Preserving Coreness while Reducing Connectivity in Network Graphs

Ahmad Zareie
School of Computer Science
The University of Sheffield
Sheffield, UK
a.zareie@sheffield.ac.uk

Rizos Sakellariou
Department of Computer Science
The University of Manchester
Manchester, UK
rizos@manchester.ac.uk

***Abstract*—Graphs, composed of nodes and edges, are widely used to model a variety of networks. On certain occasions, it is necessary to reduce connectivity within such network graphs to alleviate adverse effects, for instance, to contain undesirable or malicious events, such as the spread of viruses or rumours. However, such adjustments could alter a network's structure, potentially leading to the network's disruption and instability. In this paper, we address the problem of minimising network connectivity while ensuring that the network's coreness, a measure of its stability, remains intact. Given a graph and an integer b, the problem is to identify a set containing b edges whose removal minimises network connectivity while preserving its coreness. We outline a formal definition of the problem, establish its NP-hardness and, then, we introduce a naive greedy solution, which is improved by heuristics. A metric to assess the extent to which network connectivity can be reduced while maintaining network stability is also described. Experiments employing six real-world networks assess the effectiveness and efficiency of the proposed methods and demonstrate their suitability to solve the problem.**

***Index Terms*—coreness, network stability, network connectivity, network graphs, edge manipulation**

I. INTRODUCTION

Networks have applications in diverse fields, like computer systems, social relationships, water distribution, and power grids. Such networks are characterised by interconnected components and are typically represented by graph models. A graph contains nodes and edges, which denote components and interactions between pairs of these components, respectively.

The interconnected nature of a network facilitates the spread of undesirable or malicious phenomena across its various segments, potentially impacting the entire network adversely. When such events occur, a typical reaction is to modify the network by selectively blocking or removing interconnections (edges) within the network. This action aims to reduce the graph's overall connectivity thereby containing the spread and/or impact of such detrimental phenomena throughout the network. While numerous methods have been proposed to address network alterations to contain the dissemination of malicious phenomena in a network [1]–[5], they overlook the critical aspect of how these alterations might impact nodes' coreness and therefore network stability. In this paper, we aim to minimise the connectivity of a network while preserving its stability, something that differentiates our approach from existing methods.

Coreness serves as a widely adopted metric to assess network stability [6]–[8]. In this paper, network stability pertains to the network's ability to sustain its coreness, which implies consistent and dependable performance even after the network's connectivity is reduced. Specifically, a node's coreness is defined as the maximum value k (k-core) for which the node belongs to a subgraph where each node maintains at least k connections. In the network shown in Figure 1, nodes 1, 2, 3, and 4 exhibit a coreness value of 3, while nodes 5 and 6 have a coreness of 2. Nodes 7, 8, and 9 possess a coreness value of 1.

The paper deals with the challenge of reducing the connectivity of a graph without altering the k-core of nodes within the graph. The following examples can help understand and appreciate the motivation underpinning this work.

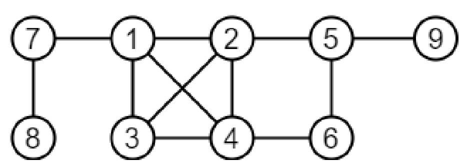

Fig. 1. An example network.

Example 1: Consider a computer network comprising servers, where the coreness of each server signifies the required collaborative connections for ensuring its service quality [9], [10]. If a server's coreness decreases, it becomes unresponsive to requests. In the event of a virus spreading through this network, removing connections becomes necessary to reduce network connectivity and slow the spread, preventing its dissemination across the entire network. However, any connection removals must

ISBN 978-3-903176-72-0 © 2025 IFIP

not compromise the stability of the network, that is, the coreness of network nodes. For instance, if the network shown in Figure 1 depicts the connectivity between servers in a computer network, removing the edge between servers 2 and 5 lowers the coreness of node 5, consequently causing a server failure. This failure then impacts nodes 6 and 9, reducing their coreness as well. Thus, removing the edge between servers 2 and 5 risks destabilising the network.

Example 2: Consider an online social network where users form connections representing friendships, and each user's coreness reflects their engagement within the network. A decrease in coreness implies reduced engagement, possibly leading to a user leaving the network [11]. When a rumour spreads through this network, blocking connections becomes essential to limit the rumour's reach and prevent its dissemination. However, blocking these connections results in reduced engagement among users, triggering a cascading effect of exits from the network. For instance, if Figure 1 depicts friendships between users in an online social network, blocking the edge between users 1 and 2 diminishes their engagement (coreness), leading to their exit and subsequently causing a cascade of exits among other connected users.

These examples, a small selection among many similar ones across diverse fields, underscore the importance of minimising network connectivity while preserving the network's stability. This paper is dedicated to addressing this critical issue. We begin by formally defining the problem and establishing its NP-hardness. We propose a greedy method to solve the problem, which we further enhance for practical applicability. Evaluating this method against some suggested baselines reveals its effectiveness in minimising network connectivity with preserved stability while also demonstrating its efficiency in achieving low running times. Additionally, we introduce a core-modularity metric that enables assessing how well a network is structured to reduce connectivity without compromising stability. This metric serves as a vital tool for evaluating a network's structural integrity, facilitating a subtle understanding of how connectivity can be strategically managed during periods of crisis.

The contributions of this paper can be summarised as follows: (i) Formulation of the problem of Preserving Stability with Reduced Connectivity (PSRC), which aims to identify a set of edges whose removal minimises network connectivity, measured by the Harary index, of a graph while preserving its coreness, a measure of stability. (ii) Proof of the problem's NP-hardness and demonstration that it cannot be approximated within a polynomial time ratio of $(1-1/e+\epsilon)$, unless P = NP. (iii) Introduction of a core-modularity metric that assesses a network's structural suitability for reducing connectivity without compromising stability. (iv) Development of a solution for the PSRC problem using a greedy method and heuristics to identify specific edges for removal from the graph. (v) Experimental evaluation of the proposed method using 6 network datasets, showcasing the superior performance of the proposed method in comparison to a set of baseline methods.

