The Recoverability of Network Controllability

Chen, Anqi ; Sun, Peng; Kooij, Robert E.

DOI
10.1109/ICSRS53853.2021.9660667

Publication date
2021

Document Version
Final published version

Published in
2021 5th International Conference on System Reliability and Safety (ICSRS)

The Recoverability of Network Controllability

Anqi Chen*, Peng Sun*, Robert E. Kooij*†
*Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Delft, The Netherlands
†TNO, Unit ICT, The Netherlands
A.Chen-2@student.tudelft.nl, {P.Sun-1, R.E.Kooij}@tudelft.nl

2021 5th International Conference on System Reliability and Safety (ICSRS) | 978-1-6654-0049-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICSRS53853.2021.9660667

***Abstract*—Network recoverability refers to the ability of a network to recover to a desired performance level after suffering topological perturbations such as link failures. The minimum number of driver nodes is a typical metric to denote the network controllability. In this paper, we propose closed-form analytic approximations for the minimum number of driver nodes to investigate the recoverability of network controllability under link-based perturbations in two scenarios: 1) only the links which are damaged in the failure process can be recovered and 2) links can be established between any pair of nodes that have no link between them after the failure process. Results show that our approximations fit well with simulation results both in synthetic networks and real-world networks, such as swarm signaling networks and some communication networks.**

Keywords—recoverability, controllability, complex networks, failures

I. Introduction

Real-world networks are often confronted with topological perturbations such as failures or malicious attacks. For instance, in power grids, the breakdown of connections between different substations can be interpreted as random failures due to circuit aging or natural disasters. In transportation networks, betweenness-based targeted attacks can have a significant impact on normal operation [1]. Network robustness is interpreted as the change of network performance in response to perturbations or challenges imposed on the network [2], which has been widely studied. As the indicators of network performance, different metrics are investigated in face of topological perturbations, such as the effective graph resistance [3], the viral conductance [4], the size of the giant component [5], betweenness and eigenvector centrality, etc.

In recent years, as the research on network controllability attracted more attention [6]-[9], the robustness of network controllability has been a hot topic. Controllability is an essential property for the safe and reliable operation of real-life infrastructures. A system is said to be controllable if it can be driven from any initial state to any desired final state by external inputs in finite time [9]. The robustness of the network controllability can be assessed by quantifying the increase in the minimum number N_D of driver nodes, under perturbation of the network topology. Pu *et al.* [10] found that degree-based attacks are more efficient on network structural controllability than random attacks and cascading failures can also do great harm to network controllability. Nie *et al.* [11] found that the vulnerability of controllability under random and intentional attacks behave differently as the removal fraction increases. Lu *et al.* [12] discovered that a betweenness-based strategy is quite efficient to harm the controllability of real-world networks. Thomas *et al.* [13] identified that the potency of a degree-based attack is directly related (on average) to the betweenness centrality of the edges being removed.

Though the work mentioned above focuses on measuring the robustness of network controllability under failures and attacks, the recovery process after failures is not considered and the investigation on the ability of a network to recover from failures is lacking. In a broad sense, network robustness is also related to the ability of a network to return to a desired performance level after failures [14] which is interpreted as network recoverability in [15]. He *et al.* [15] proposed a general topological approach and recoverability indicators to quantify the network recoverability by applying the effective graph resistance and the network efficiency as robustness metrics. Based on the types of the recovery process, two scenarios are considered: 1) only the links which are damaged in the failure process can be recovered and 2) links can be established between any pair of nodes that have no link between them after the failure process. In this paper, we inherit the general topological approach but use network controllability as the robustness metric. Furthermore, we propose closed-form analytic approximations for network controllability denoted by the minimum number of driver nodes to investigate the recoverability of network controllability.

This paper is organized as follows. In Section II, we introduce some basic concepts and definitions in network controllability proposed in [6] and illustrate the analytical approach to estimate the minimum number of driver nodes by generating functions. In Section III, we investigate the impact of topological perturbations on network controllability. In Section IV, we propose analytic approximations for the minimum number of driver nodes N_D in two scenarios and compare the performance of our approximations with simulation results. In Section V, we compare the efficiency of different recovery strategies in recovering network controllability. Section VI concludes the paper.

II. Network Controllability

A. Structural Controllability Theory

A system is controllable if it can be driven from any initial state to any desired final state by proper variable inputs in finite time [16]. Though most processes on real-world networks are non-linear, the controllability of nonlinear systems is in many

aspects structurally similar to that of linear systems [6]. We consider the linear time-invariant (LTI) dynamics of a directed network with N nodes, which is described as:

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t) \tag{1}$$

where the vector $x(t) = (x_1(t), x_2(t), \ldots, x_N(t))^T$ is the state of N nodes at time t; the $N \times N$ matrix A describes the network topology and the interaction strength between nodes. We assume that A has no self-loops, i.e. all entries on the diagonal of A are zero. The $N \times M (M \leq N)$ matrix B is the input matrix which identifies the interaction between the internal nodes and external control. The vector $u(t) = (u_1(t), u_2(t), \ldots, u_M(t))^T$ expresses the signals that are imposed on the M internal nodes each of which is controlled by an external control. The M internal nodes are referred to as driver nodes.

A LTI system is controllable if the matrix

$$C = (B, AB, A^2B, \ldots, A^{N-1}B) \tag{2}$$

has full rank, i.e., $rank(C) = N$. This criterion is called Kalman's controllability rank condition [17].

For a complex network system for which the matrix A is given, one needs to find a suitable input matrix B such that the system satisfies Kalman's controllability condition which makes the network controllable. Liu *et al.* [6] introduced a feasible method to find the minimum number of driver nodes to control the network considering: (1) the specific weights between the nodes of the networks are usually unknown while only the topology of the network is known for real-world networks; (2) all nodes can be driver nodes which are attached to external controls to make the system fully controllable. However, it would be better if fewer nodes in the network are selected to control the whole system.

Liu *et al.* [6] proved that the minimum number of driver nodes needed for structural controllability, where the input signals are injected to control the directed network, can be obtained through the ``maximum matching" of the network. The matching links of a directed graph G is a set of links such that any two links in this set do not share any start or end nodes. A node is matched if it is an end node of a matching link. Otherwise, it is unmatched. Unmatched nodes are selected as driver nodes which are attached to external controls to make the network controllable. The minimum number N_D of driver nodes to fully control a directed network depends on the maximum matching of this network:

$$N_D = max\{N - \left|M^*\right|, 1\} \tag{3}$$

where N is the size of the network and $\left|M^*\right|$ denotes the size of the maximum matching M^* of the directed network.

With the maximum matching, we are able to find the minimum number of driver nodes in our simulations as long as the topology of a network described by the matrix A is known.

