Highly-efficient minimization of network connectivity in large-scale graphs

Anonymous Author(s)*

ABSTRACT

Network connectivity minimization is a fundamental problem in controlling the spread of virus in Internet and facilitating information propagation in online social networks. The problem aims to identify a budget number of key nodes whose removal would minimize the connectivity of a network. However, the existing solutions heavily rely on the number of edges, making it challenging to handle large and densely connected social networks. In this study, we present a fast algorithm that is independent of the number of edges. To achieve this, we first introduce a surrogate matrix that approximates the residual adjacency matrix with arbitrary small predefined error. We then devise an efficient approach for calculating the key nodes by optimizing the eigenvalues of the surrogate matrix. Remarkably, the algorithm has a small time complexity of $O(knr^3)$, with r being a small tunable number. Our algorithm thereby maintains a linear scalability in terms of the number of nodes and is unaffected by the number of edges. Hence, it has the capability to efficiently handle large and dense social networks. At last, we evaluate its performance against state-of-the-art techniques using diverse real-world datasets. The experimental results demonstrate the superiority of our proposed method in terms of both solution quality and computational efficiency.

CCS CONCEPTS

• **Mathematics of computing** → **Graph algorithms**.

KEYWORDS

Network immunization; Network connectivity; complex network

ACM Reference Format:

Anonymous Author(s). 2024. Highly-efficient minimization of network connectivity in large-scale graphs. In *Proceedings of The Web Conference (Conference acronym 'XX).* ACM, New York, NY, USA, 10 pages. https://doi.org/XXXXXXX.XXXXXXX

Relevance Declaration: Our paper focuses on identifying the key nodes regarding network connectivity in complex graphs. This is crucial to protect online social networks from rumors and immunize Internet from computer viruses. Consequently, our research agrees well with the track 'Security and privacy' in the Web Conference 2024.

Conference acronym 'XX, 28 April – 2 May, 2025, Sydney, Australia

ACM ISBN 978-1-4503-XXXX-X/24/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

The optimization of network connectivity assumes great significance due to its capacity to delineate the structural properties of a complex network across a wide range of security applications [1, 2]. For instance, when examining the dynamics of SIR/SIS epidemic spreading, the occurrence of a SIR/SIS epidemic outbreak is typically contingent upon the connectivity of the principal eigenvalue of the adjacency matrix [3]. To mitigate such epidemics, it becomes imperative to minimize connectivity by means of patient isolation and the reduction of physical interactions between individuals. In the realm of Internet networks, natural connectivity acts as an evaluative metric for discerning network robustness [1]. Bolstering robustness requires the identification and preservation of critical nodes/edges within these networks. Hence, the optimization of network connectivity assumes pivotal importance in the manipulation of a network's dynamical function.

The optimization of network connectivity is commonly achieved by identifying and manipulating a collection of important nodes and edges [4]. Given a budget number k, the total number of potential node or edge sets is $\binom{n}{k}$ ($\binom{m}{k}$ for edge sets), where n and m correspond to the numbers of nodes and edges, respectively. As a result, the brute-force algorithm demonstrates exponential complexity and is impracticable for large-scale networks. To address the exponential time complexity, the most advanced methodologies can be classified into three distinct categories: (i) Heuristic algorithms[5]: These kinds of methods identify influential nodes based on various topological indices such as degree, betweenness, and PageRank. Although they usually have low time complexity, they cannot provide a guarantee regarding the quality of the solutions. (ii) Greedy algorithms [1, 6]: The algorithms select a candidate node at each step based on the marginal gain in network connectivity that occurs after removing each node. Research has shown that numerous connectivity functions adhere to the property of diminishing returns, which ensures an approximation ratio of $1-1/e$ for greedy algorithms [7, 8]. However, recalculating the marginal gain for every node at each step is computationally expensive in large-scale networks. (iii) Optimization of simplified objective functions [9, 10]: The algorithms optimize some simplified objective functions to calculate the influential nodes. However, similar to heuristic algorithms, these methods also lack a guarantee of solution quality.

In this study, we present a novel algorithm aimed at optimizing network connectivity in an efficient manner. In particular, the network connectivity can be efficiently represented by the combination of eigenvalues of a network's adjacency matrix [4]. Hence, we focus on how to accelerate the calculation of eigenvalues of the adjacency matrix after node perturbation. The primary contributions of this research can be summarized as follows: (a) *Surrogate matrix.* We introduce a straightforward *simple* surrogate matrix that closely mimics the eigenvalues of the *complex* perturbed adjacency matrix. Through rigorous analysis, we prove that the eigenvalues of the surrogate matrix and the perturbed adjacency matrix can be manipulated to predetermined small errors. (b) *A new algorithm.* By leveraging the surrogate matrix, we propose an expedient algorithm for optimizing node-level network connectivity with an approximation ratio of $1-1/e$. Our algorithm offers two significant advantages when compared to existing methods: (1) *effectiveness*, as it accounts for the perturbation of multiple eigenvalues, enabling it to effectively handle networks with small eigenvalue gaps; (2) *efficiency*, characterized by linear time complexity relative to the number of nodes. Notably, the time complexity of our algorithm remains unaffected by the number of edges, thereby ensuring scalability for large and dense networks. To the best of our knowledge, our algorithm is the most efficient solution for the connectivity minimization problem.

2 RELATED WORK

Our work touches on the network connectivity formalisms and optimization algorithms that are summarized as follows:

Network connectivity formalisms. Investigating network connectivity entails an examination of the degree to which a network is interconnected [1, 11, 12], as exemplified by various components such as graph diameter [13], clustering coefficient [14], and the presence of a giant component [15]. Conversely, from a dynamics standpoint, the focus primarily resides on investigating the dissemination models of SIS/SIR/SIRS [3], independent cascade failure [16], linear threshold model [17], and others [1]. Extensive research has demonstrated that a considerable number of dynamics are intricately tied to network connectivity, irrespective of the specific dynamics under investigation, and are solely dependent on the network's topology [18]. Chen et al. [7] has provided a comprehensive summary of connectivity metrics, elucidating how different metrics can be unified through combinations of eigenvalues derived from the adjacency matrix of a network.

Network connectivity minimization. The objective of network connectivity minimization is to identify a subset of nodes/edges whose removal can significantly reduce connectivity metrics. A commonly utilized metric for connectivity assessment is the reciprocal of the largest eigenvalue of the adjacency matrix or Laplacian matrix [19, 20]. Chen et al. [21] addressed the problem of selecting optimal nodes and edges to minimize the largest eigenvalue of the adjacency matrix. Prakash et al. [22] employed self-similar selection to propose an immunization approach for online networks. Zhang et al. [20] focused on minimizing eigenvalues of the nonbacktracking matrix through the manipulation of nodes and edges. Chen et al. [4] systematically formalized the connectivity minimization problem and introduced a fast QR-decomposition method to optimize connectivity.

A closely related problem is influence maximization, which aims to identify a subset of nodes that can maximize the activation of nodes in the final graph state [9, 17, 23–25]. Morone and Makse [9] established a connection between influence maximization and immunization problems. More recently, Fan et al. [10] introduced deep learning techniques into influence maximization. As our paper diverges from the focus of influence maximization, we limit the discussion of its details.

Our study bears some similarity to the connectivity optimization techniques discussed in refs. [4, 6]. In these references, the employed algorithm exhibited a time complexity of $O(k(mr+nr^3))$ that depends on the number of edges, rendering it computationally burdensome for dense networks. In contrast, our approach proposes to optimize the surrogate matrix. This stands in contrast to the adjacency matrix-based approach outlined in refs. [4, 6]. To the best of our knowledge, our algorithm is the fastest non-heuristic method to optimize network connectivity.

