Heliyon 9 (2023) e16112

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

Influential nodes identification method based on adaptive adjustment of voting ability

Guan Wang [a,b], Syazwina Binti Alias [b], Zejun Sun [a,*], Feifei Wang [a], Aiwan Fan [a], Haifeng Hu [a]

[a] School of Information Engineering, Pingdingshan University, Pingdingshan, Henan, China
[b] Faculty of Engineering, Built Environment & Information Technology, SEGI University, Malaysia

ARTICLE INFO

Keywords:
Complex network
Influential node
Voting ability
Adaptive adjustment

ABSTRACT

Influential nodes identification technology is one of the important topics which has been widely applied to logistics node location, social information dissemination, transportation network carrying, biological virus dissemination, power network anti-destruction, etc. At present, a large number of influential nodes identification methods have been studied, but the algorithms that are simple to execute, have high accuracy and can be better applied to real networks are still the focus of research. Therefore, due to the advantages of simple to execute in voting mechanism, a novel algorithm based on adaptive adjustment of voting ability (AAVA) to identify the influential nodes is presented by considering the local attributes of node and the voting contribution of its neighbor nodes, to solve the problem of low accuracy and discrimination of the existing algorithms. This proposed algorithm uses the similarity between the voting node and the voted node to dynamically adjust its voting ability without setting any parameters, so that a node can contribute different voting abilities to different neighbor nodes. To verify the performance of AAVA algorithm, the running results of 13 algorithms are analyzed and compared on 10 different networks with the SIR model as a reference. The experimental results show that the influential nodes identified by AAVA have high consistency with SIR model in Top-10 nodes and Kendall correlation, and have better infection effect of the network. Therefore, it is proved that AAV algorithm has high accuracy and effectiveness, and can be applied to real complex networks of different types and sizes.

1. Introduction

The influential nodes identification technology has attracted extensive attention recently. It is critical for the study of functional characteristics and practical application of a network [1,2]. The transmission, control, security, invulnerability and aggressiveness of the network can be studied by identifying the influential nodes [3–5]. At present, influential nodes identification research has played a role that cannot be ignored in the construction of emergency logistics networks, social network communication, transportation network bearings, biological virus network prevention and control and power network anti-destruction [6–11]. In logistics network, a group of influential communication nodes are selected as the influential nodes of emergency guarantees, which can ensure the fast and efficient transportation of emergency materials [12]. In real social networks, selecting influential communication nodes can speed up

* Corresponding author.
E-mail address: szj@pdsu.edu.cn (Z. Sun).

https://doi.org/10.1016/j.heliyon.2023.e16112
Received 1 November 2022; Received in revised form 30 April 2023; Accepted 5 May 2023
Available online 10 May 2023
2405-8440/© 2023 Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

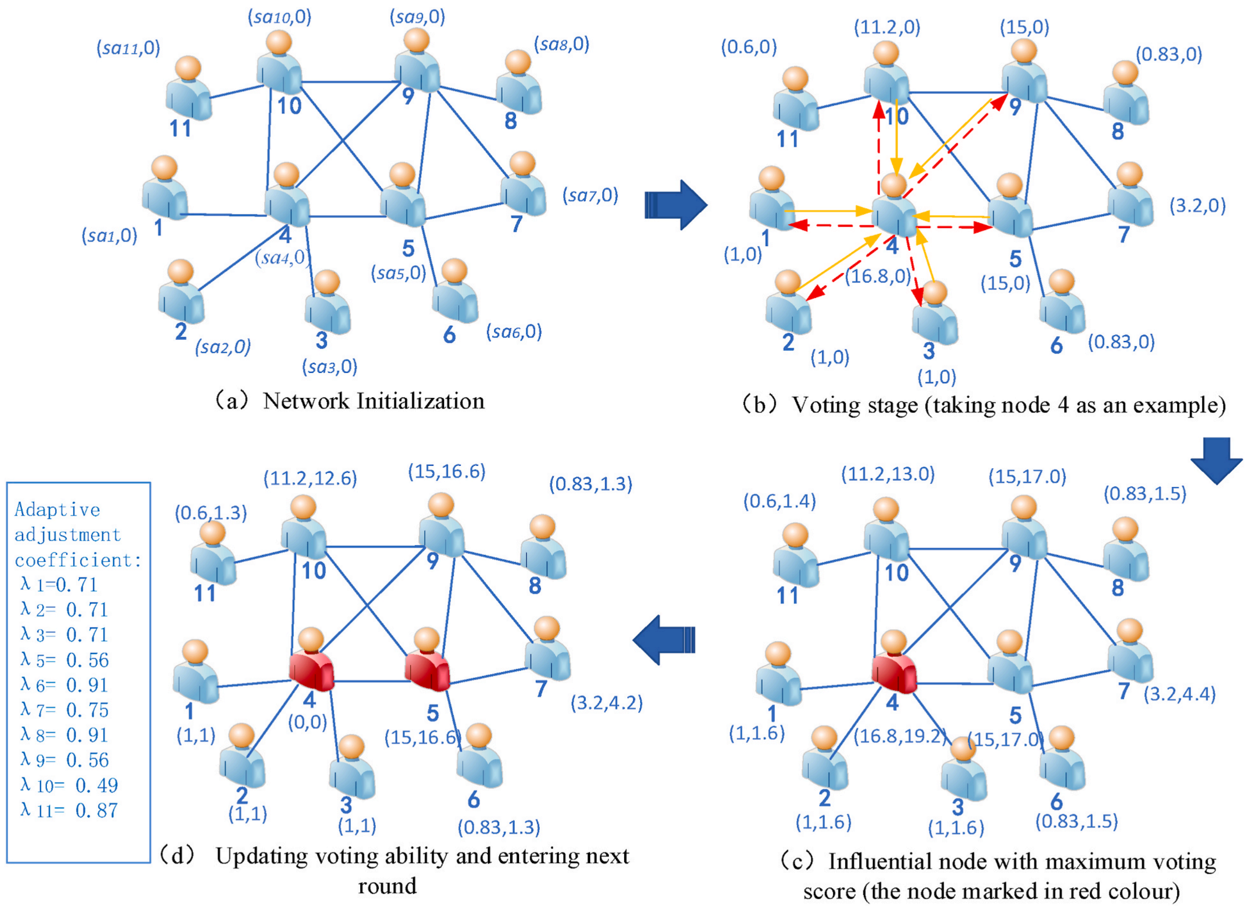


Fig. 1. Voting process diagram of AAVA algorithm.

the dissemination of information, quickly locate the key communicators in social networks, and control the spread of rumors [13,14]. In infrastructure carrying networks such as urban transportation networks, railway transport networks, aviation networks and communication networks, the network guarantee strategy can be provided for managers by identifying the critical degree of network nodes [15,16]. In biological virus prevention and control network, finding the key transmission nodes can effectively prevent and control disease transmission at the initial stage and effectively reduce the epidemic transmission capacity [17,18]. In power network, finding and protecting the key transmission line nodes in the power network improves the overall robustness and increases the invulnerability of the power network [19]. This implies that research on the identification of influential nodes can contribute to feasible decisions for the safe and efficient operation of real networks.

In the theoretical research of complex network influential nodes identification, various methods are proposed and can be mainly divided as following.

(1) Neighbor-based sorting method. The algorithm identifies the influential nodes by evaluating the number of its neighbors. Its main algorithms include degree centrality [20], semilocal centrality [21] and so on;
(2) Path-based sorting method. The algorithm sorts the information flow through the shortest path by examining the control ability of the nodes to the information flow in the network. Its main algorithms include proximity centrality [22], betweeness centrality [23] and so on;
(3) Node position-based sorting method. The algorithm measures the overall network structure and judges the criticality of nodes according to their positions. Its main algorithms include the K-shell decomposition method [24] and so on;
(4) Feature vector-based sorting method. The algorithm regards all nodes as equally important and only considers the neighbors number and their positions to judge the importance. It measures the number and the influence of the neighbor nodes. Its main algorithms include PageRank [25], eigenvector centrality [26], HITS algorithm [27] and so on.

According to the research on the latest algorithms, a voting method come from the idea of real voting in social network to identify the influential nodes [28,29,30,31,32]. Compared with nonvoting algorithm, voting algorithms are easy to perform, and the voting results can effectively describe the degree of nodes. In addition, the voting ability of neighbor nodes is reduced after voting to balance the distribution of influential nodes and to reduce the overlap of influence areas around influential nodes. Thus, it can effectively achieve maximum information dissemination through the selection of several influential nodes. Accordingly, a novel identification method based on adaptive adjustment of voting ability (AAVA) is proposed by aiming at the problems of the low discrimination and accuracy in the recent algorithms. AAVA algorithm can effectively distinguish the influence of node by measuring the self-influence of its local attributes and the voting contribution of neighbor nodes. It uses the similarity between the voting node and the voted node to dynamically adjust its voting ability without setting any parameters, so that the node can contribute different voting abilities to its neighbor nodes. A toy network is taken as an example, which describes the basic idea and voting process of AAVA algorithm. The algorithm is executed in rounds, as shown in Fig. 1.

As shown in Fig. 1(a), after network initialization, each node calculates the self-influence sa_v based on its local attribute, and the voting score is 0 and obtains (sa_v, 0). After voting starts, as shown in Fig. 1(b), each node calculates the contribution of voting ability for its neighbor nodes. When node 4 receives the voting ability from its neighbors, it also calculates the voting ability for different neighbors and votes for them. After the voting is completed, each node calculates its own voting score according to its self-influence and the voting ability of all neighbors, then, select the node with the maximum voting score as the influential node, as shown in Fig. 1 (c), node 4 was successfully selected as the influential node in this round because it has the maximum voting score. After the first round of voting, as shown in Fig. 1 (d), the self-influence and voting score of node 4 are both 0, and node 4 will no longer participate in node voting. At the same time, the neighbor nodes within two hops of node 4 adjust their voting ability by calculating the adaptive adjustment coefficient, and then update the voting score to enter the next round of influential node voting. For the detailed calculation, process of the algorithm, see section 3.

