Modelling and Analysis of Complex Networks
Robustness, Recoverability and Spreading

Wang, F.

DOI
10.4233/uuid:7f4159a0-b5e8-491b-b608-145766f29f18

Publication date
2024

Document Version
Final published version

Citation (APA)
Wang, F. (2024). *Modelling and Analysis of Complex Networks: Robustness, Recoverability and Spreading*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:7f4159a0-b5e8-491b-b608-145766f29f18

MODELLING AND ANALYSIS OF COMPLEX NETWORKS

ROBUSTNESS, RECOVERABILITY AND SPREADING

MODELLING AND ANALYSIS OF COMPLEX NETWORKS

ROBUSTNESS, RECOVERABILITY AND SPREADING

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology,
by the authority of the Rector Magnificus, Prof.dr.ir. T.H.J.J. van der Hagen,
chair of the Board for Doctorates,
to be defended publicly on
Wednesday 25 September 2024 at 15:00 o'clock

by

Fenghua WANG

Master of Science in System Theory,
Beijing Normal University, Beijing, China,
born in Hubei, China.

This dissertation has been approved by promotors.

Composition of the doctoral committee:

Rector Magnificus	chairperson
Prof.dr.ir. R.E. Kooij	Delft University of Technology, promotor
Prof.dr.ir. P.F.A. Van Mieghem	Delft University of Technology, promotor
Independent members:	
Prof.dr. H. Cherifi	University of Burgundy, France
Dr. J.L.A. Dubbeldam	Delft University of Technology
Prof.dr. N.V. Litvak	University of Twente & Eindhoven University of Technology
Prof.dr. J.L. Marzo Lazaro	University of Girona, Spain
Prof.dr. M.E. Warnier	Delft University of Technology
Prof.dr.ir. G.N. Gaydadjiev	Delft University of Technology, reserve member

Keywords: Complex Networks, Network Robustness, Network Reliability, Network recoverability, Spreading

Printed by: ProefschriftMaken

Front & Back: Designed by Fenghua Wang. Original image from shutterstock.com

Published by: Fenghua Wang

Email: 289676733@qq.com

ISBN 978-94-6366-906-1

An electronic version of this dissertation is available at
http://repository.tudelft.nl/.

To my family

CONTENTS

SUMMARY

Failures of systems are ubiquitous, such as the blackouts of electrical systems and road disruptions in transport systems. Understanding the properties of systems can benefit the control of the system and the design of robust systems. Complex networks are widely used to model complex systems. In this dissertation, we explore the performance of complex networks under link or node failures, provide strategies to recover the networks and discover the behaviors of processes spreading on networks. This dissertation comprises three parts, containing seven chapters, including the introduction chapter, five chapters discerning the research contributions, and a conclusion chapter which summarizes the main contributions and gives directions for future work.

Part I of the thesis deals with network robustness and reliability and contains two chapters. In Chapter 2, we investigate the robustness of network controllability, calculated by the minimum fraction of driver nodes required to fully control the networks. Under random and targeted node removals based on in-degree, out-degree, and total degree, we develop analytical methods using generating functions of the in-degree and out-degree distributions to approximate network controllability. Validated on synthetic and real-world networks, we find that analytical methods work well under random node removals and reasonably well for different types of targeted node removals. In Chapter 3, assuming each link has a given operational probability and nodes are always operational, we define controller reachability, which is the probability that each node can reach at least one controller. Using real-world networks, we explore how performance changes as the number of controllers increases from one to five. We propose four placement strategies to place controllers with a fixed number of controllers: the placement strategy based on graph metrics, the greedy placement strategy, and two genetic placement strategies. The placement strategies demonstrate good performance on real-world data.

After a network is degraded, recovering the system and measuring its recovery performance are essential. Therefore, Part II of this thesis studies network recoverability. This part contains two chapters. In Chapter 4, we explore how to approximate the network controllability in case of a random recovery process and how to recover the network controllability efficiently. We propose an analytical method to approximate network controllability under random node additions. The outcomes of our method closely align with numerical simulation results for both synthetic and real-world networks. Furthermore, to recover network controllability efficiently, we find that greedy strategies outperform recovery strategies based on degree centrality or betweenness centrality as well as the random recovery strategy. Chapter 5 investigates recovery strategies after random link failures on power grids. To explore the effectiveness of recovering failed links based solely on topological network metrics, we consider 13 recovery strategies. To evaluate the performance of these proposed strategies, we conducted simulations on three distinct power systems: the IEEE 30, IEEE 39, and IEEE 118 systems. Our results conclude that relying solely on a single metric to develop a recovery strategy is insufficient to restore

power grids after link failures. For comparison, recovery strategies that employ greedy algorithms are more effective choices.

Finally, Part III looks into spreading processes on networks. Chapter 6 focuses on exploring the time-dependent behaviors of the ϵ−SIS model. We find that the prevalence (the average fraction of infected nodes) as a function of time typically exhibits a "two-plateau" behavior. Furthermore, the time-dependent mean-field approximation for the complete graph performs reasonably well for relatively large self-infection rates but completely fails to mimic the typical behavior with small self-infection rates. The observations may explain why absorbing processes are hardly observed in reality, even over long intervals, because of the ignorance of the interplay of nodal self-infection with small self-infection rates and spread over links. Chapter 7 concludes the thesis by summarizing the main contributions and gives directions for future research.

SAMENVATTING

Uitval en storingen van systemen zijn alomtegenwoordig. Voorbeelden hiervan zijn stroomuitval van elektrische systemen en vertraging van verkeer binnen transportsystemen. Het begrijpen van de eigenschappen van systemen kan bijdragen aan het beheren van de systemen en het ontwerpen van robuuste systemen. Complexe netwerken worden veel gebruikt om complexe systemen te modelleren. In dit proefschrift onderzoeken we de prestaties van complexe netwerken bij uitval van verbindingen of knooppunten binnen de netwerken. We ontwikkelen strategieën om de netwerken te herstellen en beschrijven het gedrag van processen die zich verspreiden op netwerken. Dit proefschrift bestaat uit drie delen, met in totaal zeven hoofdstukken, waaronder het inleidende hoofdstuk, vijf hoofdstukken die de onderzoeksbijdragen bevatten, en een afsluitend hoofdstuk dat de belangrijkste bijdragen samenvat en duiding geeft voor toekomstig onderzoek.

Deel I van het proefschrift behandelt robuustheid en betrouwbaarheid van netwerken en bevat twee hoofdstukken. In Hoofdstuk 2 onderzoeken we de robuustheid van netwerkcontrole, berekend door de minimale fractie sturende knooppunten die nodig zijn om de netwerken volledig te controleren. Voor willekeurige en gerichte verwijdering van knooppunten, ontwikkelen we analytische methoden, gebaseerd opgenererende functies van de verdelingen van in-, uit- en totale graad van de knooppunten, om de mate waarin het netwerk controleerbaar is, te benaderen. Middels validatie op synthetische en echte netwerken blijken analytische methoden goed te werken bij willekeurige verwijderingen van knooppunten en redelijk goed voor verschillende soorten gerichte verwijderingen van knooppunten. In Hoofdstuk 3 definiëren we, ervan uitgaande dat elke verbinding met een gegeven waarschijnlijkheid operationeel is en knooppunten altijd operationeel zijn, de controller-bereikbaarheid. Dit is de waarschijnlijkheid dat elk knooppunt ten minste één controller kan bereiken. Door analyse van echte netwerken onderzoeken we hoe de controller-bereikbaarheid verandert naarmate het aantal controllers toeneemt van één tot vijf. We stellen vier strategieën voor om controllers te plaatsen met een vast aantal controllers: de plaatsingsstrategie op basis van netwerk-metrieken, een strategie gebaseerd op greedy plaatsing van controllers en twee strategieën gebaseerd op genetische algoritmes. Validatie op echte netwerken laat zien dat de plaatsingsstrategieën goed werken.

Nadat een netwerk is gedegradeerd, zijn het herstellen van het systeem en het meten van de mate van herstel van het netwerk essentieel. Daarom onderzoekt Deel II van dit proefschrift de herstelbaarheid van netwerken. Dit deel bevat twee hoofdstukken. In Hoofdstuk 4 onderzoeken we hoe we de controleerbaarheid van netwerken kunnen benaderen in geval van een willekeurig herstelproces en hoe we de controleerbaarheid van netwerken efficiënt kunnen herstellen. We stellen een analytische methode voor om de controleerbaarheid van netwerken te benaderen onder willekeurige toevoeging van knooppunten. De resultaten van onze methode sluiten nauw aan bij numerieke simulatieresultaten voor zowel synthetische als echte netwerken. Bovendien blijkt dat

greedy strategieën beter presteren bij het efficiënt herstellen van de controleerbaarheid van netwerken dan herstelstrategieën op basis van netwerk-metrieken zoals graad en betweenness, evenals de willekeurige herstelstrategie. Hoofdstuk 5 onderzoekt herstelstrategieën voor elektriciteitsnetten waar een aantal verbindingen willekeurig zijn verwijderd. Om de effectiviteit van het herstellen van de verwijderde verbindingen te verkennen op basis van alleen topologische netwerk-metrieken, beschouwen we 13 herstelstrategieën. Om de prestaties van deze voorgestelde strategieën te evalueren, voeren we simulaties uit op drie verschillende elektriciteitssystemen: de zogenaamde IEEE 30, IEEE 39 en IEEE 118 systemen. Onze resultaten tonen aan dat herstelstrategieën die gebaseerd zijn op slechts concluderen dat het vertrouwen op slechts één netwerk-metriek, niet in staat zijn om elektriciteitsnetten efficient te herstellen na willekeurig verwijderde verbindingen. Ter vergelijking, herstelstrategieën die gebruik maken van greedy gretige algoritmen, zijn effectiever.

Tot slot gaat Deel III in op verspreidingsprocessen op netwerken. Hoofdstuk 6 richt zich op het verkennen van de tijdafhankelijke gedragingen van het $\epsilon-$SIS model. We ontdekken dat de prevalentie (het gemiddelde fractie geïnfecteerde knooppunten) als functie van de tijd meestal een "twee-plateau" gedrag vertoont. Bovendien presteert de tijdsafhankelijke mean-field benadering voor de volledige graaf redelijk goed voor relatief grote zelfbesmettingspercentages, maar slaagt er niet in om het typische gedrag voor lage zelfbesmettingspercentages na te beschrijven. De observaties kunnen verklaren waarom absorberende processen nauwelijks worden waargenomen in de realiteit, zelfs over lange intervallen, vanwege het negeren van de wisselwerking tussen zelfbesmetting van knooppunten met lage besmettingspercentages en verspreiding via links. Hoofdstuk 7 sluit het proefschrift af door de belangrijkste bijdragen samen te vatten en richtingen aan te geven voor toekomstig onderzoek.

1

INTRODUCTION

Complex systems consist of interacting components that exhibit collective behaviors, often referred to as "emergent" behaviors, which go beyond the sum of individual behaviors [1]. These systems are ubiquitous, finding applications in diverse fields such as biology [2], transportation [3], finance [4], and more. Understanding the workings of complex systems is crucial for designing and controlling them. For instance, in transportation systems, a comprehensive understanding enables the development of robust transportation systems, facilitating the efficient movement of people from one location to another with reduced time and cost. Network science serves as a fundamental tool for modeling and analyzing complex systems, employing graphs to depict the topology of interconnected components [5].

Network science has its roots in graph theory, a prominent subfield of discrete mathematics [6]. A historically significant problem in graph theory is the Seven Bridges of Königsberg problem, posed by Euler in 1736 [7]: determining how to traverse each of the seven connected bridges in the city of Königsberg exactly once. Other renowned problems in graph theory encompass the coloring problem [8], the shortest-path problem [9], and more. Since the 1950s, graph theory has found applications in topology analysis in various fields such as sociology [10], engineering [11], and beyond. In comparison to graph theory, Newman *et al.* assert that network science introduces three key distinctions [12]. Firstly, network science delves into understanding the properties of real systems by utilizing real-world data, incorporating both theoretical and empirical perspectives. Secondly, networks in the context of network science may exhibit time-varying characteristics. Lastly, research in network science encompasses not only topological properties but also dynamic aspects of networks. Similarly, Van Mieghem defines a network as comprising both a topology part and a service part [13]. The topology part can be modeled as a graph, encompassing nodes (vertices) and links (edges), while the service part, for instance, involves utilizing the network topology to transport items between a group of nodes with specific constraints.

In this thesis, we have explored three topics in network science: network robustness and reliability, network recoverability, and spreading on networks.

1

1.1. NETWORK ROBUSTNESS AND RELIABILITY

Network robustness refers to the capacity of a network to maintain its functionality despite failures in certain parts of the network during a specified time interval [13][14]. Scott *et al.* classify measures for quantifying network robustness into three main categories: measures based on graph connectivity, such as binary connectivity [15], node connectivity [15], and edge connectivity [15]; measures based on the adjacency matrix spectrum, including spectral radius [16], spectral gap [17], and natural connectivity [18]; and measures based on the Laplacian matrix spectrum, such as algebraic connectivity [19], the number of spanning trees [20], and effective resistance [15], etc. [21]. Lou *et al.* present a general classification of network robustness into two types: a priori and a posteriori measures [22]. A priori measures can be determined without conducting attack simulations, such as binary connectivity, while a posteriori measures are computed through a sequence of values during an iterative process, as in the case of node or link attacks. Additionally, they have categorized a posteriori structural network robustness metrics, based on network functionality, into three main classes: connectivity metrics, like the relative size of the largest connected component [23]; controllability metrics, such as minimum number of driver nodes [24][25]; and communication ability metrics, like communicable node pairs [26]. Various research efforts have explored attack strategies, including targeted node or link removals [27], defense strategies like edge rewiring [17], and predictions of network robustness under attacks [28]. Due to the limited research on approximations of network controllability under node removals, our dissertation addresses this gap by conducting relevant research.

Controller placement problems discuss where to place controllers and how many controllers are required to reach the expected performance with or without constraints like costs, which is a hot topic in communication networks [29]. Network connectivity is a widely used metric for measuring the performance of communication networks. All connected components ensure that messages can be transmitted between any pair of nodes [13]. For networks with unreliable components, network reliability is the probability that a network remains connected with nodes or links assigned a probability of failure (survival) [30][31]. Based on the concept of network reliability, we defined the notion of controller reachability, which is the probability that all nodes can reach at least one controller to ensure the transmission of data from nodes to controllers. Using controller reachability, we investigate controller placement problems in networked systems.

1.2. NETWORK RECOVERABILITY

In the realm of network studies, network recoverability is defined as the ability of a network to return to a desired state after disruption [32]. An efficient and effective recovery strategy can enhance network recoverability, and researchers have explored and developed recovery strategies in various types of networks using different performance metrics [33]. Pan *et al.* found that different recovery strategies perform differently with respect to various performance metrics in weighted networks after links are removed [34]. Targeted recovery strategies, such as the strategy based on the targeted reinforcement path method in economic systems [35], the hybrid recovery strategy based on local betweenness and local closeness [36], and the recovery strategies against cascading failures

in interdependent networks [37], have been proposed. Heuristic recovery strategies, like greedy recovery strategies, have been found to be more efficient than metric-based strategies in recovering links. This efficiency is observed concerning indicators such as network efficiency, effective graph resistance [38], and network controllability [28]. After investigating the robustness of network controllability, we further explored research on the network recoverability of network controllability after nodes are removed.

In the context of power systems, the power grid can be represented as a complex network, where nodes represent generators, loads, and transformers, and links represent transmission lines. Numerous studies have focused on strategies for power grid recovery [39], such as black-start procedures [40], including an optimal generator start-up strategy solved through a mixed-integer linear programming (MILP) problem [41]. Machine learning methods, specifically reinforcement learning, have been developed for restoring networks after both node failures [42] and link failures [43]. Despite extensive research on power grid recovery, there is still a lack of investigation into the effectiveness of recovery strategies that rely solely on different network metrics following transmission line failures.

1.3. SPREADING ON NETWORKS

Modeling and analyzing the spreading process is a longstanding research area across various fields, including biology [44], mathematics [45], social science [46], and engineering [47]. For instance, failures spreading in transportation or power grid systems can be effectively modeled. Spreading models often consist of multiple compartments [48]. Two common compartments are susceptible (S) and infected (I). In the elementary compartmental models on the network, the susceptible-infected-susceptible (SIS) model is widely used. In the SIS model, infected nodes can infect susceptible neighboring nodes with an infection rate and can recover with the recovery rate. The steady state of the SIS model is characterized by all nodes being susceptible (healthy). To compare steady states approximated by different mean-field models of the SIS model, Li *et al.* introduced the $\epsilon-$SIS model as a benchmark model [49]. The key distinction is that a susceptible node can also be infected with its self-infection rate ϵ. This model has been applied to simulate the spreading of emotions on social networks [50]. Van Mighem [51] investigated the $\epsilon-$SIS model with an arbitrarily small but nonzero self-infection rate and found that the corresponding model exhibits an explosive phase transition. To gain a better understanding of the $\epsilon-$SIS model, we are particularly interested in the time dependence of the average fraction of infected nodes.

1.4. RESEARCH QUESTIONS

In this thesis, to better model, understand and control networked systems, we investigate research questions related to network robustness and reliability, network recoverability and spreading on networks. The research questions are defined as follows.

Using network controllability as the performance metric, we explore how network controllability evolves during random node removals and targeted node removals based on in-degree, out-degree, and total degree. Could we approximate the network controllability using the generating function of in-degree and out-degree distributions of networks

under different kinds of node removals? (Chapter 2)

Network reliability is a way to measure the performance of the networked system. Controller reachability is based upon the concept of network reliability, and we use controller reachability as a performance metric to explore controller placement problems. How many controllers does a network need to reach the expected performance? How should we place controllers on a network? (Chapter 3)

After node removals, we want to investigate how network controllability changes under random node additions. Could we approximate the network controllability during random node additions? Which recovery strategies are the most efficient among the recovery strategies considered? (Chapter 4)

Using the optimal power flow model, we would like to investigate how to recover power grids after randomly removing a fraction of links. Can we find an effective recovery strategy only using link centrality information? How does the heuristic recovery strategy perform? (Chapter 5)

The prevalence is the average fraction of infected nodes in the $\epsilon-$SIS process. What kind of time-dependent behavior does the $\epsilon-$SIS model have? (Chapter 6)

1.5. STRUCTURE OF THE THESIS

The thesis includes three parts and seven chapters: network robustness and reliability, network recoverability and spreading process on networks.

1.5.1. PART I: NETWORK ROBUSTNESS AND RELIABILITY

Chapter 2 introduces the methods based on generating functions of in-degree and out-degree distributions to approximate network controllability under random and targeted node removals.

Chapter 3 explores the network controller problem using controller reachability as a performance metric and develops different methods to place controllers.

1.5.2. PART II: NETWORK RECOVERABILITY

Chapter 4 investigates how to approximate network controllability using generating functions of in-degree and out-degree distributions under random node additions. In addition, the efficiency of different recovery strategies is assessed using the network recoverability indicator.

Chapter 5 discusses the performances of recovery strategies based on graph metrics and heuristic strategies to recover power grids using the optimal power flow model after links are randomly removed.

1.5.3. PART III: SPREADING PROCESS ON NETWORKS

Chapter 6 investigates the time-dependent behaviors of the Markovian $\epsilon-$SIS model on networks.

Chapter 7 summarizes the main contributions of the dissertation and outlines future directions for research.

1.6. PUBLICATION RELATED TO THIS THESIS

We present the published papers related to this dissertation as follows.

6. **F. Wang** and R. E. Kooij, *Robustness of Network Controllability with Respect to Node Removals*, In International Conference on Complex Networks and Their Applications pp. 383-394. Cham: Springer International Publishing, 2022. [**Chapter 2**]

5. **F. Wang** and R. E. Kooij, *Robustness of Network Controllability with Respect to Node Removals Based on In-Degree and Out-Degree*, Entroy **25**, no. 4 (2023): 656. [**Chapter 2**]

4. R. Xu, **F. Wang** and R. E. Kooij, *Controller placement with respect to controller reachability*, 7th International Conference on System Reliability and Safety (ICSRS 2023), 22-24 November, Bologna, Italy. [**Chapter 3**]

3. **F. Wang** and R. E. Kooij, *The recoverability of network controllability with respect to node additions*, New Journal of Physics **25**, no. 10 (2023): 103034. [**Chapter 4**]

2. **F. Wang**, H. Cetinay, Z. He, L. Liu, P. Van Mieghem and R. E. Kooij, *Recovering Power Grids Using Strategies Based on Network Metrics and Greedy Algorithms*, Entropy **25**, no. 10 (2023): 1455. [**Chapter 5**]

1. P. Van Mieghem and **F. Wang**, *Time dependence of susceptible-infected-susceptible epidemics on networks with nodal self-infections*, Physical Review E **101**, no. 5 (2020): 052310. [**Chapter 6**]

PART I: NETWORK ROBUSTNESS AND RELIABILITY

2

ROBUSTNESS OF NETWORK CONTROLLABILITY WITH RESPECT TO NODE REMOVALS

We developed methods based on generating functions for the in- and out-degree distributions to calculate the minimum number of driver nodes needed to control directed networks under random and targeted node removals. By validating the proposed methods on synthetic and real-world networks, we show that our methods work very well in the case of random node removals and reasonably well in the case of targeted node removals, particularly for moderate fractions of attacked nodes.

This chapter is based on the published papers [52] and [53].

2.1. INTRODUCTION

Network controllability has been investigated for different kinds of networks, like biological networks [54], transportation networks [55] and corruption networks [56]. A network is controllable if the states of nodes can be steered to any expected states in a finite time by imposing external inputs to some of the nodes. Kalman's controllability rank condition is used to judge whether a linear system is controllable or not [57]. However, sometimes, we do not know the weighted interactions within the network, which describe the strength with which a node affects other nodes. To overcome the issue, the concept of structural controllability has been proposed [58]. The interaction matrix and input matrix of the linear time-invariant system are structural if their elements are independently free parameters or some are fixed zeros. The system is called structurally controllable if it is possible to find values of structural interaction and input matrices to make the system satisfy the usual controllability condition. Besides investigating the necessary and sufficient conditions to make the specific system strongly structurally controllable [59], another research direction is to find the minimum set of inputs to make the system fully controllable [60]. Liu *et al.* [24] reduce the structural controllability problem to the optimization problem of finding a set of unmatched nodes in a maximum matching of the network. The nodes where the external input signals are imposed are named driver nodes. The number of unmatched nodes is equal to the minimum number of driver nodes needed to fully control the network. Note that the results reported in Liu *et al.* [24] critically depend on the assumption that the direct network has no self-links, i.e. a node's internal state can only be changed upon interaction with neighboring nodes [61]. We will follow this assumption throughout the chapter.

