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)[1], Lai-Jun Zhao(赵来军)[1,†], Chen Dong(董晨)[2], Gui-Qiong Xu(徐桂琼)[2], and Li-Xin Zhou(周立欣)[1]

[1] *Business School, University of Shanghai for Science and Technology, Shanghai* 200093, *China*
[2] *Department of Information Management, School of Management, Shanghai University, Shanghai* 200444, *China*

(Received 7 April 2022; revised manuscript received 21 July 2022; accepted manuscript online 1 September 2022)

The influence maximization problem in complex networks asks to identify a given size of seed spreaders set to maximize the number of expected influenced nodes at the end of the spreading process. This problem finds many practical applications in numerous areas such as information dissemination, epidemic immunity, and viral marketing. However, most existing influence maximization algorithms are limited by the "rich-club" phenomenon and are thus unable to avoid the influence overlap of seed spreaders. This work proposes a novel adaptive algorithm based on a new gravity centrality and a recursive ranking strategy, named AIGCrank, to identify a set of influential seeds. Specifically, the gravity centrality jointly employs the neighborhood, network location and topological structure information of nodes to evaluate each node's potential of being selected as a seed. We also present a recursive ranking strategy for identifying seed nodes one-by-one. Experimental results show that our algorithm competes very favorably with the state-of-the-art algorithms in terms of influence propagation and coverage redundancy of the seed set.

Keywords: influential nodes, influence maximization, gravity centrality, recursive ranking strategy

PACS: 89.75.Fb **DOI:** 10.1088/1674-1056/ac8e56

1. Introduction

The study of influence maximization (IM) in complex networks has become a hot and fundamental issue in the information science field. The IM problem comes from marketing, which attempts to persuade a subset of targeted individuals to adopt a new product or innovation, in order to spark a significant cascade of widespread adoption by other users.[1–4] Massive real-life applications make it very urgent to identify a group of influential targeted individuals (called seed nodes), including the advertising of products, immunization against epidemics, monitoring of public opinion, and predicting and intervening in the online spread of major emergencies.[5–9] The research of the IM issue can help people to better understand and accelerate the dissemination of innovations, as well as analyzing, predicting, and controlling the spread of diseases, rumors, and computer viruses.[10–14]

Existing research and solutions for solving the IM problem can be divided into three families: greedy, heuristic and topology-based algorithms.[15] In greedy algorithms, Kempe *et al.*[16] transformed the influence maximization into an optimization problem, and has proved that the IM problem is NP-hard in the linear threshold model and independent cascade model. They presented a greedy algorithm with $1-1/e$ accuracy guarantee based on a hill-climbing search strategy. Leskovec *et al.*[17] developed the CELF (cost-effective lazy-forward) greedy algorithm, which avoids redundant computations by utilizing the sub-modularity property. Subsequently, Goyal *et al.*[18] presented CELF++ to extend CELF by considering the highest prior of the marginal gains of the nodes. This algorithm achieves higher computational efficiency through better pruning. Chen *et al.*[19] proposed the NewGreedy algorithm, which propagates influence on a smaller subgraph generated by removing some edges with a certain probability. Greedy algorithms rely on time-consuming Monte Carlo simulations to guarantee the performance of solutions, which limits their application to large-scale social graphs. As a result, despite outperforming other kinds of algorithms in terms of accuracy, greedy algorithms have gradually faded out of the attention of researchers due to their limited utility.

In the past decade or so, many researchers have turned their attention to heuristic algorithms. Jung *et al.*[20] devised a heuristic algorithm to select influential seed spreaders, which takes into account the characteristics of influence estimation and ranking. Sankar *et al.*[21] developed a bio-inspired algorithm to select a set of initial seeds, which is significant for rapid convergence towards a suboptimal solution. Weskida *et al.*[22] proposed an evolutionary algorithm to identify a set of influential spreaders in social networks. By viewing the IM problem as a genetic algorithm challenge, this algorithm outperforms some classical greedy algorithms in terms of quality and efficiency in the linear threshold model. Bao *et al.*[23] presented a heuristic clustering algorithm (HC) to solve the IM problem, which classifies nodes into different clusters according to the similarity between nodes, and the initial cluster cen-

†Corresponding author. E-mail: ljzhao@usst.edu.cn

© 2023 Chinese Physical Society and IOP Publishing Ltd

ters are selected randomly. Yang *et al.*[24] designed a heuristic iterative clustering algorithm to identify the seed spreaders in complex networks, and they selected the initial cluster centers by a hill-climbing search strategy. Sheikhahmadi *et al.*[25] presented a multi-objective artificial bee colony algorithm to solve the IM problem from two aspects of node influence and budget constraint.

Although the heuristic algorithms with low computational complexity can be applied to real-world networks of increasing scale, their accuracy is unstable. Therefore, in the light of the characteristics of the network topology, many scholars try to exploit topology-based algorithms to solve this problem. The well-known algorithms are to select top-ranked nodes based on the local or global topological structure, such as degree centrality (DC[26]), betweenness centrality (BC[27]), closeness centrality (CC[28]), k-shell decomposition centrality[29] and H-index centrality.[30] Among them, DC is a basic index to determine the influence of nodes by counting the number of the neighbor nodes, but the accuracy is finite. BC and CC are related to the shortest path between each pair of nodes, which have considerable computational complexity and are not applicable to large-scale networks. The k-shell determines the cores of a network by the onion-peeling method, and H-index evaluates the influence of nodes by comparing the topological structure of nodes and their neighbors. However, choosing the top-ranked nodes as multiple initial spreaders is not an optimal strategy due to the "rich club" phenomenon (nodes with great individual influence tend to connect with each other). Chen *et al.*[31] developed the DegreeDiscount algorithm to select the multiple seed spreaders in complex networks. In each round, the node with the greatest degree is selected as a seed, and its neighbor's degree score is discounted. Zhang *et al.*[32] presented the VoteRank algorithm to recursively identify the set of decentralized spreaders. All non-seed nodes can vote for their neighbors in each round. The node with the largest voting score is selected as a seed, and its neighbors' voting power will be reduced in the following round of voting. Chen *et al.*[33] proposed the ClusterRank algorithm, which comprehensively takes into account the influence and clustering coefficient of neighbor nodes. Jiang *et al.*[34] developed an efficient algorithm called HGD to find the influential seeds by iteratively searching for dense groups. Ullah *et al.*[35] developed the local-and-global-centrality (LGC) to identify the most valuable nodes by taking into account both local and global topological properties. Gupta *et al.*[36] designed the relative local–global importance algorithm named RLGI, which considers both node ranking scores and node location to prevent important nodes from clustering.

The topology-based algorithms aim at mining network topological features without extensive Monte Carlo simulations, and thus has a tremendous advantage over the diffusion-based algorithms in terms of efficiency. However, they frequently trigger the "rich club" phenomenon, which causes significant influence overlap among the chosen nodes. If only the extremely influential users are chosen as seeds, the stable coverage of influence spread will be unacceptable. As a result, one ideal situation for multiple spreaders is for the spreaders to be both influential and dispersed across networks.

Motivated by the aforementioned observations, this work proposes a novel adaptive influence maximization algorithm based on a new gravity centrality and a recursive ranking strategy, named the AIGCrank. In this paper, we use gravity law to assess node influence. Gravity law is a basic physical rule to calculate the interaction between two objects. The interaction between two objects is proportional to their masses and inversely proportional to the square of their distance. When we choose the top-ranked nodes as multiple initial seeds, the seed set usually experiences the "rich club" phenomenon, resulting in significant influence overlap among the chosen nodes. This phenomenon is a fundamental challenge for solving the influence maximization problem. As a result, one ideal situation is for the spreaders to be both influential and dispersed across networks. In light of this, we propose a recursive ranking strategy for selecting seed nodes. We endeavor to make the selected seed spreaders as important and dispersed as possible, so as to reduce the influence coverage redundancy among seeds. First of all, we define a gravity centrality index to evaluate the influence of nodes. Specifically, we characterize the mass of nodes comprehensively from three information dimensions: neighborhood, network location and topological structure. According to the average shortest path length of a given network, the neighborhood range in the gravity centrality is determined dynamically, which is conducive to improving the accuracy of node influence evaluation. After that, to avoid the "rich-club" phenomenon and minimize the influence coverage redundancy among seed spreaders, we propose a recursive ranking strategy to enhance the dispersion of seed spreaders. The main contributions of this work are summarized as follows.

(i) The proposed gravity centrality index accurately portrays the mass of nodes by comprehensively considering their multiple properties, including neighborhood, network location and topological structure.

(ii) The proposed gravity centrality index can be adjusted adaptively to the optimal mode according to the topological features of the network, which makes the final centrality score of each node more realistic.

(iii) The AIGCrank algorithm proposed in this paper selects seed spreaders one by one through the idea of recursive ranking, so as to achieve that each seed is significant and the location of the seed set is dispersed.

