v_j \in N'(v_i)$ has with the rest of the network, and the greater the probability of node v_i spreading the disease to distant locations. In addition, the second term $g_i = \frac{2m_i}{d_i^2} \approx \frac{m_i}{(d_i(d_i-1))/2} = c_i$, where c_i is the clustering factor of node v_i. The larger c_i is, the more edges between neighbor nodes of node v_i, and the more v_i tends to spread the disease locally. From the above analysis, it is clear that the more node v_i tends to act as a "bridge", i.e., it connects several unconnected nodes of large degree, the larger the $TV(\tilde{\boldsymbol{a}}'_i)$ of node v_i, and the more likely node v_i spreads the disease outward.

Next, we illustrate how to calculate the $TV(\tilde{\boldsymbol{a}}'_i)$ of a node using two topologies. In the tree structure, as shown in Fig. 2(a), node v_1 has $p_1 = (1/d_1^2)(d_2+d_3) = 10/4$ and $g_1 = (2\times m_1)/d_1^2 = 0$, while in the cluster structure, as shown in Fig. 2(b), node v_1 has $p'_1 = (1/d_1^2)(d_2+d_3+d_4+d_5+d_6) = 26/25$ and $g'_1 = (2\times m_1)/d_1^2 = 40/25$. From the calculation results, it is clear that the value of p_i of node v_1 in the tree structure is greater than the value of p'_i in the cluster structure, and although the tree structure itself does not have many connected nodes, its probability of spreading the disease to distant places is greater when its neighbor nodes have more connected edges with the rest of the network. In addition, g'_i in Fig. 2(b) is greater than g_i in Fig. 2(a) and the number of connected edges between its neighboring nodes is larger, and local propagation tends to occur. From the above analysis, we know that the algorithm is more applicable to the local tree structure. In addition, we can learn from the analysis that when the $TV(\tilde{\boldsymbol{a}}'_i) = p_i - g_i$ value of a node is larger, it tends to spread out more, increasing the probability of propagation to distant locations and reducing the possibility of local propagation.

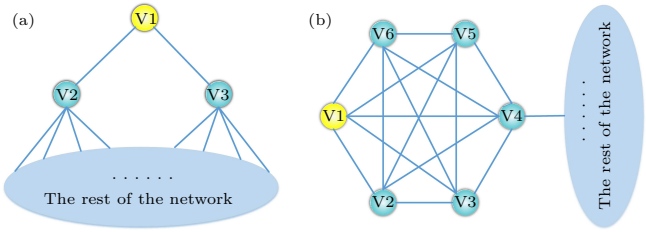


Fig. 2. Two classical topological structures. (a) Local tree structure; (b) local cluster structure.

3.2. Trade-off strategy

Currently, there are two strategies for identifying a set of critical nodes: the top-k strategy and the dispersion strategy. The top-k strategy selects the top k nodes with the largest value, but the value selected is a single measure that does not fully represent the nodes' propagation ability, causing the selected nodes to cluster together and overlap, resulting in a rich-club phenomenon. The dispersion strategy focuses on the dispersion of the selected nodes, so many critical nodes not

conducive to disease propagation are selected. In view of the limitations of the above two strategies, this paper proposes a compromise strategy that balances the importance of critical nodes and the dispersion of critical nodes.

Given the eigenvectors $\boldsymbol{u}_1$, $\boldsymbol{u}_2$,..., $\boldsymbol{u}_N$ of the Laplacian matrix, the projection s_{ij} of $\tilde{\boldsymbol{a}}_i'$ on the eigenvector $\boldsymbol{u}_j$ is defined as

$$s_{ij} = \frac{\tilde{\boldsymbol{a}}_i' \boldsymbol{u}_j}{\boldsymbol{u}_j^{\mathrm{T}} \boldsymbol{u}_j}. \tag{8}$$