B. Analytical Approximations for the Minimum Number of Driver Nodes

The generating function is an important method in combinatorics, which relates a discrete number sequence to a formal power series. Generating functions can also be used in complex networks. In Li's paper [18], the generating function is used to express the probability that all links of a randomly chosen node are in a specific state, which is written as:

$$G(x) = \sum_{k=0}^{\infty} p_k x^k \tag{4}$$

where x is the probability that a link is in a certain state, and p_k is the probability that this node has degree k. Let $x = 1$, then we obtain $G(1) = \sum_{k=0}^{\infty} p_k = 1$. Besides, the average degree k of the network can be expressed as:

$$\langle k \rangle = G'(1) = \sum_{k=0}^{\infty} k p_k. \tag{5}$$

We can also use the excess degree distribution [18] see Eq. 6, to express the probability of a node with degree k being reached through a randomly chosen link:

$$q_k = \frac{p_k k}{\sum_{k=0}^{\infty} p_k k} = \frac{p_k k}{\langle k \rangle}. \tag{6}$$

Therefore, the generating function for the excess degree distribution can be written as:

$$H(x) = \sum_{k=1}^{\infty} q_k x^{k-1} = \frac{G'(x)}{G'(1)}. \tag{7}$$

Liu *et al.* proposed a method to compute the minimum fraction $n_D = N_D / N$ of driver nodes [6]. The authors used approaches from statistical physics to derive the minimum fraction n_D of driver nodes by using generating functions of out-degree and in-degree distributions.

The general expression for the minimum fraction n_D of driver nodes that Liu *et al.* [6] obtained is:

$$\begin{aligned} n_D = \frac{N_D}{N} = \frac{1}{2}\{&G_{in}(\omega_2) + G_{in}(1-\omega_1) - 2 + G_{out}(\hat{\omega}_2) \\ &+ G_{out}(1-\hat{\omega}_1) + k(\hat{\omega}_1(1-\omega_2) + \omega_1(1-\hat{\omega}_2))\}, \end{aligned} \tag{8}$$

where N_D is the minimum number of driver nodes, N is the size of this network, k is the average out-degree, $G_{in}(\cdot)$ and $G_{out}(\cdot)$ are generating functions for the in-degree and the out-

degree distribution, respectively. Besides, ω_1 , ω_2 , $\hat{\omega}_1$, $\hat{\omega}_2$ satisfy:

$$\omega_1 = H_{out}(\hat{\omega}_2), \tag{9}$$

$$\omega_2 = 1 - H_{out}(1-\hat{\omega}_1), \tag{10}$$

$$\hat{\omega}_1 = H_{in}(\omega_2), \tag{11}$$

$$\hat{\omega}_2 = 1 - H_{in}(1-\omega_1), \tag{12}$$

where $H_{in}(\cdot)$ and $H_{out}(\cdot)$ are generating functions for the excess in-degree and the excess out-degree distribution, respectively. Letting $\omega_1 = 1-\omega_2$ and $\hat{\omega}_1 = 1-\hat{\omega}_2$ [19], the set of Eqs. 9-12 reduces to the pair of equations Eq. 9 and Eq. 12. Thus, by applying the out- and in-degree distribution of a network into Eq. 8, the minimum fraction n_D of driver nodes can be calculated.

C. Networks for Case Study

In Section II.A and II.B, we introduced the method to find the minimum number n_D of driver nodes and the analytical approximation for n_D , respectively. In this section, we introduce the networks we used in this paper for case study and compare the analytical approximation with simulation results.

1) Erdős-Rényi networks: Erdős-Rényi Network (ER network) consists of N nodes, and the probability of a link between each pair of nodes is p . The degree distribution of the ER networks has the binomial distribution which approximates the Poisson distribution:

$$p(k) = \binom{N}{k} p^k (1-p)^{N-k} \approx \frac{\langle k \rangle^k e^{-\langle k \rangle}}{k!} \tag{13}$$

where $\langle k \rangle = p(N-1)$ is the average degree.

We use the $G(N,L)$ model to generate a directed ER network, i.e., a graph with N isolated nodes is generated and then L directed links are placed randomly. For $G(N,L)$, the ER networks have N nodes and L links and its average out-degree is $\langle k_{out} \rangle = L/N$.

For ER networks, the generating functions of the degree distributions are:

$$G_{out}(x) = e^{-k(1-x)}, \tag{14}$$

$$G_{in}(x) = e^{-k(1-x)}, \tag{15}$$

$$H_{out}(x) = e^{-k(1-x)}, \tag{16}$$

$$H_{in}(x) = e^{-k(1-x)}. \tag{17}$$

Then we deduce that the expression of the minimum fraction n_D of driver nodes follows:

$$n_D = e^{-k\omega_1} + exp(-ke^{-k\omega_1}) - 1 + k\omega_1 e^{-k\omega_1} \tag{18}$$

where ω_1 satisfies:

$$\omega_1 = exp(-ke^{-k\omega_1}). \tag{19}$$

We generate 1000 ER networks with the size $N = 20000$ but with different out-degree k , ranging from 1 to 8 to compute the fraction of driver nodes. Figure 1 compares the average simulation results with the analytical approximation from Eq. 8. Each simulation result is the average value of n_D over 1000 ER networks with the same out-degree k . As shown in Figure 1, the discrepancy between simulation and analytical values is very tiny, which indicates that Eq. 8 can well estimate the minimum fraction n_D of driver nodes to control the ER networks.

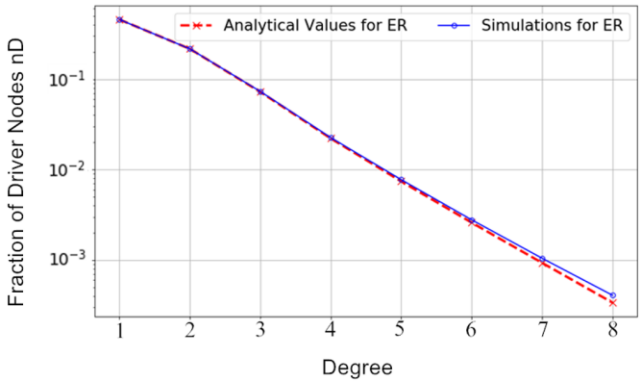


Fig. 1. Performance comparison of the approximation Eq. 8 and simulations for n_D of ER networks with 20000 nodes.

2) Swarm signalling networks (SSNs): Kamareji *et al.* [20] discussed the resilience and controllability of dynamic collective behaviors. They investigated the dynamics of information transfer channels in swarm signaling networks based on a specific topology. A SSN is modeled as a directed network with k -regular out-degree distribution and Poisson in-degree distribution with average k as:

$$p_{in}(k_{in}) = \frac{k^{k_{in}} e^{-k}}{k_{in}!}, \tag{20}$$

$$p_{out}(k_{out}) = \delta(k - k_{out}). \tag{21}$$

The basic generating algorithm of $SSN(N,k)$ is as below:

Step1: Generate a graph with N isolated nodes;

Step2: Iterate each node and randomly add k directed links pointing to k nodes that are randomly chosen.