(iv) Experiments on several real-world datasets show that

our algorithm outperforms many other state-of-the-art algorithms on the coverage and speed of influence propagation. What is more, the AIGCrank algorithm is also significantly superior to the greedy and heuristic algorithms on computational complexity.

The rest of the paper is organized as follows. Section 2 presents some preliminaries on this work. Section 3 introduces the proposed AIGCrank algorithm. Section 4 shows the experimental setup, including the data sets, spreading model, and evaluation metrics. Experiment results and discussions are shown in Section 5, followed by concluding remarks in Section 6.

2. Preliminaries

Given an unweighted and undirected network $G = (V, E)$ with vertex set V and edge set E ($|V| = n$ and $|E| = m$). The adjacency matrix of G is denoted by $A = (a_{ij})_{n\times n}$ in which $a_{ij} = 1$ if nodes v_i and v_j are adjacent, and $a_{ij} = 0$ otherwise, $i, j = 1, 2, \ldots, n$. k_i is the number of linked neighbors of node v_i in G, that is, $k_i = \sum_{v_j \in V} a_{ij}$. In other words, k_i is the degree of node v_i. The normalized degree $DC(v_i)$ of node v_i is defined as $DC(v_i) = k_i/(n-1)$.[37]

The clustering coefficient reflects the tightness of the connectivity between the neighbors of a node, and can be used to express the probability that any two neighbors of a node are connected to each other.[33] The clustering coefficient $C(v_i)$ of v_i can be expressed as follows:

$$C(v_i) = \frac{E(v_i)}{k_i(k_i-1)/2} = \frac{2E(v_i)}{k_i(k_i-1)}, \tag{1}$$

where $E(v_i)$ is the total number of linked edges between all neighbors of node v_i. If v_i has no neighbors or only has one adjacent neighbor, then we set $C(v_i) = 0$.

H-index, which is a measure of a scholar's research level, can be used to represent the influence of a scholar in his or her field. In 2016, Lü *et al.*[30] extended the H-index concept to quantify how important a node is to its network. The H-index of a node is defined to be the maximum value H such that at least H neighbors of degree no less than H exists. Here, the degree of a node denotes the number of its linked neighbors. The H-index centrality considers the two-layer neighborhood topological structure of nodes, and its mathematical formula can be expressed as follows:

$$H(v_i) = \mathcal{H}(k_{v_1^i}, k_{v_2^i}, k_{v_3^i}, \ldots, k_{v_{k_i}^i}), \tag{2}$$

where $\{v_1^i, v_2^i, v_3^i, \ldots, v_{k_i}^i\}$ are the adjacent neighbors of node v_i, and $\{k_{v_1^i}, k_{v_2^i}, k_{v_3^i}, \ldots, k_{v_{k_i}^i}\}$ are the degree of its neighbors respectively. $\mathcal{H}$ is used to return the maximum integer H of the function, which means that node v_i has at least H neighbours with a degree no less than H.

With the development of network science, scholars have found that the location of a node in the network has a greater impact on its influence than the number of neighbor nodes.[29,38] Kitsak *et al.*[29] proposed k-shell decomposition to determine the relative location of nodes in the network. Given an undirected simple network G, initially, the ks of every isolated node (i.e., $k = 0$) is defined as $ks = 0$ and these nodes are removed before the k-shell decomposition. The specific process of k-shell decomposition is as follows: (i) Remove all nodes with $k = 1$ (the degree centrality is equal to 1) and their edges, and repeat this operation until there are no nodes with $k = 1$ in the network. The removed nodes and their edges together form the 1st-shell ($ks = 1$) of the network. (ii) Next, the degree of nodes in the new network is at least 2. Remove all nodes with $k = 2$ and their edges, and repeat this operation until there are no nodes with $k = 2$ in the network. The nodes removed in this step and their edges together form the 2nd-shell ($ks = 2$) of the network. (iii) Continuing this process, the k-shell values of remaining nodes can be derived in the given network.

The k-shell is so coarse that many nodes are given with the identical value. In scale-free networks, due to the degree of most nodes being small, their k-shell has poor distinguishing ability. In view of the problem, Bae *et al.*[39] proposed the neighborhood coreness centrality. The basic assumption of this centrality index is that if the neighbors of a node are closer to the core of the network, the greater the spreading capability of the node has. The calculation formula of the neighborhood coreness centrality is as follows:

$$NC(v_i) = \sum_{v_j \in \Gamma_1(v_i)} ks(v_j), \tag{3}$$

where $\Gamma_1(v_i)$ represents the 1-layer neighborhood of node v_i, and $ks(v_j)$ denotes the k-shell of node v_j.

Inspired by Newton's gravity law, several gravity centrality indexes have been presented to evaluate the influence of nodes in complex networks, which consider both the topological structure of a node and the relationship between the node and its neighborhood.[40,41] Ma *et al.*[40] proposed a gravity centrality by viewing the k-shell value of a node as its mass and the shortest path length between two nodes as their distance. The mathematical formula of this gravity centrality is as follows:

$$G(v_i) = \sum_{v_j \in \Gamma_r(v_i)} \frac{ks(v_i) \cdot ks(v_j)}{d_{ij}^2}, \tag{4}$$

where $\Gamma_r(v_i)$ represents the r-layer neighborhood of node v_i, and d_{ij} represents the distance between nodes v_i and v_j. For simplicity, r is set to 3 in Ref. [40].

3. A novel adaptive influence maximization algorithm

When selecting a group of seed nodes in a complex network, it is crucial to take into account the seed nodes' influence as well as the coverage redundancy between them. Seed nodes should be influential nodes with high centrality scores. However, influential nodes may cluster together emerging the "rich-club" phenomenon, resulting in significant coverage redundancy in the information spreading process. How to identify a set of seed nodes in complex networks is a challenging topic. Both VoteRank and DegreeDiscount algorithms largely mitigate the "rich club" phenomenon.[31,32] They consider the effect of newly elected seeds on their neighbors, i.e., reducing the potential of neighbors being seeds. VoteRank considers the impact of a node's neighbors on its influence through a self-defined voting rule. However, they have accuracy limits due to the simplicity of the information being considered.

Inspired by the above two algorithms, we propose a novel adaptive influence maximization algorithm to identify a seed spreader set with excellent collective influence in complex networks, which consists of two phases. Firstly, we construct a new gravity centrality index to better evaluate the influence of nodes. Here, the mass of nodes integrates multiple node properties with low time complexity, including the neighborhood, location and topology information of nodes. In addition, we dynamically determine the neighborhood range in the index according to the average shortest path length of the network. Secondly, we propose a recursive ranking strategy to select a group of seeds from influential nodes. The node with the largest gravity centrality score is selected as a seed. Then, the score of the newly elected seed's neighborhood nodes will reduce outward, layer by layer, and the reduction ratio is determined according to the distance between the seed node and its neighborhood nodes. The reduction ratio decreases with increasing distance from the seed node. The affected neighborhood range is dynamically determined according to the average shortest path length of the network.

3.1. A new gravity centrality

Most existing gravity centrality indices consider node mass (associated with node importance) in a very simplistic manner, and do not investigate a node's surrounding environment at a deeper level (e.g., network location, neighbors' influence, higher-order neighbors, and so on), and thus are inaccurate in many cases.[40,41] Furthermore, they usually set the neighborhood range in gravity centrality to a fixed value, which is not suitable for various networks.[42,43] If the neighborhood range is too large, the computational complexity is high, but the opposite may lead to poor performance. Therefore, characterizing the mass of nodes and determining the neighborhood range is still an open issue.[40–43]

In this paper, a novel gravity centrality index is proposed to evaluate the influence of nodes, which requires two parameters: node mass and neighborhood range. To overcome the limitations of a single index in characterizing node mass, such as the degree only considering the most local information of nodes, H-index ignoring low-influence neighbors, and k-shell being highly coarse-grained, we evaluate node mass from multiple information dimensions. In general, there are three categories of node attributes in complex networks, namely, local attribute, global attribute and network location. Global centrality indices such as betweenness centrality (BC), closeness centrality (CC), eccentricity (ECC), Katz centrality (Katz), and others consider the rich global network information and thus usually perform well. However, the computational complexity of global centrality indices is typically very high. ECC and Katz have a computational complexity of $O(n^3)$. BC and CC have a computational complexity of $O(n \times m + n^2 \times \lg n)$. Note that any highly accurate index will become meaningless if the consuming time is unacceptable.

As a result, we select the location and local attributes to characterize node mass. Specifically, we express node local attributes using both neighborhood (H-index) and topological structure (clustering coefficient) information, and node location attribute using neighborhood coreness centrality.