Since $\boldsymbol{u}_j$ is a unit vector, then $\boldsymbol{u}_j^{\mathrm{T}} \boldsymbol{u}_j = 1$, so equation (8) can be reduced to $s_{ij} = \tilde{\boldsymbol{a}}_i' \boldsymbol{u}_j$. In fact, s_{ij} also measures the similarity between $\tilde{\boldsymbol{a}}_i'$ and $\boldsymbol{u}_j$, i.e., the larger s_{ij} is, the more similar $\tilde{\boldsymbol{a}}_i'$ is to $\boldsymbol{u}_j$. Since the N eigenvectors $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_N$ of the Laplacian matrix are orthogonal to each other, the similarity of any two eigenvectors is 0, i.e., for any $i \neq j$, $\boldsymbol{u}_i \boldsymbol{u}_j = 0$. Therefore, if the similarity s_{ij} between $\tilde{\boldsymbol{a}}_i'$ and $\boldsymbol{u}_j$ is larger, then the similarity s_{ik} between $\tilde{\boldsymbol{a}}_i$ and the other eigenvectors $\boldsymbol{u}_k$ is smaller. Assuming that $\tilde{\boldsymbol{a}}_i'$ is most similar to $\boldsymbol{u}_j$, then according to Eq. (3), $TV\left(\tilde{\boldsymbol{a}}_i'\right)$ can be approximated as

$$TV\left(\tilde{\boldsymbol{a}}_i'\right) \approx TV\left(\boldsymbol{u}_j\right). \tag{9}$$

Therefore, we can use the N eigenvectors of $\boldsymbol{L}$ as the benchmark category vectors to classify the importance of local structure of N nodes in the network. Specifically, for each node v_i, the baseline category vector ordinal number l_i with the maximum similarity s_{ij} is calculated, i.e.,

$$l_i = \arg\max_j s_{ij}. \tag{10}$$

The set $\boldsymbol{C}_j$ of nodes contained in each benchmark category vector $\boldsymbol{u}_j$ is then found, i.e.,

$$\boldsymbol{C}_j = \{v_i | l_i = j\}. \tag{11}$$

It is known from Subsection 3.1 that the larger $TV(\tilde{\boldsymbol{a}}_i')$ is, the higher the probability of node v_i spreading the disease outward. Hence, the importance of the local structure of nodes in $\boldsymbol{C}_1$, $\boldsymbol{C}_2$ and other sets decreases successively. To select a set of critical nodes, we use the constraint that nodes in the same class are not adjacent, but nodes from different classes can be adjacent to balance the importance of structure and dispersion. Hence, we give priority to the nodes in the set $\boldsymbol{C}_1$ when selecting a set of critical nodes. First, the nodes in $\boldsymbol{C}_1$ are first sorted by similarity s_{i1} from largest to smallest. Then, the first node in $\boldsymbol{C}_1$ is selected as the critical node, and next the second critical node in $\boldsymbol{C}_1$ is selected which is not adjacent to the first critical node and has the largest s_{i1}, and so on, until there are no nodes to be selected in $\boldsymbol{C}_1$. If the number of selected critical nodes is not yet satisfied, we proceed to consider the second set $\boldsymbol{C}_2$. When selecting critical nodes from $\boldsymbol{C}_2$, we only require that the selected nodes are not adjacent to the critical nodes already selected in $\boldsymbol{C}_2$. In other words, the critical nodes in each set $\boldsymbol{C}_j$ are not adjacent to each other, but the critical nodes in two sets $\boldsymbol{C}_i$ and $\boldsymbol{C}_j$ can be adjacent to each other. The above process is repeated until the given number of critical nodes is satisfied.

The pseudo-code for our algorithm is given as follows. Algorithm 1 demonstrates node classification based on the eigenvectors of the Laplace matrix, where steps 4–9 represent that calculating similarity between $\tilde{\boldsymbol{a}}_i'$ and $\boldsymbol{u}_j$ of every node, and classifying nodes by maximum similarity. After that, we can select a set of critical nodes by the constraint that nodes in the same class are not adjacent to each other while different classes of nodes can be adjacent to each other.