Network structural controllability as a generic system property is applied to measure and enhance network robustness. Measuring network robustness is usually done by measuring network performance changes during perturbations imposed upon the network [13]. The widely adopted perturbations in the research of the robustness of network controllability are random node or link removals, which are used as a benchmark compared with other perturbations. Another kind of perturbation deals with targeted attack strategies. For example, attack strategies can relate to network topology features, such as betweenness, degree and closeness. Pu *et al.* [62] demonstrate that degree-based removals are more harmful to network controllability compared to random removals. Lu *et al.* [63] find that a betweenness-based attack strategy is more harmful than a degree-based attack strategy in most real-world networks. However, Wang *et al.* [64] find that attacking bridge links, whose removal results in a disconnected network, is an effective way to destroy network controllability. Another kind of targeted attack strategy is based on critical nodes and links. Critical nodes and links are defined through the property that their removal will increase the number of driver nodes [24]. Sun *et al.* [65] report random attack under the protection of critical links is less efficient than a random link attack, and a targeted attack aiming at critical links is more harmful than a random attack. Lou *et al.* [66] propose a hierarchical attack removal framework where nodes or links are classified into critical, sub-critical and normal categories. They find that hierarchical attack strategies are more efficient than some metric-based attack strategies such as betweenness- or degree-based strategies in interdependent networks. There is also some research focusing on how to enhance the robustness of network controllability. Giulia *et al.* [67] show that network

controllability is determined by the density of nodes with in-degree and out-degree equal to one or two. Adding links to low degree nodes is beneficial to network controllability. Lou *et al.* [68] find that multi-loop structures can improve the robustness of network controllability. Zhang *et al.* [69] investigate different redundant design strategies of interdependent networks. They present that betweenness-based strategy and degree-based strategy for node backup and high degree first strategy for edge backup can optimize robustness of network controllability. Besides the aforementioned qualitative research on the robustness of network controllability, quantitative research has been conducted. Lu *et al.* [63] develop the numerical approximations of random node removals and targeted node removals based on degree on Erdös-Rényi (ER) networks. The results fit well when the fraction of nodes is below 20 %. Sun *et al.* [65] explore the closed-form approximation of the number of controllable nodes under random link removals, targeted removals and random removals with protection. Dhiman *et al.* [70] use machine learning to quantify the minimum fraction of driver nodes under random link removals and targeted link removals, which performs better than the closed-form approximation proposed by Sun *et al.* [65]. Later, Chen *et al.* [28] develop analytical approximations for the minimum number of driver nodes during random link removal by using methods based on generating functions.

However, to the best of our knowledge, analytical methods for approximating network controllability during random and targeted node removal in different types of networks are lacking. The framework to calculate the structural controllability of linear systems for directed networks has been proposed by Liu *et al.* [24]. This chapter utilizes their framework to develop analytical approximations based on degree distributions to calculate the minimum fraction of driver nodes during node removals. We consider two cases for node removal: random node removals and targeted node removals, based on the node's in-degree, out-degree, and total degree. To validate our methods, we employ two types of synthetic networks and four real-world communication networks.

This chapter is organized as follows. The second section introduces the networks used in the study for validation. The third and fourth sections provide an introduction to network controllability and targeted node removals. Analytical approximations and results demonstrating the robustness of network controllability under node removals are presented in the fifth section. The final section provides a summary.

2.2. NETWORK DATA

We will validate our theoretical results, which will be derived in the subsequent sections, on two classes of synthetic networks and on a number of real-world networks. In this section, we give details on the used networks.

2.2.1. DIRECTED SYNTHETIC NETWORKS

We choose two kinds of synthetic networks: Erdös-Rényi (ER) networks and Swarm Signalling networks (SSNs).

We generate a directed ER network on N nodes by placing a directed link between any pair of nodes, with a given probability p_{ER}. The average number of links for such ER networks satisfies $L = N(N-1)p_{ER}$. In this chapter, we have used two ER networks, with

$N = 50$, $p_{ER} = 0.07$ and $N = 100$, $p_{ER} = 0.04$.

Our topology for Swarm Signalling Networks (SSNs) was suggested in [71]. The SSN has a regular out-degree, whereas the in-degree distribution follows a Poisson distribution. To generate SSNs, we need two parameters. One is the number of nodes N, and the other is the out-degree value k. For each node, the node randomly creates k outgoing links to other nodes. This chapter uses two SSNs with $N = 10^4$, $k = 2$ and $N = 10^4$, $k = 5$.

2.2.2. REAL-WORLD NETWORKS

The real-world networks used in this study are taken from the Internet Topology Zoo [72], a collection of real-world communication networks. We change those undirected networks into directed networks by using two attributes: source node and target node [65]. The properties of the networks are shown in Table 2.1, which shows the number of nodes N, the number of links L, and the average total degree $< k >$. The total degree is the sum of the in- and out-degrees. Obviously, the average in-degree equals the average out-degree and therefore, the average total degree is twice the average in-degree (and hence the average out-degree).

Table 2.1: Properties of four real-world communication networks

Name	N	L	$< k >$
HinerniaGlobal	55	81	2.95
Syringa	74	74	2.00
Interoute	110	146	2.65
Cogentco	197	243	2.47

2.3. NETWORK CONTROLLABILITY

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

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t), \tag{2.1}$$

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

A linear, time-invariant networked system is considered controllable if its node states can be manipulated to reach any desired state within a finite time by applying a set of external inputs. The Kalman rank criterion provides a way to determine controllability, where the rank of the controllability matrix $[B, AB, A^2B, \ldots, A^{N-1}B]$ should be equal to N for the system to be fully controllable [73]. To gain an understanding of the Kalman rank criterion, we can derive the formal solution of Eq. (2.1) with an initial condition of $x(0) = \mathbf{0}$ as $x(t) = \int_0^\infty e^{A(t-\tau)} Bu(\tau)\, d\tau$. By expanding $e^{A(t-\tau)}$ into a series, we can deduce that $x(t)$ is a linear combination of the matrix $[B, AB, A^2B, \ldots, A^{N'}B, \ldots]$. According to the

Cayley-Hamilton theorem, for $N' > N$, the rank of the matrix $[B, AB, A^2B, \ldots, A^{N'}B, \ldots]$ is equivalent to the rank of the controllability matrix $[B, AB, A^2B, \ldots, A^{N-1}B]$. Consequently, if the rank of the controllability matrix is less than N, it implies that the matrix $[B, AB, A^2B, \ldots, A^{N'}B, \ldots]$ cannot span the state space of dimension N entirely. In such cases, an input $u(t)$ cannot be found to steer $x(0)$ to an arbitrary state $x(t)$ [74]. In practical applications, the implementation of the Kalman rank criterion poses challenges due to the requirement of obtaining information about the network's interaction strengths and the involvement of computationally intensive calculations, especially for large-scale networks. To mitigate these challenges, Lin [75] introduced the concept of structural controllability. Additionally, Liu *et al.* [24] presented the maximum matching method and the minimum inputs theorem to determine the minimum number of nodes (driver nodes) that must be controlled to ensure controllability. To determine the count of driver nodes, a directed network should first be transformed into a bipartite network. Subsequently, a maximum matching edge set can be derived using the maximum matching algorithm [76], consisting of N_M directed edges without shared source nodes or end nodes. The end nodes of the matching edges are termed matched nodes, while the remaining nodes are unmatched. The calculation of the minimum number N_D of driver nodes is as follows

$$N_D = \max\{1, N - N_M\}. \tag{2.2}$$

2.4. TARGETED NODE REMOVALS

Centrality analysis is an essential research area in studying network robustness [77]. Nodes with a high degree are known to have a substantial impact on network functioning and are more susceptible to targeted removals. In this study, our objective is to develop an analytical approximation of network controllability during targeted node removals based on three types of degrees: in-degree, out-degree and total degree.

Assuming that the probabilities of node attacks are proportional to some power of its in-degree, out-degree and total degree, we can express the probability of removing node i based on its in-degree k_{in_i} as p_{in_i}, based on its out-degree k_{out_i} as p_{out_i} and based on its total degree k_i as p_i. The formulas for calculating these probabilities are given as follows,

$$\begin{aligned} p_{in\_i} &= \frac{k_{in\_i}^{\alpha}}{\sum_{j \in \mathcal{N}} k_{in\_j}^{\alpha}}, \\ p_{out\_i} &= \frac{k_{out\_i}^{\alpha}}{\sum_{j \in \mathcal{N}} k_{out\_j}^{\alpha}}, \\ p_i &= \frac{{k_i}^{\alpha}}{\sum_{j \in \mathcal{N}} k_j^{\alpha}}. \end{aligned} \tag{2.3}$$

In the node removal process, after some nodes are removed, we recalculate the removal probabilities for the remaining nodes using Eq. (2.3). We then select nodes to remove based on the recalculated probabilities until all nodes are removed.

When $\alpha = 0$, the aforementioned equations become

$$\begin{aligned} p_{in\_i} &= \frac{1}{N}, \\ p_{out\_i} &= \frac{1}{N}, \\ p_i &= \frac{1}{N}, \end{aligned} \tag{2.4}$$

which indicates that each node has an equal probability of being removed, resulting in a random removal strategy. On the other hand, for $\alpha > 0$, nodes with higher degrees have a greater likelihood of being removed, while for $\alpha < 0$, nodes with lower degrees are more likely to be removed.

In this study, we investigate the impact of degree-based node removal strategies on network robustness. To this end, we focus on $\alpha > 0$, as higher-degree nodes are commonly targeted for attack in real-world scenarios. Specifically, we consider two values of α, namely $\alpha = 1$ and $\alpha = 10$, to evaluate the impact of removing nodes proportional to their degree and removing high-degree nodes more aggressively, respectively. By using Eq. (2.3), we obtain the probabilities of the node being removed based on in-degree, out-degree and total degree when $\alpha = 1$ as follows,

$$\begin{aligned} p_{in\_i} &= \frac{k_{in\_i}}{\sum_{j\in\mathcal{N}} k_{in\_j}}, \\ p_{out\_i} &= \frac{k_{out\_i}}{\sum_{j\in\mathcal{N}} k_{out\_j}}, \\ p_i &= \frac{k_i}{\sum_{j\in\mathcal{N}} k_j}. \end{aligned} \tag{2.5}$$

Analogously, the node removal probabilities based on in-degree, out-degree and total degree with $\alpha = 10$ can be calculated by

$$\begin{aligned} p_{in\_i} &= \frac{k_{in\_i}^{10}}{\sum_{j\in\mathcal{N}} k_{in\_j}^{10}}, \\ p_{out\_i} &= \frac{k_{out\_i}^{10}}{\sum_{j\in\mathcal{N}} k_{out\_j}^{10}}, \\ p_i &= \frac{{k_i}^{10}}{\sum_{j\in\mathcal{N}} k_j^{10}}. \end{aligned} \tag{2.6}$$

Our results show that, for $\alpha = 10$, the removal of high-degree nodes does not lead to a significant reduction in network robustness in the beginning stage. For several networks, there are no significant differences between the results with $\alpha = 1$ and $\alpha = 100$. Interestingly, we observe that increasing the value of α to 100 does not result in further performance gains, as the performance of removals with $\alpha = 100$ is similar to that of removals with $\alpha = 10$. Additional details on these findings can be found in Appendix A.1. Furthermore, we find that when $\alpha = 1$, the removal strategies based on in-degree or

out-degree can be more detrimental to certain networks than node removal based on the total degree. However, for some other networks, the harmful effects of these strategies are comparable. The results are presented in Appendix A.2.

2.5. ANALYTICAL APPROXIMATIONS UNDER NODE REMOVALS

This section presents how to analytically approximate network controllability in the case of different types of node removals and the corresponding results.

2.5.1. ANALYTICAL APPROXIMATIONS FOR RANDOM NODE REMOVALS

GENERAL NETWORKS

From [24], for directed network $\mathcal{G}(N,L)$ with N nodes and L links, we can determine the minimum number of driver nodes by using generating functions of the in- and out-degree distributions ($G_{in}(x)$ and $G_{out}(x)$, respectively) and of the excess in- and out-degree distributions ($H_{in}(x)$ and $H_{out}(x)$, respectively). These generating functions are defined as follows:

$$\begin{aligned} G_{in}(x) &= \sum_{k=0}^{\infty} P_{in}(k_{in}) x^{k_{in}}, \\ G_{out}(x) &= \sum_{k=0}^{\infty} P_{out}(k_{out}) x^{k_{out}}, \\ H_{in}(x) &= \frac{\sum_{k=1}^{\infty} k_{in} P_{in}(k_{in}) x^{k_{in}-1}}{< k_{in} >} = \frac{G'_{in}(x)}{G'_{in}(1)}, \\ H_{out}(x) &= \frac{\sum_{k=1}^{\infty} k_{out} P_{out}(k_{out}) x^{k_{out}-1}}{< k_{out} >} = \frac{G'_{out}(x)}{G'_{out}(1)}, \end{aligned} \tag{2.7}$$

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

$$\begin{aligned} n_d = \frac{1}{2}\{&G_{in}(\omega_2) + G_{in}(1-\omega_1) - 2 + G_{out}(\hat{\omega}_2) + G_{out}(1-\hat{\omega}_1) \\ &+ k\,[\hat{\omega}_1(1-\omega_2) + \omega_1(1-\hat{\omega}_2)]\}, \end{aligned} \tag{2.8}$$

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

$$\begin{aligned} \omega_1 &= H_{out}(\hat{\omega}_2), \\ \omega_2 &= 1 - H_{out}(1-\hat{\omega}_1), \\ \hat{\omega}_1 &= H_{in}(\omega_2), \\ \hat{\omega}_2 &= 1 - H_{in}(1-\omega_1), \end{aligned} \tag{2.9}$$

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

During the node removal process, the set of driver nodes includes two parts. One is the set containing N_D driver nodes that control the remaining part of the network, and

the other set is formed by N_r removed nodes. We assume that each removed node needs to be controlled by an individual driver node. We define the fraction of driver nodes n_D as $n_D = \frac{N_D+N_r}{N}$. After randomly removing a fraction p of nodes in the network, the fraction of driver nodes n_D satisfies

$$n_D = \frac{n_d(1-p)N + pN}{N} = n_d(1-p) + p. \tag{2.10}$$

Based on the research of Shao *et al.* [78], the generating function after randomly removing a fraction p nodes corresponds to the original generating function, with the adjusted argument $\bar{x} = p + (1-p)x$. Then the generating functions of in- and out-degree, and the excess in- and out-degree, after randomly removing a fraction p of nodes, are adjusted as follows:

$$\begin{aligned} \bar{G}_{in}(x) &= G_{in}(p + (1-p)x), \\ \bar{G}_{out}(x) &= G_{out}(p + (1-p)x), \\ \bar{H}_{in}(x) &= \frac{\bar{G}'_{in}(x)}{\bar{G}'_{in}(1)}, \\ \bar{H}_{out}(x) &= \frac{\bar{G}'_{out}(x)}{\bar{G}'_{out}(1)}. \end{aligned} \tag{2.11}$$

Next, we use Eqs. (2.8) and (2.10) to acquire the fraction of minimum number of nodes n_D after randomly removing a fraction p of nodes:

$$\begin{aligned} n_D = \frac{1}{2}(1-p)\{&\bar{G}_{in}(\omega_2) + \bar{G}_{in}(1-\omega_1) - 2 + \bar{G}_{out}(\hat{\omega_2}) + \bar{G}_{out}(1-\hat{\omega_1}) \\ &+ k(1-p)[\hat{\omega_1}(1-\omega_2) + \omega_1(1-\hat{\omega_2})]\} + p, \end{aligned} \tag{2.12}$$

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

$$\begin{aligned} \omega_1 &= \bar{H}_{out}(\hat{\omega}_2), \\ \omega_2 &= 1 - \bar{H}_{out}(1-\hat{\omega}_1), \\ \hat{\omega}_1 &= \bar{H}_{in}(\omega_2), \\ \hat{\omega}_2 &= 1 - \bar{H}_{in}(1-\omega_1), \end{aligned} \tag{2.13}$$

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

ER NETWORKS

Both the in-degree distribution $P_{in}(k_{in})$ and the out-degree distribution $P_{out}(k_{out})$ of ER networks follow a Poisson distribution with average degree k [28]. Therefore, the generating functions of in-degree and out-degree are as follows,

$$\begin{aligned} G_{in}(x) &= e^{-k(-x+1)}, \\ G_{out}(x) &= e^{-k(-x+1)}. \end{aligned} \tag{2.14}$$

The minimum fraction of driver nodes n_D after a fraction p of nodes is randomly removed in the ER networks can be obtained through Eq. (2.12) as

$$n_D = p + p\omega_2 - \omega_2 + [1 - p + k(1-p)^2(1-\omega_2)]e^{k(1-p)(\omega_2-1)} \tag{2.15}$$

where ω_2 satisfies $1 - \omega_2 - e^{-k(1-p)e^{-k(1-p)(1-\omega_2)}} = 0$.

SSNs

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

$$\begin{aligned} G_{in}(x) &= e^{-k(-x+1)}, \\ G_{out}(x) &= x^k. \end{aligned} \tag{2.16}$$

Based on Eq. (2.12), the minimum fraction of driver nodes n_D after randomly removing a fraction p of nodes can be calculated by

$$n_D = p + p\omega_2 - \omega_2 + [1 - p + (k-1)(1-p)^2(1-\omega_2)]e^{k(1-p)(\omega_2-1)} \tag{2.17}$$

where ω_2 satisfies $1 - \omega_2 - [p + (1-p)(1 - e^{-k(1-p)(1-\omega_2)})]^{k-1} = 0$.

Note that for real-world networks, the generating functions for the in- and out-degree distributions, can simply be obtained from the histograms of these distributions. We use the relative frequency of degree as the corresponding probability in generating functions.

2.5.2. ANALYTICAL APPROXIMATIONS FOR TARGETED NODE REMOVALS

CASE: $\alpha = 1$

In-degree: In undirected networks, after a fraction p of nodes has been removed based on their degree, specifically, the probability of a node removal is proportional to some power of its degree, see Eq. (2.3), the generating function of the degree distribution, $G(x)$, transforms into function $\bar{G}(x)$, which is as follows, [77],

$$\bar{G}(x) = \frac{1}{1-p}\sum_{k=0}^{\infty} p_k f^{k^\alpha}\left(1 + \frac{fG'_\alpha(f)}{< k >}(x-1)\right)^k, \tag{2.18}$$

where $f \equiv G_\alpha^{-1}(1-p)$, $G_\alpha(x) \equiv \sum_k p_k x^{k^\alpha}$ and $< k >$ is the average degree of the initial network.

We investigate the extension of prior conclusions to directed networks while removing nodes based on their in-degree. We assume that a node's in-degree and out-degree are independent and uncorrelated, such that removing a fraction p of nodes based on their in-degree results in the generating function of the in-degree distribution described by Eq. (2.18). Furthermore, the generating function of the out-degree distribution is given by $\bar{G}_{out}(x) = G_{out}(p + (1-p)x)$ following the equation of random node removals. So if we remove nodes based on in-degree, function

$\bar{G}_{in}(x)$ and function $\bar{G}_{out}(x)$ satisfy,

$$\begin{aligned}\bar{G}_{in}(x) &= \frac{1}{1-p}\sum_{k_{in}=0}^{\infty} p_{k_{in}} f^{k_{in}} \left(1+\frac{fG_1'(f)}{<k_{in}>}(x-1)\right)^{k_{in}},\\ \bar{G}_{out}(x) &= G_{out}(p+(1-p)x).\end{aligned} \tag{2.19}$$

Then, we can obtain the analytical approximation of the minimum fraction of driver nodes under node removals based on in-degree using Eq. (2.12).

Out-degree: Analogously, if we remove a fraction p of nodes based on their out-degree, we maintain the assumption that the generating function of the out-degree distribution is described by Eq. (2.18). Additionally, the generating function of the in-degree distribution can be expressed as $\bar{G}_{in}(x) = G_{in}(p+(1-p)x)$. Therefore, we have function $\bar{G}_{in}(x)$ and function $\bar{G}_{out}(x)$ as follows,

$$\begin{aligned}\bar{G}_{in}(x) &= G_{in}(p+(1-p)x),\\ \bar{G}_{out}(x) &= \frac{1}{1-p}\sum_{k_{out}=0}^{\infty} p_{k_{out}} f^{k_{out}} \left(1+\frac{fG_1'(f)}{<k_{out}>}(x-1)\right)^{k_{out}}.\end{aligned} \tag{2.20}$$

Furthermore, utilizing Eq. (2.12), we can derive an analytical approximation of the minimum fraction of driver nodes when nodes are removed based on out-degree.

Total-degree: The main challenge is to obtain expressions for the generating functions for the in- and out-degree distributions after removing a fraction p of the nodes through removals. In general, it is not possible to obtain the generating function both for the in- and the out-degree distribution, after a fraction p of nodes has been attacked. Therefore we have to come up with a heuristic to deal with this. Here we will map the targeted node attack process (based upon total degree) into a random node attack process. We suppose that the generating functions of in-degree distribution and out-degree distribution change to those corresponding to random node removal, but such that the total number of links after randomly removing a fraction $\bar{p}$ of nodes is equal to the total number of links after targeted removal of a fraction p of the nodes. As reported in [77], the fraction $\bar{p}$ can be calculated by

$$\bar{p} = 1 - \frac{fG_\alpha'(f)}{<k>}, \tag{2.21}$$

where $f \equiv G_\alpha^{-1}(1-p)$, $G_\alpha(x) \equiv \sum_k p_k x^{k^\alpha}$ and $<k>$ is the average total degree of the initial network and p_k is the probability of a node having total degree k. If $\alpha = 1$, $G_\alpha(x) \equiv \sum_k p_k x^k$, which is the generating function for the total degree distribution. For ER networks, the generating function of total degree is $G(x) = e^{-<k>(-x+1)}$ and for SSNs, the generating function of total degree is $G(x) = x^{\frac{<k>}{2}} e^{-\frac{<k>}{2}(-x+1)}$.