In comparison to DC centrality, the H-index takes into account not just the number of neighbors but also their quality, which can represent the neighborhood characteristic of nodes more reasonable. Since the poor distinguishing ability of k-shell centrality, neighborhood coreness is a better index to describe the network location of nodes compared to the k-shell. The clustering coefficient is a significant topological property that affects the spreading ability of nodes. The tightness of a node's neighbors is usually inversely proportional to its spreading ability. If the linkages between neighbors are dense, information will spread among them repeatedly, weakening the node's potential to propagate. Therefore, in this work, we argue that the clustering coefficient of nodes is negative to the spreading ability of nodes.

The mass $M(v_i)$ of node v_i is defined as follows:

$$M(v_i) = \frac{H(v_i) \cdot NC(v_i)}{1 + C(v_i)}, \tag{5}$$

where $H(v_i)$, $NC(v_i)$, and $C(v_i)$ are the H-index centrality, neighborhood coreness centrality and clustering coefficient of node v_i, respectively.

It is well known that the greater the influence of neighbors, the more influential the node itself is.[44] Therefore, we develop an extended version $M^+(v_i)$ to calculate the mass of node v_i. The $M^+(v_i)$ considers all neighbors of node v_i, and is defined as follows:

$$M^+(v_i) = \sum_{v_j \in \Gamma_1(v_i)} M(v_j), \tag{6}$$

where $\Gamma_1(v_i)$ is the nearest neighborhood set of node v_i.

The new gravity centrality $\mathrm{AGC}(v_i)$ of node v_i is defined as follows:

$$\mathrm{AGC}(v_i) = \sum_{v_j \in \Gamma_r(v_i)} \frac{M^+(v_i) \cdot M^+(v_j)}{d_{ij}^2}, \tag{7}$$

where $M^+(v_i)$ and $M^+(v_j)$ are the mass of node v_i and node v_j respectively, and d_{ij} is the length of the shortest path from node v_i to v_j. $\Gamma_r(v_i)$ is the r-layer neighborhood set of node v_i. The neighborhood range is another factor that affects the performance of the gravity centrality indices.[42,43] If r is too large, the computational complexity of the AGC will increase exponentially, which is not applicable in large-scale networks. If r is too small, only considering the information of a few neighborhood cannot fully consider the topological structure of nodes, which is not conducive to determining the real spreading capability of nodes. Therefore, to achieve a good balance between computational complexity and accuracy, this work defines r in Eq. (8) as follows:

$$r = \lceil \log_2 \langle d \rangle \rceil, \tag{8}$$

where $\langle d \rangle$ represents the average shortest path length of the network.

We also present the extended version $\mathrm{AGC}^+(v_i)$ of AGC, which considers the sum of neighbors' AGC values of node v_i, and is defined as follows:

$$\mathrm{AGC}^+(v_i) = \sum_{v_j \in \Gamma_1(v_i)} \mathrm{AGC}(v_j), \tag{9}$$

where $\Gamma_1(v_i)$ is the 1-layer neighborhood set of node v_i.

3.2. AIGCrank: Adaptive influence maximization algorithm

In real-world political elections, people tend to strongly support their ideal candidate and lessen their support for other competing candidates. Similarly, once a node has been selected as a seed, the potential of its neighbors being seeds will be discounted. Inspired by this, we propose a new adaptive influence maximization algorithm, which comprises the following: one to evaluate the influence of nodes using the gravity centrality index given in this work, and the other to select seed nodes one by one using a recursive ranking strategy suggested in this paper. We have elaborated the former in Subsection 3.1, and will focus on the latter next.

The core idea of the recursive ranking strategy is to select a group of seeds one by one according to current gravity centrality scores of nodes. In each iteration, the most influential node with the largest score is selected as seed. Once a node is added to the seed set, it will no longer participate in subsequent operations and its gravity centrality score will also be set to 0. Moreover, the score of neighborhood nodes of the newly elected seed will also be reduced gradually. And the reduction level is related to the distance between the elected seed and its each neighbor. Similarly, the range of affected neighborhood is determined adaptively according to the network's nature, namely the average shortest path of the network, and its value can be established by using Eq. (8).

The proposed adaptive influence maximization algorithm is expected effectively to avoid the emergence of "rich-club" phenomenon and expand the influence coverage of the seed set. The proposed algorithm can be divided into the following two phases, whose complete elaboration is given in Algorithm 1.

Algorithm 1 AIGCrank

Input: $A = (a_{ij})_{n\times n}$, adjacency matrix of a given network with n nodes and m edges.

Output: The seed set $S = \{v_{s_1}, v_{s_2}, \ldots, v_{s_x}\}$, where x denotes the scale of S, i.e. $x = |S|$.

/* Initialization */
for $i = 1$ to n **do**
  $H(v_i) = \mathcal{H}(k_{v_1^i}, k_{v_2^i}, k_{v_3^i}, \ldots, k_{v_{k_i}^i}) \leftarrow$ according to Eq. (1)
  $C(v_i) = 2E(v_i)/(k_i \times (k_i - 1)) \leftarrow$ according to Eq. (2)
  $NC(v_i) = \sum_{v_j \in \Gamma_1(v_i)} ks(v_j) \leftarrow$ according to Eq. (3)
  $M(v_i) = H(v_i) \times NC(v_i)/(1 + C(v_i)) \leftarrow$ according to Eq. (5)
for $i = 1$ to n **do**
  $M^+(v_i) = \sum_{v_j \in \Gamma_1(v_i)} M(v_j) \leftarrow$ according to Eq. (6)
$r = \lceil \log_2 \langle d \rangle \rceil$ // The dynamic neighborhood range of networks
for $i = 1$ to n **do**
  $\mathrm{AGC}(v_i) = \sum_{v_j \in \Gamma_r(v_i)} (M^+(v_i) \times M^+(v_j))/d_{ij}^2 \leftarrow$ according to Eq. (7)
for $i = 1$ to n **do**
  $\mathrm{AGC}^+(v_i) = \sum_{v_j \in \Gamma_1(v_i)} \mathrm{AGC}(v_j) \leftarrow$ according to Eq. (9)
for $i = 1$ to x **do**
  $v_i^* = \mathrm{argmax}\, \mathrm{AGC}^+(v_i)$, $v_i \in V$
  select v_i^* as a seed, i.e. $v_{s_i} = v_i^*$, and set $\mathrm{AGC}^+(v_i^*) = 0$
  $\mathrm{AGC}^+(v_j) = (1 - \frac{1}{2^l})\mathrm{AGC}^+(v_j)$, $v_j \in \Gamma_l(v_i^*) \setminus \Gamma_{l-1}(v_i^*)$, $l \in [1, r]$
return The seed set $S = \{v_{s_1}, v_{s_2}, \ldots, v_{s_x}\}$

Phase 1: Calculating gravity centrality

The centrality score AGC^+ of nodes in the network can be calculated by the suggested gravity centrality. The AGC^+ of a node is considered as the potential of the node being selected as the seed.

Phase 2: Recursive ranking

Step 1: All nodes in the network are ranked by centrality score AGC^+ in descending order. The node with the largest AGC^+ is selected as a seed spreader and its centrality score is set to 0. Then, according to the distance between the newly selected seed and its neighborhood nodes, the AGC^+ value of its

r-layer neighborhood nodes are reduced dynamically. The reduction ratio is half in the 1st-layer, one-fifth in the 2nd-layer, one-eighth in the 3th-layer, and until the rth-layer neighborhood. The value of r can be determined by using Eq. (8).

Step 2: Repeating the process of Step 1 recursively, until all of the seed spreaders have been selected.

To clearly illustrate the process of the proposed recursive ranking strategy, we take the network in Fig. 1 as an example. The initial AGC$^+$ value of nodes are shown in the left subfigure. Since the average shortest path length of the example network is 2.7121, we set the neighborhood range $r = \lceil \log_2 2.7121 \rceil = 2$ by using Eq. (8). In the first round of AIGCrank, node 10 with largest score 10.4106 is selected as the first seed, and the AGC$^+$ value of nodes in its 1st-layer neighborhood are reduced half, 2nd-layer neighborhood are reduced one-fourth, as shown in the right subfigure.

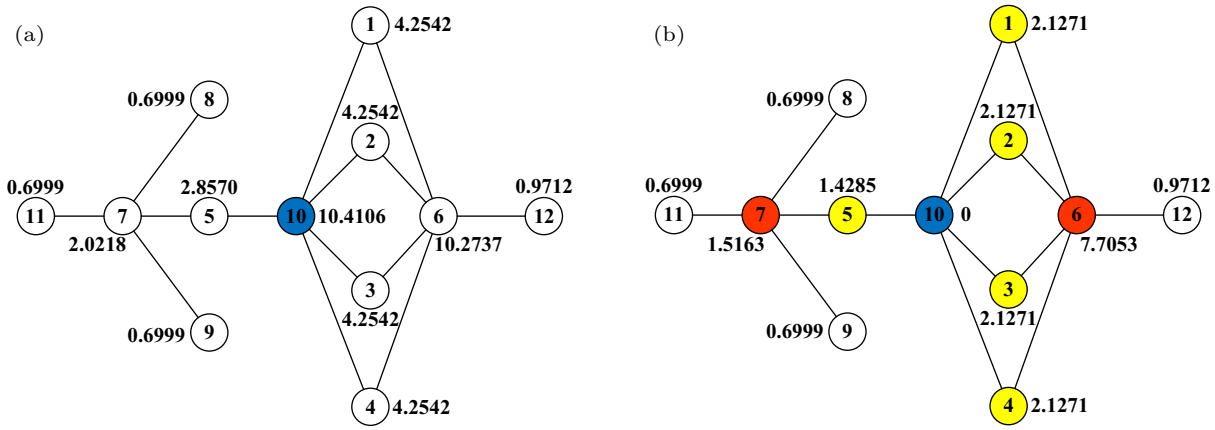


Fig. 1. An example network. Panel (a) shows the initial state of the example network. Panel (b) shows the state of adjustment of the example network after the first round of AIGCrank.

