The recoverability of network controllability with respect to node additions

Wang, Fenghua; Kooij, Robert E.

DOI
10.1088/1367-2630/ad0170

Publication date
2023

Document Version
Final published version

Published in
New Journal of Physics

New Journal of Physics
The open access journal at the forefront of physics

Deutsche Physikalische Gesellschaft DPG IOP Institute of Physics

PAPER • OPEN ACCESS

The recoverability of network controllability with respect to node additions

To cite this article: Fenghua Wang and Robert E Kooij 2023 *New J. Phys.* **25** 103034

IOP Publishing New J. Phys. 25 (2023) 103034 https://doi.org/10.1088/1367-2630/ad0170

New Journal of Physics
The open access journal at the forefront of physics

Deutsche Physikalische Gesellschaft DPG
IOP Institute of Physics

Published in partnership with: Deutsche Physikalische Gesellschaft and the Institute of Physics

OPEN ACCESS

RECEIVED
17 March 2023

REVISED
29 September 2023

ACCEPTED FOR PUBLICATION
9 October 2023

PUBLISHED
19 October 2023

PAPER

The recoverability of network controllability with respect to node additions

Fenghua Wang[1,*] **and Robert E Kooij**[1,2]

[1] Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, 2628 CD Delft, The Netherlands
[2] Unit ICT, Strategy and Policy, Netherlands Organization for Applied Scientific Research (TNO), 2595 DA Den Haag, The Netherlands
[*] Author to whom any correspondence should be addressed.

E-mail: F.wang-8@tudelft.nl

Keywords: recoverability, network controllability, network resilience, recovery strategies

Abstract

Network controllability is a critical attribute of dynamic networked systems. Investigating methods to restore network controllability after network degradation is crucial for enhancing system resilience. In this study, we develop an analytical method based on degree distributions to estimate the minimum fraction of required driver nodes for network controllability under random node additions after the random removal of a subset of nodes. The outcomes of our method closely align with numerical simulation results for both synthetic and real-world networks. Additionally, we compare the efficacy of various node recovery strategies across directed Erdös–Rényi (ER) networks, swarm signaling networks (SSNs), and directed Barabàsi Albert (BA) networks. Our findings indicate that the most efficient recovery strategy for directed ER networks and SSNs is the greedy strategy, which considers node betweenness centrality. Similarly, for directed BA networks, the greedy strategy focusing on node degree centrality emerges as the most efficient. These strategies outperform recovery approaches based on degree centrality or betweenness centrality, as well as the strategy involving random node additions.

1. Introduction

Network controllability has been extensively investigated [1], particularly due to its applicability to various complex systems that can be represented as networks. These include domains such as power grids [2], transportation systems [3], and telecommunication systems [4]. Controllability represents an important characteristic of such systems, affording the ability to achieve varied control objectives. For example, manipulating approximately 17% neurons in the *C. elegans* worm can elicit coordinated body responses, while controlling 5% of a swarm of honeybees can guide the swarm to new destinations [5]. However, controllability can falter in the face of malicious attacks or natural catastrophes, resulting from the failure of system components [6]. To improve the resilience of the system against attacks [7], bolstering its robustness becomes paramount. Moreover, there is a pressing need to explore strategies for efficiently restoring failed components to ensure controllability within the system [8].

Network controllability discussed in this research pertains to the concept of structural controllability within directed networks, which do not contain self-loops. In the domain of control theory, a system is considered controllable if it can transition from an initial state to any desired state in a finite time by applying external inputs [9]. Lin introduced the notion of structural controllability [10], where a system exhibiting structural controllability maintains a high likelihood of controllability even after modifying the weights of interconnections. Exploring the intricate interplay between network topology and controllability, Liu *et al* [11] devised the framework of structural controllability for directed networked systems. This framework focuses on injecting specific external input nodes to achieve full system controllability. Importantly, it is worth noting that network controllability differs from the widely recognized concept of 'pinning controllability' [5]. The latter explores methods for driving the system to specific states by manipulating

specific nodes. For example, network synchronization explores whether all nodes can exhibit identical dynamic trajectories [12] and investigations of network consensus problems [13] aim to determine strategies to guide all nodes towards the same state.

Errors or attacks within a system can cause a degradation in network performance [14]. Effective and efficient recovery of networks after attacks has gained considerable attention [15]. For example, Shang has explored local recovery strategies from a network percolation perspective [16], as well as strategies for restoring consensus in nonlinear multiagent systems [8]. Moreover, He *et al* [17] have defined network recoverability as a network's capacity to revert to a desired state after facing disruptions. In the context of network recoverability with a focus on network controllability, Chen *et al* [18] explored efficient recovery strategies after random link removals. Their study revealed that the greedy recovery strategy outperforms degree-based and eigenvector-based recovery strategies. Additionally, they introduced an analytical method based on degree distributions to predict network controllability during recovery. However, their investigation did not include the effective recovery strategy of nodes after node failures, a scenario frequently observed in real-life networks. Addressing this gap, our study aims to predict network controllability under random node additions and explore efficient strategies for node recovery in network controllability.

Since network recovery is the reverse process of attacking or disrupting networks, we can derive various recovery strategies by drawing insights from the attack process. Researchers have explored efficient strategies to undermine network controllability and methods to forecast network controllability under attack scenarios. Targeted attacks are generally more detrimental than random attacks [6]. Pu *et al* [19] demonstrated that node removals based on node degrees are more harmful than random node attacks in directed Erdös–Rényi (ER) and scale-free (SF) networks. Directed ER networks are synthetic networks generated by randomly placing directed links, while directed SF networks are generated to ensure that the degree distributions follow power-law distributions. Wang *et al* [20] found that intentionally attacking bridge links, whose removal can disconnect the network, effectively disrupts network controllability compared to link removals based on node degrees and distances in directed ER and SF networks. Critical nodes and links are identified based on their propensity to increase the number of driver nodes required for network controllability after removals, where driver nodes are defined as nodes where external inputs are injected [11]. Building on this, Lou *et al* [21] developed a hierarchical attack framework that incorporates critical nodes or links. In this framework, nodes or links are removed based on categorical priorities, and within the same category, nodes or links with higher centrality values are removed first. Their findings presented that the destructiveness of this attack framework is stronger than strategies that solely leverage centrality features like node degrees or betweenness when targeting nodes or links. Given that attack strategies considering degree and betweenness are extensively investigated, we aim to explore the effectiveness of different degree-based and betweenness-based recovery strategies in terms of network controllability after node removals.

Several techniques have been employed to predict the minimum number of driver nodes required under attack scenarios, including regression models, analytical methods using degree distributions, and machine learning approaches. Sun *et al* [6] introduced linear regressions for removal fractions less than l_c and quadratic regressions for fractions greater than l_c (where l_c is the fraction of critical links) to approximate the fraction of driver nodes for random and targeted link removals based on critical links. Liu *et al* [11] proposed an analytical method based on degree distributions to estimate the minimum fraction of driver nodes. Chen *et al* [18] presented an analytical method using degree distributions for random link removals. Dhiman *et al* [22] utilized an artificial neural network to predict the minimum number of driver nodes under link-targeted removals, outperforming analytical methods based on critical links. Lou *et al* [23] predicted controllability robustness under targeted attacks by utilizing convolutional neural networks, which process the adjacency matrix as a grayscale image. The performance of regression models are worse than the analytical methods using degree distributions and machine learning approaches. While machine learning methods require training data, analytical methods offer time and computational cost savings. This encourages us to develop an analytical method based on degree distributions to predict network controllability during the random node recovery process.

In this study, we focus on the recovery process of network controllability after random node removals. We propose an analytical method based on degree distributions to approximate the number of driver nodes required during random node additions. To validate our analytical approach, we apply it to both synthetic and real-world networks. Additionally, we investigate six other node recovery strategies on synthetic networks: degree-based recovery strategy, betweenness-based recovery strategy, updated degree-based recovery strategy, updated betweenness recovery strategy, greedy degree-based recovery strategy, and greedy betweenness-based recovery strategy. To measure the efficiency of a recovery strategy for network controllability, we utilize two modified recoverability indicators of the recovery process [21, 24].

The remainder of the paper is the following. Section 2 provides a detailed description of the networks utilized in this study. Section 3 outlines the attack scenario and the recovery strategies employed, the

introduction of network controllability, and the two recoverability indicators used in this study. Section 4 presents the analytical approximation for network controllability under random node additions and based on the recoverability indicators, we compare and evaluate the efficiency of different recovery strategies on synthetic networks. The last section of the article is dedicated to conclusions and discussions.

2. Network data

In this study, we evaluate the effectiveness and efficiency of our proposed methods by applying them to synthetic networks and real-world communication networks.

2.1. Synthetic networks

The synthetic networks under investigation comprise directed ER networks, swarm signaling networks (SSNs), directed Barabàsi-Albert (BA) networks and directed SF networks.

(i) Directed ER networks
Directed ER networks with N nodes are constructed by randomly placing directed links between any two nodes with a given probability p_{ER}. Both the in-degree and out-degree distributions of the generated directed ER network follow the Poisson distribution. In this research, two directed ER networks have been generated with $N = 500, p_{\mathrm{ER}} = 0.007$ and $N = 1000, p_{\mathrm{ER}} = 0.004$, respectively, where the average total degree is 7 and 8, correspondingly.

(ii) SSNs
In this study, we employ the topology of SSNs proposed and developed by [25, 26]. The SSN exhibits a regular out-degree distribution, while its in-degree distribution follows a Poisson distribution. To generate SSNs, we specify two parameters: the number of nodes N and the out-degree value k. Each node randomly creates k outgoing links to other nodes. Specifically, we generated SSNs with $N = 500, k = 2$ and $N = 500, k = 4$. The total average degree is 4 and 8, respectively.

(iii) Directed BA networks
To generate a directed BA network, we first generate an undirected BA network [27] by giving two parameters: the number of nodes N and the number of links m that a new node preferentially attaches to existing nodes with high degrees. The initial network is a star network with $m + 1$ nodes. Once the undirected BA network is established, we proceed to randomize the orientations of links, thereby transforming the network into a directed structure. We generated directed BA graphs with $N = 500, m = 2$ and $N = 500, m = 4$, respectively, where the total average degree is 4 and 8 respectively.

(iv) Directed SF network
SF networks have power-law degree distributions, which are characterized by a specific power-law exponent γ and the minimum value of the degree α. To generate SF networks, we first generate a power-law degree sequence using the Python package *powerlaw* [28]. Next, we use the configuration model [29] to generate a digraph and remove self-loop links. To ensure that the generated network conforms to the power-law distribution, we use the same Python package to fit the degree distributions. We only use generated networks that have a difference between the exponent used and the average fitting power-law exponent of the in-degree and out-degree distribution smaller than 0.01. In this study, we choose two SF networks with 10 000 nodes, one with $\gamma = 2.3, \alpha = 3$ and the other one with $\gamma = 3, \alpha = 3$. The average total degrees are around 22 and 10.3, respectively.