CASE: $\alpha = 10$

When $\alpha = 10$, we encounter difficulties in obtaining a numerical solution for $f \equiv G_\alpha^{-1}(1-p)$, where $G_\alpha(x) \equiv \sum_k p_k x^{k^\alpha}$. Consequently, it becomes challenging to determine the

evolution of the generating functions for in-degree and out-degree distributions during the node removal process. To address this challenge, we propose a heuristic approach whereby we map the targeted node removal process based on in-degree or out-degree into a random node attack process.

Specifically, for node removals based on in-degree with $\alpha = 10$, where a fraction of p nodes is to be removed, we map this process to the removal of $\bar{p}$ nodes in the in-degree distribution, while maintaining the fraction of nodes in the out-degree distribution at p. Similarly, for node removals based on out-degree with $\alpha = 10$, we map the process to the random removal of a fraction of $\bar{p}$ nodes in the out-degree distribution, as well as a fraction of p nodes in the in-degree distribution.

In-degree: In order to estimate the corresponding $\bar{p}$ of a given fraction p under node removals based on in-degree with $\alpha = 10$, we adopt the assumption that nodes are removed in descending order of in-degree. Specifically, we first sort the nodes according to their in-degree and then remove nodes starting from the node with the highest in-degree until the targeted fraction p is reached.

Next, we calculate the total in-degree of all the removed nodes by utilizing the original in-degree distribution and the targeted removal fraction p. The effective fraction $\bar{p}$ is then obtained by normalizing the total in-degree of all removed nodes with respect to the total in-degree of all nodes in the initial network. This can be calculated as follows,

$$\bar{p}_{in} = \frac{\sum_{k_{in}=k_{in_{max}}}^{k_{in}=\bar{k}_{in}} p_{k_{in}} N k_{in}}{N < k_{in} >} = \frac{\sum_{k_{in}=k_{in_{max}}}^{k_{in}=\bar{k}_{in}} p_{k_{in}} k_{in}}{< k_{in} >}, \tag{2.22}$$

where the largest in-degree value is denoted as $k_{in_{max}}$, the probability of removed nodes with degree k_{in} is denoted as $p_{k_{in}}$ and degree $\bar{k}_{in}$ satisfies $\sum_{k_{in}=k_{in_{max}}}^{k_{in}=\bar{k}_{in}} p_{k_{in}} = p$. It is worth mentioning that except removed probability $p_{\bar{k}_{in}}$, other probability $p_{k_{in}}$ is equal to probability $P_{in}(k_{in})$ in the generating function. Then we can use effective proportion $\bar{p}_{in}$ for the approximation of the minimum fraction of driver nodes as follows,

$$\begin{aligned}
\bar{G}_{in}(x) &= G_{in}(\bar{p}_{in} + (1-\bar{p}_{in})x), \\
\bar{G}_{out}(x) &= G_{out}(p + (1-p)x), \\
n_D &= \frac{1}{2}\{\bar{G}_{in}(\omega_2) + \bar{G}_{in}(1-\omega_1) - 2 + \bar{G}_{out}(\hat{\omega_2}) + \bar{G}_{out}(1-\hat{\omega_1}) \\
&\quad + k\left(1 - \frac{p+\bar{p}_{in}}{2}\right)[\hat{\omega_1}(1-\omega_2) + \omega_1(1-\hat{\omega_2})]\}\left(1 - \frac{p+\bar{p}_{in}}{2}\right) + \frac{p+\bar{p}_{in}}{2},
\end{aligned} \tag{2.23}$$

where $\omega_1, \omega_2, \bar{\omega_1}$ and $\bar{\omega_2}$ satisfy Eq. (2.13).

Out-degree Analogously, for targeted node removal based on out-degree with $\alpha = 10$, the calculation of fraction $\bar{p}_{out}$ follows the same assumption: nodes are removed from the node with the highest out-degree to the node with the lowest out-degree until

the removed fraction of nodes reaches p. The effective fraction $\bar{p}_{out}$ is the total out-degree of removed nodes normalized by the total out-degree in the original network, which can be calculated by,

$$\bar{p}_{out} = \frac{\sum_{k_{out}=k_{out_{max}}}^{k_{out}=\bar{k}_{out}} p_{k_{out}} N k_{out}}{N < k_{out} >} = \frac{\sum_{k_{out}=k_{out_{max}}}^{k_{out}=\bar{k}_{out}} p_{k_{out}} k_{out}}{< k_{out} >}, \tag{2.24}$$

where the largest degree value is denoted as $k_{out_{max}}$, and the probability of removed nodes with out-degree k_{out} as $p_{k_{out}}$. To achieve the targeted removal fraction p, we find the minimum out-degree value $\bar{k}_{out}$ satisfying $\sum_{k_{out}=k_{out_{max}}}^{k_{out}=\bar{k}_{out}} p_{k_{out}} = p$. For all out-degree values except for $\bar{k}_{out}$, their corresponding probabilities $p_{k_{out}}$ are equal to the probabilities $P_{out}(k_{out})$ in the generating function. Then, we use $\bar{p}_{out}$, the effective proportion of removed nodes based on out-degree, to estimate the minimum number of driver nodes, which is given by the following expression,

$$\begin{aligned}
\bar{G}_{in}(x) &= G_{in}(p + (1-p)x), \\
\bar{G}_{out}(x) &= G_{out}(\bar{p}_{out} + (1-\bar{p}_{out})x), \\
n_d &= \frac{1}{2}\{\bar{G}_{in}(\omega_2) + \bar{G}_{in}(1-\omega_1) - 2 + \bar{G}_{out}(\hat{\omega_2}) + \bar{G}_{out}(1-\hat{\omega_1}) \\
&\quad + k\left(1 - \frac{p+\bar{p}_{out}}{2}\right)[\hat{\omega_1}(1-\omega_2) + \omega_1(1-\hat{\omega_2})]\}\left(1 - \frac{p+\bar{p}_{out}}{2}\right) + \frac{p+\bar{p}_{out}}{2},
\end{aligned} \tag{2.25}$$

where $\omega_1, \omega_2, \bar{\omega_1}$ and $\bar{\omega_2}$ satisfy Eq. (2.13).

Total-degree: We map the fraction p of removed nodes under targeted removals for $\alpha = 10$ onto the effective proportion $\bar{p}$ of nodes under random node attack. Under the attack strategy to remove the largest degree node at each step, total degree of all removed nodes can be obtained according to the degree distribution after giving the removed fraction p. The effective proportion $\bar{p}$ is the total degree of all removed nodes normalizing by the total degree of all nodes in the initial network, which can be calculated as $\bar{p} = \frac{\sum_{k=k_{max}}^{k=\bar{k}} p_k N k}{N<k>} = \frac{\sum_{k=k_{max}}^{k=\bar{k}} p_k k}{<k>}$, where the largest degree value is denoted as k_{max}, the probability of removed nodes with degree k is denoted as p_k and degree $\bar{k}$ satisfies $\sum_{k=k_{max}}^{k=\bar{k}} p_k = p$. Similarly, except removed probability $p_{\bar{k}}$, other probability p_k is equal to probability $P(k)$ in the generating function. Then the minimum number of driver nodes can be approximated by replacing argument p by $\bar{p}$ in Eqs. (2.11) and (2.12).

2.5.3. RESULTS FOR RANDOM NODE REMOVALS

We ran simulations on various networks, as described in Section 2.2. We carried out 10000 realizations for all networks to ensure sufficient statistical power. For ER and real-world networks, which have a relatively small number of nodes, one node was removed at each step until all nodes had been removed during each realization. Then a recalculation of the minimum fraction of driver nodes was conducted by using the algorithm. On

the other hand, due to the large number of nodes in SSNs, 1% of nodes were removed at each step until all nodes had been removed during each realization. Subsequently, the minimum fraction of driver nodes was recalculated based on the modified network structure. The average value of results obtained from the 10000 realizations was taken as the final simulation output. The green lines in Figs. 2.1 and 2.2 show the simulation results.

Since we know each network's in-degree and out-degree distributions, we can compute the minimum fraction of driver nodes of a network according to the equations mentioned above for the minimum fraction n_D. The results obtained using the generating functions of the degree distributions are depicted as red dashed lines in Figs. 2.1 and 2.2.

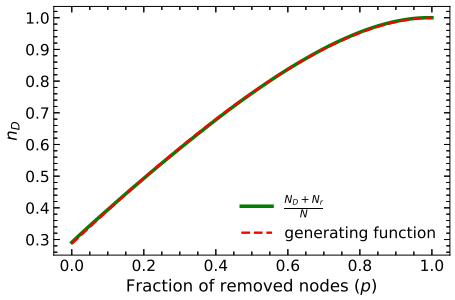
The results are shown in Figs. 2.1 and 2.2. As the predicted values in the red lines and the simulated values virtually overlap, we conclude that the analytical approximations for network controllability in the case of random node removals are very accurate. The reason for this is that, after removing a fraction p of the nodes at random, we still have expressions for the generating functions of the in- and out-degree distributions, see Eq. (2.11).

2.5.4. RESULTS FOR TARGETED NODE REMOVALS

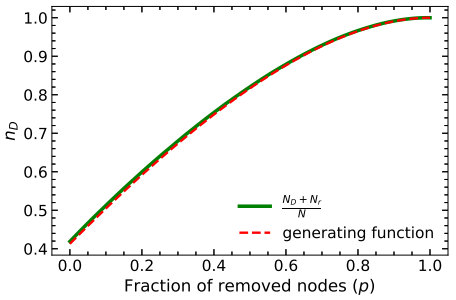
IN-DEGREE AND OUT-DEGREE WITH $\alpha = 1$

We ran simulations with 10000 realizations with $\alpha = 1$ under in-degree and out-degree node removals for all mentioned networks. Each realization is the same as described in Section 2.5.3. We present the results of targeted node removal based on in-degree and out-degree with $\alpha = 1$, as depicted in Figs. 2.3 and 2.4, and Figs. 2.5 and 2.6. The simulation results are shown in green lines, whereas the analytical results are in red. The results of random node removal are also presented in grey lines for comparison. We observe that the analytical results serve as a closed-form approximation of the minimum fraction of driver nodes (n_D), as a discrepancy exists between the predicted and simulation values during the targeted node removal process based on in-degree or out-degree. In the case of ER networks, the in-degree and out-degree distributions are identical. Consequently, the predicted values of targeted removal based on in-degree and out-degree are also the same. For SSNs, the out-degree of nodes is fixed. Therefore, the lines of analytical results of targeted node removal based on out-degree with $\alpha = 1$ in SSNs overlap with the lines of random node removal. We find that for SSNs, the simulation results of targeted removal based on in-degree and out-degree are slightly different from the simulation results of random removals.

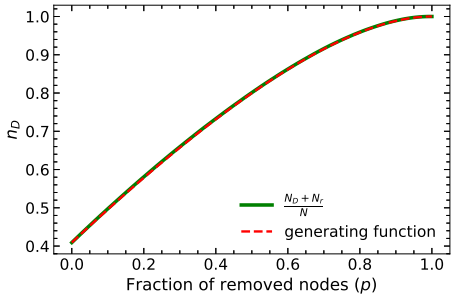
When the removed fraction p is small, the simulation results of targeted removals based on in-degree and out-degree are close to those of random removals. We verified this by calculating the Root Mean Square Error (RMSE) between the simulation results of targeted removals based on in-degree and out-degree and analytical results of randomly removing nodes below 10%, as shown in Table 2.2. Moreover, we calculated the RMSE between the simulation and analytical results of targeted node removals based on in-degree and out-degree below 10%, as shown in Table 2.3. The results indicated that both methods provide a good approximation of the simulation results, as the values in both tables for targeted node removals based on in-degree and out-degree with $\alpha = 1$ are



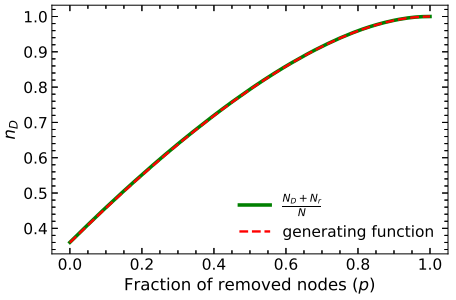
(a) HinerniaGlobal



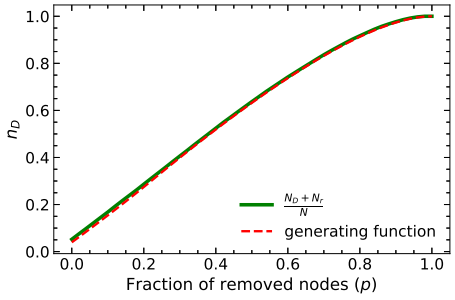
(b) Syringa



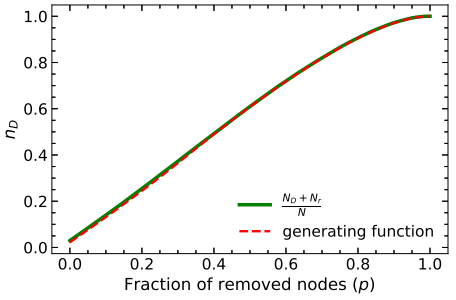
(c) Interoute



(d) Cogentco

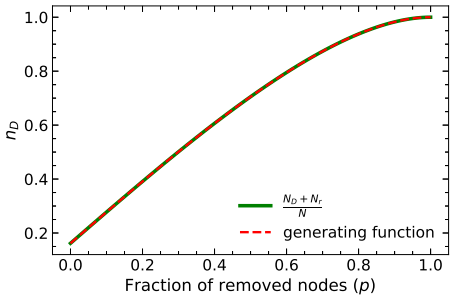


(e) ER(50,0.07)

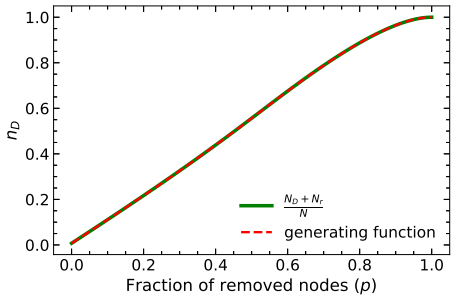


(f) ER(100,0.04)

Figure 2.1: The minimum fraction of driver nodes n_D during random node removal for different kinds of networks. The green lines are calculated by the maximum matching algorithm. The red dashed lines are obtained by the analytical methods.



(a) SSN(10^4, 2)



(b) SSN(10^4, 5)

Figure 2.2: The minimum fraction of driver nodes n_D during random node removals for SSN networks. The green lines are calculated by the maximum matching algorithm. The red dashed lines are obtained by the analytical methods.

reasonably small.

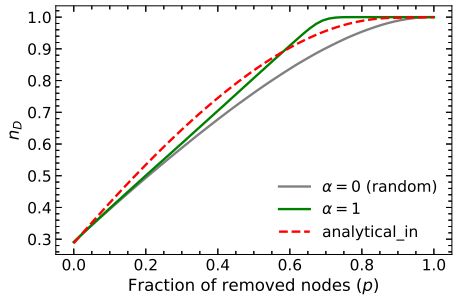
Table 2.2: The RMSE between the analytical results of random removals and the simulation results under random removals, targeted removals with $\alpha = 1$ and $\alpha = 10$, respectively, while removing up to 10% of the nodes. The column labeled "random" indicates the RMSE under random removals. The columns labeled "$\alpha = 1$" and "$\alpha = 10$" represent the RMSE under targeted node removals with $\alpha = 1$ and $\alpha = 10$, respectively. The columns labeled "in-degree", "out-degree", and "degree" represent the RMSE under targeted node removals based on in-degree, out-degree, and total degree, respectively

network	random	$\alpha = 1$			$\alpha = 10$		
		in-degree	out-degree	degree	in-degree	out-degree	degreee
ER(50, 0.07)	0.0137	0.0164	0.0156	0.0155	0.0190	0.0195	0.0223
ER(100, 0.04)	0.0079	0.0086	0.0095	0.0094	0.0126	0.0121	0.0156
HinerniaGlobal	0.0039	0.0052	0.0110	0.0084	0.0025	0.0152	0.0152
Syringa	0.0071	0.0136	0.0217	0.0179	0.0237	0.0263	0.0443
Interoute	0.0011	0.0008	0.0106	0.0056	0.0064	0.0072	0.0175
Cogentco	0.0011	0.0090	0.0053	0.0071	0.0156	0.0091	0.0248
SSN(10^4, 2)	0.0000	0.0103	0.0003	0.0052	0.0143	0.0007	0.0155
SSN(10^4, 5)	0.0000	0.0006	0.0000	0.0003	0.0008	0.0001	0.0008

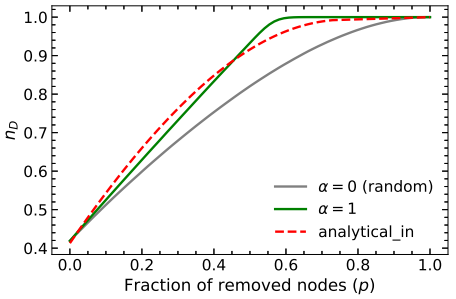
TOTAL-DEGREE WITH $\alpha = 1$

We chose the same network set to do simulations under targeted node removal, based on total degree. We did 10000 realizations for each communication network and 1000 for each synthetic network. Each realization is done as described in Section 2.5.3. The simulation results are presented as green lines in Figs. 2.7 and 2.8. For the analytical method, we employ the effective fraction of removed nodes $\bar{p}$ acquired by Eq. (2.21). Red lines in Figs. 2.7 and 2.8 represent the analytical results. We also show the simulation results under random node removals in grey lines in Figs. 2.7 and 2.8.

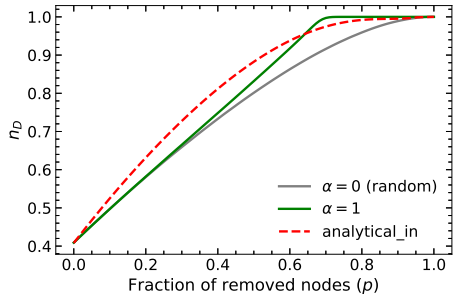
We find that the analytical results are a reasonable fit with the simulations, especially



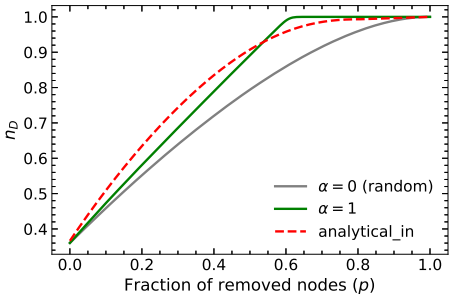
(a) HinerniaGlobal



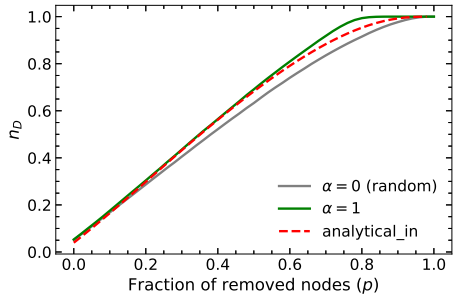
(b) Syringa



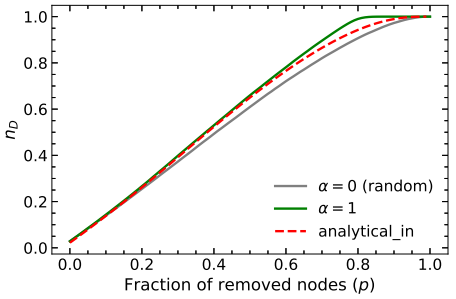
(c) Interoute



(d) Cogentco

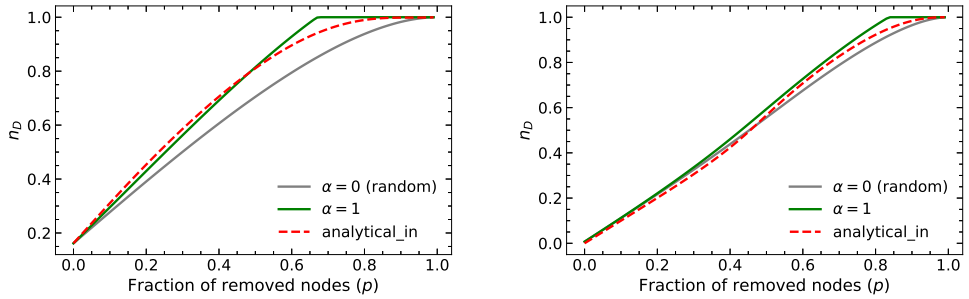


(e) ER(50,0.07)



(f) ER(100,0.04)

Figure 2.3: The minimum fraction of driver nodes n_D during targeted node removal based on in-degree with $\alpha = 1$ for different kinds of networks. The grey lines are the results of simulations under random node removal ($\alpha = 0$), and the green lines are calculated by the maximum matching algorithm. The red dashed lines are obtained by the analytical methods.



(a) SSN(10^4, 2) (b) SSN(10^4, 5)

Figure 2.4: The minimum fraction of driver nodes n_D during targeted node removal based on in-degree with $\alpha = 1$ for SSN networks. The grey lines are the results of simulations under random node removal ($\alpha = 0$), and the green lines are calculated by the maximum matching algorithm. The red dashed lines are obtained by the analytical methods.

