Technische Universiteit Delft

The Recoverability of Network Controllability

Anqi Chen

TU Delft

NAS

THE RECOVERABILITY OF NETWORK CONTROLLABILITY

by

Anqi CHEN

to obtain the degree of Master of Science
in Electrical Engineering
Track Wireless Communication and Sensing
at the Delft University of Technology,
to be defended publicly on Monday August 30, 2021 at 10:00 AM.

Student number:	5020220
Project duration:	November, 2020 – August, 2021
Thesis committee:	Prof.dr.ir. Rob Kooij, TU Delft, supervisor
	Dr. Huijuan Wang, TU Delft
	Peng Sun, TU Delft, daily supervisor

PREFACE

With this thesis project, 'The Recoverability of Network Controllability', I finished the Master of Science degree in Electrical Engineering at the Delft University of Technology. The project has been carried out at the Network Architectures and Services (NAS) group. I would like to express gratitude to my supervisor Professor Rob Kooij for providing me this opportunity to do research on this topic. I am so thankful to him for his guidance and encouragement throughout the entire duration of this project. I would like to thank Peng Sun, my daily supervisor, for his support and suggestions that helped me a lot. It was very pleasant to work with Rob and Peng in the past nine months. I would like to thank Dr. Huijuan Wang for being a part of my thesis committee. Additionally, I would like to thank my friends Xinhan Liu and Yichuang Han for their continued encouragement and accompany in the tough Covid-19 period.

Anqi Chen
Delft, August 2021

ABSTRACT

Network recoverability refers to the ability of a network to return to a desired performance level after suffering malicious attacks or random failures. A system is controllable if it can be driven from any arbitrary state to any desired state in finite time under the control of the driver nodes, which are attached to external inputs. We use the minimum number of driver nodes as the R-value, which is a typical metric to denote the network controllability. We investigate the recoverability of network controllability under link-based perturbations and node-based perturbations. For link-based perturbations, two recovery scenarios are discussed: (1) only the links which are damaged in the failure process can be recovered; (2) links can be established between any pair of nodes that have no link between them after the failure process. For node-based perturbations, we also investigate two recovery scenarios: (1) only the nodes and their original links that are removed in the failure process are recovered; (2) the nodes are removed during the failure process are recovered, and the same number of removed links are added at random. We propose analytical approximations under link-based and node-based perturbations in two recovery scenarios by using generating functions. Results show that our approximations fit well with simulation results both in synthetic networks and some real-world networks, such as swarm signaling networks and communication networks.

CONTENTS

1

INTRODUCTION

Network science [1] has attracted considerable interest and attention, as many complex networks exist in the world, such as the Internet, WWW, electricity networks, transport networks, brain networks, and social networks. The various elements in a complex system are abstracted into nodes in the network, and the connections between nodes are regarded as the functional relationship between system elements. Different types of networks in the real world can be abstracted into complex network models in order to study the commonalities of various networks that seem different and find universal methods for them. As a result, they can provide guidance for the analysis and design of real-world networks.

After several decades of development, the theoretical research of complex networks has achieved many remarkable scientific results and laid a theoretical foundation for further study. The proof of our understanding of natural or technological systems is reflected in our ability to control them [2]. Thus, the research that finds effective ways to control the behavior of the networks has attracted attention.

Many complex system problems in the real world can be abstracted into network controllability problems. For example, we can select several genes as the drug targets to make the whole biological system reach our expected state in the gene regulatory network [3]. And we can select nodes as information source nodes to produce the desired publicity effect for the entire social network [4]. Both the above problems have a common feature that is about selecting driver nodes. By applying some inputs on the driver nodes, we can control the entire network and steer it to a desired state.

Real-world networks are often confronted with topological perturbations such as link-based random failures or node-based random failures. Network robustness is interpreted as a measure of the network's response to perturbations or challenges imposed on the network [5], which has been widely studied. The ability that a network returns to the desired performance level after suffering malicious attacks and random

1

failures is defined as network recoverability [6]. Several recovery mechanisms [7] have been investigated in complex networks applications.

1.1. CHALLENGES

Liu *et al.*[2] proposed an analytical approach to express the controllability of directed networks. Based on this important theoretical framework, Sun *et al.* [8][9] expressed the approximations of the minimum fraction of driver nodes of networks after links removal or links additions. However, his method cannot approximate the minimum fraction of driver nodes during the recovery process after suffering attacks. We will propose a method to analytically express network controllability during a realization that comprises the attack progress and the subsequent recovery process in two scenarios. In addition, we will assess the performance of the analytical approximations by comparing it with simulations on various synthetic and real-world networks.

1.2. OBJECTIVES

Based on the background mentioned above, the objectives of our research are as follows:

1. Validate Sun *et al.*'s formulas [8][9] about the minimum fraction of driver nodes after removing links or adding links;

2. Propose analytical approximations of the network controllability during a realization consisting of the link-based attack process and the subsequent link-based recovery process, and validate them with simulations;

3. Compare different link-based attack strategies and recovery strategies in two separate scenarios.

4. Apply our method of analytical expressing the network controllability to the network under node-based perturbations.

1.3. CONTRIBUTION

The main contributions of this thesis are:

1. We validated Sun *et al.*'s formulas [8][9] about the minimum fraction of driver nodes after removing a fraction p of the links or adding a fraction f of the links. Then, we applied Sun *et al.*'s formula about removing a fraction p of the links to the node-based attack.

2. We proposed the general relations about generating functions of degree distributions after removing links or adding links. Based on this, the minimum fraction of driver nodes can be analytical approximated during the recovery process in two scenarios. We evaluate our approximations on real-world networks and two types of synthetic networks.

3. We compared several attack and recovery strategies for link-based attack and recovery and found the optimal recovery strategies in different scenarios.

4. We also used the general relations about generating functions of degree distributions to express network controllability of the network under node-based attack and recovery.

5. We compared three attack strategies for node-based attack. Also, we analyzed the reason that analytical approximation of minimum fraction of driver nodes cannot work after the degree-based attack and the localized attack.

1.4. THESIS OUTLINE

The structure of this thesis is as follows; Chapter 2 presents the research development of network controllability in detail. In Chapter 3, some basic knowledge of networks and the theoretical framework of network controllability are introduced. Then we propose the analytical approach for approximating the network controllability based on this fundamental theoretical framework. In Chapter 4, we compare the performance of our analytical approximations with simulation for link-based attack and link-based recovery. Besides comparing several link-based attack strategies, we compare two different link-based recovery strategies (referred to as Scenario A and B) and find the optimal strategy for each scenario. Chapter 5 deals with node-based attack and recovery. We compare the performance of analytical approximations and the simulation under node-based attack and recovery in two scenarios. Also, we try to analytically express the network controllability under the degree-based attack and the localized attack. Finally, we present our conclusions and discuss the possible scope for future research in Chapter 6.

2

BACKGROUND

In the recent ten years, the topic of network controllability has become a hot issue in the network science community. There are many critical research issues with extensive scientific significance and application value, such as whether complex networks are controllable, whether it is controllable with self-feedback, how to control networks with self-feedback, how to achieve minimum cost control, etc.

The current research on the controllability of complex networks mainly focuses on the following four aspects:

1. Research on the theoretical analysis framework of the controllability of complex networks;
2. Research on the optimization of network controllability through structural disturbance;
3. Research on the network's attack vulnerability and robustness of controllability;
4. Research on the influence of the main structural characteristics of the network on controllability.

The following sections will introduce the research progress of these four aspects in detail.

2.1. THE THEORETICAL ANALYSIS FRAMEWORK OF NETWORK CONTROLLABILITY

In 2007, Lombardi and Hörnquist firstly combined the control theory and network science [10]. They introduced Kalman's controllability criterion [11] into research and transformed the network controllability problem into calculating the rank of the controllability matrix. They concluded that the property of being downstream of the node to which the input is applied turns out to be a necessary but not a sufficient condition for controllability. Lombardi and Hörnquist's work did not get much attention

for two reasons: for most real-world networks, the weights of links are unknown, so we cannot get the controllability matrix. Even if all weights are known, computing the rank for numerous distinct combinations is a computationally prohibitive task for large networks.

In 2011, Liu *et al.* [2] introduced Lin's structural controllability [12] to bypass the need to measure the link weights. They used the 'maximum matching' to get the minimum number of driver nodes in order to solve the problems mentioned above for large-scale directed networks. They proposed an analytical approach to compute networks' controllability according to statistical physics. From the simulations and analytical results, Liu *et al.* found that the minimum number of driver nodes is determined mainly by the degree distribution, and sparse inhomogeneous networks are more difficult to control than dense homogeneous networks. Liu *et al.*'s work attracted lots of attention, and many researchers have started to focus on the topic of network controllability.

However, Liu *et al.*'s theory only suits directed networks, which cannot work for undirected networks. Yuan in 2013 introduced a general controllability paradigm for any network, which is called the exact controllability framework [13]. It identifies the minimum number of driver nodes based on the maximum geometric multiplicity of all eigenvalues of the adjacency matrix. Due to the higher computational complexity, Yuan's framework is used less than Liu *et al.*'s structural controllability in applications.

2.2. OPTIMIZATION OF NETWORK CONTROLLABILITY

The optimization of network controllability reduces the number of driver nodes needed to control the network, increasing the efficiency of controlling and reducing the cost of application. The ideal situation for controllability optimization is to achieve optimal control. There are no unmatched nodes in the network, which can be described as $N - |M*| = 0$, where N is the number of nodes and $|M*|$ denotes the size of the maximum matching in the directed network. Any node in this network can be used as a driver node to control the entire network. The network in this situation always contains a cycle of length N, but there are almost no networks with such topology in real life. Therefore, the realization of perfect matching is the goal of controllability optimization. There are two main methods currently used for optimization, changing the topology of the network [14][15] and changing the direction of the links in the network [16][17][18].

Wang [14] firstly proposed a method that can optimize the controllability of networks by structural perturbations to achieve optimal control. This method forms a new directed path by connecting the independent matching paths of the network to ensure that only one driver node is needed to control the whole network. As shown in Fig. 2.1, (a) depicts a heterogeneous network with 30 nodes, the matching links(nodes) are marked as green, and the unmatched links(nodes) are shown in gray. (b) shows that red links are added to connect the independent matching paths in sequence.

Hou proposed a method that removes redundant links and adds the same number of links randomly [15], which would not change the total number of links of the network.

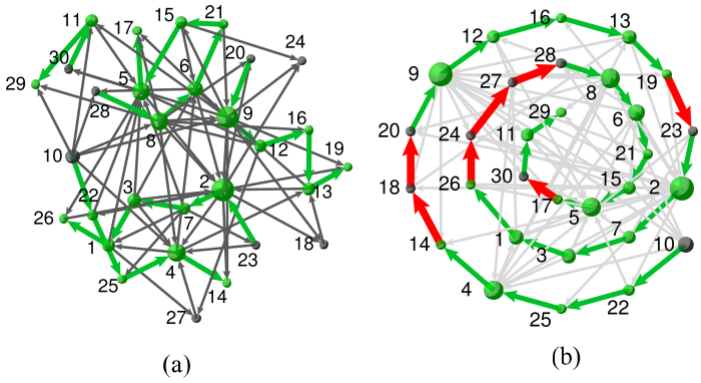


Figure 2.1: (a) The network before optimization. (b) The network after optimization. All matching paths in order. Cited from [14]

Different types of links have different influences on network controllability: critical links refer to that links whose removal will increase the number of driver nodes; ordinary links are that links whose removal will change the location of driver nodes but not change the number of driver nodes; redundant links are that links whose removal will not affect any control configuration.

The two methods discussed above are changing the topology of the network in order to optimize its controllability. However, changing the topology of networks would cost much in reality. Thus, it is necessary to find some other ways that do not need to change the topology of networks.

Based on the node residual degree, a method of assigning link direction to enhance the controllability of the network was proposed by Hou [16]. His simulations indicated that this method is more efficient than random assigning direction and can enhance the robustness of the network at the same time.

Xiao in 2014 proposed the link orientation for optimal controllability problem (EOOC) by changing the direction of links [17]. In 2015 [18], he presented a simpler link orientation method that aimed at producing more critical link directions. He also proposed a strategy that utilized only local information to enhance network controllability.

2.3. THE ROBUSTNESS OF NETWORK CONTROLLABILITY

The robustness of network controllability focuses on the change of controllability under the random attack or the targeted attack. This is also the main focus of this thesis.

Pu in 2012 [19] found that degree-based attacks are more efficient on network

structural controllability than random attacks and cascade failures also do great harm to network controllability. Nie *et al.* [20] studied the network controllability under two strategies: random and intentional attack. They found the vulnerability of controllability under random and intentional attacks behave differently as the removal fraction increases. He *et al.* [6] 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.

In this thesis, we investigate the recoverability of network controllability under link-based and node-based perturbations, respectively. The minimum fraction of driver nodes is a metric to measure controllability.

2.4. THE INFLUENCE OF NETWORK PROPERTIES ON CONTROLLABILITY

The influence of network properties on controllability mainly studies the impact of network metrics such as degree distribution, betweenness, average shortest path, etc. on network controllability. Liu *et al.* [2] in 2011 studied the influence of the degree distribution of the network on controllability. They found that the degree distribution is the most critical factor for network controllability. Meanwhile, they found that driver nodes tend to avoid hubs.

In 2013, Posfai [21] found that clustering and modularity have no impact on network controllability, but the density of the driver nodes is linearly related to the out-degree-in-degree correlation, that it is quadraticly related to the out-out-degree correlation and in-in-degree correlation, and it has no correlation with the in-out degree.

In [22], Menichetti *et al.* showed that the density of nodes with in-degree and out-degree equal to one and two, determines the number of driver nodes. Based on these results, they also proposed an algorithm to improve the controllability of networks by adding links for nodes with low degree.

In summary, the research on the controllability of complex networks is based on the structural controllability analysis framework proposed by Liu *et al.* [2] and is supplemented by the strict controllability framework proposed by Yuan *et al.* [13]. The studies of network controllability have gradually developed to optimization, robustness, and the influence of network metrics on it since the basic theoretical research work was proposed. Although some preliminary results have been achieved, there are still many problems to be solved. Overcoming and solving these problems will help people achieve the goal of controlling complex networks.

This thesis studies the recoverability of network controllability, which is extended from the research on the robustness of network controllability. We will propose a method that can analytical express the fraction of driver nodes during the attack process and the subsequent recovery process based on generating functions.

3

APPROACH

This chapter introduces the basic related theories of network controllability and methods used in this research. Section 3.1 introduces several complex network models used in this research, including Erdős-Rényi networks, regular networks, Swarm Signalling Networks, etc. Section 3.2 introduces the structural control theory in detail. Section 3.3 introduces the theoretical formulas based on statistical physics for the minimum number of driver nodes. Section 3.4 introduces the theoretical approximations proposed by Sun *et al.* [8][9] for the minimum of driver nodes of the networks after attacking or adding links randomly. Section 3.5 presents an approach for measuring the network recoverability in two scenarios.

3.1. THE BASIC NETWORK MODELS

Complex network models are usually divided into the different types according to their structure and basic properties. Regular networks have a fixed network structure in the sense that every node has the same degree. Erdős-Rényi networks have a degree sequence which is approximately a Poisson distribution and a small network diameter. Small-world networks have high clustering coefficient and short average path length. Scale-free networks have a power-law distribution degree distribution, implying there are a few nodes with very high degree values (hubs). The types of networks that are researched in this project are described in detail below.