Algorithm 1 NC (G) — Node classification

Input: Graph $G = (V, E)$
Output: Classified nodes sets $\boldsymbol{C}$

1: Delete the nodes with $K = 1$ in G using K-core algorithm, and get the new graph $\boldsymbol{G}'$.
2: $\boldsymbol{L}' \leftarrow \boldsymbol{D}' - \boldsymbol{A}'$ and $\tilde{\boldsymbol{A}}' \leftarrow \boldsymbol{D}'^{-1} \boldsymbol{A}'$.
3: $\boldsymbol{C}_n \leftarrow \emptyset$ for $n = 0, 1, 2, \ldots, [V']$.
4: **for** each row vector $\tilde{\boldsymbol{a}}_i'$ of $\tilde{\boldsymbol{A}}'$ **do**
5: $\quad l_{\max} \leftarrow i$ and $s_{\max} \leftarrow 0$.
6: $\quad$ **for** each eigenvectors $\boldsymbol{u}_j$ of $\boldsymbol{L}'$ **do**
7: $\quad\quad$ Calculate the similarity s_{ij} of $\tilde{\boldsymbol{a}}_i'$ and eigenvectors $\boldsymbol{u}_j$.
8: $\quad\quad$ **if** $s_{ij} > s_{\max}$ **then**
9: $\quad\quad\quad l_{\max} \leftarrow j$ and $s_{\max} \leftarrow s_{ij}$
10: $\quad\quad$ **end if**
11: $\quad$ **end for**
12: $\quad \boldsymbol{C}_{l_{\max}} \leftarrow \boldsymbol{C}_{l_{\max}} \cup \{i\}$.
13: **end for**
14: **return** $\boldsymbol{C}$

4. Experimental results

To verify the effectiveness of the proposed algorithm (LA) in this paper, experiments are conducted on seven real networks and four synthetic networks generated by the LFR model[29] in this section, and their performance is evaluated in comparison with seven benchmark algorithms. To better demonstrate the comparative effects between these algorithms, we divided the experiments into two main groups: top-k strategy and dispersion strategy. The experiments are carried out on the SIR model, and the infection probability is uniformly set as $\alpha = \langle \omega \rangle / (\langle \omega^2 \rangle - \langle \omega \rangle)$, where $\langle \omega \rangle$ is the average degree of the network.

4.1. Benchmark algorithm

Seven benchmark algorithms are selected for comparison with the LA algorithm, namely, BC,[13] CC,[14,15] DC,[11] CR,[19] CI,[25] WVR,[27] and FINDER.[28] The details of the seven algorithms are described as follows.

4.1.1. Top-k strategy

Betweenness centrality (BC), which measures the importance of a node by whether the shortest paths all pass through it, and selects the top k nodes that have been passed through more often as the initial spreaders.

Closeness centrality (CC), which measures the importance of a node by the sum of its distances from other nodes in the network, and selects the top k nodes with smaller sum of distances as the initial spreaders.

Degree centrality (DC), which measures the importance of a node by the number of its neighbors, and selects the top $k$nodes with larger degree values as the initial spreaders.

Cycle ratio (CR), which defines a cycle number matrix in order to represent the cycle information of the network, by which a metric called cycle ratio can be calculated — quantifying the importance of a single node by measuring the extent to which a node participates in the shortest cycle associated with other nodes.

4.1.2. Dispersion strategy

Collective influence (CI), which selects the nodes through a scalable algorithm and believes low-degree nodes play a major broker role in the network, and despite being weakly connected, can be powerful influencers.

WVoteRank (WVR), which selects the nodes with the highest number of votes and not the neighboring nodes of the already selected nodes as the initial propagators in each round of voting by iteration.