Table 2.3: The RMSE between the analytical results of the proposed analytical methods and the simulation results under different kinds of removals while removing up to 10% of the nodes. The column labeled "random" indicates the RMSE under random removals. The columns labeled "$\alpha = 1$" and "$\alpha = 10$" represent the RMSE under targeted node removals with $\alpha = 1$ and $\alpha = 10$, respectively. The columns labeled "in-degree", "out-degree", and "degree" represent the RMSE under targeted node removals based on in-degree, out-degree, and total degree, respectively

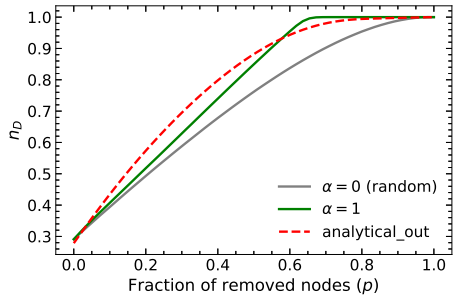
network	random	$\alpha = 1$			$\alpha = 10$		
		in-degree	out-degree	degree	in-degree	out-degree	degreee
ER(50, 0.07)	0.0137	0.0122	0.0113	0.0095	0.0588	0.0595	0.0543
ER(100, 0.04)	0.0079	0.0058	0.0067	0.0039	0.0189	0.0193	0.0284
HinerniaGlobal	0.0039	0.0089	0.0136	0.0025	0.0281	0.0349	0.0354
Syringa	0.0071	0.0096	0.0143	0.0061	0.0157	0.0235	0.0142
Interoute	0.0011	0.0151	0.0242	0.0009	0.0265	0.0454	0.0229
Cogentco	0.0011	0.0233	0.0244	0.0050	0.0314	0.0330	0.0322
SSN(10^4, 2)	0.0000	0.0085	0.0002	0.0027	0.0331	0.0006	0.0343
SSN(10^4, 5)	0.0000	0.0094	0.0000	0.0024	0.0167	0.0001	0.0264

for small values of the fraction p of attacked nodes. It indicates that the proposed method of calculating the effective proportion $\bar{p}$ is inaccurate in the late removal stage.

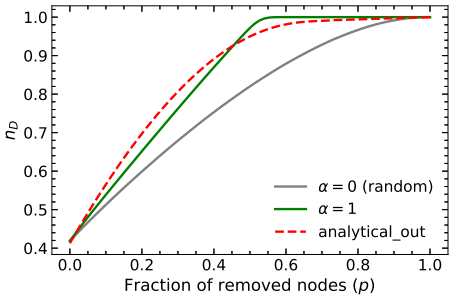
IN-DEGREE AND OUT-DEGREE WITH $\alpha = 10$

We ran the simulations of 10000 realizations with $\alpha = 10$ under in-degree and out-degree node removals in mentioned networks. Each realization of every network is the same as described in Section 2.5.3. The simulation results of network controllability are shown in the green lines in Figs. 2.9 and 2.10, and Figs. 2.11 and 2.12. As before, the analytical results are depicted in red lines, while the simulation results of network controllability under random node removals are shown in grey lines.

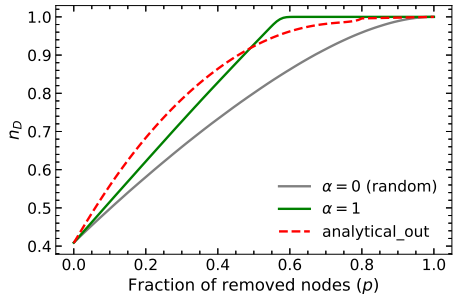
In addition to targeted node removals based on out-degree in SSNs with fixed out-



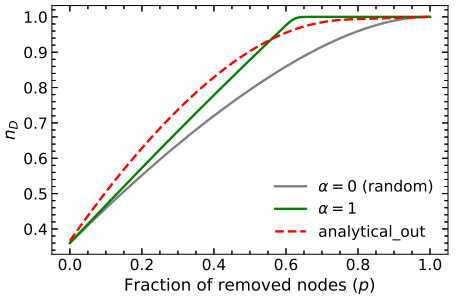
(a) HinerniaGlobal



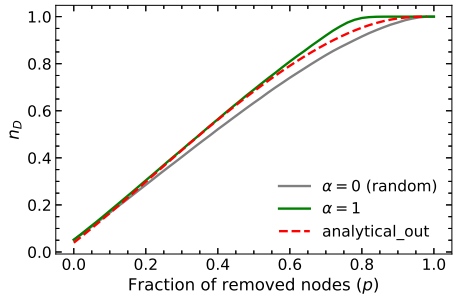
(b) Syringa



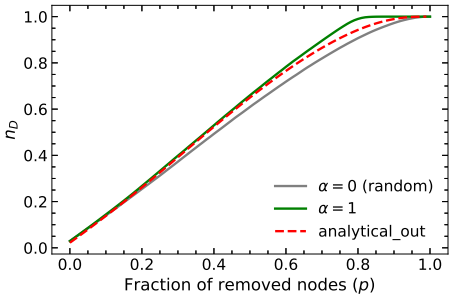
(c) Interoute



(d) Cogentco

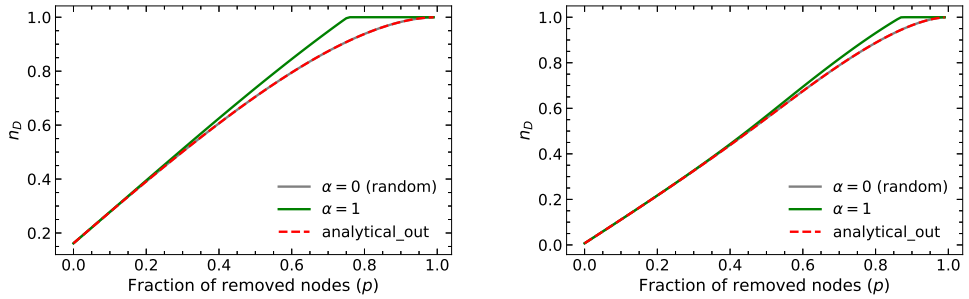


(e) ER(50,0.07)



(f) ER(100,0.04)

Figure 2.5: The minimum fraction of driver nodes n_D during targeted node removal based on out-degree with $\alpha = 1$ for different kinds of networks. The grey lines are the results of simulations under random node removal ($\alpha = 0$), and the green lines are calculated by the maximum matching algorithm. The red dashed lines are obtained by the analytical methods.



(a) SSN(10^4, 2)

(b) SSN(10^4, 5)

Figure 2.6: The minimum fraction of driver nodes n_D during targeted node removal based on out-degree with $\alpha = 1$ for SSN networks. The grey lines are the results of simulations under random node removal ($\alpha = 0$), and the green lines are calculated by the maximum matching algorithm. The red dashed lines are obtained by the analytical methods.

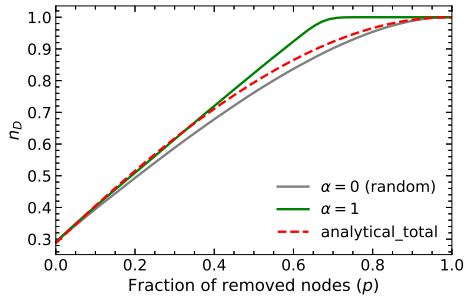
degree, the analytical results are consistent with random node removals. Notably, the analytical results exhibit a similar pattern for $\alpha = 10$, where they initially surpass the simulation results before eventually intersecting and becoming inferior to the targeted node attack lines but superior to the random node attack lines as the fraction of removed nodes approaches one. We find the proposed methods can closely approximate network controllability using a closed-form approach but do not precisely align with simulation results.

Upon examining Tables 2.2 and 2.3, we observe that both the proposed analytical methods and the analytical results of random node removal demonstrate satisfactory performance for targeted node removal based on in-degree and out-degree with $\alpha = 10$ when the fraction of removed nodes p is below 10%. However, the values obtained for $\alpha = 10$ are comparatively inferior to those obtained for $\alpha = 1$ and random node removal. These outcomes highlight the limitations of our proposed approach. Specifically, our method assumes that nodes are removed from the node with the highest degree to the node with the lowest degree, which is true when α is large enough like infinity. In this context, we choose $\alpha = 10$ and the node with the highest degree is much more likely to be removed but still can not be guaranteed to be removed at each step.

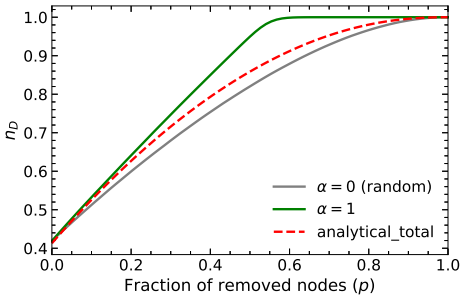
TOTAL-DEGREE $\alpha = 10$

Analogously, we do the simulations with $\alpha = 10$ under total degree targeted node removal, 10000 realizations for each communication network and 1000 realizations for each synthetic network. Each realization is carried out as described in Section 2.5.3. The simulation results are shown in the green lines. We present the analytical results in red lines. The simulation results of network controllability under random node removals are depicted in grey lines. The results with $\alpha = 10$ for total degree targeted node removal are shown in Figs. 2.13 and 2.14.

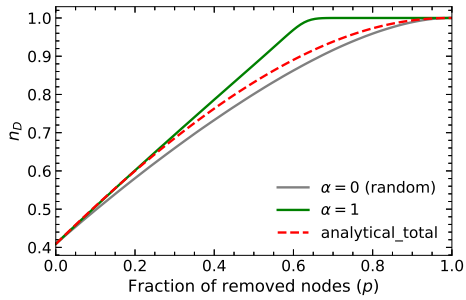
The proposed approaches for the case $\alpha = 10$ can approximate network controllability



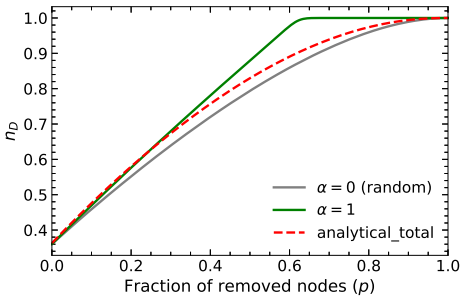
(a) HinerniaGlobal



(b) Syringa



(c) Interoute



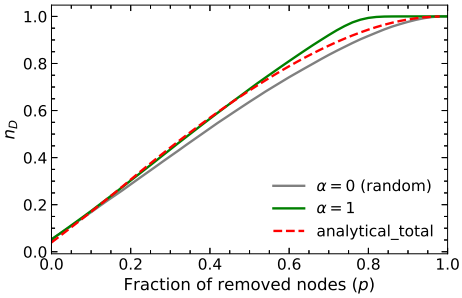
(d) Cogentco

Figure 2.7: The minimum fraction of driver nodes n_D during targeted node removal based on the total degree with $\alpha = 1$ for real-world networks. The grey lines are the results of simulations under random node removal ($\alpha = 0$), and the green lines are calculated by the maximum matching algorithm. The red dashed lines are obtained by the analytical methods.

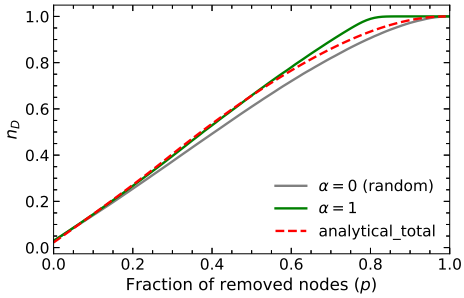
in a closed-form but do not perfectly fit the simulation results. The analytical result lines are first above the targeted attack lines, then below the targeted attack lines but still above the random attack lines, until the fraction of removed nodes approaches one.

2.6. CHAPTER SUMMARY

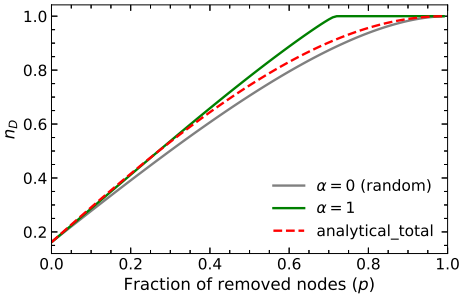
In this chapter, we propose analytical methods based on generating functions to compute the minimum fraction of the number of driver nodes in directed networks, subject to node removals. Analytical methods fit simulation results very well for random node removals. Moreover, we develop analytical methods for three cases during targeted node removal based on in-, out- and total degrees. Our proposed analytical methods demonstrate reasonable results to predict the minimum fraction of driver nodes under targeted removals. Furthermore, our investigation indicates that random node removals may also serve as a reliable predictor of the results of various targeted node removals, particularly when the fraction of removed nodes is minimal (below 10%).



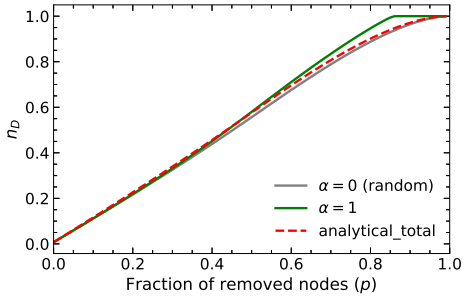
(a) ER(50, 0.07)



(b) ER(100, 0.04)

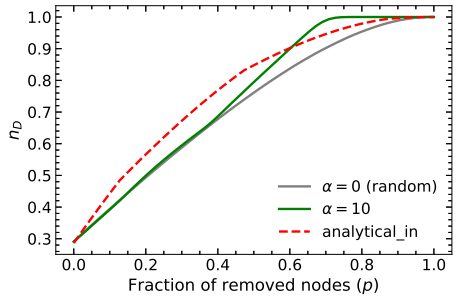


(c) SSN(10^4, 2)

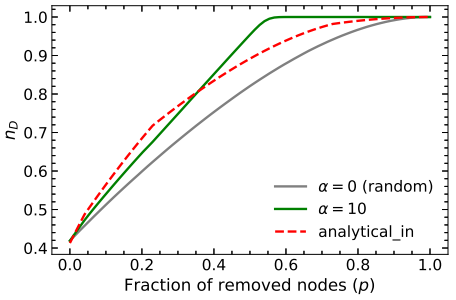


(d) SSN(10^4, 5)

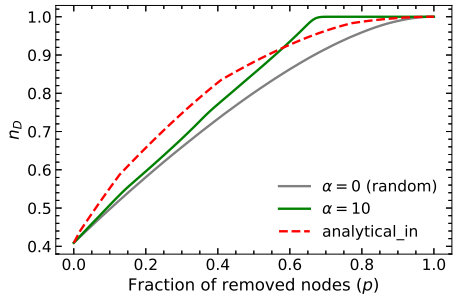
Figure 2.8: The minimum fraction of driver nodes n_D during targeted node removal based on the total degree with $\alpha = 1$ for synthetic networks. The grey lines are the results of simulations under random node removal ($\alpha = 0$), and the green lines are calculated by the maximum matching algorithm. The red dashed lines are obtained by the analytical methods.



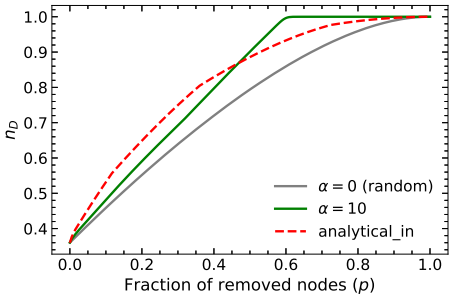
(a) HinerniaGlobal



(b) Syringa

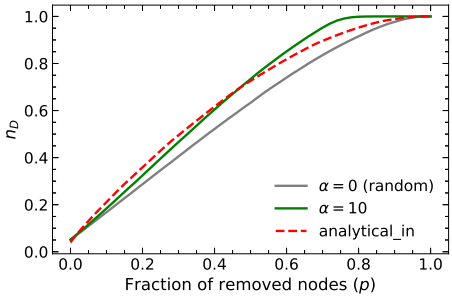


(c) Interoute

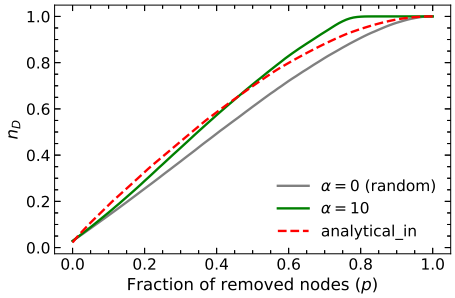


(d) Cogentco

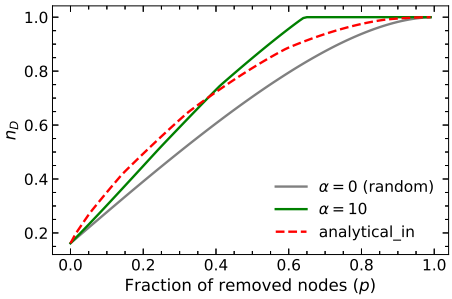
Figure 2.9: The minimum fraction of driver nodes n_D during targeted node removal based on in-degree with $\alpha = 10$ for real-world networks. The grey lines are the results of simulations under random node removal ($\alpha = 0$), and the green lines are calculated by the maximum matching algorithm. The red dashed lines are obtained by the analytical methods.



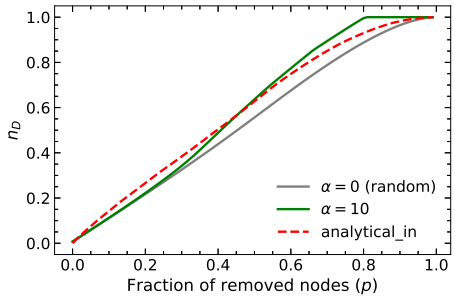
(a) ER(50,0.07)



(b) ER(100,0.04)

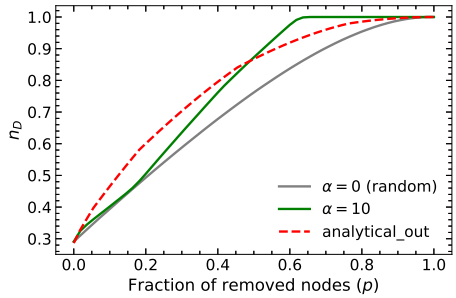


(c) SSN(10^4,2)

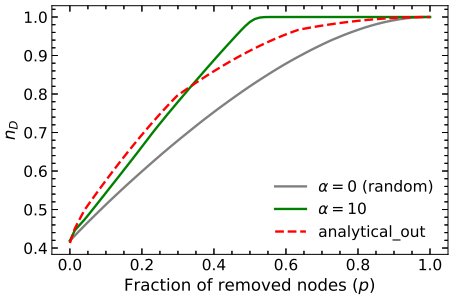


(d) SSN(10^4,5)

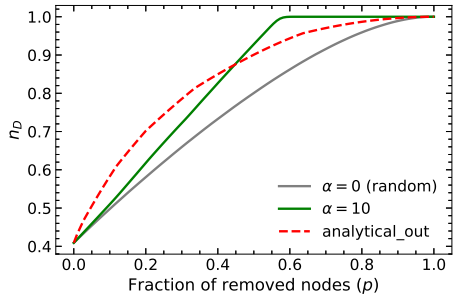
Figure 2.10: The minimum fraction of driver nodes n_D during targeted node removal based on in-degree with $\alpha = 10$ for synthetic networks. The grey lines are the results of simulations under random node removal ($\alpha = 0$), and the green lines are calculated by the maximum matching algorithm. The red dashed lines are obtained by the analytical methods.



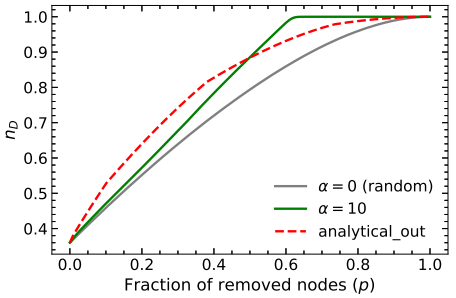
(a) HinerniaGlobal



(b) Syringa

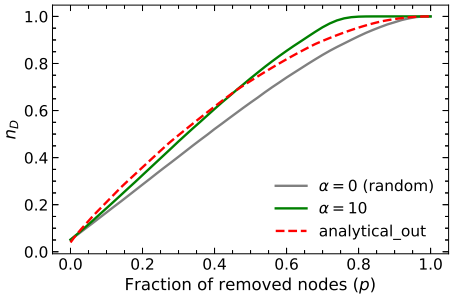


(c) Interoute

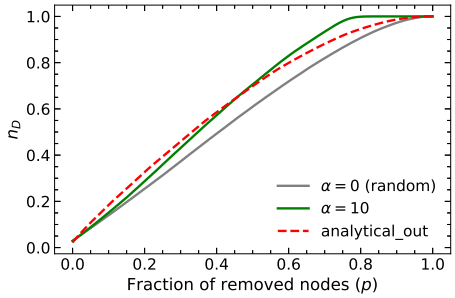


(d) Cogentco

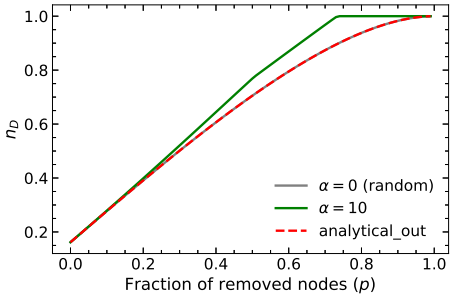
Figure 2.11: The minimum fraction of driver nodes n_D during targeted node removal based on out-degree with $\alpha = 10$ for real-world networks. The grey lines are the results of simulations under random node removal ($\alpha = 0$), and the green lines are calculated by the maximum matching algorithm. The red dashed lines are obtained by the analytical methods.