The generating functions of SSN's degree distributions can be expressed as:

$$G_{out}(x) = x^k, \tag{22}$$

$$G_{in}(x) = e^{-k(1-x)}, \tag{23}$$

$$H_{out}(x) = x^{k-1}, \tag{24}$$

$$H_{in}(x) = e^{-k(1-x)}. \tag{25}$$

We deduce that for SSNs, the minimum fraction n_D of driver nodes follows:

$$n_D = (1-e^{-k(1-\omega_2)})^k - 1 + e^{-k(1-\omega_2)} + k(1-\omega_2)e^{-k(1-\omega_2)} \tag{26}$$

where ω_2 satisfies:

$$1-\omega_2 = (1-e^{-k(1-\omega_2)})^{k-1}. \tag{27}$$

In our simulations, we generate 1000 SSNs with the same number of nodes 20000 but with different out-degree k, ranging from 1 to 8, to compute the fraction of driver nodes by applying the maximum matching algorithm. The performance comparison of the average results from simulations and the analytical approximation is shown in Figure 2. Each simulation result is the average value of n_D over 1000 SSNs with the same out-degree k. As shown in Figure 2, the approximation fits very well with simulation results, which means Eq. 8 has high accuracy in estimating the minimum fraction n_D of driver nodes for SSNs.

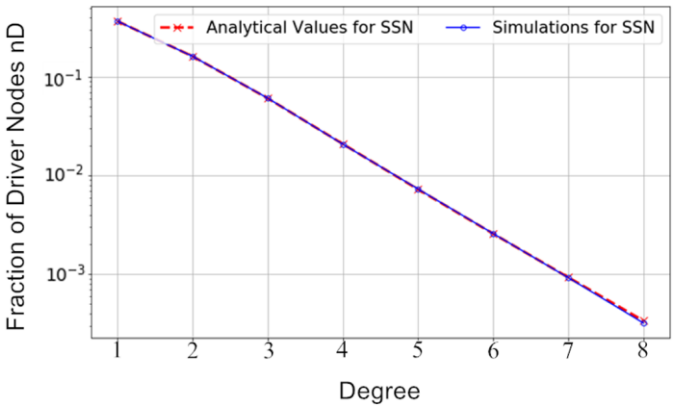


Fig. 2. Performance comparison of the approximation Eq. 8 and simulations for n_D of SSNs with 20000 nodes.

3) Real-world networks: We use some communication networks from the Topology Zoo [21] and the Network Repository [22] for the case study. The properties of the 4 real-world networks are illustrated in Table I. where $\langle k \rangle$ is the average out-degree, which equals the average in-degree.

TABLE I. TOPOLOGICAL PROPERTIES OF 4 REAL-WORLD NETWORKS

Networks	N	L	$\langle k \rangle$
Cogentco	197	243	1.234
kdl	754	895	1.187
routers	2114	6632	3.137
WHOIS	7500	56900	7.587

For real-world networks, the generating function of the degree distribution satisfies:

$$G(x) = \frac{\mathcal{N}(0) + \mathcal{N}(1)\cdot x + \cdots + \mathcal{N}(n-1)\cdot x^{n-1}}{N} \tag{28}$$

where N is the total number of nodes in the network, $\mathcal{N}(m)$ is the number of nodes whose degree equals m.

III. NETWORK CONTROLLABILITY UNDER PERTURBATIONS

As discussed in Section II.B, Eq. 8 provides us a method to analytically calculate the fraction of driver nodes in a network when the degree distribution is known. However, Eq. 8 is not directly applicable when perturbations occur in the network, such as when a fraction p of links are randomly removed in an attack or a fraction f of links are randomly added to the original network. In this section, we propose methods to calculate the minimum fraction of driver nodes for networks under perturbations.

A. Removal of a Fraction p of the Links at Random

According to [23], the degree distribution after randomly removing a fraction p of links, is given by

$$Pr[D_G = i] = (1-p)^i \sum_{j=i}^{N-1} \binom{j}{i} p^{j-i} Pr[D_{G_0} = j] \tag{29}$$

where $p = i/L$, i is the number of removed links and L is the initial number of links in the network. $Pr[D_{G_0} = j]$ is the degree distribution of the original network.

Given the generating function $G(x)$ for the initial network, the generating function $\bar{G}(x)$ of the degree distribution for the resulting network after removing at random a fraction p of the links, satisfies:

$$\bar{G}(x) = G(p + (1-p)x), \tag{30}$$

See also [23]. It follows from Eq. 30 that the average degree after removal of a fraction p of the links becomes $(1-p)G'(1)$. From Eq. 7, we also obtain $\bar{H}(x) = H(p + (1-p)x)$. By replacing the generating function $G(x)$ in Eq. 8 with the generating function $\bar{G}(x)$, we can calculate the fraction of driver nodes in a network where a fraction p of links is randomly removed.

1) Results for SSNs: After substituting the new generating functions $\bar{G}(x)$ into Eq. 8, the fraction n_D of driver nodes in SSNs after a fraction p of the links are removed satisfies [19]:

$$n_D = (p+(1-p)(1-e^{-k(1-p)(1-\omega_2)}))^k - 1 + e^{-k(1-p)(1-\omega_2)} + k(1-p)(1-\omega_2)e^{-k(1-p)(1-\omega_2)}. \quad (31)$$

Applying Eq. 9, Eq. 12 and using the expressions for $\bar{H}_{in}$ and $\bar{H}_{out}$, we obtain ω_2 by solving:

$$1-\omega_2 = (p+(1-p)(1-e^{-k(1-\omega_2)(1-p)}))^{k-1}. \quad (32)$$

In our simulations, we generate SSNs with the same number of nodes 10000 but with different fixed out-degree k, which ranges from 1 to 8. For a SSN with a specific out-degree, we randomly remove a fraction p of links, where $p=0$, 0.2 or 0.5. The simulation results are the average values of n_D for 1000 different attacked SSNs. In Figure 3, we compare the average simulation values and analytical values.

When there is no link removal, i.e. $p=0$, Eq. 31 gives the minimum fraction n_D of driver nodes in the original network. As the value of p increases from 0.2 to 0.5, the value of n_D increases since more driver nodes are needed to make the network controllable. Figure 3 also illustrates that dense networks are easier to control than sparse networks, which have a smaller average degree. As Figure 3 shows, the simulations fit very well with the approximation Eq. 31.

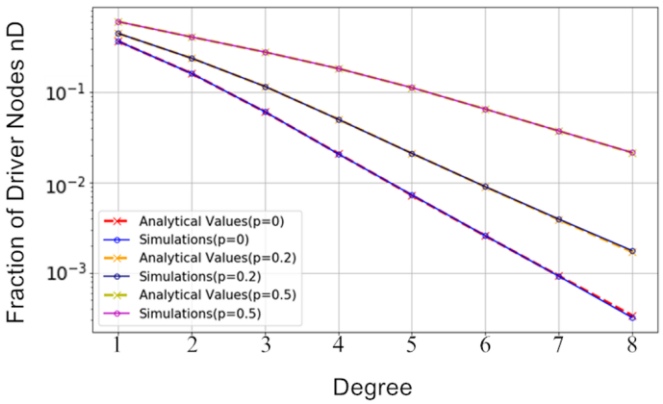


Fig. 3. Fraction of driver nodes for SSN with 10000 nodes as function of k for different values of p under links removal.

