Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Finding the key nodes to minimize the victims of the malicious information in complex network

Mingyang Zhou, Hongwu Liu, Hao Liao [*], Gang Liu, Rui Mao

College of Computer and Software, Shenzhen University, Shenzhen, 518060, China

ARTICLE INFO

Keywords:
Complex network
Targeted immunization
Submodularity
Optimization

ABSTRACT

Safeguarding crucial nodes provides a direct approach to impede the dissemination of malicious information in complex networks, such as the Internet. However, determining the optimal budget size, represented as k, for protecting nodes is a challenging problem classified as NP-hard. In this study, we investigate the origin of the NP-hard property, known as the influence redundancy mechanism, as a means to address this problem. The influence redundancy characterizes the intricate interactions among key nodes. Subsequently, we introduce an objective function that allows for the optimization of the set of key nodes. Our objective is to minimize the spectral radius of the adjacency matrix after removing these key nodes. We mathematically prove that the objective function exhibits the submodular property, and our proposed method achieves an approximation ratio of $(1 - 1/e)$ with a time complexity of $O(N \log N)$, where N represents the size of the network. Experimental results show that the identified key nodes outperform classical methods in 20 empirical networks, specifically in the Susceptible-Infected-Recovered (SIR) model and Independent Cascade (IC) model, thus confirming their improved performance quality.

1. Introduction

In recent years, there has been significant attention given to the proliferation of epidemics and the dissemination of malicious information, such as the monkeypox outbreaks and the COVID-19 pandemic [1]. Malicious information often spreads through the contact–contact relationships between agents, which can be observed in scenarios like the transmission of infectious diseases in human-to-human social networks or the proliferation of computer viruses on the Internet. The consequences of a malicious information outbreak can result in severe economic crises and trigger cascading failures in critical infrastructure systems, such as airline disruptions and road congestion [2,3]. Understanding the dynamics of malicious information spread relies on the inherent characteristics of the malicious content and the underlying network structure. Since the behavior of a network is mainly influenced by a small fraction of key nodes, one effective approach to impede the dissemination of malicious information is to target these key nodes for immunization [4–6]. However, determining the optimal selection of key nodes remains a challenging NP-hard problem.

The identification of key nodes for impeding the spread of malicious information can generally be categorized into three main approaches: Firstly, Monte Carlo simulation methods involve conducting numerous simulations utilizing specific information diffusion models to assess the influence of nodes [7]. Nevertheless, the computational time required makes these methods impractical for large networks. Secondly, heuristic-based methods rely on intuition and network centrality measures to identify key nodes, without considering the specific spreading model of the malicious information [8]. Examples include high degree [9], betweenness [10], and k-shell centrality [11]. However, the performance of these methods cannot be guaranteed due to their fluctuating nature, limiting their applicability in real-world scenarios. Lastly, analytic-based methods employ specific spreading models and optimize objective functions to identify key nodes [12–14]. Typically, these methods exhibit superior performance compared to heuristic-based approaches. Traditionally, classical methods [8] for identifying key nodes select one node at a time based on their importance, which is evaluated using heuristic or analytic methods. In some cases, the importance is re-calculated once a new key node is determined, which can alleviate redundancy issues. However, most existing methods primarily focus on the individual importance of nodes, disregarding the redundancy of importance between different nodes.

For the collective influence of multiple nodes, the marginal gain of a node's importance decreases when the node is added to a larger node-set, which is attributed to the redundancy of importance. However, most existing methods use either simple greedy methods [13,15] or

[*] Corresponding author.
E-mail address: haoliao@szu.edu.cn (H. Liao).

https://doi.org/10.1016/j.knosys.2024.111632
Received 22 August 2023; Received in revised form 28 February 2024; Accepted 10 March 2024
Available online 16 March 2024

deep neural networks to identify influential nodes [16]. We still lack a method to systematically investigate the influence redundancy between nodes.

This study examines the detection of crucial nodes in intricate networks by exploiting the redundancy mechanism. Our primary aim is to minimize the maximum eigenvalue of the network's adjacency matrix, as these significant nodes play a crucial role in achieving this objective. Their importance is determined by the relative variation in the maximum eigenvalue upon their removal. To accurately evaluate the significance of a set of nodes, we propose a methodology that utilizes path lengths to assess the impact and redundancy of important nodes. Subsequently, we present an iterative algorithm, denoted as the "Set Influence Algorithm" (SIA), which optimizes an objective function to identify the key nodes. The SIA method effectively maximizes the influence of the set of nodes while simultaneously minimizing influence redundancy. In comparison to traditional approaches, the SIA demonstrates superior performance without escalating the time complexity.

We summarize our main contributions as follows:

- We propose a method to characterize the set influence and influence redundancy among key nodes.
- We prove that the set influence function has the submodular property that guarantees $(1 - 1/e)$ approximation ratio.
- we propose an efficient algorithm to optimize the set influence and calculate the key nodes, with time complexity $O(N \log N)$.
- We compare the proposed method with state-of-the-art methods. Experimental results validate that our method outperforms other methods.

The rest of the paper is organized as follows: Section 2 presents the related work. Section 3 describes the preliminary problem. We present the method in Section 4. In Section 5, we evaluated the performance of SIA on 20 real networks and compared it with the state-of-the-art methods. Finally, Section 6 summarizes the paper.

2. Related work

The related work could be divided into three classes: The influence maximization problem (IMP), node immunization, and influence redundancy.

The influence maximization problem: The Influence Maximization Problem (IMP) in social networks was initially studied by Kempe et al. [7]. They demonstrated the NP-hardness of the problem under the Independent Cascade (IC) and Linear Threshold (LT) models. In the existing literature, several node importance measurements have been proposed, including degree centrality (HD) [9], betweenness centrality (BC), PageRank [17], eigenvector centrality (EC) [18], K-shell [11], and other heuristic methods [8]. These methods assign scores to nodes by leveraging the topological structure of networks, with nodes having high scores being considered as key nodes. In addition to these methods, Borgs et al. [19] introduced Reverse Influence Sampling (RIS) as a concept to address the IMP. Building upon this, Tang et al. [15] proposed the Influence-based Multi-armed Bandit (IMM) algorithm, which incorporates the martingale technique to enhance performance and provides a guaranteed $(1 - 1/e - \epsilon)$-approximation. Later on, much attention was paid to reducing the time complexity of IMM [13,16]. Chen et al. [20,21] summarized the recent achievement of influence maximization. Furthermore, Morone and Makse [22] mapped the IMP to percolation theory and proposed the Collective Influence (CI) greedy algorithm. The CI algorithm measures the importance of a node based on the number of ℓ-length neighbors and selects the most significant nodes during each round. Lastly, Fan et al. [23,24] introduced a deep reinforcement learning framework called FINDER. FINDER can be trained on small synthetic networks generated by toy models and subsequently applied to a wide range of application scenarios.

Node immunization: Identifying key nodes to prevent the spread of malicious information can address the network robustness problem [12,25,26] as well as the node immunization problem [5,6,27]. The literature on the spread of malicious information and epidemic thresholds has extensively examined the relationship between epidemic thresholds and the largest eigenvalue [28,29]. For instance, Tong et al. [4] investigated strategies for removing nodes in a network to impede epidemic spread, using the largest eigenvalue of the adjacency matrix to quantify node importance and propagation efficiency. Expanding on the concept of node removal, Tong et al. [30] explored the optimal placement of a set of edges in the underlying graph, including edge deletion and addition, to optimize the largest eigenvalue. Their research revealed the intrinsic relationship between edge deletion and node removal problems. In a similar vein, Tariq et al. [31] formulated a monotone and submodular objective function to minimize the largest eigenvalue of the graph. They also proposed a randomized approximation algorithm to estimate the score of each vertex. Other related works [32–34] have studied the influence of spectral radius of non-backtracking matrix on the immunization problem.

Influence redundancy: The interactions among key nodes are complex, leading to unstable performance on different structured networks. Zhou et al. [35] addressed the influence overlap and proposed a novel framework to enhance the collective influence of multiple nodes. Zhang et al. [36] introduced VoteRank, a method to identify key nodes using a voting mechanism. When a node is selected as a key node and removed from the network, the selection probability of its neighbors and neighbors' neighbors decreases, effectively reducing influence redundancy. To capture the impact of topological overlap between second-order neighborhoods, Zhao et al. [37] proposed a second-order neighborhood (SN) index. This index characterizes the importance of edges in the network. Yu et al. [38] considered the overlap of communities within edge neighborhoods and introduced a novel and effective index called subgraph overlap (SO). To quantify influence redundancy, Wang et al. [39] presented a theoretical method that measures the influence redundancy from the perspective of non-spreader nodes based on their infected probability. They calculate the total influence exerted on each non-spreader node to quantify influence redundancy. Furthermore, Zhou et al. [40] conducted a deeper analysis of the influence redundancy metric by investigating the perturbation of the principal eigenvector of the adjacency matrix.