4. Experimental setup

4.1. Data sets

To evaluate the performance of the proposed AIGCrank algorithm, we conduct the comprehensive experiments in comparison with the state-of-the-art algorithms on six real-world networks, including URVemail,[45] Yeast,[46] Router,[47] CA-GrQc,[48] PairsFSG,[49] and Days.[50] URVemail refers to the network of e-mail interchanges between members of the Univeristy Rovira i Virgili (URV).[51] Yeast refers to protein–protein interaction network.[52] Router refers to a network representation of a portion of the router-level Internet.[53] CA-GrQc refers to scientific collaborations between authors papers submitted to general relativity and quantum cosmology (GrQc) category.[54] PairsFSG refers to the University of South Florida word association, rhyme, and word fragment norms.[55] Days refers to Reuters terror news network concerning the September 11 attack on the US.[56]

Table 1. The main statistical properties of six data sets.

Network	n	m	$\langle k \rangle$	$\langle d \rangle$	$\langle C \rangle$	HeC
URVemail	1133	5451	9.6222	3.6060	0.2540	1.9421
Yeast	2361	7182	6.0840	4.3762	0.2710	2.9439
Router	5022	6258	2.4920	6.4488	0.0330	5.5031
CA-GrQc	4158	13425	6.4570	6.0490	0.5296	3.0512
PairsFSG	10617	63784	12.016	4.0120	0.1870	3.1710
Days	13308	148035	11.124	3.1247	0.3860	10.032

Table 1 provides the main characteristics of the statistical topology of these networks. Column "Network" indicates the name of the network. For each network, columns "n" and "m" refer to the number of vertices and edges, respectively. Column "$\langle k \rangle$" indicates the average degree. Column "$\langle d \rangle$" represents the mean of all shortest paths. Column "$\langle C \rangle$" is the mean clustering coefficient, and column "HeC $= \langle k^2 \rangle / \langle k \rangle^2$" is the heterogeneity coefficient of the degree distribution.[57]

4.2. Reference algorithms

We adopt the following five state-of-the-art baseline algorithms as the reference algorithms in the computational comparisons.

(i) **HC (2017)**[23] A clustering-based heuristic algorithm named HC to select the central node of each community as the seed.

(ii) **VoteRank (2016)**[32] An iterative voting method named VoteRank to vote for a set of decentralized spreaders with the best spreading ability.

(iii) **HGD (2019)**[34] A heuristic group discovery algorithm named HGD to transfer the problem of influential spreaders selection into a problem to find groups with dense connections.

(iv) **LGC (2021)**[35] A local-and-global centrality (LGC) measuring algorithm to identify the vital nodes through handling local as well as global topological aspects of a network simultaneously.

(v) **RLGI (2021)**[36] A topology-based heuristic algorithm to identify the top k influential seed nodes by considering both the location of nodes in the network and the interaction effects between nodes.

4.3. Evaluation metrics

The independent cascade (IC) model is one of the most classical information propagation models, whose details are as follows: (i) When $t = 0$, set the propagation probability α. Set the nodes in S to be active and the remaining nodes to be inactive. The number of active nodes is $A(0) = |S|$. (ii) When $t \neq 0$, each node activated at $t-1$ has one chance to activate its inactive neighbors independently with α. $A(t)$ is the number of active nodes at t. (iii) If there are no newly activated nodes at t, that is $A(t) - A(t-1) = 0$, then the propagation ends, and we mark this time as t_c. Otherwise, repeat step (ii).

We employ three metrics to evaluate the performance of our algorithm.

Propagation influence Given an undirected graph $G = (V, E)$ with n nodes, the propagation influence $F(t)$ at time t under IC model can be defined as

$$F(t) = \frac{A(t)}{n}, \tag{10}$$

where $A(t)$ is the number of active nodes at time t.

Stable propagation influence The stable propagation influence $F(t_c)$ under IC model can be define as

$$F = \frac{A(t_c)}{n}, \tag{11}$$

where $A(t_c)$ represents the number of active nodes at time t_c, i.e., the information propagation reaching a stable state.

Mean distance The mean distance between seed nodes can be used to portray the dispersion of seed nodes.[58] The more dispersed the seed nodes are, the smaller the redundancy of their propagation coverage is. The mean distance d_{mean} between seed nodes can be defined as

$$d_{\text{mean}} = \frac{2}{x(x-1)} \sum_{v_i \neq v_j \in S} d_{ij}, \tag{12}$$

where $x = |S|$ is the number of the seed nodes, and d_{ij} is the distance between nodes v_i and v_j.

5. Results and analysis

To test and verify the performance of the proposed algorithm, we will employ the above three evaluation metrics to compare AIGCrank with five state-of-the-art algorithms. Five contrast algorithms include HC,[23] VoteRank,[32] HGD,[34] LGC,[35] and RLGI.[36] In the following, we conduct four groups of experiments on six real networks. The former three concentrate on comparing propagation influence of these six algorithms under the IC model. The last one focuses on contrasting influence coverage redundancy of these six algorithms.

5.1. Influence propagation

The number of seed nodes should be different in different scale networks. Here we define the seed ratio p ($p = x/n$) to describe the scale of seed nodes, where p is the ratio of the number of seed nodes to the number of all nodes in a network. Moreover, if the propagation probability α is too large, the propagation can easily extend to the entire network. Due to the difference in network topology, we conduct the comparative experiments with the α value close to the epidemic threshold $\alpha_c = \langle k \rangle / \langle k^2 \rangle$, i.e., $\alpha = q \cdot \alpha_c$ ($q \in [0.5, 1.5]$).

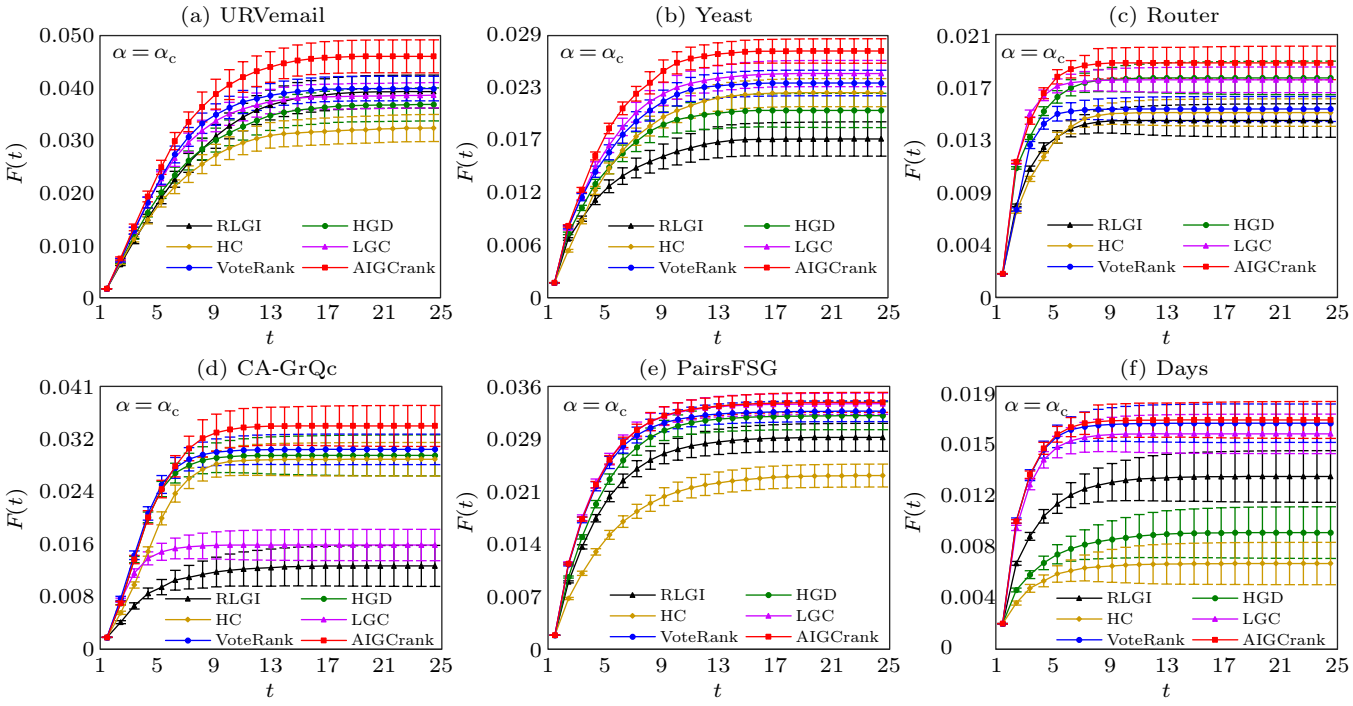


Fig. 2. The propagation influence $F(t)$ (at time t, $\alpha = \alpha_c$ and $p = 0.002$) on six networks ((a) URVemail, (b) Yeast, (c) Router, (d) CA-GrQc, (e) PairsFSG, and (f) Days) for different algorithms. Note that the experiment runs 100 times independently.