3.1.1. 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. Erdős-Rényi Network is named after two mathematicians, Pál Erdős and Alfréd Rényi, who have done much fundamental works on graph theory. The degree distribution of the ER networks has the binomial distribution which approximates the Poisson distribution:

$$P(k) = \binom{N}{k} p^k (1-p)^{N-k} \approx \frac{\langle k \rangle^k e^{-\langle k \rangle}}{k!} \tag{3.1}$$

where $< k >$ is the average degree.
There are two methods to generate a directed ER network:

1. $G(N, p)$ model
   Step1: Generate a graph with N isolated nodes;
   Step2: Iterate each pair of nodes and add a directed link between each pair with probability p.

2. $G(N, L)$ model
   Step1: Generate a graph with N isolated nodes;
   Step2: Place L directed links randomly.

The directed network which is generated by the $G(N, p)$ model has N nodes and $pN(N-1)$ links and its average out-degree is $< k_{out} >= p(N-1)$, which always equals the average in-degree. For $G(N, L)$, the ER networks have N nodes and L links and its average out-degree is $< k_{out} >= \frac{L}{N}$. In this thesis, we use $G(N, L)$ to generate ER networks for simulation.

3.1.2. REGULAR NETWORKS

Regular networks are networks whose nodes all have the same degree. If the degree of all nodes is $< k >$, this graph is called a k-regular graph. Its degree distribution satisfies the Dirac delta function:

$$P(k) = \delta(\langle k \rangle - k) \tag{3.2}$$

In the directed k-regular graph, we assume that both the out- and in-degree are fixed ($< k_{out} >=< k_{in} >=< k >$). It has N nodes and $N < k >$ links.

3.1.3. SWARM SIGNALLING NETWORKS (SSNS)

In 2013, Kamareji *et al.* [23] discussed the resilience and controllability of dynamic collective behaviors. They devised the swarm signaling networks based on the topology to research the dynamics of information transfer channels. A SSN is modeled as a directed network with k-regular out-degree distribution and Poisson in-degree distribution with average k as:

$$P_{in}(k_{in}) = \frac{k^{k_{in}} e^{-k}}{k_{in}!} \tag{3.3}$$

$$P_{out}(k_{out}) = \delta(k - k_{out}) \tag{3.4}$$

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.

3.1.4. REAL-WORLD NETWORKS

There are many networks in the real world, such as social networks, information networks, technology networks, and biological networks [24]. We select some real communication networks from the Topology ZOO [25] and the Network Repository [26]

for the case study. The 4 real-world networks that are used in this thesis are described in Table 3.1. $< k >$ is the average out-degree, which equals the average in-degree.

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

Table 3.1: Topological properties of 4 real-world networks

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

$$G(x) = \frac{N(k=0) + N(k=1) \times x + N(k=3) \times x^2 + \cdots + N(k=n-1) \times x^{n-1}}{n} \tag{3.5}$$

where n is the total number of nodes in the network, $N(k=0)$ is the number of nodes whose degree is zero, and so on.

3.2. THE STRUCTURAL CONTROL THEORY

3.2.1. THE SYSTEM DYNAMICS OF NETWORKS' CONTROLLABILITY

In real life, most processes running on complex networks are non-linear, which system dynamics are challenging to express in a general mathematical equation. However, the performance of non-linear systems is similar to that of linear systems in many aspects [27]. Considering the linear time-invariant (LTI) dynamics of the complex network with N nodes:

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t) \tag{3.6}$$

where the vector $x(t) = (x_1(t), x_2(t), ..., x_N(t))^T$ is the state of N nodes at time t; the $N \times N$ matrix A describes the interaction strength between nodes; 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 $u(t) = (u_1(t), u_2(t), ..., u_M(t))^T$ expresses the signals that are imposed on the M nodes which are controlled by an outside controller.

According to the classical control theory, the system, which is expressed in Eq. 3.6, is controllable if it can be driven from any initial state to any desired final state in finite time. In control theory, there are two conditions that are always used for identifying a system, whether controllable or not: Kalman's controllability rank condition [11] and PBH controllability condition [28]. In the following, we will introduce Kalman's condition in detail.

3.2.2. KALMAN'S CONTROLLABILITY RANK CONDITION

A LTI sytem is controllable if the matrix

$$C = [B, AB, A^2B, ..., A^{N-1}B] \tag{3.7}$$

has full rank, which means,

$$rank(C) = N \tag{3.8}$$

According to Kalman's controllability criterion, for a given complex network system, matrix A is given. Therefore, it is critical to find a suitable input matrix B so that the system satisfies Kalman's controllability condition, which determines that the network is controllable. If an external control signal is applied to every node in the network, the system must be fully controllable. The reason is that matrix B is a diagonal matrix with rank N, satisfying Kalman's controllability condition. However, it would be better if fewer nodes in the network are selected to control the whole system.

3.2.3. STRUCTURAL CONTROLLABILITY THEOREM

The specific weights between the nodes of the networks are usually unknown for real-world networks. And in most cases, only the topology of the network is known. Thus, Liu *et al.* [2] proposed it is feasible to use structural controllability to avoid the problem that many real-world networks' weights are unknown. They regarded the network as a structural matrix where non-zero entries represent a link between two nodes and a zero represents that there is no link between the two nodes.

The structural controllability shows that for a system composed of a structural matrix, if it is possible to fix the free parameters in A and B to specific values so that the system is controllable, the system is called structure controllable. If a system is structurally controllable, it is completely controllable for almost all parameter values, except for the all-zero state and some proper algebraic variety.

3.2.4. MATCHING PROBLEM

The matching M of an undirected graph G is a link set such that any two links in this set do not have any common nodes. A node is matched if it is incident to a link in the set M. Otherwise, it is unmatched.

The matching M of a directed graph G is a link set 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 link that belongs to this set M. Otherwise, it is unmatched.

For undirected and directed graphs, maximum matching is the matching set that includes the maximum number of links. It is worth noting that there can be several different maximum matching sets. However, the maximum numbers of these sets are the same, which means the maximum number of matched nodes is fixed. If all nodes of a graph are matched, this is called the perfect match.

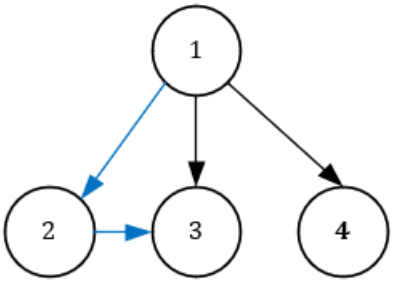
The minimum number of driver nodes (N_D) to fully control a directed network depends on the maximum matching of this network:

$$N_D = max\{N - |M*|, 1\} \tag{3.9}$$

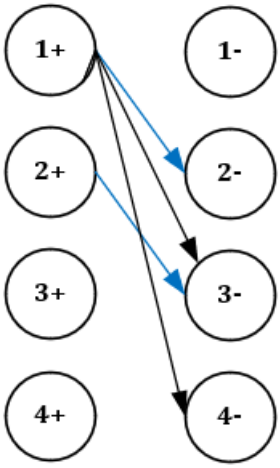
where N is the size of the network and $|M*|$ denotes the size of the maximum matching in the directed network.

3.2.5. SIMULATIONS FOR DRIVER NODES

In this thesis, we research the controllability of directed networks. To determine the maximum matching of directed networks, a bipartite network G_B with $2N$ nodes and L links is constructed to represent the directed network G with N nodes and L links, as shown in Fig. 3.1:



(a) A directed graph



(b) Bipartite representation of the digraph in (a)

Figure 3.1: Matching in digraph and its bipartite representation, the matching links are shown in blue

In Fig.3.1(b), the left column with signature "+" represents source nodes, and "-" represents target nodes. The links between them are still original links of the directed graph. There are many algorithms that can calculate the maximum matching of a bipartite graph efficiently. In our simulations, the algorithm *g.maximum_bipartite_matching()* in igraph-python is used directly to find the size of the maximum matching.

3.3. ANALYTICAL APPROXIMATIONS FOR DRIVER NODES

The generating function is an important method in combinatorics, which corresponds the discrete number sequence to the formal power series. Generating functions can also be used in complex networks. In Li's paper [29], he used the generating function to express the probability that all links of a randomly chosen node are in a specific state, which is written as:

$$G(x) = \sum_{k=0}^{\infty} p_k x^k \tag{3.10}$$

x is the probability that a link is in a certain state, and p_k is the probability that this node has degree k.

We can also use the excess degree distribution [29] as Eq. 3.11 to express the probability of a node with degree k reached by a randomly chosen link.

$$q_k = \frac{p_k k}{\sum_{k=0}^{\infty} p_k k} = \frac{p_k k}{\langle k \rangle} \tag{3.11}$$

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

$$H(x) = \sum_{k=1}^{\infty} q_k x^{k-1} \tag{3.12}$$

Liu *et al.* in 2011, proposed a way to compute the minimum number of driver nodes [2]. The authors used the method in statistical physics to derive the minimum fraction of driver nodes with given generating functions of out-degree and in-degree distributions.

The general function for the minimum fraction of driver nodes n_D that Liu *et al.* [2] obtained is:

$$\begin{aligned} n_D &= \frac{N_D}{N} \\ &= \frac{1}{2}\{G_{in}(\omega_2) + G_{in}(1-\omega_1) - 2 + G_{out}(\hat{\omega}_2) + G_{out}(1-\hat{\omega}_1) + k(\hat{\omega}_1(1-\omega_2) + \omega_1(1-\hat{\omega}_2))\} \end{aligned} \tag{3.13}$$

where N_D is the number of driver nodes, N is the size of this network, k is the average out-degree, and ω_1, ω_2, $\hat{\omega}_1$, $\hat{\omega}_2$ satisfy:

$$\omega_1 = H_{out}(\hat{\omega}_2) \tag{3.14}$$

$$\omega_2 = 1 - H_{out}(1-\hat{\omega}_1) \tag{3.15}$$

$$\hat{\omega}_1 = H_{in}(\omega_2) \tag{3.16}$$

$$\hat{\omega}_2 = 1 - H_{in}(1-\omega_1) \tag{3.17}$$

3.3.1. MINIMUM NUMBER OF DRIVER NODES FOR SSN

In the following, we apply Liu *et al.*'s equation [2] of n_D Eq. 3.13 on SSN as an example. The out-degree distribution of SSN is regular, which is given in Eq. 3.4, where k is the average out-degree. The in-degree distribution approximates the Poisson distribution, which is given in Eq. 3.3, where k is the average in-degree. After substituting in-degree distribution and out-degree distribution into the generating functions Eq. 3.10 and Eq. 3.12, the generating functions of SSN's degree distributions can be expressed as:

$$G_{out}(x) = x^k \tag{3.18}$$

$$G_{in}(x) = e^{-k(1-x)} \tag{3.19}$$

$$H_{out}(x) = x^{k-1} \tag{3.20}$$

$$H_{in}(x) = e^{-k(1-x)} \tag{3.21}$$

and $\omega_1, \omega_2, \hat{\omega}_1, \hat{\omega}_2$ satisfy:

$$\omega_1 = \hat{\omega}_2^{k-1} \tag{3.22}$$
$$\omega_2 = 1 - (1 - \hat{\omega}_1)^{k-1} \tag{3.23}$$
$$\hat{\omega}_1 = e^{-k(1-\omega_2)} \tag{3.24}$$
$$\hat{\omega}_2 = 1 - e^{-k\omega_1} \tag{3.25}$$

If we assume:

$$\omega_1 = 1 - \omega_2 \tag{3.26}$$
$$\hat{\omega}_2 = 1 - \hat{\omega}_1, \tag{3.27}$$

then the pair of equations Eq. 3.23 and Eq. 3.24 follows from Eq. 3.22 and Eq. 3.25. As a result, the minimum fraction of driver nodes n_D follows:

$$n_D = (1 - e^{-k(1-\omega_2)})^k - 1 + e^{-k(1-\omega_2)} + k(1-\omega_2)e^{-k(1-\omega_2)} \tag{3.28}$$

where ω_2 satisfies:

$$1 - \omega_2 = (1 - e^{-k(1-\omega_2)})^{k-1} \tag{3.29}$$

In our simulations, we generate 1000 SSNs with the same number of nodes 20,000 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 simulation and the analytical approximations is shown in Fig. 3.2 and Table 3.2. The simulation and approximations fit very well, which means Eq. 3.28 can be used to determine the minimum fraction of driver nodes for SSN.

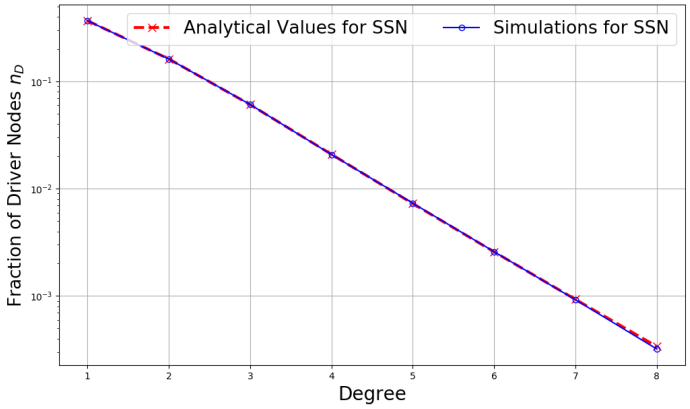


Figure 3.2: Performance comparison of the approximation Eq. 3.28 and simulations for n_D of SSN with 20,000 nodes

k	Simulation	Eq. 3.28
1	0.36765	0.36788
2	0.16176	0.16190
3	0.06096	0.06076
4	0.02081	0.02092
5	0.00732	0.00726
6	0.00259	0.00258
7	0.00092	0.00093
8	0.00032	0.00034

Table 3.2: Comparing values from Eq. 3.28 with simulation for SSN with 20,000 nodes

3.3.2. MINIMUM NUMBER OF DRIVER NODES FOR ER NETWORKS

For directed Erdős-Rényi networks, the generating functions of the degree distributions are:

$$G_{out}(x) = e^{-k(1-x)} \tag{3.30}$$

$$G_{in}(x) = e^{-k(1-x)} \tag{3.31}$$

$$H_{out}(x) = e^{-k(1-x)} \tag{3.32}$$

$$H_{in}(x) = e^{-k(1-x)} \tag{3.33}$$

and the expression of the minimum fraction of driver nodes n_D is:

$$n_D = e^{-k\omega_1} + exp(-ke^{-k\omega_1}) - 1 + k\omega_1 e^{-k\omega_1} \tag{3.34}$$

where ω_1 satisfies:

$$\omega_1 = exp(-ke^{-k\omega_1}) \tag{3.35}$$

We generate 1000 ER networks with the same number of nodes 20,000 but with different out-degree k, ranging from 1 to 8 to compute the fraction of driver nodes. Fig. 3.3 and Table 3.3 compare the fraction of driver nodes by the average simulation results and the analytical approximations from Eq. 3.34. The discrepancy between simulation and analytical values is very tiny, which indicates that Eq. 3.34 can estimate the minimum fraction of driver nodes that are needed to control the ER network.

3.4. ANALYTICAL APPROXIMATIONS FOR CONTROLLABILITY UNDER PERTURBATIONS

3.4.1. REMOVE A FRACTION p OF THE LINKS AT RANDOM

In this section, we deduce the analytical approximations for the minimum fraction of driver nodes of a network where a fraction p of the links is removed at random.