2) Results for ER networks: Following the same procedure as before, we obtain the analytical expression of n_D for ER networks as follows:

$$n_D = \exp(-k(1-p)e^{-k(1-p)(1-\omega_2)}) - 1 + e^{-k(1-p)(1-\omega_2)} + k(1-p)(1-\omega_2)e^{-k(1-p)(1-\omega_2)}, \quad (33)$$

where ω_2 satisfies the equation:

$$\omega_2 = 1-\exp(-k(1-p)e^{-k(1-p)(1-\omega_2)}). \quad (34)$$

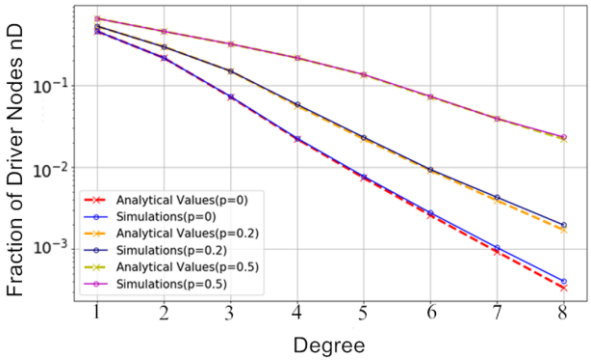


Fig. 4. Fraction of driver nodes for ER networks with 10000 nodes as function of k for different values of p under links removal.

We then compare the analytical results obtained by Eq. 33 with simulation results. Each simulation result is the average values of n_D over 1000 different attacked ER networks. As shown in Figure 4, our approximation for ER networks still fits well with the simulation results.

B. Addition of a Fraction f of the Links at Random

According to [23], the degree distribution for adding a fraction f of links at random, is given by

$$Pr[D_G = i] = (1-f)^{N-i-i}\sum_{j=0}^{i}\binom{N-1-j}{i-j}f^{j-i}Pr[D_{G_0}=j] \quad (35)$$

where $f = K\Big/(\binom{N}{2}-L)$, K is the number of added links, $\binom{N}{2}-L$ is the number of all possible links to add. Given the generating function $G(x)$ for the initial network, the generating function $\hat{G}(x)$ for the resulting network after randomly adding a fraction f of links satisfies:

$$\hat{G}(x) = (1-f(1-x))^{N-1}G(\frac{x}{1-f(1-x)}) \quad (36)$$

See [24]. It follows from Eq. 36 that the average degree after randomly adding a fraction f of links becomes $(1-f)G'(1)+f(N-1)$. By replacing the generating function $G(x)$ in Eq. 8 by the generating function $\hat{G}(x)$, we can calculate the fraction of driver nodes in a network where a fraction f of links is added at random.

1) Results for SSNs: After replacing the original generating function $G(x)$ with the new generating functions $\hat{G}(x)$ into Eq. 8, the fraction n_D of driver nodes in SSNs after a fraction f of the links is randomly added satisfies [24]:

$$n_D = e^{-\bar{k}(1-\omega_2)} + (1-e^{-\bar{k}(1-\omega_2)})^k(1-fe^{-\bar{k}(1-\omega_2)})^{N-1-k} - 1 + \bar{k}(1-\omega_2)e^{-\bar{k}(1-\omega_2)}, \quad (37)$$

where $\bar{k} = k + f(N-1-k)$ and ω_2 satisfies:

$$\bar{k}(1-\omega_2) = (1-e^{-\bar{k}(1-\omega_2)})^{k-1} \cdot (\bar{k} - f(N-1)e^{-\bar{k}(1-\omega_2)}) \cdot (1-fe^{-\bar{k}(1-\omega_2)})^{N-2-k}. \quad (38)$$

In our simulations, we generate SSNs with 10000 nodes but with different fixed out-degree k that ranges from 1 to 8. For 1000 SSN with specific degree, we randomly add a fraction f of links, where $f = 3\times10^{-6}$ and $f = 3\times10^{-4}$. Figure 5 compares the values from simulation and analytical approximations.

The approximations exhibit a very good fit for the simulation when $f = 3\times10^{-6}$. However, when $f = 3\times10^{-4}$, there is a gap between the tail of the analytical approximation and that of the simulation, which means that when $f = 3\times10^{-4}$, the analytical approximations do not fit with the simulations well when the degree is large. The reason is that according to Eq. 3, the number of driver nodes obtained through simulation is at least 1. Therefore n_D is always at least $1/N$ where N is the size of the network. Therefore, for $N = 10000$, we can only validate values of n_D larger than 10^{-4}.

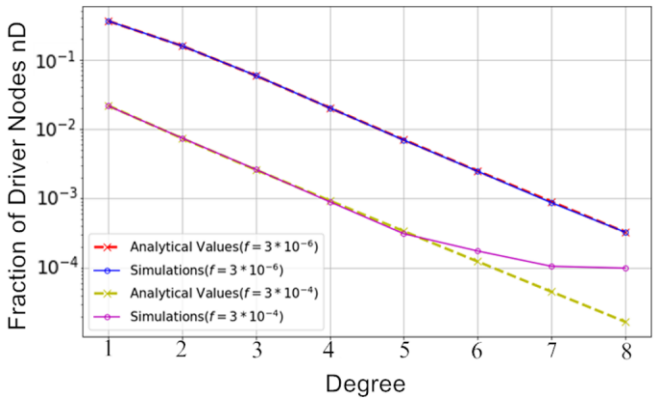


Fig. 5. Fraction of driver nodes for SSNs with 10000 nodes as function of k for adding probability f under links recovery.

2) Results for ER networks: Following the same procedure as before, we obtain the analytical expression of n_D for ER networks:

$$n_D = e^{-\bar{k}\omega_1} + exp(-\bar{k}e^{-\bar{k}\omega_1}) - 1 + \bar{k}\omega_1 e^{-\bar{k}\omega_1} \quad (39)$$

where $\bar{k} = k + f(N-1-k)$ and $\omega_1 = exp(-\bar{k}e^{-\bar{k}\omega_1})$. We also generate ER networks with 10000 nodes but with different out-degree that ranges from 1 to 8. The randomly links addition probability f is set as $f = 3\times10^{-6}$. For a specific average degree k and the probability f, 1000 ER networks are used for simulation. Figure 6 compares the average values of simulations with analytical approximations, and again shows a good fit between them.

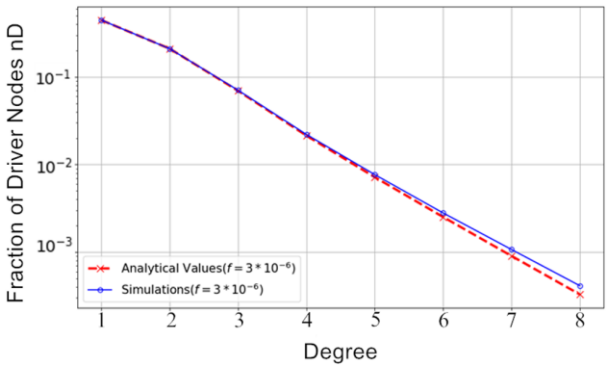


Fig. 6. Fraction of driver nodes for ER networks with 10000 nodes as function of k for adding probability f under links recovery.