3. Preliminary

3.1. Symbols and notations

Consider an undirected and unweighted network $G(V, E)$, where V and E denote the sets of nodes and edges respectively, with edge $e_{uv} = (u, v) \in E$, $u \in V$, $v \in V$, we define the adjacency matrix of G as $A = (a_{ij})_{N \times N}$, if an edge exists between node i and node j, then $a_{ij} = 1$, otherwise $a_{ij} = 0$. The terms λ_i denote the eigenvalues of A (with $|\lambda_1| \geq |\lambda_2| \geq \cdots |\lambda_N| \geq 0$). We select a set of key nodes $S \subset V$ from G, and obtain the remaining network $G'(V, E')$ by removing the edges attached to the key nodes. The adjacency matrix A' of the remaining network is $A' = A - R$ where $R = (r_{ij})_{N \times N}$, $r_{ij} = a_{ij}$ if $i \in S$ or $j \in S$, otherwise $r_{ij} = 0$. Similarly, λ'_i denotes the eigenvalues of A'.

In addition, we denote the path $p = \langle u, \dots, v \rangle$ as a sequence of connected nodes from u to v, and the path set $P^{(h)}_{u \longrightarrow v} = \{p | p = \langle u, \dots, v \rangle\}$ represents all paths of length h from u to v. Furthermore, the paths that pass through specific nodes are denoted as $P^{(h)}_{\overset{s}{u \longrightarrow v}} = \{p | p = \langle u, \dots, s, \dots, v \rangle\}$ and $P^{(h)}_{\overset{S}{u \longrightarrow v}} = \bigcup_{s \in S} P^{(h)}_{\overset{s}{u \longrightarrow v}}$. Table 1 summarizes the symbols and notations used in the paper.

Table 1
Important symbols used in this paper.

Symbol	Description
$G(V,E)$	An undirected and unweighted network
V	Node set
E	Edge set
N, M	Size of nodes and edges in network G
A, R	Adjacency matrix
$A' = A - R$	The adjacency matrix of the remaining network
λ_A	The largest eigenvalue of the matrix A
$\lambda_{A'}$	The largest eigenvalue of the matrix A'
S	The set of key nodes
$P^{(h)}_{u\longrightarrow v}$	All paths from u to v with length h
$P^{(h)}_{u\overset{s}{\longrightarrow}v}$	All paths from u to v that pass through s with length h
$P^{(h)}_{u\overset{S}{\longrightarrow}v}$	All paths from u to v that pass through at least one node in S with length h
$D^{(h)}_S$	The lost paths by removing the edges attached to key nodes with length h
$\mathcal{N}^{(h)}_u$	The neighbor node set of u with distance h

3.2. Problem definition

As described in Section 1, node immunization is employed as a preventive measure against the dissemination of malicious information. In this study, we utilize the SIR mode to characterize the spreading dynamics of malicious information within the network. The SIR model classifies individuals into three states: Susceptible (S), Infected (I), and Recovered (R). At each time step t, infected individuals have a probability α of infecting their susceptible neighbors, while at the same time, infected individuals have a probability γ of recovering from the infection. Subsequently, recovered individuals may lose their immunity with a probability β and transition back to a susceptible state. The determinative factor for the spread of malicious information across the network lies in its threshold, which is characterized by the reciprocal of the largest eigenvalue [28]. A higher eigenvalue corresponds to superior immunization performance. It is worth noting that the largest eigenvalue also governs immunization performance in other similar information spreading models [33,41,42].

Apart from the epidemic SIR model, as mentioned in ref [28], the largest eigenvalue derived from the network's adjacency matrix is a useful metric for determining the epidemic threshold in cascading models on various types of networks. Furthermore, larger eigenvalues indicate higher levels of network connectivity. Consequently, in order to mitigate the impact of malicious information, it becomes crucial to safeguard specific key nodes that hinder the propagation of such information. In this study, our objective is to minimize the largest eigenvalue $\lambda_{A'}$ of the residual network by selecting a budget size k of key nodes S. We refer to this task as the *Eigenvalue Minimization Problem* (EMP).

Problem 1. EMP(G,k): How to find a budget size k of key nodes to minimize $\lambda_{A'}$?

Input: The adjacency matrix A of a network and a budget integer k.

Output: A subset S of k nodes that, when removed, could minimize $\lambda_{A'}$.

Theorem 1. *EMP is NP-hard.*

Proof. See Appendix A. □

3.3. Challenge

Performance quality: In the identification of key nodes, it is observed that different traditional methods exhibit significant performance variations within a single network. Additionally, a specific method may demonstrate fluctuating performance across different networks. This inconsistency arises primarily due to the fact that most existing methods solely focus on individual node importance, while neglecting the crucial aspect of importance interaction between nodes, also known as influence redundancy [35]. Therefore, it becomes imperative to systematically analyze influence redundancy and incorporate it into the key node identification process. By considering the influence interactions between nodes, we can enhance the accuracy and robustness of key node identification methods.

Time complexity: According to Theorem 1, it has been established that Problem 1 is NP-hard. A straightforward approach is to use a simple greedy algorithm that selects the best node in each round. However, the natural algorithm is computationally intractable, as its time complexity is $O(\binom{N}{k}M)$ [4]. Although some advanced methods have been proposed to reduce the time complexity to $O(Nk^2 + M)$, they still suffer from precision loss and significant time consumption, especially in dense networks [4]. Hence, there is a compelling necessity to develop faster methods that can maintain performance quality. Efficient algorithms are required to strike a balance between computational speed and accuracy, making them more practical and applicable for large-scale network analysis.

4. The proposed method

4.1. Motivation

In traditional network analysis, the selection of key nodes primarily hinges on their individual significance, often neglecting the critical aspect of influence redundancy [8,11,18]. This concept of influence redundancy emerges when key nodes, as exemplified by Nodes A and B in Fig. 1, exhibit overlapping roles within the network's structure. More precisely, these nodes can concurrently act in vital roles in the spread of information, thereby creating intersecting spheres of influence. This intersection leads to an overlap of potential information dissemination pathways. Existing methods fall short of adequately addressing this redundancy, which could lead to sub-optimal strategies in network immunization. Our approach seeks to evaluate the influence of nodes in a more holistic manner, incorporating the dynamics of influence redundancy.

4.2. Analysis of objective function

According to the EMP, the objective is to determine the optimal key nodes S to minimize the $\lambda_{A'}$. $\lambda_{A'}$ actually represents the growth rate of an arbitrary vector $\mathbf{x}_0$ after n iterations of A'. It can be calculated using the expression $|\mathbf{x}_n| = |A'^n\mathbf{x}_0| = (\mathbf{x}_0^T A'^{2n}\mathbf{x}_0)^{\frac{1}{2}} \sim e^{n log \lambda_{A'}}$ [22], $\lambda_{A'}$ is then calculated by the power method:

$$\lambda_{A'} = \lim_{n\to\infty}\left(\frac{|A'^n\mathbf{x}_0|}{|\mathbf{x}_0|}\right)^{\frac{1}{n}} \approx \left(\frac{\mathbf{x}_0^T A'^{2n}\mathbf{x}_0}{|\mathbf{x}_0|^2}\right)^{\frac{1}{2n}}. \tag{1}$$

In our case, we set $\mathbf{x}_0 = \mathbf{1}$ and $h = 2n$, for a finite value of h. As h increases, the right hand side of Eq. (1) converges rapidly to $\lambda_{A'}$. However, it is important to note that the computational complexity sharply increases with the value of h.

Lemma 1. *We define*

$$\Delta\lambda(S,h) = \mathbf{1}^T A^h\mathbf{1} - \mathbf{1}^T(A-R)^h\mathbf{1}, \tag{2}$$

where h is a positive number and R is the adjacency matrix of S, $R_{ij} = 1$ is either $i \in S$ or $j \in S$; otherwise $R_{ij} = 0$. The problem of EMP is equivalent to maximizing $\Delta\lambda(S,h)$.

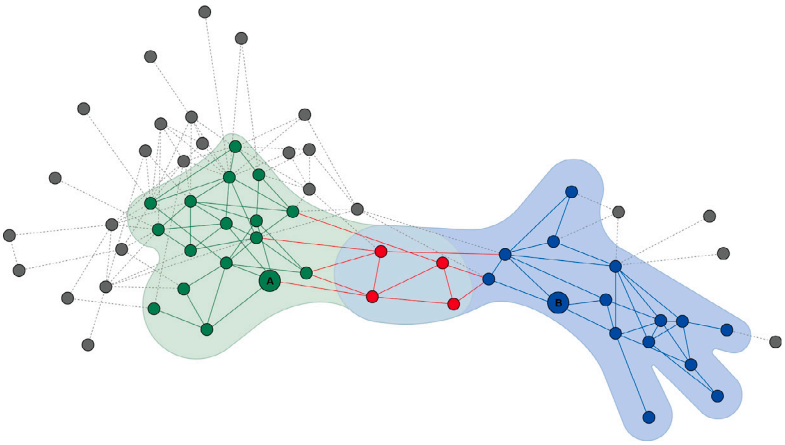


Fig. 1. An example of influence redundancy in a network. The nodes in the network represent individuals and the edges represent the existence of contact relationships. In the beginning, A and B are victim sources and malicious information spreads through the network. Within a short period of time, green nodes and blue nodes become new victims activated by A and B respectively. While red nodes are simultaneously influenced by both the green and blue nodes.