The rest of the paper is organised as follows. Section II presents an overview of related work. Section III formally defines the problem and establishes its NP-hardness. The specifics of the proposed greedy method, along with the implemented heuristics to enhance its performance, are detailed in Section IV. Section V introduces the metric of *core-modularity* and suggests a method to calculate the metric. Section VI delves into the experimental settings and presents the evaluation results. Finally, Section VII concludes the paper.

II. Related Work

A. Connectivity Minimisation

From a broad perspective, *network connectivity* indicates how effectively nodes are linked together. Various metrics like diameter [12], eigenvalues [13], clustering coefficient [14], total pairwise connectivity [15], and Harary index [16] have been employed to evaluate network connectivity. Connectivity is a pivotal factor in networks, offering benefits such as resilience, robustness and enhanced node interaction. For example, edge failures may not disrupt interactions in highly connected networks due to available alternate pathways. However, excessive connectivity could yield adverse effects, such as cascading failures and vulnerability to attacks, necessitating a reduction in graph connectivity to mitigate these impacts. The subsequent paragraphs review studies dedicated to edge removal to optimise network structures, an approach that often leads to decreased network connectivity as a consequence.

The removal of network edges has been considered as an approach to mitigate the spread of something undesired. For instance, in [17], breaking the network into smaller clusters is explored as an approach to minimise connectivity and contain rumour propagation. Additionally, the work in [18] identifies critical edges based on the structural properties of edges and their connected nodes to control the spread of rumours. In some other studies [19]–[21], minimising the spectral radius of a graph is addressed to minimise the connectivity and contain the spread of diseases in human interaction networks. Edge removal serves various purposes across diverse fields, impacting network connectivity, such as, maximising diameter [22], reducing total pairwise connectivity [15], enhancing resistance against malicious attacks [23] or minimising the number of triangles within the network [24].

The Harary index has been considered as a connectivity measure in some studies, where the goal is to remove edges to optimise the index in the network. The problem in [2] focuses on determining whether there exists a subset of edges whose removal ensures that the Harary index of the network remains below a specified threshold. This approach is relevant in scenarios where weakening the network's connectivity without completely fragmenting it is desired. The problem in [25] aims to identify a set of edges whose removal maximises the Harary index in the network. The authors propose an approach grounded in the Stackelberg game model to tackle the problem. In [26], the focus is on maximising the Harary index to decrease network connectivity, thereby minimising the spread of negative phenomena such as viruses and rumours. To address this problem, the authors evaluate the impact of removing edges based on their structural properties. They demonstrate that removing edges with high betweenness on the shortest paths between pairs of nodes is the most effective strategy for minimising the Harary index. In this paper, we also use the Harary index as the connectivity measure.

The research described in this paper distinguishes itself from prior studies through its main objective. While earlier work concentrates on altering network structures essentially without accounting for any impact on stability, our aim is unique. We seek to minimise network connectivity while safeguarding the coreness of all nodes, ensuring both the post-alteration functionality of all nodes and the preservation of network stability.

B. Network Stability

From a comprehensive viewpoint, *network stability* denotes the nodes' ability to uphold their functions or characteristics in the network despite any disturbances or alterations in the network structure. Network stability may be assessed based on node characteristics. Various centrality measures, such as degree, closeness [27], betweenness [28] and coreness [29], have been employed to assess node characteristics within a network. These measures assign specific values to nodes, expressing their characteristics based on diverse structural properties in the network. Degree centrality assesses a node's characteristics by considering the number of edges connected to it. Closeness centrality evaluates a node's centrality based on its proximity to all other nodes in the network. Betweenness centrality measures a node's characteristics by assessing its role as a mediator along the shortest paths between all pairs of nodes. Coreness assesses a node's characteristics based on its level of engagement within the network. The coreness of a node represents the highest k value in the largest subgraph containing this node, such that every node within that subgraph maintains a degree of at least k.

Coreness has found an application across various fields like biology, ecology, computer science, and social network analysis [30] and it is acknowledged as a metric for network stability in numerous studies [6]–[8], [11], [31]–[33]; we also use it in this paper. The next paragraph reviews studies on edge removal to optimise network structures, while specifically considering the coreness centrality measure.

Several studies [32]–[34] have examined edge removal as a means to minimise the number of nodes with a specified coreness value, referred to as the k-core minimisation problem. First introduced in [33], the authors proposed a greedy algorithm augmented with heuristics to tackle this problem. In contrast, the paper in [34] delves into the resilience of the k-core structure against edge removal, introducing a game-theoretic algorithm to address the k-core minimisation problem. The work in [32] extends the k-core minimisation problem across various metrics, aiming to minimise the collapsed k-core concerning node count, edge count, and overall coreness. The paper presents a baseline method, optimised with heuristics, to solve these extended problems.

The study in [35] highlights issues with the coreness accuracy to determine nodes' ability in spreading processes; it states that this inaccuracy is due to core-like groups with high coreness indices but low spreading ability. They introduced a measure for edge redundancy in spreading processes and showed that removing redundant edges improves the accuracy of node ranking and network core structure identification. In [36], the authors introduce the concept of skeletal core subgraph to explore the backbone of k-core structures within graphs. They propose an approach to categorise graphs based on their skeletal core subgraph and demonstrate the relationship between the skeletal core subgraph and various graph properties, including core resilience.

The problem addressed in this paper differs from the studies discussed above by focusing on minimising network connectivity while preserving the coreness of all nodes.

III. Problem Definition

A network can be modelled using an undirected, unweighted graph $G(V, E)$, where the node set V represents the components and the edge set $E \subseteq V \times V$ indicates the interactions between pairs of components. The number of nodes and edges in the graph is denoted as $|V|$ and $|E|$, respectively. An edge between nodes v_i and v_j denotes the interaction between these nodes; these nodes are considered as neighbours. The number of v_i's neighbours, denoted by γ_i, signifies the degree of v_i. The distance d_{ij} between two nodes v_i and v_j represents the minimum number of edges to traverse from v_i to v_j while $1/d_{ij}$ denotes the inverse distance between these nodes. Given the undirected nature of the graph, $d_{ij} = d_{ji}$. In cases where there is no path between v_i and v_j, then $d_{ij} = \infty$. When a set of edges $S \subset E$ is removed from the graph G, the resulting reduced graph is denoted as $G^{(S)} = G(V, E - S)$.