2. Relater works

In this section, we summarize typical voting algorithms in recent years and analyze their shortcomings. In the recent years, a variety of new representative algorithms have appeared based on research on the above-mentioned classical algorithms, for example: the algorithms based on the idea of voting method [28,29,33,34], gravity model [30,35], node propagation ability [36,37], information entropy [31,38,39] and other method are proposed. In 2016, the VoteRank algorithm [28] is proposed by introducing the voting mechanism, and the node calculates the voting score by obtaining the neighbor's votes and then finally selecting the influential node with the maximum score. In VoteRank, the voting ability of node has the same value, which is equal to 1. It's simple to design and easy to implement and takes into account the voting ability of the neighbor nodes. However, it has a low discrimination for nodes, and the coefficient of voting ability adjustment is fixed and cannot be adjusted automatically. In 2019, Sun Hong-liang et al. proposed a voting method in weighted networks-WVoteRank [33]. It increases the consideration of the neighbor number and link weights for the voting score. After voting, the voting ability of neighboring nodes in one hop and two hops of selected node are weakened uniformly. In the WVoteRank algorithm, the weakening coefficient is the same in the whole network, which is related to the reciprocal of the average degree in network. In 2020, Sanjay Kumar et al. proposed a neighbor core value-based voting algorithm, NCVoteRank [29]. In 2021, Liu Peng-feng et al. proposed VoteRank Plus algorithm [34], which considers the different proximities between nodes and the fact that nodes may vote differently from their neighbors. When voting is completed, the algorithm weakens the voting ability within two-hop neighbors to distribute the influential nodes properly, but it still needs to improve the discrimination of influential nodes. In 2022, Ramya D. Shetty proposed global structure influence-GSI algorithm [35], which consider the global and local attributes of nodes by using COVID-19 as a case. In 2019, Shuang Xuet al. Proposed SpectralRank (SR)algorithm [36], which measures the nodes' propagation capability by considering the nature of spread dynamics. In 2020, Guo et al. proposed a votinng method based on Information Entropy-EnRenew algorithm [39], which uses the information entropy formula to obtain the voting score, the probability in information entropy is calculated by taking the proportion of the degree value of the neighbor nodes. In 2017, Fan Yang et al. proposed an ELKSS algorithm, which extends the local K-shell and centrality [32]. This algorithm considers the K-shell values of neighbor nodes and extends the local K-shell sum of the neighbor nodes. In 2019, Zhongjing Yu et al. proposed a ProfitLeader (PL) algorithm [40], which calculates the profitability of nodes to describe the importance of nodes. In 2020, Zhao Jie et al. measured the global importance (GIN) of each node based on a quantitative model [41]. The significance of the node in GIN is calculated by own importance and connected node, which takes into account the node degree and distance to neighbor node.

In summary, the above mentioned voting and non-voting algorithms can effectively identify the influence nodes, based on the advantages and disadvantages of the above algorithm, the research motivations of this paper are as follows.

(1) Explore an algorithm with high accuracy and discrimination of node importance recognition. By using the advantages of simple and easy execution of the voting algorithm, the influence of the node itself and the voting contribution of the neighbor nodes are considered comprehensively.
(2) In combination with the fact that nodes in the real complex network have different voting preferences for different neighbor nodes, the relationship between nodes is reasonably utilized to achieve different voting abilities when voting for different neighbor nodes.
(3) In order to balance the distribution of influential nodes in the whole network, after the voting, nodes should adjust their voting ability according to the situation of the selected influential nodes of their neighbors, and realize the dynamic update of voting ability and avoid setting parameters.

In this paper, we propose a novel voting method AAVA algorithm by combining the above problems and considering the effectiveness and complexity of the algorithm. The specific description is shown in Section III. The main contributions include the following.

(1) The proposed AAVA algorithm integrates the self-influence of nodes and the voting contribution of neighbors. When calculating the voting score, the voting contribution of neighbors is not only measured, but also the self-influence of node is comprehensively evaluated by integrating their local attributes, which can solve the shortcoming of the voting algorithms of coarse-graining and improve the accuracy of ranking results.
(2) The voting contribution of neighbor is determined by its voting ability, which integrates the degree value and the voting probability between nodes, through the use of the normalized similarity coefficient to present the voting probability for neighbor nodes, which can effectively reflect the different contributions for different neighbor nodes.
(3) After the voting process, the similarity relationship between selected node and its neighbor nodes within two-hop is used to realize the adaptive adjustment of the voting ability of neighbor nodes, without setting or adjusting any parameters, so as to increase the applicability of the algorithm in complex networks.
(4) Through the comparison of experimental results, the AAVA algorithm is superior to several classical routing algorithms and the recent new algorithms in accuracy, effectiveness and the results of the top-10 nodes.

The remaining content is organized: Section 3 describes AAVA algorithm in detail, focusing on the idea of the algorithm and the voting process of the specific work; Section 4 carries out simulation experiments on different datasets between AAVA algorithm and several classical algorithms and the voting algorithms, then analyzes and discusses the results; Section 5 concludes the main contributions of this research.

3. The proposed method

A method based on adaptive adjustment of voting ability is proposed, the importance of a node is represented by its voting score, which is composed of two aspects. The first aspect is the self-influence of the node, each node calculates the neighbors number and degrees according to its local attribute. The larger the number and degrees of neighbors, and the more powerful the self-influence of the node. The second aspect is the voting contribution of neighbor nodes. Meanwhile, the proposed algorithm introduces into the Jaccard similarity [40] between the voting node and voted node to realize the contribution of voting ability can be adjusted adaptively for each neighbor. After voting, the adaptive adjustment coefficient is calculated to reduce the voting ability of the two-hop neighbors of the selected node. Therefore, AAVA algorithm considers the self-influence of nodes to solve the problem of low differentiation of ranking values, and it can make use of the similarity to realize different voting probabilities for different neighbor nodes in the voting process, meanwhile, the neighbor nodes of the elected node can realize the adaptive adjustment of their voting ability by using the similarity, which avoids the problem that the probability of important neighbor nodes being elected in the next round is too small, so as to ensure the accuracy and effectiveness of the ranking results. The definitions and an example of AAVA algorithm are made in this section.

3.1. Preliminaries

Assume an undirected and powerless network $G=(V,E)$, and determine the tuple of node v as (sa_v, vs_v), where sa_v represents the self-influence and vs_v represents the voting score of node v. $\Gamma(v)$ is defined as the neighbors set of v, node u is the neighbor of node v.

Definition 1 (Self-influence): $DC(u)$ and DC_{max} represent the degree and maximum degree value in network, respectively. The self-influence of a node sa_v is closely related to its own local attribute, which is calculated as:

$$sav = |\Gamma(v)| * \frac{\sum_{u\in\Gamma(v)} DC(u)}{DC\max} \tag{1}$$

Definition 2 (Similarity): There has a certain tendency between nodes when voting, the tendency between nodes is calculated by the Jaccard similarity, which is calculated as:

$$J(u,v) = \frac{|\Gamma(u)\cap\Gamma(v)|}{|\Gamma(u)\cup\Gamma(v)|} \tag{2}$$

Definition 3 (Voting probability): The voting probability of each node voting for different neighbor nodes is different, and the calculation of voting probability is obtained by similarity. The voting probability $vp\,(u, v)$ of node u for node v is defined as the ratio of $J\,(u, v)$ to the sum of the similarities of node u and its neighbors $\Gamma(u)$. The more similar node v is to its neighbor u, the greater the voting probability of node u for node v. The voting probability that node u will vote for node v is calculated as:

$$vp(u,v) = \frac{J(u,v)}{\sum_{\omega\in\Gamma(u)} J(u,\omega)} \tag{3}$$

Definition 4 (Contribution of voting ability): The voting contribution of neighbor node is related to its voting ability. The contribution of voting ability of node u is calculated by its local attribute and the voting probability for node v:

$$va(u,v) = \frac{\sum_{u\in\Gamma(v)} DC(u)}{DC\max} * vp(u,v) \tag{4}$$

Definition 5 (Voting score): The voting score is related to its self-influence and the contribution of voting ability of all neighbor nodes, which is calculated as:

$$vsv = sav + \sqrt{\sum_{u \in \Gamma(v)} va(u, v)} \tag{5}$$

Definition 6 (Adjustment of voting ability): Node v is defined as the influential node. Node u is defined as the neighbor within two-hop of node v. The adaptive adjustment coefficient of voting ability of node u is λ_u, which mainly calculates the Jaccard similarity J (u, v) according to definition 2. The voting ability of node u for its neighbor w is adjusted as:

$$\begin{cases} \mathrm{va}(\mathrm{u}, \mathrm{w}) = \lambda \mathrm{u} * \mathrm{va}(\mathrm{u}, \mathrm{w}) \\ \lambda \mathrm{u} = 1 - \mathrm{J}(\mathrm{u}, \mathrm{v}) \end{cases} \tag{6}$$

3.2. Process description of AAVA algorithm

AAVA algorithm measures the self-influence and the voting ability of neighbor nodes in voting process comprehensively. AAVA algorithm consists four stages in its execution.