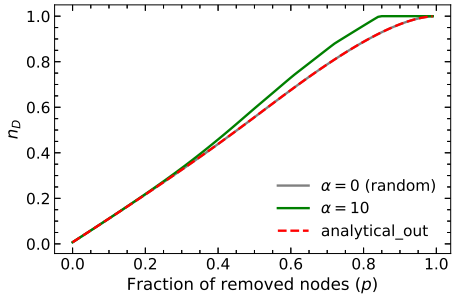
(a) ER(50,0.07)



(b) ER(100,0.04)

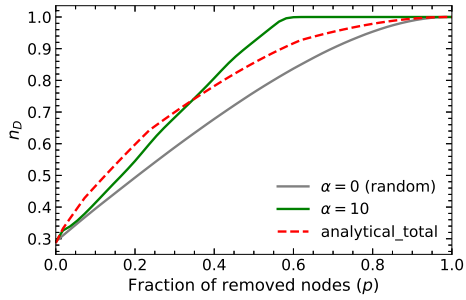


(c) SSN(10^4,2)

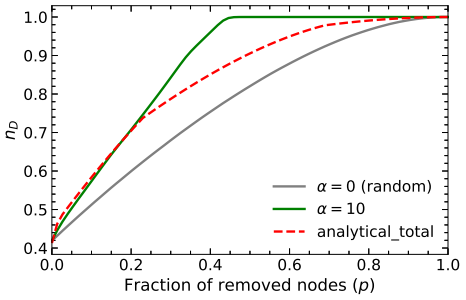


(d) SSN(10^4,5)

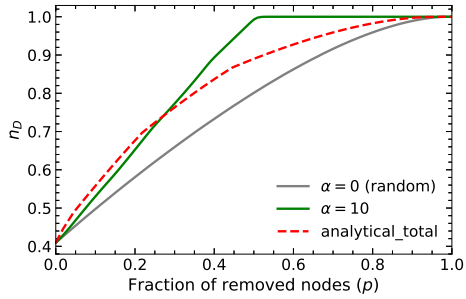
Figure 2.12: The minimum fraction of driver nodes n_D during targeted node removal based on out-degree with $\alpha = 10$ for synthetic networks. The grey lines are the results of simulations under random node removal ($\alpha = 0$), and the green lines are calculated by the maximum matching algorithm. The red dashed lines are obtained by the analytical methods.



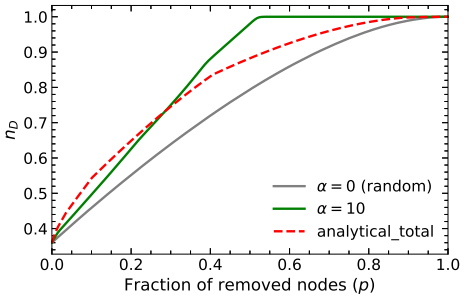
(a) HinerniaGlobal



(b) Syringa

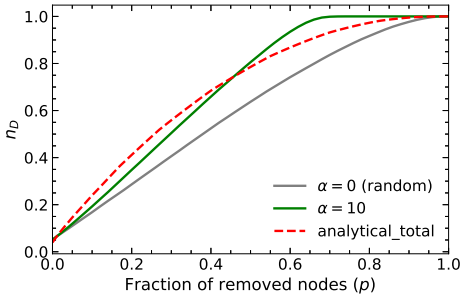


(c) Interoute

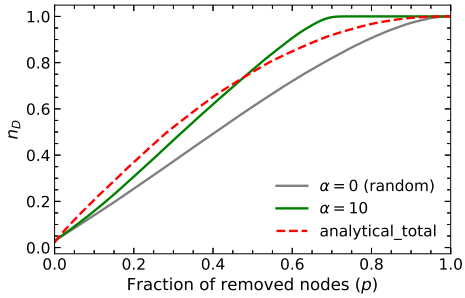


(d) Cogentco

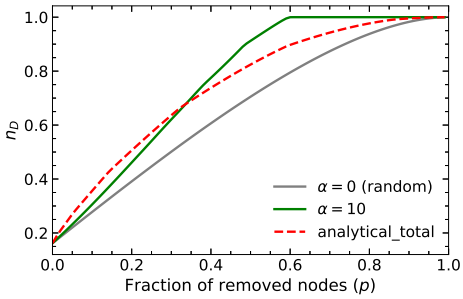
Figure 2.13: The minimum fraction of driver nodes n_D during targeted node removal based on the total degree with $\alpha = 10$ for real-world networks. The grey lines are the results of simulations under random node removal ($\alpha = 0$), and the green lines are calculated by the maximum matching algorithm. The red dashed lines are obtained by the analytical methods.



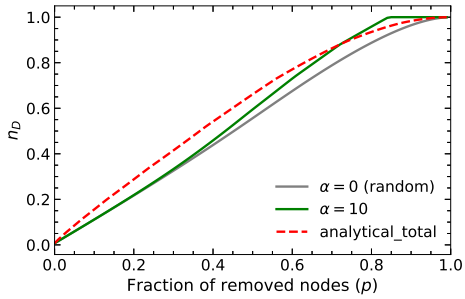
(a) ER(50, 0.07)



(b) ER(100, 0.04)



(c) SSN(10^4, 2)



(d) SSN(10^4, 5)

Figure 2.14: The minimum fraction of driver nodes n_D during targeted node removal based on the total degree with $\alpha = 10$ for synthetic networks. The grey lines are the results of simulations under random node removal ($\alpha = 0$), and the green lines are calculated by the maximum matching algorithm. The red dashed lines are obtained by the analytical methods.

3

CONTROLLER PLACEMENT WITH RESPECT TO CONTROLLER REACHABILITY

Using controller reachability as a network performance metric, we investigate the controller placement problem on networks. Controller reachability is the probability that each node can reach at least one controller, given that each link is operational with a fixed probability. By exploring placements for more than 100 real-world networks and by varying the number of controllers from two to five, we find that controller reachability varies greatly with different placements. Obviously, increasing the number of controllers increases the controller reachability. However, the extent of this increase depends on the strategy with which the controllers are placed. The findings indicate that efficient controller placement strategies should be developed to ensure good network performance. In this research, we propose four controller placement strategies. One strategy is based on topological network metrics: the degree of the node and the path length between the controllers and nodes. The other three heuristic strategies are the greedy algorithm, a classic genetic algorithm, and a heuristic genetic algorithm. By validating strategies on real-world networks, we find that all four strategies work well to solve the controller placement problem with respect to controller reachability.

This chapter is based on the published paper [79].

3.1. INTRODUCTION

The Controller Placement Problem (CPP) on networks belongs to the class of facility location problems, which looks to place facilities in potential locations, such that certain performance metrics are optimized, often under constraints of cost. The study of controller placement problems has been widely investigated in the setting of Software-Defined Networking (SDN). The SDN network can be divided into the data plane, the control plane, and the application plane, where the data plane corresponds to the networking devices that are responsible for forwarding data such as switches. The control plane decides how to handle network traffic using controllers, and the application plane remotely monitors and configures the control functionality through the interface (northbound API) [80]. SDN has a logically centralized control architecture achieved by decoupling the network control plane and the data plane [81].

Within the Controller Placement Problem (CPP) in Software-Defined Networking (SDN), two fundamental research questions arise, each crucial to optimizing network performance [29]. The first question centers on determining the appropriate number of controllers to be deployed in the network. Although a single controller is typically sufficient for smaller networks, larger networks require the addition of multiple controllers to maintain optimal performance levels [82]. The use of a multi-controller design has shown promising results in improving the performance of the SDN network [83], albeit at the expense of increased deployment costs. Hence, determining the optimal number of controllers becomes imperative for striking a balance between performance and cost considerations.

The second research question relates to the strategic placement of controllers within the network. The performance of the network is inherently influenced by the specific locations chosen for the controllers, given a fixed number of controllers. As the number of controllers increases, the number of potential placement combinations increases exponentially, intensifying the challenge of identifying optimal configurations. Consequently, the question of where to place the controllers emerges as a critical consideration. Addressing this question is vital to achieving superior network performance by minimizing latency, optimizing resource utilization, and ensuring efficient communication between the control plane and the data plane. The solutions to the research questions heavily rely on the formulation approaches employed for the CPP in SDN.

The formulation of the CPP in SDN requires the consideration of various performance metrics that directly impact the design of effective controller placement strategies. Three fundamental metrics, namely latency, reliability, and cost, play critical roles in the evaluation and optimization of the placement of controllers [81]. The first performance metric, latency, captures the time delays experienced during data transmission and processing. Transmission latency encompasses delays that occur between switches and controllers, as well as inter-controller communication. It is often quantified using the average or maximum distances between the network components. Processing latency arises when controllers become overloaded, exceeding their processing capacity. Minimizing latency is crucial to ensure efficient communication between the control plane and the data plane, resulting in improved network responsiveness. Reliability constitutes the second performance metric and is concerned with the probability of successfully reaching controllers in the event of network component failures. Low reliability levels can lead to

packet losses due to the absence of viable transmission paths. Evaluating the probabilities associated with multiple control paths and determining the availability of such paths are vital in ensuring network robustness and fault tolerance. The third performance metric encompasses cost considerations. Economic costs encompass expenses related to construction, maintenance, operation, and other financial factors. Environmental costs, such as energy consumption and CO_2 emissions, are also taken into account. Balancing costs while optimizing controller placements is essential for achieving economically and environmentally sustainable SDN deployments. The performance metrics can be used in isolation or in combination to formulate the CPP as either a single-objective optimization problem, aiming to optimize a specific metric, or a multi-objective optimization problem, considering multiple metrics simultaneously [84].

Determining the number of controllers required is a fundamental challenge, and it is closely intertwined with the problem of where to place the controllers. The NP-hard nature of the controller placement problem [29][85] necessitates the exploration of diverse algorithms and methodologies to identify optimal placements. One commonly employed method is the brute force algorithm, which aims to find the optimal solution, simply by considering the full state space of all possible placements. However, this algorithm is inefficient and suitable primarily for small-scale networks. To address larger-scale networks, heuristic algorithms, including the greedy algorithm, simulated annealing, genetic algorithm, and others, have been widely adopted. Additionally, linear programming and graph-based algorithms have been proposed as alternative approaches to tackle the complexity of the controller placement problem [86].

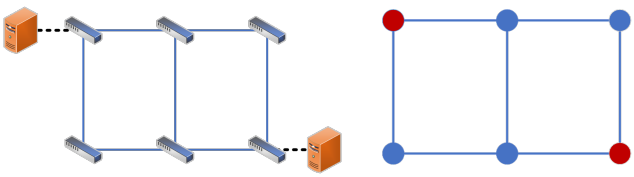
Heller *et al.* [29] propose the controller placement problem with a focus on minimizing propagation delays in wide-area networks. They utilize average latency (k-median) and worst-case latency (k-center) as performance metrics to optimize controller placements. By exhaustively enumerating every combination of controllers, they find that random placement falls significantly short of the optimal placement in nearly all network topologies, thereby underscoring the substantial impact of placement on network performance. Furthermore, in most topologies, it is challenging to optimize both average latency and worst-case latency simultaneously. Hu *et al.* [87] formulate the reliability-aware controller placement (RCP) problem. The expected percentage of control path loss is considered as a metric to reflect the reliability of networks. In the RCP problem, they look for the number of controllers and the placement in a network with a given failure probability of each component such that the reliability is optimized. They find that reliability and latency cannot be optimized at the same time for most networks. However, the placement with optimal reliability exhibits quite a good average latency as well. Ros *et al.* [88] introduce the Fault Tolerant Controller problem, wherein a heuristic algorithm is developed to determine the optimal controller placement, considering the lower bound reliability as a performance metric. Zhong *et al.* [89] define a reliability metric in the control network as the average number of disconnected switches when a single edge fails. They propose a min-cover-based method to minimize the number of controllers and maximize reliability. The concept of "cover" is employed, representing a set of nodes with controllers to ensure that every switch in the network belongs to at least one controller's neighborhood, guaranteeing low propagation delays.

In this research, we aim to investigate efficient controller placement strategies using a

reliability metric. We assume that nodes, including controllers, are always operational. At the same time, links between nodes and controllers or between controllers are also assumed to be always operational. However, links among nodes are subject to operational probabilities. In addition to traditional reliability metrics based on paths, we introduce one novel reliability metric called controller reachability. Controller reachability quantifies the probability that all nodes can reach at least one controller. To calculate the exact controller reachability value, we employ a path composition algorithm. By analyzing more than one hundred real-world networks and considering various controller placement scenarios, we aim to highlight the differences among placements and emphasize the importance of efficient placement techniques. Subsequently, we propose different controller placement strategies, including a strategy based on graph metrics, a greedy algorithm, and two strategies utilizing genetic algorithms. To validate the effectiveness of the proposed placement strategies, we conduct experiments on five representative medium-sized real-world networks.

3.2. CONTROLLER REACHABILITY

The data plane in SDN can be mathematically represented as an undirected graph, denoted as $G(\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ represents the set of nodes corresponding to network devices, and $\mathcal{L}$ represents the set of links representing connections between network devices. The graph consists of N nodes and L links. Each link l_{ij} indicates the presence of a connection between node i and node j. Notably, in the data plane, controllers are co-located with nodes, assuming that connections between controllers and nodes are always operational. Fig. 3.1 provides a graphical representation of an SDN network using a graph model. Each node in the graph represents a network device, and the presence of a red node signifies the co-location of a controller. We assume that nodes within the network always remain operational and links have an independent and identical operational probability p.



(a) SDN network (b) Graph representation

Figure 3.1: Graph representation of SDN network.

We define controller reachability $C_r(K, p, s)$ as the probability that each node can reach at least one controller, assuming each link is operational with probability p, when placing K controllers in the graph G on the node set s, where the cardinality of set s is K and $s \subseteq \mathcal{N}$. To calculate controller reachability, we enumerate the cases of $m, 1 \leq m \leq L$ operational links and count the number of case I_m in which each node can reach at least one controller. The probability that each node can reach at least one controller given m operational links and $L-m$ nonoperational links is $I_m p^m (1-p)^{L-m}$. By considering all

possible cases, the controller reachability with K controllers placed at node set s can be obtained as follows,

$$C_r(K,p,s) = \sum_{m=1}^{L} I_m p^m (1-p)^{L-m}. \tag{3.1}$$

The aforementioned enumeration method is suitable for small-scale networks, but it becomes computationally infeasible for larger networks due to the exponential growth in running time with an increasing number of links. To address this issue, we can estimate the controller reachability $\hat{C}_r(K,p,s)$ for a given operational probability value p using Monte Carlo simulation [31]. This estimation is obtained by performing a large number (n) of independent sampling trials as follows:

$$\hat{C}_r(K,p,s) = \frac{1}{n} U_n \tag{3.2}$$

where U_n represents the number of samples in which every node successfully reaches at least one controller. However, it is worth noting that Monte Carlo Simulation may not be computationally efficient when estimating rare event probabilities [90]. Achieving a good approximation often requires running the simulation for a significant number of iterations.

Given the limitations of the enumeration and Monte Carlo simulation methods in terms of network size and computation time, we propose a modified approach to calculate controller reliability by utilizing a path decomposition algorithm. The path decomposition algorithm offers an efficient means to compute the exact value of all-terminal reliability as long as the width of a decomposition is sufficiently small [91]. A path decomposition of a graph $G = (V, E)$ is given by a sequence of vertex subsets $(X_1, X_2, ..., X_r)$ that satisfy three conditions: (1) $\cup_{1\le i\le r} X_i = V$;(2) For every edge $(v, w) \in E$, there exists an index i such that $1 \le i \le r$, where both endpoints v and w are elements of X_i;(3) For any three indices i, j, k: if $i \le j \le k$, then $X_i \cap X_k \subseteq X_j$. The width of a decomposition is given by $\max_{1\le i\le r} |X_i| - 1$ [92]. Pathwidth is the minimum width over all possible decompositions of a graph, a problem known to be NP-hard [93]. All-terminal reliability $Rel(p)$ is defined as the probability that all terminals (nodes) can reach each or the probability that the network is connected [94]. With the assumption that all links are independently operational with probability p and all nodes are always operational, all-terminal reliability can be calculated as follows,

$$Rel(p) = \sum_{m=1}^{L} F_m p^m (1-p)^{L-m}, \tag{3.3}$$

where F_m is the number of operating network states with m operational links. In the context of our problem, we leverage this algorithm to calculate controller reachability, which is equivalent to all-terminal reliability when a single controller is present in the graph. To apply the path decomposition algorithm for cases where there are multiple controllers in the network, we introduce a modification of the problem at hand. Specifically, we consider merging all controllers together as a super-controller node. The links connecting controllers and nodes are modified to connect the super-controller node with nodes. To eliminate duplicate links, we adjust the link operational probabilities

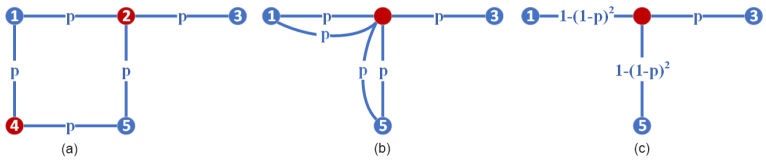


Figure 3.2: Example of controllers merging. (a) two controllers are placed on node 2 and node 4 in the graph with five nodes; (b) merging node 2 and node 4 to a super-controller node presented in red resulting in two links between node 1 and the super-controller node, and two links between node 5 and the super-controller node; (c) adjusting the link operational probabilities to eliminate duplicate links.

accordingly. The resulting network comprises a single controller and does not contain duplicate links, allowing us to apply the path decomposition algorithm to calculate the all-terminal reliability efficiently. To illustrate the process of controllers merging, we provide an example in Fig. 3.2.

3.3. DATA SET

In this study, we utilized real-world communication networks from the Topology Zoo dataset [72]. We specifically selected a subset of networks consisting of 100 small-sized networks with the number of nodes ranging from 11 to 30, as well as five medium-sized networks with over 50 nodes. To provide an overview of the networks used, we present the average node degree and network sizes in Fig. 3.3. The average node degree spans from 1.875 to 4.48, and among the networks, the smallest in size is Abilene with 11 nodes and 14 links, while the largest network is Cogentco with 197 nodes and 243 links. Additionally, we present the number of nodes (N), the number of links (L), and the average degree d_{av}, the number of nodes with degrees equal to one and two ($d = 1$ and $d = 2$) for the five medium-sized networks in Table 3.1.

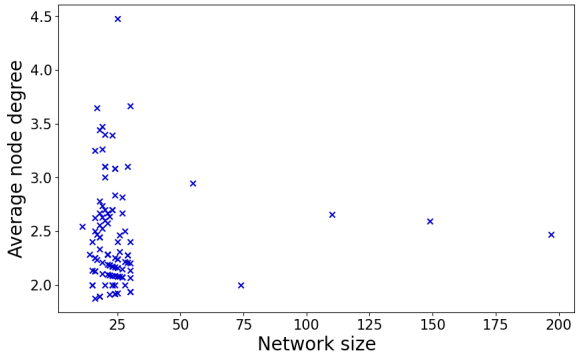


Figure 3.3: Average node degree with respect to network size for 100 small real-world networks and five medium real-world networks from the Topology Zoo.

Table 3.1: Properties of five medium-sized real-world networks from the Topology Zoo

Name	N	L	d_{av}	$d=1$	$d=2$
HinerniaGlobal	55	81	2.945	1	20
Syringa	74	74	2.000	23	34
Interoute	110	146	2.655	8	53
Cogentco	197	243	2.467	22	95
GtsCe	149	193	2.591	12	80

3.4. PLACEMENT STRATEGIES

3.4.1. CONTROLLER PLACEMENT STRATEGY BASED ON DEGREE AND DISTANCE

Nodes with low degrees, specifically degrees equal to one and two, exhibit higher vulnerability to disconnection within a network. Testing on three distinct sets of graphs, denoted as $\Omega(N,L)$, where each set encompasses all non-isomorphic connected graphs with the same numbers of nodes and links, we find that the optimal placements prefer to include nodes with low degrees. Specifically, we investigated the sets $\Omega(7,10)$, $\Omega(10,12)$, and $\Omega(9,18)$, which contain 132, 8548, and 33366 graphs, respectively. The optimal placements with varying numbers of controllers (2, 3, 4, and 5) are determined for every graph within these selected sets. During this process, the degrees of the nodes corresponding to the optimal controller placements are recorded. The acquired information produces statistical findings, which are shown in Fig. 3.4. The results reveal a significant trend suggesting a preference for nodes with lower degrees in the optimal placement of controllers.

To enhance controller reachability, it is effective to assign higher priorities to nodes with lower degrees. Moreover, considering the distance between nodes and controllers is also important. Fig. 3.5 presents a comprehensive analysis of the distances between two controllers across our set of 100 small real-world graphs. Each column within the figure represents a specific graph, while each data point signifies a possible distance between two controllers. Notably, the red data points correspond to the distances observed when the optimal placements are implemented. By examining the positioning of these red data points within their respective columns, valuable insights can be gained regarding the relationship between optimal controller placements and the distance. Specifically, if a red point is at the top of its column, it indicates that the optimal placement of this graph is the node pair with the maximum distance. When the link operational probability $p = 0.99$, the optimal placements of 67 graphs are observed to be the node pairs with the maximum distance, while the optimal placements of 14 graphs correspond to the node pairs with the second-largest distance. None of the analyzed graphs exhibit an optimal placement where the selected node pair consists of two neighboring nodes, i.e. with a distance of one. It can be concluded that distance is a significant graph metric that influences controller reachability. When placing two controllers, the optimal placement tends to occur at the node pair with the maximum distance. This preference for maximizing distance helps ensure that no node within the network is located too far away from the controllers.

Considering these two graph metrics, we propose an effective strategy for placing K controllers. The core concept of this strategy involves partitioning nodes into distinct

sets based on their degrees, with a preference for selecting nodes from sets with smaller degrees. In cases where nodes have the same degree, the algorithm aims to maximize the distance between controllers, thereby minimizing the overall distance between controllers and nodes.

Specifically, in the graph G, we denote the lowest degree as d_1, the second lowest degree as d_2 (where $d_1 \neq d_2$), the i-th lowest degree as d_i and the highest degree as d_M ($d_1 < d_2 < \ldots < d_M$). We consider n_1 nodes with degree d_1 as set $S(d_1)$, n_2 nodes with degree d_2 as set $S(d_2)$, n_i nodes with degree d_i as set $S(d_i)$. Besides, we use $d_0 = 0$, $n_0 = 0$, and $S(d_0) = \emptyset$ to indicate that there are no nodes with a degree of 0 in the network. Consequently, we obtain M non-empty sets of nodes that do not overlap and collectively cover every node in graph G. We aim to find the node set $S(d_k)$ such that $\sum_{i=0}^{k-1} n_i < K \leq \sum_{i=0}^{k} n_i$. The node sets with degree lower than d_k are defined as the initial existing controllers set $S_C = \bigcup_{i=0}^{k-1} S(d_i)$, the controllers are placed on every node in this set due to their low degree. The number of remaining controllers is defined as $K' = K - \sum_{i=0}^{k-1} n_i$. The nodes in $S(d_k)$ are considered as potential locations for placing K' controllers according to the distance.