3 PROBLEM DEFINITION

In this study, we adopt a standardized notation system to enhance clarity and consistency in our mathematical framework. Specifically, the inequality $A < 0$ implies that the matrix A is negative definite, while the inequality $A < B$ implies that $A - B$ is negative definite. Let $G = (\mathcal{V}, \mathcal{E})$ be a finite undirected and unweighted graph of n nodes and m edges without self-loops, with node set $\mathcal{V} = \{1, 2, ..., n\}$ and edge set $\mathcal{E} = \{(i, j)|i, j \in \mathcal{V}\}$. The connections of nodes are represented by the adjacency matrix $A = (a_{ij})_{n\times n}$ with the entry a_{ij} denoting the adjacency relation between nodes i and j. If nodes i and j are linked to either by an edge $e \in \mathcal{E}$ with unit weight, then $a_{ij} = a_{ji} = 1$. Otherwise, $a_{ij} = a_{ji} = 0$. d_i is the degree of node i and d_{max} denotes the largest degree. We suppose that the adjacency matrix A has n distinct eigenvalues $\lambda_1 > \lambda_2 > ... > \lambda_n$ ($\Lambda^{(n)} = diag\{\lambda_1, \lambda_2, ..., \lambda_n\}$) with the corresponding eigenvectors $\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_n$ ($U^{(n)} = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_n]$), $A = U^{(n)}\Lambda^{(n)}{U^{(n)}}^T$. The largest eigenvalue of A is also denoted as λ_A ($\lambda_A = \lambda_1$). When we find a budget size k of influential nodes and remove the nodes from the graph, we obtain the adjacency matrix A' of the remaining graph by setting zero for the rows and columns of A that correspond to the removed nodes. The eigenvalues and eigenvectors of A' are denoted as eigen-pair $(\lambda'_i, \mathbf{v}'_i), i = 1, 2, ..., n$. If we specify a matrix notation, $\lambda_i(B)$ means the i-th largest eigenvalue of B. Without specification, $\lambda_i = \lambda_i(A)$, $\lambda'_i = \lambda_i(A')$.

Definition of network connectivity[4, 6]: The network connectivity is usually defined as

$$\zeta(G) = \sum_{\pi \in G} f(\pi), \tag{1}$$

where π is a subgraph of G, $f(\cdot)$ is a nonnegative mapping function $f : \pi \to \mathbb{R}^+$, $f(\emptyset) = 0$ for empty set; otherwise $f(\pi) > 0$. $\zeta(G)$ actually means the aggregation of all subgraphs in the network. In a large number of real scenarios, $\zeta(G)$ could be represented by a combination function of eigenvalues of A, $\zeta(G) = F(\Lambda^{(r)})$, where $F(\Lambda^{(r)})$ is a function of the top-r eigenvalues of A. Choosing appropriate function $f(\cdot)$ plays a crucial role in accurately emulating diverse connectivity metrics such as path capacity, triangle capacity, natural capacity, and others (See ref. [4]). In this paper, we minimize the connectivity based on the optimization of top-r eigenvalues. The problem could be formalized as:

Problem 1(Connectivity minimization): Finding k optimal nodes that, when removed, could minimize $\zeta(G \backslash \mathcal{S})$.

Input: The adjacency matrix A of a network, a budget integer k, and a connectivity measure $\zeta(G)$;

Output: A subset $\mathcal{S}$ of k nodes that minimize $\zeta(G \backslash \mathcal{S})$.

Previous research [6] established the NP-hardness of the connectivity minimization problem. Consequently, the widely employed approach is the greedy algorithm. At each step, the greedy algorithm calculates the marginal gain of $\zeta(G)$ for each node and chooses a candidate node that could minimize $\zeta(G)$. The current best implementation of the greedy algorithm [4, 6] demonstrates a time complexity of $O(k(mr + nr^3))$ that depends on the number of edges and proves to be impractical for dense networks. Consequently, to accelerate the greedy algorithm, the central issue of the greedy algorithm is to simplify the evaluation complexity of marginal gains, which is the main concern of our paper.

4 THE PROPOSED ALGORITHM

In order to efficiently calculate the marginal gains of $\zeta(G)$ for each node in the greedy algorithm, we first propose a surrogate matrix to approximate $\lambda_{A'}$ and the surrogate matrix has simplifier formalism than A. We then introduce a fast algorithm to minimize the connectivity metric based on the surrogate matrix.

4.1 The proposed surrogate matrix

Given a candidate node set $\mathcal{S}$, we first define the surrogate matrix as follows:

Definition 1 (Surrogate matrix): The surrogate matrix is defined as $A - D$, where D represents a diagonal matrix $D = diag\{c, c, 0, c..., 0\}_{n \times n}$, $D(i, i) = c$ if node $i \in \mathcal{S}$; $D(i, i) = 0$ otherwise. c is a real positive number.

We highlight the relative ease in calculating the surrogate matrix in comparison to the more complex process of obtaining the residual adjacency matrix. Specifically, the residual adjacency matrix is derived by assigning zeros to the rows and columns corresponding to the nodes within the set $\mathcal{S}$.

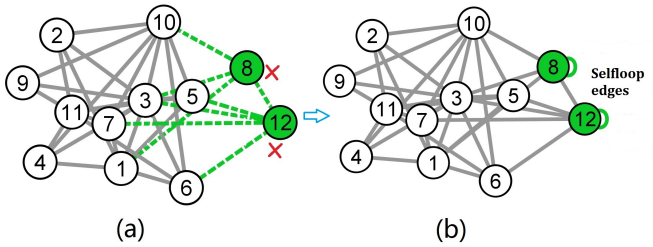

Figure 1: Schematic translation of the node removal problem. Removing a set of nodes is equivalent to the addition of negative self-loop edges.

On the other hand, the surrogate matrix is obtained through the addition of a single self-loop edge for each node within $\mathcal{S}$ (refer to Fig. 1).

We then provide an example to help understand the utilization of the surrogate matrix. Let us consider the spreading dynamics of an epidemic in an example network in Fig. 1, and let us consider the SIS spreading model. Let ρ_i be the infection probability of node i, $\rho_i = 0$ means susceptible state, $\rho_i = 1$ means complete infection. Supposing that two nodes labeled 8 and 12 (green) need to be immunized, we explore two strategies: (1) Removing the connections associated with nodes 8 and 12 (Fig. 1(a)). (2) Considering the application of negative feedback to these nodes using pinning control techniques [26] (Fig. 1(b)), where the feedback is formally defined as $c(0 - \rho_i)$ (see ref. [26] for the pinning control details). Both approaches successfully achieve immunization of the network (More details are shown in the *Appendix A*). In this study, we employ the concept of thresholds to examine the spreading dynamics of networks [3, 27]. For the first case, where two nodes are removed, the threshold is denoted as $\lambda_{A'}$ [27]. In the second case, the threshold is characterized by the largest eigenvalue λ_{A-D} of the matrix $A - D$, with D being the feedback diagonal matrix [26, 28]. The diagonal elements of matrix D indicate the feedback applied to the nodes. Since both strategies prevent the two specified nodes 8 and 12 involved in the spreading dynamics, the two approaches should share similar thresholds, i.e., $\lambda_{A'} \approx \lambda_{A-D}$. However, this conjecture currently lacks rigorous analysis and we will provide strict proofs in the following part.

Theorem 1: The eigenvalues of the surrogate matrix approximate the eigenvalues of A' at $c > \lambda_A$,

$$\lambda_i(A') = \lambda_i(A - D) + O(c^{-1}), \quad i = 1, 2, ..., n - |\mathcal{S}|. \tag{2}$$

By appropriately tuning the parameter c, we can ensure that the eigenvalues of $A - D$ closely approximate those of A' with an arbitrarily small error. Additionally, as $c \to +\infty$, the top$-r$ eigenvalues of $A - D$ coincide with those of A'. Before proving Theorem 1, we first prove that the largest eigenvalue of $A - D$ and A' satisfies Eq. 2.