Proof. EMP requires to choose the best nodes S to minimize $\lambda_{A'}$. Minimizing $\lambda_{A'}$ is equivalent to maximizing $\Delta\lambda = \lambda_A - \lambda_{A'}$. Here, we use $\Delta\lambda(S,h)$ to approximate $\Delta\lambda$. Supposing that $\mathbf{1}^T = \alpha_1 \mathbf{v}_1' + \cdots + \alpha_N \mathbf{v}_N'$, we have

$$\begin{aligned}\left(\mathbf{1}^T A'^h \mathbf{1}\right)^{\frac{1}{h}} &= \lambda_1' \left[\alpha_1^2 + \alpha_2^2 \left(\frac{\lambda_2'}{\lambda_1'}\right)^h + \cdots + \alpha_N^2 \left(\frac{\lambda_N'}{\lambda_1'}\right)^h\right]^{\frac{1}{h}} . \\ &< \lambda_1' \left[\alpha_1^2 + \left(N - \alpha_1^2\right)\left(\frac{\lambda_2'}{\lambda_1'}\right)^h\right].\end{aligned} \tag{3}$$

Since $|\frac{\lambda_2'}{\lambda_1'}| < 1$, $\left(\mathbf{1}^T A'^h \mathbf{1}\right)^{\frac{1}{h}}$ approximates $\lambda_{A'}$ with exponential decay with h, $\left(\mathbf{1}^T A'^h \mathbf{1}\right)^{\frac{1}{h}} \propto \lambda_1' + O(\left(\frac{\lambda_2'}{\lambda_1'}\right)^h)$. Then we translate the EMP into the following problem:

Problem 2. Given a positive integer h, how to find a budget size k of key nodes to maximize $\Delta\lambda(S,h)$.

Input: The adjacency matrix A of a network, a positive integer h and a budget integer k.

Output: A subset S of k nodes that maximize $\Delta\lambda(S,h)$.

We transform the NP-hard problem of EMP into the NP-complete problem. The solution to Problem 2 also implies a near-optimal solution to the EMP, thereby leading us to Lemma 1. When $h \to +\infty$, $\lim_{h\to+\infty} \Delta\lambda(S,h) = \lambda_A - \lambda_{A'}$. However, for finite h, $\Delta\lambda(S,h)$ could still achieve high precision. We validate the influence of h in the experiment. □

Theorem 2. *Maximizing $\Delta\lambda(S,h)$ is NP-complete.*

Proof. See Appendix B. □

Definition 1. $\Delta\lambda(S,h)$ can be characterized by the number of paths that pass through the key nodes, that is

$$\Delta\lambda(S,h) = |D_S^{(h)}| = \sum_{i,j\in V, P_{i\longrightarrow j}^{(h)} \subset D_S^{(h)}} |P_{i\longrightarrow j}^{(h)}|, \tag{4}$$

where $|D_S^{(h)}|$ represents the lost paths by removing the edges attached to key nodes with length h, and $|P_{i\longrightarrow j}^{(h)}|$ represents the number of different paths from node i to j with length h, i.e., $|P_{i\longrightarrow j}^{(h)}| = \mathbf{a}_{ij}$, with $A^h = (\mathbf{a}_{ij})_{N\times N}^h$. The quantity $\mathbf{1}^T A^h \mathbf{1}$ denotes the number of paths from node i to j with length h. Similarly, A'^h and $\mathbf{1}^T A'^h \mathbf{1}$ can be represented. After $\mathbf{1}^T A^h \mathbf{1} - \mathbf{1}^T A'^h \mathbf{1}$, we obtain $D_S^{(h)}$. It is important to note that the removal of key nodes leads to the loss of all paths passing through these nodes, which is represented by $D_S^{(h)}$. By simplifying Eq. (2), we obtain Eq. (4).

Here, we introduce the term $\Delta\lambda(S,h)$ as the 'Set Influence', which represents the number of paths in the network that traverse at least one node from a predefined set of key nodes. It is important to note that certain paths in the network might pass through multiple key nodes, and the removal of any of these nodes would result in the loss of those paths. To account for this phenomenon, we acknowledge the presence of redundancy among the key nodes. We define the 'influence overlap' as the paths that traverse more than one key node, which provides a quantitative measure of the extent of influence redundancy.

Definition 2. The *influence overlap* is defined as the cumulative influence of each key node minus the set influence:

$$\begin{aligned}\Delta\lambda(S,h)_{ov} &= \Delta\lambda(S,h)_{cum} - \Delta\lambda(S,h) \\ &= \sum_{i,j\in V} |P_{i\xrightarrow{S} j}^{(h)}| - \sum_{i,j\in V, P_{i\longrightarrow j}^{(h)} \subset D_S^{(h)}} |P_{i\longrightarrow j}^{(h)}|,\end{aligned} \tag{5}$$

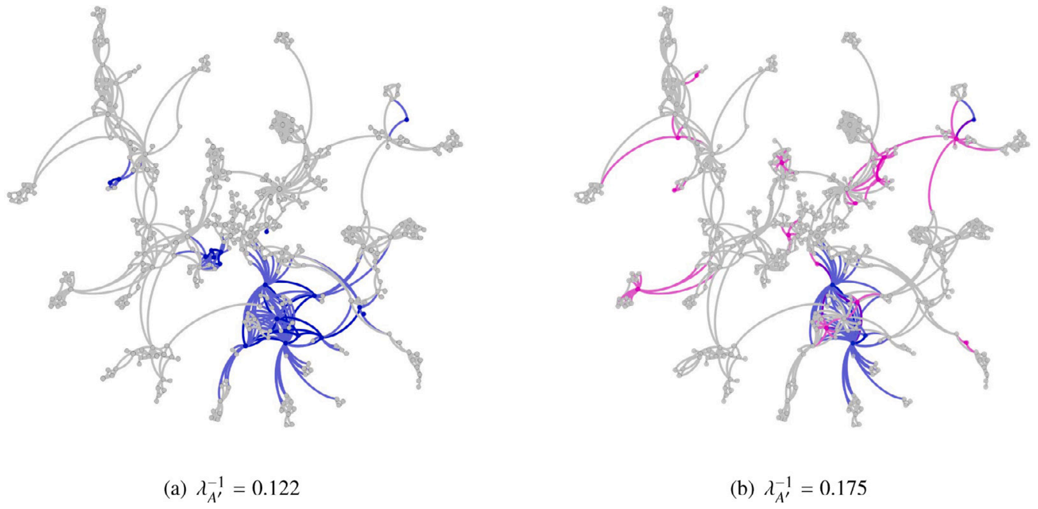
where the cumulative influence $\Delta\lambda(S,h)_{cum} = \sum_{i,j\in V} |P_{i\xrightarrow{S} j}^{(h)}|$ represents all paths from node $i \in V$ to other nodes $j \in V$ that pass through at least one key node in S with length h.

In Definition 2, it is specified that each path should be counted only once. Nevertheless, when a path crosses multiple key nodes, there is a possibility of double-counting in the cumulative influence calculation. The additional instances of counting in such cases refer to the influence overlap. Hence, the concept of influence overlap provides a quantitative measure of the extra instances of path counting that arise due to the inclusion of multiple key nodes in the path.

4.3. SIA: The optimization algorithm

In this section, the proposed optimization algorithm is introduced, followed by an analysis of its accuracy and efficiency (see Fig. 2). To address Problem 2, the Set Influence Algorithm (SIA) is proposed as a solution, which is outlined in Algorithm 1. The algorithm commences by initializing an empty set, denoted as S. Subsequently, at each iteration, a node $u \notin S$ is chosen and included in S in a manner that maximizes the increment of $\Delta\lambda(S\cup\{u\}, h)$. This process is repeated until a predetermined number of nodes have been added to S.

Lemma 2. *The time complexity of the SIA method is $O(kN^4 \log h)$, and the space complexity is $O(N^2 + k)$.*



(a) $\lambda_{A'}^{-1} = 0.122$

(b) $\lambda_{A'}^{-1} = 0.175$

Fig. 2. (a) Key nodes(blue) selected by the eigenvector centrality(EC) method in the Science network. (b) When the parameter $h = 10$, key nodes are selected from the network by the SIA method, where the nodes selected by SIA but not by EC are marked in magenta. Compared with the nodes selected by the EC method, the nodes selected by SIA are scattered uniformly, which implies that their influence redundancy is less.