FINDER (FINDER), which introduces a deep reinforcement learning framework that incorporates inductive graph representation learning to represent graph states and actions, and a deep Q' (quality of action score) network that combines reinforcement learning and deep neural networks to automatically learn the strategy that optimizes the objective to find critical nodes in a complex network.

4.2. Real networks

The seven real network datasets selected in this paper are C.elegans,[33] crime,[34] Email,[35] Yeast,[36] HI-II-14,[37] ca-HepPh,[38] and ca-AstroPh,[38] which are complex networks with nodes ranging from a few hundred to more than 10000. Among them, the C.elegans network is a neural network, in which nodes represent neurons in the nervous system of the worm, and edges represent connections between neurons. The crime network is a criminal network. The Email network is an email communication network at the University of Rovira, Spain, in which nodes represent users and edges indicate that users have sent emails to each other. The Yeast network and HI-II-14 network are protein–protein interaction networks. The Ca-HepPh network is a collaborative network, in which nodes are authors of published papers, and the edges represent that there are co-authored papers between two or more authors. The Ca-AstroPh network is a collaborative network, which covers scientific collaborations between authors of papers submitted to the astrophysics category. The basic topological properties of the seven real networks are shown in Table 1.

Table 1. Four basic topological properties of the real networks, including the number of nodes ($|V|$), the number of edges ($|E|$), the average degree of the nodes ($\langle\omega\rangle$), and the network clustering coefficient (c).

Network	$\|V\|$	$\|E\|$	$\langle\omega\rangle$	c
C.elegans	297	2148	14.464	0.292
Crime	829	1473	3.553	0.006
Email	1133	5451	9.622	0.220
Yeast	2375	11693	9.847	0.306
HI-II-14	4165	13087	6.284	0.44
Ca-HepPh	12008	118489	19.738	0.612
Ca-AstroPh	18771	198050	21.102	0.631

4.2.1. Top-k strategy

Firstly, we investigate the effect of changing the percentage of source influential spreaders p on the final affected scale $F(n)$. Another variable parameter λ is set to 1.2. As can be seen in Figs. 3(a)–3(f), on the C.elegans network, the LA algorithm outperforms the other four algorithms overall. In particular, when $p \in [0.03, 0.10]$, the LA algorithm always performs

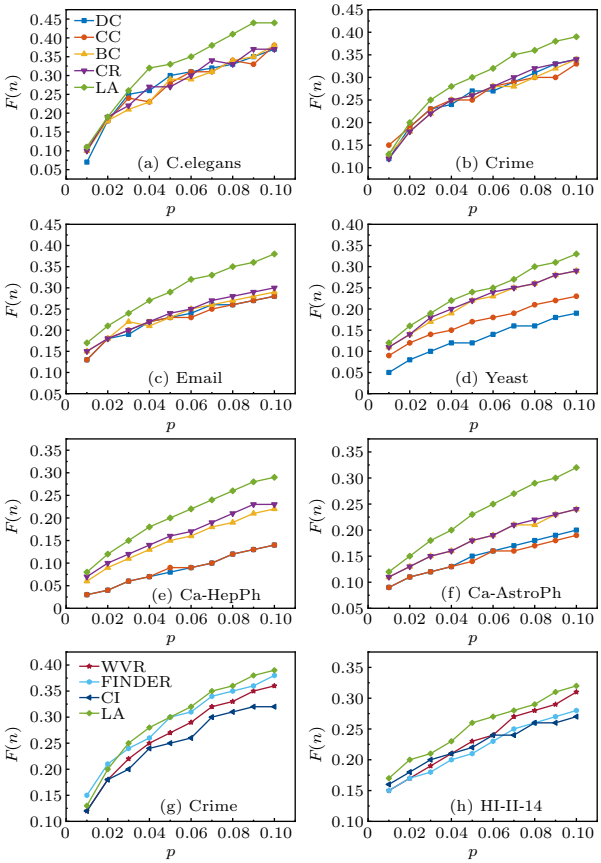


Fig. 3. Final affected scale $F(n)$ with different percentages of source influential spreaders p on seven real networks, where (a)–(f) show the effects of top-k strategy algorithms and (g) and (h) show the effects of dispersion strategy algorithms. Experiments are carried out on the SIR model, where $\lambda = 1.2$, $\alpha = \langle\omega\rangle/(\langle\omega^2\rangle - \langle\omega\rangle)$. Each data point is the average of 100 independent runs.