Lemma 1: The largest eigenvalue $\lambda_{A'}$ of the residual network has the upper and lower boundary $\lambda_{A-D} - \frac{d^2_{max}}{c - \lambda_A} \leq \lambda_{A'} \leq \lambda_{A-D}$ at $c > \lambda_A$. If $c \to +\infty$, $\lambda_{A'} = \lambda_{A-D}$.

Proof: Without loss of generality, we suppose that the node set $\mathcal{S} = \{1 : k\}$ and consider two cases.

Case 1: We rewrite matrix A using block matrix formalism and construct the following quadratic form,

$$\begin{aligned} f_1(\begin{bmatrix} \mathbf{0} \\ \mathbf{x} \end{bmatrix}) &= \begin{bmatrix} \mathbf{0} \\ \mathbf{x} \end{bmatrix}^T (A - D) \begin{bmatrix} \mathbf{0} \\ \mathbf{x} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{0} \\ \mathbf{x} \end{bmatrix}^T \begin{pmatrix} A_{11} - cI & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{x} \end{bmatrix} \\ &= \mathbf{x}^T A_{22} \mathbf{x}, \end{aligned} \tag{3}$$

where A_{11} is the adjacency matrix within the nodes in $\mathcal{S}$, A_{22} actually represents the adjacency matrix of the residual network, $A_{22} = A'$, and $A_{12} = A_{21}^T$ is the adjacency matrix between set $\mathcal{S}$ and the residual nodes.

Recalling that for arbitrary matrix Q and arbitrary vector $\mathbf{x}$, the Rayleigh entropy $R(Q, \mathbf{x}) = \frac{\mathbf{x}^T Q \mathbf{x}}{\mathbf{x}^T \mathbf{x}}$ has the property $\lambda_{Q,min} \leq R(Q, \mathbf{x}) \leq \lambda_{Q,max}$, where $\lambda_{Q,min}$ and $\lambda_{Q,max}$ are the minimum and maximum eigenvalues of Q respectively. If we set $\mathbf{x}$ as the eigenvector corresponding to the largest eigenvalue of A_{22}, we have

$$f_1(\begin{bmatrix} \mathbf{0} \\ \mathbf{x} \end{bmatrix}) = \lambda_{A_{22}} \mathbf{x}^T \mathbf{x} \leq \lambda_{A-D} \begin{bmatrix} \mathbf{0} \\ \mathbf{x} \end{bmatrix}^T \begin{bmatrix} \mathbf{0} \\ \mathbf{x} \end{bmatrix}. \tag{4}$$

Hence, we arrive at $\lambda_{A_{22}} \leq \lambda_{A-D}$.

Case 2: Supposing $c > \lambda_A$, we construct the following quadratic form,

$$\begin{aligned} f_2(\begin{bmatrix} \mathbf{y} \\ \mathbf{x} \end{bmatrix}) &= \begin{bmatrix} \mathbf{y} \\ \mathbf{x} \end{bmatrix}^T (A - D) \begin{bmatrix} \mathbf{y} \\ \mathbf{x} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{y} \\ \mathbf{x} \end{bmatrix}^T \begin{pmatrix} A_{11} - cI & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{x} \end{bmatrix} \\ &= \mathbf{y}^T A_{11} \mathbf{y} - c\mathbf{y}^T \mathbf{y} + 2\mathbf{y}^T A_{12} \mathbf{x} + \mathbf{x}^T A_{22} \mathbf{x}. \end{aligned} \tag{5}$$

We focus on the terms,

$$\begin{aligned} & - c\mathbf{y}^T \mathbf{y} + 2\mathbf{y}^T A_{12} \mathbf{x} \\ &= -c \sum_i y_i^2 + \sum_{1 \leq i \leq k} \sum_{1 \leq j \leq n-k} 2A_{12}(i,j) y_i x_j \\ &= - \sum_{1 \leq i \leq k} \sum_{1 \leq j \leq n-k} A_{12}(i,j)^2 (\sqrt{\frac{\alpha c}{d_{max}}} y_i - \sqrt{\frac{d_{max}}{\alpha c}} x_j)^2 \\ &- c \sum_{1 \leq i \leq k} (1 - \alpha \sum_{1 \leq j \leq n-k} \frac{A_{12}(i,j)}{d_{max}}) y_i^2 \\ &+ \sum_{1 \leq i \leq k} \sum_{1 \leq j \leq n-k} (A_{12}(i,j) \sqrt{\frac{d_{max}}{\alpha c}} x_j)^2 \\ &\leq -c \sum_{1 \leq j \leq n-k} (1 - \alpha) y_j^2 + \sum_{1 \leq j \leq n-k} \frac{d_{max}^2}{\alpha c} x_j^2 \\ &= -c(1 - \alpha) \mathbf{y}^T \mathbf{y} + \frac{d_{max}^2}{\alpha c} \mathbf{x}^T \mathbf{x}, \end{aligned} \tag{6}$$

where α is a positive number. Hence, we have

$$\begin{aligned} f_2(\begin{bmatrix} \mathbf{y} \\ \mathbf{x} \end{bmatrix}) &\leq \mathbf{y}^T (A_{11} - c(1 - \alpha) I) \mathbf{y} + \mathbf{x}^T (A_{22} + \frac{d_{max}^2}{\alpha c} I) \mathbf{x} \\ &\leq \begin{bmatrix} \mathbf{y} \\ \mathbf{x} \end{bmatrix}^T B \begin{bmatrix} \mathbf{y} \\ \mathbf{x} \end{bmatrix}, \end{aligned} \tag{7}$$

where $B = \begin{pmatrix} A_{11} - c(1 - \alpha) I & 0 \\ 0 & A_{22} + \frac{d_{max}^2}{\alpha c} I \end{pmatrix}$ is a block diagonal matrix. Since $\lambda_{A_{11}} < \lambda_A$, when we set $c(1 - \alpha) = \lambda_A$, we have $A_{11} - c(1 - \alpha) I < 0$. If we set $\begin{bmatrix} \mathbf{y} \\ \mathbf{x} \end{bmatrix}$ as the eigenvector corresponding to the largest eigenvalue of $A - D$, we have

$$\lambda_{A-D} \begin{bmatrix} \mathbf{y} \\ \mathbf{x} \end{bmatrix}^T \begin{bmatrix} \mathbf{y} \\ \mathbf{x} \end{bmatrix} \leq \begin{bmatrix} \mathbf{y} \\ \mathbf{x} \end{bmatrix}^T B \begin{bmatrix} \mathbf{y} \\ \mathbf{x} \end{bmatrix}. \tag{8}$$

Based on the property of *Rayleigh entropy*, we can get

$$\begin{aligned} \lambda_{A-D} \leq \lambda_B &= max\{\lambda_{A_{11} - \lambda_A I}, \lambda_{A_{22} + \frac{d_{max}^2}{\alpha c} I}\} \\ &= \lambda_{A_{22}} + \frac{d_{max}^2}{c - \lambda_A}. \end{aligned} \tag{9}$$

Recalling that $\lambda_{A_{22}} \leq \lambda_{A-D}$ and Eq. 9, we have

$$\lambda_{A-D} - \frac{d_{max}^2}{c - \lambda_A} \leq \lambda_{A'} = \lambda_{A_{22}} \leq \lambda_{A-D}. \tag{10}$$

In Eq. 10, when $c \to +\infty$, $\lambda_{A'} = \lambda_{A-D}$. Hence $\lambda_{A'}$ satisfies the lemma. □

Next, we show that the other eigenvalues also satisfy Theorem 1.