We deduce the analytical expression for the fraction n_D of driver nodes in SSNs where a fraction p of links is randomly removed. An important step is to find the degree distribution of the resulting network $G(N, L(1-p))$ after a fraction p of links are randomly

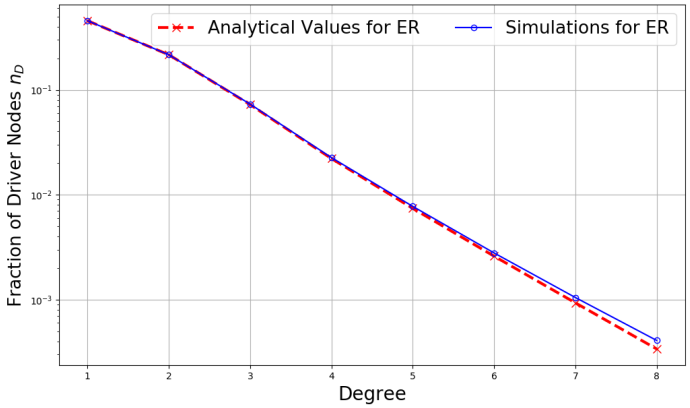


Figure 3.3: Performance comparison of the approximation Eq. 3.34 and simulations for n_D of ER networks with 20,000 nodes

k	Simulation	Eq. 3.34
1	0.45610	0.45594
2	0.21608	0.21607
3	0.07314	0.07231
4	0.02261	0.02216
5	0.00775	0.00742
6	0.00279	0.00260
7	0.00105	0.00093
8	0.00041	0.00034

Table 3.3: Comparing values from Eq. 3.34 with simulation for ER networks with 20,000 nodes

removed from the original graph $G_0(N,L)$. The degree distribution $Pr[D_G = i]$ [30] of the resulting network can be expressed as:

$$Pr[D_G = i] = (1-p)^i \sum_{j=i}^{N-1} \binom{j}{i} p^{j-i} Pr[D_{G_0} = j] \tag{3.36}$$

where $p = \frac{m}{L}$ is the fraction of removed links in the original network, and m is the number of links that are removed randomly.

The new degree distribution is used to obtain the new generating functions of SSNs in which a fraction of p of the links are removed at random. The generating functions of

the new out-degree and in-degree distributions satisfy:

$$\bar{G}_{out}(x) = (p + (1-p)x)^k \tag{3.37}$$

$$\bar{G}_{in}(x) = e^{-k(1-p)(1-x)} \tag{3.38}$$

$$\bar{H}_{out}(x) = (p + (1-p)x)^{k-1} \tag{3.39}$$

$$\bar{H}_{in}(x) = e^{-k(1-p)(1-x)} \tag{3.40}$$

The average out- and in-degree after removing a fraction p of the links are denoted by $\bar{k}$, which satisfies:

$$\bar{k} = \bar{k}_{out} = \bar{k}_{in} = k(1-p) \tag{3.41}$$

After substituting the new generating functions into Eq. 3.13, the fraction of driver nodes of SSN after a fraction p of the links have been removed satisfies [8]:

$$n_D = (p + (1-p)(1 - e^{-k(1-p)(1-\omega_2)}))^k - 1 + e^{-k(1-p)(1-\omega_2)} + k(1-p)(1-\omega_2)e^{-k(1-p)(1-\omega_2)} \tag{3.42}$$

where ω_2 satisfies:

$$1 - \omega_2 = (p + (1-p)(1 - e^{-k(1-p)(1-\omega_2)}))^{k-1} \tag{3.43}$$

The complete proof of the formulas given above is given in the Appendix A.
When $p = 0$, there is no links removal. Eq. 3.42 and Eq. 3.43 become Eq. 3.28 and Eq. 3.29.

3.4.2. ADD A FRACTION f OF THE LINKS AT RANDOM

In this section, we generalized the analytical approximations for the graph with m links being added randomly by considering SSNs again. The fraction of added links is denoted as f, which satisfies:

$$f = \frac{m}{N(N-1) - L} \tag{3.44}$$

The average out- and in-degree is denoted by $\bar{k}$, which is expressed as:

$$\bar{k} = k + f(N - 1 - k) \tag{3.45}$$

$f(N-1-k)$ means each node in the original network has $N-1-k$ out-going links that can be added randomly with probability f. The degree distribution $Pr[D_G = i]$ [30] of the network after a fraction f of links addition is expressed as:

$$Pr[D_G = i] = (1-f)^{N-1-i} \sum_{j=0}^{i} \binom{N-1-j}{i-j} f^{j-i} Pr[D_{G_0} = j] \tag{3.46}$$

Then, the new generating functions of the out- and in-degree distributions respectively satisfy:

$$\bar{G}_{out}(x) = x^k (1 - f(1-x))^{N-1-k} \tag{3.47}$$

$$\bar{G}_{in}(x) = e^{-\bar{k}(1-x)} \tag{3.48}$$

$$\bar{H}_{out}(x) = \frac{x^{k-1}}{\bar{k}} (\bar{k} - f(N-1)(1-x))(1 - f(1-x))^{N-2-k} \tag{3.49}$$

$$\bar{H}_{in}(x) = e^{-\bar{k}(1-x)} \tag{3.50}$$

Therefore, the minimum fraction of driver nodes of SSN after additions is expressed as [9]:

$$n_D = e^{-\bar{k}(1-\omega_2)} + (1-e^{-\bar{k}(1-\omega_2)})^k (1-fe^{-\bar{k}(1-\omega_2)})^{N-1-k} - 1 + \bar{k}(1-\omega_2)e^{-\bar{k}(1-\omega_2)} \tag{3.51}$$

where ω_2 satisfies:

$$\bar{k}(1-\omega_2) = (1-e^{-\bar{k}(1-\omega_2)})^{k-1}(\bar{k} - f(N-1)e^{-\bar{k}(1-\omega_2)})(1-fe^{-\bar{k}(1-\omega_2)})^{N-2-k} \tag{3.52}$$

The complete proof of the formulas given above is given in the Appendix B.
When $f = 0$, there is no links addition. Eq. 3.51 and Eq. 3.52 become Eq. 3.28 and Eq. 3.29.

3.5. R-VALUE AND TWO RECOVERY SCENARIOS

In this thesis, the recoverability of the network controllability can be assessed by the efficiency that the minimum fraction n_D of driver nodes return to the original state under perturbations of the network topology.

3.5.1. 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 [5]. 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 thesis is:

$$R = \frac{1-n_D}{1-n_{D_0}} \tag{3.53}$$

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 R-value equals 0, it means the network controllability is completely destroyed, and all nodes need to be controlled to control the whole network.

In the following chapters, a challenge indicates an event that changes the network topology and thus possibly changes the R-value. In this thesis, 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. An elementary challenge for node-based attack and recovery is one node removal and its links removal in the attack phase and one node addition and adding the number of its original links 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 $R[k]_{1\leq k\leq M}$, where k is the sequence number of challenges.

3.5.2. RECOVERY IN SCENARIO A

In this thesis, 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 or the nodes are removed one by one until the R-value reaches a predefined threshold $R_{threshold}$. The number of removed links or removed nodes that makes R-value reach the predefined threshold is denoted as K_a. Then the recovery process starts from the remaining network $G_{attacked}(N, L-K_a)$ or $G_{attacked}(N-K_a, L_{remained})$. 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. If the elementary challenges are node-based removals and additions, the recovery process in Scenario A assumes that the removed nodes are added, and k out-links and k in-links are added between the added node and any random nodes, where k is the average out- and in-degree of the original network before the attack process.

For link-based random attack and random recovery in Scenario A, the generating function during the attack process [30] is denoted as $\bar{G}(x)$ and the generating function during the subsequent recovery process [9] is denoted as $\bar{\bar{G}}(x)$:

$$\begin{cases} \text{Attack process: } \bar{G}(x) = G(p+(1-p)x), p = \frac{m}{L} \\ \text{Recovery process: } \bar{\bar{G}}(x) = (1-f(1-x))^{N-1}\bar{G}(\frac{x}{1-f(1-x)}), f = \frac{m}{N(N-1)-L-K_a} \end{cases} \tag{3.54}$$

For the above general relations of generating functions, we still use SSN as an example to illustrate the method. The generating functions of an original SSN's out- and in-degree distributions follow Eq. 3.18 and Eq. 3.19. After a fraction p of the links is removed, the generating functions are written as Eq. 3.37 and Eq. 3.38. Then, we substitute the attacked SSN's generating functions of the degree distributions (Eq. 3.37 and Eq. 3.38) to the general relations of generating functions about recovery (Eq. 3.54). As a result, the generating functions of out- and in-degree distributions of recovered networks are expressed as:

$$\bar{\bar{G}}_{out}(x) = (1-f(1-x))^{N-1}(p+(1-p)\frac{x}{1-f(1-x)})^k \tag{3.55}$$

$$\bar{\bar{G}}_{in}(x) = (1-f(1-x))^{N-1}e^{-k(1-p)(1-\frac{x}{1-f(1-x)})} \tag{3.56}$$

According to these new generating functions and Eq. 3.13 by Liu *et al.*, the minimum fraction of driver nodes to control the network during the recovery phase can be analytically expressed.

For node-based random attack and random recovery in Scenario A, the generating function in the attack process is the same as that for link-based attack, while p is the fraction of removed nodes. During the recovery process, there are many ways of adding nodes in Scenario A. In this thesis, we assume that one removed node is added, and k out-links and k in-links are established randomly between this node and any random nodes in each recovery step. Thus, we still use the general formula about generating

functions for link-based recovery in node-based recovery process, while $f = \frac{2km}{N(N-1)}$.

$$\begin{cases} \text{Attack process}: \bar{G}(x) = G(p + (1-p)x), p = \frac{m}{N} \\ \text{Recovery process}: \bar{\bar{G}}(x) = (1 - f(1-x))^{N-1} \bar{G}(\frac{x}{1-f(1-x)}), f = \frac{2km}{N(N-1)} \end{cases} \tag{3.57}$$

where $2km$ is the number of links are added in each step.

3.5.3. 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 added until the network returns to the original state under the link-based recovery. For the node-based recovery, all removed nodes and their original links are added to return to the original state. A symmetric method is used in Scenario B to express the generating function in the recovery process. Eq. 3.58 and Eq. 3.59 are generating functions under the link-based challenges and node-based challenges, respectively. By using the same notation as before, $\bar{G}(x)$ [30] and $\bar{\bar{G}}(x)$ refer to the generating functions in the attack process and the subsequent recovery process, respectively.

$$\begin{cases} \text{Attack process}: \bar{G}(x) = G(p + (1-p)x), p = \frac{m}{L} \\ \text{Recovery process}: \bar{\bar{G}}(x) = G(p + (1-p)x), p = \frac{2K_a - m}{L} \end{cases} \tag{3.58}$$

In the link-based attack process, p 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 at the R-threshold, and m is the number of added links.

$$\begin{cases} \text{Attack process: } \bar{G}(x) = G(p + (1-p)x), p = \frac{m}{N} \\ \text{Recovery process: } \bar{\bar{G}}(x) = G(p + (1-p)x), p = \frac{2K_a - m}{N} \end{cases} \tag{3.59}$$

In the node-based attack process, p is the fraction of removed nodes, and m is the number of removed nodes. During the node-based recovery process, $p = \frac{2K_a - m}{N}$ where K_a is the number of removed nodes that makes the R-value drop to the R-threshold, and m is the number of added nodes.

4

RESULTS FOR REMOVAL AND SUBSEQUENT ADDITIONS OF LINKS

In this chapter, we consider the link-based attack and the subsequent link-based recovery and show the performance of the analytical approximations compared with simulation.

4.1. REMOVING AT RANDOM A FRACTION p OF THE LINKS

In Chapter 3, Eq. 3.42 gives the analytical expression of the fraction of the minimum number of driver nodes of SSN after removing a fraction p of links uniformly at random. In our simulations, we generate SSNs with the same number of nodes 10,000 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. Fig. 4.1 and Table 4.1 compare the average simulation values and analytical values.

	Simulation			Eq. 3.42		
k	p=0	p=0.2	p=0.5	p=0	p=0.2	p=0.5
1	0.36765	0.44973	0.60652	0.36788	0.44933	0.60653
2	0.16176	0.23894	0.41013	0.16190	0.23883	0.41012
3	0.06096	0.11632	0.27945	0.06076	0.11628	0.279212
4	0.02081	0.05021	0.18341	0.02092	0.05034	0.18344
5	0.00732	0.02111	0.11286	0.00726	0.02114	0.11270
6	0.00259	0.00904	0.06526	0.00258	0.00900	0.06539
7	0.00092	0.00395	0.03747	0.00093	0.00390	0.03738
8	0.00032	0.00177	0.02157	0.00034	0.00171	0.02150

Table 4.1: Comparing Eq. 3.42 with simulation results for SSNs with 10,000 nodes under links removal

In the case that there is no link removal, i.e. $p = 0$, Eq. 3.28 becomes Eq. 3.42. As

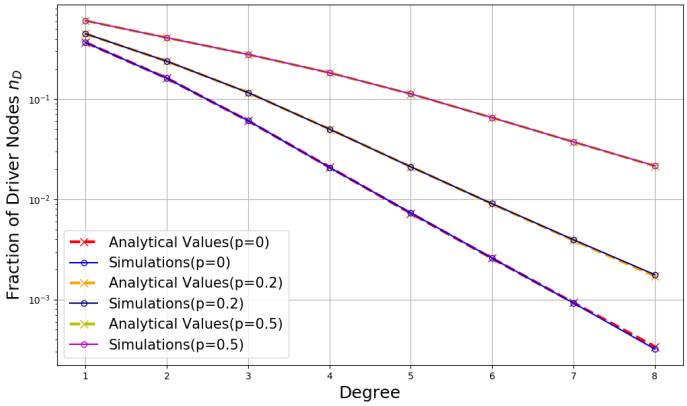


Figure 4.1: Fraction of driver nodes for SSN with 10,000 nodes as function of k for several values of p under links removal

the value of p increases from 0.2 to 0.5, the value of n_D also increases since more driver nodes are needed to make the network controllable. Fig. 4.1 also illustrates that dense networks are easier to control than sparse networks, which have a smaller degree. As Table 4.1 shows, the simulations fit very well with the approximation Eq. 3.42.

The calculation can also be applied to other directed networks when the value of p and the degree distribution $P(k_{in}, k_{out})$ are given. Through any given degree distribution $P(k_{in}, k_{out})$ of the network, the analytical approximation of the n_D of the network in which a fraction p of the links is randomly removed can be deduced and expressed. To simplify the discussion, the analytical formula for ER networks is given as Eq. 4.1 directly.

$$n_D = exp(-k(1-p)e^{-k(1-p)(1-\omega_2)}) - 1 + e^{-k(1-p)(1-\omega_2)} + k(1-p)(1-\omega_2)e^{-k(1-p)(1-\omega_2)} \quad (4.1)$$

where ω_2 satisfies the equation:

$$\omega_2 = 1 - exp(-k(1-p)e^{-k(1-p)(1-\omega_2)}) \quad (4.2)$$

A shown in Fig. 4.2 and Table 4.2, the performance of our approximation for ER networks after after a fraction p of the links is removed is very well. The results from simulation are the average values of n_D for 1000 different attacked ER networks .

4.2. ADDING AT RANDOM A FRACTION f OF THE LINKS

In the previous chapter, Eq. 3.51 provides the analytical results of minimum fraction n_D of driver nodes for randomly adding a fraction f of links to SSNs with the k-regular out-degree. In our simulations, we generate SSNs with 10,000 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 to simulate, where $f = 3 \times 10^{-6}$ and $f = 3 \times 10^{-4}$. Fig.