2.2. Real-world networks

For the real-world networks, we choose 202 communication networks from the Internet Topology Zoo data set [30], whose number of nodes ranges from 11 to 754. To change undirected communication networks into directed networks, based on the node attribute: source node or target node [6], we assign the direction of the link from the source node to the target node. The properties of the 202 communication networks, in terms of number of nodes, number of links and average degree, are depicted in figure 1. Apart from the small and medium-sized communication networks, we incorporate an additional seven larger directed networks obtained from the network data repository [31] and the SNAP dataset collection [32]. These selected networks originate from diverse domains, such as the world wide web (WebSpam [33] and Indochina [34]), a Wikipedia adminship election dataset (Wiki Vote [35]), a retweet network dataset (Qatif [36]), an E-mail network dataset (Email Eu core [37]), and internet peer-to-peer network datasets (p2p Gnutella25 [38] and p2p Gnutella08 [39]). Essential details such as the number of nodes (N) and links (L) and the average degree (d_{av}) of these seven larger networks are presented in table 1.

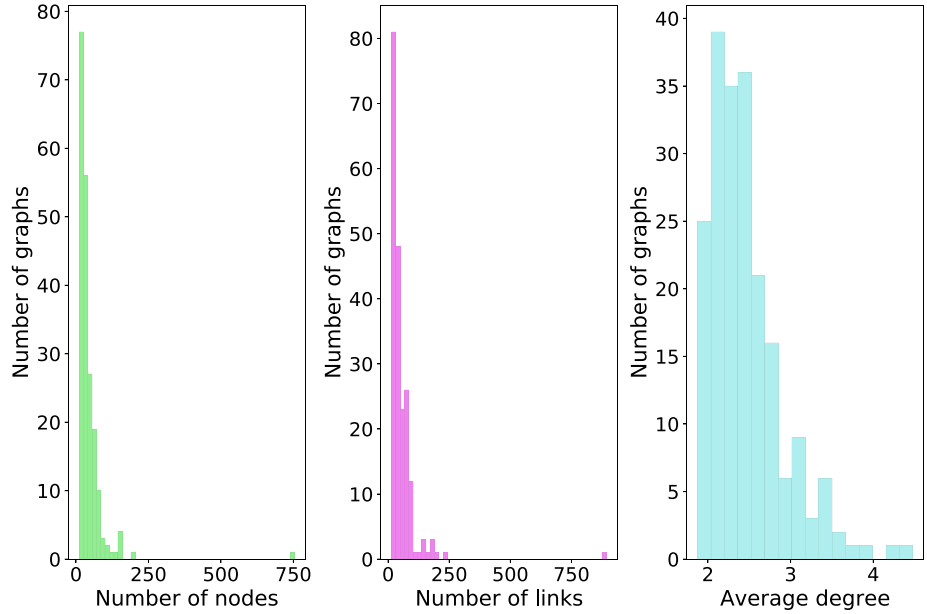


Figure 1. Properties of 202 networks from the Internet Topology Zoo data set.

Table 1. Properties of seven real-world networks.

Name	N	L	d_{av}
Qatif [31]	7537	8568	2.274
p2p Gnutella25 [32]	22 663	54 693	4.827
p2p Gnutella08 [32]	6299	20 776	6.597
Indochina [31]	11 358	47 606	8.383
WebSpam [31]	4767	37 375	15.681
Wiki Vote [32]	7066	141 779	40.130
Email Eu core [32]	986	46 771	94.870

3. Preliminaries

3.1. Attack and recovery scenarios

In this study, the network attack process is executed iteratively. At each time step, a node is uniformly and randomly chosen and subsequently removed. Concurrently, the links connected to other nodes are eliminated when the selected node is removed. We stop removing nodes when 15% of the nodes are removed from the network.

In the recovery phase, we employ seven distinct recovery strategies within our investigation: random recovery strategy, degree-based recovery strategy, betweenness-based strategy, updated degree recovery strategy, updated betweenness recovery strategy, greedy-degree recovery strategy, and greedy-betweenness recovery strategy. When implementing these strategies, we focus on restoring the nodes. At each step, a single node is recovered along with its previously removed links that connect to nodes still present in the attacked graph based on the recovery strategy. We persist in adding back the removed nodes until the original network is fully restored.

The random recovery strategy involves selecting a node uniformly and randomly from the set of removed nodes at each step. This chosen node is then added to the attacked network. On the other hand, the degree-based recovery strategy relies on the degree information derived from the initial graph (i.e. the graph prior to the attack). The procedure entails ranking the removed nodes based on their degree values in the original network. Throughout the recovery phase, these nodes are gradually reintroduced to the network in accordance with their degree ranks and their original connections. Similarly, the betweenness recovery strategy is rooted in betweenness centrality in the original graph. Nodes that have been removed are ranked according to their betweenness values as calculated from the original network. Nodes with higher betweenness centrality rankings are afforded higher priority during the recovery process, and they are added into the network earlier, including their original connections with other existing nodes in the attacked graph.

The updated degree recovery strategy involves selecting a removed node at each time step that, upon reintegration into the attacked network, will possess the highest degree compared to other removed nodes undergoing the same process. In cases where multiple nodes would have the same highest degree after reintegration, their degrees in the original network are compared. The node with the highest original degree is prioritized for the addition. Should the degrees in the original network be equal, a random selection between the nodes is made. Similarly, the updated betweenness recovery strategy follows a comparable approach. The key distinction lies in the use of betweenness values instead of degree values at each step for selecting the node to be reintroduced into the network.

The greedy-degree recovery strategy operates by selecting a removed node from the set in each step to minimize the number of driver nodes most effectively. If multiple nodes offer the same potential reduction in the minimum number of driver nodes, the original degrees of the removed nodes are compared. The node with the higher initial degree is given priority for reintegration. If removed nodes yield an equal reduction in the minimum number of driver nodes and have identical initial degrees, a random selection determines which node is added back. Similarly, the greedy-betweenness recovery strategy follows a similar approach. However, instead of relying on initial degrees as a determining factor for reintegration, the initial betweenness values of the removed nodes are used.

3.2. Network controllability

Consider a linear, time-invariant networked system composed of N nodes, governed by the following equation:

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = Ax(t) + Bu(t). \tag{1}$$

Here, the $N \times 1$ vector $x(t) = (x_1(t), x_2(t), \ldots, x_N(t))^T$ represents the state of each node. The matrix A, with dimensions $N \times N$, characterizes the connections between nodes with corresponding strength. Furthermore, the $N \times M$ matrix B serves as the input matrix, indicating which nodes are under direct control through the $M \times 1$ control input vector $u(t) = (u_1(t), u_2(t), u_3(t), \ldots, u_M(t))^T$.

A linear, time-invariant networked system is considered controllable if its node states can be manipulated to reach any desired state within a finite time by applying a set of external inputs. The Kalman rank criterion provides a way to determine controllability, where the rank of the controllability matrix $[B, AB, A^2B, \ldots, A^{N-1}B]$ should be equal to N for the system to be fully controllable [9]. To gain an understanding of the Kalman rank criterion, we can derive the formal solution of equation (1) with an initial condition of $x(0) = \mathbf{0}$ as $x(t) = \int_0^\infty \mathrm{e}^{A(t-\tau)}Bu(\tau)\,\mathrm{d}\tau$. By expanding $\mathrm{e}^{A(t-\tau)}$ into a series, we can deduce that $x(t)$ is a linear combination of the matrix $[B, AB, A^2B, \ldots, A^{N'}B, \ldots]$. According to the Cayley–Hamilton theorem, for $N' > N$, the rank of the matrix $[B, AB, A^2B, \ldots, A^{N'}B, \ldots]$ is equivalent to the rank of the controllability matrix $[B, AB, A^2B, \ldots, A^{N-1}B]$. Consequently, if the rank of the controllability matrix is less than N, it implies that the matrix $[B, AB, A^2B, \ldots, A^{N'}B, \ldots]$ cannot span the state space of dimension N entirely. In such cases, an input $u(t)$ cannot be found to steer $x(0)$ to an arbitrary state $x(t)$ [40]. In practical applications, the implementation of the Kalman rank criterion poses challenges due to the requirement of obtaining information about the network's interaction strengths and the involvement of computationally intensive calculations, especially for large-scale networks. To mitigate these challenges, Lin [10] introduced the concept of structural controllability. Additionally, Liu *et al* [11] presented the maximum matching method and the minimum inputs theorem to determine the minimum number of nodes (driver nodes) that must be controlled to ensure controllability. To determine the count of driver nodes, a directed network should first be transformed into a bipartite network. Subsequently, a maximum matching edge set can be derived using the maximum matching algorithm [41], consisting of N_M directed edges without shared source nodes or end nodes. The end nodes of the matching edges are termed matched nodes, while the remaining nodes are unmatched. The calculation of the minimum number N_D of driver nodes is as follows

$$N_D = \max\{1, N - N_M\}. \tag{2}$$

3.3. Analytical approximations of the number of driver nodes

According to Liu *et al* [11], under the assumptions of no self-loops and absence of degree correlations among nodes, for a directed network represented by $\mathcal{G}(N, L)$ with N nodes and L links, the minimum fraction of driver nodes can be approximated by using generating functions of in- and out-degree distributions ($G_{\mathrm{in}}(x)$ and $G_{\mathrm{out}}(x)$, respectively) as well as excess in- and out-degree distributions ($H_{\mathrm{in}}(x)$ and $H_{\mathrm{out}}(x)$, respectively). The aforementioned generating functions are defined as follows:

$$
\begin{aligned}
G_{\text{in}}(x) &= \sum_{k=0}^{\infty} P_{\text{in}}(k_{\text{in}})\, x^{k_{\text{in}}}, \\
G_{\text{out}}(x) &= \sum_{k=0}^{\infty} P_{\text{out}}(k_{\text{out}})\, x^{k_{\text{out}}}, \\
H_{\text{in}}(x) &= \frac{\sum_{k=1}^{\infty} k_{\text{in}} P_{\text{in}}(k_{\text{in}})\, x^{k_{\text{in}}-1}}{< k_{\text{in}} >} = \frac{G'_{\text{in}}(x)}{G'_{\text{in}}(1)}, \\
H_{\text{out}}(x) &= \frac{\sum_{k=1}^{\infty} k_{\text{out}} P_{\text{out}}(k_{\text{out}})\, x^{k_{\text{out}}-1}}{< k_{\text{out}} >} = \frac{G'_{\text{out}}(x)}{G'_{\text{out}}(1)},
\end{aligned}
\tag{3}
$$

where k_{in} and k_{out} correspond to in- and out-degree, respectively, and $P_{\text{in}}(\cdot)$ and $P_{\text{out}}(\cdot)$ signify in- and out-degree probability distribution, respectively. Then the minimum fraction of driver nodes is given by:

$$
\begin{aligned}
n_d = \frac{1}{2} \{ & G_{\text{in}}(\omega_2) + G_{\text{in}}(1-\omega_1) - 2 + G_{\text{out}}(\hat{\omega}_2) + G_{\text{out}}(1-\hat{\omega}_1) \\
& + k[\hat{\omega}_1(1-\omega_2) + \omega_1(1-\hat{\omega}_2)]\},
\end{aligned}
\tag{4}
$$

where ω_1, ω_2, $\hat{\omega}_1$ and $\hat{\omega}_2$ satisfy

$$
\begin{aligned}
\omega_1 &= H_{\text{out}}(\hat{\omega}_2), \\
\omega_2 &= 1 - H_{\text{out}}(1-\hat{\omega}_1), \\
\hat{\omega}_1 &= H_{\text{in}}(\omega_2), \\
\hat{\omega}_2 &= 1 - H_{\text{in}}(1-\omega_1),
\end{aligned}
\tag{5}
$$

and k denotes half of the average degree equal to the average in-degree and the average out-degree, $k = \frac{1}{2} < k > = < k_{\text{in}} > = < k_{\text{out}} >$.