IV. Recoverability of Network Controllability

In Section III, we investigated the impact of random link failures and random link additions on the fraction n_D of driver nodes separately. However, it is natural and common in real life to consider recovering a network after failures occur in the network. He *et al.* [15] proposed a general topological approach to quantify the network recoverability which refers to the ability of a network to return to a desired performance level after suffering topological perturbations such as link failures. Based on the types of the recovery process, two scenarios are considered; Scenario A: links can be established between any pair of nodes that have no link between them after the failure process and Scenario B: only the links which are damaged in the failure process can be recovered. In this paper, we inherit the general topological approach but use network controllability as the robustness metric.

A. R-Value

The robustness of a network can be expressed in a mathematical way, through the so-called *R*-value, which quantifies the robustness of a network [2]. In our work, we use the normalized value of n_D as the *R*-value whose value is between 0 and 1. The definition of *R*-value in this paper is:

$$R - value = \frac{1-n_D}{1-n_{D_0}}, \quad (40)$$

where n_{D_0} is the fraction of driver nodes in the original network, n_D is the fraction of driver nodes during the attack phase and recovery phase. When n_D is equal to n_{D_0}, *R* equals 1, which reflects the network's controllability does not change. When the *R*-value equals 0, it means the network controllability is completely destroyed, and all nodes need to be controlled ($n_D = 1$) to control the whole network.

A challenge indicates an event that changes the network topology and thus possibly changes the *R*-value. In this paper, we assume that changes do not happen at the same time. For link-based attack and recovery, an elementary challenge is one

link removal in the attack phase or one link addition in the recovery phase. Each challenge can change the network topology and the *R*-value. As a result, every perturbation in the attack and recovery process has its associated n_D and *R*-value. A sequence of *R*-values can describe any realization with a number M of elementary challenges, denoted by $R[k]_{1\le k\le M}$, where k is the sequence number of challenges.

B. Envelopes

As we discussed above, the impact of any realization of a failure and subsequent recovery process on the network's functionality can be expressed as a sequence of *R*-values $R[k]$, where k is the sequence number of elementary challenges. To investigate the recoverability of networks, we need to know the number of challenges needed to make the original *R*-value (which is normalized to 1) decrease to a predefined *R*-threshold $\rho<1$ in the failure process and also the number of challenges needed to increase the *R*-threshold ρ back to the original *R*-value. This confines us to investigate the number of challenges K as a function of a specific *R*-value r, i.e., $\{K[r]\}$. Thus, each value in $\{K[r]\}$ is the number of challenges that is needed to change the *R*-value to a specific *R*-value r for each realization. Considering that it is impossible to list all values of r between the *R*-threshold ρ and the original *R*-value, we evenly sampled $H=1000$ different r values in the interval $[\rho,1]$. Thus, $r_j=\rho+\frac{(j-1)(1-\rho)}{H-1}$ where j is the j-th value of r. The *envelope* is constructed using all sequences $\{K[r]\}$ for $r\in\{r_1,r_2,...,r_H\}$. The boundaries of the envelope are given by the extreme number of challenges K

$$K_{\min}[r]\in\{\min(K[r_1]),\min(K[r_2]),...,\min(K[r_H])\}, \quad (41)$$

$$K_{\max}[r]\in\{\max(K[r_1]),\max(K[r_2]),...,\max(K[r_H])\}, \quad (42)$$

which gives the best- and worst-case values of the robustness metrics for a network after a given number of recovery challenges. The expected number of challenges K leading to the topological approach r_j is

$$K_{avg}[r]\in\{E(K[r_1]),E(K[r_2]),...,E(K[r_H])\}. \quad (43)$$

Since $K[r]$ defines a probability density function (pdf), we are interested in the percentiles of $K[r]$

$$K_{m\%}[r]\in\{K_{m\%}[r_1],K_{m\%}[r_2],...,K_{m\%}[r_H]\}, \quad (44)$$

where $K_{m\%}[r]$ are the points at which the cumulative distribution of $K[r]$ crosses $m/100$, namely $K_{m\%}[r]=t\Leftrightarrow \Pr[K[r]\le t]=\frac{m}{100}$.

We apply the envelopes to present the behavior of the failure and recovery processes on a network [2], [25]. The envelope profiles the pdf of the random variables of the number of challenges K, which is the probability of a random variable to fall within a particular region. The area of the envelope can be regarded as the variation of the robustness impact of a certain series of challenges, which quantifies the uncertainty or the amount of risk due to perturbations [26].

C. Recovery in Scenario A

In this paper, the *R*-value is the controllability metric of a network $G(N,L)$. Attacking this network would make its minimum fraction n_D of driver nodes increase. Thus, the *R*-value decreases, which denotes the degradation of network controllability. The links are removed one by one until the *R*-value reaches a predefined threshold *R*-threshold. The number of removed links that makes the *R*-value reach the predefined threshold is denoted as K_a. Then the recovery process starts from the remaining network $G_{attacked}(N,L-K_a)$. Scenario A assumes that the recovered links can be added between any two nodes in the complement of the graph after attacks if the elementary challenges are link-based removals and additions.

For link-based random attack and random recovery in Scenario A, the generating function $\bar{G}(x)$ (given in Eq. 30) during the attack process and the generating function $\bar{\bar{G}}(x)$ during the subsequent recovery process [24] can be deduced following the method introduced in Section III:

$$\begin{cases} \text{Attack}: & \bar{G}(x)=G(p+(1-p)x), \\ \text{Recovery}: & \bar{\bar{G}}(x)=(1-f(1-x))^{N-1}\cdot\bar{G}\left(\frac{x}{1-f(1-x)}\right), \end{cases} \quad (45)$$

where $p=m_a/L$, $f=\frac{m_r}{N(N-1)-L+K_a}$, m_a is the number of removed links during the attack process and m_r is the number of recovered links in the recovery process. Recall that K_a denotes the number of removed links at the end of the attack process. The generating function $\bar{\bar{G}}(x)$ in Eq. 45 can be deduced by replacing the generating function $G(x)$ in Eq. 36 with the generating function $\bar{G}(x)$. By applying Eq. 45 to Eq. 8, we can approximate the fraction n_D of driver nodes and the corresponding *R*-values in the random attack and recovery process. When a fraction p of links is randomly attacked, the approximation for the fraction n_D of driver nodes in SSNs follows Eq. 31. When a fraction f of links is randomly recovered in the network after attack, the approximation for the fraction n_D of driver nodes in SSNs follows:

$$\begin{aligned} n_D=&\omega_1(1-\hat{\omega}_2)\cdot(k(1-p^*)+f(N-1-k(1-p^*))) \\ &+G_{in}(1-\omega_1)-1+G_{out}(\hat{\omega}_2), \end{aligned} \quad (46)$$

where $p^*=K_a/L$,

$$G_{out}(x) = (1 - f(1-x))^{N-1} \cdot (p^* + (1-p^*)\frac{x}{1-f(1-x)})^k, \quad (47)$$

$$G_{in}(x) = (1 - f(1-x))^{N-1} \cdot e^{-k(1-p^*)(1-\frac{x}{1-f(1-x)})}, \quad (48)$$

and ω_1 and $\hat{\omega}_2$ follow from Eq. 9 and Eq. 12, after using the appropriate expressions for $H_{out}(x)$ and $H_{in}(x)$, which can be derived by applying Eq. 7 to $G_{out}(x)$ and $G_{in}(x)$ given above.