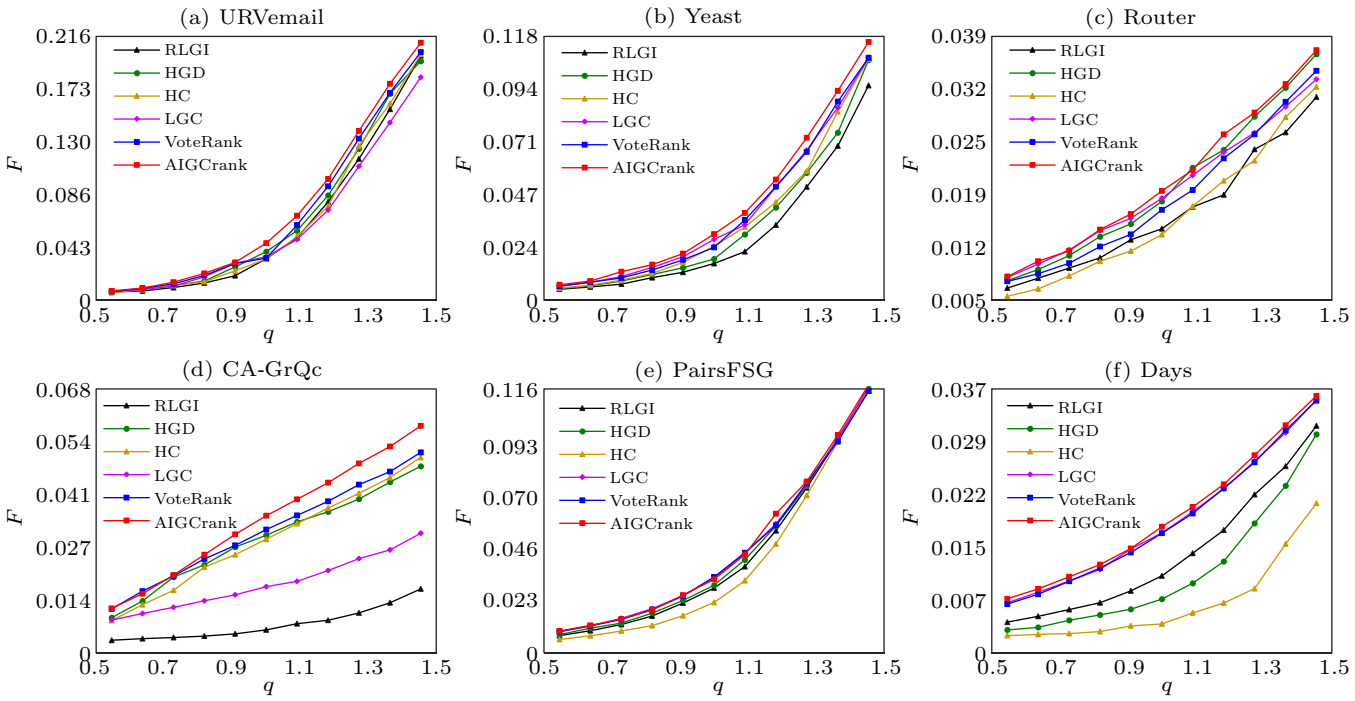


Fig. 3. The stable propagation influence F ($p = 0.002$, eleven different propagation probabilities $\alpha = q \cdot \alpha_c$, $q \in [0.5, 1.5]$) on six networks for different algorithms. Note that the experiment runs 100 times independently.

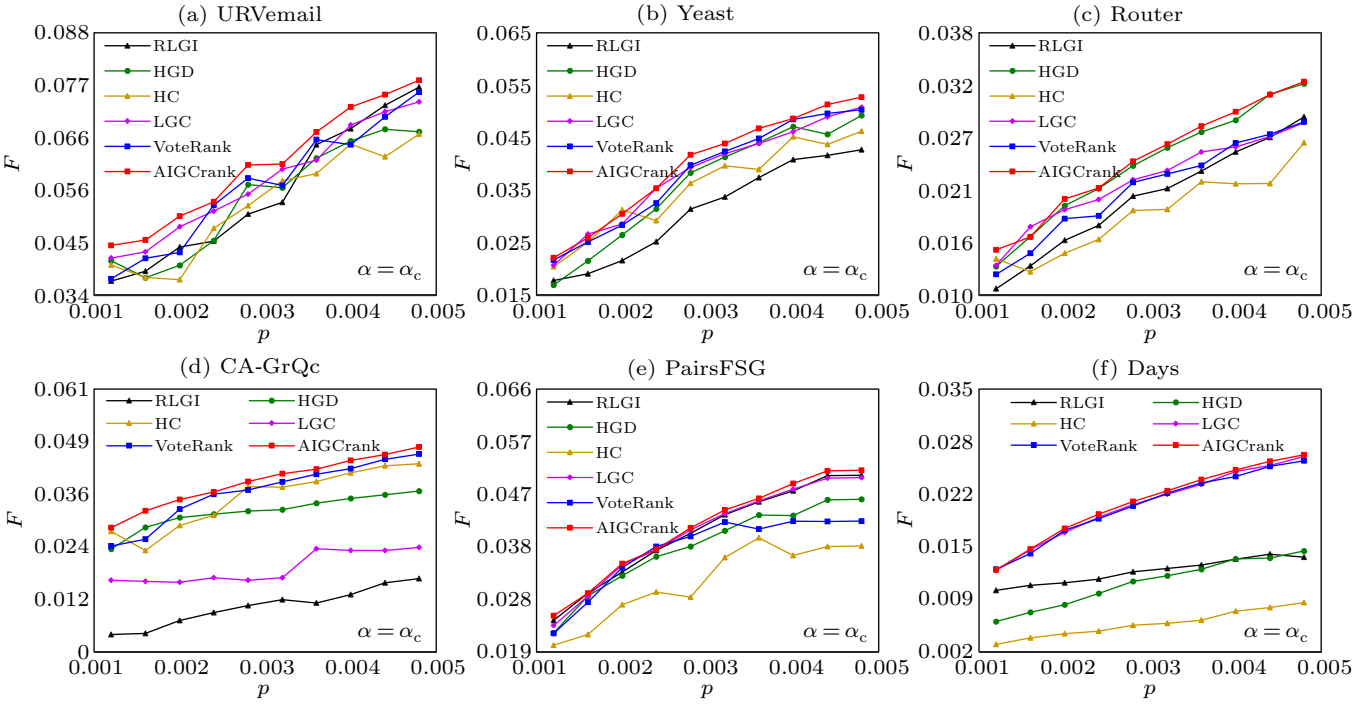


Fig. 4. The stable propagation influence F ($\alpha = \alpha_c$, ten different seed ratios $p \in [0.001, 0.0055]$) obtained by each compared algorithm on six real-world networks. Note that the experiment runs 100 times independently.

Figure 2 shows the propagation influence $F(t)$ (at time t, and $p = 0.002$ and $\alpha = \alpha_c$) on six networks for different algorithms. As shown in Fig. 2, although the propagation influence $F(t)$ of the proposed AIGCrank may not be the best in the propagation process, it is always the best when the propagation is stable. In Days network, VoteRank compares favorably to other four contrast algorithms. This is because the degree heterogeneity coefficient of Days network is high, which means that a few nodes have extremely high degree values. As a result, VoteRank, which emphasizes neighbor nodes voting contributions, can outperform other four algorithms.

To make the experiments more convincing, we use Figs. 3 and 4 to manifest that the proposed AIGCrank is not sensitive to the parameters (propagation probability α, seed ratio p). Figure 3 shows the experiment results when $\alpha = q \cdot \alpha_c$ ($q \in [0.5, 1.5]$) in URVemail, Yeast, Router, CA-GrQc, Pairs-

FSG, and Days. Figure 4 presents the comparative experiment results when $p \in [0.001, 0.0055]$ in six real-world networks.