Under node removals, the driver nodes can be classified into two categories. The first category consists of N_D driver nodes controlling the remaining network, while the second category includes N_r removed nodes with the assumption that each removed node should be controlled separately. We define the fraction of driver nodes n_D as $n_D = \frac{N_D + N_r}{N}$. After randomly removing a fraction p of nodes in the network, the fraction of driver nodes n_D satisfies

$$
n_D = \frac{n_d(1-p)N + pN}{N} = n_d(1-p) + p. \tag{6}
$$

3.3.1. Analytical approximations under random node removals and random node additions

Based on the research of Shao *et al* [42], the generating function following the random removal of a fraction p of nodes is analogous to the initial generating function but with the modified argument $\bar{x} = p + (1-p)x$. Consequently, the generating functions of in- and out-degree, as well as excess in- and out-degree, are updated as follows after randomly removing a proportion p of nodes:

$$
\begin{aligned}
\bar{G}_{\text{in}}(x) &= G_{\text{in}}(p + (1-p)x), \\
\bar{G}_{\text{out}}(x) &= G_{\text{out}}(p + (1-p)x), \\
\bar{H}_{\text{in}}(x) &= \frac{\bar{G}'_{\text{in}}(x)}{\bar{G}'_{\text{in}}(1)}, \\
\bar{H}_{\text{out}}(x) &= \frac{\bar{G}'_{\text{out}}(x)}{\bar{G}'_{\text{out}}(1)}.
\end{aligned}
\tag{7}
$$

Next, we use equations (4) and (6) to acquire the fraction of minimum number of nodes n_D after randomly removing a fraction p of nodes:

$$
\begin{aligned}
n_D = \frac{1}{2}(1-p) \{ & \bar{G}_{\text{in}}(\omega_2) + \bar{G}_{\text{in}}(1-\omega_1) - 2 + \bar{G}_{\text{out}}(\hat{\omega}_2) + \bar{G}_{\text{out}}(1-\hat{\omega}_1) \\
& + k(1-p)[\hat{\omega}_1(1-\omega_2) + \omega_1(1-\hat{\omega}_2)]\} + p,
\end{aligned}
\tag{8}
$$

where ω_1, ω_2, $\hat{\omega}_1$ and $\hat{\omega}_2$ satisfy

$$
\begin{aligned}
\omega_1 &= \bar{H}_{\text{out}}(\hat{\omega}_2), \\
\omega_2 &= 1 - \bar{H}_{\text{out}}(1-\hat{\omega}_1), \\
\hat{\omega}_1 &= \bar{H}_{\text{in}}(\omega_2), \\
\hat{\omega}_2 &= 1 - \bar{H}_{\text{in}}(1-\omega_1),
\end{aligned}
\tag{9}
$$

and k is half of the average degree equal to the average in-degree and the average out-degree, $k=\frac{1}{2}<k>=<k_{\text{in}}>=<k_{\text{out}}>$.

In this study, we will denote a network perturbation, either a node removal or a node addition, as a challenge. This study uses challenge K to present the number of manipulations under node removals or additions. A manipulation represents a node removal or a node addition. Challenge K represents that a fraction $p=\frac{K}{N}$ of nodes was removed during the removal process. Hence $K=0$ corresponds to the graph in the initial state before the attack. Then the minimum fraction of driver nodes at challenge K satisfies

$$\begin{aligned} n_D(K)=\frac{1}{2}\left(1-\frac{K}{N}\right)\Big\{&\bar{G}_{\text{in}}(\omega_2)+\bar{G}_{\text{in}}(1-\omega_1)-2+\bar{G}_{\text{out}}(\hat{\omega_2})+\bar{G}_{\text{out}}(1-\hat{\omega_1})\\ &+k\left(1-\frac{K}{N}\right)[\hat{\omega_1}(1-\omega_2)+\omega_1(1-\hat{\omega_2})]\Big\}+\frac{K}{N}, \end{aligned} \tag{10}$$

satisfying equation (9).

Under random node additions, suppose the total number of removed nodes (challenges) during the attack process is K_a, the total number of nodes added back at challenge K is $K-K_a$, and the fraction of removed nodes p at challenge K is equal to $p=\frac{2K_a}{N}-\frac{K}{N}$. Therefore, during random additions, the minimum fraction of driver nodes at challenge K is

$$\begin{aligned} n_D(K)=\frac{1}{2}\left(1-\frac{2K_a}{N}+\frac{K}{N}\right)\Big\{&\bar{G}_{\text{in}}(\omega_2)+\bar{G}_{\text{in}}(1-\omega_1)-2+\bar{G}_{\text{out}}(\hat{\omega_2})+\bar{G}_{\text{out}}(1-\hat{\omega_1})\\ &+k\left(1-\frac{2K_a}{N}+\frac{K}{N}\right)[\hat{\omega_1}(1-\omega_2)+\omega_1(1-\hat{\omega_2})]\Big\}+\frac{2K_a}{N}-\frac{K}{N}, \end{aligned} \tag{11}$$

satisfying equation (9).

(i) Directed ER networks

Both the in-degree distribution $P_{\text{in}}(k_{\text{in}})$ and the out-degree distribution $P_{\text{out}}(k_{\text{out}})$ of ER networks follow a Poisson distribution with average degree k. Therefore, the generating functions of in-degree and out-degree are as follows,

$$G_{\text{in}}(x)=\mathrm{e}^{-k(-x+1)}, G_{\text{out}}(x)=\mathrm{e}^{-k(-x+1)}. \tag{12}$$

The minimum fraction of driver nodes n_D at challenge K under random removals in the ER networks can be obtained through equations (7), (10) and (9) as

$$n_D(K)=\frac{K}{N}+\frac{K}{N}\omega_2-\omega_2+\left[1-\frac{K}{N}+k\left(1-\frac{K}{N}\right)^2(1-\omega_2)\right]\mathrm{e}^{k\left(1-\frac{K}{N}\right)(\omega_2-1)} \tag{13}$$

where ω_2 satisfies $1-\omega_2-\mathrm{e}^{-k\left(1-\frac{K}{N}\right)}\mathrm{e}^{-k\left(1-\frac{K}{N}\right)(1-\omega_2)}=0$.

Then, the minimum fraction of driver nodes n_D at challenge K under random additions satisfies

$$\begin{aligned} n_D(K)=&\left[1-\frac{2K_a}{N}+\frac{K}{N}+k\left(1-\frac{2K_a}{N}+\frac{K}{N}\right)^2(1-\omega_2)\right]\mathrm{e}^{k\left(1-\frac{2K_a}{N}+\frac{K}{N}\right)(\omega_2-1)}\\ &+\frac{2K_a}{N}-\frac{K}{N}+\left(\frac{2K_a}{N}-\frac{K}{N}\right)\omega_2-\omega_2 \end{aligned} \tag{14}$$

where ω_2 satisfies $1-\omega_2-\mathrm{e}^{-k\left(1-\frac{2K_a}{N}+\frac{K}{N}\right)}e^{-k\left(1-\frac{2K_a}{N}+\frac{K}{N}\right)(1-\omega_2)}=0$.

(ii) SSNs

In SSNs with N nodes and average in-degree and out-degree equal to k, the in-degree distribution resembles a Poisson distribution with mean value k and the out-degree distribution follows a Dirac delta function. As a result, the generating functions of in-degree and out-degree distribution can be denoted as follows,

$$G_{\text{in}}(x)=\mathrm{e}^{-k(-x+1)}, G_{\text{out}}(x)=x^k. \tag{15}$$

Based on equations (7), (10) and (9), the minimum fraction of driver nodes n_D at challenge K under random removals can be calculated by

$$n_D(K)=\frac{K}{N}+\frac{K}{N}\omega_2-\omega_2+\left[1-\frac{K}{N}+(k-1)\left(1-\frac{K}{N}\right)^2(1-\omega_2)\right]\mathrm{e}^{k\left(1-\frac{K}{N}\right)(\omega_2-1)} \tag{16}$$

where ω_2 satisfies $1-\omega_2-[\frac{K}{N}+\left(1-\frac{K}{N}\right)(1-\mathrm{e}^{-k\left(1-\frac{K}{N}\right)(1-\omega_2)})]^{k-1}=0$.
Then the minimum fraction of driver nodes n_D at challenge K under random additions can be obtained by

$$n_D(K)=\left[1-\frac{2K_a}{N}+\frac{K}{N}+(k-1)\left(1-\frac{2K_a}{N}+\frac{K}{N}\right)^2(1-\omega_2)\right]\mathrm{e}^{k\left(1-\frac{2K_a}{N}+\frac{K}{N}\right)(\omega_2-1)}+\frac{2K_a}{N}-\frac{K}{N}+\left(\frac{2K_a}{N}-\frac{K}{N}\right)\omega_2-\omega_2 \tag{17}$$

where ω_2 satisfies $1-\omega_2-[\frac{2K_a}{N}-\frac{K}{N}+(1-\frac{2K_a}{N}+\frac{K}{N})(1-\mathrm{e}^{-k\left(1-\frac{2K_a}{N}+\frac{K}{N}\right)(1-\omega_2)})]^{k-1}=0$.