better than the benchmark algorithms. On the Email network, Yeast network, ca-HepPh network, and ca-AstroPh network, the LA algorithm always brings a better influence range than DC, BC, CC, and CR when $p \in [0.01, 0.10]$. In particular, on the crime network, the LA algorithm obtains an improvement of spreading influence ranging from 5.53% to 17.30% over the second best algorithm CR. Similarly, the LA algorithm outperforms the second best algorithm CR with an improvement of spreading influence ranging from 15.77% to 29.10% on the ca-HepPh network. Moreover, on the ca-AstroPh network, the LA algorithm obtains improvements of 25.09%, 16.71%, 52.29% and 54.49% compared with CR, BC, DC and CC, respectively.

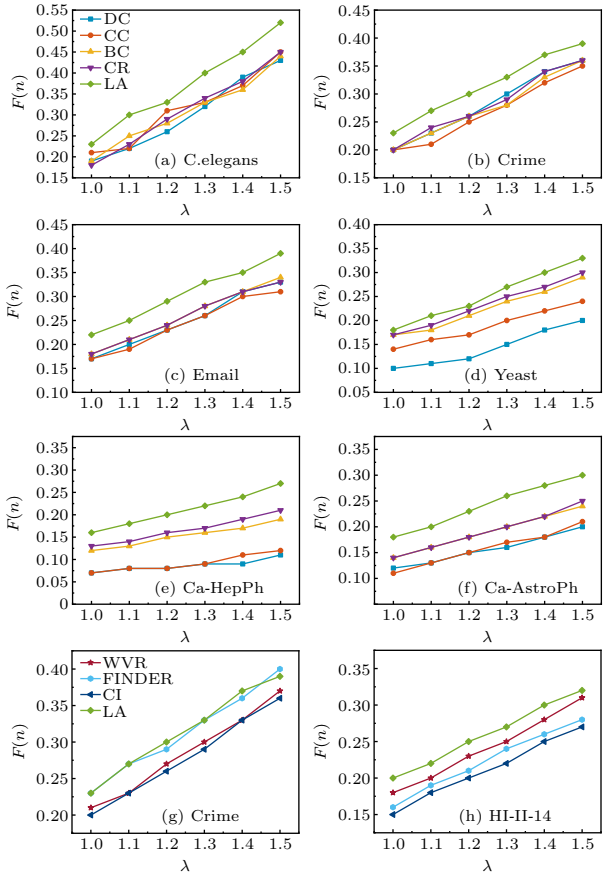


Fig. 4. Final affected scale $F(n)$ with different λ on six real networks, where (a)–(f) show the effects of top-k strategy algorithms and (g) and (h) show the effects of dispersion strategy algorithms. Experiments are carried out on the SIR model, where $p = 0.05$, $\alpha = \langle\omega\rangle/(\langle\omega^2\rangle - \langle\omega\rangle)$. Each data point is the average of 100 independent runs.

Secondly, we investigate the effect of changing λ on the final affected scale $F(n)$. In the experiment, the percentage of source influential spreaders is set to a fixed value of $p = 0.05$. As can be seen in Figs. 4(a)–4(f), it can be concluded that the LA algorithm performs better than the other four pure node centrality indicators algorithms on the C.elegans network, crime network, Yeast network, Email network, ca-HepPh network, and ca-AstroPh network. In particular, on the ca-AstroPh network, the LA algorithm obtains an improvement of spreading influence ranging from 23.61% to 28.56% over the second best algorithm CR.