In this paper, we calculate the connectivity of graph G using the Harary index [16], also known as inverse geodesic length, which is based on the inverse distance between all pairs in the graph. The Harary index of graph G is calculated using Eq. (1) as follows:

$$C(G) = \sum_{(v_i,v_j)\in V} \frac{1}{d_{ij}} \quad (1)$$

Higher values of $C(G)$ indicate stronger connectivity [37]. The selection of the Harary index is influenced by two key factors [2], [25]. First, this index has been extensively studied in the literature as a global measure of network connectivity. Second, the index remains effective regardless of the input graph structure.

The coreness of node v_i is denoted as k_i. When referring to the coreness of a node within a graph or its modified version, we use the notation $k_i(G)$ to explicitly refer to the coreness of node v_i in graph G. The coreness of nodes in a graph can be determined using the iterative algorithm proposed in [29]. This algorithm starts with $k = 1$ and removes all nodes of degree 1 until no node with degree 1 is left. A value of coreness equal to 1 is assigned to the removed nodes at this stage. At the next stage, the value of k is increased by one and the process is repeated for this value of k. The algorithm continues until all nodes are removed. In this paper, network stability is assessed through coreness, indicating that a network remains stable if all nodes maintain their coreness despite a reduction in network connectivity. For a network modelled as a graph G, stability is achieved if the coreness of all nodes remains unchanged when removing a set of edges S; this means that $k_i(G) = k_i(G^{(S)})$ for every node $v_i \in V$.

Problem Statement: Given a network modelled as an undirected, unweighted graph $G(V, E)$ with an integer value b, the problem of *Preserving Stability with Reduced Connectivity (PSRC)* seeks to identify a set $S \subset E$ consisting of at most b edges. Removing these edges from the graph minimises its connectivity while maintaining the coreness of all nodes unchanged, thereby preserving the network stability. This problem is formally defined by Eq. (2).

$$\begin{aligned} S^* &= \underset{S\subset E}{arg\,min}\ C(G^{(S)}) \\ &\text{s.t.}(|S| \leq b) \text{ and } (k_i(G) = k_i(G^{(S)})\ \forall v_i \in V) \end{aligned} \quad (2)$$

The first constraint, $|S| \leq b$, ensures that the edges removed from the network do not exceed a specified budget b. Meanwhile, the second constraint, $k_i(G) = k_i(G^{(S)})\ \forall v_i \in V$, guarantees the preservation of coreness and hence network stability.

Problem Hardness: PSRC is NP-hard for any given b.

Proof: To establish the NP-hardness of the PSRC problem, we reduce this problem from a well-known NP-hard problem, the Knapsack problem [38]. In the Knapsack problem, the goal is to select items from a set, each with its weight and value, to maximise the total value while adhering to a weight limit. In a variant of the Knapsack problem known as the Position-Dependent Knapsack Problem (PDKP) [39], the weight and value of an item are functions that depend on the other selected items. Consider the set of edges in a network, denoted as E. Each edge $e \in E$ is treated analogously as an item in PDKP. For an edge e at position r in the sequence of items packed in the knapsack, two functions $\mathcal{F}_w(e)$ and $\mathcal{F}_v(e)$ are defined to determine the weight and value of the edge.

The function $\mathcal{F}_w(e)$ evaluates to 1 if the removal of the edge e from the network, along with all other edges positioned before the position r of edge e in the knapsack, does not compromise the stability of the network. However, if removing the edge would compromise network stability, $\mathcal{F}_w(e)$ is set to $+\infty$, indicating that this edge cannot be added to the knapsack as it compromises network stability. The function $\mathcal{F}_v(e)$ determines the value of edge e in three steps. First, the function identifies all pairs P of nodes for which the edge e lies on their shortest path(s) after removing the edges preceding r in the item packing sequence. Second, it calculates the shortest distance between each pair $(v_i, v_j) \in P$ both when e is present and when e is removed; these distances are denoted by d_{ij} and $d_{ij}^{(e)}$, respectively. Finally, the value of the edge is determined using Eq. (3).

$$\mathcal{F}_v(e) = \sum_{(v_i,v_j)\in P} \left(\frac{1}{d_{ij}} - \frac{1}{d_{ij}^{(e)}}\right) \quad (3)$$

The PSRC problem is reformulated as the selection of a set S of items, which meets the condition that the aggregate weight of items in S remains at most b while maximising the collective value of items in S. This particular variant is directly reducible to the PDKP. Therefore, the PSRC problem is NP-hard and cannot be efficiently approximated within a ratio of $(1 - 1/e + \epsilon)$ unless P = NP. □

IV. Solution

This section introduces a naive greedy method that iteratively selects edges, ensuring the network stability remains unaffected, while minimising the network connectivity upon their removal. Some heuristics are also suggested to enhance the efficiency of this greedy method.

A. A Naive Greedy Method

The pseudocode of the naive greedy method is shown in Algorithm 1.

In Algorithm 1, the solution set S begins as an empty set (line 1). The coreness of all nodes is computed, along with the set of candidate edges (lines 2-3). An edge is a candidate edge if its removal does not alter

Algorithm 1: Naive Greedy (NG) method

```
  Data: Graph G(V,E) and budget b
  Result: S, a set of b edges
1 S = {};
2 compute the coreness, k_i, of every node v_i ∈ V;
3 compute set R, edges whose removal does not alter the
    coreness of any node;
4 while |R| > 0 and |S| < b do
5   |  e = argmin_{r∈R} C(G^(r)) ;
6   |  S = S ∪ {e};
7   |  update the graph as G(V, E − {e});
8   |  update the set R;
9 return S;
```

the coreness of any node. Then, edges for removal are selected and added to S in iterations (lines 4-8). During each iteration, the impact of removing every candidate edge on network connectivity is assessed. The edge whose removal minimises the connectivity is added to S (lines 5-6). The graph is updated by removing the selected edge, and consequently, the candidate edge set is updated. The determination of candidate edges (in line 3) follows a procedure outlined in [36]. First, the core-strength of each node $v_i \in V$ is calculated using $CS_i = \gamma_i^{(k)} - k_i + 1$, where $\gamma_i^{(k)}$ denotes the number of neighbours of v_i with a coreness equal to or greater than that of v_i. The removal of an edge e, connecting nodes v_i and v_j, impacts the coreness of at least one node solely under two conditions: either (1) when $CS_i = 1$ and $k_j \leq k_i$, or (2) when $CS_j = 1$ and $k_i \leq k_j$. In all other cases, the edge removal does not alter the coreness of any node, maintaining network stability, and hence qualifies as a candidate edge.