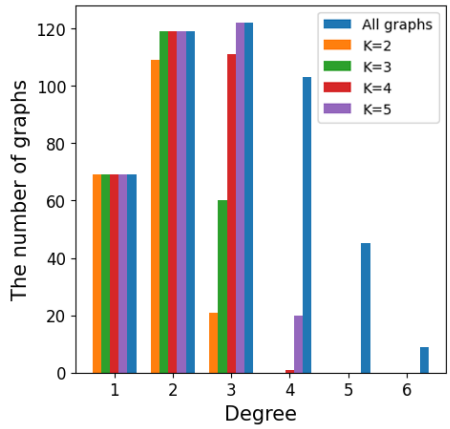
If $K \leq n_1$, $S_C = \emptyset$, the nodes in $S(d_1)$ are considered as potential locations to place $K' = K$ controllers. For each node in set $S(d_1)$, we compute the sum of its distance to other nodes in G. The node with the highest sum of distances is selected as the location for the first controller. From the placement of the second controller onwards, we select the node that has the longest distance to the existing controllers as the location for the next controller.

If $K > n_1$, the node set that we are going to place K' controllers in is determined as follows: If $n_1 < K \leq n_1 + n_2$, $S_C = S(d_1)$, the nodes in $S(d_2)$ are considered as potential locations for placing $K' = K - n_1$ controllers. If $n_1 + n_2 < K \leq n_1 + n_2 + n_3$, $S_C = S(d_1) \cup S(d_2)$, the nodes in $S(d_3)$ are considered as potential locations for placing $K' = K - n_1 - n_2$ controllers, etc. Using this method, we can identify the locations of $K - K'$ controllers based on the node degree, and then place the remaining K' controllers based on the distance within a smaller set of nodes. Within this set, the distance between each node and the existing controllers is computed. The node with the longest distance to the existing controllers is selected as the location for the next controller. This process is repeated until K nodes are identified. The pseudo-code of the algorithm is presented in Algorithm 1.

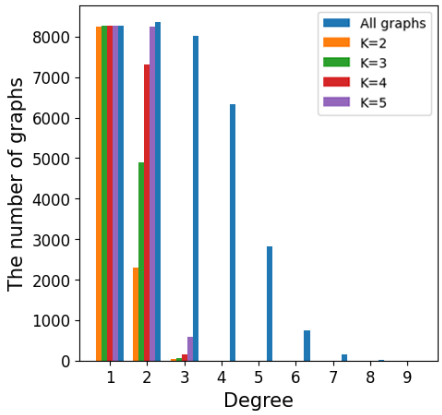
3.4.2. GREEDY ALGORITHM

The greedy algorithm is employed to determine controller placements in a step-by-step manner. At each iteration, the algorithm selects a location that maximizes the improvement in the performance metric, which in our case is the controller reachability. Since each decision is made based solely on the available information at that step, the greedy algorithm may yield a solution that is locally optimal but not necessarily globally optimal.

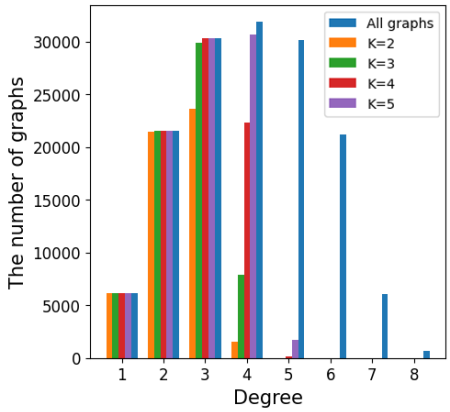
When placing the first controller in the network, there is no distinction in terms of controller reachability among the available options. However, the choice made for the first controller will impact the subsequent decisions. To address this "black start" problem encountered in the greedy algorithm, we explore four different approaches to start the algorithm:



(a) $\Omega(7,10)$



(b) $\Omega(10,12)$



(c) $\Omega(9,18)$

Figure 3.4: Statistics on optimal placement for three sets of graphs. The blue bars represent the number of graphs containing nodes with a specific degree, while the remaining bars indicate the number of graphs where the optimal placement (at $p = 0.99$) of K controllers includes at least one node with that degree.

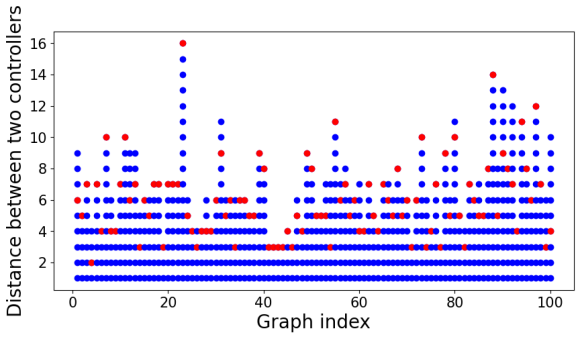


Figure 3.5: Statistics on optimal placement for 100 real-world graphs ($K = 2, p = 0.99$). The x-axis denotes the index of graphs, and the y-axis denotes the distance between the two controllers. In this figure, each point represents a possible distance between two controllers. The red points represent the distance between two controllers when the optimal placements are employed.

- Randomly choose a node,
- Randomly choose a node with the lowest degree,
- Use Algorithm1 to determine the first node,
- Enumerate all possible placements for $K = 2$ controllers and select the optimal solution as the first two controllers to be placed.

Not surprisingly, randomly choosing a node performs poorly when placing the first few controllers, but it gradually approaches the performance of other methods. Enumerating all possible placements with two controllers yields the best performance, but it is rather time-consuming. The remaining two methods both select a node with the lowest degree, but Algorithm 1 also takes into consideration the distance. Therefore, we have ultimately decided to use Algorithm 1 to determine the first node in the greedy algorithm. The computational time required for determining the first node is short, almost negligible compared to the running time of the greedy algorithm.

Starting from the second controller, the greedy algorithm iterates over all nodes without controllers and selects the node that offers the highest improvement in controller reachability. This process is repeated until K controllers have been placed. The pseudo-code for the greedy algorithm is provided in Algorithm 2.

3.4.3. GENETIC ALGORITHMS

The genetic algorithm is a well-known meta-heuristic algorithm that draws inspiration from biological evolution. In each generation, individuals from the current population are selected as parents, and through their reproduction, new children are generated. The algorithm aims to gradually progress towards the optimal solution by continually improving the fitness of the individuals in the population. The genetic algorithm encompasses several key elements, namely chromosome representation, selection, crossover, mutation, and fitness function computation [95]. Selection, mutation, and crossover

Algorithm 1 Controller Placement Algorithm based on degree and distance

Input: network G, number of controllers K
Output: Set S_C

Define set S_C as the set of nodes with controllers
Define d_i as the i-th lowest degree
Define n_i as the number of nodes with degree d_i
Define $S(d_i)$ as the set of nodes with degree d_i
Define $n_0 = 0$, $S(d_0) = \emptyset$
Define $distance(u, v)$ as the shortest path length between u and v
Find the set $S(d_k)$ such that $\sum_{i=0}^{k-1} n_i < K \leq \sum_{i=0}^{k} n_i$
$S_C = \bigcup_{i=0}^{k-1} S(d_i)$
$K' = K - \sum_{i=0}^{k-1} n_i$
if $K < n_1$ **then**
  for $v \in S(d_1)$ **do**
    $SumDistance(v) = \sum_{u \in G, u \neq v} (distance(u, v))$
  end for
  Add the node v with the highest $SumDistance(v)$ into S_C
  $K' = K' - 1$
end if
while $K' > 0$ **do**
  for $v \in S(d_k)$ **do**
    $D(v) = \min(distance(u, v))$ where $u \in S_C$
  end for
  Add the node v with the highest $D(v)$ into S_C
  $K' = K' - 1$
end while
return Set S_C

are often referred to as biologically-inspired operators. In this research, we developed two genetic algorithms: the classic genetic algorithm and the heuristic genetic algorithm. Both algorithms employ the same method for generating the initial population, and the fitness value is determined based on the controller reachability.

In the context of the controller placement problem, an individual in the genetic algorithm population is represented as a sequence of K genes, where each gene corresponds to the node that hosts a controller. To generate the initial population, we employ the method proposed in [96], which ensures that each gene has a similar frequency of occurrence in the population. Specifically, the gene frequency f in the initial population is determined by the following equation,

$$f = \max\left\{2, \left\lceil \frac{N}{100} \cdot \frac{\ln(n_s)}{d} \right\rceil \right\} \tag{3.4}$$

where N is the number of nodes, K is the number of controllers, $n_s = \binom{N}{K}$ is the number of possible placements, $d = \lceil \frac{N}{K} \rceil$ is the rounded-up density of the problem. The gene frequency is at least two. The initial population size P can be calculated by $P = f \cdot d$.

Algorithm 2 Greedy algorithm

Input: network G, number of controllers K
Output: Set S_C

Define set S_C as the set of nodes with controllers
The first controller S_{C1} is chosen based on algorithm 1
$K = K - 1$
while $K > 0$ **do**
  for node $v \notin S_C$ **do**
    Compute the controller reachability $C_r(v)$ if controllers are placed at node v and nodes in S_C
  end for
  Add node v with the highest $C_r(v)$ into set S_C
  $K = K - 1$
end while
return Set S_C

After determining the frequency and the population size, the nodes are assigned to each solution. In the first set of d solutions, the nodes $1, 2, \ldots, K$ are assigned to the first solution, the nodes $K+1, K+2, \ldots, 2K$ are assigned to the second solution, etc. The process is repeated until all nodes are assigned to solutions, resulting in d solutions. If N/K is not an integer, random non-repeating genes are selected to fill the last solution. In the second set of d solutions, the nodes $1, 3, \cdots 2K-1$ are assigned to the first solution, the nodes $2K+1, 2K+3, \cdots 4K-1$ are assigned to the second solution, etc. The process continues until f sets of solutions are obtained. By adjusting the increment of nodes when generating different sets of solutions, we ensure that there are no repeated solutions in the initial population. Next, we will present how the two algorithms work in the subsequent steps.

CLASSIC GENETIC ALGORITHM

- **Selection operator** Binary tournament selection, a selection method known for its fast convergence and ease of implementation [97], is employed in this study. Two individuals are randomly sampled from the population, and the tournament is conducted based on their fitness values. The individual with the highest fitness value is selected to proceed with the following operation.

- **Crossover operator** Partial Mapped Crossover (PMX) is employed as the crossover operator in this study. It is recognized as the most commonly used technique for permutation-encoded chromosomes. PMX ensures that the generated offspring do not contain any duplicate genes [95], making it particularly suitable for this context. Compared to other crossover operators, PMX has demonstrated superior performance. The crossover rate is set to $p_c = 0.8$.

- **Mutation operator** The mutation operator is employed to preserve the genetic diversity within the population, thereby preventing the algorithm from converging to a locally optimal solution. Randomly selecting a gene from an individual, the

mutation operator replaces it with another gene that is not present in that individual. This process guarantees that the offspring do not contain any duplicate genes. In this research, the mutation rate is set to $p_m = 0.1$.

The classic genetic algorithm will terminate after the iteration times reaches the preset value. The pseudo-code of the classic genetic algorithm is presented in Algorithm 3.

Algorithm 3 Classic genetic algorithm

Input: network G, number of controllers K, maximum number of iterations MAX
Output: Set S_C

```
Define set S_C as the set of nodes with controllers
Determine the population size P
Initializing the population
Compute the fitness value of each individual
Set iteration counter t = 0
while t < MAX do
    Select P individuals from the population using tournament selection.
    Apply crossover on P/2 pairs of individuals with crossover probability.
    Apply mutation on the offspring with mutation probability.
    New population with size P is generated
    t = t + 1
end while
Add nodes in the best solution in the last iteration to set S_C
return Set S_C
```


HEURISTIC GENETIC ALGORITHM

The heuristic genetic algorithm employed in this research is based on the algorithm proposed by Alp *et al.* for solving the facility location problem, where evolution is facilitated by a greedy heuristic [96]. In the heuristic genetic algorithm, only the crossover operator is utilized. Unlike the traditional crossover method that involves exchanging genes between two individuals to produce offspring, this novel genetic algorithm begins by combining the genes of the two parents. Subsequently, a greedy deletion heuristic is applied to reduce the number of genes until the solution contains exactly K genes. Furthermore, instead of generating a new population at each iteration, the algorithm continuously updates the initial population.

- **Crossover operator**

  Two individuals are randomly selected as parents to initiate the crossover process. The genes of the parents are combined to create a temporary offspring. However, this offspring cannot be directly utilized in the subsequent steps as it contains $2K$ genes, which exceeds the desired number of K genes. To refine the offspring and ensure it consists of only K genes, a removal procedure is applied. The gene removals follow two rules until K genes are kept:

  – The gene that is present twice is kept.

– The gene that contributes least to improving controller reachability is removed.

Although the crossover increases the time demands, the quality of the offspring is improved.

Whenever a new individual is produced through crossover, it undergoes a comparison process with the existing members of the population. If the generated individual is not identical to any of the current population members and exhibits a better fitness value (controller reachability) than the worst fitness value in the population, it will replace the worst individual. This replacement mechanism facilitates the gradual improvement of the average quality of the entire population. Furthermore, the population maintains good diversity as it consists of non-duplicate individuals. The termination condition for the heuristic algorithm is reached when the best fitness value remains unchanged in successive iterations. The pseudo-code of the heuristic genetic algorithm is presented in Algorithm 4.

Algorithm 4 Heuristic genetic algorithm

Input: network G, controllers' number K
Output: Set S_C

```
Determine the population size P
Initializing the population
Compute the fitness value of each individual
Store the best value and the worst value
Set iteration counter t = 0
while t < n do
    Select two individuals from the population randomly
    Apply crossover and generate one offspring
    if offspring not in population then
        Compute the fitness value of offspring
        if fitness value > worst value then
            Update population
        else
            t = t + 1
        end if
    else
        t = t + 1
    end if
end while
Add nodes of the best solution to set S_C
return Set S_C
```

Table 3.2: The number of graphs based on the scaled average reachability values in each interval with $K = 2, 3, 4, 5$

K	(0, 0.25)	(0.25, 0.5)	(0.5, 0.75)	(0.75, 1)
2	31	52	11	6
3	25	48	21	6
4	17	50	27	6
5	8	54	31	7

3.5. RESULTS

3.5.1. HOW DOES PLACEMENT AFFECT CONTROLLER REACHABILITY

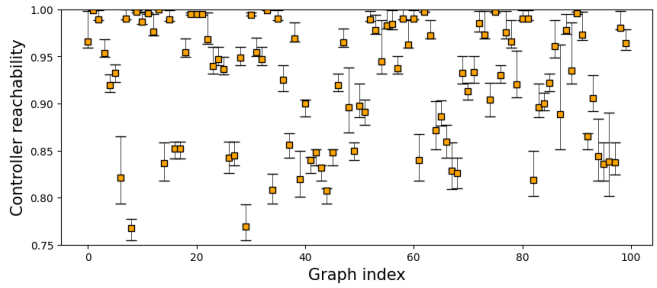
To investigate performance disparities among different controller placement strategies, we conducted a comprehensive analysis of controller reachability for the small-sized real-world networks with the number of nodes ranging from 11 to 30. Specifically, we considered the placement of 2, 3, 4, and 5 controllers with an operational probability p set to 0.99. The resulting controller reachability values were utilized to determine the maximum, minimum, and average reachability, represented by error bars in Figs. 3.6 and 3.7. Analysis of the figure reveals that only a few networks achieve near-optimal controller reachability when randomly placing controllers. For most networks, the average reachability values deviate significantly from the optimal levels. To further substantiate our observations, we employed max-min scaling on our set of 100 graphs to quantify the deviation from the average reachability value (random placements) to the optimal value (optimal placement). The scaling process involved dividing the range between the scaled maximum value (1) and the scaled minimum value (0) into four intervals: (0, 0.25), (0.25, 0.5), (0.5, 0.75), and (0.75, 1). When the scaled value equals 0, the average controller reachability is the farthest from the optimal reachability and when the scaled value equals 1, the average controller reachability is the optimal reachability. The number of graphs falling within each interval based on the scaled average reachability values is shown in Table 3.2.

Our analysis reveals that an increasing number of networks fall within the intervals (0.5, 0.75) and (0.75, 1) as the number of controllers (K) increases, which suggests that the performance of random placement approaches closer to that of the optimal placement with a larger K. However, for most networks, the controller reachability of random placements is still far away from the controller reachability with the optimal placement, regardless of the value of K. Consequently, it is imperative to employ efficient strategies for controller placement to achieve enhanced performance for most networks.

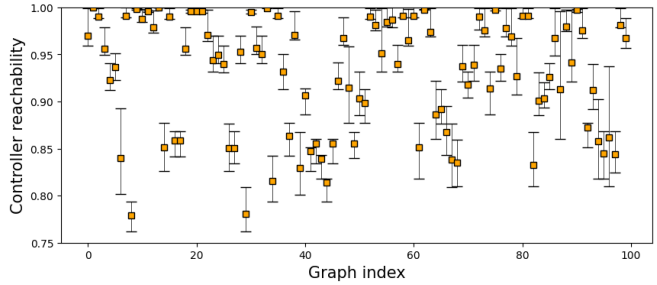
3.5.2. HOW MANY CONTROLLERS ARE NEEDED

To investigate the optimal number of controllers required in the networks, we analyzed our 100 small real-world networks to determine the optimal controller placements for varying values of K, ranging from one to five, as depicted in Fig. 3.8. Although the sizes of the networks are close, the controller reachability with a single controller exhibits a significant range, spanning from 0.75 to 0.9995.

Upon setting a controller reachability threshold of 0.995, we observed that among the analyzed networks, six networks achieved this criterion with a single controller, while 26



(a) $K = 2$



(b) $K = 3$

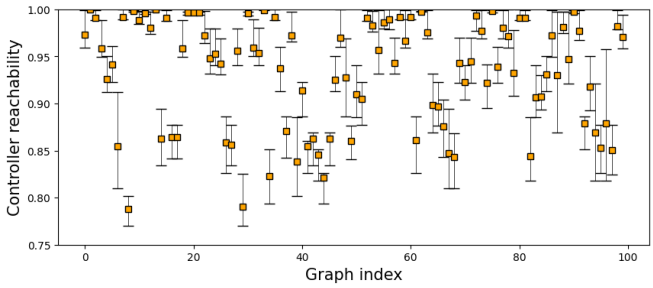
Figure 3.6: Controller reachability error bars for 100 small real-world graphs when $K = 2, 3$ and link operational probability equal to 0.99. The x-axis denotes the index of the graph; the y-axis denotes the controller reachability. Each error bar represents the maximum, minimum, and average controller reachability of all possible placements of K controllers.

networks necessitated two controllers. Furthermore, ten networks met the requirement with three controllers, one network required four controllers, and four networks necessitated five controllers. Notably, 53 networks could not attain a controller reachability of 0.995 even with five controllers placed.

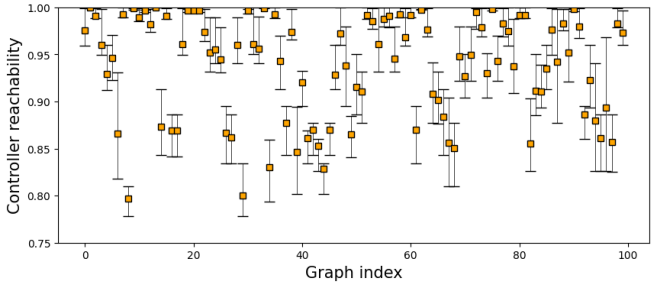
Determining the optimal number of controllers should be based on specific requirements, link probability p, and network topology. It is impractical to establish a universal number that applies to all networks. Nevertheless, our observations indicate that deploying multiple controllers consistently enhances controller reachability across all networks. Notably, when $K = 2$, all the curves show a significant improvement in controller reachability with just two controllers.

3.5.3. COMPARE DIFFERENT PLACEMENT STRATEGIES

To assess the effectiveness of the proposed four placement strategies, we conducted experiments using five medium-sized networks. Considering the large number of nodes in each graph and the observation that optimal placements often involve nodes with low degrees, we investigated the feasibility of utilizing nodes with degrees below the average degree as potential candidates instead of all nodes for controller placements. In addition to the four strategies, we compared the results with random placements



(a) $K = 4$



(b) $K = 5$

Figure 3.7: Controller reachability error bars for 100 small real-world graphs when $K = 4,5$ and link operational probability equal to 0.99. The x-axis denotes the index of the graph; the y-axis denotes the controller reachability. Each error bar represents the maximum, minimum, and average controller reachability of all possible placements of K controllers.

among all nodes, which were generated through 10000 simulations, where the average value was computed over the 10000 realizations. For the HinerniaGlobal network and the Syringa network, we also obtained the optimal placements for different values of K within the potential nodes set, comprising nodes with degrees equal to one and two. The outcomes for the five medium-sized networks are presented in Figs. 3.9 and 3.10. Notably, all four placement strategies yielded satisfactory results for the HinerniaGlobal, Interoute, and GtsCe networks. However, the graph metric-based strategy exhibited relatively poor performance compared to the other methods for the Syringa and Cogentco networks. The discrepancy can be attributed to the influence of network topology.