4.2.2. Dispersion strategy

Firstly, we investigate the effect of changing the percentage of source influential spreaders p on the final affected scale $F(n)$. Another variable parameter λ is set to 1.2. As can be seen in Figs. 3(g)–3(h), LA algorithm always outperforms the other three dispersion strategy algorithms. On the HI-II-14 network, LA algorithm outperforms algorithm WVR with an improvement of spreading influence ranging from 3.26% to 13.04%, outperforms algorithm FINDER with an improvement of spreading influence ranging from 10.43% to 20.02%, and outperforms algorithm CI with an improvement of spreading influence ranging from 4.43% to 16.73%.

Secondly, we investigate the effect of changing λ on the final affected scale $F(n)$. In this experiment, the percentage of source influential spreaders is set to a fixed value of $p = 0.05$. As can be seen in Figs. 4(g)–4(h), the LA algorithm performs better than the dispersion strategy algorithms. In particular, on the HI-II-14 network, the LA algorithm obtains an improvement of spreading influence ranging from 6.18% to 14.51% over the second best algorithm WVR.

4.3. Synthetic networks

To verify the impact of community structure on our algorithm, we choose to compare it with the CR algorithm, since our algorithm tends to choose the local tree-like structure, while CR tends to choose the local cluster structure. The experiment is performed on four synthetic networks generated by the LFR model.[29] The LFR model accounts for the heterogeneity in the distributions of node degrees and of community sizes, which can be used to test the effect of community topology on algorithms. On the LFR model, both the degree and the community size distributions obey power laws, with exponents ε_1 and ε_2, respectively. In our experiments we generate the LFR networks we need by controlling a number of topologies, as follows: the four synthetic networks have the same ε_1 and ε_2, but different mixing parameters μ. The smaller the μ, the stronger the community structure.

The four networks have the same number of nodes and approximate number of edges. The experimental results are shown in Fig. 5. On the LFR1 network and LFR2 network with higher clustering coefficients, the LA algorithm outperforms the CR algorithm. In particular, on the LFR1 network, the LA algorithm obtains an improvement of spreading influence ranging from 4.87% to 7.86% over the CR algorithm. On the LFR2 network, the LA algorithm obtains an improvement of 6.35% on average. However, on the LFR3 and LFR4 networks with smaller clustering coefficients, there is no significant improvement or decrease in the final affected scale of

the LA algorithm, but for the CR algorithm, there is an improvement compared to that on the LFR1 and LFR2 networks. Hence, when the local clustering structure increases, it has no obvious influence on our algorithm.

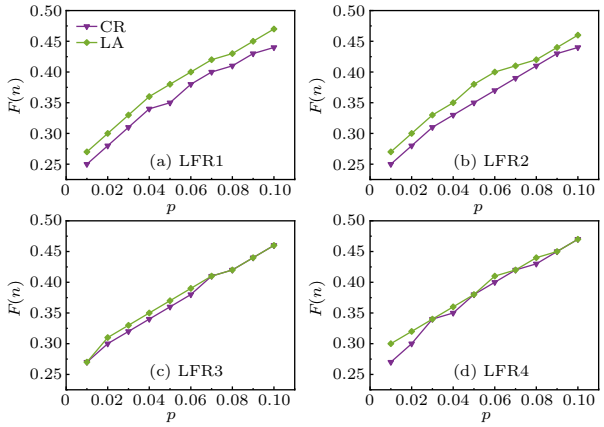


Fig. 5. Final affected scale $F(n)$ with different percentages of source influential spreaders p on the synthetic network, where $\lambda = 1.2$, $\alpha = \langle\omega\rangle/(\langle\omega^2\rangle - \langle\omega\rangle)$. Each data point is the average of 100 independent runs.