In Algorithm 1, computing the coreness of all nodes and determining the candidate edge set each requires $O(|E|)$ [36], [40]. In each iteration of the algorithm, calculating the impact of removing every candidate edge on network connectivity costs $O(|R|\cdot(|V|+|E|))$, due to the Breadth-First Search algorithm used to calculate the distance between pairs of nodes requires $O(|V| + |E|)$. Therefore, considering b iterations, the total time complexity of Algorithm 1 amounts to $O(|E|+b\cdot|R|\cdot(|V|+|E|))$. This complexity can be impractical, especially in large, dense graphs. Subsequent improvements aim to reduce this time complexity by employing some heuristics.

B. Heuristics

Line 5 of Algorithm 1 computes the impact of removing every candidate edge to identify the edge whose removal minimises network connectivity. Yet, this impact calculation for every candidate edge can be performed once and stored outside the *for* loop. Consequently, in each iteration, only the impact of edges influenced by the removal of the edge selected in that iteration requires updating. To identify the candidate edges whose impact is affected after the removal of an edge, we make the following observation. For an edge e, if this edge is on the path with the shortest distance between a pair, removal of e may affect the distance between the pair; otherwise, this removal does not affect the distance between the pair. Thus, considering the betweenness of edges on the shortest paths between pairs of nodes can help identify the candidate edges whose impact is influenced by the removal of an edge and update their impact.

To enhance the time complexity of the naive method, we initiate the calculation of the impact of removing every candidate edge before the *for* loop of Algorithm 1 and store the calculation of the impact for each edge. In each iteration, using the stored impact values, we remove the edge with the highest impact from the graph. Subsequently, we identify candidate edges whose betweenness is affected (increased or decreased) due to the removal, as their impact on network connectivity may change. To efficiently identify and update the betweenness of these affected edges, we employ the method proposed in [41]. This method relies on storing essential graph structure data, including node distances from each other and lists of predecessors on shortest paths between node pairs. This approach minimises redundant computations by focusing solely on edges whose betweenness is affected by the removal of an edge. For further enhancement, if betweenness is not significantly affected by the removal of an edge (as determined by a threshold), no updates take place.

By incorporating these heuristics into the naive greedy method, we propose an enhanced greedy approach presented in Algorithm 2.

Algorithm 2: Enhanced Greedy (EG) method

```
Data: Graph G(V,E), budget b, threshold a
Result: S, a set of b edges
S = {};
compute the coreness, k_i, of every node v_i ∈ V;
compute set R, edges whose removal does not alter the
  coreness of any node;
compute B_r, betweenness of every edge r ∈ R;
compute I_r, the impact of removing every edge r ∈ R on
  the network connectivity;
while |R| > 0 and |S| < b do
  |  e = argmin_{r∈R} I_r;
  |  S = S ∪ {e};
  |  update the graph as G(V, E − e);
  |  update set R;
  |  compute R', the candidate edges whose betweenness is
  |    affected after removing e;
  |  for r ∈ R' do
  |    |  compute B'_r as the betweenness of edge r in the
  |    |    updated graph;
  |    |  if |B_r − B'_r| ≥ a then
  |    |    |  update the B_r as B_r = B'_r;
  |    |    |  update the I_r as the impact of removing r in
  |    |    |    the updated graph;
return S;
```

Initially, the enhanced algorithm calculates the coreness values for each node, identifies candidate edges for potential removal, computes the betweenness values for these candidates, and stores the impact of removing each edge (lines 2-5). Through iterations, the algorithm iterates as follows: it selects the candidate edge e with the highest impact, adds it to the solution set S, eliminates it from the graph G, and excludes it from the candidate set R (lines 7-10). Subsequently, the algorithm employs the method proposed in [41] to determine a set R' that includes candidate edges whose betweenness is affected by the removal of edge e (line 11). Within lines 12-16, for each edge $r \in R'$, the algorithm calculates the edge betweenness. Following the removal of an edge, if the difference between the stored and calculated betweenness values for an edge r does not exceed a threshold a (line 14), its betweenness value and its impact remain unchanged.

Algorithm 2 initially requires $O(2 \cdot |E|)$ to compute the coreness of every node and the candidate edge set in lines 2 and 3. Additionally, $O(|V| \cdot |E|)$ and $O(|R| \cdot (|V| + |E|))$ are needed to calculate betweenness and the impact of candidate edges in lines 4 and 5, respectively. During each iteration, updating the betweenness and impact of affected candidate edges requires $O(|R'| \cdot (|V| + |E|))$, where the size of $|R'|$ depends on the threshold a and the graph density. Therefore, the total time complexity of the enhanced greedy method is $O(|V| \cdot |E| + |R| \cdot (|V| + |E|) + b \cdot |R'| \cdot (|V| + |E|))$.

V. Core-Modularity

In this section, we introduce the *core-modularity* metric designed to quantify the extent to which network connectivity can be reduced while maintaining its stability. This metric evaluates a network's structural integrity, allowing controlled connectivity reduction during crises without compromising stability. The core-modularity metric offers a clear and quantifiable means to evaluate the stability of network structures under varying conditions. Ranging between 0 and 1, a high core-modularity value signifies a strong network structure, enabling decreased connectivity while preserving stability.

The *core-modularity* of a graph G is calculated using Eq. (4).

$$\Phi(G) = \sum_{v_i \in V} \frac{k_i}{\phi_i} \tag{4}$$

In this equation, k_i is the coreness of node v_i and ϕ_i represents the count of nodes necessary for node v_i to maintain its coreness within the network. For instance, in Figure 1, node 9 requires node 5 to retain its coreness. Node 5, in turn, relies on nodes 2 and 6 for its coreness, and nodes 2 and 6 need nodes 1, 3, and 4 for theirs. Therefore, node 9 necessitates the presence of nodes 5, 2, 6, 1, 3, and 4, totalling seven nodes (including itself) to sustain its coreness ($\phi_9 = 7$). Additionally, these seven nodes form an independent module within the network as they do not rely on nodes outside their group to retain their coreness. In general, when a node v_i is part of a module comprising n nodes, the ϕ_i value equates to n.