In our simulation, the *R*-threshold is set to 0.9. We generate 100 SSNs with the size $N = 500$ and the out-degree $k = 2$. Each realization consists of an attack process and the subsequent recovery process.

Based on Eq. 45, the controllability of the attacked network can be analytically expressed during the subsequent recovery process in Scenario A. The top two sub-figures in Figure. 7 exemplify the envelopes of the challenges in SSN for the controllability metric *R*-value in Scenario A, under the random attack and recovery strategy. The approximation fits very well with the simulation, which indicates the general formula Eq. 45 works well. As shown in the bottom two sub-figures of Figure 7, our approximation also fits well with the simulation results in real-world networks. We notice that our analytical approximations for network controllability perform better for *kdl* than *Cogentco*, as the method is based on statistical physics and performs better for large networks.

D. Recovery in Scenario B

The attack process in Scenario B is the same as in Scenario A. In the recovery process in Scenario B, all the links that are removed in the attack process are randomly added until the network returns to the original state under the link-based recovery. A symmetric method is used in Scenario B to express the generating function in the recovery process. By using the same notation as before, $\bar{G}(x)$ [27] and $\bar{\bar{G}}(x)$ refer to the generating functions in the attack process and the subsequent recovery process, respectively.

$$\begin{cases} \text{Attack:} & \bar{G}(x) = G(p + (1-p)x), \\ \text{Recovery:} & \bar{\bar{G}}(x) = \bar{G}(p^* + (1-p^*)x). \end{cases} \quad (49)$$

In the link-based attack process, $p = m / L$ is the fraction of the removed links, and m is the number of removed links. In the link-based recovery process, $p^* = \frac{2K_a - m}{L}$, where K_a is the number of removed links that makes the *R*-value reach the *R*-threshold. After applying Eq. 49 to Eq. 8, we can approximate the fraction n_D of driver nodes and the corresponding *R*-values for Scenario B.

When a fraction p of links is randomly attacked, the approximation for the fraction n_D of driver nodes in SSNs still follows Eq. 31. When the attacked links are randomly recovered, the approximation for the fraction n_D of driver nodes in SSNs follows:

$$n_D = G_{in}(1-\omega_1) - 1 + G_{out}(\hat{\omega}_2) + k(1-p^*) \cdot \omega_1(1-\hat{\omega}_2), \quad (50)$$

where

$$G_{out}(x) = (p^* + (1-p^*)x)^k, \quad (51)$$

$$G_{out}(x) = e^{-k(1-p^*)(1-x)}. \quad (52)$$

In our simulations, we generate 100 SSNs with specific nodes number ($N = 500$) and a specific out-degree ($k_{out} = 2$ or $k_{out} = 4$). Each network is simulated 100 times. We also use two real-world networks for simulations. For a specific real-world network, we simulate 10000 times. Each realization consists of a link-based random attack process and a subsequent link-based random recovery process in Scenario B. Figure 8 illustrates the method predicts the network controllability well during the whole process, not only for SSN, but also for real-world networks.

Comparing the figure for Scenario A and Scenario B, although the attack process is the same, the total number of challenges $K_a + K_r$ in Scenario A is larger than that in Scenario B. It means Scenario B can recover the network's controllability faster than Scenario A because Scenario B assumes it just recovers the attacked links.

V. Recovery Strategies

For simplicity, we only consider the random attack strategy in the attack process and investigate the influence of different recovery strategies on network controllability.

A. Scenario A

In Scenario A, links can be added between any two nodes in the complement of the graph after attacks. Thus, the possible number of steps that is needed to recover the network controllability under the metric-based recovery strategies can be very large. Thus they are not suitable for Scenario A. In the following, three recovery strategies are discussed:

Random Recovery. Random recovery is the easiest way that can be regarded as a self-repairing method after failures or a recovery method without scheduling.

Greedy Recovery. The greedy recovery strategy is adding the link that makes the *R*-value increase the most in each challenge. However, there are many options to add links in each step. Thus, it is a computationally prohibitive task for large networks as the greedy strategy needs to compute all results and pick the best choice.

Connect Recovery. The Connect recovery strategy is extended from [28], which proposed a general approach to optimize the controllability of complex networks by judiciously perturbing the network structure. There are three steps to use the connect recovery strategy:

1. finding the minimum number of independent matching paths;
2. randomly ordering all found matching paths;

3. linking the ending nodes of each matching path to the starting nodes of the matching paths next to it in order.

There are three topologically structural cases [28] of a matching path, shown in Figure 9:

(a) a chain: a path starts from an unmatched node and ends at a matched node without outgoing link belonging to the set of maximum matching;

(b) a directed loop: a path starts from an arbitrary node in a directed loop and ends at the "superior" node that points at the starting node;

(c) isolated node: a node without any link belonging to the set of the maximum matching.

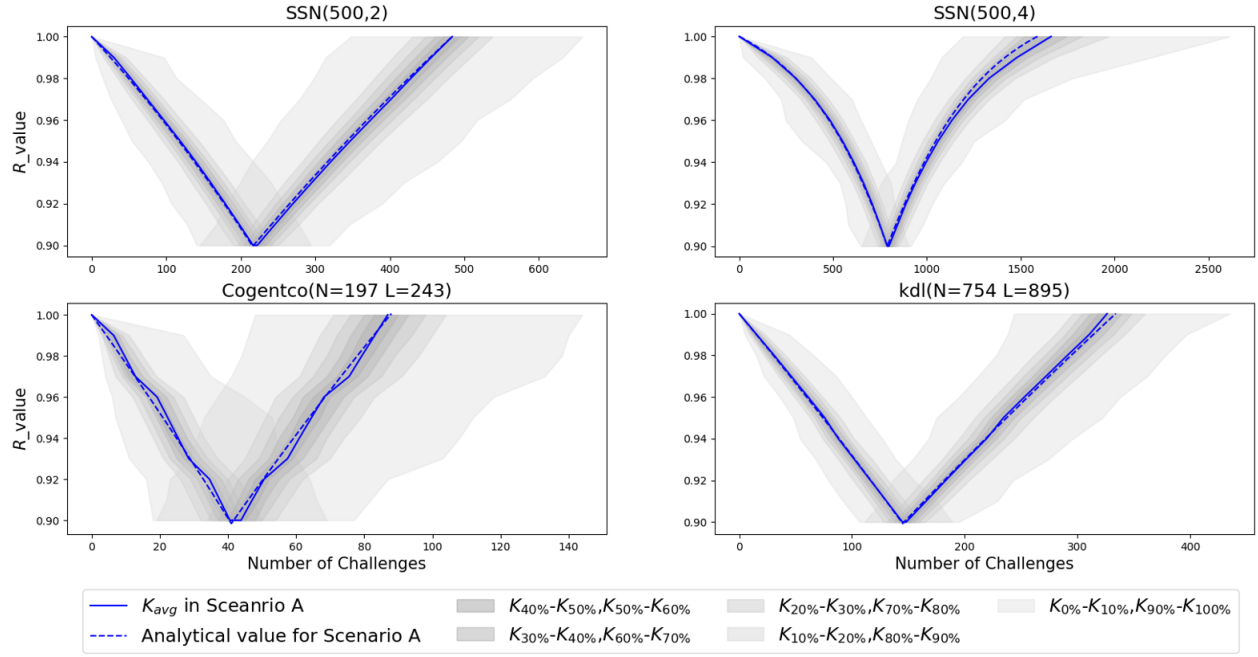


Fig. 7. Envelopes of the challenges for SSNs with 500 nodes and different average out-degree ($k_{out}=2$ and $k_{out}=4$) and two real-world networks (*Cogentco* and *kdl*) in Scenario A, by random attack and random recovery strategy. The threshold of the *R*-value is 0.9. Each envelope is based on 10^4 realizations.