To elucidate the limitations of the graph metric-based strategy in certain scenarios, we compared its node selection approach with the greedy method, which also involves selecting nodes one by one. Here, we present an example using the Syringa network with ten controllers and a link operational probability of $p = 0.99$, as depicted in Fig. 3.11. In this example, both the greedy algorithm and the graph metric-based strategy initially select node 5 as the first controller. However, as $K = 2$, the subsequent controller selections differ between the two strategies. The greedy algorithm chooses node 18 as the location for the second controller, whereas the graph metric-based method selects node 21. Upon closer examination of the network topology, it becomes evident that the greedy

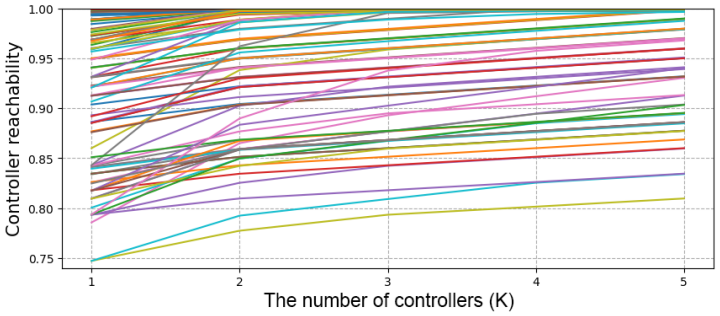
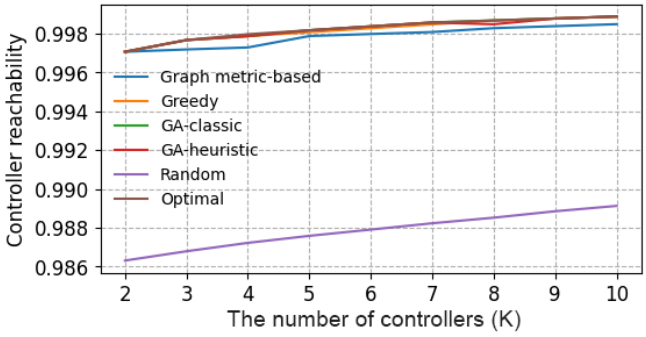
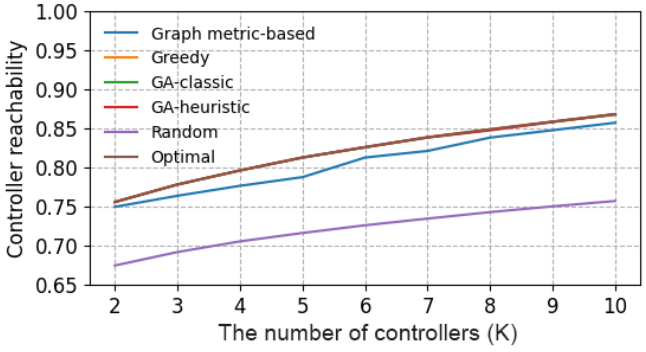


Figure 3.8: The optimal controller reachability of 100 graphs. In this figure, the x-axis denotes the number of controllers, the y-axis denotes the controller reachability. Each curve represents the optimal controller reachability of a graph with $K = 1, 2, 3, 4, 5$ at $p = 0.99$.

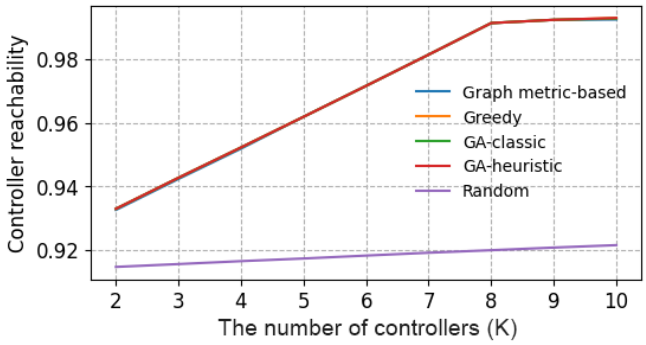


(a) HinerniaGlobal

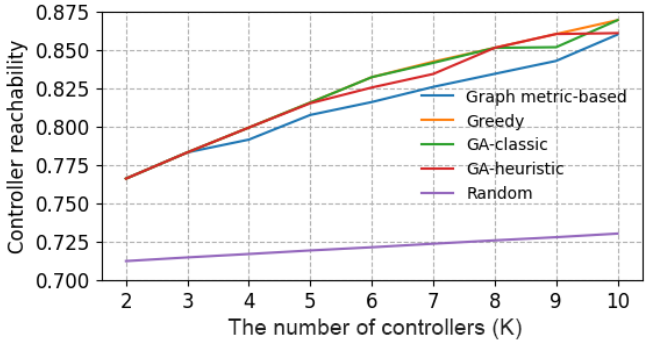


(b) Syringa

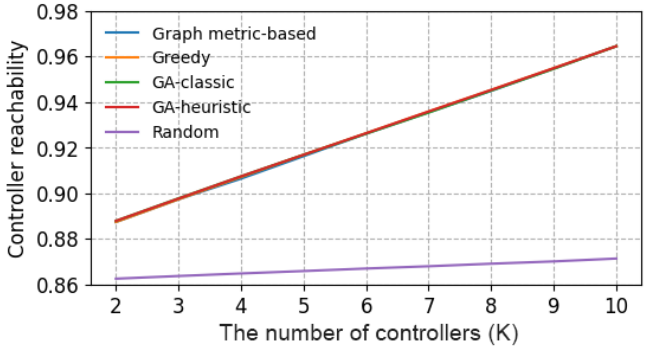
Figure 3.9: Comparison of placement strategies on two of five real-world networks by placing different numbers of controllers with link operational probability p equal to 0.99. The x-axis denotes the number of controllers, the y-axis denotes the controller reachability.



(a) Interoute



(b) Cogentco



(c) GtsCe

Figure 3.10: Comparison of placement strategies on three of five real-world networks by placing different numbers of controllers with link operational probability p equal to 0.99. The x-axis denotes the number of controllers, the y-axis denotes the controller reachability.

algorithm makes a more advantageous choice. Although node 21 has the greatest distance

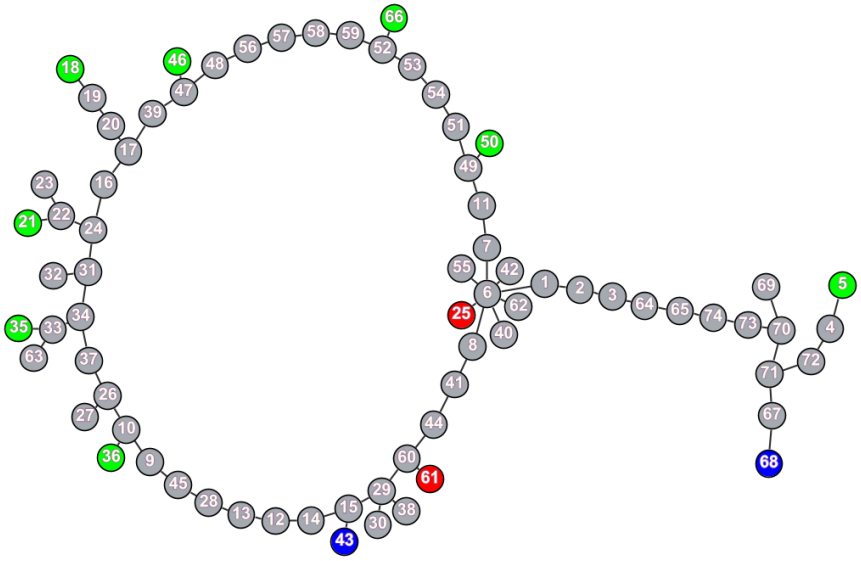


Figure 3.11: Syringa: ten controllers chosen by the graph metric-based strategy and the greedy algorithm. The green nodes are found by both strategies, the red nodes are only found by the greedy algorithm, and the blue nodes are only found by the graph metric-based strategy.

from node 5, with a distance of 31, node 18 is more prone to disconnection as the hop count from node 18 to the "ring" portion of the network is three, which is higher than the hop count from node 21 and the "ring" portion. In the Syringa network, the "ring" portion can be considered relatively more reliable. Consequently, in scenarios where similar situations arise, optimal decisions cannot be solely based on node degree and distance metrics. The provided example serves to emphasize the limitations of the graph metric-based strategy in specific network topologies.

3.6. CHAPTER SUMMARY

In this chapter, we adopt controller reachability as the performance metric for addressing the controller placement problem. Controller reachability is defined as the probability that each node can reach at least one controller. To evaluate controller reachability, we employ the path decomposition algorithm, commonly used for computing all-terminal reliability. By applying this algorithm, we can calculate the controller reachability and quantify the impact of different placements on network performance. Our analysis reveals that the choice of placement strategy can significantly influence controller reachability, highlighting the importance of efficient placement strategies. Moreover, we investigate how many controllers are necessary to ensure satisfactory performance. For small-size networks, introducing a second controller yields substantial improvements in controller reachability for most networks. However, it is important to note that the number of controllers required depends on the specific network topology and the performance re-

quirements in real-world scenarios. Thus, determining the optimal number of controllers necessitates consideration of both network topology and the desired performance levels.

Upon analyzing the optimal placements for various graphs, we have determined that two topological graph metrics, namely degree and distance, heavily influence controller reachability. Building upon this insight, we propose a controller placement strategy that leverages degree and distance as key factors. Additionally, we develop three heuristic algorithms: the greedy algorithm, the classic genetic algorithm, and the heuristic genetic algorithm, to facilitate controller placement. To evaluate the effectiveness of these strategies, we conducted experiments using real-world graphs from the Topology Zoo dataset. The strategy based on graph metrics proves to be a time-efficient approach, delivering controller placements comparable to those obtained using other heuristic methods. Although the strategy based on graph metrics exhibits certain limitations for specific network topologies, its overall performance is satisfactory. Among the three heuristic algorithms, the greedy algorithm demonstrates superior performance across all networks. Both the classic genetic algorithm and the heuristic genetic algorithm are more computationally intensive, yet they do not surpass the performance of the greedy algorithm. Considering both performance and algorithm complexity, we conclude that the strategy based on graph metrics and the greedy algorithm are suitable approaches for addressing the controller placement problem.

Part II: Network recoverability

4

RECOVERABILITY OF NETWORK CONTROLLABILITY WITH RESPECT TO NODE REMOVALS

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

This chapter is based on the published paper [79].

4.1. INTRODUCTION

Network controllability has been extensively investigated [98], particularly due to its applicability to various complex systems that can be represented as networks. These include domains such as power grids [99], transportation systems [100], and telecommunication systems [101]. Controllability represents an important characteristic of such systems, affording the ability to achieve varied control objectives. For example, manipulating approximately 17% neurons in the C. elegans worm can elicit coordinated body responses, while controlling 5% of a swarm of honeybees can guide the swarm to new destinations [102]. However, controllability can falter in the face of malicious attacks or natural catastrophes, resulting from the failure of system components [65]. To improve the resilience of the system against attacks [103], bolstering its robustness becomes paramount. Moreover, there is a pressing need to explore strategies for efficiently restoring failed components to ensure controllability within the system [104].

Network controllability discussed in this research pertains to the concept of structural controllability within directed networks, which do not contain self-loops. In the domain of control theory, a system is considered controllable if it can transition from an initial state to any desired state in a finite time by applying external inputs [105]. Lin introduced the notion of structural controllability [75], where a system exhibiting structural controllability maintains a high likelihood of controllability even after modifying the weights of interconnections. Exploring the intricate interplay between network topology and controllability, Liu *et al.* [24] devised the framework of structural controllability for directed networked systems. This framework focuses on injecting specific external input nodes to achieve full system controllability. Importantly, it is worth noting that network controllability differs from the widely recognized concept of "pinning controllability" [102]. The latter explores methods for driving the system to specific states by manipulating specific nodes. For example, network synchronization explores whether all nodes can exhibit identical dynamic trajectories [106] and investigations of network consensus problems [107] aim to determine strategies to guide all nodes towards the same state.

Errors or attacks within a system can cause a degradation in network performance [108]. Effective and efficient recovery of networks after attacks has gained considerable attention [109]. For example, Shang has explored local recovery strategies from a network percolation perspective [110], as well as strategies for restoring consensus in nonlinear multi-agent systems [104]. Moreover, He *et al.* [32] have defined network recoverability as a network's capacity to revert to a desired state after facing disruptions. In the context of network recoverability with a focus on network controllability, Chen *et al.* [28] explored efficient recovery strategies after random link removals. Their study revealed that the greedy recovery strategy outperforms degree-based and eigenvector-based recovery strategies. Additionally, they introduced an analytical method based on degree distributions to predict network controllability during recovery. However, their investigation did not include the effective recovery strategy of nodes after node failures, a scenario frequently observed in real-life networks. Addressing this gap, our study aims to predict network controllability under random node additions and explore efficient strategies for node recovery in network controllability.

Since network recovery is the reverse process of attacking or disrupting networks,

we can derive various recovery strategies by drawing insights from the attack process. Researchers have explored efficient strategies to undermine network controllability and methods to forecast network controllability under attack scenarios. Targeted attacks are generally more detrimental than random attacks [65]. Pu *et al.* [62] demonstrated that node removals based on node degrees are more harmful than random node attacks in directed Erdös-Rényi (ER) and scale-free (SF) networks. Directed ER networks are synthetic networks generated by randomly placing directed links, while directed scale-free networks are generated to ensure that the degree distributions follow power-law distributions. Wang *et al.* [64] found that intentionally attacking bridge links, whose removal can disconnect the network, effectively disrupts network controllability compared to link removals based on node degrees and distances in directed ER and SF networks. Critical nodes and links are identified based on their propensity to increase the number of driver nodes required for network controllability after removals, where driver nodes are defined as nodes where external inputs are injected [24]. Building on this, Lou *et al.* [111] developed a hierarchical attack framework that incorporates critical nodes or links. In this framework, nodes or links are removed based on categorical priorities, and within the same category, nodes or links with higher centrality values are removed first. Their findings presented that the destructiveness of this attack framework is stronger than strategies that solely leverage centrality features like node degrees or betweenness when targeting nodes or links. Given that attack strategies considering degree and betweenness are extensively investigated, we aim to explore the effectiveness of different degree-based and betweenness-based recovery strategies in terms of network controllability after node removals.

Several techniques have been employed to predict the minimum number of driver nodes required under attack scenarios, including regression models, analytical methods using degree distributions, and machine learning approaches. Sun *et al.* [65] introduced linear regressions for removal fractions less than l_c and quadratic regressions for fractions greater than l_c (where l_c is the fraction of critical links) to approximate the fraction of driver nodes for random and targeted link removals based on critical links. Liu *et al.* [24] proposed an analytical method based on degree distributions to estimate the minimum fraction of driver nodes. Chen *et al.* [28] presented an analytical method using degree distributions for random link removals. Dhiman *et al.* [112] utilized an artificial neural network to predict the minimum number of driver nodes under link-targeted removals, outperforming analytical methods based on critical links. Lou *et al.* [113] predicted controllability robustness under targeted attacks by utilizing convolutional neural networks, which process the adjacency matrix as a grayscale image. The performance of regression models is worse than the analytical methods using degree distributions and machine learning approaches. While machine learning methods require training data, analytical methods offer time and computational cost savings. This encourages us to develop an analytical method based on degree distributions to predict network controllability during the random node recovery process.

In this study, we focus on the recovery process of network controllability after random node removals. We propose an analytical method based on degree distributions to approximate the number of driver nodes required during random node additions. To validate our analytical approach, we apply it to both synthetic and real-world networks. Additionally,

we investigate six other node recovery strategies on synthetic networks: degree-based recovery strategy, betweenness-based recovery strategy, updated degree-based recovery strategy, updated betweenness recovery strategy, greedy degree-based recovery strategy, and greedy betweenness-based recovery strategy. To measure the efficiency of a recovery strategy for network controllability, we utilize two modified recoverability indicators of the recovery process [38], [111].

The remainder of the chapter is the following. Section 4.2 provides a detailed description of the networks utilized in this study. Section 4.3 outlines the attack scenario and the recovery strategies employed, the analytical approximation for network controllability under random node additions, and the two recoverability indicators used in this study. Section 4.4 presents results of the analytical approximation for network controllability under random node additions, and based on the recoverability indicators, we compare and evaluate the efficiency of different recovery strategies on synthetic networks. The chapter is summarized in Section 5.6.

4.2. NETWORK DATA

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

4.2.1. SYNTHETIC NETWORKS

The synthetic networks under investigation comprise directed Erdös-Rényi (ER) networks, swarm signalling networks (SSNs), directed Barabási-Albert (BA) networks and directed power-law (PL) distributed networks.

1. Directed ER networks

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

2. SSNs

   In this study, we employ the topology of SSNs proposed and developed by [71][114]. The SSN exhibits a regular out-degree distribution, while its in-degree distribution follows a Poisson distribution. To generate SSNs, we specify two parameters: the number of nodes N and the out-degree value k. Each node randomly creates k outgoing links to other nodes. Specifically, we generated SSNs with $N = 500$, $k = 2$ and $N = 500$, $k = 4$. The total average degree is 4 and 8, respectively.

3. Directed PL distributed networks

   Directed power-law distributed networks have power-law degree distributions, which are characterized by a specific power-law exponent γ and the minimum value of the degree α. To generate directed power-law distributed networks, we first generate a power-law degree sequence using the Python package *powerlaw* [115].

Next, we use the configuration model [116] to generate a digraph and remove self-loop links. To ensure that the generated network conforms to the power-law distribution, we use the same Python package to fit the degree distributions. We only use generated networks that have a difference between the exponent used and the average fitted power-law exponent of the in-degree and out-degree distribution smaller than 0.01. In this study, we choose two PL distributed networks with 10000 nodes, one with $\gamma = 2.3, \alpha = 3$ and the other one with $\gamma = 3, \alpha = 3$. The average total degrees are around 22 and 10.3, respectively.

Directed BA networks also exhibit power-law degree distributions. To generate a directed BA network, we first generate an undirected BA network [117] by giving two parameters: the number of nodes N and the number of links m that a new node preferentially attaches to existing nodes with high degrees. The initial network is a star network with $m+1$ nodes. Once the undirected BA network is established, we proceed to randomize the orientations of links, thereby transforming the network into a directed structure. We generated directed BA graphs with parameters $N = 500, m = 2$ and $N = 500, m = 4$, respectively, where the average total degree is 4 and 8. As the number of nodes N increases in the directed BA networks with $m = 2$ and $m = 4$, the average power-law exponents of the in-degree and out-degree distributions, obtained using the aforementioned Python package, converge to around 2.9 over 10000 networks. These results are presented in Appendix B, Table B.1. We opted for directed BA graphs with 500 nodes rather than power-law distributed graphs with 10000 nodes to reduce the computational time required for simulations while investigating the performance of different recovery strategies in directed PL distributed networks.

4.2.2. REAL-WORLD NETWORKS

For the real-world networks, we choose 202 communication networks from the Internet Topology Zoo data set [72], whose number of nodes ranges from 11 to 754. To change undirected communication networks into directed networks, based on the node attribute: source node or target node [65], we assign the direction of the link from the source node to the target node. The properties of the 202 communication networks, in terms of the number of nodes, number of links and average degree, are depicted in Fig. 4.1. Apart from the small and medium-sized communication networks, we incorporate an additional seven larger directed networks obtained from the Network Data Repository [118] and the SNAP dataset collection [119]. These selected networks originate from diverse domains, such as the world wide web (WebSpam [120] and Indochina [121]), a Wikipedia adminship election dataset (Wiki Vote [122]), a retweet network dataset (Qatif [123]), an E-mail network dataset (Email Eu core [124]), and internet peer-to-peer network datasets (p2p Gnutella25 [125] and p2p Gnutella08 [126]). Essential details such as the number of nodes (N) and links (L) and the average degree (d_{av}) of these seven larger networks are presented in Table 4.1.

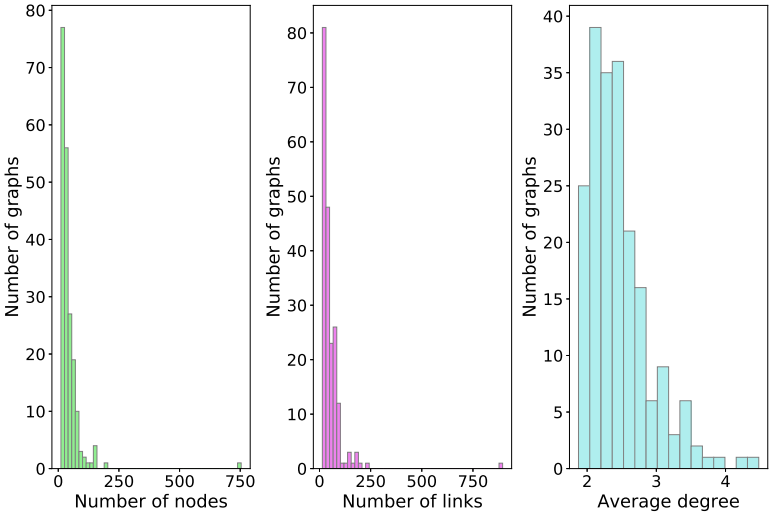


Figure 4.1: Properties of 202 real-world networks from the Internet Topology Zoo data set

Table 4.1: Properties of seven real-world networks

Name	N	L	d_{av}
Qatif [118]	7537	8568	2.274
p2p Gnutella25 [119]	22663	54693	4.827
p2p Gnutella08 [119]	6299	20776	6.597
Indochina [118]	11358	47606	8.383
WebSpam [118]	4767	37375	15.681
Wiki Vote [119]	7066	141779	40.130
Email Eu core [119]	986	46771	94.870

4.3. PRELIMINARIES

4.3.1. ATTACK AND RECOVERY SCENARIOS

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

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

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

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

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

nodes are used.

4.3.2. ANALYTICAL APPROXIMATIONS OF THE NUMBER OF DRIVER NODES

Network controllability has been introduced in Chapter 2 Section 2.3. The analytical approximation for network controllability with respect to random node removals has been described in Chapter 2 Section 2.5.1. In this study, we will denote a network perturbation, either a node removal or a node addition, as a challenge. This study uses challenge K to present the number of manipulations under node removals or additions. A manipulation represents a node removal or a node addition. Challenge K represents that a fraction $p = \frac{K}{N}$ of nodes was removed during the removal process. Hence $K = 0$ corresponds to the graph in the initial state before the attack. Then the minimum fraction of driver nodes at challenge K satisfies

$$\begin{aligned} n_D(K) = \frac{1}{2}\left(1-\frac{K}{N}\right)\{\bar{G}_{in}(\omega_2)+\bar{G}_{in}(1-\omega_1)-2+\bar{G}_{out}(\hat{\omega_2})+\bar{G}_{out}(1-\hat{\omega_1}) \\ +k\left(1-\frac{K}{N}\right)[\hat{\omega_1}(1-\omega_2)+\omega_1(1-\hat{\omega_2})]\}+\frac{K}{N}, \end{aligned} \tag{4.1}$$

satisfying Eq. (2.13).