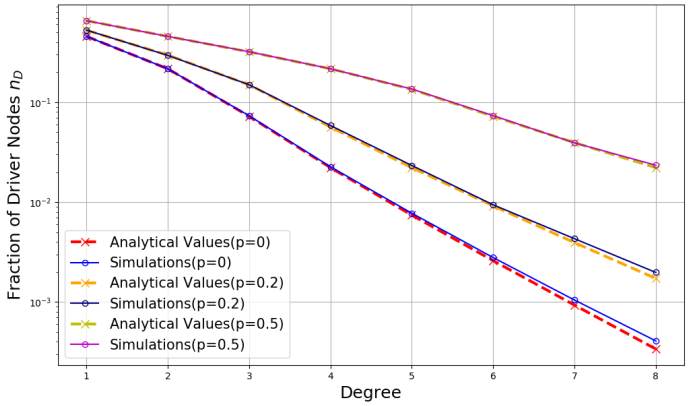


Figure 4.2: Fraction of driver nodes for ER networks with 10,000 nodes as function of k for several values of p under links removal

	Simulation			Eq. 4.1		
k	p=0	p=0.2	p=0.5	p=0	p=0.2	p=0.5
1	0.45610	0.52675	0.65341	0.45594	0.52538	0.65437
2	0.21608	0.29440	0.45565	0.21607	0.29628	0.45594
3	0.07315	0.14961	0.32024	0.07231	0.14942	0.31906
4	0.02261	0.05849	0.21615	0.02216	0.05649	0.21607
5	0.00775	0.02330	0.13560	0.00742	0.02216	0.13442
6	0.00279	0.00941	0.07344	0.00260	0.00919	0.07231
7	0.00105	0.00431	0.03920	0.00093	0.00394	0.03943
8	0.00041	0.00198	0.02343	0.00034	0.00172	0.02216

Table 4.2: Comparing Eq. 4.1 with simulation results for ER networks with 10,000 nodes under links removal

4.3 and Table 4.3 compare the values from simulation and analytical approximations.

The approximations exhibit a very good fit for the simulation when $f = 3 \times 10^{-6}$. However, when $f = 3 \times 10^{-4}$, there is a gap between the tail of the analytical approximations and that of the simulation. In other words, in the case, $f = 3 \times 10^{-4}$, the analytical approximations cannot fit with the simulations well when the degree is large. The reason is that when the degree is large, the number of driver nodes is at least 1, which follows from Eq. 3.9. Thus, n_D is always at least $\frac{1}{N}$ where N is the size of the network.

The analytical expression can also be applied in ER networks. The approximation for n_D satisfies:

$$n_D = e^{-\bar{k}\omega_1} + exp(-\bar{k}e^{-\bar{k}\omega_1}) - 1 + \bar{k}\omega_1 e^{-\bar{k}\omega_1} \tag{4.3}$$

where $\bar{k} = k + f(N-1-k)$ and $\omega_1 = exp(-\bar{k}e^{-\bar{k}\omega_1})$.

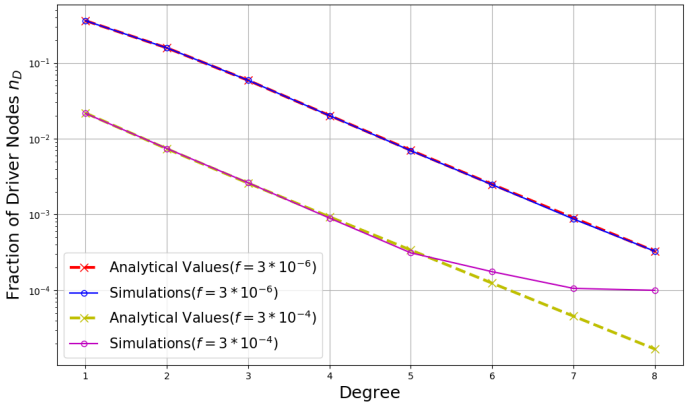


Figure 4.3: Fraction of driver nodes for SSN with 10,000 nodes as function of k for adding probability f under links recovery

	Simulation		Eq. 3.51	
k	$f = 3 \times 10^{-6}$	$f = 3 \times 10^{-4}$	$f = 3 \times 10^{-6}$	$f = 3 \times 10^{-4}$
1	0.36069	0.02158	0.36095	0.02183
2	0.15801	0.00744	0.15806	0.00736
3	0.05910	0.00262	0.05895	0.00259
4	0.02020	0.00090	0.02026	0.00093
5	0.00696	0.00031	0.00704	0.00034
6	0.00248	0.00018	0.00250	0.00012
7	0.00087	0.00011	0.00090	0.00005
8	0.00033	0.00010	0.00033	0.00002

Table 4.3: Comparing Eq. 3.51 with simulation for SSNs with 10,000 nodes

We also generate ER networks with 10,000 nodes but with different out-degree that ranges from 1 to 8. The randomly links addition probability f is set as $f = 3 \times 10^{-6}$. For a specific degree k and the probability f. 1000 ER networks are used to simulate. Fig. 4.4 and Table 4.4 compare the average values from simulations and analytical approximations, and show a good fit between them.

4.3. RECOVERABILITY IN SCENARIO A AND SCENARIO B

4.3.1. SCENARIO A

Scenario A assumes that the recovery links can be added between any two nodes if there is no link between them after attacks. The whole process is divided into two steps. The first step is attacking links one by one randomly until the R-value decreases to the threshold $R_{threshold}$, which is predefined, Then the second step is adding links one by

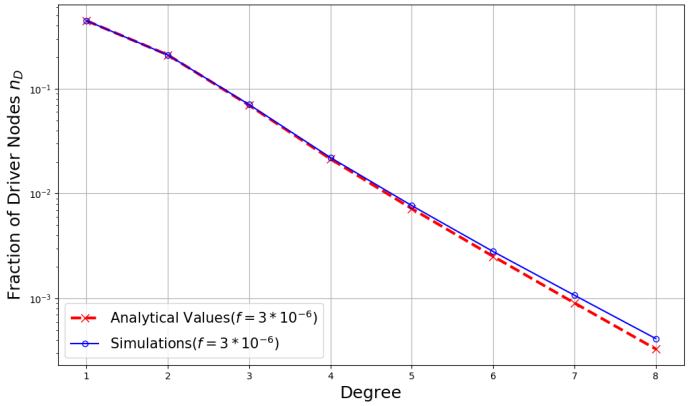


Figure 4.4: Fraction of driver nodes for ER networks with 10,000 nodes as function of k after adding a fraction f of the links

k	Simulation	Eq. 4.3
1	0.44639	0.44674
2	0.21066	0.21041
3	0.06965	0.07071
4	0.02142	0.02210
5	0.00719	0.00768
6	0.00252	0.00280
7	0.00091	0.00107
8	0.00033	0.00041

Table 4.4: Fraction of driver nodes for ER networks with 10,000 nodes as function of k after adding a fraction f of the links

one in Scenario A until the R-value returns to 1.

The R-threshold is set to 0.9 in all simulations in this thesis. In our simulation, we generate 100 SSNs with 500 nodes and out-degree $k = 2$, and each of them is simulated for 100 realizations. Each realization of processes consists of an attack process and the subsequent recovery process.

Based on Eq. 3.54, the controllability of the attacked network can be analytically expressed during the subsequent recovery process in Scenario A. The top two figures in Fig. 4.5 exemplify the envelopes of the challenges in SSN for the controllability metric R-value in Scenario A, under the random attack strategy and the recovery strategy. The approximation fits very well with the simulation, which indicates the general formulas Eq. 3.54 about generating functions that are discussed in the previous chapter work well. As shown in the bottom two figures of Fig. 4.5, the method can easily also be applied to

real-world networks and also performs well. We can find our analytical approximations of network controllability for *kdl* predict the R-value better than that for *Cogentco*, as the method is based on statistical physics and performs better for large networks.

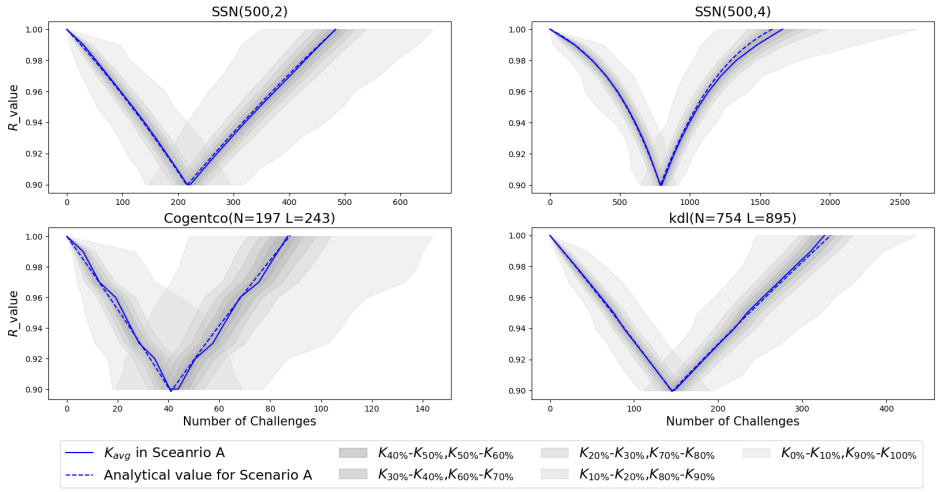


Figure 4.5: Envelopes of the challenges for SSNs with 500 nodes and different average out-degree($k_{out} = 2$ and $k_{out} = 4$) and two real-world networks (*Cogentco* and *kdl*) in Scenario A, by random attack and random recovery strategy. The threshold of R-value is 0.9. Each envelope is based on 10^4 realizations.

4.3.2. SCENARIO B

Scenario B assumes that the recovery links can only be added between the two nodes which had a connection before attacks. In other words, only the links removed in the attack process are added one by one until the network returns to the original topology.

For link-based recovery in Scenario B, we use the symmetric method Eq. 3.59 for generating functions to approximate n_D and the R-value. 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 10,000 times. Each realization consists of a link-based random attack process and a subsequent link-based random recovery process in Scenario B. Fig. 4.6 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.

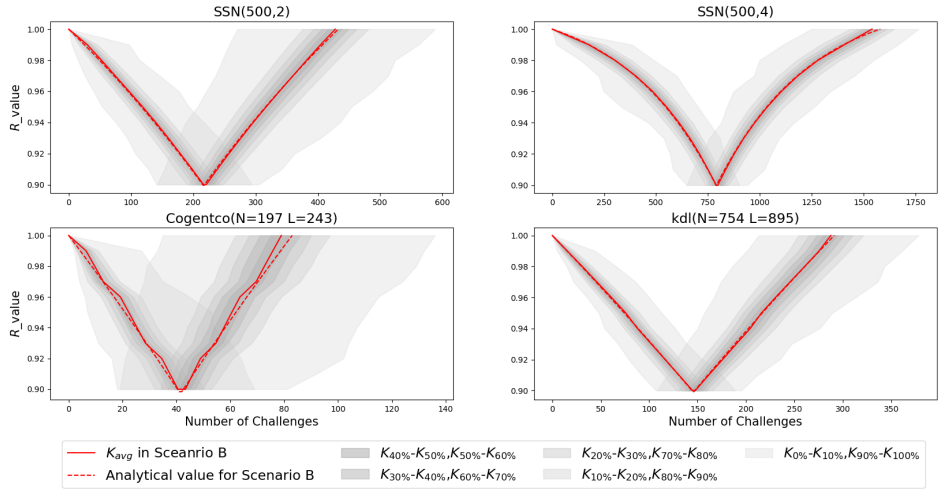


Figure 4.6: Envelopes of the challenges for SSNs with 500 nodes and different average out-degree($k_{out} = 2$ and $k_{out} = 4$) and two real networks (*Cogentco* and *kdl*) in Scenario B, by random attack and random recovery strategy. The threshold of R-value is 0.9. Each envelope is based on 10^4 realizations.

4.4. ATTACK STRATEGIES

This section studies the network controllability under six link-based attack strategies. In the general case, the damage of link-based attack to the controllability of the network is not as significant as node-based attack. For a network with N nodes and average out-degree k, removing a fraction p of the links only removes pNk links, while removing a fraction p of the nodes means removing Np nodes and about $2pNk$ links. The links that are removed under the link-based attack are about half of that under the node-based attack. It should be noted that the study here is different from the study [31] based on link cascading failure: the removal of one link will trigger the removal of other links in a cascading of failures. In this thesis, the cascading failures are not considered. Thus, removing one link will not affect the removal of other links. There are three different attack strategies that are discussed:

- Random Attack. For this strategy, the links are removed randomly and uniformly.
- Metric-based Attack. The metric-based strategy refers to the sequence of removing links by the topological metrics of links. We consider attack strategies based on metrics of links between node i and j: 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(evc) ($min(c_i c_j)$), and the maximum product of eigenvector centrality ($max(c_i c_j)$). In each challenge during the attack process under a specific strategy, a link with the related metric is removed.
- Greedy Attack. The greedy attack strategy involves removing the link that makes the R-value decrease the most in each challenge.

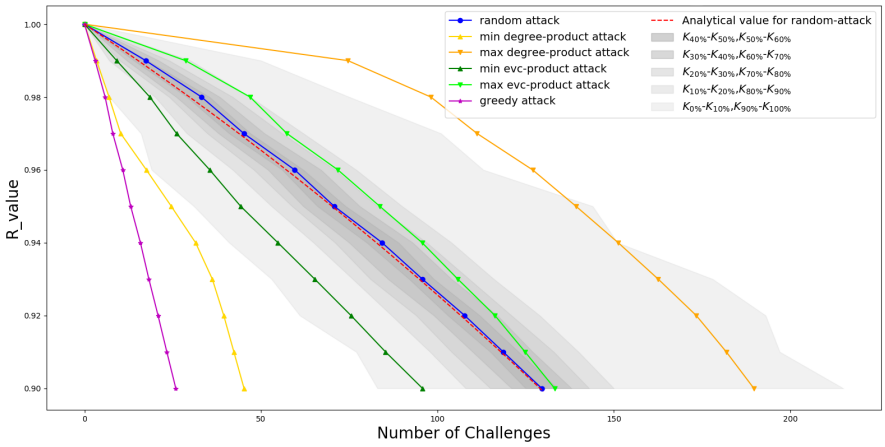


Figure 4.7: R-value as function of number of challenges under six different attack strategies in SSN(300,2)

We generate 100 SSNs with 300 nodes and with an average out-degree of 2. Each SSN is attacked by six different strategies respectively for 100 times. From Fig. 4.7, the envelope of random attack covers a large surface, which implies that the performance of a realization can deviate much from the other. The greedy strategy decreases the R-value to the threshold with the least steps. The strategy based on attacking the link with the minimum degree-product also performs well and is faster than any realization of the random attack. Although the minimum evc-product strategy can reduce the R-value faster than random attack, it performs much worse than the minimum degree-product strategy, which means the minimum fraction of driver nodes is more related to the degree than eigenvector centrality. The strategies based on maximum metrics perform worse than the random attack, especially the max degree-product strategy.

4.5. RECOVERY STRATEGIES

For simplicity, we use the random attack strategy in the attack process, and different recovery strategies are applied after the random attack. In the following, we also consider two scenarios: Scenario A and Scenario B.

4.5.1. 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 [14], 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 topology structural cases [14] of a matching path, shown in Fig. 4.8.

  1. 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;
  2. 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;
  3. isolated node: a node without any link belonging to the set of the maximum matching.