Algorithm 1: Set Influence Algorithm (SIA)

```
Input: the adjacency matrix $A$, an integer $h$ and the budget size $k$
Output: key node set $S$
Initialize $S = \emptyset$;
for $count = 1$ to $k$ do
    for $u \notin S$ do
        Calculate matrix $R$ that correspond to $u$;
        $\Delta\lambda(u,h) = \mathbf{1}^T A^h \mathbf{1} - \mathbf{1}^T (A-R)^h \mathbf{1}$.;
    end
    Add $i = argmax_i \Delta\lambda(i,h)$ to $S$;
    // Remove $i$ from $A$
    $A(i,:) = 0$;
    $A(:,i) = 0$;
end
return $S$
```

Proof. The SIA algorithm utilizes a greedy approach to select nodes that have not yet been included in set S, and at each step, calculates $\Delta\lambda(u,h)$. The time complexity for calculating $\Delta\lambda(u,h)$ is determined to be $O(N^3 \log h)$, while the time complexity for N rounds of matrix multiplication for $\Delta\lambda(u,h)$ is $O(N^4 \log h)$ is estimated to be $O(kN^4 \log h)$. The selection process continues until a fixed number, denoted as k, of nodes has been added to set S. Consequently, the overall time complexity of the SIA algorithm amounts to $O(kN^4 \log h)$.

The algorithm necessitates the storage of the adjacency matrix A, which occupies $O(N^2)$ space. In each iteration, the algorithm selects a node $i \notin S$, thus necessitating the tracking of the nodes in S, requiring an additional $O(k)$ space. The computation of $\Delta\lambda(u,h)$ necessitates the storage matrix A^h, which can be dynamically computed through the deletion of rows and columns that correspond to the nodes in S. This operation requires an additional $O(N^2)$ space. Thus, the space complexity of the SIA algorithm amounts to $O(N^2 + k)$. □

Despite the simplicity of Algorithm 1, its computational complexity poses challenges when applied to large graphs. To mitigate this drawback, we have concentrated our efforts on setting $h = 3$, thereby achieving a reduction in complexity. Within this framework, we have successfully obtained the decline of $\Delta\lambda(S,3)$ for the removal of a single node, as outlined below:

Algorithm 2: SIA-3

```
   Input: the adjacency matrix $A$ and the budget size $k$
   Output: key node set $S$
1  Initialize $S = \emptyset$ ;
2  Compute the degree; let $d$ be the corresponding degree $d(j)(j = 1, \dots, N)$;
3  for each $u \in V$ do
4      Using Eq. (6) to calculate $\Delta\lambda(u,3)$ that correspond to $u$;
5  end
6  while $|S| < k$ do
7      Add $i = argmax_{i,i \notin S} \Delta\lambda(i,3)$ to $S$;
8      // Remove $i$ from $A$
9      $A(i,:) = 0$;
10     $A(:,i) = 0$;
11     Update $d(j)$ that $j \in \mathcal{N}_i^{(1)}$ ;
12     for each $u \in \mathcal{N}_i^{(3)}$ do
13         Using Eq. (6) to update $\Delta\lambda(u,3)$ that correspond to $u$;
14     end
15 end
16 return $S$
```

$$\Delta\lambda(u,3) = \sum_{i \in \mathcal{N}_u^{(1)}} a_{ui} \Delta d_u \Delta d_i + \sum_{i \in \mathcal{N}_u^{(1)}, j \in \mathcal{N}_i^{(1)}} a_{ij} \Delta d_i d_j, \tag{6}$$

In Eq. (6), where Δd_u and Δd_i represent the changes in the degrees of u and its neighbor node i, resulting from the removal of node u from the network.

In Algorithm 2, the SIA-3 method is proposed. The algorithm initializes an empty set $S = \emptyset$ and calculates the degree of each node (lines 1–2). Then, at the first iteration, it calculates the drop of each node (lines 3–5). From lines 6–14, the algorithm iteratively adds the key node that contributes the maximum drop to $\Delta\lambda(S,3)$ (line 7), and removes that node from the network (lines 9–10). Additionally, it updates the degree of each neighbor of the key node (line 11). This removal of the key node results in a change in $\Delta\lambda(u,3)$, so the algorithm also updates the drop of the affected nodes (lines 12–14).

Theorem 3. *The objective function $\Delta\lambda(S,h)$ is monotonically non-decreasing and submodular. As a result, Algorithm 1 and Algorithm 2* *possess the capability of achieving an approximation ratio of at least* $(1-1/e)$ *with respect to the optimal* $\Delta\lambda(S,h)$*.*

Proof. We use the property of submodular function to show the proof. A submodular function is a set function $f : 2^V \to \mathbb{R}$, satisfying the following condition: for any subset $S \subseteq V$ and any of its supersets T and any element $u \in V \setminus T$, $f(S \cup \{u\}) - f(S) \geq f(T \cup \{u\}) - f(T)$. The submodular function has pretty property [4,43]: Let S be the node set by simple greedy algorithm(Algorithms 1 or 2) and let $\tilde{S}$ be the best set. Then $\Delta\lambda(S,h) > (1-1/e)\Delta\lambda(\tilde{S},h)$. In the following part, we first prove that $\Delta\lambda(S,h)$ satisfies submodular function and then use the property to get the theorem.

As for $\Delta\lambda(S,h)$, the contribution of an individual node u to $\Delta\lambda(S,h)$ is defined as $\Delta\lambda(u,h) = \Delta\lambda(S \cup \{u\},h) - \Delta\lambda(S,h)$. And, according to Eq. (5), $\Delta\lambda(S,h) = \Delta\lambda(S,h)_{cum} - \Delta\lambda(S,h)_{ov}$. It is obvious that

$$\begin{aligned}&\Delta\lambda(S \cup \{u\},h)_{cum} - \Delta\lambda(S,h)_{cum}\\ &\geq \Delta\lambda(S \cup \{u\},h)_{ov} - \Delta\lambda(S,h)_{ov}.\end{aligned} \tag{7}$$

Therefore, with the addition of new node u, $\Delta\lambda(S,h)$ is non-decreasing. Next, we show $\Delta\lambda(S,h)$ is submodular. Accordingly, we have

$$\begin{aligned}&(\Delta\lambda(S \cup \{u\},h) - \Delta\lambda(S,h))\\ &-(\Delta\lambda(T \cup \{u\},h) - \Delta\lambda(T,h))\\ &=(\Delta\lambda(T \cup \{u\},h)_{ov} - \Delta\lambda(T,h)_{ov})\\ &-(\Delta\lambda(S \cup \{u\},h)_{ov} - \Delta\lambda(S,h)_{ov}).\end{aligned} \tag{8}$$

There are two cases for the newly added node u: (a) node u overlaps with the key nodes. (b) node u does not overlap with the key nodes. It is observed that as the number of key nodes increases, the influence overlap also increases. Consequently, compared to a node $u \in S$, a node $u \in T$ will exhibit higher influence overlap. we have

$$\begin{aligned}&\Delta\lambda(T \cup \{u\},h)_{ov} - \Delta\lambda(T,h)_{ov}\\ &\geq \Delta\lambda(S \cup \{u\},h)_{ov} - \Delta\lambda(S,h)_{ov}.\end{aligned} \tag{9}$$

Therefore, the objective function is submodular. According to the property of submodular function [43], our algorithms have $(1-1/e)$ approximation ratio, which completes the proof. □

Lemma 3. *The time complexity of the SIA-3 is* $O(N\langle d\rangle^2 + N\log N + k(\langle d\rangle^3 + \langle d\rangle \log N))$*. For sparse networks, the time complexity is* $O(N\log N)$*.*

Proof. For the initialization, lines 1–2 calculate the degree of each node, which has time complexity $O(M)$. For lines 3–5, we first calculate the drop of each node which requires time complexity $O(N\langle d\rangle^2)$. Meanwhile, we maintain a heap with nodes in the network, and the node u of the heap stores its drop value, which represents its contribution to $\Delta\lambda(S,3)$. The created heap to sort the contribution of each node has time complexity $O(N\log N)$. Lines 12–14 update the drop of nodes adjacent to the removed node and readjust the heap, which requires time complexity $O(\langle d\rangle^3 + \langle d\rangle \log N)$. Therefore, the overall time complexity is $O(N\langle d\rangle^2 + N\log N + k(\langle d\rangle^3 + \langle d\rangle \log N))$. In addition, for large-scale sparse networks, $\langle d\rangle$ could be considered as constants, and the time complexity could become $O(N\log N)$. □

Lemma 4. *The space complexity of the SIA-3 method is* $O(N+M)$*.*

Proof. In Algorithm 2, we need memory space $O(M)$ for matrix A, and $O(N)$ to store the degree and drop value of each node. Therefore, the total space complexity is $O(N+M)$. □

5. Experiment

This section presents an evaluation of the proposed method on empirical networks. The assessment encompasses a description of the data sets and evaluation metrics employed. Subsequently, a comprehensive comparison between the proposed method and other state-of-the-art techniques is conducted. Finally, an analysis of the parameter h in the proposed method is provided.