(iii) SF directed networks
For SF networks, we suppose the in-degree distribution and out-degree distribution both follow the pure power-law distribution with minimum degree a and exponent γ, which can be denoted as follows,

$$P_{\mathrm{in}}(k_{\mathrm{in}})=C_{\mathrm{in}}k_{\mathrm{in}}^{-\gamma},\quad P_{\mathrm{out}}(k_{\mathrm{out}})=C_{\mathrm{out}}k_{\mathrm{out}}^{-\gamma}, \tag{18}$$

where $C_{\mathrm{in}}=\frac{1}{\sum_{k_{in}=a}^{\infty}k_{in}^{-\gamma}}$ and $C_{\mathrm{out}}=\frac{1}{\sum_{k_{\mathrm{out}}=a}^{\infty}k_{\mathrm{out}}^{-\gamma}}$, in short $C_{\mathrm{in}}=C_{\mathrm{out}}=\frac{1}{\zeta(\gamma,a)}$ where $\zeta(\gamma,a)$ is the Hurwitz Zeta function. The average degree satisfies $k=\frac{\zeta(\gamma-1,a)}{\zeta(\gamma,a)}$. Correspondingly, the generating functions can be obtained by

$$G_{\mathrm{in}}(x)=\frac{x^a\Phi(x,\gamma,a)}{\zeta(\gamma,a)},\quad G_{\mathrm{out}}(x)=\frac{x^a\Phi(x,\gamma,a)}{\zeta(\gamma,a)}, \tag{19}$$

where $\Phi(z,s,\alpha)$ is the Lerch transcendent function. Together with equations (7), (10) and (9), the fraction of the minimum fraction of driver nodes n_D at challenge K under random removals can be calculated by

$$\begin{aligned}n_D=&\frac{\left(1-\frac{K}{N}\right)\left(-\frac{K}{N}\omega_2+\frac{K}{N}+\omega_2\right)^a\Phi\left(-\frac{K}{N}\omega_2+\frac{K}{N}+\omega_2,\gamma,a\right)}{\zeta(\gamma,a)}\\&+\frac{\left(1-\frac{K}{N}\right)\Phi\left(\frac{\left(\frac{K}{N}-1\right)\Phi\left(-\frac{K}{N}\omega_2+\frac{K}{N}+\omega_2,\gamma-1,a\right)\left(-\frac{K}{N}\omega_2+\frac{K}{N}+\omega_2\right)^{a-1}}{\zeta(\gamma-1,a)}+1,\gamma,a\right)}{\zeta(\gamma,a)}\\&\times\left(\frac{\left(\frac{K}{N}-1\right)\left(-\frac{K}{N}\omega_2+\frac{K}{N}+\omega_2\right)^{a-1}\Phi\left(-\frac{K}{N}\omega_2+\frac{K}{N}+\omega_2,\gamma-1,a\right)}{\zeta(\gamma-1,a)}+1\right)^a\\&+\frac{k\left(2\frac{K}{N}-1-\left(\frac{K}{N}\right)^2\right)(\omega_2-1)\left(-\frac{K}{N}\omega_2+\frac{K}{N}+\omega_2\right)^{a-1}\Phi\left(-\frac{K}{N}\omega_2+\frac{K}{N}+\omega_2,\gamma-1,a\right)}{\zeta(\gamma-1,a)}+2\frac{K}{N}-1,\end{aligned} \tag{20}$$

where $1-\omega_2-\bar{H}_{out}(1-\bar{H}_{in}(\omega_2))=0$.
Then the fraction of the minimum fraction of driver nodes n_D at challenge K under random additions can be acquired by

$$\begin{aligned}n_D=&\frac{\left(1-\frac{2K_a-K}{N}\right)\left(\frac{2K_a-K}{N}(1-\omega_2)+\omega_2\right)^a\Phi\left(\frac{2K_a-K}{N}(1-\omega_2)+\omega_2,\gamma,a\right)}{\zeta(\gamma,a)}\\&+\frac{\left(1-\frac{2K_a-K}{N}\right)\Phi\left(\frac{\left(\frac{2K_a-K}{N}-1\right)\Phi\left(\frac{2K_a-K}{N}(1-\omega_2)+\omega_2,\gamma-1,a\right)\left(\frac{2K_a-K}{N}(1-\omega_2)+\omega_2\right)^{a-1}}{\zeta(\gamma-1,a)}+1,\gamma,a\right)}{\zeta(\gamma,a)}\\&\times\left(\frac{\left(\frac{2K_a-K}{N}-1\right)\left(\frac{2K_a-K}{N}(1-\omega_2)+\omega_2\right)^{a-1}\Phi\left(\frac{2K_a-K}{N}(1-\omega_2)+\omega_2,\gamma-1,a\right)}{\zeta(\gamma-1,a)}+1\right)^a\\&+\frac{\left(\frac{2K_a-K}{N}(1-\omega_2)+\omega_2\right)^{a-1}\Phi\left(\frac{2K_a-K}{N}(1-\omega_2)+\omega_2,\gamma-1,a\right)}{\zeta(\gamma-1,a)}\\&\times\frac{k\left(\frac{4K_a-2K}{N}-1-\left(\frac{2K_a-K}{N}\right)^2\right)(\omega_2-1)}{\zeta(\gamma-1,a)}+2\frac{2K_a-K}{N}-1,\end{aligned} \tag{21}$$

where $1-\omega_2-\bar{H}_{\mathrm{out}}(1-\bar{H}_{\mathrm{in}}(\omega_2))=0$.

3.4. Recoverability indicators

To facilitate the comparison of various recovery strategies, recoverability indicators are necessary. One such indicator is the recovery energy E, as proposed by Sun *et al* [24]. We adopt the recovery energy as a measure of recoverability. The calculation of recovery energy is outlined as follows:

$$E = \sum_{K=K_a}^{2K_a} n_D(K), \tag{22}$$

where K_a is the number of challenges occurring during the attack process.

Taking inspiration from the robustness metric suggested in [21, 43] to assess the effectiveness of attack strategies, our study introduces a recoverability metric denoted by R to quantify the recoverability. This robustness metric quantifies the impact of an attack strategy by averaging the network controllability at each step during an attack [21]. Similarly, we compute the recoverability metric R for a recovery strategy by averaging the network controllability at each step during the recovery process. This allows us to quantify the performance of different recovery strategies with respect to network controllability:

$$R = \frac{1}{K_a} \sum_{K=K_a}^{2K_a} n_D(K) = \frac{E}{K_a}. \tag{23}$$

Since the value of K_a remains constant for different recovery strategies for a given network, the ranking of recovery strategies remains consistent across both recoverability indicators. The physical significance of the recovery energy lies in its representation of the total minimum number of required driver nodes throughout the recovery process. A higher recovery energy signifies a greater demand for driver nodes during recovery, whereas a lower recovery energy implies that network controllability can be regained with fewer driver nodes. In essence, recovery strategies with lower recovery energy E or a reduced unique recoverability measure R are deemed more efficient in restoring network controllability.

4. Results

4.1. Validations of the analytical method

To validate the proposed analytical method for random removals and additions, we conducted simulations on both synthetic and real-world networks to determine the minimum fraction of required driver nodes. Each simulation realization involved a sequence of attacking and recovering the network, with this process repeated 10 000 times. During the attack phase, a single node was randomly removed at each step, and the recalculated minimum fraction of necessary driver nodes for network controllability was recorded. Subsequently, in the recovery phase, we reintroduced one removed node along with its original connections at each step and recalculated the minimum fraction of driver nodes. The recovery process concluded upon the restoration of all initially removed nodes. For synthetic networks characterized by specific parameters, a new network was generated for each simulation realization. However, for real-world networks, the same network was employed across all realizations.

In figure 2, we present the simulation results and analytical predictions of the proposed method on synthetic networks. The results of random node removal and addition are displayed in blue and red, respectively. The analytical approximations are represented by dashed lines, and the algorithm results are presented with solid lines. Our analysis shows that the analytical approximations accurately predict the minimum fraction of driver nodes in most synthetic networks, except for SF networks with $N = 10\,000$, $\gamma = 2.3$, and $a = 3$, where a gap is observed between the dashed lines and solid lines.

We also observe the discrepancies between simulation and analytical results for three real-world networks, which is depicted in figure 3 where the solid lines represent the simulation results and the dashed lines represent the analytical results. To further explore the reasons for these discrepancies, we conduct two experiments on each of the three networks.

In the first experiment, we conduct the degree preserving rewiring strategy that we maintain the original degree distribution of each network and randomly rewire two links in the graph. We then recalculate the minimum fraction of driver nodes and repeat this process for 10 000 iterations. We record the minimum fraction of driver nodes for each iteration and present the results in the form of a box plot. In the second experiment, we generate 10 000 graphs using the degree distributions obtained from the real-world networks by employing the configuration model [29]. We then calculate the minimum fraction of driver nodes for each generated graph and represent the results as a box plot.

As depicted in figure 4, the results of both experiments reveal that the minimum fraction of driver nodes can vary for the networks with the same in- and out-degree distributions. Additionally, we observe that the