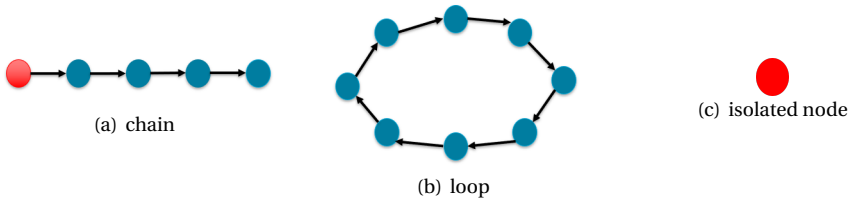
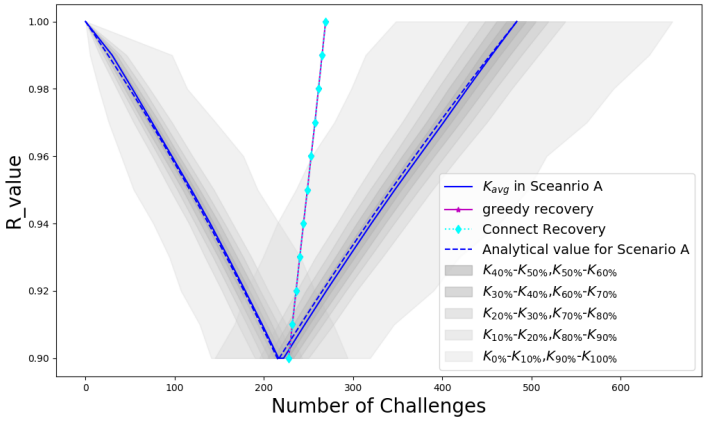


Figure 4.8: Three cases of independent path. Unmatched nodes are shown in red and matched nodes are shown in blue.

As shown in Fig. 4.9, 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 8531 s. In comparison, connect recovery strategy only costs 0.04 s 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.



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

Figure 4.9: Comparisons of different recovery strategies for SSN(500,2) in Scenario A

4.5.2. 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 randomly and uniformly 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.

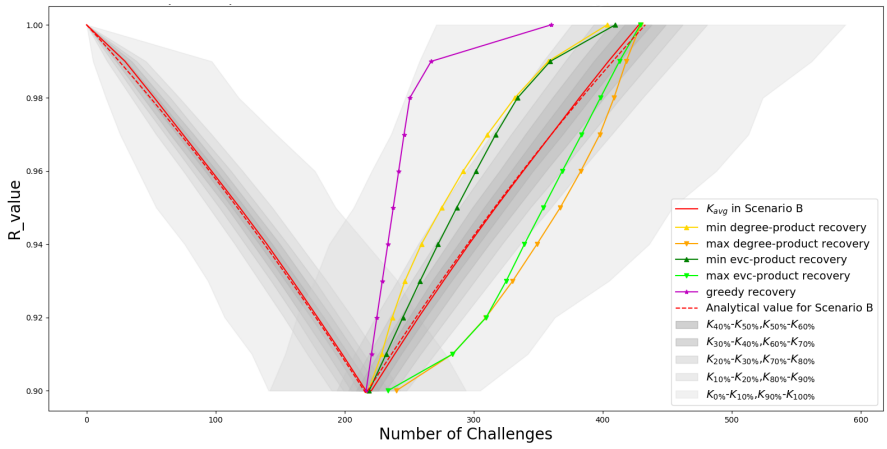


Figure 4.10: Comparisons of different recovery strategies for SSN(500,2) in Scenario B

As shown in Fig. 4.10, 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.

5

RESULTS FOR REMOVAL AND SUBSEQUENT ADDITIONS OF NODES

As mentioned in the previous chapter, the node-based attack is more harmful than the link-based attack since all links attached to the attacked node are removed under the node-based attack. In this chapter, we present the results under the node-based attack and recovery.

5.1. REMOVING AT RANDOM A FRACTION p OF THE NODES

Eq. 3.42 in Chapter 3 can also express the approximation for the minimum fraction n_D of driver nodes of SSN after removing a fraction p of the nodes at random. In this case, $p = \frac{m}{N}$, where m is the number of removed nodes, and N is the original number of nodes of the SSN. However, n_D, which is computed from Eq. 3.42, cannot be used directly, as this is the minimum fraction of driver nodes of the remained network after attacks. However, we assume that the removed nodes also need to be controlled separately to control the whole network so that the minimum fraction n'_D of driver nodes of the attacked network follows:

$$n'_D = \frac{n_D \times (N-m) + m}{N} \tag{5.1}$$

By combining the Eq. 3.42 and Eq. 5.1, the minimum fraction of driver nodes n'_D of SSN after removing a fraction p of the nodes at random satisfies:

$$\begin{aligned} n'_D =& (((p+(1-p)(1-e^{-k(1-p)(1-\omega_2)}))^k - 1 + e^{-k(1-p)(1-\omega_2)} + k(1-p)(1-\omega_2)e^{-k(1-p)(1-\omega_2)}) \\ & \times (N-Np)+Np)/N \end{aligned} \tag{5.2}$$

where $p = \frac{m}{N}$.

In our simulations, we generate SSNs with 10,000 nodes but with different out-degree k, ranging from 1 to 8. For SSN with a specific degree, we randomly remove a

fraction p of nodes, where $p = 0.2$ and $p = 0.5$. We simulate 1000 times for each situation with specific out-degree k and probability p. When p is 0, for which case there is no node and no link removal, $n'_D = n_D$, and Eq. 5.2 becomes Eq. 3.28. As shown in Fig. 5.1 and Table 5.1, n'_D of SSN after at random a fraction p of the nodes is approximated by the expression Eq. 5.2 accurately.

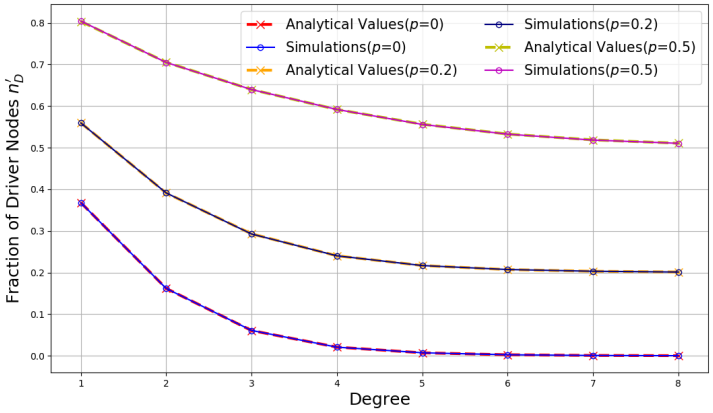


Figure 5.1: Fraction of driver nodes for SSN with 10,000 nodes as function of degree k for several values of p under nodes removal

	Simulation			Eq. 5.2		
k	p=0	p=0.2	p=0.5	p=0	p=0.2	p=0.5
1	0.36765	0.55928	0.80408	0.36788	0.55946	0.80327
2	0.16176	0.39123	0.70468	0.16190	0.39106	0.70506
3	0.06096	0.29262	0.63961	0.06076	0.29302	0.63961
4	0.02081	0.24027	0.59184	0.02092	0.24027	0.59172
5	0.00732	0.21682	0.55583	0.00726	0.21691	0.55635
6	0.00259	0.20732	0.53247	0.00258	0.20720	0.53270
7	0.00092	0.20314	0.51827	0.00093	0.20312	0.51869
8	0.00032	0.20129	0.51048	0.00034	0.20137	0.51075

Table 5.1: Comparing Eq. 5.2 with simulation for SSN with 10,000 nodes after removing a fraction p of the nodes

We apply the above method to ER networks. The analytical expression of n'_D of ER networks after a fraction p of the nodes is removed follows:

$$n'_D = (exp(-k(1-p)e^{-k(1-p)(1-\omega_2)}) - 1 + e^{-k(1-p)(1-\omega_2)} + k(1-p)(1-\omega_2)e^{-k(1-p)(1-\omega_2)} \times (N - Np) + Np)/N \tag{5.3}$$

We generate ER networks with 10,000 nodes and with different out-degree k, ranging from 1 to 8. Also, we randomly remove ER networks' nodes with probability p, where p is 0.2 or 0.5. The results of simulation are the average values of n'_D after 1000 times simulation. As shown in Fig. 5.2 and Table 5.2, we find our approximations results have a good fit with the simulations. This indicates that Eq. 5.3 is suitable for expressing n'_D of ER networks after removing a fraction p of the nodes at random.

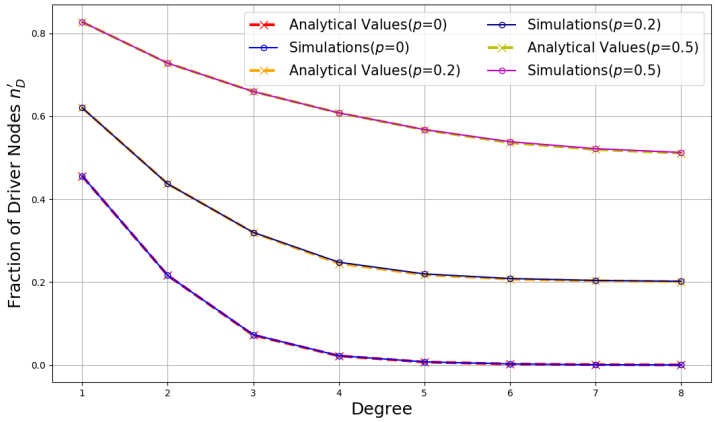


Figure 5.2: Fraction of driver nodes for ER networks with 10,000 nodes as function of degree k for several values of p under node-based removal

	Simulation			Eq. 5.3		
k	p=0	p=0.2	p=0.5	p=0	p=0.2	p=0.5
1	0.45610	0.62015	0.82724	0.45594	0.62030	0.82718
2	0.21608	0.43681	0.72792	0.21607	0.43703	0.72797
3	0.07315	0.31984	0.65951	0.07231	0.31954	0.65953
4	0.02261	0.24795	0.60798	0.02216	0.24519	0.60804
5	0.00775	0.21976	0.56780	0.00742	0.21773	0.56721
6	0.00279	0.20868	0.53858	0.00260	0.20735	0.53616
7	0.00105	0.20404	0.52169	0.00093	0.20315	0.51971
8	0.00041	0.20203	0.51262	0.00034	0.20138	0.51108

Table 5.2: Comparing Eq. 5.3 with simulation for ER networks with 10,000 nodes after removing a fraction p of the nodes

5.2. RECOVERABILITY IN SCENARIO A AND SCENARIO B

5.2.1. SCENARIO A

In Scenario A, there is no certain way to recover the nodes and their links. This thesis assumes that the removed nodes are added one by one in the recovery process. At the same time, we add k out-links and k in-links between the added node and other random nodes in each step during the recovery process, where k is the average out-degree and in-degree. If the R-value still does not return to 1 after all removed nodes are added, we keep choosing an existing node randomly from the network and adding k out-links and k in-links between this picked node and any other nodes.

In our simulations, we generate SSN and ER networks with a specific number of nodes($N = 500$) but with different out-degree ($k_{out} = 2$ and $k_{out} = 4$). We generate 100 SSNs and 100 ER networks with 500 nodes and a specific out-degree and run 100 simulations for each network. Besides synthetic networks, two real-world networks are also used for running 10,000 simulations. Each realization consists of a random attack process and a subsequent random recovery process in Scenario A. In Fig. 5.3, we compare simulation results for SSN and ER networks with 500 nodes and with different out-degree($k_{out} = 2$ and $k_{out} = 4$) to the analytical approximations Eq. 3.57. We find the analytical approximations do not fit the simulations well, especially for sparse ER networks with average out-degree $k_{out} = 2$. Besides, there is a twist at the end of the envelopes, which indicates the recovery method that only adding links randomly for existing nodes after all removed nodes are added will slow down the network controllability recovery.

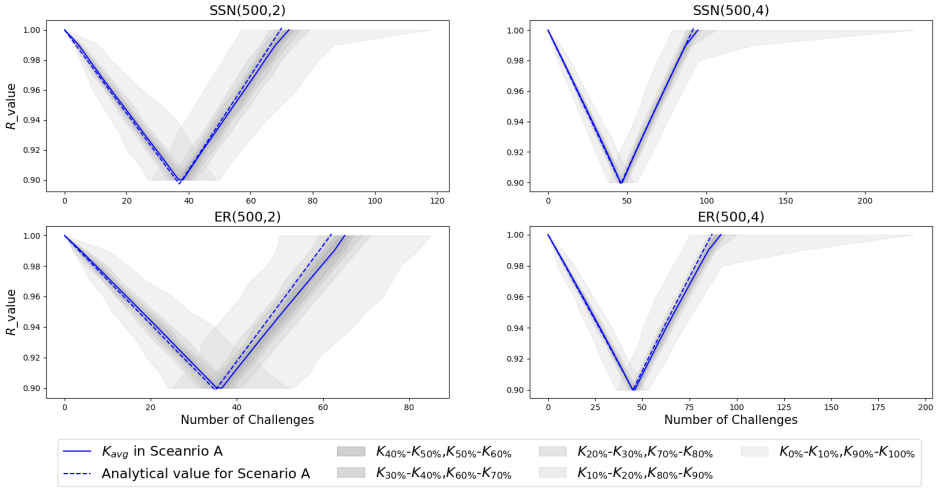


Figure 5.3: Envelopes of the challenges for SSNs and ER networks with different degree in Scenario A, by random attack and random recovery strategy. The threshold of R-value is 0.9. Each envelope is based on 10^4 realizations.

When the formulas are applied to real-world networks, as shown in Fig. 5.4, the discrepancy between the approximations and the simulation results is more obvious. The reason for the bad performance could be from two aspects. Firstly, the method of node-based recovery in Scenario A is hard to define. There are many possible ways to recover nodes and links. And it is uncertain how to recover the network controllability if the R-value still does not return to 1 after all original nodes are added. Secondly, we assume that $2k$ links are added randomly in each step for analytical approximations, which is a rather crude assumption.

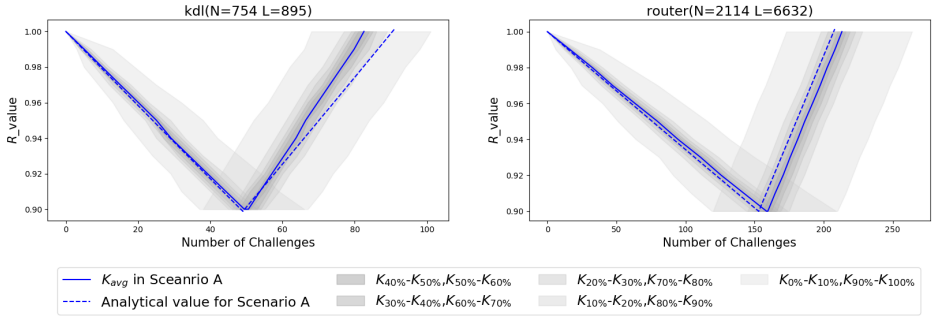


Figure 5.4: Envelopes of the challenges for two real networks in Scenario A, by random attack and random recovery strategy. The threshold of R-value is 0.9. Each envelope is based on 10^4 realizations.

5.2.2. SCENARIO B

The attack process in Scenario B is the same as in Scenario A. In the recovery process in Scenario B, we add the removed nodes one by one in each step. Also, the node's original links are added when the node is recovered. Thus, the network would return to its original state after the recovery process in Scenario B.

For node-based recovery in Scenario B, we still use the symmetric method Eq. 3.59 for generating functions and Eq. 5.1 to approximate n'_D and the R-value. In our simulations, we generate 100 SSNs and 100 ER networks with a specific number of nodes number($N = 500$) and a specific out-degree($k_{out} = 2$ or $k_{out} = 4$). Each network is simulated 100 times. Besides synthetic networks, we also use two real-world networks for simulations. For a specific real-world network, we run 10,000 simulations. Each realization consists of a random attack process and a subsequent random recovery process in Scenario B. Fig. 5.5 and Fig. 5.6 indicate that the symmetric method used in Scenario B works well for both synthetic and real-world networks.