(1) Initialization stage. Initialize the self-influence and voting score of each nodes. The initial values of sa_v and vs_v in node v (sa_v, vs_v) are (0,0).
(2) Voting stage. When entering the voting election, node v integrates the similarity and the voting probability to calculate the voting ability va (v,u) for its neighbor node u. Then, the node that has the maximum voting score vs_v is selected as the influential node.
(3) Voting ability update stage. The self-influence and voting score of the selected node v are updated to 0, and the neighbor nodes within the two hops of node v will adjust and update the voting ability adaptively.
(4) Iterative cycle stage. After the voting ability update is completed, the network will enter a new round of voting. Only one influential node was selected in each round, the iteration of the algorithm stops until the preset number of influential nodes is reached.

The implementation process of AAVA.

Algorithm AAVA
Input: G=(V, E);
Output:
Influence values of all ranked nodes;
calculate degree value DC of all nodes;
calculate the maximum degree $DC_{\max}$;
for each node v in the set V **do**
use Formula (1) to calculate self-influence sa_v;
for each node u in the neighbor set Γ(v) **do**
calculate degree value $DC(u)$ of node u;
use Formula (2) to calculate similarity between nodes v and u;
use Formula (3) to calculate voting probability vp (u,v);
use Formula (4) to calculate voting ability va (u,v);
endfor
use Formula (5) to calculate voting score vs_v;
endfor
select *maximum* (vs_s) and add the corresponding node s to rank;
while (rank$\leq|V|$)
for each node u in the neighbor set $\Gamma(s)$ of the selected node s **do**
for each node w in the neighbor set of $\Gamma(u)$;
$X = \Gamma(s) \cup \Gamma(u)$;
endfor
endfor
for each node v in the set X **do**
use Formula (6) to adjust and update the voting ability;
for each node u in the neighbor set Γ(v) **do**
use Formula (5) to calculate voting score vs_u;
endfor
endfor
select *maximum* (vs_s) and add the corresponding node s to rank;
endwhile
return rank

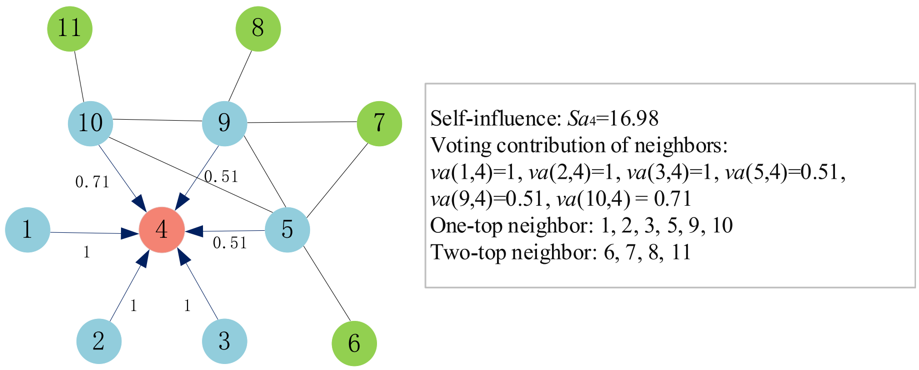


Fig. 2. Take node 4 as an example in the whole network.

Table 1
Node ranking results and values of three algorithms and SIR.

VoteRank	VoteRank value	VoteRank Plus	VoteRank Plus value	AAVA	AAVA value	SIR	SIR value
4	6	4	2.506	4	19.176	4	1.733
5	3.214	5	0.845	5	16.562	5	1.686
9	1.429	9	0.379	9	16.278	9	1.635
1	0	10	0.140	10	11.988	10	1.534
2	0	1	0	7	3.333	7	1.301
3	0	2	0	1	1.0	1	1.177
6	0	3	0	2	1.0	3	1.169
7	0	6	0	3	1.0	6	1.168
8	0	7	0	6	0.833	11	1.16
10	0	8	0	8	0.833	2	1.143
11	0	11	0	11	0.667	8	1.128

3.3. Complexity analysis

After network initialization, in the first cycle, the self-influence, voting probability, voting ability and score are calculated according to Eqs. (1)–(5), and complexity is $O(n<k>)$, where n represents the nodes number and $<k>$ represents the average degree. Then, the voting ability and voting score of nodes are updated through the second cycle, in which the first inner cycle calculates the set X of one-hop and two-hop neighbor nodes of the selected node, and the complexity is $O(<k>^2)$. In the second inner cycle, the voting ability and voting score is updated according to Eqs. (6) and (5), and the complexity is $O(x<k>^2)$, where $x \leq 2<k>$. Finally, the influential nodes are elected, and the complexity is $O(2n<k>^2)$. Thus, the time complexity of AAVA algorithm is $O(n<k>^2+2n(x<k>^2)$, which is equivalent to $O(n<k>^2)$.

3.4. Example of AAVA algorithm

Node 4 is taken as an example in Fig. 1. The neighbors of node 4 have 6 nodes, which are node 1, 2, 3, 5, 9 and 10, and as shown in Fig. 2.

The detailed process of AAVA is calculated as.

(1) Calculate the self-influence: in Fig. 1(a), after the network is initialized, nodes calculate self-influence according to Eq. (1), and the voting score is 0. They are (sa_1,vs_1)=(1,0), (sa_2,vs_2)=(1,0), (sa_3,vs_3)=(1,0), (sa_4,vs_4)= (16.98,0), (sa_5,vs_5)= (15,0), (sa_9,vs_9)= (15,0), and (sa_{10},vs_{10})=(11.2,0).
(2) Calculate the voting contribution of neighbors: in Fig. 1 (b), according to Eqs. (2) and (3), the 6 neighbor nodes calculated the probability of voting for node 4 as follows: $vp(1,4) = 1$, $vp(2,4) = 1$, $vp(3,4) = 1$, $vp(5,4) = 0.17$, $vp(9,4) = 0.17$, $vp(10,4) = 0.25$. The voting ability of each neighbor for node 4 can be obtained according to Eq. (4), that is, $va(1,4) = 1$, $va(2,4) = 1$, $va(3,4) = 1$, $va(5,4) = 0.51$, $va(9,4) = 0.51$, $va(10,4) = 0.71$.
(3) Calculate the voting scores: in Fig. 1(c), combined with the self-influence of node 4 and the voting contribution of its neighbors, its voting score vs_4 is calculated according to Eq. (5), namely:

$$vs4 = 16.98 + \sqrt{1 + 1 + 1 + 0.51 + 0.51 + 0.71} \approx 19.2$$

Combined with all the voting score, node 4 is the highest in the toy network, so node 4 is selected as the influential node successfully.

Table 2
The topological statistics of ten real networks.

DataSets	$d_{average}$	$d_{\max}$	$<CC>$	$\|V\|$	$\|E\|$
Karate	4.588	17	0.588	34	78
Dolphins	5.129	12	0.303	62	159
Football	10.6	12	0.403	115	616
E-mail	9.622	71	0.254	1133	5451
Euroroad	2.414	5	0.02	1174	1417
Friendships	13.492	272	0.167	1858	12,534
Hamster	13.31	273	0.231	2426	16,631
Facebook	2.064	769	0.803	2888	2981
Powergrid	2.669	19	0.107	4941	6594
Ca-astroph	21.102	236	0.677	18,771	198,050

(4) Adjust the voting ability of selected node within two hops: in Fig. 1(d), both the voting ability and voting score of node 4 are set to 0, and no longer join in the subsequent voting. As one-hop neighbors of node 4, nodes 1, 2, 3, 5, 9, and 10 weaken their voting ability by Eq. (6) and the adaptive adjustment coefficient are $\lambda_1 = 0.71$, $\lambda_2 = 0.71$, $\lambda_3 = 0.71$, $\lambda_5 = 0.56$, $\lambda_9 = 0.56$ and $\lambda_{10} = 0.49$, respectively. At the same time, as two-hop neighbors of node 4, nodes 6, 7, 8, and 11 also weaken their voting ability according to Eq. (6), which are updated as $\lambda_6 = 0.91$, $\lambda_7 = 0.75$, $\lambda_8 = 0.91$ and $\lambda_{11} = 0.87$, respectively. After that, the network moves into the next round of voting.

After 11 rounds of execution of the network in Fig. 1, according to AAVA algorithm process, the ranking of the critical degree of the network nodes and the voting score are obtained according to the voting score of each round. The two compared algorithms of VoteRank and VoteRank Plus are running to get the value of voting score of each node. Table 1 shows the ranking results and voting score statistics of the nodes of AAVA algorithm, VoteRank algorithm and VoteRank Plus algorithm, as well as the ranking of node influence and weight values in the SIR model. In particular, among the voting score of nodes 1, 6, 7 and 11, from the red and blue data in Table 1.

We have color-coded the nodes whose ranking results are consistent with SIR model, from Table 1 we can see that VoteRank algorithm has 3 nodes in the same ranking order as SIR model, and both AAVA and VoteRank Plus algorithm has 6 nodes in the same order. Meanwhile, we can obtain that the VoteRank and VoteRank Plus algorithms cannot effectively distinguish their degree because the voting score of these nodes is 0, the ranking values in VoteRank algorithm are 0 from the 4th node to the 11th node, and the ranking values in VoteRank Plus algorithm are 0 from the 5th node to the 11th node. However, the ranking values of AAVA are clearly differentiated, that because AAVA algorithm not only considers the voting of neighbor nodes, but also increases the judgment of the self-influence of node, it can effectively avoid the situation where the node voting score is 0. Therefore, AAVA algorithm can better solve the problem of low differentiation of ranking values while maintaining accurate ranking results of nodes.

4. Experimental evaluation

We select ten representative real networks in the public website to compare AAVA algorithm with the voting algorithms and classical algorithms, including DC [20], CC [22], BC [23], K-shell [24], PR [25], EC [26] and the new algorithms, including VoteRank [28], VoteRank Plus [34], GSI [35], ELKSS [32], PL [40] and GIN [41]. All trials in this experiment are running Python on a Windows 10 operating system. The hardware configuration is 8G RAM, Intel Core i7 processor, and 256G SSD hard disk.