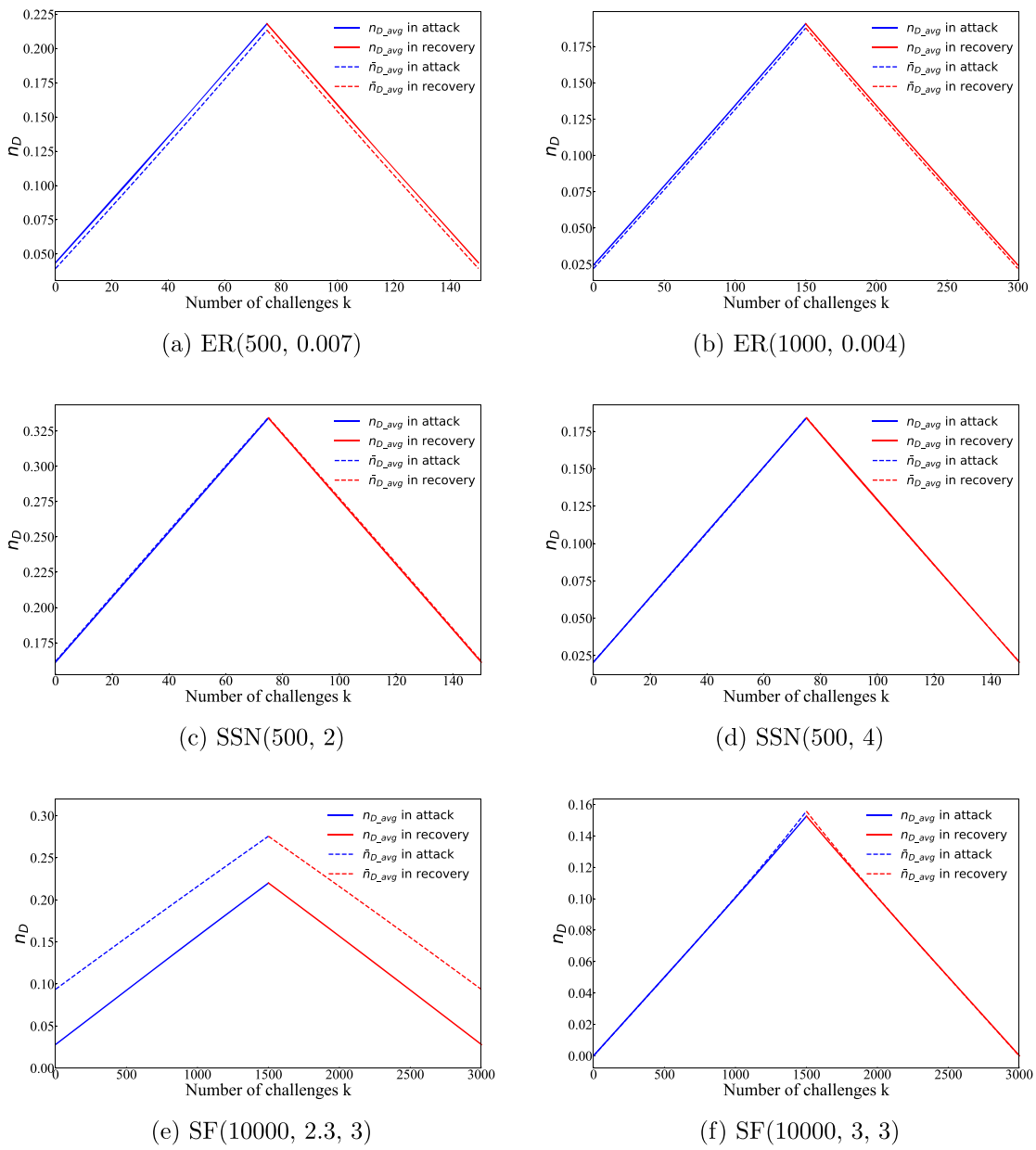


Figure 2. The minimum fraction of driver nodes n_D during random node removals and random node additions in synthetic networks. The blue lines depict node removals, while the red lines represent node additions. The solid lines are obtained by simulations. The blue and red dashed lines are the analytical approximations under random node removals and additions, respectively. We use $n_{D\text{avg}}$ to denote the mean minimum fraction of driver nodes in the simulations at each challenge and use $\bar{n}_{D\text{avg}}$ to denote the analytical values of the minimum fraction of driver nodes at each challenge.

mean value of the minimum number of driver nodes in the networks generated by the configuration model is equivalent to that obtained using the analytical approximation method. It should be noted that the analytical approximation method represents the expected value of the minimum number of driver nodes for graphs that have the same in-degree and out-degree distributions. Conversely, a real-world network is merely a single instance of networks that satisfy the specific in-degree and out-degree distributions. This fundamental difference between the analytical and real-world networks accounts for the gaps between the simulation and analytical results.

4.2. Analytical method with shifting

In order to reduce the discrepancies between the predicted and simulated values, we propose to adjust our original analytical model by applying a shift. First, we determine the exact value of the minimum fraction of driver nodes $n_D[0]'$ by applying the maximum matching algorithm. The shifting term β is then calculated as the difference between $n_D[0]'$ and the original analytical approximation $n_D[0]$, i.e. $\beta = n_D[0]' - n_D[0]$. Consequently, if the shifted analytical result of the minimum fraction of driver nodes at a particular

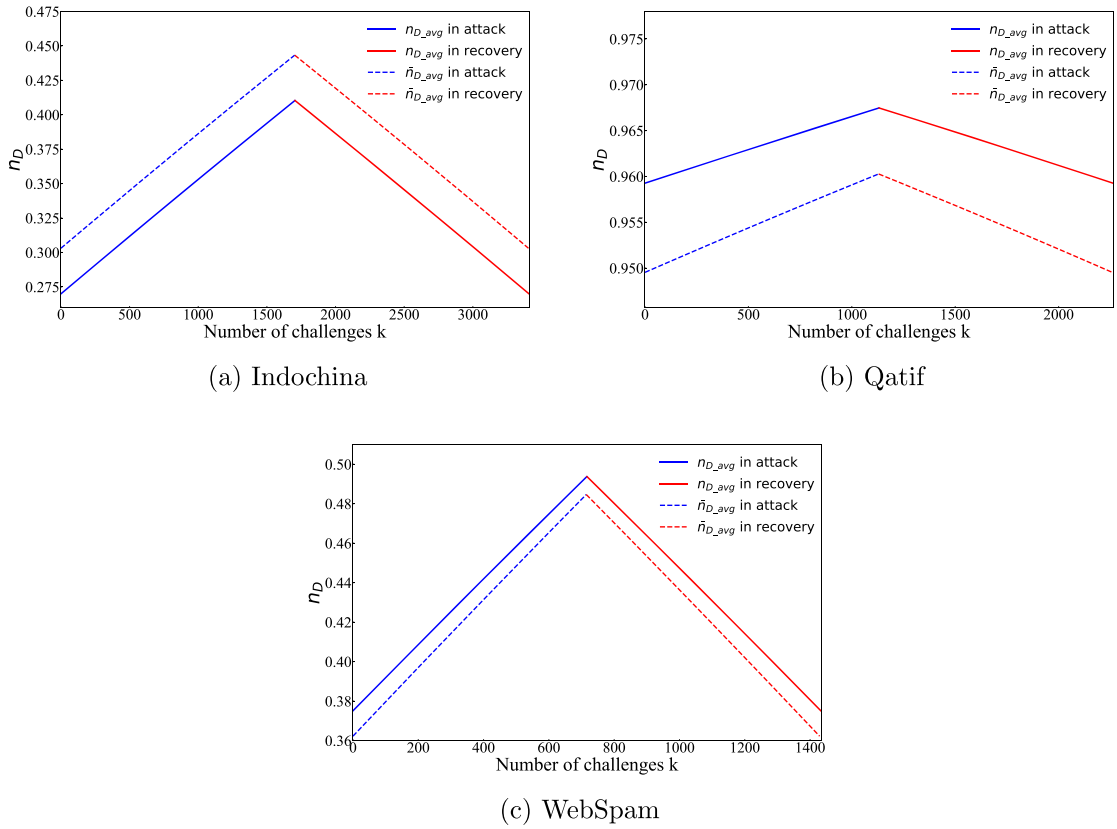


Figure 3. The minimum fraction of driver nodes n_D during random node removals and random node additions in three real-world networks. The blue lines depict node removals while the red lines represent node additions. Both are obtained by the maximum matching algorithm over 10 000 realizations. The blue dashed lines represent the analytical approximations under node removals and the red dashed lines represent the analytical approximations under node additions. $n_{D\text{avg}}$ presents the mean minimum fraction of driver nodes using the algorithm at each challenge and $\bar{n}_{D\text{avg}}$ presents the analytical values of the minimum fraction of driver nodes at each challenge.

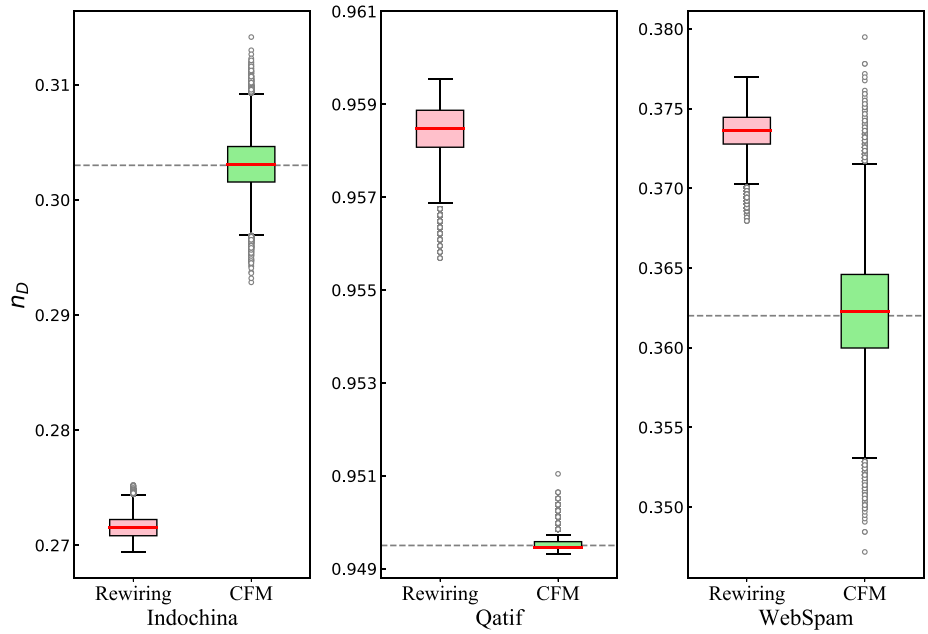


Figure 4. Rewiring links and generating graphs using the configuration model can yield different values for the minimum fraction of driver nodes. 'Rewiring' presents rewiring results, and 'CFM' represents results for the configuration model. The analytical results obtained using the in-degree and out-degree distributions are represented by the grey dashed lines. The mean minimum fraction of driver nodes calculated by the algorithm for networks after rewiring and for networks generated by using the configuration model are indicated by the red lines.

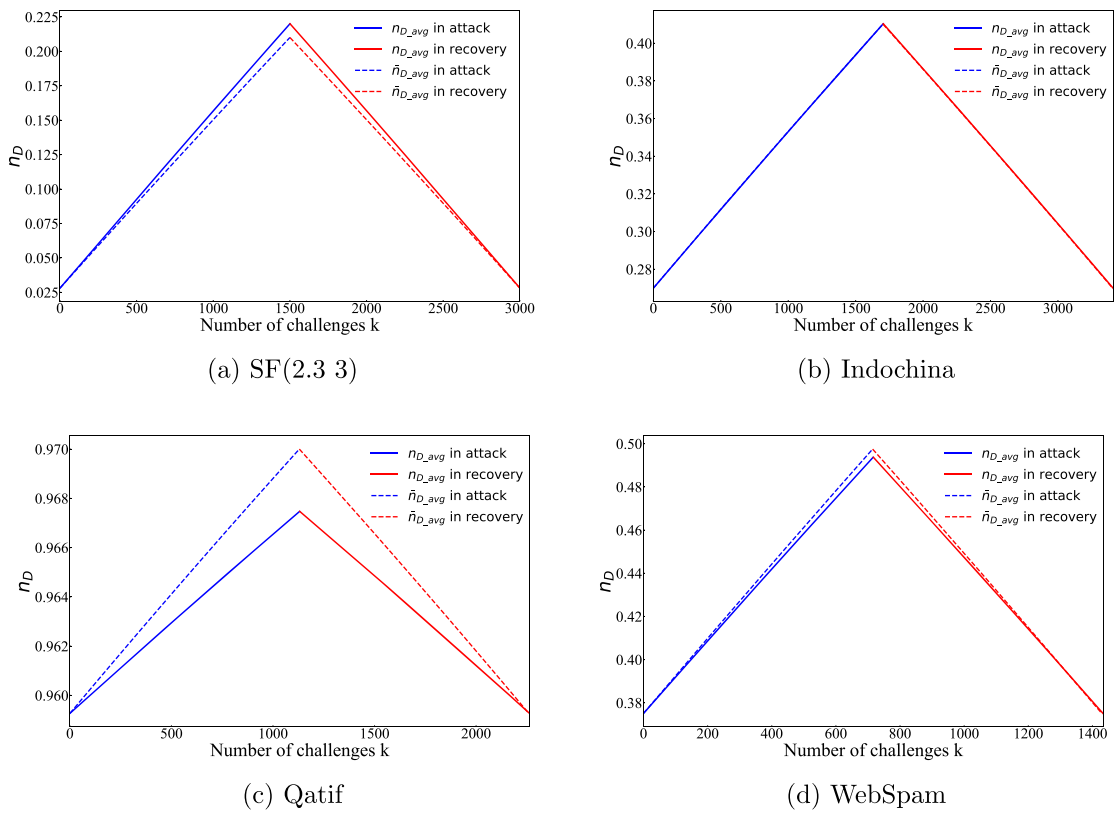


Figure 5. The minimum fraction of driver nodes n_D during random node removals and random node additions in three real-world networks and one SF network after shifting. The blue lines depict node removals, while the red lines represent node additions. Simulation results (solid lines) are obtained by the maximum matching algorithm over 10 000 realizations. The blue dashed lines are the shifted analytical approximations under node removals, while the red dashed lines are the shifted analytical approximations under node additions. $n_{D\text{avg}}$ denotes the mean minimum fraction of driver nodes in the simulations at each challenge and $\bar{n}_{D\text{avg}}$ denotes the shifted analytical values of the minimum fraction of driver nodes at each challenge.