Lemma 2: The eigenvalues of A' satisfies $\lambda_i(A') = \lambda_i(A - D) + O(c^{-1}), \quad i = 2, ..., n - |\mathcal{S}|$.

Proof: We first focus on the case $i = 2$. Let $\mathbf{x}_i$ and $\mathbf{y}_i$ ($|\mathbf{x}_i| = 1, |\mathbf{y}_i| = 1$) be the eigenvectors corresponding to the i-th largest eigenvalues of A' and $A - D$ respectively. Based on Lemma 1, we have $\lambda_{A - \lambda_1 \mathbf{x}_1 \mathbf{x}_1^T - D} - \frac{d_{max}^2}{c - \lambda_A} \leq \lambda_{A' - \lambda_1 \mathbf{x}_1 \mathbf{x}_1^T} \leq \lambda_{A - \lambda_1 \mathbf{x}_1 \mathbf{x}_1^T - D}$. We concern the matrix$[(A - D - \lambda_{A-D} \mathbf{y}_1 \mathbf{y}_1^T) + (\lambda_{A-D} \mathbf{y}_1 \mathbf{y}_1^T - \lambda_1 \mathbf{x}_1 \mathbf{x}_1^T)]$. Note that for arbitrary vector $\mathbf{v}$ ($|\mathbf{v}_i| = 1$), $|\mathbf{v}^T (\lambda_{A-D} \mathbf{y}_1 \mathbf{y}_1^T - \lambda_1 \mathbf{x}_1 \mathbf{x}_1^T) \mathbf{v}| < 2|\lambda_1 - \lambda_{A-D}|$. Since the largest eigenvalue of matrix $(A - D - \lambda_{A-D} \mathbf{y}_1 \mathbf{y}_1^T)$ is the second largest eigenvalue of $A - D$, we have $\lambda_2(A - D) = \lambda_1(A - D - \lambda_{A-D} \mathbf{y}_1 \mathbf{y}_1^T) + O(|\lambda_1 - \lambda_{A-D}|)$ and hence $\lambda_2(A') = \lambda_2(A - D) + O(c^{-1})$.

For the case $i > 2$, we consider the matrix $[A - D - \sum_{t=1}^{i} \lambda_{t,A-D} \mathbf{y}_t \mathbf{y}_t^T + (\sum_{t=1}^{i} \lambda_{t,A-D} \mathbf{y}_t \mathbf{y}_t^T - \sum_{t=1}^{i} \lambda_t \mathbf{x}_t \mathbf{x}_t^T)]$. Iteratively using the derivation of case 2 could arrive at lemma 2. □

4.2 Fast spectral calculation of the surrogate matrix

The core issue of the greedy algorithm is to calculate the eigenvalues of the residual adjacency matrix. Recognizing the remarkable approximation between the eigenvalues of $A - D$ and those of the residual matrix A', we introduce a

time-efficient approach to calculate the eigenvalues of $A-D$ as a viable alternative to computing those of A'.

In the process of identifying a suitable candidate node j, we construct a vector $\mathbf{v}_{n\times 1}$ that has only one nonzero entry $\mathbf{v}_j = 1$. We then execute partial-QR decomposition $[U^{(r)}, \mathbf{v}] = QR$. Importantly, it should be noted that the column vectors incorporated within $U^{(r)}$ are orthonormal, we only need to orthogonalize $\mathbf{v}$ from $U^{(r)}$ as

$$\mathbf{u} = \mathbf{v} - \sum_{i=1}^{r} (\mathbf{v}^T U^{(r)}(:,i)) U^{(r)}(:,i) = \mathbf{v} - \sum_{i=1}^{r} U^{(r)}(j,i) U^{(r)}(:,i). \tag{11}$$

We can also calculate the magnitude of $\mathbf{u}$ as $||\mathbf{u}|| = \sqrt{1 - \sum_i U^{(r)}(j,i)^2}$.

Then the partial-QR decomposition on $[U^{(r)}, \mathbf{v}]$ is

$$[U^{(r)}, \mathbf{v}] = [U^{(r)}, \mathbf{u}] \begin{bmatrix} I & \mathbf{r} \\ 0 & ||\mathbf{u}|| \end{bmatrix}, \tag{12}$$

where I is the identity matrix and $\mathbf{r} = U^{(r)}(j,:)^T$. For the partial-QR decomposition $[U^{(r)}, \mathbf{v}] = QR$, $Q = [U^{(r)}, \mathbf{u}]$ and $R = \begin{bmatrix} I & \mathbf{r} \\ 0 & ||\mathbf{u}|| \end{bmatrix}$.

Theorem 2: Matrix A could be effectively approximated by considering its top-r eigenvalues and their corresponding eigenvectors, along with an accompanying error denoted as E, i.e., $A = U^{(r)} \Lambda^{(r)} {U^{(r)}}^T + E$. When adding a new node j to set $\mathcal{S}$, the eigenvalues of modified matrix $A-D$ are equal to that of $Z = R \begin{bmatrix} \Lambda^{(r)} & 0 \\ 0 & -c \end{bmatrix} R^T$.

Proof: Since $A = U^{(r)} \Lambda^{(r)} {U^{(r)}}^T + E$, we have

$$A - D = [U^{(r)}, \mathbf{v}] \begin{bmatrix} \Lambda^{(r)} & 0 \\ 0 & -c \end{bmatrix} [U^{(r)}, \mathbf{v}]^T + E. \tag{13}$$

The eigenvalues of $A-D$ are determined by the first part of the right-hand side of Eq. 13. Since $[U^{(r)}, \mathbf{v}]$ has the QR decomposition formalism $[U^{(r)}, \mathbf{v}] = QR$, we can arrive at

$$A - D = QR \begin{bmatrix} \Lambda^{(r)} & 0 \\ 0 & -c \end{bmatrix} R^T Q^T + E. \tag{14}$$

Let $Z = R \begin{bmatrix} \Lambda^{(r)} & 0 \\ 0 & -c \end{bmatrix} R^T$, Z is a symmetric matrix and could be decomposed as $Z = U_Z \Lambda_Z U_Z^T$. Hence, we can arrive at

$$A - D = Q U_Z \Lambda_Z U_Z^T Q^T + E. \tag{15}$$

Since Q and U_Z are orthonormal matrices, Λ_Z and QU_Z actually mean the top eigenvalues and corresponding eigenvectors of $A-D$.□

Remark: Theorem 2 presents a practical approach for evaluating all the eigenvalues of the matrix $A-D$. Specifically, explicitly calculating the λ_{A-D} possesses time complexity $O(n^3)$. In contrast, the computation of the eigenvalues of Z exhibits a time complexity of $O(r^3)$. Given the fact that $r \ll n$, it is evident that the calculation of Z is a notably more efficient procedure.

4.3 The proposed Connectivity Optimization Algorithm (COA)

In this section, we introduce the details of the connectivity optimization method. The proposed algorithm is shown in Algorithm 1. In Algorithm 1, we initialize the set $\mathcal{S}$ empty. In lines 3-13, at each step, we first calculate the marginal gains of $\zeta(\cdot)$ for each node (lines 4–8) and then choose the best node corresponding to the maximum marginal gain (line 9). We then update the eigenvalues and eigenvectors of $A-D$ (lines 10–12).