4.1. Data description

Ten representative networks are selected to verify the accuracy and effectiveness of AAVA in this experiment. **Karate**: A social network represents the friendship of American karate club. **Dolphin**: A social network represents the interactive relationship among bottlenose dolphins. **Football**: A real social network based on the American Football League. **Email**: A communication network represents the mail exchanges among users in a Spain university. **Euroroad**: A European electronic road network structure. **Friendships**: A social network built on the basis of friendships on website. **Hamster**: A social network built on the user relationships on Hamster. **Facebook**: A social network based on the friendship between users on Facebook. **Powergrid**: A grid network based on grid equipment and power supply lines in US. **Ca-astroph**: A collaborative network based on the scientific collaborative relationship between the authors of the astrophysics category. The topological statistics of the ten real networks are shown in Table 2.

In Table 2, $d_{average}$ represents the average degree, $d_{\max}$ represents the maximum degree, $<CC>$ represents the average clustering coefficient, $|V|$ represents the nodes number, $|E|$ represents the edges number.

4.2. Evaluation indicators

In the simulation experiment, the SIR propagation model [42] is used to evaluate the performance of AAVA algorithm and the comparison algorithms. In SIR model, the susceptible node (S) refers to the uninfected node, which is easily infected by the infected node (I); the recovery node (R) refers to the infected node that has recovered and is no longer infected. Initially, a specified number of

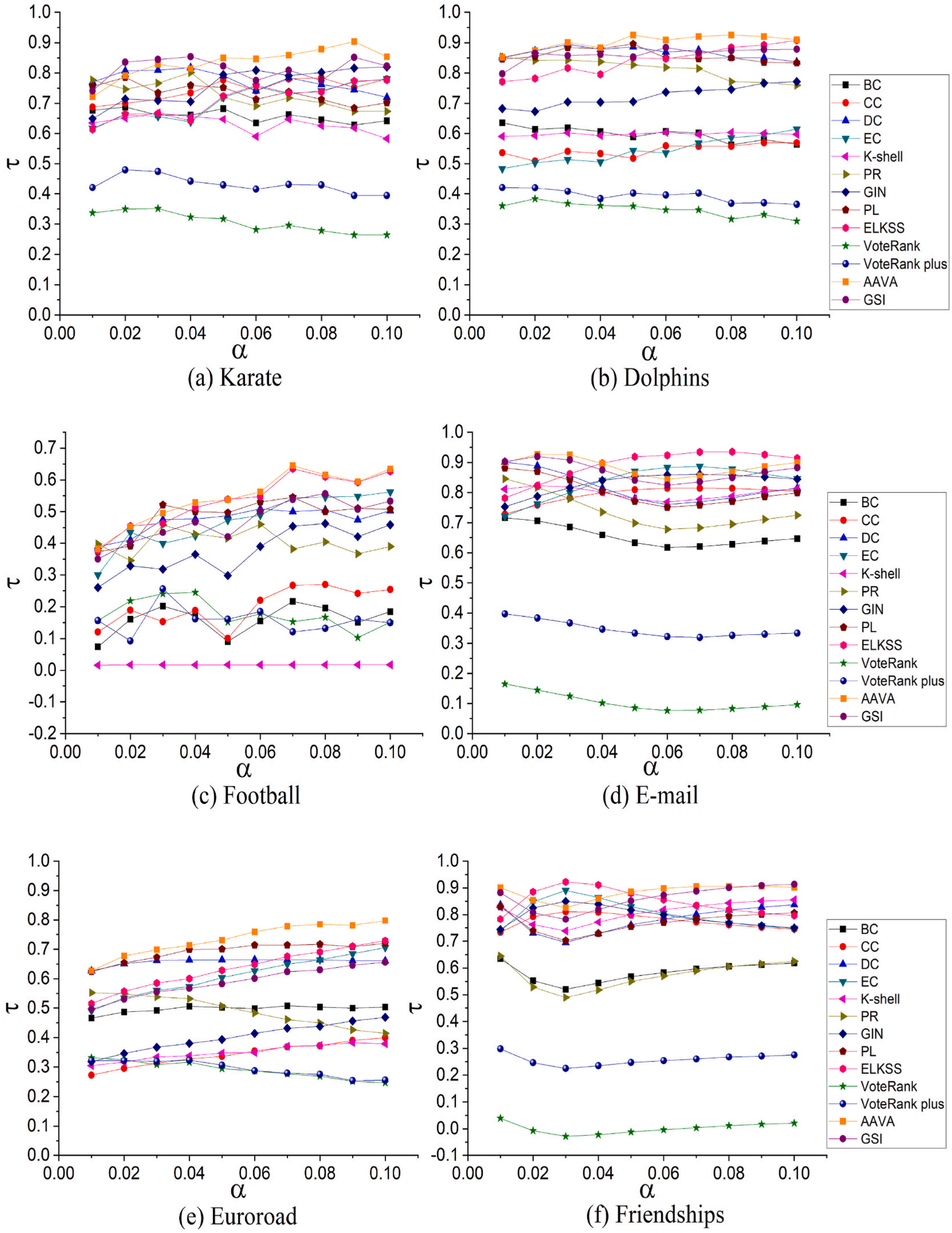


Fig. 3. The Kendall τ values of 13 algorithms on 10 networks are compared with the infection probability α = [0.01, 0.1].

infected nodes are selected, at each time step, infected node I infects the S node with probability α, and each node I transmits to the recovery node R with probability β. The influence of each node is ranked by calculating the infected nodes number.

Based on the SIR Model, firstly, we will calculate the correlation of Kendall coefficient [43] and the top-10 nodes to study the effect of node sorting results, so as to verify the accuracy of the algorithms. Secondly, we will calculate the total infection nodes number F(t) of all nodes and the top-10 nodes at time t [28,34], so as to analyze the infection ability in the whole network and verify the effectiveness of the algorithms.

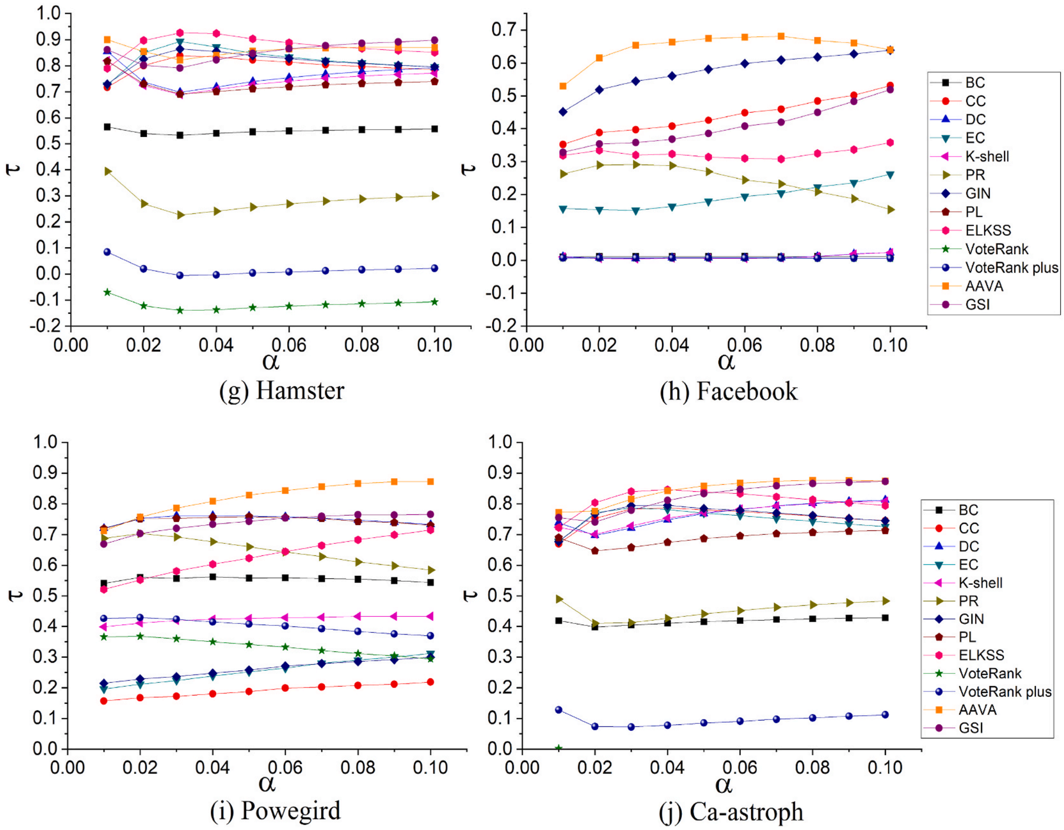


Fig. 3. (*continued*).

(1) Kendall τ

Based on SIR model, we selected different infection probabilities α to evaluate the effectiveness of AAVA algorithm, the range of α is set to [0.01–0.1]. Fig. 3 shows the value line of Kendall τ between AAVA and other 11 compared algorithms on the ten representative networks with different infection probabilities.