The pseudo-code shown in Algorithm 3 is designed

Algorithm 3: Core-module computation

```
Data: Graph G(V, E)
Result: M, a set of modules
M = {};
z = 0;
compute the coreness of every node v_i ∈ V;
while |V| > 0 do
    initialise an empty queue Q;
    select a node v_i randomly from V;
    Q.add(v_i);
    while |Q| > 0 do
        v_i = Q.remove();
        M_z.append(v_i);
        V = V − {v_i};
        for v_j in neighbours of v_i do
            if removal of v_i affects the coreness of v_j then
                Q.add(v_j);
    z = z + 1;
while there exist M_i ∩ M_j ≠ {} ∀M_i, M_j ∈ M do
    merge M_i and M_j;
return M
```

to determine a set of modules, denoted as M, within a graph G. The process of identifying a module begins by randomly selecting a node and adding the node to the module. Subsequently, a set of nodes that may have their coreness affected by the removal of the node is identified and added to the module. This iterative process continues using a queue mechanism until no nodes within the module affect a node outside the module. To identify all modules in the graph, this process repeats until no nodes remain in the graph. Ultimately, the algorithm merges modules that share common nodes; this merging process recurs until no pairs of modules contain shared nodes.

VI. Evaluation

To assess the performance of the proposed methods, namely the naive greedy method (NG) (Algorithm 1) and the enhanced greedy method (EG) (Algorithm 2), six distinct real-world networks sourced from diverse domains available in the Network Repository [42] are utilised. These networks exhibit varying properties, encompassing differences in domain applied, structural complexity, size and density, providing a broad spectrum for performance evaluation. The properties of these networks are reported in Table I. Following the name of the network, the next three columns of the table show number of nodes, number of edges, and density of the network. The column $Max(k)$ shows the maximum coreness in the network and the column $C(G)$ shows the connectivity in the original network (the network before removing any edge).

TABLE I
PROPERTIES OF THE NETWORKS USED IN THE EVALUATION

Network Name	Node size	Edge size	Density	$Max(k)$	$C(G)$
power-494-bus (PWR)	494	586	0.0048	3	14,415.88
soc-physicians (PHY)	241	923	0.0319	6	4,298.22
rt-twitter-copen (TWT)	761	1,029	0.0036	5	60,332.36
web-polblogs (WEB)	643	2,280	0.0110	13	61,860.96
fb-pages-food (FFD)	620	2,091	0.0109	12	46,375.64
ca-CSphd (CSP)	1,882	1,740	0.0010	3	57,635.01

TABLE II
RUNNING TIME (SECONDS) OF THE EG METHOD FOR DIFFERENT VALUES OF a

Network	$a = 10^{-1}$	$a = 10^{-2}$	$a = 10^{-3}$	$a = 10^{-4}$	$a = 10^{-5}$
PWR	54.85	96.80	231.36	452.99	512.70
PHY	11.64	11.72	11.97	24.12	56.14
TWT	176.74	185.80	353.83	1237.43	2815.52
WEB	385.57	396.89	679.71	1985.61	5498.41
FFD	344.37	358.93	531.78	1398.38	4016.23
CSP	438.29	440.71	585.44	1099.33	1872.34

In the absence of any alternative methods to solve the PSRC problem, we consider five baseline methods for comparative evaluation. These methods follow a simple strategy to iteratively select b edges. In each iteration, an edge is chosen for removal if its elimination does not compromise network stability. The described baseline methods are: (i) Random method (RM): Randomly selects an edge and removes it; (ii) Degree method (DM): Eliminates the edge connecting two nodes with the maximum degree and updates the nodes' degrees. (iii) Coreness method (CM): Removes the edge connecting nodes with the highest coreness; (iv) Betweenness method (BM): Removes the edge with the maximum betweenness centrality and updates the betweenness values for all edges; (v) Closeness method (CLM): Eliminates the edge connecting nodes with the highest closeness centrality and updates the closeness metric for all nodes.

Four sets of experiments are designed to assess the two proposed greedy methods, NG and EG. The first set aims to evaluate the impact of varying the threshold a on the enhanced greedy method (EG) in order to determine a good threshold value. The second set assesses the effectiveness of different methods in reducing network connectivity while maintaining stability. The third set of experiments investigates the efficiency of the methods based on their running time. Finally, the fourth set investigates the suitability of the *core-modularity* metric, defined in Eq. (4) to quantify the extent to which network connectivity can be reduced while maintaining its stability.

The first set of experiments involves varying the budget b (number of edges to be removed) from 5 to 50 in steps of 5. For each case, we apply the EG method using five different values for the threshold a: $10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}$. The connectivity obtained for different values of b and a is shown in Figure 2. Additionally, to analyse the impact of threshold variation on the running time of the EG method, we compute the method running time for each threshold value a with a fixed budget size of $b = 50$. The results are presented in Table II.

Observing Figure 2, it is evident that the EG method becomes notably more effective in reducing network connectivity as the threshold a increases. However, beyond $a > 10^{-3}$, the method shows marginal improvements. An analysis of the running times presented in Table II highlights a substantial increase in the method's execution duration for a values exceeding 10^{-3} across all networks. Consequently, considering the trade-off between effectiveness and efficiency, we opt to utilise a threshold value of $a = 10^{-3}$ in the EG method for subsequent experiments.

The second set of experiments involves comparing the effectiveness of the proposed methods against baseline methods in reducing connectivity (while maintaining stability). Across this set, the budget b ranges from 5 to 50 in steps of 5. Each case applies the seven methods to tackle the PSRC problem. The connectivity achieved by the seven methods for varying values of b is illustrated in Figure 3.

The results shown in Figure 3 demonstrate the consistent superiority of the NG method over other methods across various networks and for different b values. This superiority can be attributed to NG's comprehensive evaluation of the impact of candidate edge removals, selecting edges with the highest impact on network connectivity. However, in smaller networks, some exceptions arise. For instance, BM outperforms NG for $b = 10$ in the PWR network, and EG surpasses NG for $b = 5$ in the PHY network. These exceptions might occur because of NG's inability to guarantee optimality, stemming from its nature of looking for locally optimal choices in each iteration. Comparing EG and NG, it is apparent that the heuristics applied by EG do not harm its effectiveness in addressing the PSRC problem. Notably, the threshold a chosen for EG closely approximates the results achieved by NG.