5.1. Dataset

We conducted experiments using a total of 20 empirical networks[1] for evaluation purposes. These networks can be briefly described as follows: (1) Email: Reflecting the email communications among members of Rovira I Virgili University. (2) Caster: Depicting a social network among users of the caster.com website. (3) DaysAll: Comprising the main connectivity components obtained by converting the Reuters Terrorist News Network into a combined network across all 66 days (syndication at all time points). (4) Email-enron: Constructed from a dataset of email addresses, where each node represents an email address and edges denote communication between two addresses. (5) Euroroad: A representation of the international E-road network with nodes indicating cities and edges indicating connectivity between them. (6) Hamster: Reflecting the friendship network among users of the hamsterster.com website. (7) LastFm: Derived from the LastFm user social network, collected from the public API in March 2020. (8) Vidal: Representing an initial version of a proteome-scale map illustrating binary proteinCprotein interactions in humans. (9) Yeast: An undirected network encompassing protein interactions within yeast, with nodes representing proteins and edges representing metabolic interactions between them. (10) Science-email: Focusing on email communication between scientists. (11) Minnesota: Depicting the road network of Minnesota. (12) USAir97: Illustrating the airline network between US airports in 1997. (13) Email-univ: Emphasizing a university's email network. (14) Science: Capturing the collaborative network of scientists. (15) Openflights: Comprising flights collected by the OpenFlights.org project, where each node represents an airport and edges indicate flights operated by specific airlines. (16) Dogster: Reflecting friendships between users of the website dogster.com. (17) Astrophysics: Derived from the co-authorship network within the “astrophysics” section (Astro-ph) of arXiv, with nodes representing authors and edges representing collaborations. (18) Flickr: An undirected network where Flickr images share public metadata, with nodes denoting images and edges indicating common metadata between them. (19) Brightkite: Composed of user-user friendships from Brightkite. (20) CiteSeer: Extracted from CiteSeer, representing an authorship network where nodes correspond to publications, and edges indicate that an author has published a given publication. The statistical characteristics of these 20 empirical networks are presented in Table 2.

5.2. Baseline methods

In our experiment, we compare the performance of the SIA method with 11 state-of-the-art methods: high-degree(HD) [8], betweenness centrality(BC) [10], PageRank(PR) [17], eigenvector centrality(EC) [18], K-shell [11], collective influence(CI) [22], non-backtracing matrix(NBM) [44], FINDER [23], IMM [15], SUBSIM [13], and DeepIM [16].

5.3. Evaluation metrics

The performance of the proposed method is evaluated in three spreading models: the eigenvalue minimization problem, the IC model [7], and the SIR spreading model.

Eigenvalue minimization problem: The eigenvalue minimization problem is to minimize the largest eigenvalue of the network. Since the reciprocal of the largest eigenvalue of the remaining network

[1] Konect Dataset Collection, http://konect.cc/networks/

Table 2
Basic description of network properties that include the network size (N), total edge numbers (M), the average degree ($\langle d\rangle$), the maximum degree (d_{max}), the clustering coefficient ($\langle C\rangle$), the average length between nodes ($\langle L\rangle$), and the sparsity ($2M/(N(N-1))$).

Dataset	N	M	$\langle d\rangle$	d_{max}	$\langle C\rangle$	$\langle L\rangle$	Sparsity
Email	1133	5451	9.62	71	0.22	3.61	8.5×10^{-3}
Caster	4438	22 652	10.21	3171	0.41	2.54	4.6×10^{-3}
DaysAll	3217	13 332	8.29	575	0.31	3.25	5.2×10^{-3}
Email-enron	143	623	8.71	42	0.43	2.97	6.1×10^{-2}
Euroroad	1174	1417	2.51	10	0.033	19.18	2.1×10^{-3}
Hamster	1858	12 534	13.49	272	0.09	3.39	7.3×10^{-3}
Lastfm	2272	5577	4.91	125	0.21	6.39	4.3×10^{-3}
Vidal	3133	6726	4.29	129	0.035	4.80	1.4×10^{-3}
Yeast	1870	3896	2.44	56	0.055	7.07	2.2×10^{-3}
Science-email	998	2030	2.03	45	0.004	11.62	4.1×10^{-3}
Minnesota	2152	5398	2.51	5	0.017	32.31	2.3×10^{-3}
USAir97	332	2126	12.81	139	0.62	2.74	3.9×10^{-2}
Email-univ	1100	5500	5.602	71	0.16	3.57	9.1×10^{-3}
Science	379	1828	4.82	34	0.74	6.04	2.6×10^{-2}
Openflights	2903	30 501	10.78	242	0.397	4.10	7.2×10^{-3}
Dogster	260 390	2 148 179	16.49	22 139	0.01	3.36	6.3×10^{-5}
Astrophysics	16 046	121 251	15.11	360	0.43	15.11	9.1×10^{-4}
Flickr	105 938	2 316 948	43.74	5425	0.40	43.74	4.1×10^{-4}
Brightkite	58 228	214 078	7.35	1134	0.11	4.86	1.3×10^{-4}
CiteSeer	286 748	512 267	3.57	385	0.05	6.35	0.9×10^{-4}

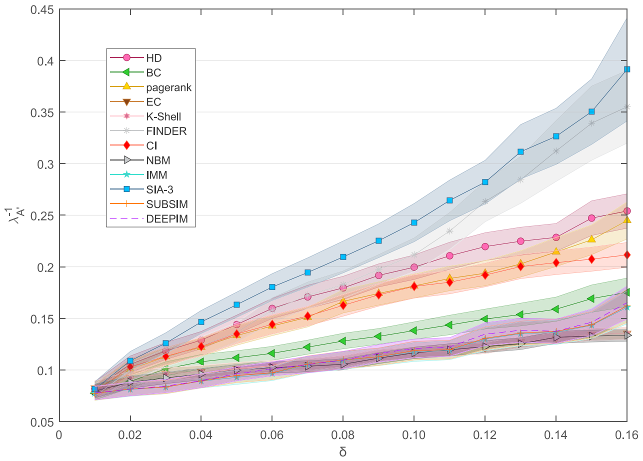


Fig. 3. The reciprocal of the largest eigenvalue $\lambda^{-1}_{A'}$ of the remaining network as a function of the proportion δ of key nodes, $\delta = k/N$. The result is the average on 20 empirical networks.

determines the immunization performance, the evaluation metric of this problem is $\lambda^{-1}_{A'}$.

IC model simulation: The fraction g of the giant component of the independent cascade(IC) model is an important measure of cascade failure. In the IC simulation, we use the identified nodes as initial activated seeds. When the spread of information finishes, we count the number of inactive nodes. Since the function of a network is usually determined by its giant component, we use the giant component of inactive nodes as the evaluation.

SIR model simulation: The evaluation metric of the SIR model is the final coverage of propagation, which is defined as:

$$F(t) = \frac{N_{victims}(t)}{N}, \tag{10}$$

where N is the number of nodes, $N_{victims}(t)$ denote the number of nodes infected and recovered at time t. A smaller $F(t)$ indicates a more effective immunization strategy, which means better key nodes.

5.4. Parameter settings

The parameters are set as follows:

- *SIA*. The method features a crucial adjustable parameter h, which is systematically evaluated across a range spanning from 1 to 10 in the experiments focused on the eigenvalue minimization problem. Furthermore, for the comparative analysis with other baseline methods, a meticulous selection process leads us to set h at the optimal value of 3.
- *PageRank*. There is a damping factor parameter in the method, which we set 0.85.
- *CI*. The objective function in the method has one parameter of radius ℓ that we set 2.
- *IMM*. The method has two parameters, the propagation probability p and the number of samples mc. In all experiments, we set p_i of the sampled node i as $p_i = \frac{1}{d_i}$, mc is obtained by the calculation described in [45].
- *SUBIM*. We use the parameters of the original paper [13].
- *DeepIM*. We use the parameters of the original paper [16].
- *SIR model*. In the SIR model, people can be in one of three states: Susceptible, Infected, or Recovered. At each time step, infected people can infect their susceptible neighbors with probability α. Meanwhile, the infected people can recover with probability γ. Recovered people can also lose immunity with probability β and become susceptible again. Here, we set (1) $\alpha = 0.1$, $\gamma = 0.5$, $\beta = 0$. (2) 0.01 N initially infected nodes and 0.16 N immunization nodes, the immunization nodes were selected by different methods.

For the other methods, the parameters were set the same as the original papers [10,11,18,23].

5.5. Results

Here, we conducted an analysis on a set of 20 empirical networks. For each of these networks and for every evaluation metric employed, we extracted the performance of 11 different methods. By aggregating the overall performance of these 11 methods across the 20 empirical networks, we critically evaluated their rankings based on the average $\lambda^{-1}_{A'}$ (including error comparison), the average g, and the average $F(t)$.

Eigenvalue minimization problem: In Fig. 3, we compared the average scores of the SIA-3 with the other 11 methods on 20 empirical networks for the eigenvalue minimization problem. Fig. 2 vividly shows an example of the identified nodes. Our findings demonstrate that the SIA-3 outperforms all other methods across all 20 analyzed empirical networks. However, the performance of the CI method is relatively poor due to its neglect of the prevalent loop structures in empirical networks [22]. Although the CI method is known for its exceptional performance on highly sparse networks, it struggles to achieve optimality in the presence of abundant loops resulting from rich-club [46] and community phenomena evident in real networks. Similarly, the performance of the IMM method is unstable, possibly stemming from variations in the number of sampled random reverse reachable sets among the 20 empirical networks. FINDER demonstrates commendable performance on the eigenvalue minimization problem due to its ability to learn and generalize from small synthetic networks to a wide array of real-world networks. However, FINDER relies on the quality of the synthetic network generated by the utilized toy model for training, and its performance may be compromised if the distribution of the synthetic network significantly deviates from that of the real network. Another intriguing observation is that the simple heuristic PageRank method and HD method also display satisfactory performance. This can be attributed to these methods selecting a larger proportion of common nodes. Nonetheless, as the cardinality of $|S|$ increases, the substantial overlap in influence significantly diminishes their performance.