The Kendall τ value in Fig. 3 to verify the accuracy of the algorithms. Clearly, the effect of AAVA algorithm is better in each kind of network, and the overall effect is the best in the Karate (Fig. 3(a)), Dolphins (Fig. 3(b)), Football (Fig. 3(c)), Euroroad (Fig. 3(e)), Facebook (Fig. 3(h)) and Powergrid (Fig. 3(i)) networks. Although the compared result of AAVA algorithm isn't the best on the E-mail (Fig. 3(d)) and Hamster (Fig. 3(g)) networks, the value of Kendall τ of AAVA algorithm is still the highest at some probabilities. In the E-mail network, the effect of AAVA algorithm is the highest when the infection probability is between 0.01 and 0.04. In Hamster network, AAVA algorithm is higher than the other comparison algorithms from 0.08. In the Friendship (Fig. 3(f)) and Ca-astroph (Fig. 3(j)) networks, the Kendall value of AAVA algorithm is on the rise since the infection probability is 0.03 and is higher than that of the other comparison algorithms from 0.05. Among all the comparison algorithms, VoteRank and VoteRank Plus algorithms using the voting method, we can see that the Kendall value is negative in Friendships and Hamster networks and in a low level of the other eight networks. This is due to the VoteRank and VoteRank Plus algorithms suppress the voting ability of the neighbor nodes of the elected node, the important neighbor nodes also will have a small election probability. Therefore, they are not good at sorting all the nodes across the network. In contrast, AAVA algorithm improves the differentiation of voting scores because it measures the self-influence of nodes and the voting ability of its neighbors, so it can better identify the influential nodes and rank them accurately.

(2) Propagation capability

According to the running results in the algorithm, a node sequence is sorted by the influential values from high to low. Based on SIR, the corresponding average value of the infected nodes in SIR is assigned to influential values of the node sequence, and then the corresponding value of the node is drawn into a curve according to the order of the node sequence. This curve represents the propagation capability of each node under the SIR model, the ideal running result of the curve should show a steady downward trend, the smoother the curve of the results is, the more consistent sequence with SIR, and the more efficient the algorithm is. In this experiment, the infection and recovery probability is 0.1 and 1, respectively. Among the 10 networks, the number of iterations is set to 1000 except the Ca-astroph network, because Ca-astroph is a large scale network and should run a long time, therefore, its iterations of simulation is

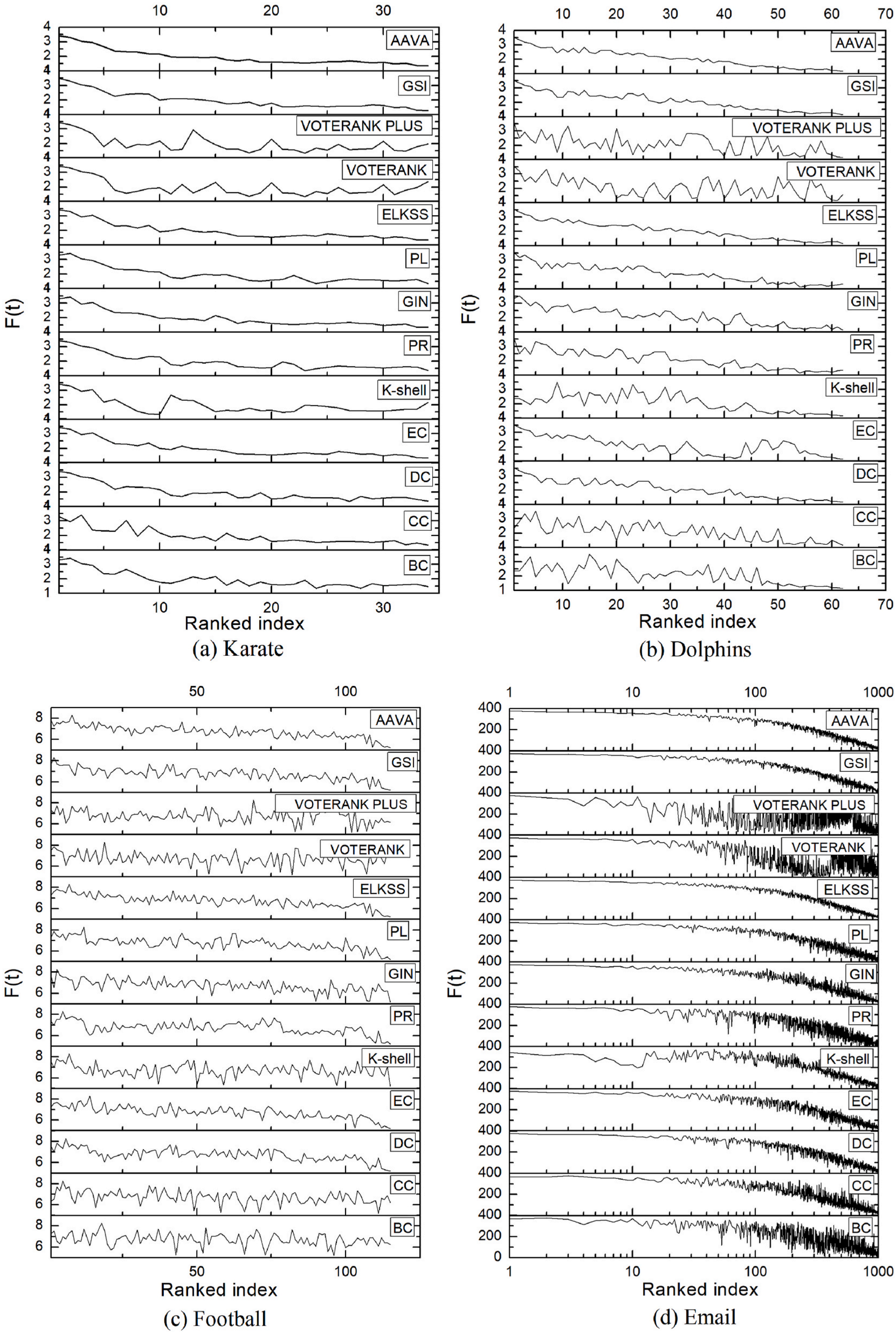
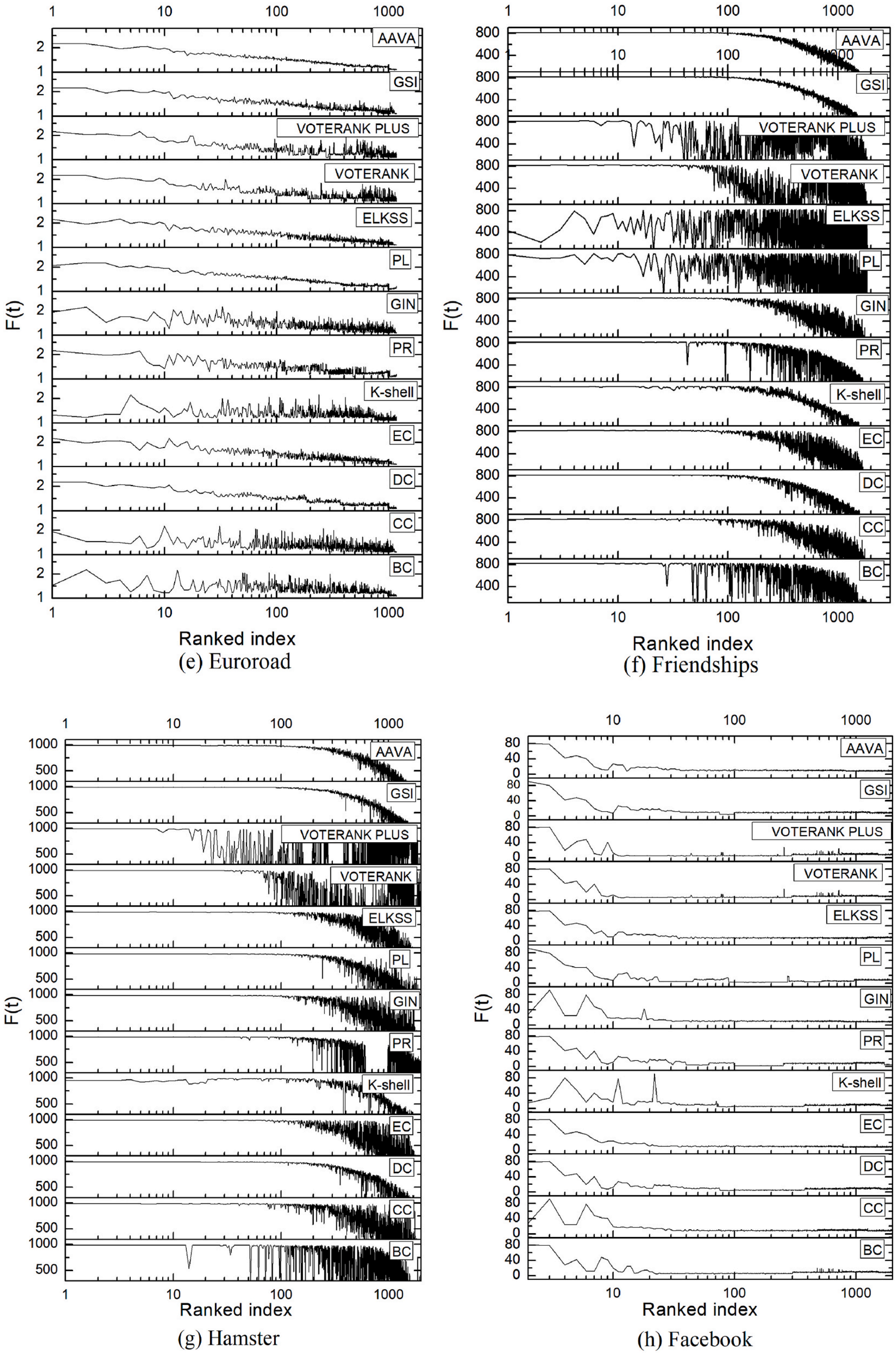


Fig. 4. The propagation capability of 13 algorithms on 10 networks are compared based on SIR model. F(t) represents the values of propagation capability.