5.3. ATTACK STRATEGIES

We consider three node-based attack strategies:

1. Random Attack. It refers to the strategy that the nodes are removed uniformly at random. With the removal of nodes, the links that belong to these nodes are

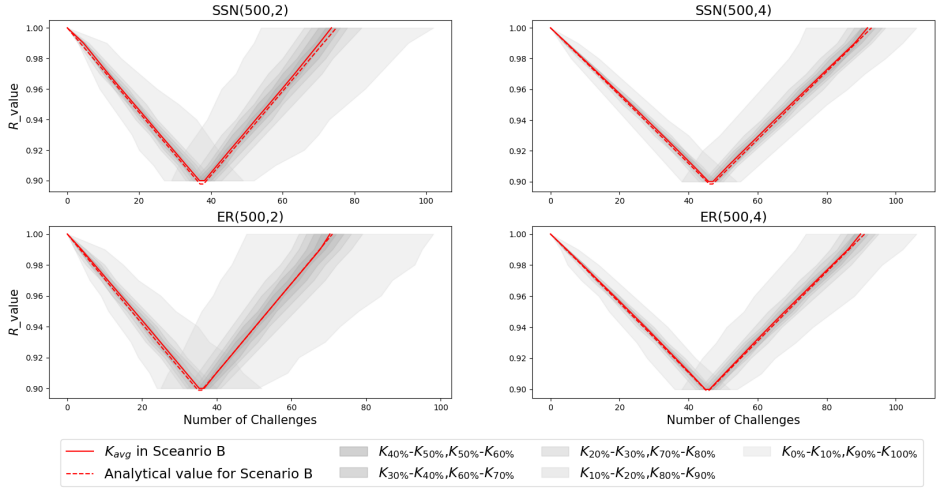


Figure 5.5: Envelopes of the challenges for SSNs and ER networks with different degree in Scenario B, by random attack and random recovery strategy. The threshold of R-value is 0.9. Each envelope is based on 10^4 realizations.

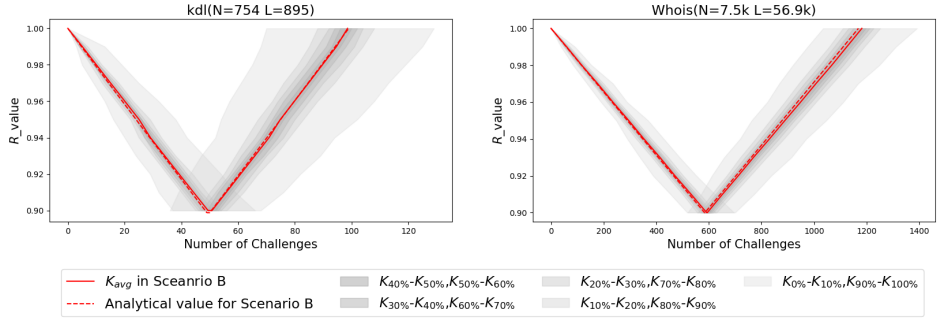


Figure 5.6: Envelopes of the challenges for two real-world networks in Scenario B, by random attack and random recovery strategy. The threshold of R-value is 0.9. Each envelope is based on 10^4 realizations.

removed at the same time.

2. Degree-based Attack [32]. This strategy attacks nodes based on degree distribution, which means nodes are attacked with probability $W(k_i) = \frac{k_i}{\sum_{j=1}^{N} k_j} = \frac{k_i}{2L}$, where k_i is the degree of node i.

3. Localized Attack [33]. The localized attack strategy describes a realistic attack method: one node's removal would influence its neighbors. It assumes that the nodes are chosen to be removed, starting from the root node that is chosen randomly, then its nearest neighbors, the next nearest neighbors, and so on.

The degree-based attack and localized attack were proposed for undirected networks. In this project, we adjust them and apply them on directed networks. For the degree-based attack, we regard the sum of nodes' in-degree and out-degree as the degree used in the computation of attacking probability. For localized attack, we ignore the direction of the links.

As shown in Fig. 5.7, the degree-based attack destroys the network controllability faster than the random attack for SSNs with average out-degree ($\bar{k}_{out} = 2$). The localized attack needs the most steps to decrease the R-value to the threshold. The difference among the steps that are taken to diminish the R-value under the three strategies is tiny. For the real-world networks *kdl*, we can find the degree-based attack is still the quickest, and the steps that localized attack needs to destroy *kdl* is the upper bound for the envelope of the random attack.

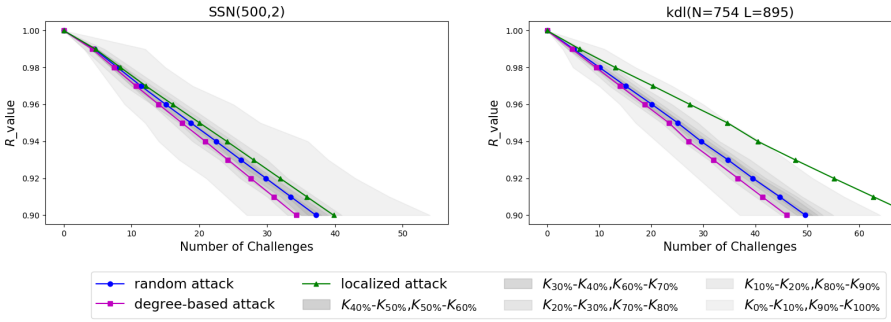


Figure 5.7: Comparisons of three node-based attack strategies in SSN(500,2) and kdl network. The threshold of R-value is 0.9.

The performance of random attack and degree-based attack is the same for random regular networks, as every node in a random regular network has the same degree; thus, the probability of being attacked under degree-based attack is the same. For Erdős–Rényi network, the localized attack performs the same as the random attack. These are shown in Fig. 5.8.

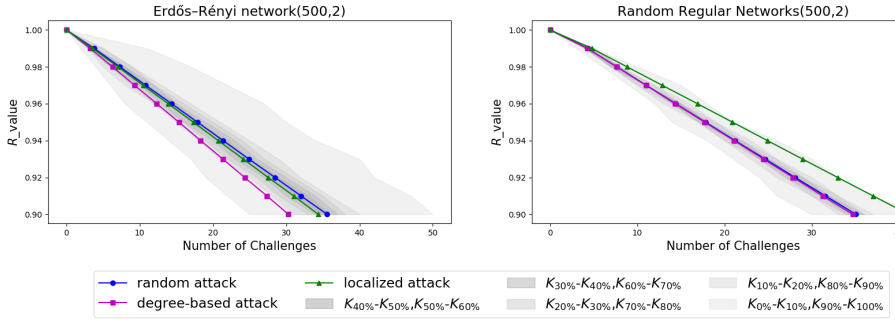


Figure 5.8: Comparisons of three node-based attack strategies in Erdős–Rényi network(500,2) and Random Regular Network(500,2). The threshold of R-value is 0.9.

5.4. DEGREE-BASED ATTACK

Let $W(k_i)$ denote the probability that a node with degree k_i is removed under the attack:

$$W(k_i) = \frac{k_i}{\sum_{j=1}^{N} k_j} \tag{5.4}$$

$$= \frac{k_i}{2L} \tag{5.5}$$

where k_i is the sum of the out-degree and in-degree of node i.
The generating function of the degree distribution of the unperturbed network satisfies:

$$G(x) = \sum_{k=0}^{\infty} p_k x^k \tag{5.6}$$

The generating function of the degree distribution of the remaining network after degree-based attack follows:

$$\bar{G}(x) = \frac{1}{\hat{p}} G(f + f^2 \frac{G'(f)}{< k >}(x-1)) \tag{5.7}$$

where $\hat{p}$ is the fraction of nodes that are not attacked, $f \equiv G^{-1}(\hat{p})$, and $< k >$ is the average degree of the original networks.

We use Eq. 5.7, Eq. 3.13, and Eq. 5.1 to compute network controllability n'_D and R-value after degree-based attack. Fig. 5.9 shows the comparison of the analytical approximations and simulations. From the figure, we can find there is an obvious discrepancy between them. To figure out which step in our model contains mistakes, we compare the analytical degree distribution of SSN after the degree-based attack and the simulations.

As shown in Fig. 5.10 the degree distribution from the generating function Eq. 5.7 cannot predict the actual degree distribution under simulations accurately, which has

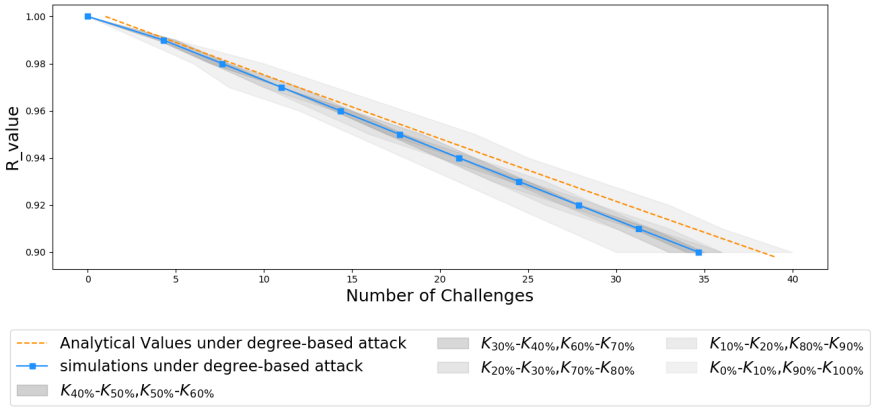


Figure 5.9: Comparison of approximations and simulation of SSN(500,2) under degree-based attack

apparent gaps between analytical values and simulations. However, the expression of degree distribution is the critical step to deduce the minimum fraction of the driver node. The generating function Eq. 5.7 cannot be used in perturbed directed networks.

Then, we investigate the out-degree-based attack and in-degree-based attack for directed networks. The difference between the two strategies and the degree-based attack is that k_i in Eq. 5.5 is out-degree or in-degree rather than the sum of out-degree and in-degree. In our simulation, we generate 100 SSNs with 500 nodes and an out-degree of 2. Each SSN is attacked by the out-degree-based attack and in-degree-based attack 100 times, respectively. Fig. 5.11 compares the degree distributions of SSN after out-degree-based attack and in-degree-based attack with the analytical values. We find the approximations for out-degree distribution can fit well with the simulations under out-degree-based attack. The in-degree distribution of SSN under in-degree-based attack is close to the analytical in-degree distribution. Thus, we can only predict one of the two degree distributions accurately.

5.5. LOCALIZED ATTACK

As introduced before, the localized attack is attacking nodes from the root node chosen at random, then its nearest neighboring nodes, the next nearest neighbors, and so on. This strategy was proposed by Shao *et al.* [33] for undirected connected networks. The authors also analytically studied the robustness of complex networks under the localized attack. Shao *et al.* found that the generating function $\bar{G}(x)$ of the degree distribution of the remaining network after the localized attack follows [33]:

$$\bar{G}(x) = \frac{1}{G(f)} G\left[f + \frac{G'(f)}{G'(1)}(x-1)\right] \tag{5.8}$$

where $G(x)$ is the generating function of the original network, $\hat{p}$ is the fraction of nodes that are not removed, and $f \equiv G^{-1}(\hat{p})$.

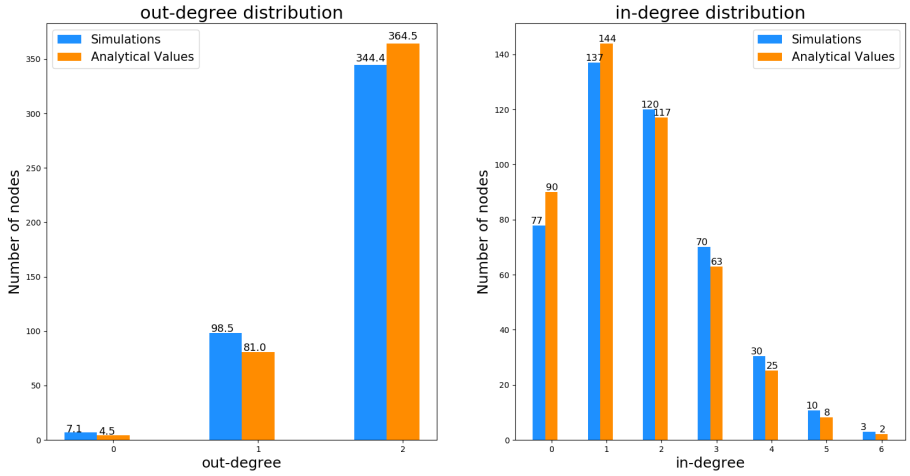


Figure 5.10: Comparisons of degree distributions from simulation and analytical values of SSN(500,2) after a fraction $p = 0.1$ of the nodes are removed by the degree-based attack

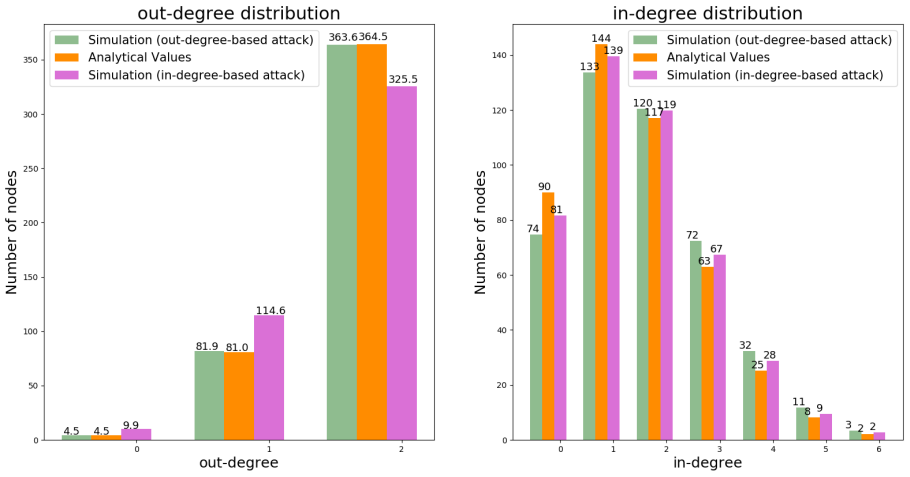


Figure 5.11: Comparisons of degree distributions from simulation under out-degree-based attack and in-degree-based attack and analytical values of SSN(500,2) after a fraction $p = 0.1$ of the nodes are removed by the degree-based attack

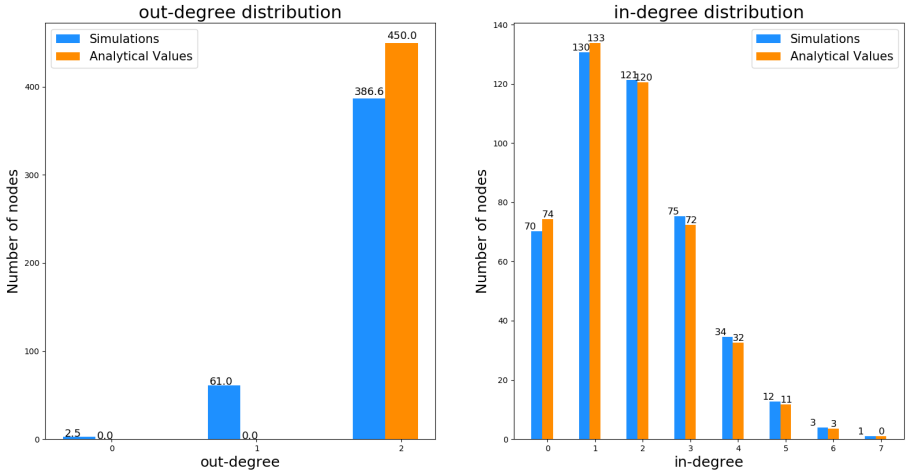


Figure 5.12: Comparisons of degree distributions from simulation under localized attack and analytical values of SSN(500,2) after a fraction $p = 0.1$ of the nodes are removed by the localized-based attack

5

In our simulation, we generate 100 SSNs with 500 nodes and an average out-degree of 2. For aach SSN, a fraction 0.1 of the nodes is removed 100 times by localized attack, which ignores the direction of links. The comparison of the out-degree and in-degree distribution from simulations and approximations is shown in Fig. 5.12. We can find that Eq. 5.8 cannot predict the generating function of degree distributions well for directed networks under the localized attack.