The third set of experiments focuses on assessing the running time of the seven methods. For this assessment, a fixed budget of $b = 50$ is used, and each method is applied to solve the problem. The running time of the methods on various networks is shown in Table III.

TABLE III
RUNNING TIME (SECONDS) OF THE SEVEN METHODS

Network	RM	DM	CM	BM	CLM	NG	EG
PWR	1.11	1.86	2.84	21.06	9.46	918.76	231.36
PHY	0.33	0.34	0.35	2.84	1.28	356.66	11.97
TWT	2.14	4.34	5.36	50.60	21.50	4377.47	353.83
WEB	3.58	4.08	4.20	52.08	19.38	15102.21	679.71
FFD	2.16	3.81	3.88	45.24	18.11	12359.98	531.78
CSP	4.74	5.22	5.31	106.26	28.53	5633.36	585.44

Table III reveals that the NG method exhibits the slowest running time, attributed to its reliance on com-

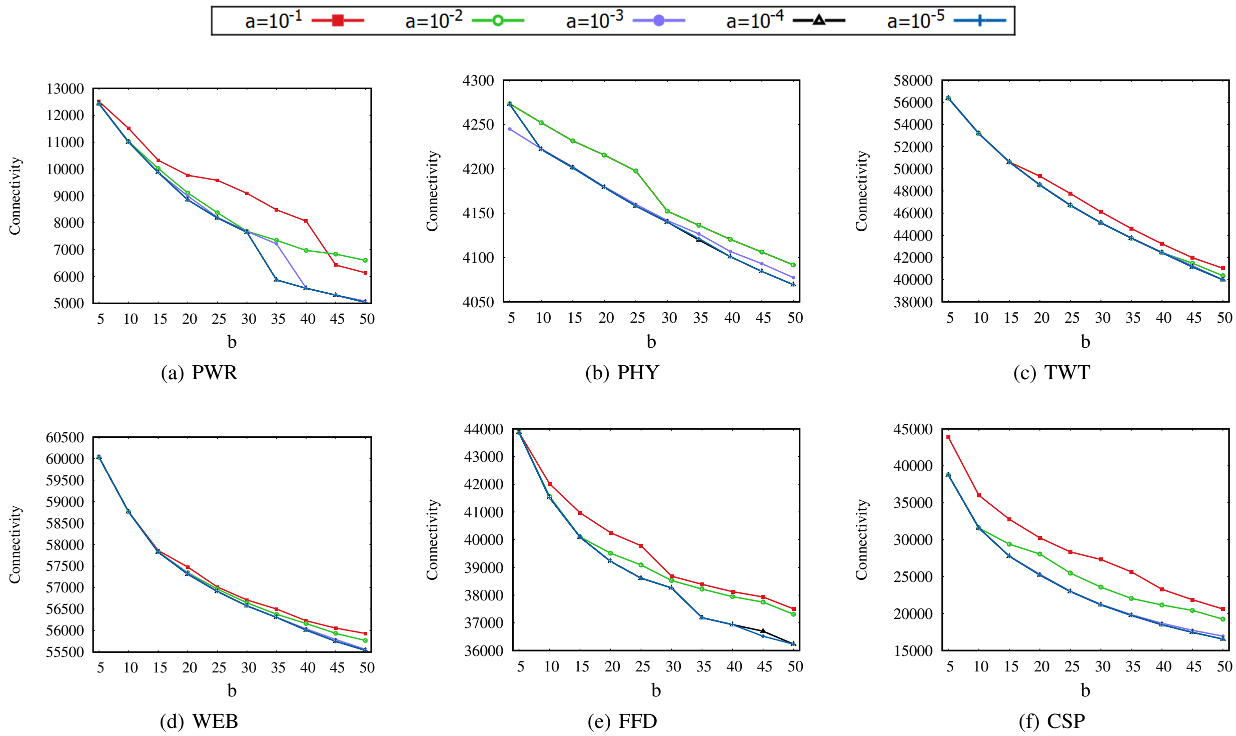

Fig. 2. Connectivity achieved by the EG method for different values of a

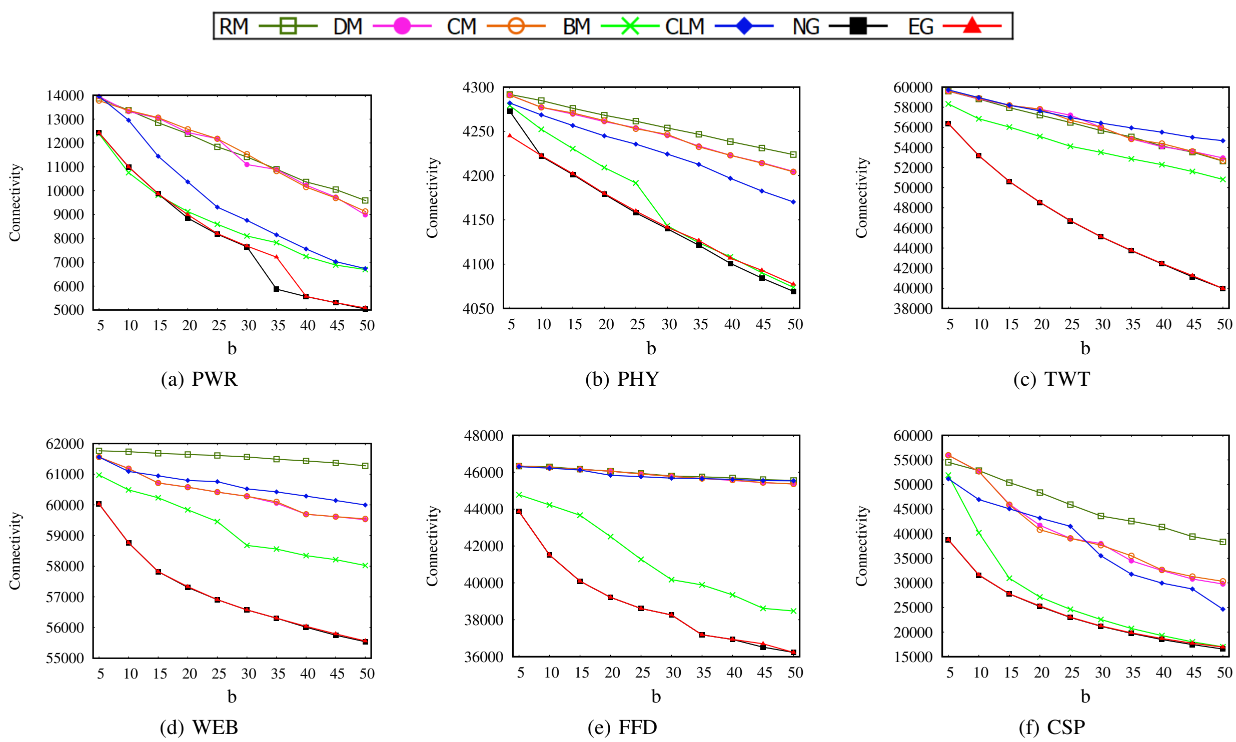

Fig. 3. Connectivity achieved by different methods

puting the impact of edges in each iteration. Conversely, the DM and CM methods, utilising simple edge selection techniques and requiring minimal computation for updates after each removal, emerge as the fastest methods after RM. It is also observed that the developed heuristics have notably enhanced the running time of EG compared to NG.

The fourth set of experiments is dedicated to exploring the *core-modularity* metric. This experiment seeks to determine the ability of the metric to quantify the extent to which network connectivity can be reduced without compromising stability. The *core-modularity* value for each network, as computed using Eq. (4), is presented in the first row of Table IV. We compare each of these values with a value indicating the reduction in network connectivity achieved by the EG method, which is the method that generally gives the best results. We calculate the reduction in connectivity of a network G using Eq. (5) as follows:

$$\text{Reduction in Connectivity}(G) = \frac{C(G) - C(G^{(S)})}{C(G)} \quad (5)$$

where $C(G)$ is the connectivity of the original network and $C(G^{(S)})$ is the connectivity of the network after $b = 50$ edges were removed using the EG method. The reduction is reported in the second row of Table IV.

TABLE IV
CORE-MODULARITY AND REDUCTION IN CONNECTIVITY

Network	PWR	PHY	TWT	WEB	FFD	CSP
Core-Modularity	0.3073	0.0940	0.2033	0.0409	0.0912	0.4723
Reduction in C	0.6475	0.0514	0.3373	0.1018	0.2188	0.7059

Table IV reveals a notable trend: networks with higher values of *core-modularity* exhibit correspondingly higher reduction values. This suggests that networks with greater *core-modularity* values can lead to a greater reduction in connectivity compared to networks with lower *core-modularity*. The calculated correlation coefficient between *core-modularity* and reduction in connectivity, reported as 0.9455, underscores the accuracy of the metric. This high correlation indicates the effectiveness of the metric in quantifying a network's ability to be managed during crises by decreasing its connectivity while maintaining its stability and functionality.

VII. Conclusion