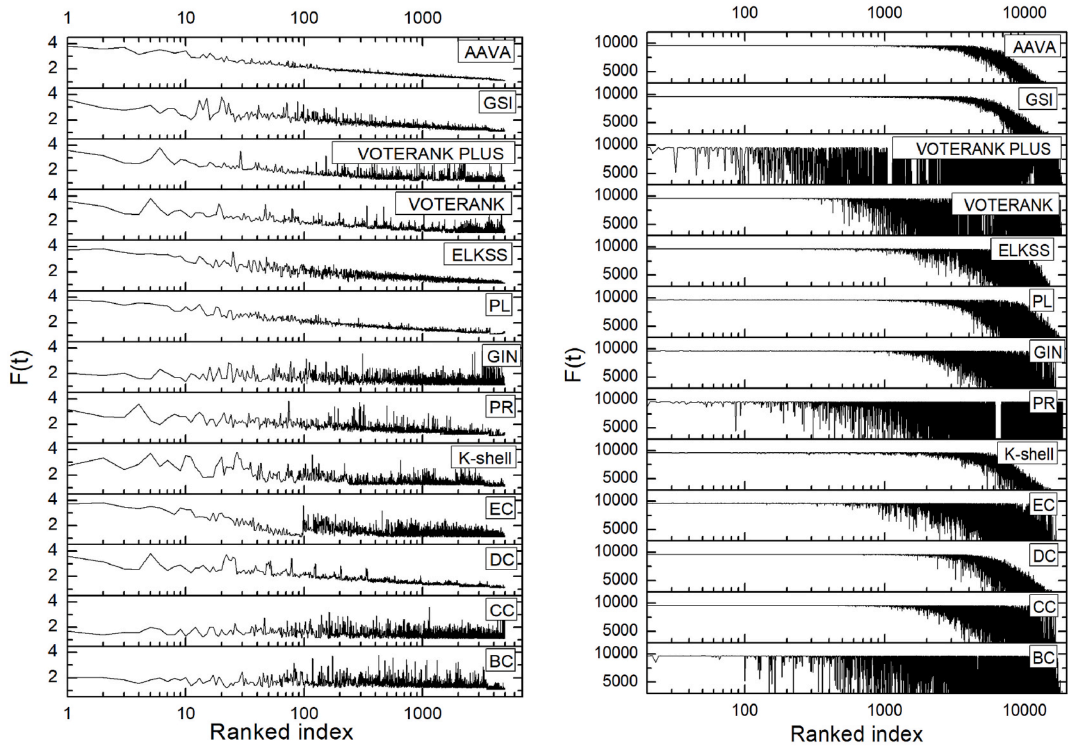
(e) Euroroad

(f) Friendships

(g) Hamster

(h) Facebook

Fig. 4. (*continued*).



(i) Powergrid (j) Ca-astroph

Fig. 4. (*continued*).

Table 3
Top-10 nodes in Karate.

BC	DC	CC	EC	PR	Ks	VoteRank	ELKSS	PL	GIN	VoteRank Plus	GSI	AAVA	SIR
1	34	1	34	34	1	34	34	1	1	34	34	34	34
34	1	3	1	1	2	1	1	34	34	1	1	1	1
33	33	34	3	33	3	33	3	33	3	33	33	33	33
3	3	32	33	3	4	3	33	3	33	2	3	3	3
32	2	9	2	2	8	2	2	2	2	6	2	2	2
9	4	14	9	32	9	6	14	32	32	32	4	32	32
2	32	33	14	4	14	26	9	9	9	7	9	4	9
14	9	20	4	24	31	7	4	14	14	30	14	9	14
20	14	2	32	9	33	28	32	4	4	28	32	14	4
6	24	4	31	14	34	30	8	24	20	4	31	24	24
8	10	9	9	10	8	5	9	10	9	6	9	10	Num

set to 100.

Because the small nodes number of the Karate, Dolphins and Football networks, the result of F(t) will be displayed in linear form in these networks, while the result will be displayed in log10 form in the other 7 networks which have more nodes, to focus on the more influential nodes. As we can see from Fig. 4, the data change of AAVA algorithm in the Karate (Fig. 4(a)), Dolphins (Fig. 4(b)), Euroroad (Fig. 4(e)), Friendships (Fig. 4(f)), Hamster (Fig. 4(g)), Powergrid (Fig. 4(i)) and Ca-astroph (Fig. 4(j)) networks is small, the curve of AAVA algorithm has the smoothest downward trend and fewer quivering burrs, so its effect is the best. In the E-mail network (Fig. 4 (d)), the infection effect of AAVA algorithm has no obvious advantage compared with the ELKSS algorithm, although the curve of AAVA algorithm has a few burrs, the amplitude of burrs are not very dramatic, the infection effect of AAVA algorithm is obviously better than others. In the Football network, due to the large difference in the value of nodes, the curve change of all algorithms has little difference. In the Football network (Fig. 4(c)), the infection effects of the AAVA and PL algorithms are the best because they have smoother curves and fewer burrs, thus the effects of these two algorithms are better than those of the other algorithms. In the Facebook network (Fig. 4(h)), the infection effect of the EC algorithm is the best. Because the degree value of nodes are particularly uneven, so the amount of infected nodes is large and fluctuates greatly. But in AAVA algorithm, the amplitude of burrs are smaller than that of other algorithms. Therefore, the curves of AAVA algorithm are smoother and have fewer quivering burrs in most networks and can well infect all nodes in the network.

Table 4
Top-10 nodes in Euroroad.

BC	DC	CC	EC	PR	Ks	VoteRank	ELKSS	PL	GIN	VoteRank Plus	GSI	AAVA	SIR
402	284	401	7	284	2	284	7	107	401	284	284	7	284
284	7	402	43	137	3	7	43	284	7	137	7	284	7
277	39	403	499	236	17	39	107	7	402	107	401	107	107
453	137	432	107	39	4	137	284	39	411	39	137	43	499
452	107	1019	181	107	855	107	181	499	453	236	107	39	137
403	236	253	454	7	6	236	499	43	454	7	236	137	39
401	43	452	39	204	7	141	39	137	232	141	43	499	181
404	141	404	180	768	880	401	401	181	43	401	499	181	236
837	499	232	411	164	8	499	137	236	253	81	181	401	43
836	181	284	8	587	22	43	236	141	400	265	141	236	401
2	8	1	6	6	0	9	10	9	3	6	8	10	Num

Table 5
Top-10 nodes in Powergrid.

BC	DC	CC	EC	PR	Ks	VoteRank	ELKSS	PL	GIN	VoteRank Plus	GSI	AAVA	SIR
651	2847	1378	4422	602	4422	2847	4422	4436	2781	2847	2847	4436	4436
559	602	1678	4436	932	4415	602	4436	4422	2685	602	558	2847	4422
1365	932	2944	4419	3411	4452	932	4452	4453	559	3930	556	4422	2847
2824	3411	1377	4417	2847	4453	3411	4453	2847	2944	3411	2926	602	4452
2685	4436	2781	4452	1210	4427	4436	4419	4452	2824	2926	602	4434	4419
1324	558	1365	4453	691	4428	558	4417	4434	2956	4436	2866	4452	4453
1378	2287	1368	4427	2287	4451	2287	4434	4417	651	558	2543	4453	4434
1213	2926	1380	4421	2865	4454	2926	4427	2926	1378	2287	2783	4438	4417
433	2865	2685	4434	2554	4417	2865	4438	2543	2782	2543	2719	4417	4438
2781	3930	2795	4438	3930	4418	3930	4421	4438	1678	2865	2956	4419	4437
0	2	0	8	1	4	2	8	8	0	2	1	9	Num

(3) Top-10 nodes

The sorted results of the top-10 influential nodes of 13 algorithms and SIR model are compared, so as to evaluate the accuracy and efficiency of AAVA algorithm. The infection and recovery probability is 0.1 and 1. The top-10 important nodes of the 13 algorithms on 10 networks are sorted in descending order. Only Karate, Euroroad and Powergrid as representative networks are shown as follows, the colored data in each algorithm represent the node numbers that are the same as the top-10 nodes in the SIR Model. The last row in Tables 3–5 counts the same top-10 nodes number of each algorithm with SIR model, which are represented by underscores.

In Tables 3 and it is obviously shows that the Top-10 influential nodes selected by the 13 algorithms have little difference because the Karate network structure is relatively simple. However, from the point of view of position ranking, the set of 10 nodes selected by AAVA, PL, PR, and DC is the same as that of SIR, but the order of the first six nodes of AAVA and PR is the same as that of SIR, while that of PL is only the first five nodes. However, Voterank and Voterank Plus algorithms only have 5 and 6 nodes respectively, which are the same as SIR model. Generally, there is little difference between the other comparison algorithm due to Karate is a small network. Therefore, it is concluded that the AAVA method works best.

Table 4 shows the comparison under the medium-sized Euroroad network. According to the selected nodes, the top-10 nodes are exactly the same as those selected by SIR in the ELKSS and AAVA algorithms, however, the top-10 nodes of Ks and GSI algorithms are completely different from SIR model. From the location ordering of nodes, the first two nodes of VoteRank and DC are the same as SIR, but the location order of the first three nodes of AAVA algorithm is highly consistent with SIR in the comprehensive comparison. This proves that the AAVA method has better performance.

Table 5 shows the comparison under medium and large Powergrid networks. According to the selected nodes, the results of the 13 algorithms are quite different because of the complexity of the network structure. The top-10 nodes selected by AAVA are highly consistent with those selected by SIR, and the PL algorithm with the best effect in the comparison algorithm has 8 nodes, however, the top-10 nodes of BC, CC, GIN algorithms are completely different from SIR model. In addition, the location ranking of the nodes indicates that the location order of the first three nodes of AAVA algorithm is highly consistent with SIR, and the selected nodes and their locations are sorted comprehensively. In general, the AAVA method has better performance.