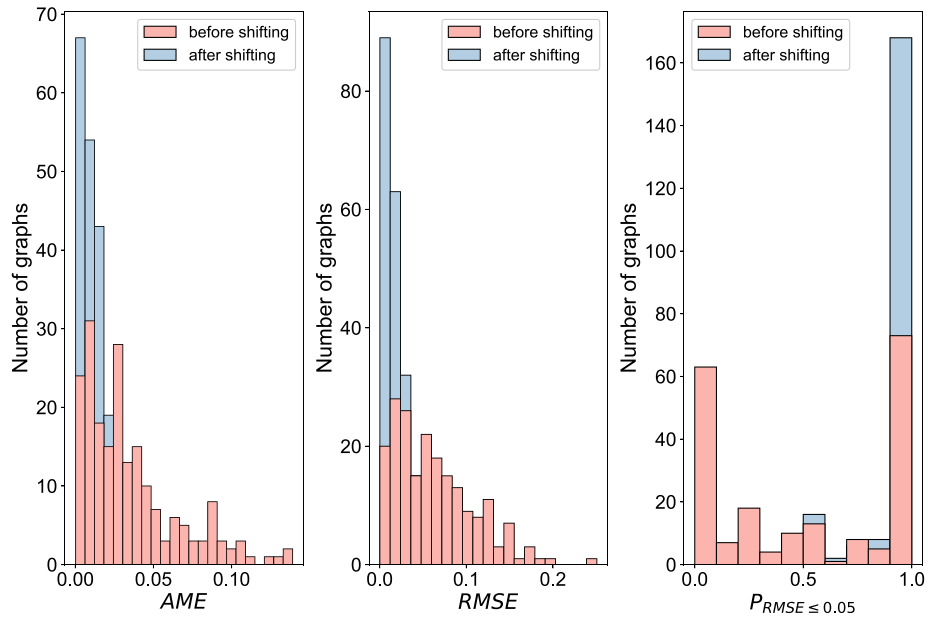


Figure 6. Results before and after shifting for the Topology Zoo data set. Clearly the shifted model exhibits better performance for this data set.

Table 2. Results for the shifted model for seven large scale real-world networks.

Name	AME	RMSE	$P_{\mathrm{RMSE}\leqslant 0.05}$	AME'	RMSE'	$P'_{\mathrm{RMSE}\leqslant 0.05}$
Qatif	0.0085	0.0088	1.0000	0.0015	0.0013	1.0000
p2p Gnutella25	0.0002	0.0003	1.0000	0.0000	0.0000	1.0000
p2p Gnutella08	0.0363	0.0533	0.2326	0.0030	0.0037	1.0000
Indochina	0.0338	0.1005	0.0000	0.0009	0.0019	1.0000
WebSpam	0.0104	0.0240	1.0000	0.0056	0.0106	1.0000
Wiki Vote	0.0001	0.0002	1.0000	0.0001	0.0001	1.0000
Email Eu core	0.0014	0.0096	1.0000	0.0000	0.0002	1.0000

challenge k is denoted as $n_D[k]'$, the shifted result $n_D[k]'$ can be calculated as follows:

$$n_D[k]' = \beta + n_D[k]. \tag{24}$$

It should be noted that the analytical method using shifting will incur an additional computational cost due to the use of the maximum matching algorithm, which has a time complexity of $O\left(L\sqrt{N}\right)$ [41]. Here, L denotes the number of links in the network and N represents the number of nodes in the network.

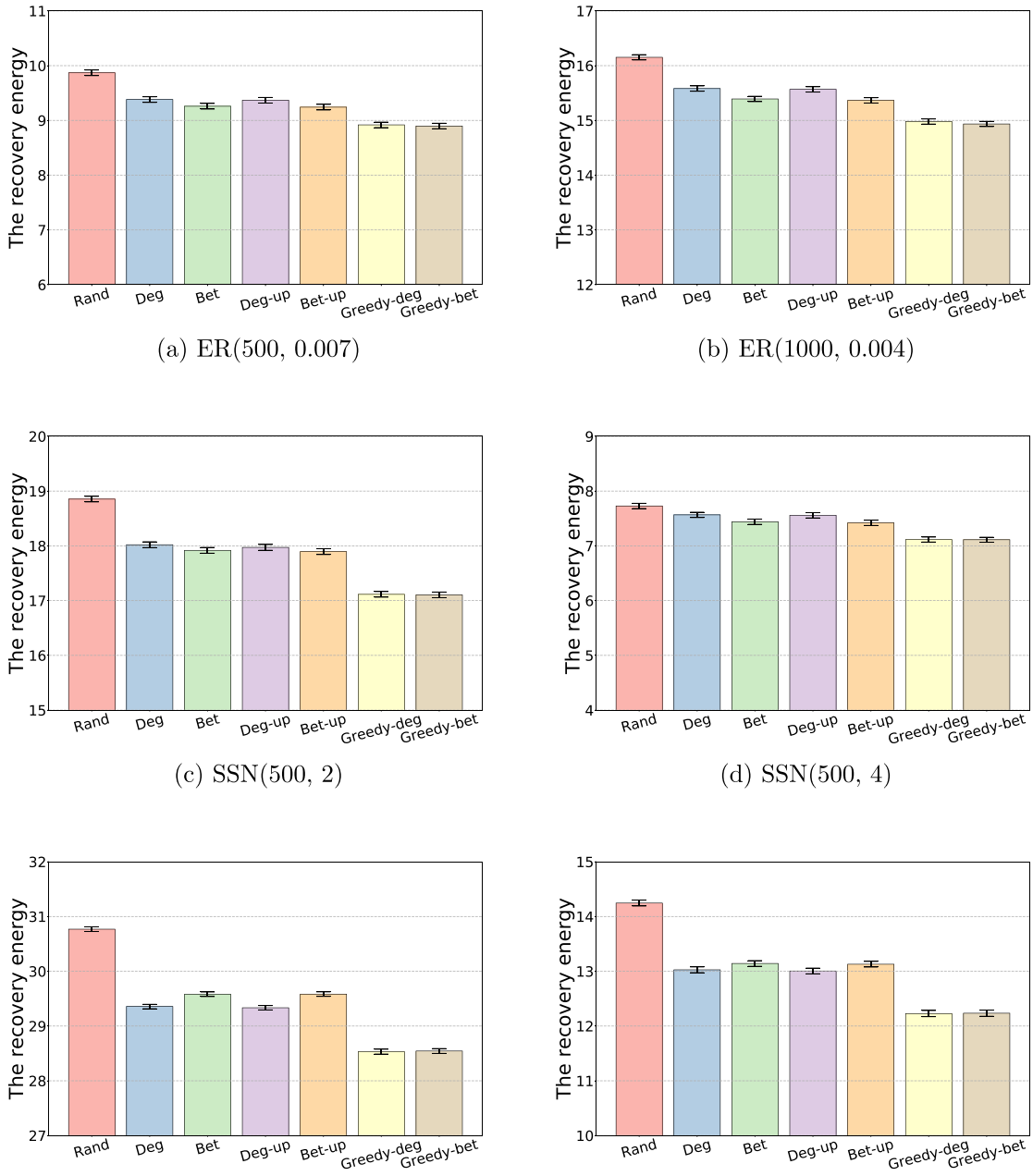
We show the results of the adjusted model for three real-world networks and SF(2.3, 3) in figure 5, where the prediction results are much better than those before shifting. Then we validate the shifting method using the dataset comprising 202 small-scale real-world networks from the Topology Zoo and seven large-scale networks. Validation involves calculating the absolute mean error (AME) and root mean square error (RMSE) between the results obtained from simulations and analytical results, both before and after applying shifting. AME is defined as the absolute difference between the simulation and the analytical results. Additionally, we calculate the proportion of challenges where RMSE was smaller than 5%, denoted as $P_{\mathrm{RMSE}\leqslant 0.05}$. We compare the results before and after shifting to demonstrate the effectiveness of the method.

We present the results of the shifted model for the Topology Zoo dataset in a histogram (figure 6), where the results before and after shifting are depicted in orange and blue, respectively. Furthermore, we observe an improvement in the approximations for the seven large graphs by comparing the results obtained before and after shifting (table 2). The results for the shifted model exhibit smaller AME and RMSE values, and higher $P_{\mathrm{RMSE}\leqslant 0.05}$ values, thus indicating the effectiveness of the shifting method. Besides using the shifting term $\beta = n_D[0]' - n_D[0]$, we can also attempt to use the ratio of $n_D[0]'$ and $n_D[0]$ as a scaling factor, i.e. $\gamma = \frac{n_D[0]'}{n_D[0]}$, to construct a rescaled model $n_D[k]' = \gamma n_D[k]$. However, the results for the rescaled model for the Topology Zoo and seven large-scale networks are less good than those for the shifted model. This discrepancy could be attributed to the rescaled model's heightened sensitivity to scaling factors at each point, as opposed to the fixed modifications offered by the shifted model. The results for the rescaled model are reported in appendix A.

4.3. The efficiency of recovery strategies

We adopt two recoverability indicators to assess the effectiveness of distinct recovery strategies. For each synthetic network category with different sets of parameters, such as the directed ER network with parameters $N = 500$, $p_{\mathrm{ER}} = 0.007$, we generate a total of 10 000 networks. For each network instance, we proceed by randomly removing 15% of the nodes and subsequently employ diverse recovery strategies to restore the network. During the recovery phase, we reintroduce one node at each step in accordance with the chosen recovery method. We then recalibrate the minimum fraction of driver nodes required for network controllability until all previously eliminated nodes are reinstated.