We adjust and extend the localized attack to out-localized attack and in-localized attack. The out-localized attack is attacking nodes according to the distances from the root node to the other nodes. On the contrary, the in-localized attack is that we attack the nodes whose distance from itself to the root node with ascending order. We also generate 100 SSNs with 500 nodes and an average out-degree of 2 and attack these SSNs 100 times by the out-localized attack and the in-localized attack, respectively. As shown in Fig. 5.13, the generating function Eq. 5.8 of out-degree distribution can predict the out-degree distribution of SSN under in-localized attack better. The in-degree distributions from approximation Eq. 5.8 and simulations under out-localized attack can fit well.

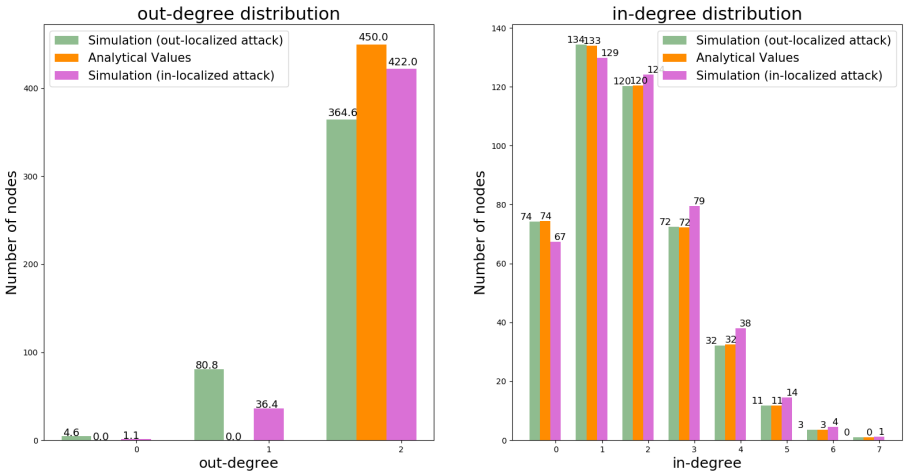


Figure 5.13: Comparisons of degree distributions from simulation under out-localized attack and in-localized attack and analytical values of SSN(500,2) after a fraction $p = 0.1$ of the nodes are removed by the localized-based attack

6

CONCLUSIONS AND FUTURE WORK

This chapter is the last chapter summarizing our findings and proposing some suggestions for future research.

6.1. CONCLUSIONS

In this thesis, we assessed the recoverability of networks based on network controllability, in terms of the fraction of minimum number of driver nodes n_D. We focused on analytical expressions for network controllability during the attack and the subsequent recovery process in Scenario A and Scenario B.

In Chapter 4, we mainly discussed the case under link-based attack and recovery. Firstly, we validated Sun *et al*'s formulas [8][9] of analytically expressing the minimum fraction of driver nodes of networks after removing links and adding links with simulations. We used the method that describes the general relations of generating functions during the attack and the subsequent recovery process to approximate the network controllability in the whole process. The recovery process has two scenarios: Scenario A refers to the case that links that can be established between any two disconnected nodes after attacking; for Scenario B only the links that are removed during the attack process are recovered. Comparing analytical values from our equations and the average simulation results shows a very good performance, both in Scenario A and Scenario B. We also compared some attacking strategies and some recovery strategies in Scenario A and Scenario B, respectively. In the attack process, the greedy attack strategy is the most destructive. For the recovery process in Scenario A, the connect strategy is optimal as it needs a minimal number of steps to recover its controllability, just like the greedy strategy, while it costs much less computation time than the greedy strategy. For the recovery process in Scenario B, the greedy recovery strategy exhibits the best performance.

In Chapter 5, we researched the case under node-based attack and node-based recovery. The formula expressing the minimum fraction of driver nodes of networks

after removing links was also applied to that of networks after removing nodes. The general relations of generating functions about attacking and recovering were also used under the node-based perturbations. However, the minimum number of driver nodes computed from the equations is the N_D of the remaining networks after attacking. The removed nodes also need to be controlled separately to control the whole network so that the number of removed nodes needs to be added. The general relations of generating functions under the recovery process in Scenario A cannot work as it considers k out-links and k in-links are added, where k is the average out-degree and in-degree, in each step in analytical expressions for simplification. The symmetric method used in Scenario B still performs accurately. We also compare three node-based attack strategies: random attack, degree-based attack, and localized attack. We tried to express the network controllability under degree-based attack and localized attack analytically but failed. Then we analyzed the reasons why it did not work.

6.2. FUTURE RESEARCH

This project represents a very small part in exploring combining network controllability and recoverability. There are still many open questions left, waiting to be explored and solved. In this section, we propose some directions for future research.

Firstly, we mainly focused on network controllability and did not quantify the recoverability in this thesis. In [6], He *et al.* quantified network recoverability based on robustness metrics, which can also be applied to network controllability. Quantifying recoverability based on network controllability can help people identify whether a network has a good ability to recover its network controllability after attacks.

Secondly, there are many possible ways to recover the nodes and links under the node-based recovery process in Scenario A. How to recover the links when recovering a node? How to recover if all removed nodes are added but the network controllability does not return to the original state? Besides the way of simulations for Scenario A, the analytical expression of network controllability under the recovery process in Scenario A also needs optimization.

Thirdly, we failed to express the network controllability of directed networks under degree-based attack or localized attack. We were not able to find expressions for the generating functions of out- and in-degree distributions of networks after degree based or localized attacks.

Lastly, this project mainly focuses on the analytical approximation of network controllability for synthetic networks and some other real-world networks. In the future, we can combine the theoretical framework of network controllability and apply it to some real-world cases, such as gene control networks, transport networks, etc.

BIBLIOGRAPHY

[1] D. J. Watts, "The "new" science of networks," *Annual Review of Sociology*, vol. 30, no. 1, pp. 243–270, 2004. [Online]. Available: https://doi.org/10.1146/annurev.soc.30.020404.104342

[2] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási, "Controllability of complex networks," *Nature*, vol. 473, no. 7346, pp. 167–173, 2011.

[3] V. Ravindran, S. V., and G. Bagler, "Identification of critical regulatory genes in cancer signaling network using controllability analysis," *Physica A: Statistical Mechanics and its Applications*, vol. 474, pp. 134–143, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378437117300699

[4] M. Cremonini and F. Casamassima, "Controllability of social networks and the strategic use of random information," *Computational social networks*, vol. 4, no. 1, pp. 1–22, 2017.

[5] P. Van Mieghem, C. Doerr, H. Wang, J. M. Hernandez, D. Hutchison, M. Karaliopoulos, and R. Kooij, "A framework for computing topological network robustness," *Delft University of Technology, Report20101218*, pp. 1–15, 2010.

[6] 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.

[7] T. Afrin and N. Yodo, "A concise survey of advancements in recovery strategies for resilient complex networks," *Journal of Complex Networks*, vol. 7, pp. 393–420, 06 2019.

[8] S. Peng, E. K. Robert, and B. Roland, "Controllability of a class of swarm signalling networks: impact of removing links," *in preparation*, 2021.

[9] S. Peng and E. K. Robert, "Controllability of a class of swarm signalling networks: impact of adding links," *in preparation*, 2021.

[10] A. Lombardi and M. Hörnquist, "Controllability analysis of networks," *Phys. Rev. E*, vol. 75, p. 056110, May 2007. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.75.056110

[11] 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. [Online]. Available: https://doi.org/10.1137/0301010

[12] C.-T. Lin, "Structural controllability," *IEEE Transactions on Automatic Control*, vol. 19, no. 3, pp. 201–208, 1974.

[13] Z. Yuan, C. Zhao, Z. Di, W.-X. Wang, and Y.-C. Lai, "Exact controllability of complex networks," *Nature Communications*, vol. 4, no. 1, Sep 2013. [Online]. Available: http://dx.doi.org/10.1038/ncomms3447

[14] 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. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.85.026115

[15] H. Lvlin, L. Songyang, B. Jiang, and B. Liang, "Enhancing complex network controllability by rewiring links," in *2013 Third International Conference on Intelligent System Design and Engineering Applications*, 2013, pp. 709–711.

[16] L.-L. Hou, S.-Y. Lao, G. Liu, and L. Bai, "Controllability and directionality in complex networks," *Chinese Physics Letters*, vol. 29, no. 10, p. 108901, oct 2012. [Online]. Available: https://doi.org/10.1088/0256-307x/29/10/108901

[17] Y.-D. Xiao, S.-Y. Lao, L.-L. Hou, and L. Bai, "Edge orientation for optimizing controllability of complex networks," *Phys. Rev. E*, vol. 90, p. 042804, Oct 2014. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.90.042804

[18] Y. Xiao, S. Lao, L. Hou, M. Small, and L. Bai, "Effects of edge directions on the structural controllability of complex networks," *PLOS ONE*, vol. 10, no. 8, pp. 1–15, 08 2015. [Online]. Available: https://doi.org/10.1371/journal.pone.0135282

[19] 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

[20] 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

[21] M. Pósfai, Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási, "Effect of correlations on network controllability," *Scientific reports*, vol. 3, no. 1, pp. 1–7, 2013.

[22] G. Menichetti, L. Dall'Asta, and G. Bianconi, "Network controllability is determined by the density of low in-degree and out-degree nodes," *Phys. Rev. Lett.*, vol. 113, p. 078701, Aug 2014. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.113.078701

[23] M. Komareji and R. Bouffanais, "Resilience and controllability of dynamic collective behaviors," *PLOS ONE*, vol. 8, no. 12, p. e82578, 2013. [Online]. Available: https://app.dimensions.ai/details/publication/pub.1030450849andhttps://journals.plos.org/plosone/article/file?id=10.1371/journal.pone.0082578&type=printable

[24] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, p. 167–256, Jan 2003. [Online]. Available: http://dx.doi.org/10.1137/S003614450342480

[25] 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.

[26] 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

[27] J.-J. E. Slotine and W. Li, *Applied nonlinear control*. Prentice hall Englewood Cliffs, NJ, 1991, vol. 199, no. 1.

[28] M. Hautus, "Controllability and observability conditions of linear autonomous systems," *Nederlandse Akademie van Wetenschappen. Proceedings. Series A. Indagationes Mathematicae*, vol. 72, 01 1969.

[29] 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

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

[31] 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, p. e89066, 2014.

[32] D. Y. Kenett, J. Gao, X. Huang, S. Shao, I. Vodenska, S. V. Buldyrev, G. Paul, H. E. Stanley, and S. Havlin, "Network of interdependent networks: overview of theory and applications," *Networks of Networks: The Last Frontier of Complexity*, pp. 3–36, 2014.

[33] S. Shao, X. Huang, H. E. Stanley, and S. Havlin, "Percolation of localized attack on complex networks," *New Journal of Physics*, vol. 17, no. 2, p. 023049, feb 2015. [Online]. Available: https://doi.org/10.1088/1367-2630/17/2/023049

A

APPENDIX A

This appendix shows the procedure of deducing Eq. 3.42 in detail. The out-degree distribution $P_{out}()$ for an unperturbed SSN is:

$$P_{out}(k_{out}) = \delta(k - k_{out}) \tag{A.1}$$

The expression for the degree distribution after removing m links randomly is expressed as:

$$Pr[D_G = i] = (1-p)^i \sum_{j=i}^{N-1} \binom{j}{i} p^{j-i} Pr[D_{G_0} = j] \tag{A.2}$$

Thus, the out-degree distribution of network after perturbations $\bar{P}_{out}()$ follows:

$$\bar{P}_{out}(k_{out}) = (1-p)^{k_{out}} \sum_{j=k_{out}}^{N-1} \binom{j}{k_{out}} p^{j-k_{out}} \delta(k-j) \tag{A.3}$$

After removing links, $k_{out} \leq k$, and

$$\bar{P}_{out}(k_{out}) = (1-p)^{k_{out}} \binom{k}{k_{out}} p^{k-k_{out}} \tag{A.4}$$

From the degree distribution, we can express the generating function G_{out} as:

$$\begin{aligned} \bar{G}_{out}(x) == & \sum_{k_{out}=0}^{\infty} \bar{P}_{out}(k_{out}) x^{k_{out}} \\ = & \sum_{k_{out}=0}^{k} (1-p)^{k_{out}} \binom{k}{k_{out}} p^{k-k_{out}} x^{k_{out}} \\ = & \sum_{k_{out}=0}^{k} \binom{k}{k_{out}} ((1-p)x)^{k_{out}} p^{k-k_{out}} \\ = & (p + (1-p)x)^k \end{aligned} \tag{A.5}$$

A

The in-degree distribution of unperturbed SSN $P_{in}()$ follows a Poisson distribution:

$$P_{in}(k_{in}) = \frac{k^{k_{in}} e^{-k}}{k_{in}!} \tag{A.6}$$

Following the Eq. A.2, the in-degree distribution of the perturbed SSN $\bar{P}_{in}()$ is expresses as:

$$\bar{P}_{in}(k_{in}) = (1-p)^k_{in} \sum_{j=k_{in}}^{\infty} \binom{j}{k_{in}} p^{j-k_{in}} \frac{k^j}{j!} e^{-k} \tag{A.7}$$

Then, the generating function of in-degree distribution for the perturbed SSN follows:

$$\begin{aligned}
\bar{G}_{in}(x) == & \sum_{k_{in}=0}^{\infty} \bar{P}_{in}(k_{in}) x^{k_{in}} \\
= & \sum_{k_{in}=0}^{\infty} (1-p)^{k_{in}} \sum_{j=k_{in}}^{\infty} \binom{j}{k_{in}} p^{j-k_{in}} \frac{k^j}{j!} e^{-k} x^{k_{in}} \\
= & e^{-k} \sum_{k_{in}=0}^{\infty} (\frac{(1-p)x}{p})^{k_{in}} \sum_{j=k_{in}}^{\infty} \binom{j}{k_{in}} \frac{(pk)^j}{j!} \\
= & e^{-k} \sum_{k_{in}=0}^{\infty} (\frac{(1-p)x}{p})^{k_{in}} \sum_{j=k_{in}}^{\infty} \frac{j!}{k_{in}!(j-k_{in})!} \frac{(pk)^j}{j!} \\
= & e^{-k} \sum_{k_{in}=0}^{\infty} (\frac{(1-p)x}{p})^{k_{in}} \sum_{j=k_{in}}^{\infty} \frac{(pk)^j}{k_{in}!(j-k_{in})!} \\
= & e^{-k} \sum_{k_{in}=0}^{\infty} (\frac{(1-p)x}{p})^{k_{in}} \frac{1}{k_{in}!} \sum_{j=k_{in}}^{\infty} \frac{(pk)^{j-k_{in}} (pk)^{k_{in}}}{(j-k_{in})!} \\
= & e^{-k} \sum_{k_{in}=0}^{\infty} (\frac{(1-p)x}{p})^{k_{in}} \frac{(pk)^{k_{in}}}{k_{in}!} \sum_{\hat{j}=0}^{\infty} \frac{(pk)^{\hat{j}}}{\hat{j}!} \\
= & e^{-k} \sum_{k_{in}=0}^{\infty} \frac{(k(1-p)x)^{k_{in}}}{k_{in}!} e^{pk} \\
= & e^{-k} e^{k(1-p)x} e^{pk} \\
= & e^{-k(1-p)(1-x)}
\end{aligned} \tag{A.8}$$