We can explain the above phenomenon from the following two aspects. First, from the basic topological data of the four synthetic networks in Table 2, we can see that the modularity (Q) of the LFR1 network and LFR2 network is higher than that of the LFR3 network and LFR4 network. For a network, the greater the modularity is, the stronger the community structure is, and the higher the local clustering coefficient. For the CR algorithm considering the shortest cycle ratio, the selected nodes are basically located in the core of the community. When the community structure is strong, the CR algorithm tends to cause local propagation within the community, while when the community structure is weak, the connected edges between communities increase, which increases the possibility of outward propagation, so the CR algorithm performs better on the LFR3 network and LFR4 network than on the LFR1 network and LFR2 network. Secondly, the nodes selected by the LA algorithm act as "bridges" and do not necessarily have many neighbors, but the neighbors of the selected nodes tend to have a very large degree and can propagate far away. Hence, on networks with a strong community structure, the LA algorithm tends to select the middle node of the local tree-like structure, which connects several communities and can expand the affected scale; on networks with weak community structure, the possibility of forming a local tree-like structure between nodes and nodes or nodes and communities increases, so the LA algorithm also chooses "bridge" nodes to expand the affected scale. In short, the LA algorithm always selects the "bridge" nodes of the local tree-like structure regardless of the community structure, and the affected scale is not influenced. The CR algorithm tends to select the core nodes in the communities, which results in a strong community structure network and reduces the affected scale.

Table 2. Eight basic topological properties of the synthetic network, including the number of nodes ($|V|$), number of edges ($|E|$), average degree ($\langle\omega\rangle$), node power-law exponent (ε_1), community power-law exponent (ε_2), mixing parameter (μ), modularity (Q), and network clustering coefficient (c).

Net	$\|V\|$	$\|E\|$	$\langle\omega\rangle$	ε_1	ε_2	μ	Q	c
LFR1	8000	68383	17.096	2.5	1.5	0.1	0.7378	0.1837
LFR2	8000	66235	16.559	2.5	1.5	0.2	0.6040	0.1106
LFR3	8000	71563	17.891	2.5	1.5	0.3	0.4252	0.0963
LFR4	8000	73167	18.292	2.5	1.5	0.4	0.3225	0.0588

5. Conclusion

In this paper, we propose a new algorithm for identifying multiple influential spreaders that balances the top-k strategy and the dispersion strategy. To validate the performance of the LA algorithm, we evaluate it on seven real networks and four synthetic networks for comparison with seven benchmark algorithms. On the real networks, we observe the effect on the final affected scale by adjusting p and λ. From the above, we know that adjusting $p \in [0.01, 0.10]$, the LA algorithm shows better performance on the C.elegans, crime, Email, Yeast, ca-HepPh and ca-AstroPh networks; when adjusting $\lambda \in [1.0, 1.5]$, similarly, the LA algorithm outperforms the other seven benchmark algorithms. On the LFR networks, we observe the effect on the algorithm performance by generating networks with different community structures. This paper mainly shows the comparison between the LA algorithm and the CR algorithm. The experimental results show that the LA algorithm always selects the "bridge" nodes of the local tree-like structure, and outperforms other benchmark algorithms.

The LA algorithm can improve the final affected scale to some extent. However, finding multiple influential spreaders is still an open research issue. In this paper, we focus on the effects of tree and cluster structures on the LA algorithm, while the way other network topologies, such as star and ring structures, affect the LA algorithm is still a far-reaching topic to be studied.

Acknowledgments

Project supported by the National Natural Science Foundation of China (Grant No. 62176217), the Program from the Sichuan Provincial Science and Technology, China (Grant No. 2018RZ0081), and the Fundamental Research Funds of China West Normal University (Grant No. 17E063).