This paper addresses the challenge of reducing network connectivity while maintaining network stability. It introduces a formal model and proposes an approach leveraging a naive greedy method, supplemented by heuristics. This method identifies specific edges for removal in order to minimise network connectivity without impacting any node's coreness. Experimental evaluations demonstrate the proposed method's efficacy and efficiency compared to different baseline approaches. Additionally, the paper proposed a core-modularity metric that provides insights into the extent to which targeted interventions can preserve network stability, ensuring that critical functions continue to operate smoothly even under challenging circumstances.

There are several directions for future exploration and investigation. First, additional investigations can explore the problem in networks with diverse properties, including directed, weighted, multilayered, or dynamic network graphs. Second, exploring alternative network connectivity metrics like diameter or eigenvalues, as well as stability metrics such as closeness or betweenness, could enrich the problem's analysis. Additionally, studying methods to increase a network's core-modularity to enhance its resilience during crises warrants further exploration. Overall, the problem examined in this paper, coupled with the core-modularity metric, offers novel insights into network design and manipulation, potentially motivating a broader analysis and enhancing comprehension of different real-world networks.

References

[1] H. Tong, B. A. Prakash, T. Eliassi-Rad, M. Faloutsos, and C. Faloutsos, "Gelling, and melting, large graphs by edge manipulation," in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. New York, NY, USA: Association for Computing Machinery, 2012, p. 245–254.

[2] S. Gaspers and K. Najeebullah, "Optimal surveillance of covert networks by minimizing inverse geodesic length," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 533–540, 7 2019.

[3] C. Nowzari, V. M. Preciado, and G. J. Pappas, "Analysis and control of epidemics: A survey of spreading processes on complex networks," *IEEE Control Systems Magazine*, vol. 36, no. 1, pp. 26–46, 2016.

[4] Q. Zhu, Z. Zhu, Y. Qi, H. Yu, and Y. Xu, "Optimization of cascading failure on complex network based on NNIA," *Physica A: Statistical Mechanics and its Applications*, vol. 501, pp. 42–51, 2018.

[5] A. Zareie and R. Sakellariou, "Minimizing the spread of misinformation in online social networks: A survey," *Journal of Network and Computer Applications*, vol. 186, p. 103094, 2021.

[6] Z. Zhou, F. Zhang, X. Lin, W. Zhang, and C. Chen, "K-core maximization: An edge addition approach," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. AAAI Press, 2019, p. 4867–4873.

[7] S. Teng, J. Xie, F. Zhang, C. Lu, J. Fang, and K. Wang, "Optimizing network resilience via vertex anchoring," in *Proceedings of the ACM Web Conference 2024*. New York, NY, USA: Association for Computing Machinery, 2024, p. 606–617.

[8] F. Zhang, Q. Linghu, J. Xie, K. Wang, X. Lin, and W. Zhang, "Quantifying node importance over network structural stability," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2023, p. 3217–3228.

[9] P. Dey, S. K. Maity, S. Medya, and A. Silva, "Network robustness via global k-cores," in *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2021, p. 438–446.