(4) Infection capacity of the Top-10 nodes

The top-10 nodes of 13 algorithms are the seed nodes to infect other nodes, and the accuracy of the algorithm is verified by evaluating the number of infected nodes F(t). The infection and recovery probability are set to 0.01 and 1, respectively. After 30 rounds of independent operation, the average number of infected nodes is taken for 1000 iterations.

According to Fig. 5, the value of F(t) increases rapidly at the initial time in each network, then it increases with the number of

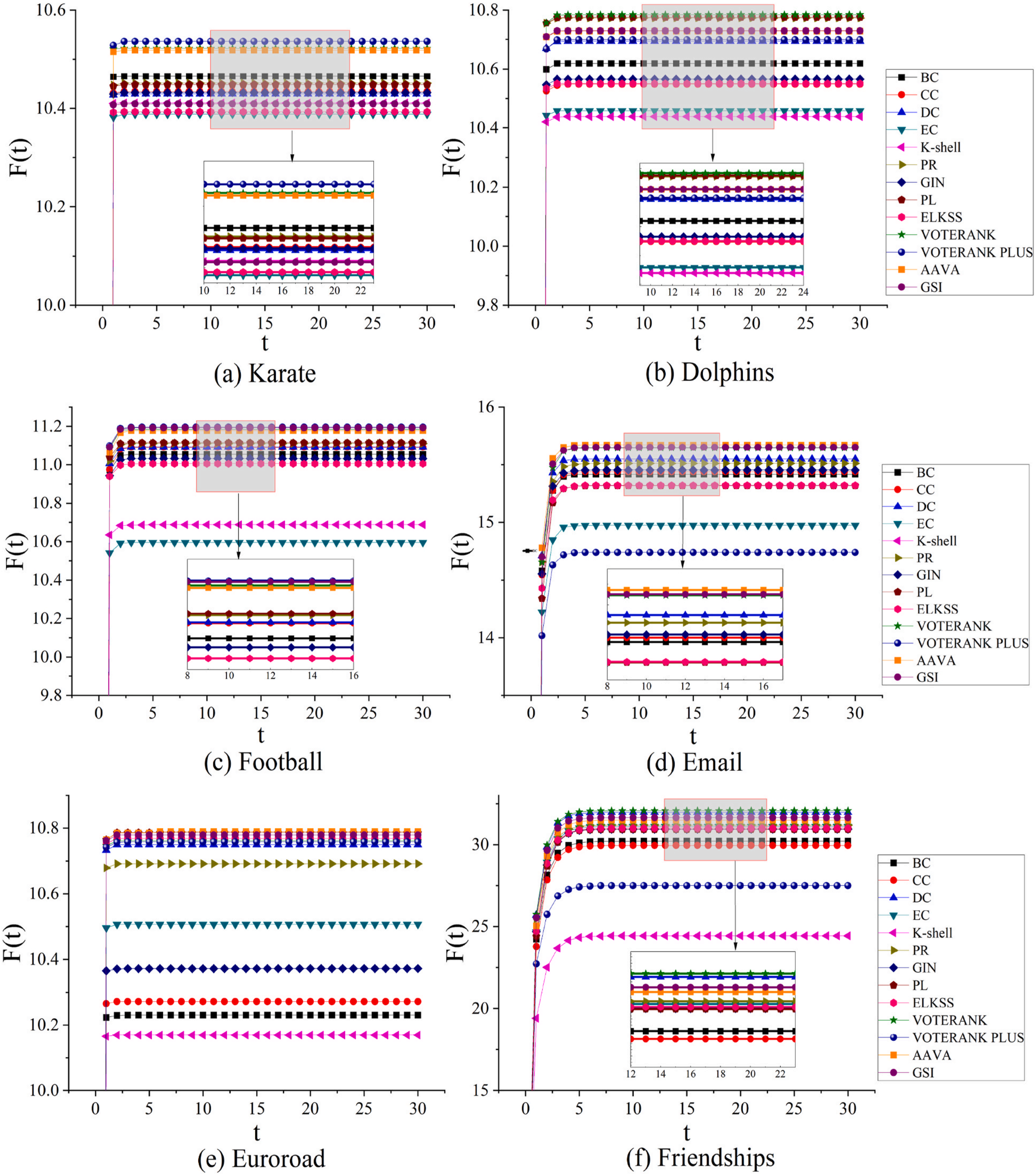


Fig. 5. The infection capacity of the top-10 nodes of 13 algorithms on 10 networks, F(t) represents the infected nodes number.

rounds; finally, it reaches a stable state after a certain period of infection. The comparison results show that the performance of the VoteRank, Voterank Plus and AAVA algorithms is better than that of the other algorithms in the Karate (Fig. 5(a)), Dolphins (Fig. 5(b)), Football (Fig. 5(c)), Euroroad (Fig. 5(e)), Facebook (Fig. 5(h)) and Powergrid (Fig. 5(i)) networks. This is because the voting method can effectively weaken the ability of neighbor nodes of selected influential node, which can make the elected influential nodes more dispersed and more conducive to diffusion between nodes. At the same time, the overall effect of AAVA algorithm is the best in the E-mail (Fig. 5(d)), Euroroad (Fig. 5(e)), Powergrid (Fig. 5(i)) and Ca-astroph (Fig. 5(j)) networks. The effect of AAVA algorithm is general

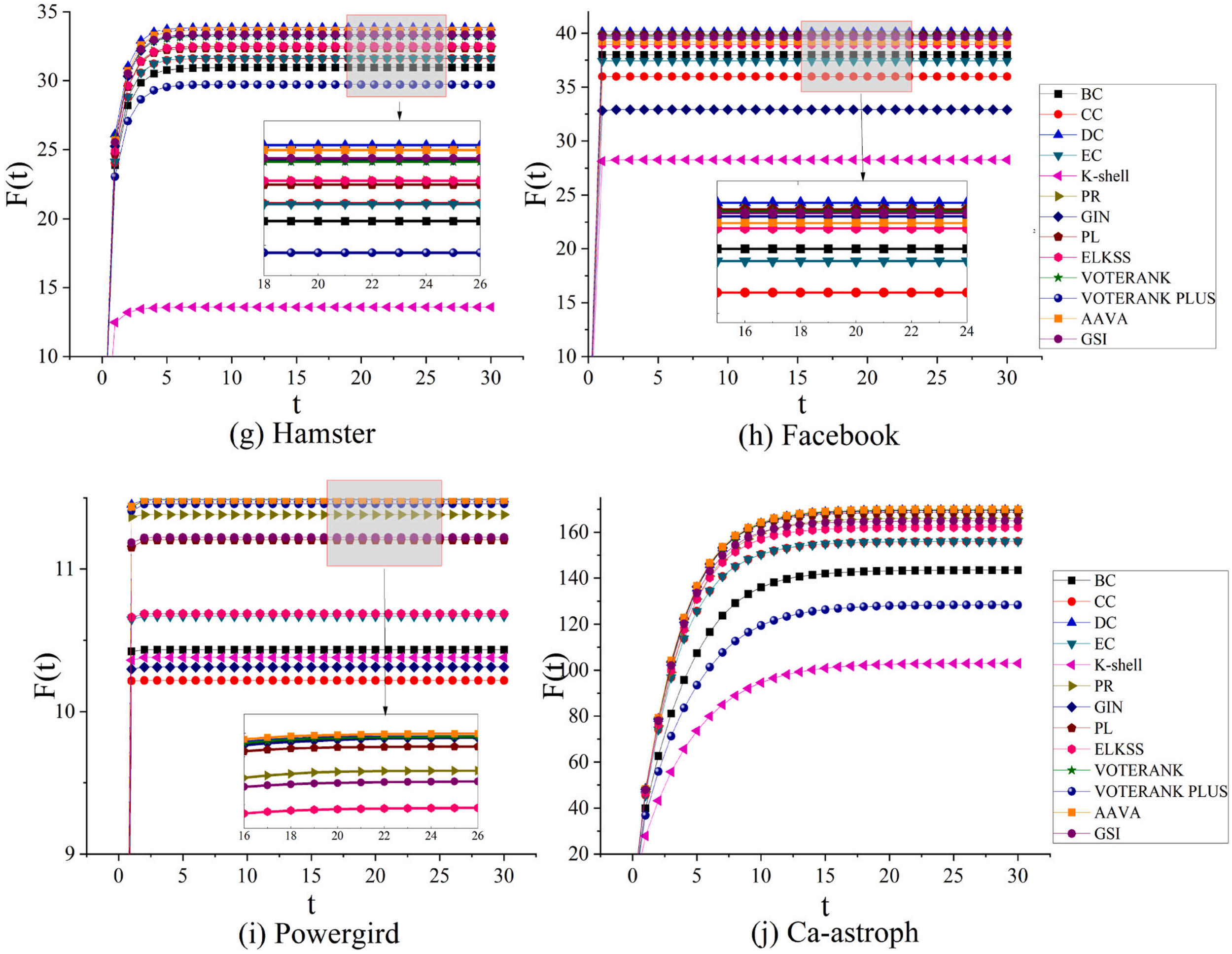


Fig. 5. (*continued*).

in other networks, but it is still at a high level. In the Facebook, Hamster (Fig. 5(g)) and Friendships (Fig. 5(f)) social networks, the effect of the DC algorithm performs better due to it sorts nodes according to their degree. Due to the voting ability of nodes can be adjusted adaptively in AAVA algorithm, so the distribution of influential nodes can be better balanced, and performs a better effect as a whole.