Specifically, figure 3 shows the stable propagation influence F under eleven different propagation probabilities α in six networks. Without loss of generality, the source seed ratio p is set to be 0.002. As shown in Fig. 3, AIGCrank can obtain larger the stable propagation influence F than other five algorithms under different propagation probabilities. In CA-GrQc and Days, the performance of AIGCrank is significantly better than five contrast algorithms. In URVemail, Yeast, Router, and PairsFSG, AIGCrank has slight advantages over the other five algorithms. It is worth noting that the performance of the five compared algorithms is not always constant, but the VoteRank algorithm performs well most of the time, which reflects, to some extent, the feasibility of recursive selecting idea.

Moreover, figure 4 presents the stable propagation influence F under ten different source seed ratios p in six real networks, where propagation probability $\alpha = \alpha_c$. As shown in Fig. 4, AIGCrank can achieve larger stable propagation influence F than those of state-of-the-art algorithms under different seed ratios. AIGCrank algorithm achieves a significant comparative advantage in the CA-GrQc network. In Router, PairsFSG and Days, the performance of HC algorithm is not good, and these several networks have no obvious community characteristics, which indicates that the performance of HC depends on the community structure characteristics of the given network. To sum up, figures 3 and 4 show that the proposed algorithm is insensitive to parameters α and p, and the performance of AIGCrank is independent of the settings of these parameters.

5.2. Coverage redundancy

As pointed out Jiang *et al.*,[34] the more dispersed the seed nodes are, the smaller the possibility of coverage redundancy in their information propagation. We further analyze the dispersion of seed nodes through the average distance among them, as displayed in Fig. 5. It can be observed from Fig. 5 that the dispersion of AIGCrank is almost the best in Router and PairsFSG. In other remaining networks, the dispersion of AIGCrank is almost the second best, where the HC algorithm is the best. This is due to the fact that the HC algorithm selects the seed nodes from different clusters, which can better ensure the dispersion of the seed set. However, HC is obviously inferior to AIGCrank in the other two metrics, namely, propagation influence coverage and speed under the IC model, due to the fact that the performance of HC depends on the spherical community characteristics of the network.

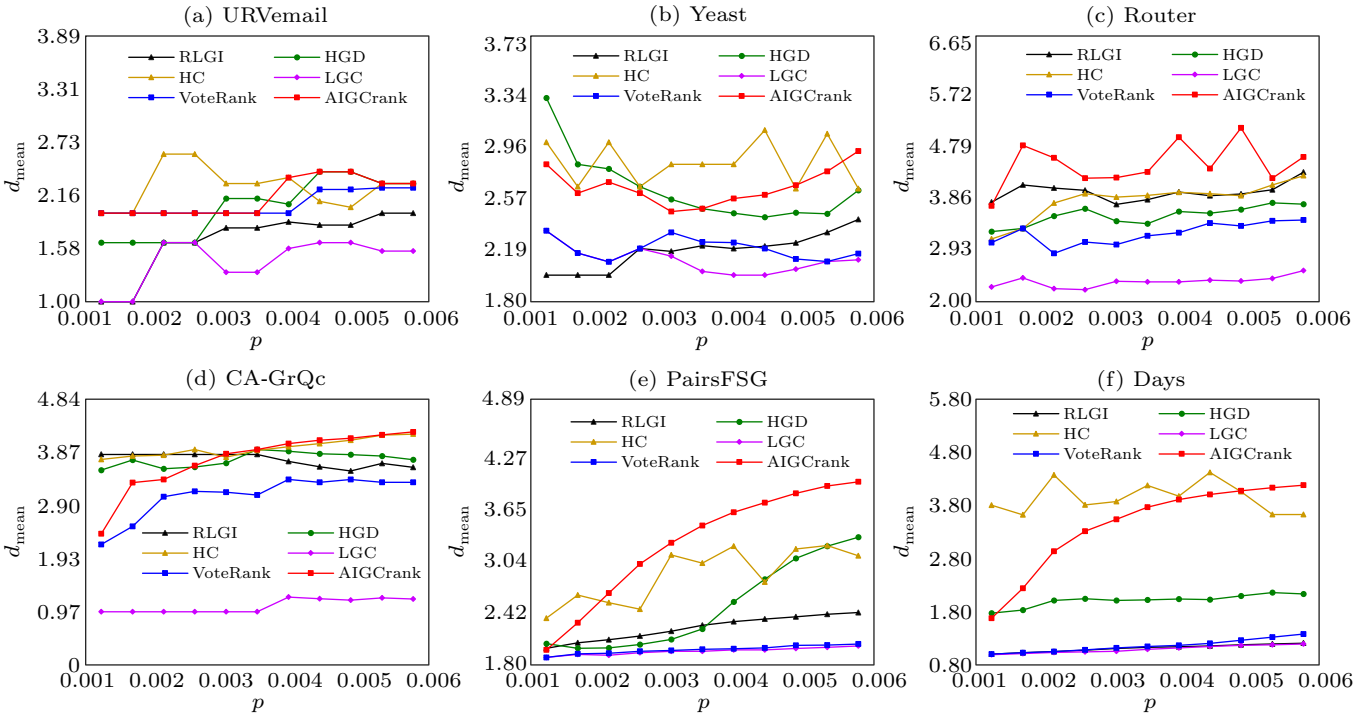


Fig. 5. The average distance d_{mean} (eleven different seed ratios $p \in [0.001, 0.006]$) between each pair of seed nodes derived from each compared algorithms.

In summary, the comparative experiments on six real networks show that the proposed algorithm outperforms those of state-of-the-art algorithms on propagation influence and coverage redundancy.

5.3. Complexity analysis

For a given network, the total number of nodes and the average degree are assumed to be n and $\langle k \rangle$, respectively. The proposed algorithm consists of the following three parts: cal-

culating the mass, initializing the gravity centrality score, and recursive ranking. For the first part, calculating the mass requires the H-index centrality, neighborhood coreness centrality and clustering coefficient of nodes in complex networks. All three indicators have computational complexity $O(n)$ respectively, so the computational complexity of calculating the mass is $O(n+n+n) = O(n)$. When calculating a node's gravity centrality score, the r-layer neighborhood must be taken into account. Therefore, the computational complexity of the second part is $O(\langle k\rangle^r \times n)$. For the third part, in the recursive ranking strategy, a seed node is selected for each round. Here, only updating the gravity centrality score of the r-layer neighborhood of the newly selected seed. Hence, the computational complexity is $O(x \times \langle k\rangle^r)$, where x is the number of seed nodes. As a result, the computational complexity of the third portion is significantly lower than $O(n)$. So the total computational complexity of the proposed AIGCrank algorithm is $O(3 \times n + \langle k\rangle^r \times n)) = O(\langle k\rangle^r \times n)$. Most of large-scale networks are sparse, and their average degree $\langle k\rangle$ is very small in comparison to the number n of nodes, i.e., $\langle k\rangle \ll n$. According to the small-world theory,[59] the average shortest path in most real networks is small, so r is a small value. Therefore, if networks have small-world and sparsity characteristics, the computational complexity of AIGCrank can be reaching $O(n)$.

The computational complexity of five contrast algorithms is shown in Table 2. It is clearly seen that our algorithm and four contrast algorithms are with low computational complexity, i.e., $O(n)$. Although LGC performs well in terms of influence propagation, particularly in Days network, its computational complexity ($O(n^2)$) is relatively high and it is not applicable to large-scale networks.

Table 2. The complexity of algorithms for comparison.

Network	Time complexity
AIGCrank	$O(n)$
HC	$O(n)$
VoteRank	$O(n)$
HGD	$O(n)$
LGC	$O(n^2)$
RLGI	$O(n)$

6. Conclusions

Influence maximization problem covers massive aspects in daily life, such as viral marketing, epidemic prevention and so on, which is an important research topic in the information science field. Most of the existing influence maximization algorithms suffer from high computational complexity or insufficient accuracy. The algorithms based on topological information have a low computational complexity and can be applied to large-scale networks, but their accuracy needs further improvement. To effectively avoid the phenomenon of the "rich-club" in spreading and improve the accuracy of influence maximization solutions, in this work, we propose the AIGCrank algorithm to select the influential multiple seed spreaders in complex networks. One key to our algorithm first lies in calculating the score of each node according to the proposed gravity centrality, which takes into full consideration from the perspective of the neighborhood, network location and topological structure information of nodes. Note that the gravity centrality index can be adjusted adaptively to the optimal mode according to the topological features of the network. The other key to our algorithm lies in selecting the seed nodes one by one by the strategy of recursive ranking, the core of which is to adaptively determine the range of neighborhoods affected by the selected seeds based on the extracted network properties. Experiments in six real-world networks demonstrate that the suggested algorithm outperforms the five state-of-the-art algorithms in terms of accuracy. Our algorithm is also significantly superior to other greedy and heuristic influence maximization algorithms on computational complexity, and presents a new exploration for the influence maximization issue in the information science field.