Next, we compute the mean value of the minimum fraction of driver nodes across the 10 000 networks for each specific recovery strategy at every step. Subsequently, we sum these mean values to derive the recovery energy associated with the recovery strategy. For the various recovery strategies employed on synthetic networks, we present the corresponding recovery energy in figure 7. The recoverability metric R is summarized in table 3. In the case of directed ER networks and SSNs, the greedy-betweenness recovery strategy demonstrates the lowest recovery energy, followed by the greedy-degree recovery strategy. The remaining recovery strategies, ranked in order of increasing recovery energy, are updated betweenness recovery strategy, updated degree recovery strategy, betweenness-based recovery strategy, degree-based recovery strategy, and random recovery strategy. Regarding the directed BA network, the degree-related recovery strategies outperform the betweenness-related recovery strategies. The order of recovery energy



(a) ER(500, 0.007) (b) ER(1000, 0.004)

(c) SSN(500, 2) (d) SSN(500, 4)

(e) BA(500, 2) (f) BA(500, 4)

Figure 7. Recovery energy for different recovery strategies in synthetic networks. Bar 'Rand' presents the recovery energy of random recovery strategy; Bar 'Deg' presents the recovery energy of degree-based recovery strategy; Bar 'Bet' presents the recovery energy of betweenness-based recovery strategy; Bar 'Deg-up' presents the recovery energy of updated degree recovery strategy; Bar 'Bet-up' presents the recovery energy of updated betweenness recovery strategy; Bar 'Greedy-deg' presents the recovery energy of greedy-degree recovery strategy and Bar 'Greedy-bet' presents the recovery energy of greedy-betweenness recovery strategy. The numbers above each bar demonstrate different recovery strategies' values of recovery energy.

ranking for different recovery strategies, from lowest to highest, is as follows: greedy-degree recovery strategy, greedy-betweenness recovery strategy, updated degree recovery strategy, updated betweenness recovery strategy, degree-based recovery strategy, betweenness-based recovery strategy, and random recovery strategy. Indeed, it is worth highlighting that the performance improvements brought about by the updated degree (or betweenness) recovery strategy are not substantial when compared to the performance of the corresponding degree-based (or betweenness-based) recovery strategy. On the other hand, the greedy-degree (or greedy-betweenness) recovery strategy significantly enhances performance in comparison to the degree-based (or betweenness-based) recovery strategy. The recovery strategy outcomes for small-sized networks, as presented in appendix B, are consistent with the results discussed here.

Table 3. The recoverability metric R for different recovery strategies for different kinds of synthetic networks. 'Rand' is an abbreviation of random recovery strategy; 'Deg' is an abbreviation of degree based recovery strategy; 'Bet' presents betweenness based recovery strategy; 'Deg-up' is an abbreviation of updated degree based recovery strategy; 'Bet-up' presents updated betweenness recovery strategy; 'Greedy-deg' is an abbreviation of greedy-degree recovery strategy; 'Greedy-bet' presents greedy-betweenness recovery strategy.

Name	Rand	Deg	Bet	Deg-up	Bet-up	Greedy-deg	Greedy-bet
ER(500, 0.007)	0.129 86	0.123 43	0.121 83	0.123 26	0.121 61	0.117 35	0.117 08
ER(1000, 0.004)	0.106 97	0.103 19	0.101 93	0.103 09	0.101 76	0.099 21	0.098 91
SSN(500, 2)	0.248 07	0.237 11	0.235 76	0.236 47	0.235 45	0.225 19	0.225 06
SSN(500, 4)	0.101 69	0.099 55	0.097 90	0.099 44	0.097 63	0.093 61	0.093 57
BA(500, 2)	0.404 85	0.386 27	0.389 21	0.385 97	0.389 25	0.375 42	0.375 58
BA(500, 4)	0.187 44	0.171 41	0.172 93	0.171 13	0.172 75	0.160 89	0.160 98

5. Conclusion and discussion

In this study, we have introduced an analytical approach based on degree distributions to estimate the minimum fraction of driver nodes needed for achieving network controllability through random node additions. We have also employed two recoverability indicators to assess the efficiency of seven recovery strategies after random node removals. These strategies include the random recovery strategy, degree-based recovery strategy, betweenness-based recovery strategy, updated degree recovery strategy, updated betweenness recovery strategy, greedy-degree recovery strategy, and greedy-betweenness recovery strategy.

Upon analysis, we have observed a difference between our initial analytical predictions and simulation results in both synthetic and real-world networks. To address this inconsistency, we propose an adjustment to the original method to align the outcomes more closely. Regarding the seven recovery strategies, we have determined that the greedy-betweenness recovery strategy demonstrates superior efficiency in directed ER networks and SSNs, while the greedy-degree recovery strategy proves most efficient in directed BA networks.

With the investigation into approximating network controllability under random node additions complete, future research endeavors could delve into the development of analytical techniques for estimating network controllability under various recovery strategies. For instance, Wang and Kooij [44] have laid the groundwork for potential analytical methods to approximate network controllability under targeted node additions based on degree. Furthermore, considering the additional computation cost associated with the shifted model, there is potential for enhancing its effectiveness. One promising avenue is the exploration of algorithms with lower complexity to calculate the initial minimum number of driver nodes, thereby optimizing the performance of the shifted model.

Moreover, considering that cycles play a critical role in network controllability [45], we can consider the method proposed by Fan *et al* [46] to measure node centrality based on cycles, which could lead to the development of a cycle ratio recovery strategy, potentially offering improved recovery efficiency. In addition, the concept of the l-shell of a given node, defined as the set of nodes at a distance l from the focal node [42], presents an intriguing avenue for further research. Exploring localized attacks and subsequent recovery strategies based on the shell distance l could offer insights into strategies that leverage localized information. These investigations hold the potential to deepen our understanding of the efficacy of diverse recovery methods and contribute to the evolution of more efficient network recovery techniques in the context of network controllability.

Data availability statement

The data cannot be made publicly available upon publication because no suitable repository exists for hosting data in this field of study. The data that support the findings of this study are available upon reasonable request from the authors.

Acknowledgments

We would like to acknowledge the financial support of the China Scholarship Council (Grant No. 201906040194).

Appendix A. Analytical method based upon rescaling

Besides using the difference between the simulation result $n_D[0]'$ and the analytical approximation $n_D[0]$, we also constructed a model using a rescaling factor γ, defined as $\gamma = \frac{n_D[0]'}{n_D[0]}$. Consequently, if the analytical

result of the minimum fraction of driver nodes at a particular challenge k is denoted as $n_D[k]$, the rescaled result $n_D[k]'$ can be computed as follows:

$$n_D[k]' = \gamma n_D[k]. \tag{A.1}$$

We present the results of the rescaled model for the Topology Zoo dataset in a histogram (figure A1), where the results obtained before and after rescaling are depicted in orange and blue, respectively. We observe a noticeable improvement in the approximations for the seven large real-world graphs by comparing the results obtained before and after rescaling (table A1). The results for the rescaled model exhibit smaller AME and RMSE values, and higher $P_{\mathrm{RMSE}\leqslant 0.05}$ values, thus indicating the effectiveness of the rescaling method. However, compared to the results using the shifted model, the prediction improvements are slightly less.

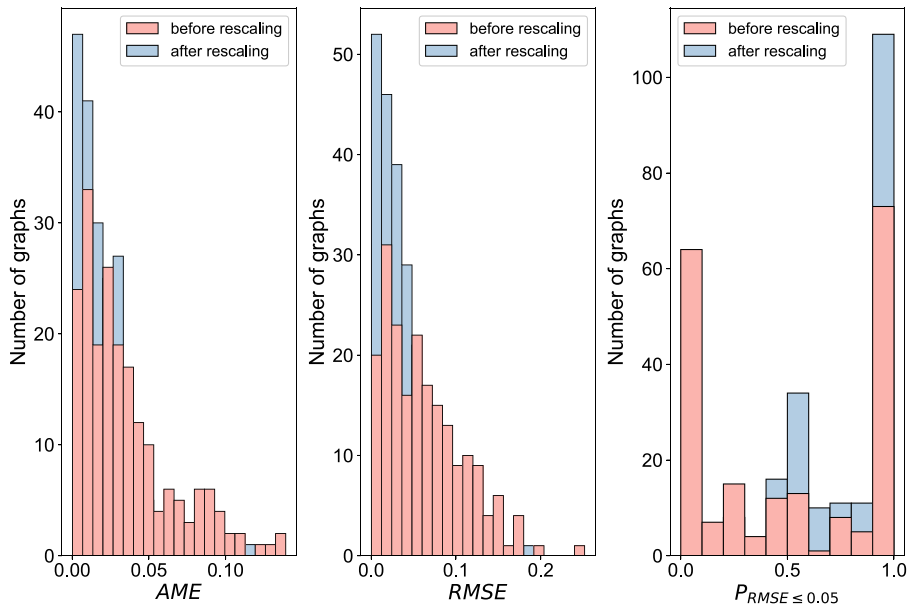


Figure A1. The method based upon rescaling has better approximation performance for the Topology Zoo data set than the original analytical model.

Table A1. The results of the rescaled model for seven large scale real-world networks.

Name	AME	RMSE	$P_{\mathrm{RMSE}\leqslant 0.05}$	AME'	RMSE'	$P'_{\mathrm{RMSE}\leqslant 0.05}$
Qatif	0.0085	0.0088	1.0000	0.0016	0.0014	1.0000
p2p Gnutella25	0.0002	0.0003	1.0000	0.0000	0.0000	1.0000
p2p Gnutella08	0.0363	0.0533	0.2315	0.0053	0.0066	1.0000
Indochina	0.0338	0.1005	0.0000	0.0081	0.0197	1.0000
WebSpam	0.0104	0.0240	1.0000	0.0056	0.0106	1.0000
Wiki Vote	0.0001	0.0002	1.0000	0.0001	0.0001	1.0000
Email Eu core	0.0014	0.0096	1.0000	0.0014	0.0066	1.0000

Appendix B. The performance of different recovery strategies in smallsized synthetic networks

To investigate whether the performance of different recovery strategies in small-sized synthetic networks remains consistent, we calculate the recovery energy and recoverability metric of different recovery strategies in small-sized synthetic networks. The results of the recovery energy are presented in table B1 and the results of the recoverability metric are shown in table B2.

Table B1. The recovery energy E of different recovery strategies for different kinds of synthetic networks. 'Rand' is an abbreviation of random recovery strategy; 'Deg' is an abbreviation of degree based recovery strategy; 'Bet' presents betweenness based recovery strategy; 'Deg-up' is an abbreviation of updated degree based recovery strategy; 'Bet-up' presents updated betweenness recovery strategy; 'Greedy-deg' is an abbreviation of greedy-degree recovery strategy; 'Greedy-bet' presents greedy-betweenness recovery strategy.