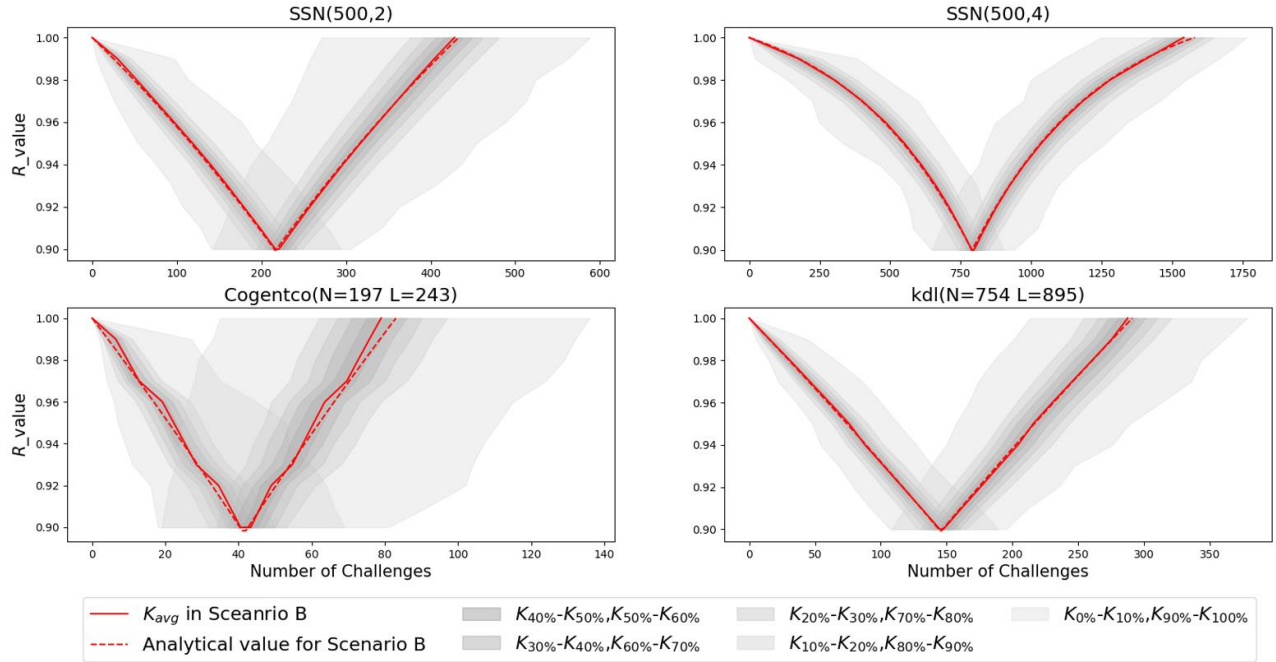


Fig. 8. Envelopes of the challenges for SSNs with 500 nodes and different average out-degree ($k_{out}=2$ and $k_{out}=4$) and two real-world networks (*Cogentco* and *kdl*) in Scenario B, by random attack and random recovery strategy. The threshold of the *R*-value is 0.9. Each envelope is based on 10^4 realizations.

As shown in Figure 10, both the greedy strategy and the connect strategy recover the controllability at the fastest speed. The number N_D of driver nodes becomes one less after every step under the two strategies. And their recovery speed is upper bounded by the random recovery envelopes. However, greedy recovery is a computationally prohibitive task for large networks as it needs to compute all possible outcomes and pick the best choice. The average computation time used for one realization is 8531s. In comparison, connect recovery strategy only costs 0.04s for one realization on average. The reason that the connect strategy just needs a little time to compute is that it only computes once before recovery to find all independent paths. Considering both the steps and time, the connect strategy is optimal for Scenario A. The second recommendation is the greedy strategy if the time is less important than the number of steps and the network is not too large.

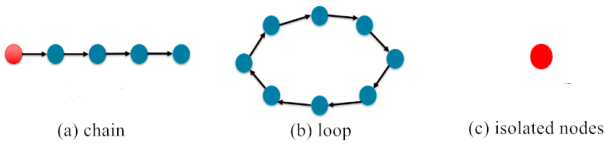
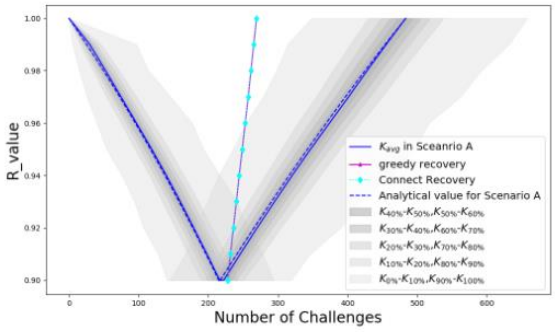


Fig. 9. Three cases of independent path. Unmatched nodes are shown in red and matched nodes are shown in blue.



Strategies	Time used in one realization
Random Recovery	15.2 s
Connect Recovery	0.04 s
Greedy Recovery	8531.9 s

Fig. 10. Comparisons of different recovery strategies for SSN(500,2) in Scenario A.

B. Scenario B

Scenario B assumes that it only recovers the links that are removed during the attack process. Thus, the computational effort is much less than for Scenario A. We also divide the strategies into three categories:

Random Recovery. The random recovery strategy refers to adding the removed links uniformly at random during the recovery process.

Metric-based Recovery. The metric-based strategy determines the sequence of adding links that were attacked, by the topological metrics of links. Four recovery strategies based on metrics of links between node i and node j are considered: the minimum product of degree ($min(d_i d_j)$), the maximum product of degree ($max(d_i d_j)$), the minimum product of eigenvector centrality ($min(c_i c_j)$), and the maximum product of eigenvector centrality ($max(c_i c_j)$). In each challenge step during the recovery process under a specific strategy, a link with the related optimal metric is added.

Greedy Recovery. The greedy recovery strategy is choosing the link to add in each step to increase the *R*-value the most from the links removed during the attack process.

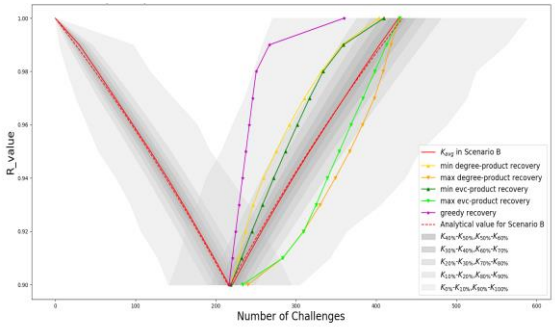


Fig. 11. Comparisons of different recovery strategies for SSN(500,2) in Scenario B.