In the context of the SIA method, the accuracy of its calculation is influenced by the parameter h, wherein a larger value of h corresponds to a higher time complexity. To strike a balance between performance and time complexity, we conducted experiments focused on the eigenvalue minimization problem, specifically exploring the impact of varying values of h on the SIA method.

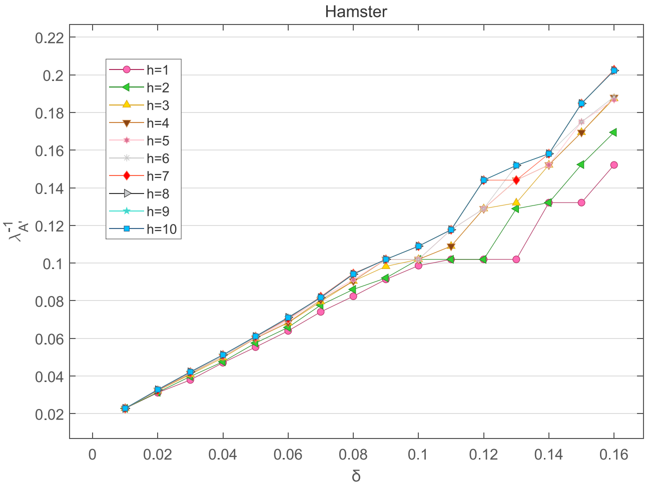


Fig. 4. The eigenvalue minimization problem evaluation on the Hamster network in the range of h from 1 to 10.

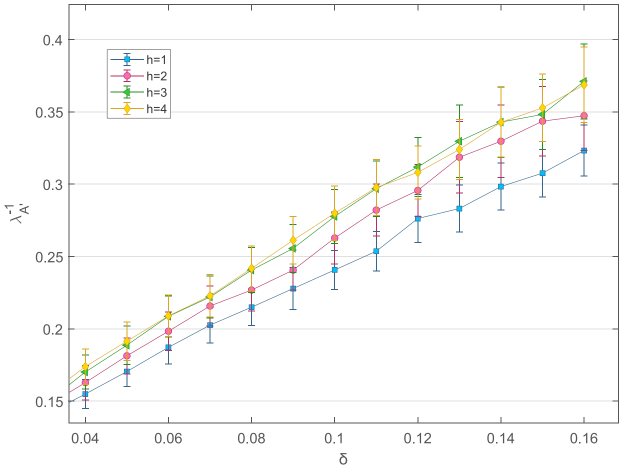


Fig. 5. The eigenvalue minimization problem evaluation on 20 empirical networks in the range of h from 1 to 4. The result is the average on 20 empirical networks.

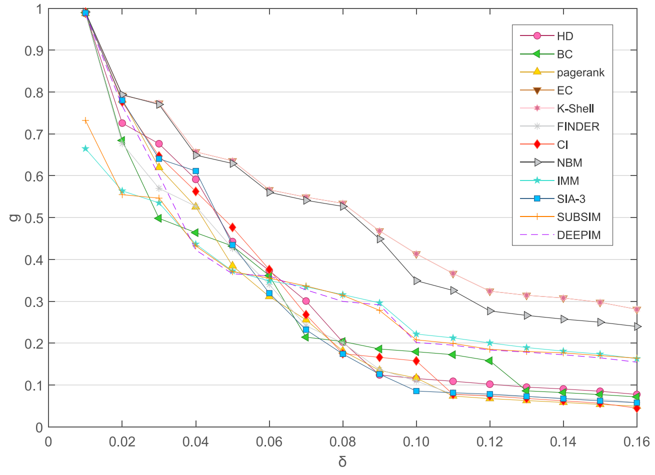


Fig. 6. The giant component g of the inactive nodes as a function of the proportion δ of key nodes under IC model. In the simulation, we use the identified nodes as initial activated seeds. When the spread of information finishes, we use the giant component of inactive nodes as the evaluation.

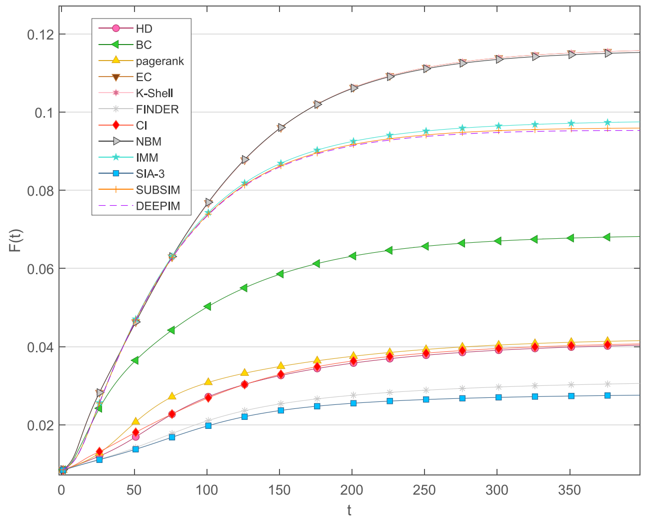


Fig. 7. The propagation $F(t)$ on SIR model at time step t. The result is the average on 20 empirical networks.

Fig. 4 shows the influence of h on $\lambda^{-1}_{A'}$ in Hamster network. In general, as h increases, $\lambda^{-1}_{A'}$ performs better. Another finding is that $h = 3$ corresponds to a significant improvement in performance compared to $h = 1, 2$. When $h > 3$, the performance gains are very small with the increase of h. Besides, we investigate the influence of h on more networks in Fig. 5. Here, we only consider $h = 1, 2, 3, 4$ because the time consumption increases sharply with h and large h is prohibitive for large networks. As shown in Fig. 5, there is a remarkable performance improvement of $h = 3$ compared to $h = 1$. But the performance gain of $h = 4$ is minimal compared to $h = 3$. In general, $h = 1$ corresponds to relatively low performance and low time complexity; for $h = 3, 4$ corresponds to relatively better performance and moderate time complexity, which scales for large networks.

Malicious information simulation: We proceeded to conduct a simulation experiment employing the IC model. In the IC simulation, we use the identified nodes as initial activated seeds. When the spread of information finishes, we count the number of inactive nodes. Since the function of a network is usually determined by its giant component, we use the giant component of inactive nodes as the evaluation in Fig. 6. In Fig. 6, it became evident that the SIA-3 method consistently exhibited near-optimal or optimal performance. Out of the 20 networks observed, the SIA-3 method outperformed the rest in 7 instances (35%), while the CI method achieved superior results in 4 instances (20%). It is notable that despite the fact that the SIA method was not explicitly tailored for IC model, it demonstrated remarkably close-to-optimal performance. This implies that the SIA method possesses superior generalization capabilities compared to alternative approaches.

Finally, we proceeded to conduct a simulation experiment utilizing the SIR model. By conducting 100 independent simulations, we were able to derive average scores. As depicted in Fig. 7, it becomes evident that the SIA-3 method consistently outperforms alternative methods as the number of susceptible nodes ($|S|$) increases. This can be attributed to the fact that SIA-3 is associated with a larger threshold. Higher epidemic thresholds pose greater difficulty in the propagation of malicious information. Interestingly, the experimental results indicate that the K-shell method exhibits the poorest performance. This can be attributed to its tendency to identify key nodes that are often clustered together, resulting in significant overlap of influence. Conversely, the SIA method not only selects nodes with substantial influence but also effectively reduces influence overlap.

To better understand the superiority of the SIA-3 method compared to the state-of-the-art methods, we compare the set influence of key

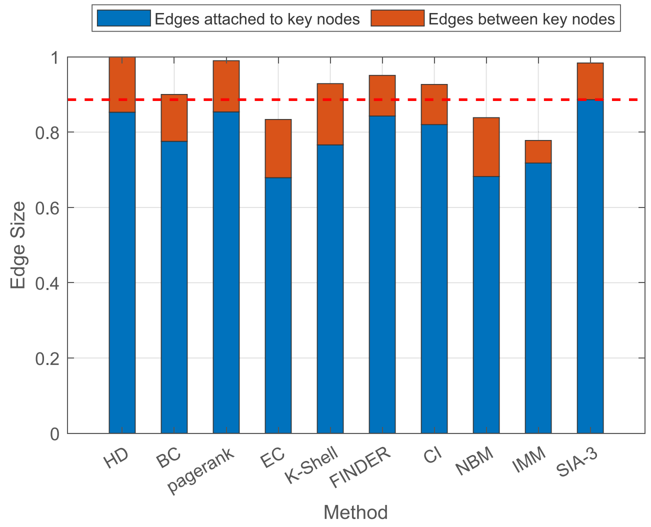


Fig. 8. We compared the number of edges attached to key nodes (set influence and height of blue bars) with the number of overlapping edges between key nodes (influence overlap and height of orange bars). We selected 18% of the key nodes in 9 networks. The SIA-3 method demonstrated a small percentage of influence overlap, whereas, for the HD centrality, BC, EC, K-shell, and NBM methods, a large percentage of influence overlap was observed. Although the HD and PR methods have a higher cumulative sum of node degrees (cumulative influence and total height of bars) than SIA-3, they have a smaller set influence than the SIA-3 due to the greater redundancy of their impacts.