Name	Rand	Deg	Bet	Deg-up	Bet-up	Greedy-deg	Greedy-bet
ER(50, 0.07)	1.299 25	1.248 32	1.231 36	1.247 08	1.228 50	1.195 86	1.194 62
ER(100, 0.04)	1.791 39	1.733 17	1.706 36	1.731 72	1.702 56	1.659 79	1.657 14
SSN(50, 2)	2.239 36	2.155 38	2.137 79	2.151 46	2.133 14	2.065 63	2.064 26
SSN(50, 4)	0.983 94	0.974 22	0.962 45	0.973 83	0.960 52	0.949 80	0.949 63
SSN(100, 2)	3.927 23	3.769 35	3.740 61	3.7614	3.734 18	3.598 17	3.595 57
SSN(100, 4)	1.614 84	1.589 19	1.560 30	1.588 03	1.556 44	1.517 31	1.516 74
BA(50, 2)	3.396 99	3.250 94	3.262 15	3.248 86	3.260 49	3.174 85	3.175 40
BA(50, 4)	1.442 35	1.343 77	1.335 73	1.342 72	1.333 53	1.289 32	1.288 62
BA(100, 2)	6.201 76	5.926 66	5.955 79	5.922 76	5.954 77	5.779 48	5.780 79
BA(100, 4)	2.634 04	2.424 88	2.420 39	2.422 08	2.416 71	2.302 61	2.302 16

Table B2. The recoverability metric R of different recovery strategies for different kinds of synthetic networks. 'Rand' is an abbreviation of random recovery strategy; 'Deg' is an abbreviation of degree based recovery strategy; 'Bet' presents betweenness based recovery strategy; 'Deg-up' is an abbreviation of updated degree based recovery strategy; 'Bet-up' presents updated betweenness recovery strategy; 'Greedy-deg' is an abbreviation of greedy-degree recovery strategy; 'Greedy-bet' presents greedy-betweenness recovery strategy.

Name	Rand	Deg	Bet	Deg-up	Bet-up	Greedy-deg	Greedy-bet
ER(50, 0.07)	0.144 36	0.138 70	0.136 82	0.138 56	0.136 50	0.132 87	0.132 74
ER(100, 0.04)	0.111 96	0.108 32	0.106 65	0.108 23	0.106 41	0.103 74	0.103 57
SSN(50, 2)	0.248 82	0.239 49	0.237 53	0.239 05	0.237 02	0.229 51	0.229 36
SSN(50, 4)	0.109 33	0.108 25	0.106 94	0.108 20	0.106 72	0.105 53	0.105 51
SSN(100, 2)	0.245 45	0.235 58	0.233 79	0.235 09	0.233 39	0.224 89	0.224 72
SSN(100, 4)	0.100 93	0.099 32	0.097 52	0.099 25	0.097 28	0.094 83	0.094 80
BA(50, 2)	0.377 44	0.361 22	0.362 46	0.360 98	0.362 28	0.352 76	0.352 82
BA(50, 4)	0.160 26	0.149 31	0.148 41	0.149 19	0.148 17	0.143 26	0.143 18
BA(100, 2)	0.387 61	0.370 42	0.372 24	0.370 17	0.372 17	0.361 22	0.361 30
BA(100, 4)	0.164 63	0.151 56	0.151 27	0.151 38	0.151 04	0.143 91	0.143 89

References

[1] D'Souza R M, di Bernardo M and Liu Y-Y 2023 Controlling complex networks with complex nodes *Nat. Rev. Phys.* **5** 250–62
[2] Cuadra L, Salcedo-Sanz S, Del Ser J, Jiménez-Fernández S and Woo Geem Z 2015 A critical review of robustness in power grids using complex networks concepts *Energies* **8** 9211–65
[3] Lin J and Ban Y 2013 Complex network topology of transportation systems *Transp. Rev.* **33** 658–85
[4] Zhang D, Shi P, Wang Q-G and Yu Li 2017 Analysis and synthesis of networked control systems: a survey of recent advances and challenges *ISA Trans.* **66** 376–92
[5] Chen G 2017 Pinning control and controllability of complex dynamical networks *Int. J. Autom. Comput.* **14** 1–9
[6] Sun P, Kooij R E and Van Mieghem P 2021 Reachability-based robustness of controllability in sparse communication networks *IEEE Trans. Netw. Serv. Manage.* **18** 2764–75
[7] Sun P, Kooij R E, He Z and Van Mieghem P 2019 Quantifying the robustness of network controllability *2019 4th Int. Conf. on System Reliability and Safety (ICSRS)* (IEEE) pp 66–76
[8] Shang Y 2013 Consensus recovery from intentional attacks in directed nonlinear multi-agent systems *Int. J. Nonlinear Sci. Numer. Simul.* **14** 355–61
[9] Kalman R E 1960 On the general theory of control systems *Proc. 1st Int. Conf. on Automatic Control (Moscow, USSR)* pp 481–92
[10] Lin C-T 1974 Structural controllability *IEEE Trans. Autom. Control* **19** 201–8
[11] Liu Y-Y, Slotine J-J and Barabási A-L 2011 Controllability of complex networks *Nature* **473** 167–73
[12] Arenas A, Díaz-Guilera A, Kurths J, Moreno Y and Zhou C 2008 Synchronization in complex networks *Phys. Rep.* **469** 93–153
[13] Amirkhani A and Barshooi A H 2022 Consensus in multi-agent systems: a review *Artif. Intell. Rev.* **55** 3897–935

[14] Albert R, Jeong H and Barabási A-L 2000 Error and attack tolerance of complex networks *Nature* **406** 378–82
[15] Hosseini S, Barker K and Ramirez-Marquez J E 2016 A review of definitions and measures of system resilience *Reliab. Eng. Syst. Saf.* **145** 47–61
[16] Shang Y 2016 Localized recovery of complex networks against failure *Sci. Rep.* **6** 30521
[17] He Z, Sun P and Van Mieghem P 2019 Topological approach to measure network recoverability *2019 11th Int. Workshop on Resilient Networks Design and Modeling (RNDM)* pp 1–7 (https://doi.org/10.1109/RNDM48015.2019.8949119)
[18] Chen A, Sun P and Kooij R E 2021 The recoverability of network controllability *2021 5th Int. Conf. on System Reliability and Safety (ICSRS)* pp 198–208 (https://doi.org/10.1109/ICSRS53853.2021.9660667)
[19] Pu C-L, Pei W-J and Michaelson A 2012 Robustness analysis of network controllability *Physica* A **391** 4420–5
[20] Wang L, Zhao G, Kong Z and Zhao Y 2020 Controllability and optimization of complex networks based on bridges *Complexity* **2020** 1–10
[21] Lou Y, Wang L and Chen G 2021 A framework of hierarchical attacks to network controllability *Commun. Nonlinear Sci. Numer. Simul.* **98** 105780
[22] Dhiman A, Sun P and Kooij R 2021 Using machine learning to quantify the robustness of network controllability *Machine Learning for Networking* ed Eric Renault, S Boumerdassi and P Mühlethaler (Springer) pp 19–39
[23] Lou Y, He Y, Wang L and Chen G 2022 Predicting network controllability robustness: a convolutional neural network approach *IEEE Trans. Cybern.* **52** 4052–63
[24] Sun P, He Z, Kooij R E and Van Mieghem P 2021 Topological approach to measure the recoverability of optical networks *Opt. Switch. Netw.* **41** 100617
[25] Komareji M and Bouffanais R 2013 Resilience and controllability of dynamic collective behaviors *PLoS One* **8** 1–15
[26] Shang Y and Bouffanais R 2014 Influence of the number of topologically interacting neighbors on swarm dynamics *Sci. Rep.* **4** 4184
[27] Barabási A-L and Albert R 1999 Emergence of scaling in random networks *Science* **286** 509–12
[28] Alstott J, Bullmore E and Plenz D 2014 Powerlaw: a Python package for analysis of heavy-tailed distributions *PLoS One* **9** 1–11
[29] Newman M E J, Strogatz S H and Watts D J 2001 Random graphs with arbitrary degree distributions and their applications *Phys. Rev.* E **64** 026118
[30] Knight S, Nguyen H X, Falkner N, Bowden R and Roughan M 2011 The internet topology zoo *IEEE J. Sel. Areas Commun.* **29** 1765–75
[31] Rossi R A and Ahmed N K 2015 The network data repository with interactive graph analytics and visualization *AAAI* (available at: https://networkrepository.com)
[32] Leskovec J and Krevl A 2014 SNAP datasets: Stanford large network dataset collection (available at: http://snap.stanford.edu/data)
[33] Castillo C, Chellapilla K and Denoyer L 2008 Web spam challenge 2008 *Proc. 4th Int. Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*
[34] Boldi P, Codenotti B, Santini M and Vigna S 2004 UbiCrawler: a scalable fully distributed web crawler *Softw. Pract. Exper.* **34** 711–26
[35] Leskovec J, Huttenlocher D and Kleinberg J 2010 Signed networks in social media *Proc. SIGCHI Conf. on Human Factors in Computing Systems* pp 1361–70
[36] Rossi R A, Gleich D F, Gebremedhin A H and Patwary M A 2014 Fast maximum clique algorithms for large graphs *Proc. 23rd Int. Conf. on World Wide Web (WWW)*
[37] Yin H, Benson A R, Leskovec J and Gleich D F 2017 Local higher-order graph clustering *Proc. 23rd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining* pp 555–64
[38] Leskovec J, Kleinberg J and Faloutsos C 2007 Graph evolution: densification and shrinking diameters *ACM Trans. Knowl. Discovery Data* **1** 2–es
[39] Ripeanu M and Foster I 2002 Mapping the Gnutella network: macroscopic properties of large-scale peer-to-peer systems *Peer-to-Peer Systems: First Int. Workshop (IPTPS 2002) (Cambridge, MA, USA, 7–8 March 2002) (Revised Papers 1)* (Springer) pp 85–93
[40] Liu Y-Y and Barabási A-L 2016 Control principles of complex systems *Rev. Mod. Phys.* **88** 035006
[41] Hopcroft J E and Karp R M 1973 An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs *SIAM J. Comput.* **2** 225–31
[42] Shao J, Buldyrev S V, Braunstein L A, Havlin S and Eugene Stanley H 2009 Structure of shells in complex networks *Phys. Rev.* E **80** 036105
[43] Schneider C M, Moreira A A, Andrade J S, Havlin S and Herrmann H J 2011 Mitigation of malicious attacks on networks *Proc. Natl Acad. Sci.* **108** 3838–41
[44] Wang F and Kooij R 2023 Robustness of network controllability with respect to node removals *Complex Networks and Their Applications XI* ed H Cherifi, R N Mantegna, L M Rocha, C Cherifi and S Micciche (Springer) pp 383–94
[45] Jiang S, Zhou J, Small M, Lu J-A and Zhang Y 2023 Searching for key cycles in a complex network *Phys. Rev. Lett.* **130** 187402
[46] Fan T, Lü L, Shi D and Zhou T 2021 Characterizing cycle structure in complex networks *Commun. Phys.* **4** 272