Algorithm 1: Connectivity optimization algorithm (COA)

Input: The adjacency matrix A, the budget size k, and a connectivity measure $\zeta(\cdot) = F(\cdot)$
Output: Node set $\mathcal{S}$

1: Initialize $S = \emptyset$ and c as the largest degree of nodes.
2: Calculate top-r eigenvalues $\Lambda^{(r)}$ and eigenvectors $U^{(r)}$ of A.
3: **while** count =1 to k **do**
4: **while** $i \notin \mathcal{S}$ **do**
5: Calculate matrix R and Z that correspond to i.
6: Compute the whole eigenvalues Λ_Z of Z.
7: $I(i) = F(\Lambda^{(r)}) - F(\Lambda_Z)$.
8: **end while**
9: Add $j = argmax_j I(j)$ to $\mathcal{S}$.
10: Calculate matrix R and Z that correspond to j.
11: Compute the eigen decomposition of $Z = U_Z \Lambda_Z U_Z^T$.
12: $\Lambda^{(r)} = {\Lambda_Z}^{(r)}$, $U^{(r)} = (QU_Z)^{(r)}$.
13: $A = A - D$.
14: **end while**
15: **return** set $\mathcal{S}$.

Lemma 3: The time complexity of COA is $O(knr^3 + mr)$, among which the greedy iteration costs $O(knr^3)$.

proof: For the initialization, lines 1–2 calculate the top-r eigenvalues and eigenvectors, which have time complexity $O(nr^2 + mr)$. For lines 4–8, at each round, calculating the connectivity drop of each node requires traversing all eigenvalues with complexity $O(nr^3)$, and choosing a candidate node requires $O(n)$ (lines 9). Lines 10–12 update the eigenvalues and eigenvectors after the removal of a node. Hence, the overall time complexity is $O(knr^3 + mr)$. □

Lemma 4: The space complexity of the COA method is $O(nr + m)$.

proof: In the algorithm, we need $O(m)$ for matrix A, $O(nr)$ for matrix U and Q, $O(r^2)$ for R and Z. Hence, the total space complexity is $O(nr + m)$.□

Lemma 5: The COA algorithm could arrive at $(1 - 1/e)$ approximation ratio at arbitrary small error.

Proof: Based on Theorem 1, $\lambda_{A'} = \lambda_{A-D} + O(c^{-1})$. For arbitrary small error ε, if we set $c = \lambda_A + O(1/\varepsilon)$, the eigenvalue difference between $A-D$ and A' is $O(\varepsilon)$. Based on the diminishing return property of the objective function $\zeta(\cdot)$

[4, 21], the greedy algorithm has $(1 - 1/e)$ approximation ratio at arbitrary small error ε. Hence, we have Lemma 5.□

Parameter determination: Our algorithm COA (in Algorithm 2) has two parameters r and c. For the determination of parameter r, previous investigations have consistently demonstrated that the number of prominent eigenvalues of real networks aligns with the number of the communities [29, 30]. Given this, we use the modularity-based method [31] to detect communities and determine r. In practical applications, it is often feasible to set r to a considerably smaller value than the actual number of communities, thereby optimizing time efficiency. For instance, experimental results have verified that even with $r = 100$, an error rate lower than 1% can be achieved. For the determination of parameter c, larger c yields a diminished level of approximation error in the surrogate matrix. In the experiments, we could set $c = 2d_{max}$ that is empirically enough to have high solution quality. Moreover, empirical evidence has indicated that smaller values of c contribute to achieving higher precision. Significantly, the experimental results affirm that the approximation error between the eigenvalues of $A - D$ and A' is consistently below 1%.

5 EXPERIMENT

Here, we are interested in the performance of real data. Our experiments run on a computer with 4 2.4GHz Intel(R) i7 CPUs, 128GB memory, and 64bit Ubuntu 20.04.

5.1 Experimental setup

Datasets. We perform experiments on six different social networks [1]: (1) BitcoinSocial: A user trust/distrust network from the *Bitcoin OTC* platform with 5875 nodes and 21489 edges. (2) Hamstersterfriends: A social network containing friendships between users of the website *Hamsterster* with 1788 nodes and 12476 edges. (3) Twitter: A network representing user-user following information on *Twitter* with 22322 nodes and 31823 edges. (4) Facebook: The friendship of Facebook users with 63392 nodes and 816831 edges. (5) Epinions: The social network of the online product rating site *Epinions* with 119130 nodes and 704267 edges. (6) Flixster: The social network of *Flixster*, an online movie rating site with 2523386 nodes and 7918801 edges. All these networks are treated as undirected and unweighted.

Comparing methods. We compare our algorithm with seven state-of-the-art methods. (1) Degree: Nodes are ranked by their degree. (2) K-shell [32]: The method is based on the K-shell decomposition of the network. (3) PageRank [33]: The PageRank scores of nodes are calculated by the recursive PageRank equation. (4) Eigenvector [34]: The importance of a node is characterized by the entry of the principal eigenvector. (5) Netshield [21]: It is a method aiming at minimizing the largest eigenvalue of the adjacency matrix. (6) Contain [4]: The method optimizes the connectivity of a network based on top-r eigenvalues. (7) Finder [10]: It is a method based on reinforcement learning. (8) Brute force (Exact) method. It costs more than 24 hours in networks with millions of nodes. Hence, we omit its connectivity performance and only discuss its time consumption.

Parameter settings. The PageRank method has a parameter to tune the probability of teleportation, which is tuned to the optimal performance. For other parameters, we set $c = 2d_{max}$ and r as the number of communities unless otherwise stated. In the experiments, we also investigate the influence of c and r on the results. See the code on https://anonymous.4open.science/ r/connectivityoptimization-0DE2 .

Evaluation Metrics. We use two connectivity metrics: the largest eigenvalue λ_1' of the residual networks and the size of triangles in the residual networks [4, 35]. If different methods have similar performance, we compare their time consumption and smaller time consumption is better.

5.2 Experiment results

In the experiments, the performance of the COA algorithm was assessed from three distinct viewpoints: solution quality, time consumption, and parameter sensitivity.

Solution quality. The proposed COA method is initially compared with baseline methods using two connectivity metrics, namely the largest eigenvalue λ_1' and the number of triangles. It is worth noting that smaller values of these connectivity metrics indicate superior performance.

Figure 2 visually demonstrates that our COA method consistently achieves the smallest value for λ_1' across the tested networks. Consequently, the proposed COA outperforms the existing methods. The performance of classical heuristic methods, such as *Degree*, *K-shell*, *PageRank*, *Eigenvector*, and *Finder*, exhibits fluctuations across different networks, failing to guarantee solution quality. It is worth noting that our method attains nearly equivalent performance to *Contain*. However, *Contain* explicitly computes the top-r eigenvalues of the adjacency matrices of the residual networks, requiring the manipulation of all edges connected to the nodes in the influential set $\mathcal{S}$, resulting in increased time consumption. In contrast, our COA adopts a simple surrogate matrix $A - D$ to mitigate the time complexity.

Figure 3 shows the triangle numbers of different methods. We observe that our method has the largest drop of the triangle number, revealing the effectiveness of our method. The number of triangles agrees well with that of the largest eigenvalue λ_1' in Fig. 3. The results also demonstrate that our method outperforms the existing methods.

Time complexity result. We conducted a time consumption comparison between the COA method, *Contain*, and the brute force methods (refer to Fig. 4). This comparison is justified as only these three methods are reliant on the calculation of top-r eigenvalues.