This work only simulates the propagation process using the IC model and does not consider other propagation models such as SI, SIR, and others. Empirical analysis of more real networks, propagation models, and well-known and newly proposed algorithms are all extremely beneficial for gaining knowledge and experience. We expect that a clear picture of this issue will be completed in the future by assembling many fragments from various empirical studies. In recent years, many researchers have focused on the structure and dynamics of temporal networks.[60,61] Identifying a set of influential nodes in temporal networks is an important and fascinating topic. It is worthwhile to investigate how to address the challenge of the time-dependent node activity and to extend our work to temporal networks. We also intend to investigate the influence maximization problem considering information type and topic community characteristics in the future, i.e., how different information types and topic communities affect the propagation process.

Appendix A: Comparative experiments of node influence indices for the recursive ranking strategy

To further test and verify the performance of the proposed algorithm, we devise four contrast algorithms based on the proposed recursive ranking strategy and conduct comparative experiments. On the one hand, to validate the benefit of combining multiple attributes for evaluating node mass, three of the compared algorithms characterize node mass separately using single-attribute indices such as H-index, neighborhood coreness, and k-shell. These three indices are implemented in

the proposed algorithm, which correspond to three compared algorithms named AIGCrank-1, AIGCrank-2, and AIGCrank-3. On the other hand, to further verify the performance of the proposed recursive ranking strategy, we devise the fourth compared algorithm (named GCrank) in which the gravity centrality proposed by Ma *et al.*[40] is implemented in our recursive ranking strategy.

Four contrast algorithms based on our recursive ranking strategy use the gravity raw framework to evaluate node influence. Their main distinction is in how node mass is defined. AIGCrank-1, AIGCrank-2, and AIGCrank-3 characterize node mass using H-index, neighborhood coreness, and clustering coefficient, respectively. GCrank employs k-shell to characterize node mass. The proposed AIGCrank characterize node mass from multiple information dimensions such as neighborhood, network location and topological structure. The performance of the five algorithms reflects the accuracy with which node mass is characterized.

In Section 5, the performance of AIGCrank has been tested through four group competitive experiments in terms of the three evaluation metrics such as propagation influence, stable propagation influence and mean distance. In this section, we use the most representative evaluation metric, stable propagation influence (under different propagation probabilities α and seed ratios p), to validate the effects of different node influence indices in a nutshell.

Figure A1 shows the comparative experiment results when $\alpha = q \cdot \alpha_c$ ($q \in [0.5, 1.5]$) in URVemail, Yeast, Router, CA-GrQc, PairsFSG and Days. Figure A2 presents the comparative experiment results when $p \in [0.001, 0.0055]$ in six real-world networks. Figure A1 shows the stable propagation influence F under eleven different propagation probabilities α in six networks. Without loss of generality, we fix the seed ratio p as 0.002. As shown in Fig. A1, AIGCrank can obtain larger stable propagation influence F than four other compared algorithms under different propagation probabilities. AIGCrank-2, the neighborhood coreness index being implemented in the proposed algorithm, performs slightly worse than our AIGCrank, but better than three other contrast algorithms.

Figure A2 presents the stable propagation influence F under ten different seed ratios p in six real networks, where propagation probability $\alpha = \alpha_c$. As shown in Fig. A2, AIGCrank can achieve larger stable propagation influence F than the four compared algorithms under different seed ratios. AIGCrank-3, the clustering coefficient index being implemented in the proposed algorithm, performs the worst, indicating that using clustering coefficients alone to assess node importance is not a good idea. In fact, the clustering coefficient is typically used in conjunction with the degree index when determining node importance.

In conclusion, our algorithm outperforms the four contrast algorithms under various propagation probabilities and seed ratios. Moreover, the computational complexity of the five algorithms is approximately linear. This indicates that the gravity centrality index proposed in this paper, which is based on multiple information dimensions, can accurately assess node influence with low computational cost. It also validates the effectiveness of the proposed recursive ranking strategy for selecting seed nodes.

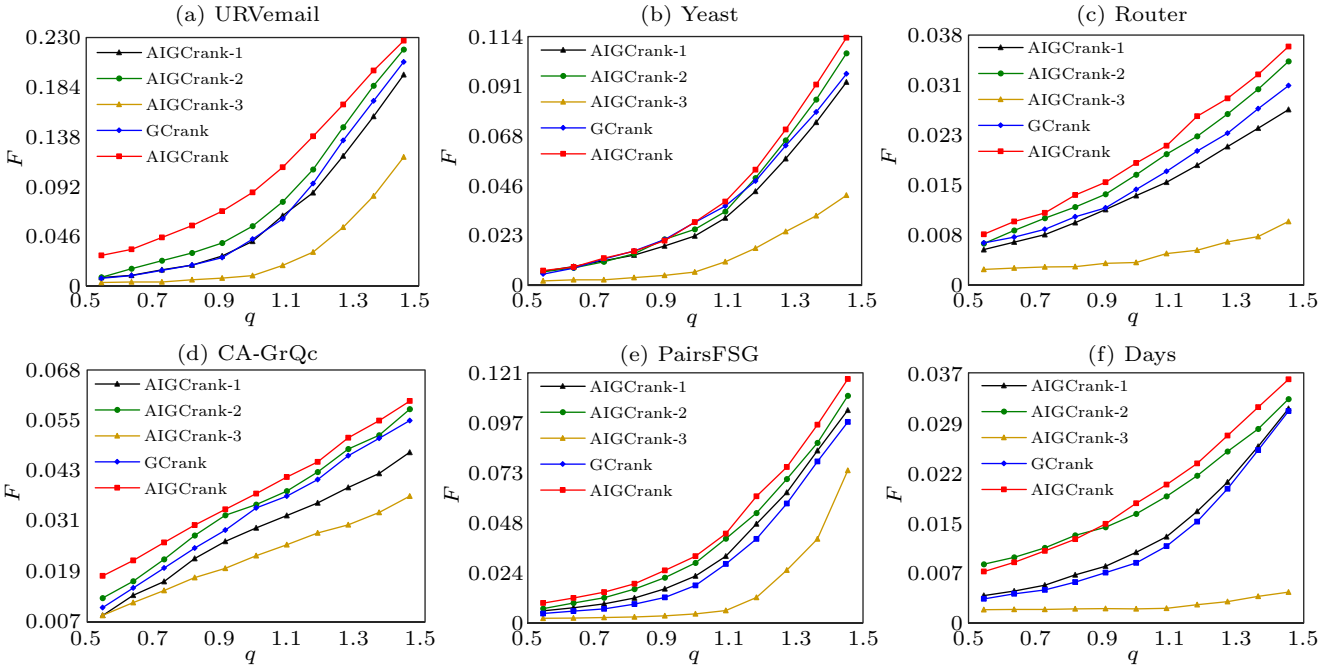


Fig. A1. The stable propagation influence F ($p = 0.002$, eleven different propagation probabilities $\alpha = q \cdot \alpha_c$, $q \in [0.5, 1.5]$) on six networks for different algorithms. Note that the experiment runs 100 times independently.