References

[1] Domingos P and Richardson M 2001 *Mining the Network Value of Customers* (San Francisco), p. 57
[2] Hébert-Dufresne L, Allard A, Young J and Dubé L J 2013 *Sci. Rep.* **3** 2171
[3] Carrington P, John S and Stan W 2005 *Models and Methods in Social Network Analysis* (Cambridge: Cambridge University Press)

[4] Richardson M and Domingos P 2002 *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, July, 2002, Edmonton, Alberta, Canada, p. 61
[5] Kitsak Maksim, Gallos L K, Havlin S, Liljeros F, Muchnik L, Stanley H E and Makse H A 2010 *Nat. Phys.* **6** 888
[6] Lü L, Chen D, Ren X, Zhang Q, Zhang Y and Zhou T 2016 *Phys. Rep.* **650** 1
[7] Morone F and Makse H A 2015 *Nature* **524** 65
[8] Pei S, Wang J, Morone F and Makse H A 2019 *Complex Networks* **8** 2051
[9] Costa G S and Ferreira S C 2020 *Phys. Rev.* **101** 022311
[10] Bovet A and Makse H A 2019 *Nat. Commun.* **10** 1
[11] Bonacich P 1972 *Math. Soc.* **2** 113
[12] Chen D, Lü L, Shang M, Zhang Y and Zhou T 2012 *Physica A* **391** 1777
[13] Freeman L C 1978 *Social Networks* **1** 215
[14] Sabidussi Gert 1966 *Psychometrika* **31** 581
[15] Dangalchev C 2006 *Physica A* **365** 556
[16] Chen D, Gao H, Lü L and Zhou T 2013 *PLoS ONE* **8** 1
[17] Brin S and Page L 1998 *Computer Networks and ISDN Systems* **30** 107
[18] Lü L, Zhang Y, Yeung C H and Zhou T 2011 *PLoS ONE* **6** 1
[19] Fan T, Lü L, Shi D and Zhou T 2021 *Commun. Phys.* **4** 272
[20] Colizza V, Flammini A, Serrano M A and Vespignani A 2006 *Nat. Phys.* **2** 110
[21] Kempe D, Kleinberg J and Tardos É 2003 *Proceedings of the* 9*th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August, 2003, Washington, DC, USA, p. 137
[22] Chen W, Wang Y and Yang S 2009 *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, June, 2009, Paris, France, p. 199
[23] Zhao X, Huang B, Tang M, Zhang H and Chen D 2014 *Europhys. Lett.* **108** 68005
[24] Guo L, Lin J, Guo Q and Liu J 2016 *Phys. Lett. A* **380** 837
[25] Morone F and Makse H 2015 *Nature* **524** 65
[26] Zhang J, Chen D, Dong Q and Zhao Z 2016 *Sci. Rep.* **6** 1
[27] Sun H, Chen D, He J and Ch'ng E 2019 *Physics A* **519** 303
[28] Fan C, Zeng L, Sun Y and Liu Y 2020 *Nat. Mach. Intell.* **2** 317
[29] Lancichinetti A, Fortunato S and Radicchi F 2009 *Phys. Rev. E* **80** 1
[30] Hethcote H W 2000 *SIAM Rev.* **42** 599
[31] Yang R, Wang B, Ren J, Bai W, Shi Z, Wang W and Zhou T 2007 *Phys. Lett. A* **364** 189
[32] Kong Y, Shi G, Wu R and Zhang Y 2019 *Physics Reports* **832** 1
[33] Ryan R and Nesreen A 2015 *Proceedings of the* 29*th AAAI Conference on Artificial Intelligence*, January 25–30, 2015, Austin, Texas USA, p. 4292
[34] https://doi.org/10.24433/CO.3005605.v1
[35] Guimerà R, Danon L, Díaz-Guilera A, Giralt F and Arenas A 2003 *Phys. Rev. E* **68** 065103
[36] Jeong H, Mason S P, Barabási A L and Oltvai Z N 2001 *Nature* **411** 41
[37] http://interactome.dfci.harvard.edu/H_sapiens/
[38] Leskovec J, Kleinberg J and Faloutsos C 2007 *Graph Evolution: Densification and Shrinking Diameters*, March, 2007, New York, USA, 1