Figure 4 illustrates the time consumption analysis of our COA algorithm compared to *Contain* and the brute force methods across various networks. The results confirm the efficiency of COA, as it consistently exhibits the lowest time consumption. Notably, the time consumption of COA increases

[1] Konect Data Collection, Http://konect.cc/networks/

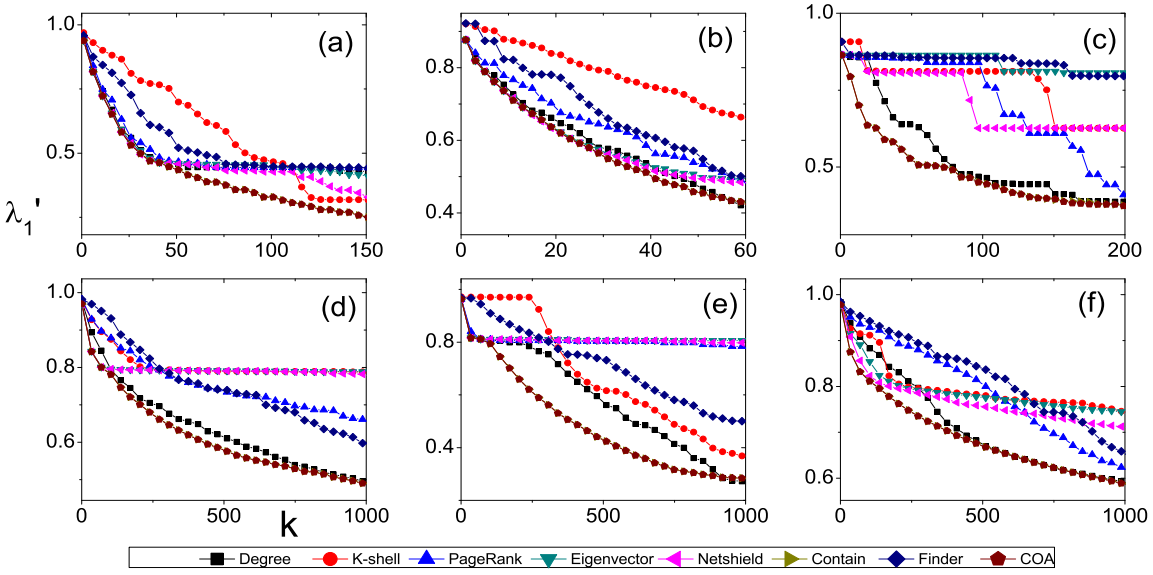

Figure 2: The comparison of largest eigenvalues λ_1' of different methods. Smaller λ_1' is better. (a) BitcoinSocial. (b) Hamstersterfriends. (c) Twitter. (d) FacebookWOSN. (e) Epinions. (f) Flixster.

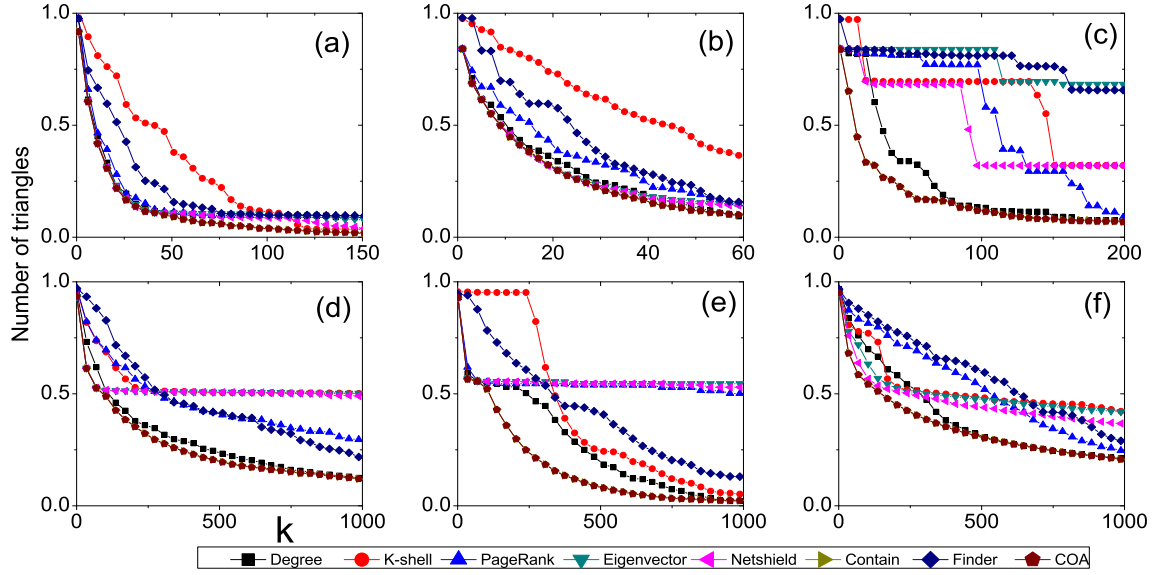

Figure 3: The comparison of triangle numbers of different methods. Smaller triangle number is better. (a) BitcoinSocial. (b) Hamstersterfriends. (c) Twitter. (d) FacebookWOSN. (e) Epinions. (f) Flixster.

gradually with the parameter k, owing to the algorithm's iteration (lines 4-8 in Algorithm 2), which has a linear relationship with the number of nodes and is independent of the number of edges. In contrast, the time complexities of *Contain* and the brute force methods rely on the number of edges, resulting in a higher time consumption overall compared to COA. Consequently, COA outperforms the other methods in terms of time consumption. It is worth mentioning that in certain networks (Figures 4(a)-(b)), *Contain* exhibits higher time consumption than the brute force method. This discrepancy can be attributed to the intricate connections within these networks and the varying time consumption involved in the spectral decomposition of parameter Z (line 6 in Algorithm 2) within these smaller networks. However, for larger networks (Figures 4(c)-(f)), *Contain* consistently displays

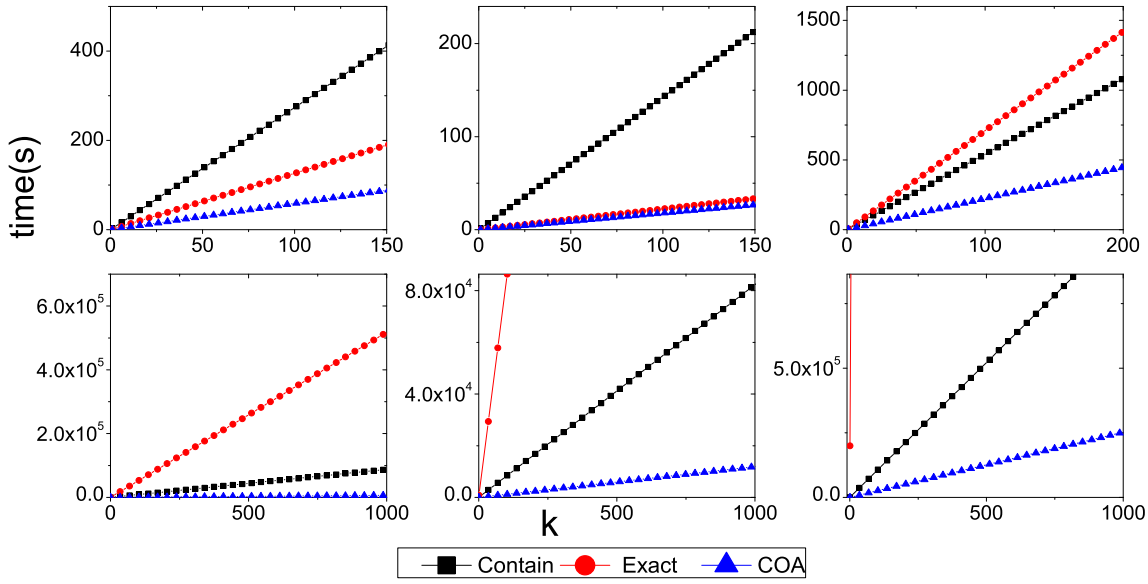

Figure 4: The time consumption of different methods. Smaller time is better. (a) BitcoinSocial. (b) Hamsterfriends. (c) Twitter. (d) FacebookWOSN. (e) Epinions. (f) Flixster.

lower time consumption than the brute force method, aligning with previous findings in the literature.