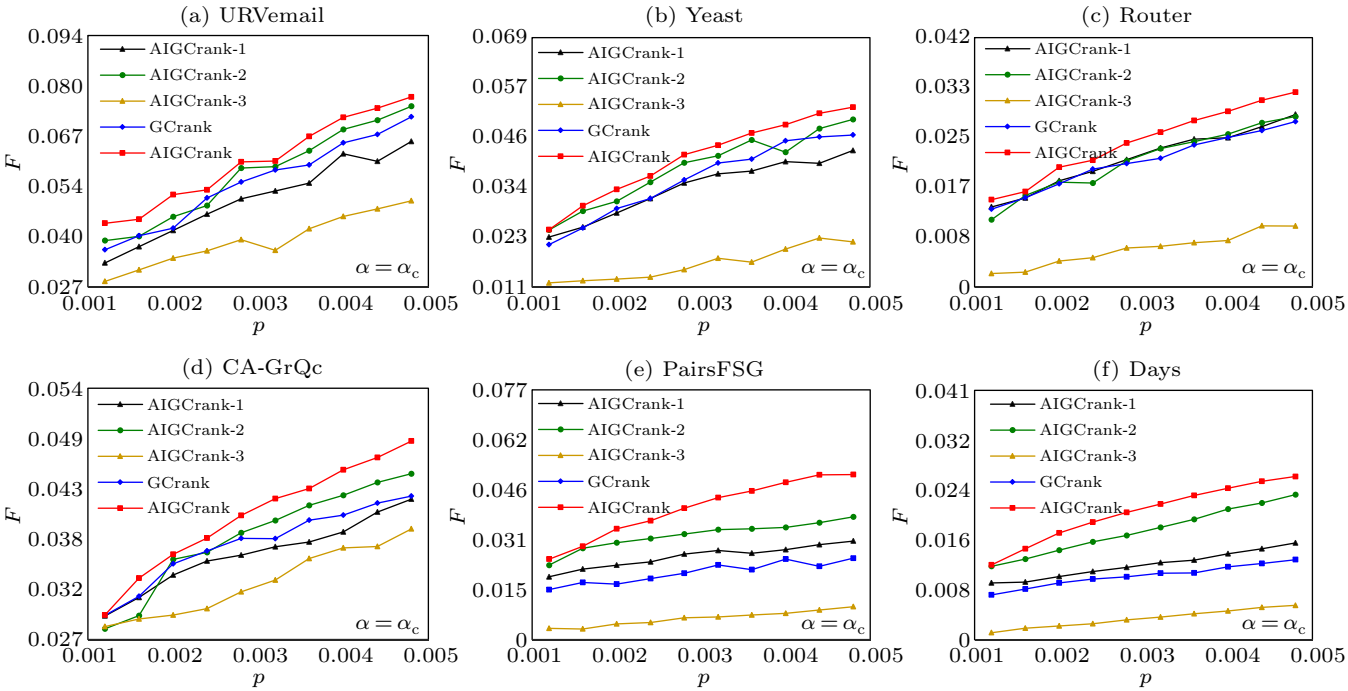


Fig. A2. The stable propagation influence F ($\alpha = \alpha_c$, ten different seed ratios $p \in [0.001, 0.0055]$) obtained by each compared algorithm on six real-world networks. Note that the experiment runs 100 times independently.

Acknowledgements

Project supported by the National Social Science Foundation of China (Grant Nos. 21BGL217 and 18AZD005) and the National Natural Science Foundation of China (Grant Nos. 71874108 and 11871328).

References

[1] Wang C, Shi Q H, Xiao W Z, Feng Y and Chen C 2021 *Knowl-Based Syst.* **213** 106692
[2] Wang Z X, Sun C C, Xi J K and Li X C 2021 *Future Gener. Comp. Sy.* **118** 327
[3] Guo J X and Wu W L 2021 *ACM. T. Knowl. Discob. D.* **15** 1
[4] Yan X L, Cui Y P and Ni S J 2020 *Chin. Phys. B* **29** 048902
[5] Caliò A and Tagarelli A 2021 *Knowl-Based Syst.* **546** 1273
[6] Li H, Zhang R S, Zhao Z L, Liu X and Yuan Y N 2021 *Appl. Intell.* **51** 7749
[7] Güney E, Leitner M, Ruthmair M and Sinnl M 2021 *Eur. J. Oper. Res.* **289** 144
[8] Wang X F, Tong X, Fan H, Wang C Y, Li J X and Wang X 2021 *Knowl-Based Syst.* **221** 106944
[9] Yang Y Z, Hu M and Huang T Y 2020 *Chin. Phys. B* **29** 088903
[10] Wang Z X, Sun C C, Yuan G, Rui X B and Yuan X D 2020 *J. Comput. Sci.* **43** 101129
[11] Du Y X, Gao C, Hu Y, Sankaran M and Deng Y 2014 *Physica A* **399** 57
[12] Wang J and Wang Y Q 2015 *Chin. J. Phys.* **53** 020702
[13] Domingos P and Richardson M 2001 *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 26–29, 2001, San Francisco, USA, p. 57
[14] Bian T and Deng Y 2018 *Chaos* **28** 043109
[15] Olivares R, Muñoz F and Riquelme F 2021 *Knowl-Based Syst.* **212** 106623
[16] Kempe D, Kleinberg J and Tardos É 2003 *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 24–27, 2003, Washington, USA, p. 137
[17] Leskovec J, Krause A, Guestrin C, Faloutsos C, Vanbriesen J and Glance N 2007 *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 12–15, 2007, New York, USA, p. 420
[18] Goyal A, Lu W and Lakshmanan L V 2011 *Proceedings of the 20th International Conference Companion on World Wide Web*, March 25–28, 2011, New York, USA, p. 47
[19] Chen W, Wang Y and Yang S 2009 *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, June 28–30, 2009, Paris, France, p. 199
[20] Jung K, Heo W and Chen W 2012 *Proceedings of the 12th International Conference on Data Mining*, December 10–13, 2012, Brussels, Belgium, p. 918
[21] Sankar C P and Kumar K S 2016 *PLoS One* **11** e0168125
[22] Weskida M and Michalski R 2016 *Proceedings of the* 2016 *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, August 18–21, 2016, San Francisco, USA, p. 1189
[23] Bao Z K, Liu J G and Zhang H F 2017 *Phys. Lett. A* **381** 976
[24] Yang P L, Xu G Q, Yu Q and Guo J W 2020 *Chaos* **30** 093106
[25] Sheikhahmadi A and Zareie A 2020 *Appl. Soft Comput.* **94** 106436
[26] Freeman L C 1978 *Soc. Netw.* **1** 215
[27] Freeman L C 1977 *Sociometry* **40** 35
[28] Xu G Q, Meng L, Tu D Q and Yang P L 2021 *Chin. Phys. B* **30** 088901
[29] Maksim K, Gallos L K, Havlin S, Liljeros F, Muchnik L, Stanley H E and Makse H A 2010 *Nat. Phys.* **6** 888
[30] Lü L Y, Zhou T, Zhang Q M and Stanley H E 2016 *Nat. Commun.* **7** 10168
[31] Chen W, Wang C and Wang Y J 2010 *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, July 24–28, 2012, Washington, USA, p. 1029
[32] Zhang J X, Chen D B, Dong Q and Zhao Z D 2016 *Sci. Rep.* **6** 27823
[33] Chen D B, Gao H, Lü L Y and Zhou T 2013 *PLoS One* **8** e77455
[34] Jiang L C, Zhao X, Ge B, Xiao W D and Ruan Y R 2019 *Physica A* **516** 58
[35] Ullah A, Wang B, Sheng J F, Long J, Khan N and Sun Z J 2021 *Expert Syst. Appl.* **186** 115778
[36] Gupta M and Mishra R 2021 *Decis. Support Syst.* **149** 113608
[37] Wasserman S and Faust K 1994 *Social network analysis: Methods and applications* (Cambridge: Cambridge University Press) p. 178
[38] Namtirtha A, Dutta A and Dutta B 2018 *Physica A* **499** 310
[39] Bae J and Kim S 2014 *Physica A* **395** 549
[40] Ma L L, Ma C, Zhang H F and Wang B H 2016 *Physica A* **451** 205
[41] Wang Z X, Du C J, Fan J P and Xing Y 2017 *Neurocomputing* **260** 466
[42] Li H W, Shang Q Y and Deng Y 2021 *Chaos Soliton Fract.* **143** 110456
[43] Yang X and Xiao F Y 2021 *Knowl-Based Syst.* **227** 107198
[44] Yang P L, Liu X and Xu G Q 2019 *Int. J. Mod. Phys. C* **30** 1940008
[45] http://konect.cc/networks/arenas-email/
[46] http://vlado.fmf.uni-lj.si/pub/networks/data/bio/Yeast/Yeast.htm
[47] http://math.bu.edu/people/kolaczyk/datasets.html

[48] http://snap.stanford.edu/data/ca-GrQc.html
[49] http://vlado.fmf.uni-lj.si/pub/networks/data/dic/fa/FreeAssoc.htm
[50] http://vlado.fmf.uni-lj.si/pub/networks/data/CRA/terror.htm
[51] Guimerà R, Danon L, Díaz-Guilera A, Giralt F and Arenas A 2003 *Phys. Rev. E* **68** 065103
[52] Bu D B, Zhao Y, Cai L, Xue H, Zhu X P, Lu H C, Zhang J F, Sun S W, Ling L J, Zhang N et al. 2003 *Nucleic Acids Res.* **31** 2443
[53] Hand D J 2010 *Int. Stat. Rev.* **78** 135
[54] Leskovec J, Kleinberg J and Faloutsos C 2007 *ACM Trans. Knowl. Discov. Data* **1** 2
[55] Nelson D L, McEvoy C L and Schreiber T A 2004 *Behav. Res. Meth. Ins. C.* **36** 402
[56] Corman S R, Kuhn T, Mcphee R D and Dooley K J 2002 *Hum. Commun. Res.* **28** 157
[57] Yuan X, Shao S, Stanley H E and Havlin S 2015 *Phys. Rev. E* **92** 032122
[58] Kundu S and Pal S K 2015 *Inform. Sciences* **316** 107
[59] Uzzi B, Amaral L A and Reed-Tsochas F 2007 *Eur. Manag. Rev.* **4** 77
[60] Qu C Q, Zhan X X, Wang G H, Wu J L and Zhang Z K 2019 *Chaos* **29** 033116
[61] Song J, Weng T F, Gu C G and Yang H J 2020 *Chin. Phys. B* **29** 080501