[10] R. Chitnis and N. Talmon, "Can we create large k-cores by adding few edges?" in *International Computer Science Symposium in Russia*, Springer. Switzerland: Springer, Cham, 2018, pp. 78–89.

[11] R. Laishram, A. E. Sar, T. Eliassi-Rad, A. Pinar, and S. Soundarajan, "Residual core maximization: An efficient algorithm for maximizing the size of the k-core," in *Proceedings of the 2020 SIAM International Conference on Data Mining*. SIAM, 2020, pp. 325–333.

[12] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, 2003.

[13] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak Mathematical Journal*, vol. 23, no. 2, pp. 298–305, 1973.

[14] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world'networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[15] Y. Shen, N. P. Nguyen, Y. Xuan, and M. T. Thai, "On the discovery of critical links and nodes for assessing network vulnerability," *IEEE/ACM Transactions on Networking*, vol. 21, no. 3, pp. 963–973, 2013.

[16] O. Ivanciuc, T.-S. Balaban, and A. T. Balaban, "Reciprocal distance matrix, related local vertex invariants and topological indices," *Journal of Mathematical Chemistry*, vol. 12, no. 1, pp. 309–318, 1993.

[17] C. M. Schneider, T. Mihaljev, S. Havlin, and H. J. Herrmann, "Suppressing epidemics with a limited amount of immunization units," *Phys. Rev. E*, vol. 84, p. 061911, Dec 2011.

[18] A. Zareie and R. Sakellariou, "Rumour spread minimization in social networks: A source-ignorant approach," *Online Social Networks and Media*, vol. 29, p. 100206, 2022.

[19] S. Saha, A. Adiga, B. A. Prakash, and A. K. S. Vullikanti, "Approximation algorithms for reducing the spectral radius to control epidemic spread," in *Proceedings of the 2015 SIAM International Conference on Data Mining*. SIAM, 2015, pp. 568–576.

[20] Z. Zhang, Z. Zhang, and G. Chen, "Minimizing spectral radius of non-backtracking matrix by edge removal," in *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*. New York, NY, USA: Association for Computing Machinery, 2021, p. 2657–2667.

[21] P. Van Mieghem, D. Stevanović, F. Kuipers, C. Li, R. van de Bovenkamp, D. Liu, and H. Wang, "Decreasing the spectral radius of a graph by link removals," *Phys. Rev. E*, vol. 84, p. 016101, Jul 2011.

[22] A. A. Schoone, H. L. Bodlaender, and J. Van Leeuwen, "Diameter increase caused by edge deletion," *Journal of Graph Theory*, vol. 11, no. 3, pp. 409–427, 1987.

[23] M. Rayatidamavandi, F. Conlon, and M. Rahnamay-Naeini, "Reducing network vulnerability to malicious attacks," in *2018 International Conference on Computing, Networking and Communications*. IEEE, 2018, pp. 718–723.

[24] R.-H. Li and J. X. Yu, "Triangle minimization in large networks," *Knowledge and Information Systems*, vol. 45, no. 3, pp. 617–643, 2015.

[25] K. Najeebullah, "Complexity of optimally defending and attacking a network," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[26] P. Holme, B. J. Kim, C. N. Yoon, and S. K. Han, "Attack vulnerability of complex networks," *Phys. Rev. E*, vol. 65, p. 056109, May 2002.

[27] L. C. Freeman *et al.*, "Centrality in social networks: Conceptual clarification," *Social network: critical concepts in sociology*, vol. 1, pp. 238–263, 2002.

[28] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, vol. 40, no. 1, pp. 35–41, 1977.

[29] S. B. Seidman, "Network structure and minimum degree," *Social Networks*, vol. 5, no. 3, pp. 269–287, 1983.

[30] Y.-X. Kong, G.-Y. Shi, R.-J. Wu, and Y.-C. Zhang, "k-core: Theories and applications," *Physics Reports*, vol. 832, pp. 1–32, 2019.

[31] R. Sun, C. Chen, X. Wang, Y. Zhang, and X. Wang, "Stable community detection in signed social networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 10, pp. 5051–5055, 2022.

[32] C. Chen, Q. Zhu, R. Sun, X. Wang, and Y. Wu, "Edge manipulation approaches for k-core minimization: Metrics and analytics," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 1, pp. 390–403, 2023.

[33] W. Zhu, C. Chen, X. Wang, and X. Lin, "K-core minimization: An edge manipulation approach," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. New York, NY, USA: Association for Computing Machinery, 2018, p. 1667–1670.

[34] S. Medya, T. Ma, A. Silva, and A. Singh, "A game theoretic approach for k-core minimization," in *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2020, p. 1922–1924.

[35] Y. Liu, M. Tang, T. Zhou, and Y. Do, "Improving the accuracy of the k-shell method by removing redundant links: From a perspective of spreading dynamics," *Scientific reports*, vol. 5, no. 1, p. 13172, 2015.

[36] R. Laishram and S. Soundarajan, "On finding and analyzing the backbone of the k-core structure of a graph," in *2022 IEEE International Conference on Data Mining*, 2022, pp. 1017–1022.

[37] S. Freitas, D. Yang, S. Kumar, H. Tong, and D. H. Chau, "Graph vulnerability and robustness: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 6, pp. 5915–5934, 2023.

[38] S. Martello and P. Toth, *Knapsack problems: algorithms and computer implementations*. John Wiley & Sons, Inc., 1990.

[39] S. Gawiejnowicz, N. Halman, and H. Kellerer, "Knapsack problems with position-dependent item weights or profits," *Annals of Operations Research*, vol. 326, no. 1, pp. 137–156, 2023.

[40] D. Wen, L. Qin, Y. Zhang, X. Lin, and J. X. Yu, "I/o efficient core graph decomposition at web scale," in *2016 IEEE 32nd International Conference on Data Engineering*, 2016, pp. 133–144.

[41] N. Kourtellis, G. D. F. Morales, and F. Bonchi, "Scalable online betweenness centrality in evolving graphs," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 9, pp. 2494–2506, 2015.

[42] R. A. Rossi and N. K. Ahmed, "The network data repository with interactive graph analytics and visualization," in *AAAI*, 2015. [Online]. Available: http://networkrepository.com