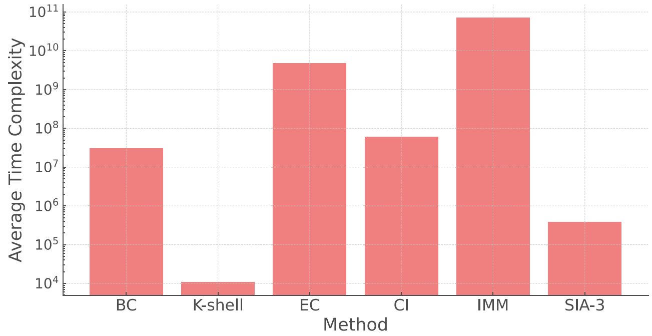


Fig. 9. We compared the time complexity of SIA-3 with some of the state-of-the-art methods in network datasets. The horizontal coordinate is the different methods and the vertical coordinate is the time overhead. The chart shows that SIA-3 has a significantly lower average time complexity compared to other methods.

nodes in 9 network datasets in Fig. 8. The set influence consists of two parts: the cumulative influence of a node and the influence overlap of the node (edges between key nodes). Since $\Delta\lambda(S,3) = \Delta\lambda(S,3)_{cum} - \Delta\lambda(S,3)_{ov}$, to maximize $\Delta\lambda(S,3)$, we should maximize $\Delta\lambda(S,3)_{cum}$ while minimizing $\Delta\lambda(S,3)_{ov}$. In Fig. 8, the SIA-3 method maximizes $\Delta\lambda(S,3)$, with the highest set influence, but with lower influence overlap. In contrast, the state-of-the-art methods consider only the cumulative influence of nodes and show considerable influence overlap of nodes. As a result, they do not maximize set influence.

To effectively illustrate the advantages of the SIA-3 method, we conducted a comparative analysis of run time across different methodologies within four distinct network datasets (Caster, Euroroad, Yeast, and Minnesota). The results in Fig. 9 highlight the superior performance of the SIA-3 method in comparison to its counterparts. Notably, in our experimental observations, both the CI and SIA-3 methods demonstrated impressive efficiency in terms of runtime, particularly within sparser network datasets. The SIA-3 method's time complexity was observed to closely approximate $O(N\,log\,N)$, marking a significant efficiency milestone. Additionally, it is worth mentioning that the IMM method, despite its effectiveness, necessitates a substantial number of samples to maintain its performance standards, consequently leading to a higher runtime. This comparison underscores the efficiency and robustness of the SIA-3 method in network analysis, especially in sparse datasets.

6. Conclusion

This study aims to examine the problem of identifying crucial nodes within intricate networks for the purpose of impeding the dissemination of malicious information. Our approach involves the utilization of the influence redundancy mechanism, which effectively captures the intricate interactions among these key nodes. Furthermore, we introduce an objective function that optimizes the selection of these key nodes by minimizing the maximum eigenvalue of the adjacency matrix. We demonstrate the submodularity of this function, enabling us to achieve a $(1 - 1/e)$ approximation ratio. Additionally, we propose the Set Influence Algorithm (SIA), an efficient algorithm capable of calculating the set influence of key nodes in large networks. The proposed methodology offers the following advantages: (a) providing an effective immunization strategy, (b) applicability to large networks, (c) excellent generalization capabilities, and (d) consistent outperformance of state-of-the-art methods. Thus, our approach presents a promising solution to the problem of identifying key nodes within complex networks.

CRediT authorship contribution statement

Mingyang Zhou: Writing – original draft, Methodology, Investigation, Conceptualization. **Hongwu Liu:** Writing – original draft, Methodology, Investigation, Data curation. **Hao Liao:** Writing – review & editing. **Gang Liu:** Writing – review & editing. **Rui Mao:** Writing – review & editing.

Declaration of competing interest

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

Data availability

I have shared links to my data.

Acknowledgments

The authors acknowledge financial support from the Shenzhen Fundamental Research-General Project, China (Grant Nos. 20220811155803001, JCYJ20190808162601658), National Natural Science Foundation of China (Grant Nos. 62276171, 62002233, 61972145), Guangdong Basic and Applied Basic Research Foundation, China (Grant Nos. 2024A1515011938, 2020B1515120028), Guangdong Peral River Recruitment Program of Talents, China (Grant Nos. 2019ZT08X603), and Swiftlet Fund Fintech funding.

Appendix A. Proof of Theorem 1

We prove that Problem 1 is NP-hard by constructing a polynomial reduction from a well-known NP-hard problem, the max k-hitting set problem (MaxHit(n, m, k)) [12], which is defined as follows.

Problem 3. Max k-Hitting Set Problem.

Input: (1) a set $\mathcal{U}$ of n elements; (2) a collection $B = \{\mathcal{B}_1, \mathcal{B}_2, \dots, \mathcal{B}_m\}$ of m distinct subsets of $\mathcal{U}$, which are not mutually exclusive; (3) a positive integer k.

Output: A set $\mathcal{H} \subseteq \mathcal{U}$ of k elements, with $\{\mathcal{B}_i | \mathcal{B}_i \cap \mathcal{H} \neq \phi\}$ having the maximum cardinality.

First step, without loss of generality, we assume that $1 < k < min\{n, m\}$. This assumption is reasonable when $k = 1$, MaxHit(n, m, k) can be trivially solved by picking the element in $\mathcal{U}$ with the most associated sets from B; when $k \geq m$, recalling the pigeonhole principle, we can hit all m sets with at most m elements from $\mathcal{U}$; when $k \geq n$, the entire element set $\mathcal{U}$ could be picked.

Given an instance of MaxHit(n, m, k) with $1 < k < min\{n, m\}$, we can construct a network G with n nodes, each corresponds to one element in $\mathcal{U}$, and each subset $\mathcal{B}_i \subseteq \mathcal{U}$, we construct a connected component G_i with $|\mathcal{B}_i|$ nodes, the nodes in G_i corresponding to the elements in $\mathcal{B}_i$. The removal of nodes in G_i would cause the largest eigenvalue drop of G_i until the corresponding component is invalid. Then, we can construct G as the union of m valid components as $G = G_1 \cup G_2 \cdots \cup G_m$. Since the sets in B are distinct and not mutually exclusive, the resulting valid component set cannot be independent. Therefore, the solution of MaxHit(n, m, k) would be equivalent to the solution of EMP(G, k), which completes the proof. □

Appendix B. Proof of Theorem 2

We prove that Problem 2 is NP-complete by constructing a polynomial reduction from a well-known NP-Complete problem, the vertex cover problem (VC(G, k)) [12], which is defined as follows.

Problem 4. Vertex Cover Problem.

Input: (1) A graph represents by an undirected and unweighted network $G(V, E)$; (2) a positive integer k.

Output: A subset of k vertexes, for any edge in G, with a vertex falling inside this subset (covering all edges with vertexes).

Given an instance of the Vertex Cover Problem, $G = (V, E)$, where V is the set of vertices and E is the set of edges, consider the following cases: Case 1: The edge set E is composed of a single edge, denoted as E_1, connecting vertices v_1 and v_2, where one must be selected between v_1 and v_2 for coverage. In this case, we create an instance of the $\Delta\lambda$-Max(G, h, k) problem, by setting $h = 1$ and k equal to the size of the minimum vertex cover of G. The problem instance is to find a subset S of k nodes such that $\Delta\lambda(S, 1)$ is maximized. Case 2: The edge set E is composed of edges connecting every pair of vertices among v_1, v_2, and v_3, denoted as E_2. Because these three vertexes are fully connected and form a triangle, any node coverage must take two of these three vertexes. In this case, we set $h = 2$ and the problem instance is to find a subset S of k nodes such that $\Delta\lambda(S, 2)$ is maximized. Case 3: The edge set E is composed of edges connecting a number of vertices greater than 3, denoted as $E_i (i > 2)$, where E_i can be regarded as a combination of the subproblems of case 1 and case 2. In this case, we set $h > 2$ and the problem instance is to find a subset S of k nodes such that $\Delta\lambda(S, h)$ is maximized. Therefore, the solution of the Vertex Cover problem, VC(G, k), would be equivalent to the solution of the $\Delta\lambda$-Max(G, h, k) problem, for each case. Since the Vertex Cover problem is NP-Complete, and we have proven that there exists a polynomial-time reduction from the Vertex Cover problem to the $\Delta\lambda$-Max(G, h, k) problem, we can conclude that the problem is also NP-Complete. □

References

[1] Y. Liu, H. Sanhedrai, G. Dong, L.M. Shekhtman, F. Wang, S.V. Buldyrev, S. Havlin, Efficient network immunization under limited knowledge, Natl. Sci. Rev. 8 (1) (2021) nwaa229.