As shown in Figure 11, the greedy strategy outperforms other strategies as expected. And because links to be added are the removed links, the greedy strategy is scalable for large networks. The strategies which select and restore the link with the minimum degree product or minimum eigenvector centrality product perform better than random recovery. It is worth noting that the *R*-value as function of the number of challenges k under the greedy strategy, minimum-degree product, and minimum-evc product are concave in the recovery process, which demonstrates the returns property of the recovery measures are diminishing. In contrast, the functions under the recovery strategies based on maximum degree-product and maximum evc-product are convex, and the function under random recovery is approximately linear. What is more, the number of steps needed to make the *R*-value return to 1 in random recovery, the maximum degree-product strategy, and the maximum-evc product is the same because Scenario B recovers the links that are removed in the attack phase.

VI. Conclusion

In this study, we derived analytical approximations for the minimum number N_D of driver nodes needed to control networks during link-based random attacks and random additions respectively. Results show that our approximations fit well with simulation results both for random attacks and random link additions. Besides, we inherit the general topological approach but use network controllability as the robustness metric. Furthermore, we propose closed-form analytic approximations for network controllability to investigate the recoverability of network controllability for two scenarios. We validated that our approximations have high accuracy in estimating the fraction of driver nodes in some real-world and swarm signalling networks. We also found that the Connect Recovery strategy is comparable with Greedy

Recovery strategy in recovering controllability of networks but takes much less time for Scenario A.

REFERENCES

[1] B. Berche, C. von Ferber, T. Holovatch, and Y. Holovatch, "Resilience of public transport networks against attacks," *The European Physical Journal B*, vol. 71, no. 1, pp. 125–137, 2009.

[2] P. Van Mieghem, C. Doerr, H. Wang, J. M. Hernandez, D. Hutchison, M. Karaliopoulos, and R. E. Kooij, "A framework for computing topological network robustness," Delft University of Technology, Re- port20101218, pp. 1–15, 2010.

[3] X. Wang, E. Pournaras, R. E. Kooij, and P. Van Mieghem, "Improving robustness of complex networks via the effective graph resistance," *The European Physical Journal B*, vol. 87, no. 9, p. 221, 2014.

[4] A. Socievole, F. De Rango, C. Scoglio, and P. Van Mieghem, "Assessing network robustness under SIS epidemics: The relationship between epidemic threshold and viral conductance," Computer Networks, vol. 103, pp. 196–206, 2016.

[5] X. Wang, Y. Koc¸, S. Derrible, S. N. Ahmad, W. J. Pino, and R. E. Kooij, "Multi-criteria robustness analysis of metro networks," Physica A: Statistical Mechanics and its Applications, vol. 474, pp. 19–31, 2017.

[6] Y.-Y. Liu, J.-J. Slotine, and A.-L. Baraba´si, "Controllability of complex networks," Nature, vol. 473, no. 7346, pp. 167–173, 2011.

[7] Z. Yuan, C. Zhao, Z. Di, W.-X. Wang, and Y.-C. Lai, "Exact controllability of complex networks," Nature Communications, vol. 4, p. 2447, 2013.

[8] T. Jia, Y.-Y. Liu, E. Cso´ka, M. Po´sfai, J.-J. Slotine, and A.-L. Baraba´si, "Emergence of bimodality in controlling complex networks," Nature Communications, vol. 4, p. 2002, 2013.

[9] T. Nepusz and T. Vicsek, "Controlling edge dynamics in complex networks," Nature Physics, vol. 8, no. 7, p. 568, 2012.

[10] C.-L. Pu, W.-J. Pei, and A. Michaelson, "Robustness analysis of network controllability," Physica A: Statistical Mechanics and its Applications, vol. 391, no. 18, pp. 4420–4425, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378437112003135.

[11] S. Nie, X. Wang, H. Zhang, Q. Li, and B. Wang, "Robustness of controllability for networks based on edge-attack," PLOS ONE, vol. 9, no. 2, pp. 1–8, 02 2014. [Online]. Available: https://doi.org/10.1371/journal.pone.0089066

[12] Z.-M. Lu and X.-F. Li, "Attack vulnerability of network controllability," PloS one, vol. 11, no. 9, p. e0162289, 2016.

[13] J. Thomas, S. Ghosh, D. Parek, D. Ruths, and J. Ruths, "Robustness of network controllability to degree-based edge attacks," in International Workshop on Complex Networks and their Applications. Springer, 2016, pp. 525–537.

[14] X. Pan and H. Wang, "Resilience of and recovery strategies for weighted networks," PloS ONE, vol. 13, no. 9, p. e0203894, 2018.

[15] Z. He, P. Sun, and P. Van Mieghem, "Topological approach to measure network recoverability," in 2019 11th International Workshop on Resilient Networks Design and Modeling (RNDM), 2019, pp. 1–7.

[16] A. Lombardi and M. Ho¨rnquist, "Controllability analysis of networks," Physical Review E, vol. 75, no. 5, p. 056110, 2007.

[17] R. E. Kalman, "Mathematical description of linear dynamical systems," Journal of the Society for Industrial and Applied Mathematics, Series A: Control, vol. 1, no. 2, pp. 152–192, 1963.

[18] M. Li and B.-H. Wang, "Generating function technique in complex networks," Journal of Physics: Conference Series, vol. 604, p. 012013, apr 2015. [Online]. Available: https://doi.org/10.1088/1742-6596/604/1/012013

[19] P. Sun, R. E. Kooij, and R. Bouffanais, "Controllability of a class of swarm signalling networks: impact of removing links," in preparation, 2021.

[20] M. Komareji and R. Bouffanais, "Resilience and controllability of dynamic collective behaviors," PloS ONE, vol. 8, no. 12, p. e82578, 2013.

[21] S. Knight, H. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The internet topology zoo," Selected Areas in Communications, IEEE Journal on, vol. 29, no. 9, pp. 1765 –1775, october 2011.

[22] 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

[23] P. Van Mieghem, Performance analysis of complex networks and systems. Cambridge University Press, 2014.

[24] P. Sun and R. E. Kooij, "Controllability of a class of swarm signalling networks: impact of adding links," in preparation, 2021.

[25] S. Trajanovski, J. Martin-Hernandez, W. Winterbach, and P. Van Mieghem, "Robustness envelopes of networks," Journal of Complex Networks, vol. 1, no. 1, pp. 44–62, 2013.

[26] P. Sun, Z. He, R. E. Kooij, and P. Van Mieghem, "Topological approach to measure the recoverability of optical networks," Optical Switching and Networking, vol. 41, p. 100617, 2021.

[27] P. Van Mieghem, Performance analysis of complex networks and systems. Cambridge University Press, 2014.

[28] W.-X. Wang, X. Ni, Y.-C. Lai, and C. Grebogi, "Optimizing controllability of complex networks by minimum structural perturbations," Phys. Rev. E, vol. 85, p. 026115, Feb 2012.