5. Conclusion

Identifying influential nodes has important application value for the research of invulnerability, security and propagation in complex networks. In this paper, a novel voting method based on adaptive adjustment of voting ability is proposed. At the beginning of voting, we first use the local attributes of node to measure the self-influence. Secondly, in the voting process and the voting end stage, the node adaptively adjusts its voting ability when voting for different neighbors by using the similarity relationship between the nodes and its neighbors. Finally, the importance of the node is comprehensively evaluated by integrating its self-influence and the voting contribution of its neighbors. To verify the performance of AAVA algorithm, the running results is compared with the other 11 representative algorithms on 10 different types of networks, and taking SIR as a reference model, the Kendall value and the scale of node infection are analyzed and compared. After the analysis of the experimental data, it shows that AAVA algorithm can solve the problem of coarse-graining of node differentiation in traditional algorithms and is effectively suitable for the identification and ranking the influential nodes. The overall experimental results show that AAVA algorithm performs better and can be applied to most of the complex networks. However, the future work still presents great challenges, because the network in the real world is dynamically changing, therefore, further optimization of the algorithm should be carried out in future research to further improve the performance and applicability of the algorithm combined with the characteristics of real networks.

Author contribution statement

Guan Wang and Zejun Sun: Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Wrote the paper.

Syazwina Binti Alias: Conceived and designed the experiments; Wrote the paper.

Feifei Wang: Performed the experiments; Analyzed and interpreted the data.

Aiwan Fan and Haifeng Hu: Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data.

Data availability statement

Data associated with this study has been deposited at https://github.com/Crown0702/AAVA.

Declaration of competing interest

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

Acknowledgment

This work was supported by the Science and Technology Research Project of the Science and Technology Department of Henan Province under Grant (232102210099, 232102210041, 222102210129, 222102210160), the"14th Five-Year Plan" Project of Educational Science in Henan Province of China under Grant (2021YB0232), the Scientific Research Projects of Colleges and Universities in Henan Province of China under Grant (23A520051).

References

[1] J. Shao, K. Wang, J. Lu, et al., Learning Evolving Concepts with Online Class Posterior Probability, International Conference on Database Systems for Advanced Applications, Springer, Cham, 2022, pp. 639–647.
[2] Z. Sun, Y. Sun, X. Chang, Q. Wang, X. Yan, Z. Pan, Community detection based on the Matthew effect, Knowl. Base Syst. 205 (2020), 106256.
[3] K. Hajarathaiah, M.K. Enduri, S. Anamalamudi, Efficient algorithm for finding the influential nodes using local relative change of average shortest path, Phys. Stat. Mech. Appl. 591 (2021), 126708.
[4] F. Wang, Z. Sun, Q. Gan, A. Fan, H. Shi, H. Hu, Influential node identification by aggregating local structure information, Physica A 593 (2022), 126885.
[5] H. Hu, Z. Sun, F. Wang, L. Zhang, G. Wang, Exploring influential nodes using global and local information, Sci. Rep. 12 (2022), 22506.
[6] A. Ullah, B. Wang, J. Sheng, et al., A novel relevance-based information interaction model for community detection in complex networks, Expert Syst. Appl. 196 (2022), 116607.
[7] L. Meng, G. Xu, P. Yang, D. Tu, A novel potential edge weight method for identifying influential nodes in complex networks based on neighborhood and position, Journal of Computational Science 60 (2022), 101591.
[8] R. Chen, B.C. Fung, S.Y. Philip, B.C. Desai, Correlated network data publication via differential privacy, The VLDB Journal 23hy rumors spre 4 (2014) 653–676.
[9] M.M. Tulu, M.E. Mkiramweni, R. Hou, S. Feisso, T. Younas, Influential nodes selection to enhance data dissemination in mobile social networks: a survey, J. Netw. Comput. Appl. 169 (2020), 102768.
[10] J. Shao, Z. Zhang, Z. Yu, J. Wang, Y. Zhao, Q. Yang, Community Detection and Link Prediction via Cluster-Driven Low-Rank Matrix Completion, IJCAI, 2019, pp. 3382–3388.
[11] R. Pastor-Satorras, A. Vespignani, Epidemic spreading in scale-free networks, Phys. Rev. Lett. 86 (2001) 3200.
[12] A. Caunhye, S. Niex, Optimization models in emergency logistics:a literature review, Socieo-ecomomic Planning Sciences 46 (2012) 4–13.
[13] M. Li, J. Peng, s. Ju, et al., Efficient algorithms for finding diversified top-k structural hole spanners in social networks, Inf. Sci. 602 (2022) 236–258.
[14] G. Maji, A. Namtirtha, A. Dutta, M.C. Malta, Influential spreaders identification in complex networks with improved k-shell hybrid method, Expert Syst. Appl. 144 (2020), 113092.
[15] M. Zhang, T. Huang, Z. Guo, Z. He, Complex-network-based traffic network analysis and dynamics: a comprehensive review, Physica A 607 (2022), 128063.
[16] M. Jong, M. Rui, D. Stead, Y. Ma, B. Xi, Introducing public–private partnerships for metropolitan subways in China: what is the evidence, J. Transport Geogr. 18 (2) (2010) 301–313.
[17] S. Wen, J. Jiang, B. Liu, Y. Xiang, W. Zhou, Using epidemic betweenness to measure the influence of users in complex networks, J. Netw. Comput. Appl. 78 (2017) 288–299.
[18] P. Zhu, Q. Zhi, Y. Guo, Z. Wang, Analysis of epidemic spreading process in adaptive networks, IEEE Transactions on Circuits and Systems II: Express Briefs 66 (7) (2019) 1252–1256.
[19] J. Fang, J. Wu, Z. Zheng, K.T. Chi, Revealing structural and functional vulnerability of power grids to cascading failures, IEEE Journal on Emerging and Selected Topics in Circuits and Systems 11 (1) (2019) 133–143.
[20] P. Bonacich, Factoring and weighting approaches to status scores and clique identification, J. Math. Sociol. 2 (1) (1972) 113–120.
[21] D. Chen, L. Lu, M. Shang, Y. Zhang, T. Zhou, Identifying influential nodes in complex networks, Phys. Stat. Mech. Appl. 391 (4) (2012) 1777–1787.
[22] G. Sabidussi, The centrality index of a graph, Psychometrika 31 (4) (1966) 581–603.
[23] L.C. Freeman, Centrality in social networks conceptual clarification, Soc. Network. 1 (3) (1978) 215–239.
[24] M. Kitsak, L.K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H.E. Stanley, H.A. Makse, Identification of influential spreaders in complex networks, Nat. Phys. 6 (11) (2010) 888–893.
[25] S. Brin, L. Page, The anatomy of a large-scale hypertextual web search engine, Comput. Netw. ISDN Syst. 30 (1–7) (1998) 107–117.
[26] T. Wen, W. Jiang, Identifying influential nodes based on fuzzy local dimension in complex networks, Chaos, Solitons & Fractals 119 (2019) 332–342.
[27] J. Kleinberg, Authoritative sources in a hyperlinked environment, J. ACM 46 (5) (1999) 604–632.
[28] J. Zhang, D. Chen, Q. Dong, Z. Zhao, Identifying a set of influential spreaders in complex networks, Sci. Rep. 6 (1) (2016), 27823.
[29] S. Kumar, B. Panda, Identifying influential nodes in social networks: neighborhood coreness based voting approach, Physica A 553 (2020), 124215.
[30] A. Ullah, B. Wang, J. Sheng, J. Long, N. Khan, Z. Sun, Identification of nodes influence based on global structure model in complex networks, Sci. Rep. 11 (1) (2021) 1–11.
[31] H. Zhang, S. Zhong, Y. Deng, H.C. Kang, LFIC: identifying influential nodes in complex networks by local fuzzy information centrality, IEEE Trans. Fuzzy Syst. 30 (8) (2021) 3284–3296.

[32] F. Yang, R. Zhang, Z. Yang, R. Hu, M. Li, Y. Yuan, K. Li, Identifying the most influential spreaders in complex networks by an extended local K-shell sum, Int. J. Mod. Phys. C 28 (1) (2017), 1750014.
[33] H. Sun, D. Chen, J. He, E. Ch'ng, A voting approach to uncover multiple influential spreaders on weighted networks, Physica A 519 (2019) 303–312.
[34] P. Liu, L. Li, S. Fang, Y. Yao, Identifying influential nodes in social networks: a voting approach, Chaos, Solit. Fractals 152 (2021), 111309.
[35] D. Ramya, Shrutilipi Shetty, Bhattacharjee, GSI: an influential node detection approach in heterogeneous network using covid-19 as use case, IEEE Transactions on Computational Social Systems 9 (2022) 1–15.
[36] S. Xu, P. Wang, C.X. Zhang, J. Jinhu Lu, Spectral learning algorithm reveals propagation capability of complex networks, IEEE Trans. Cybern. 49 (12) (2019) 4253–4261.
[37] L. Lü, T. Zhou, Q.M. Zhang, H.E. Stanley, The H-index of a network node and its relation to degree and coreness, Nat. Commun. 7 (2016), 10168.
[38] M. Lei, H.C. Kang, Node influence ranking in complex networks: a local structure entropy approach, Chaos, Solit. Fractals 160 (2022), 112136.
[39] N. Zhao, J. Bao, N. Chen, Ranking Influential Nodes in Complex Networks with Information Entropy Method, 2020, 5903798.
[40] Z. Yu, J. Shao, Q. Yang, Z. Sun, ProfitLeader: identifying leaders in networks with profit capacity, World Wide Web 22 (2) (2019) 533–553.
[41] J. Zhao, Y. Wang, Y. Deng, Identifying influential nodes in complex networks from global perspective, Chaos, Solitons & Fractals 133 (2020), 109637.
[42] R. Pastor-Satorras, A. Vespignani, Epidemic dynamics and endemic states in complex networks, Phys. Rev. 63 (6) (2001), 066117.
[43] M.G. Kendall, A new measure of rank correlation, Biometrika 30 (1938) 81–93.