[2] D. Helbing, Globally networked risks and how to respond, Nature 497 (7447) (2013) 51–59.

[3] D. Lian, Z. Gao, X. Song, Y. Li, Q. Liu, E. Chen, Training recommenders over large item corpus with importance sampling, IEEE Transactions on Knowledge and Data Engineering (2023).

[4] C. Chen, H. Tong, B.A. Prakash, C.E. Tsourakakis, T. Eliassi-Rad, C. Faloutsos, D.H. Chau, Node immunization on large graphs: Theory and algorithms, IEEE Trans. Knowl. Data Eng. 28 (1) (2015) 113–126.

[5] Y. Ren, M. Jiang, Y. Yao, T. Wu, Z. Wang, M. Li, K.-K.R. Choo, Node immunization in networks with uncertainty, in: 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering, TrustCom/BigDataSE, IEEE, 2018, pp. 1392–1397.

[6] R. Yan, D. Li, W. Wu, D.-Z. Du, Y. Wang, Minimizing influence of rumors by blockers on social networks: algorithms and analysis, IEEE Trans. Netw. Sci. Eng. 7 (3) (2019) 1067–1078.

[7] D. Kempe, J. Kleinberg, É. Tardos, Maximizing the spread of influence through a social network, in: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2003, pp. 137–146.

[8] H. Liao, M.S. Mariani, M. Medo, Y.-C. Zhang, M.-Y. Zhou, Ranking in evolving complex networks, Phys. Rep. 689 (2017) 1–54.

[9] L. Lü, D. Chen, X.-L. Ren, Q.-M. Zhang, Y.-C. Zhang, T. Zhou, Vital nodes identification in complex networks, Phys. Rep. 650 (2016) 1–63.

[10] R. Albert, I. Albert, G.L. Nakarado, Structural vulnerability of the North American power grid, Phys. Rev. E 69 (2) (2004) 025103.

[11] 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.

[12] C. Chen, R. Peng, L. Ying, H. Tong, Network connectivity optimization: Fundamental limits and effective algorithms, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 1167–1176.

[13] Q. Guo, S. Wang, Z. Wei, M. Chen, Influence maximization revisited: Efficient reverse reachable set generation with bound tightened, in: Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, 2020, pp. 2167–2181.

[14] S.M. Nikolakaki, A. Ene, E. Terzi, An efficient framework for balancing submodularity and cost, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 1256–1266.

[15] Y. Tang, Y. Shi, X. Xiao, Influence maximization in near-linear time: A martingale approach, in: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, 2015, pp. 1539–1554.

[16] C. Ling, J. Jiang, J. Wang, M.T. Thai, L. Xue, J. Song, M. Qiu, L. Zhao, Deep graph representation learning and optimization for influence maximization, in: Proceedings of the 40th International Conference on Machine Learning, ICML '23, JMLR.org, 2023.

[17] L. Page, S. Brin, R. Motwani, T. Winograd, The PageRank Citation Ranking: Bringing Order to the Web, Tech. Rep., Stanford InfoLab, 1999.

[18] A.M. Amani, M. Jalili, X. Yu, L. Stone, Finding the most influential nodes in pinning controllability of complex networks, IEEE Trans. Circuits Syst. II 64 (6) (2017) 685–689.

[19] C. Borgs, M. Brautbar, J. Chayes, B. Lucier, Maximizing social influence in nearly optimal time, in: Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2014, pp. 946–957.

[20] W. Chen, C. Castillo, L.V. Lakshmanan, Information and Influence Propagation in Social Networks, Springer Nature, 2022.

[21] M. Azaouzi, W. Mnasri, L.B. Romdhane, New trends in influence maximization models, Comp. Sci. Rev. 40 (2021) 100393.

[22] F. Morone, H.A. Makse, Influence maximization in complex networks through optimal percolation, Nature 524 (7563) (2015) 65–68.

[23] C. Fan, L. Zeng, Y. Sun, Y.-Y. Liu, Finding key players in complex networks through deep reinforcement learning, Nat. Mach. Intell. 2 (6) (2020) 317–324.

[24] L. Ma, Z. Shao, X. Li, Q. Lin, J. Li, V.C.M. Leung, A.K. Nandi, Influence maximization in complex networks by using evolutionary deep reinforcement learning, IEEE Transactions on Emerging Topics in Computational Intelligence 7 (4) (2023) 995–1009, http://dx.doi.org/10.1109/TETCI.2021.3136643.

[25] A. Logins, Y. Li, P. Karras, On the robustness of cascade diffusion under node attacks, in: Proceedings of the Web Conference 2020, 2020, pp. 2711–2717.

[26] S. Freitas, D. Yang, S. Kumar, H. Tong, D.H. Chau, Graph vulnerability and robustness: A survey, IEEE Trans. Knowl. Data Eng. (2022).

[27] M. Ahmad, S. Ali, J. Tariq, I. Khan, M. Shabbir, A. Zaman, Combinatorial trace method for network immunization, Inform. Sci. 519 (2020) 215–228.

[28] D. Chakrabarti, Y. Wang, C. Wang, J. Leskovec, C. Faloutsos, Epidemic thresholds in real networks, ACM Trans. Inf. Syst. Secur. 10 (4) (2008) 1–26.

[29] Y. Lin, W. Chen, Z. Zhang, Assessing percolation threshold based on high-order non-backtracking matrices, in: Proceedings of the 26th International Conference on World Wide Web, 2017, pp. 223–232.

[30] C. Chen, H. Tong, B.A. Prakash, T. Eliassi-Rad, M. Faloutsos, C. Faloutsos, Eigen-optimization on large graphs by edge manipulation, ACM Trans. Knowl. Discov. Data (TKDD) 10 (4) (2016) 1–30.

[31] J. Tariq, M. Ahmad, I. Khan, M. Shabbir, Scalable approximation algorithm for network immunization, 2017, arXiv preprint arXiv:1711.00784.

[32] Z. Zhang, Z. Zhang, G. Chen, Minimizing spectral radius of non-backtracking matrix by edge removal, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, 2021, pp. 2657–2667.

[33] R. Shang, W. Zhang, L. Jiao, X. Zhang, R. Stolkin, Dynamic immunization node model for complex networks based on community structure and threshold, IEEE Trans. Cybern. (2020).

[34] C. Wu, D. Lian, Y. Ge, M. Zhou, E. Chen, D. Tao, Boosting factorization machines via saliency-guided mixup, IEEE Transactions on Pattern Analysis and Machine Intelligence (2024).

[35] M.-Y. Zhou, W.-M. Xiong, X.-Y. Wu, Y.-X. Zhang, H. Liao, Overlapping influence inspires the selection of multiple spreaders in complex networks, Physica A 508 (2018) 76–83.

[36] J.-X. Zhang, D.-B. Chen, Q. Dong, Z.-D. Zhao, Identifying a set of influential spreaders in complex networks, Sci. Rep. 6 (1) (2016) 1–10.

[37] N. Zhao, J. Li, J. Wang, T. Li, Y. Yu, T. Zhou, Identifying significant edges via neighborhood information, Physica A 548 (2020) 123877.

[38] E.-Y. Yu, Y. Fu, J.-L. Zhou, D.-B. Chen, Finding important edges in networks through local information, in: 2021 7th International Conference on Computer and Communications, ICCC, IEEE, 2021, pp. 2225–2229.

[39] N. Wang, Z.-Y. Wang, J.-G. Liu, J.-T. Han, Maximizing spreading influence via measuring influence overlap for social networks, 2019, arXiv preprint arXiv:1903.00248.

[40] M. Zhou, J. Tan, H. Liao, Z. Wang, R. Mao, Dismantling complex networks based on the principal eigenvalue of the adjacency matrix, Chaos 30 (8) (2020) 083118.

[41] R. Pastor-Satorras, C. Castellano, P. Van Mieghem, A. Vespignani, Epidemic processes in complex networks, Rev. Modern Phys. 87 (3) (2015) 925.

[42] D.-p. Gao, N.-j. Huang, Threshold dynamics of an SEIR epidemic model with a nonlinear incidence rate and a discontinuous treatment function, Rev. R. Acad. Cienc. Exactas Fís. Nat. Ser. A Mat. 114 (1) (2020) 5.

[43] A. Krause, C. Guestrin, Near-optimal observation selection using submodular functions, in: AAAI, Vol. 7, 2007, pp. 1650–1654.

[44] F. Krzakala, C. Moore, E. Mossel, J. Neeman, A. Sly, L. Zdeborová, P. Zhang, Spectral redemption in clustering sparse networks, Proc. Natl. Acad. Sci. 110 (52) (2013) 20935–20940.

[45] W. Chen, An issue in the martingale analysis of the influence maximization algorithm imm, in: Computational Data and Social Networks: 7th International Conference, CSONET 2018, Shanghai, China, December 18–20, 2018, Proceedings 7, Springer, 2018, pp. 286–297.

[46] A. Ma, R.J. Mondragón, Rich-cores in networks, PLoS One 10 (3) (2015) e0119678.