The generating functions of excess out-degree distribution $H_{out}()$ can be written as:

$$H_{out}(x) = \sum_{k_{out}=1}^{\infty} \frac{P_{out}(k_{out}) k_{out}}{< k_{out} >} x^{k_{out}-1} \tag{A.9}$$

After attacks, the generating function $\bar{H}_{out}(x)$ of SSN is given by:

$$\begin{aligned}
\bar{H}_{out}(x) &= \sum_{k_{out}=1}^{\infty} \frac{\bar{P}_{out}(k_{out})k_{out}}{<k_{out}>} x^{k_{out}-1} \\
&= \sum_{k_{out}=1}^{k} \frac{(1-p)^{k_{out}} \binom{k}{k_{out}} p^{k-k_{out}} k_{out}}{k(1-p)} x^{k_{out}-1} \\
&= \sum_{k_{out}=1}^{k} \binom{k-1}{k_{out}-1} p^{k-k_{out}} ((1-p)x)^{k_{out}-1} \\
&\overset{m=k_{out}-1}{=} \sum_{m=0}^{k-1} \binom{k-1}{m} p^{k-1-m} ((1-p)x)^{m} \\
&= (p+(1-p)x)^{k-1}
\end{aligned} \tag{A.10}$$

The same procedure for deducing the generating function $\bar{H}_{in}(x)$:

$$\begin{aligned}
\bar{H}_{in}(x) &= \sum_{k_{in}=1}^{\infty} \frac{\bar{P}_{in}(k_{in})k_{in}}{<k_{in}>} x^{k_{in}-1} \\
&= \sum_{k_{in}=1}^{\infty} \frac{k_{in}(1-p)^{k_{in}}}{k(1-p)} \sum_{j=k_{in}}^{\infty} \binom{j}{k_{in}} \frac{k^j}{j!} e^{-k} p^{j-k_{in}} x^{k_{in}-1} \\
&= e^{-k} \sum_{k_{in}=1}^{\infty} \frac{k_{in}(1-p)^{k_{in}}}{k(1-p)} \sum_{j=k_{in}}^{\infty} \frac{j!}{k_{in}!(j-k_{in})!} \frac{p^{j-k_{in}} k^j}{j!} x^{k_{in}-1} \\
&= e^{-k} \sum_{k_{in}=1}^{\infty} \frac{k_{in}(1-p)^{k_{in}}}{k(1-p)} \sum_{j=k_{in}}^{\infty} \frac{k^{k_{in}}}{k_{in}!} \frac{(kp)^{j-k_{in}}}{(j-k_{in})!} \frac{x^{k_i n}}{x} \\
&= e^{-k} \sum_{k_{in}=1}^{\infty} \frac{k_{in}(1-p)^{k_{in}}}{k(1-p)} \frac{k^{k_{in}}}{k_{in}!} e^{pk} \frac{x^{k_{in}}}{x} \\
&= e^{-k} \sum_{k_{in}=1}^{\infty} \frac{k_{in}(kx(1-p))^{k_{in}}}{xk(1-p)k_{in}!} e^{pk} \\
&= e^{-k+pk} \sum_{k_{in}=1}^{\infty} \frac{(kx(1-p))^{k_{in}-1}}{(k_{in}-1)!} \\
&\overset{m=k_{in}-1}{=} e^{-k+pk} \sum_{m=0}^{\infty} \frac{(kx(1-p))^{m}}{m!} \\
&= e^{-k(1-p)(1-x)}
\end{aligned} \tag{A.11}$$

Thus, the set of equations Eq. 3.22-Eq. 3.25 becomes:

$$\omega_1 = (p+(1-p)\hat{\omega}_2)^{k-1} \tag{A.12}$$

$$\omega_2 = 1-(p+(1-p)(1-\hat{\omega}_1))^{k-1} \tag{A.13}$$

$$\hat{\omega}_1 = e^{-k(1-p)(1-\omega_2)} \tag{A.14}$$

$$\hat{\omega}_2 = 1-e^{-k(1-p)\omega_1} \tag{A.15}$$

A

By setting $\hat{\omega}_2 = 1 - \hat{\omega}_1$ and $\omega_2 = 1 - \omega_1$, the pair of Eq. A.12 and Eq. A.15 is equivalent to the pair of Eq. A.13 and Eq. A.14. Then, the n_D in Eq. 3.13 follows:

$$\begin{aligned} n_D &= \bar{G}_{out}(1-\hat{\omega}_1) + \bar{G}_{in}(\omega_2) - 1 + \bar{k}\hat{\omega}_1(1-\omega_2) \\ &= (p + (1-p)(1 - e^{-k(1-p)(1-\omega_2)}))^k - 1 + e^{-k(1-p)(1-\omega_2)} + k(1-p)(1-\omega_2)e^{-k(1-p)(1-\omega_2)} \end{aligned} \tag{A.16}$$

where ω_2 satisfies:

$$1 - \omega_2 = (p + (1-p)(1 - e^{-k(1-p)(1-\omega_2)}))^{k-1} \tag{A.17}$$

B

APPENDIX B

Here we deduce the Eq. 3.51 in detail. The degree distribution $Pr[D_G = i]$ of SSN after randomly adding links can be expressed as:

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

The out-degree distribution for the perturbed SSN $\bar{P}_{out}()$ follows:

$$\begin{aligned} \bar{P}_{out}(k_{out}) &= (1-f)^{N-1-k_{out}} \sum_{j=0}^{k_{out}} \binom{N-1-j}{k_{out}-j} f^{k_{out}-j} \delta(k-j) \\ &= (1-f)^{N-1-k_{out}} \binom{N-1-k}{k_{out}-k} f^{k_{out}-k} \end{aligned} \tag{B.2}$$

if $k_{out} \geq k$. From the above out-degree distribution, we can get the generating function of the out-degree distribution:

$$\begin{aligned} \bar{G}_{out}(x) &= \sum_{k_{out}=0}^{\infty} \bar{P}_{out}(k_{out}) x^{k_{out}} \\ &= \sum_{k_{out}=k}^{N-1} (1-f)^{N-1-k_{out}} \binom{N-1-k}{k_{out}-k} f^{k_{out}-k} x^{k_{out}} \\ &\overset{i=k_{out}-k}{=} \sum_{i=0}^{N-1-k} (1-f)^{N-1-k-i} \binom{N-1-k}{i} f^{i} x^{k+i} \\ &= \sum_{i=0}^{N-1-k} x^{k} \binom{N-1-k}{i} (fx)^{i} (1-f)^{N-1-k-i} \\ &= x^{k}(fx+1-f)^{N-1-k} \\ &= x^{k}(1-f(1-x))^{N-1-k} \end{aligned} \tag{B.3}$$

B

The in-degree distribution of SSN with N nodes and with average in-degree of k follows a binomial distribution. Thus the in-degree distribution of the perturbed SSN $\bar{P}_{in}()$ follows:

$$\bar{P}_{in}(k_{in}) = (1-f)^{N-1-k_{in}} \sum_{j=0}^{k_{in}} \binom{N-1-j}{k_{in}-j} f^{k_{in}-j} \binom{N-1}{j} p^j (1-p)^{N-1-j} \tag{B.4}$$

Then, the generating function of in-degree distribution can be expressed:

$$\begin{aligned}
\bar{G}_{in}(x) &= \sum_{k_{in}=0}^{\infty} \bar{P}_{in}(k_{in}) x^{k_{in}} \\
&= \sum_{k_{in}=0}^{N-1} (1-f)^{N-1-k_{in}} \sum_{j=0}^{k_{in}} \binom{N-1-j}{k_{in}-j} f^{k_{in}-j} \binom{N-1}{j} p^j (1-p)^{N-1-j} x^{k_{in}} \\
&= \sum_{k_{in}=0}^{N-1} (1-f)^{N-1-k_{in}} (fx)^{k_{in}} \sum_{j=0}^{k_{in}} \frac{(N-1-j)!}{(k_{in}-j)!(N-1-k_{in})!} \frac{(N-1)!}{(N-1-j)!j!} (\frac{p}{j})^j (1-p)^{N-1-j} \\
&= \sum_{k_{in}=0}^{N-1} (1-f)^{N-1-k_{in}} (fx)^{k_{in}} \sum_{j=0}^{k_{in}} \frac{(N-1)!}{(N-1-k_{in})!k_{in}!} \frac{k_{in}!}{(k_{in}-j)!j!} (\frac{p}{f})^j (1-p)^{N-1-j+k_{in}-k_{in}} \\
&= \sum_{k_{in}=0}^{N-1} \binom{N-1}{k_{in}} ((1-f)(1-p))^{N-1-k_{in}} (fx)^{k_{in}} \sum_{j=0}^{k_{in}} \binom{k_{in}}{j} (\frac{p}{f})^j (1-p)^{k_{in}-j} \\
&= \sum_{k_{in}=0}^{N-1} \binom{N-1}{k_{in}} ((1-f)(1-p))^{N-1-k_{in}} (fx)^{k_{in}} (\frac{p}{f}+1-p)^{k_{in}} \\
&= \sum_{k_{in}=0}^{N-1} \binom{N-1}{k_{in}} ((1-f)(1-p))^{N-1-k_{in}} (x(p+f(1-p)))^{k_{in}} \\
&= ((1-f)(1-p)+x(p+f(1-p)))^{N-1} \\
&= (1-f(1-x)-(1-f)p(1-x))^{N-1} \\
&\overset{p=\frac{k}{N-1}}{=} (1-f(1-x)-(1-f)\frac{k}{N-1}(1-x))^{N-1} \\
&= (1-f(1-x)-(1-f)\frac{\bar{k}-f(N-1-k)}{N-1}(1-x))^{N-1} \\
&= (1-\frac{(1-x)\bar{k}}{N-1})^{N-1} \\
&\overset{N\to\infty}{=} e^{-\bar{k}(1-x)}
\end{aligned} \tag{B.5}$$

where $p = \frac{k}{N-1}$ and $\bar{k} = k + f(N-1-k)$.
The generating function of excess out-degree distribution of the perturbed SSN is:

$$\bar{H}_{out}(x) = \sum_{k_{out}=1}^{\infty} \frac{\bar{P}_{out}(k_{out}) k_{out}}{< k_{out} >} x^{k_{out}-1} \tag{B.6}$$

$$= \sum_{k_{out}=k}^{N-1} \frac{k_{out}(1-f)^{N-1-k_{out}} \binom{N-1-k}{k_{out}-k} f^{k_{out}-k}}{\bar{k}} x^{k_{out}-1} \tag{B.7}$$

$$\overset{i=k_{out}-k}{=} \sum_{i=0}^{N-1-k} \frac{(k+i)(1-f)^{N-1-k-i} \binom{N-1-k}{i} f^{i}}{\bar{k}} x^{k+i-1} \tag{B.8}$$

$$= \frac{x^{k-1}}{\bar{k}} \sum_{i=0}^{N-1-k} (k+i) \binom{N-1-k}{i} (fx)^{i} (1-f)^{N-1-k-i} \tag{B.9}$$

$$= \frac{x^{k-1}}{\bar{k}} (k(fx+1-f)^{N-1-k} + f(N-1-k)x(fx+1-f)^{N-2-k}) \tag{B.10}$$

$$= \frac{x^{k-1}}{\bar{k}} (kfx+k-kf+Nfx-fx-kfx)(fx+1-f)^{N-2-k} \tag{B.11}$$

$$= \frac{x^{k-1}}{\bar{k}} (\bar{k} - f(N-1)(1-x))(1-f(1-x))^{N-2-k} \tag{B.12}$$

The same procedure for deducing the generating function $\bar{H}_{in}(x)$ of the perturbed SSN:

$$\begin{aligned}
\bar{H}_{in}(x) &= \sum_{k_{in}=1}^{\infty} \frac{\bar{P}_{in}(k_{in})k_{in}}{< k_{in} >} x^{k_{in}-1} \\
&= \frac{1}{< k_{in} >} \sum_{k_{in}=1}^{N-1} (1-f)^{N-1-k_{in}} \sum_{j=0}^{k_{in}} \binom{N-1-j}{k_{in}-j} f^{k_{in}-j} \binom{N-1}{j} p^{j} (1-p)^{N-1-j} x^{k_{in}-1} \\
&= \frac{1}{\bar{k}x} \sum_{k_{in}=1}^{N-1} k_{in} \binom{N-1}{k_{in}} ((1-f)(1-p))^{N-1-k_{in}} (fx)^{k_{in}} \sum_{j=0}^{k_{in}} \binom{k_{in}}{j} (\frac{p}{f})^{j} (1-p)^{k_{in}-j} \\
&= \frac{1}{\bar{k}x} \sum_{k_{in}=1}^{N-1} k_{in} \binom{N-1}{k_{in}} ((1-f)(1-p))^{N-1-k_{in}} (fx)^{k_{in}} (\frac{p}{f}+1-p)^{k_{in}} \\
&= \frac{1}{\bar{k}x} \sum_{k_{in}=1}^{N-1} k_{in} \binom{N-1}{k_{in}} ((1-f)(1-p))^{N-1-k_{in}} (x(p+f(1-p)))^{k_{in}} \\
&= \frac{1}{\bar{k}x} (N-1)(x(p+f(1-p)))(x(p+f(1-p))+(1-f)(1-p))^{N-2} \\
&= \frac{(N-1)(p+f(1-p))}{\bar{k}} (1-(p+f(1-p))(1-x))^{N-2} \\
&\overset{p=\frac{k}{N-1}}{=} \frac{(N-1)(\frac{k}{N-1}+\frac{f(N-1-k)}{N-1})}{\bar{k}} (1-(\frac{k}{N-1}+\frac{f(N-1-k)}{N-1})(1-x))^{N-2} \\
&= (1-\frac{\bar{k}}{N-1}(1-x))^{N-2} \\
&\overset{N\to\infty}{=} e^{-\bar{k}(1-x)}
\end{aligned} \tag{B.13}$$

B

Thus, the set of equations Eq. 3.22-Eq. 3.25 follows:

$$\omega_1 = \frac{\hat{\omega}_2^{k-1}}{\bar{k}}(\bar{k} - f(N-1)(1-\hat{\omega}_2))(1-f(1-\hat{\omega}_2))^{N-2-k} \tag{B.14}$$

$$\omega_2 = 1 - \frac{(1-\hat{\omega}_1)^{k-1}}{\bar{k}}(\bar{k} - f(N-1)\hat{\omega}_1)(1-f\hat{\omega}_1)^{N-2-k} \tag{B.15}$$

$$\hat{\omega}_1 = e^{-\bar{k}(1-\omega_2)} \tag{B.16}$$

$$\hat{\omega}_2 = 1 - e^{-\bar{k}\omega_1} \tag{B.17}$$

By setting $\hat{\omega}_2 = 1-\hat{\omega}_1$ and $\omega_2 = 1-\omega_1$, the n_D in Eq. 3.13 follows:

$$n_D = \bar{G}_{out}(1-\hat{\omega}_1) + \bar{G}_{in}(\omega_2) - 1 + \bar{k}\hat{\omega}_1(1-\omega_2) \tag{B.18}$$

$$= e^{-\bar{k}(1-\omega_2)} + (1-e^{-\bar{k}(1-\omega_2)})^k(1-fe^{-\bar{k}(1-\omega_2)})^{N-1-k} - 1 + \bar{k}(1-\omega_2)e^{-\bar{k}(1-\omega_2)} \tag{B.19}$$

where ω_2 is the solution of the equation:

$$\bar{k}(1-\omega_2) = (1-e^{-\bar{k}(1-\omega_2)})^{k-1}(\bar{k} - f(N-1)e^{-\bar{k}(1-\omega_2)})(1-fe^{-\bar{k}(1-\omega_2)})^{N-2-k} \tag{B.20}$$