Parameter sensitivity. There are two parameters, r and c, associated with COA approach. First, we validate the impact of r on the connectivity performance. We employ the COA algorithm to identify influential nodes under various values of r and subsequently evaluate λ_1' of the residual network A', as demonstrated in Fig. 5(a). The findings exhibited in Fig. 5(a) demonstrate that with an increase in the value of r, the achieved λ_1' on the BitcoinSocial network decreases, indicating improved performance. Through extensive experimentation, we discover that a value of $r = 50$ yields highly satisfactory results, with errors less than 2%. Moreover, similar trends are observed in the other networks and we omit them due to space constraints.

To analyze the impact of parameter c on the approximation error between the largest eigenvalue of $A-D$ and A', we present the results in Figure 5(b). The figure demonstrates that as c increases, λ_{A-D} rapidly converges to λ_1' in the BitcoinSocial network. Notably, when $c = 10\lambda_A$, the disparity between the largest eigenvalues of $A-D$ and A' is less than 2%, providing empirical evidence for the effectiveness of Theorem 1. The results are similar for the other four networks, we omit four networks due to space limitation.

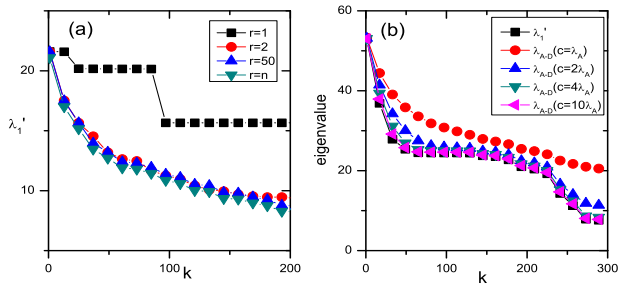

Figure 5: Parameter sensitivity results. (a) Influence of r on λ_1' on BitcoinSocial network. (b) Influence of r on λ_1' on BitcoinSocial network.

6 CONCLUSION

In this paper, we study the problem of how to efficiently minimize network connectivity. Initially, we propose a *simple* surrogate matrix as an approximation for the *complex* residual adjacency matrix. Through rigorous proof, we establish that the top-r eigenvalues of the surrogate matrix closely resemble those of the residual networks, with negligible predefined errors. To optimize network connectivity, we subsequently devise a rapid algorithm that calculates the top-r eigenvalues of the residual networks and selects the optimal node-set. Notably, our algorithm boasts a small time complexity of $O(knr^3)$, which remains invariant to the number of edges. Consequently, it demonstrates scalability for large and dense networks. Experimental results validate the superior performance of our algorithm compared to existing methods, thereby validating our theoretical analysis.

The primary focus of our study concerns the resolution of the largest eigenvalue and the number of triangles. As part of future work, we aim to explore alternative approaches for connectivity optimization.

REFERENCES

[1] Scott Freitas, Diyi Yang, Srijan Kumar, Hanghang Tong, and Duen Horng Chau. Graph vulnerability and robustness: A survey. *IEEE Trans. Knowl. Data Eng.*, 35(6):5915–5934, 2023.

[2] Cesar H Comin, Thomas Peron, Filipi N Silva, Diego R Amancio, Francisco A Rodrigues, and Luciano da F Costa. Complex systems: features, similarity and connectivity. *Physics Reports*, 861:1–41, 2020.

[3] Ming Li, Run-Ran Liu, Linyuan Lü, Mao-Bin Hu, Shuqi Xu, and Yi-Cheng Zhang. Percolation on complex networks: Theory and application. *Physics Reports*, 907:1–68, 2021.

[4] Chen Chen, Ruiyue Peng, Ying Lei, and Hanghang Tong. Network connectivity optimization: Fundamental limits and effective algorithms. In *the 24th ACM SIGKDD International Conference*, pages 19–23, 2018.

[5] Linyuan Lü, Duanbing Chen, Xiao-Long Ren, Qian-Ming Zhang, Yi-Cheng Zhang, and Tao Zhou. Vital nodes identification in complex networks. *Physics Reports*, 650:1–63, 2016.

[6] C. Chen, R. Peng, L. Ying, and H. Tong. Fast connectivity minimization on large-scale networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 15(3):1–25, 2021.

[7] Chen Chen, Jingrui He, Nadya Bliss, and Hanghang Tong. On the connectivity of multi-layered networks: Models, measures and optimal control. In *2015 IEEE International Conference on Data Mining*, pages 715–720. IEEE, 2015.

[8] Wenruo Bai and Jeff Bilmes. Greed is still good: maximizing monotone submodular+ supermodular (bp) functions. In *International Conference on Machine Learning*, pages 304–313. PMLR, 2018.

[9] Flaviano Morone and Hernán A Makse. Influence maximization in complex networks through optimal percolation. *Nature*, 524(7563):65–68, 2015.

[10] Changjun Fan, Li Zeng, Yizhou Sun, and Yang-Yu Liu. Finding key players in complex networks through deep reinforcement learning. *Nature machine intelligence*, 2(6):317–324, 2020.

[11] Hau Chan, Shuchu Han, and Leman Akoglu. Where graph topology matters: the robust subgraph problem. In *Proceedings of the 2015 SIAM International Conference on Data Mining*, pages 10–18. SIAM, 2015.

[12] Thang N Dinh, Ying Xuan, My T Thai, Panos M Pardalos, and Taieb Znati. On new approaches of assessing network vulnerability: hardness and approximation. *IEEE/ACM Transactions on Networking*, 20(2):609–619, 2011.

[13] Mukkai S Krishnamoorthy and Balaji Krishnamurthy. Fault diameter of interconnection networks. *Computers & Mathematics with Applications*, 13(5-6):577–582, 1987.

[14] Oded Green and David A Bader. Faster clustering coefficient using vertex covers. In *2013 International Conference on Social Computing*, pages 321–330. IEEE, 2013.

[15] Yingying Duan and Feng Lu. Robustness of city road networks at different granularities. *Physica A: Statistical Mechanics and its Applications*, 411:21–34, 2014.

[16] Paolo Crucitti, Vito Latora, and Massimo Marchiori. Model for cascading failures in complex networks. *Physical Review E*, 69(4):045104, 2004.

[17] David Kempe, Jon Kleinberg, and Éva 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*, pages 137–146, 2003.

[18] Yang Wang, Deepayan Chakrabarti, Chenxi Wang, and Christos Faloutsos. Epidemic spreading in real networks: An eigenvalue viewpoint. In *22nd International Symposium on Reliable Distributed Systems, 2003. Proceedings.*, pages 25–34. IEEE, 2003.

[19] B Aditya Prakash, Deepayan Chakrabarti, Nicholas C Valler, Michalis Faloutsos, and Christos Faloutsos. Threshold conditions for arbitrary cascade models on arbitrary networks. *Knowledge and information systems*, 33(3):549–575, 2012.

[20] Zuobai Zhang, Zhongzhi Zhang, and Guanrong Chen. Minimizing spectral radius of non-backtracking matrix by edge removal. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2657–2667, 2021.

[21] Chen Chen, Hanghang Tong, B Aditya Prakash, Charalampos E Tsourakakis, Tina Eliassi-Rad, Christos Faloutsos, and Duen Horng Chau. Node immunization on large graphs: Theory and algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 28(1):113–126, 2015.

[22] B Aditya Prakash, Lada Adamic, Theodore Iwashyna, Hanghang Tong, and Christos Faloutsos. Fractional immunization in hospital-transfer graphs. pages 1–9, 2011.

[23] Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 199–208, 2009.

[24] Yuchen Li, Ju Fan, Yanhao Wang, and Kian-Lee Tan. Influence maximization on social graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 30(10):1852–1872, 2018.

[25] Wei Chen, Chi Wang, and Yajun Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1029–1038, 2010.

[26] Xiaofan Wang, Xiang Li, and Jinhu Lu. Control and flocking of networked systems via pinning. *IEEE Circuits and Systems Magazine*, 10(3):83–91, 2010.

[27] Deepayan Chakrabarti, Yang Wang, Chenxi Wang, Jurij Leskovec, and Christos Faloutsos. Epidemic thresholds in real networks. *ACM Transactions on Information and System Security (TISSEC)*, 10(4):1, 2008.

[28] X.F. Wang and G. Chen. Pinning control of scale-free dynamical networks. *Physica A: Statistical Mechanics and its Applications*, 310(3):521–531, 2002.

[29] Xiaofan Wang, Xiang Li, and Guangrong Chen. *Network Science: An Introduction*. High education press, 2012.

[30] C. Sarkar and S. Jalan. Spectral properties of complex networks. *CHAOS*, 28(10):102101, 2018.

[31] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69(6):066133, 2004. PRE.

[32] Maksim Kitsak, Lazaros K Gallos, Shlomo Havlin, Fredrik Liljeros, Lev Muchnik, H Eugene Stanley, and Hernán A Makse. Identification of influential spreaders in complex networks. *Nature physics*, 6(11):888–893, 2010.

[33] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[34] Ali Moradi Amani, Mahdi Jalili, Xinghuo Yu, and Lewi Stone. A new metric to find the most vulnerable node in complex networks. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. IEEE, 2018.

[35] Charalampos E Tsourakakis. Fast counting of triangles in large real networks without counting: Algorithms and laws. In *2008 Eighth IEEE International Conference on Data Mining*, pages 608–617. IEEE, 2008.

[36] Panpan Shu, Wei Wang, Ming Tang, and Younghae Do. Numerical identification of epidemic thresholds for susceptible-infected-recovered model on finite-size networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 25(6):063104, 2015.

[37] Deepayan Chakrabarti, Yang Wang, Chenxi Wang, Jurij Leskovec, and Christos Faloutsos. Epidemic thresholds in real networks. *ACM Transactions on Information and System Security (TISSEC)*, 10(4):1, 2008.

[38] Romualdo Pastor-Satorras and Alessandro Vespignani. Epidemic spreading in scale-free networks. *Physical review letters*, 86(14):3200, 2001.

[39] Yang-Yu Liu and Albert-László Barabási. Control principles of complex systems. *Reviews of Modern Physics*, 88(3):035006, 2016.

A APPENDIX A: MOTIVATION DETAILS

In this part, we only brief the SIS model and pinning control of the SIS model, and provide some outline of the critical thresholds of the dynamics. For proof details, please refer to refs. [26, 29, 36, 37].

SIS model. Considering the susceptible-infected-susceptible (SIS) spreading model [38] in a network $G(N, E)$ denoted by an adjacency matrix $\mathsf{A} = (a_{ij})_{N\times N}$. Let $\rho_i(t)$ represent the

infection probability of node i at time t. The general dynamics of each node could be written as [36, 38]

$$\frac{\partial \rho_i(t)}{\partial t} = -\rho_i(t) + \beta[1-\rho_i(t)] \sum_{j \in N_i} \rho_j(t), \tag{16}$$

where N_i is the neighboring set of node i.

Here, we utilize a convenient approach to calculate the threshold of the network. When protecting a network, we usually desire controlling the state of all nodes onto the susceptible state, $\rho_i(t) \approx 0, i = 1, 2, ..., N$. When all the $\rho_i(t) \to 0$, the high-order terms $\rho_i(t) \cdot \rho_j(t)$ of the right hand side (r.h.s) of Eq. 16 is much smaller than the terms $\rho_i(t)$ and $\rho_j(t)$. Thus, we neglect the influence of the high-order terms $\rho_i(t) \cdot \rho_j(t)$ of the r.h.s of Eq. 16 and obtain

$$\frac{\partial \rho_i(t)}{\partial t} = -\rho_i(t) + \beta \sum_{j \in N_i} \rho_j(t). \tag{17}$$

Equation 17 could be written in matrix formalism,

$$\frac{\partial \rho(t)}{\partial t} = (\beta \mathsf{A} - I)\rho(t), \tag{18}$$

where $\rho(t) = [\rho_1(t), \rho_2(t), ..., \rho_N(t)]^T$. Equation 18 is a linear dynamical system. According to the controllability principle [39], $\rho(t)$ could be controlled onto $\rho(t) = \mathbf{0}$ when the largest eigenvalue of $(\beta \mathsf{A} - I)$ is negative that is

$$\beta_c = \frac{1}{\lambda_1}, \tag{19}$$

where λ_1 is the largest eigenvalue of A. The critical threshold of Eq. 19 could be characterized the reciprocal of its largest eigenvalue.

When we remove the nodes in $\mathcal{S}$, only the nodes in the residual network follows the above equation. Hence, the critical threshold of the residual network is characterized by λ_1'.

More strict derivation of Eq. 19 could also be found in ref. [37].

Pinning control. We design negative feedback for the chosen influential nodes. Without loss of generality, we suppose $\mathcal{S} = \{1 : k\}$ and modified SIS dynamics for the nodes in $\mathcal{S}$ is

$$\frac{\partial \rho_i(t)}{\partial t} = -\rho_i(t) + \beta[1-\rho_i(t)] \sum_{j \in N_i} \rho_j(t) + c[0-\rho_i(t)], \tag{20}$$

where the last term of the r.h.s of Eq. 20 represents feedback. The feedback is inclined to control the $\rho_i(t)$ to 0.

For the nodes in $\mathcal{S}$, when the dynamics arrive at stationary state, $\frac{\partial \rho_i(t)}{\partial t} = 0$, we have $\rho_i(t) = \frac{\beta \sum_{j \in N_i} \rho_j(t)}{c+1+\beta \sum_{j \in N_i} \rho_j(t)}$. Since $\rho_i(t) \in [0, 1]$ and $\sum_{j \in N_i} \rho_j(t) < d_{max}$, we have $\rho_i(t) < \frac{\beta d_{max}}{c+1+\beta d_{max}}$. When $c \to +\infty$, $\rho_i(t) \to 0$. When c is large, the nodes in $\mathcal{S}$ are rarely influenced by its neighbors, the following equation 22 has similar stationary state with Eq. 20. We note that the second terms of the r.h.s of Eq. 21 and Eq. 20 influence the dynamics little, because when $c \to +\infty$, $\rho_i(t) \to 0$ in Eq. 21.

$$\frac{\partial \rho_i(t)}{\partial t} = -\rho_i(t) + \beta \sum_{j \in N_i} \rho_j(t) + c[0-\rho_i(t)]. \tag{21}$$

The dynamics of the other nodes still follows Eq. 17. Hence, the dynamics of the whole nodes could be modeled as

$$\frac{\partial \rho(t)}{\partial t} = (\beta \mathsf{A} - D - I)\rho(t). \tag{22}$$

$\rho(t)$ could be controlled onto $\rho(t) = \mathbf{0}$ when the largest eigenvalue of $(\beta \mathsf{A} - D - I)$ is negative that is $\beta_c = \frac{1}{\lambda_{1,\beta A - D}}$. If we set $c' = D/\beta$ and $D' = D/\beta$, $\beta_c = \frac{1}{\lambda_{1,A-D'}}$. Hence, the critical threshold of the residual network is characterized by $\lambda_{1,A-D'}$.

Combining the analysis of SIS and pinning control models, we shows that the critical threshold of the residual network could be characterized by λ_1' and $\lambda_{1,A-D'}$ respectively. The two formalisms should be equivalent. Here, we only outline the SIS and pinning control, which provides intuition to conjecture that $A-D$ has similar eigenvalues of A'. See Theorem 1 for the strict proof.