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

$$\begin{aligned} n_D(K) = \frac{1}{2}\left(1-\frac{2K_a}{N}+\frac{K}{N}\right)\{\bar{G}_{in}(\omega_2)+\bar{G}_{in}(1-\omega_1)-2+\bar{G}_{out}(\hat{\omega_2})+\bar{G}_{out}(1-\hat{\omega_1}) \\ +k\left(1-\frac{2K_a}{N}+\frac{K}{N}\right)[\hat{\omega_1}(1-\omega_2)+\omega_1(1-\hat{\omega_2})]\}+\frac{2K_a}{N}-\frac{K}{N}, \end{aligned} \tag{4.2}$$

satisfying Eq. (2.13).

1. Directed ER networks

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

   $$G_{in}(x) = e^{-k(-x+1)}, G_{out}(x) = e^{-k(-x+1)}. \tag{4.3}$$

   The minimum fraction of driver nodes n_D at challenge K under random removals in the ER networks can be obtained through Eqs. (2.11), (4.1) and (2.13) as

   $$n_D(K) = \frac{K}{N}+\frac{K}{N}\omega_2-\omega_2+\left[1-\frac{K}{N}+k\left(1-\frac{K}{N}\right)^2(1-\omega_2)\right]e^{k\left(1-\frac{K}{N}\right)(\omega_2-1)} \tag{4.4}$$

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

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

$$n_D(K) = \left[1 - \frac{2K_a}{N} + \frac{K}{N} + k\left(1 - \frac{2K_a}{N} + \frac{K}{N}\right)^2 (1-\omega_2)\right] e^{k\left(1-\frac{2K_a}{N}+\frac{K}{N}\right)(\omega_2-1)} + \frac{2K_a}{N} - \frac{K}{N} + \left(\frac{2K_a}{N} - \frac{K}{N}\right)\omega_2 - \omega_2 \tag{4.5}$$

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

2. SSNs

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

$$G_{in}(x) = e^{-k(-x+1)}, G_{out}(x) = x^k. \tag{4.6}$$

Based on Eqs. (2.11), (4.1) and (2.13), the minimum fraction of driver nodes n_D at challenge K under random removals can be calculated by

$$n_D(K) = \frac{K}{N} + \frac{K}{N}\omega_2 - \omega_2 + \left[1 - \frac{K}{N} + (k-1)\left(1 - \frac{K}{N}\right)^2 (1-\omega_2)\right] e^{k\left(1-\frac{K}{N}\right)(\omega_2-1)} \tag{4.7}$$

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

Then the minimum fraction of driver nodes n_D at challenge K under random additions can be obtained by

$$n_D(K) = \left[1 - \frac{2K_a}{N} + \frac{K}{N} + (k-1)\left(1 - \frac{2K_a}{N} + \frac{K}{N}\right)^2 (1-\omega_2)\right] e^{k\left(1-\frac{2K_a}{N}+\frac{K}{N}\right)(\omega_2-1)} + \frac{2K_a}{N} - \frac{K}{N} + \left(\frac{2K_a}{N} - \frac{K}{N}\right)\omega_2 - \omega_2 \tag{4.8}$$

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

3. Directed PL distributed networks

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

$$P_{in}(k_{in}) = C_{in} k_{in}^{-\gamma}, \quad P_{out}(k_{out}) = C_{out} k_{out}^{-\gamma}, \tag{4.9}$$

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

$$G_{in}(x) = \frac{x^a \Phi(x,\gamma,a)}{\zeta(\gamma,a)}, \quad G_{out}(x) = \frac{x^a \Phi(x,\gamma,a)}{\zeta(\gamma,a)}, \tag{4.10}$$

4

where $\Phi(z,s,\alpha)$ is the Lerch transcendent function. Together with Eqs. (2.11), (4.1) and (2.13), the fraction of the minimum fraction of driver nodes n_D at challenge K under random removals can be calculated by

$$\begin{aligned} n_D = & \frac{\left(1-\frac{K}{N}\right)\left(-\frac{K}{N}\omega_2+\frac{K}{N}+\omega_2\right)^a \Phi\left(-\frac{K}{N}\omega_2+\frac{K}{N}+\omega_2,\gamma,a\right)}{\zeta(\gamma,a)} + \\ & \frac{\left(1-\frac{K}{N}\right)\Phi\left(\frac{\left(\frac{K}{N}-1\right)\Phi\left(-\frac{K}{N}\omega_2+\frac{K}{N}+\omega_2,\gamma-1,a\right)\left(-\frac{K}{N}\omega_2+\frac{K}{N}+\omega_2\right)^{a-1}}{\zeta(\gamma-1,a)}+1,\gamma,a\right)}{\zeta(\gamma,a)} \\ & \times\left(\frac{\left(\frac{K}{N}-1\right)\left(-\frac{K}{N}\omega_2+\frac{K}{N}+\omega_2\right)^{a-1}\Phi\left(-\frac{K}{N}\omega_2+\frac{K}{N}+\omega_2,\gamma-1,a\right)}{\zeta(\gamma-1,a)}+1\right)^a \\ & + \frac{k\left(2\frac{K}{N}-1-\left(\frac{K}{N}\right)^2\right)(\omega_2-1)\left(-\frac{K}{N}\omega_2+\frac{K}{N}+\omega_2\right)^{a-1}\Phi\left(-\frac{K}{N}\omega_2+\frac{K}{N}+\omega_2,\gamma-1,a\right)}{\zeta(\gamma-1,a)} \\ & + 2\frac{K}{N}-1, \end{aligned} \tag{4.11}$$

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

$$
\begin{aligned}
n_D = & \frac{\left(1-\frac{2K_a-K}{N}\right)\left(\frac{2K_a-K}{N}(1-\omega_2)+\omega_2\right)^a \Phi\left(\frac{2K_a-K}{N}(1-\omega_2)+\omega_2,\gamma,a\right)}{\zeta(\gamma,a)} + \\
& \frac{\left(1-\frac{2K_a-K}{N}\right)\Phi\left(\frac{\left(\frac{2K_a-K}{N}-1\right)\Phi\left(\frac{2K_a-K}{N}(1-\omega_2)+\omega_2,\gamma-1,a\right)\left(\frac{2K_a-K}{N}(1-\omega_2)+\omega_2\right)^{a-1}}{\zeta(\gamma-1,a)}+1,\gamma,a\right)}{\zeta(\gamma,a)} \times \\
& \left(\frac{\left(\frac{2K_a-K}{N}-1\right)\left(\frac{2K_a-K}{N}(1-\omega_2)+\omega_2\right)^{a-1}\Phi\left(\frac{2K_a-K}{N}(1-\omega_2)+\omega_2,\gamma-1,a\right)}{\zeta(\gamma-1,a)}+1\right)^a \\
& + \frac{\left(\frac{2K_a-K}{N}(1-\omega_2)+\omega_2\right)^{a-1}\Phi\left(\frac{2K_a-K}{N}(1-\omega_2)+\omega_2,\gamma-1,a\right)}{\zeta(\gamma-1,a)} \times \\
& \frac{k\left(\frac{4K_a-2K}{N}-1-\left(\frac{2K_a-K}{N}\right)^2\right)(\omega_2-1)}{\zeta(\gamma-1,a)} + 2\frac{2K_a-K}{N} - 1,
\end{aligned}
\tag{4.12}
$$

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

4.3.3. RECOVERABILITY INDICATORS

To facilitate the comparison of various recovery strategies, recoverability indicators are necessary. One such indicator is the recovery energy E, as proposed by Sun *et al.* [38]. We adopt the recovery energy as a measure of recoverability. The calculation of recovery energy is outlined as follows:

$$
E = \sum_{K=K_a}^{2K_a} n_D(K), \tag{4.13}
$$

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

Taking inspiration from the robustness metric suggested in [127], [111] to assess the effectiveness of attack strategies, our study introduces a recoverability metric denoted by R to quantify the recoverability. This robustness metric quantifies the impact of an attack strategy by averaging the network controllability at each step during an attack [111]. Similarly, we compute the recoverability metric R for a recovery strategy by averaging the network controllability at each step during the recovery process. This allows us to quantify the performance of different recovery strategies with respect to network controllability:

$$
R = \frac{1}{K_a}\sum_{K=K_a}^{2K_a} n_D(K) = \frac{E}{K_a}. \tag{4.14}
$$

Since the value of K_a remains constant for different recovery strategies for a given network, the ranking of recovery strategies remains consistent across both recoverability indicators. The physical significance of the recovery energy lies in its representation of

the total minimum number of required driver nodes throughout the recovery process. A higher recovery energy signifies a greater demand for driver nodes during recovery, whereas a lower recovery energy implies that network controllability can be regained with fewer driver nodes. In essence, recovery strategies with lower recovery energy E or a reduced unique recoverability measure R are deemed more efficient in restoring network controllability.

4.4. RESULTS

4.4.1. VALIDATIONS OF THE ANALYTICAL METHOD

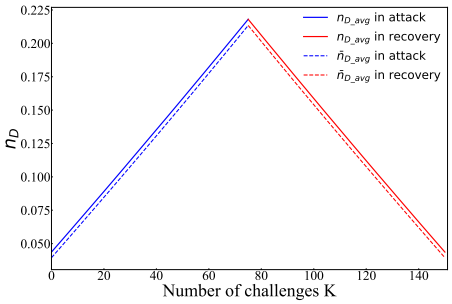
To validate the proposed analytical method for random removals and additions, we conducted simulations on both synthetic and real-world networks to determine the minimum fraction of required driver nodes. Each simulation realization involved a sequence of attacking and recovering the network, with this process repeated 10000 times. During the attack phase, a single node was randomly removed at each step, and the recalculated minimum fraction of necessary driver nodes for network controllability was recorded. Subsequently, in the recovery phase, we reintroduced one removed node along with its original connections to the existing nodes in the network at each step and recalculated the minimum fraction of driver nodes. The recovery process concluded upon the restoration of all initially removed nodes. For synthetic networks characterized by specific parameters, a new network was generated for each simulation realization. However, for real-world networks, the same network was employed across all realizations.

In Fig. 4.2, we present the simulation results and analytical predictions of the proposed method on synthetic networks. The results for random node removals and additions are displayed in blue and red, respectively. The analytical approximations are represented by dashed lines, and the algorithm results are presented with solid lines. Our analysis shows that the analytical approximations accurately predict the minimum fraction of driver nodes in most synthetic networks, except for PL networks with $N = 10000$, $\gamma = 2.3$, and $a = 3$, where a gap is observed between the dashed lines and solid lines.

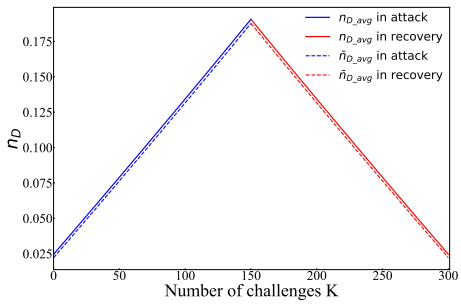
We also observe the discrepancies between simulation and analytical results for three real-world networks, which is depicted in Fig. 4.3 where the solid lines represent the simulation results and the dashed lines represent the analytical results. To further explore the reasons for these discrepancies, we conduct two experiments on each of the three real-world networks.

In the first experiment, we apply the degree preserving rewiring, which maintains the original degree distribution of each network, while randomly rewiring two links in the graph. We then recalculate the minimum fraction of driver nodes and repeat this process for 10000 iterations. We record the minimum fraction of driver nodes for each iteration and present the results in the form of a box plot. In the second experiment, we generate 10000 graphs using the degree distributions obtained from the real-world networks by employing the configuration model [128]. We then calculate the minimum fraction of driver nodes for each generated graph and represent the results as a box plot.

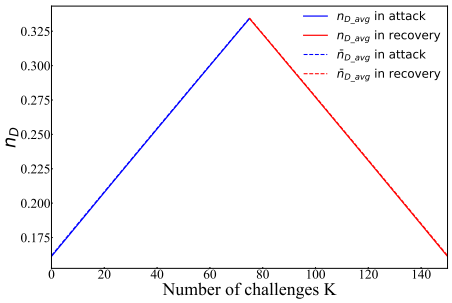
As depicted in Fig. 4.4, the results of both experiments reveal that the minimum fraction of driver nodes can vary for the networks with the same in- and out-degree distributions. Additionally, we observe that the mean value of the minimum number



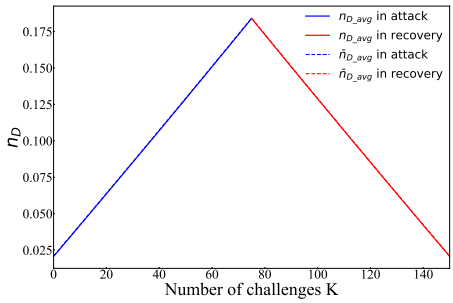
(a) ER(500, 0.007)



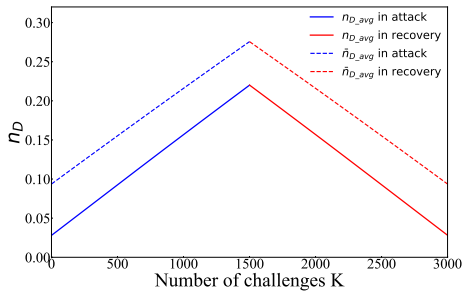
(b) ER(1000, 0.004)



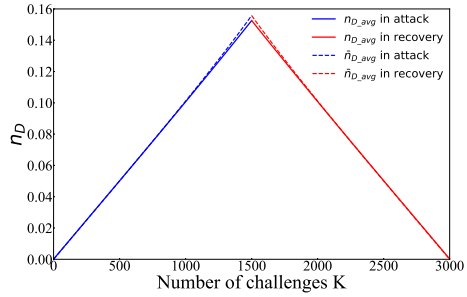
(c) SSN(500, 2)



(d) SSN(500, 4)



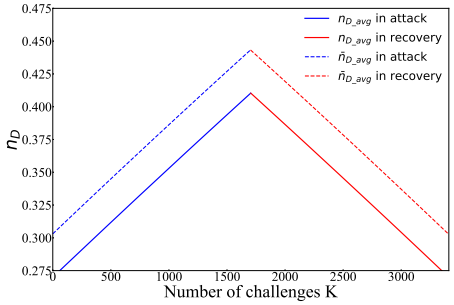
(e) PL(10000, 2.3, 3)



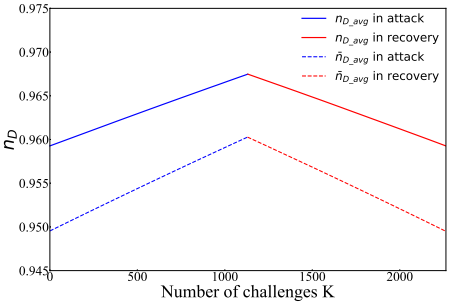
(f) PL(10000, 3, 3)

Figure 4.2: The minimum fraction of driver nodes n_D during random node removals and random node additions in synthetic networks. The blue lines depict node removals, while the red lines represent node additions. The solid lines are obtained by simulations. The blue and red dashed lines are the analytical approximations under random node removals and additions, respectively. We use n_{Davg} to denote the mean minimum fraction of driver nodes in the simulations at each challenge and use $\tilde{n}_{Davg}$ to denote the analytical values of the minimum fraction of driver nodes at each challenge.

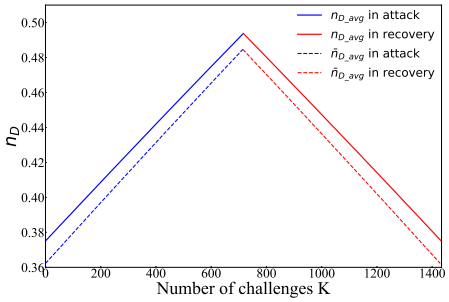
of driver nodes in the networks generated by the configuration model is equivalent to that obtained using the analytical approximation method. It should be noted that the analytical approximation method represents the expected value of the minimum number of driver nodes for graphs that have the same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



(a) Indochina



(b) Qatif

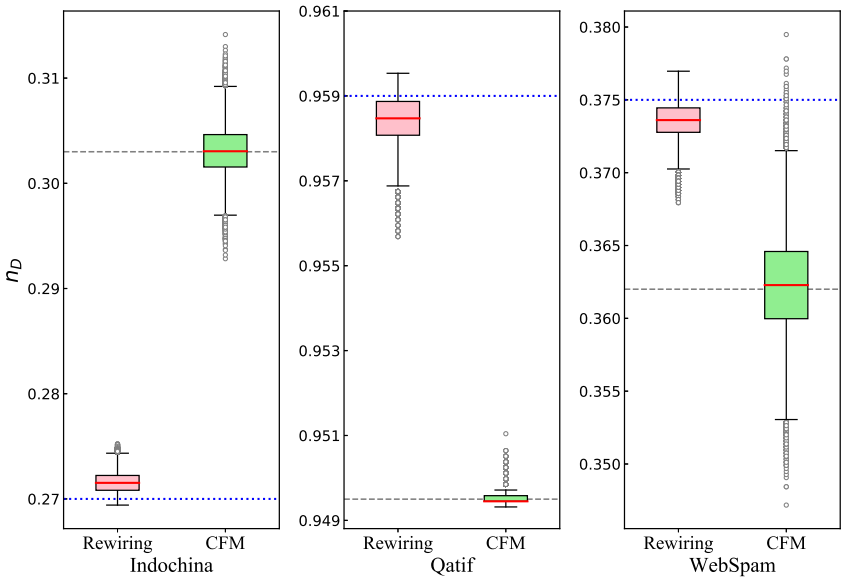


(c) WebSpam

Figure 4.3: The minimum fraction of driver nodes n_D during random node removals and random node additions in three real-world networks. The blue lines depict node removals while the red lines represent node additions. Both are obtained by the maximum matching algorithm over 10000 realizations. The blue dashed lines represent the analytical approximations under node removals and the red dashed lines represent the analytical approximations under node additions. n_{Davg} presents the mean minimum fraction of driver nodes using the algorithm at each challenge and $\tilde{n}_{Davg}$ presents the analytical values of the minimum fraction of driver nodes at each challenge.

4.4.2. ANALYTICAL METHOD WITH SHIFTING

In order to reduce the discrepancies between the predicted and simulated values, we propose to adjust our original analytical model by applying a shift. First, we determine



(a) Two experiments on three real-world graphs

Figure 4.4: Rewiring links and generating graphs using the configuration model can yield different values for the minimum fraction of driver nodes. "Rewiring" presents rewiring results, and "CFM" represents results for the configuration model. The analytical results obtained using the in-degree and out-degree distributions are represented by the grey dashed lines. The values of the minimum fraction of driver nodes calculated by the maximum matching algorithm for the real-world networks are represented by the blue dotted lines. The mean minimum fraction of driver nodes calculated by the algorithm for networks after rewiring and for networks generated by using the configuration model are indicated by the red lines.

the exact value of the minimum fraction of driver nodes $n_D[0]'$ by applying the maximum matching algorithm. The shifting term β is then calculated as the difference between $n_D[0]'$ and the original analytical approximation $n_D[0]$, i.e., $\beta = n_D[0]' - n_D[0]$. Consequently, if the shifted analytical result of the minimum fraction of driver nodes at a particular challenge k is denoted as $n_D[k]'$, the shifted result $n_D[k]'$ can be calculated as follows:

$$n_D[k]' = \beta + n_D[k]. \tag{4.15}$$

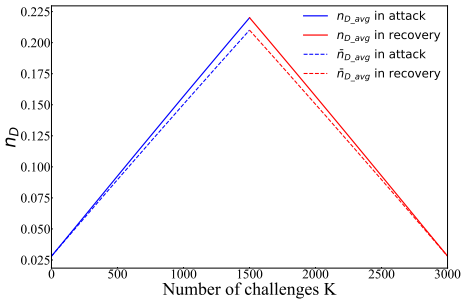
It should be noted that the analytical method using shifting will incur an additional computational cost due to the use of the maximum matching algorithm, which has a time complexity of $O\left(L\sqrt{N}\right)$ [76]. Here, L denotes the number of links in the network and N represents the number of nodes in the network.

We show the results of the adjusted model for three real-world networks and PL(2.3, 3) in Fig. 4.5, where the prediction results are much better than those before shifting. Then we validate the shifting method using the dataset comprising 202 small-scale real-world networks from the Topology Zoo and seven large-scale networks. Validation involves calculating the absolute mean error (AME) and root mean square error (RMSE) between the results obtained from simulations and analytical results, both before and after applying shifting. AME is defined as the absolute difference between the simulation and the analytical results. Additionally, we calculate the proportion of challenges where RMSE was smaller than 5%, denoted as $P_{RMSE\leq 0.05}$. We compare the results before and after shifting to demonstrate the effectiveness of the method.

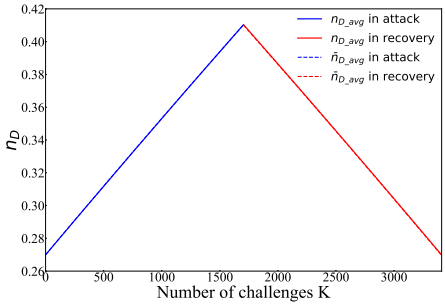
We present the results of the shifted model for the Topology Zoo dataset in a histogram (Fig. 4.6), where the results before and after shifting are depicted in orange and blue, respectively. Furthermore, we observe an improvement in the approximations for the seven large graphs by comparing the results obtained before and after shifting (Table 4.2). The results for the shifted model exhibit smaller AME and RMSE values and higher $P_{RMSE\leq 0.05}$ values, thus indicating the effectiveness of the shifting method. Besides using the shifting term $\beta = n_D[0]' - n_D[0]$, we can also attempt to use the ratio of $n_D[0]'$ and $n_D[0]$ as a scaling factor, i.e. $\gamma = \frac{n_D[0]'}{n_D[0]}$, to construct a rescaled model $n_D[k]' = \gamma n_D[k]$. However, the results for the rescaled model for the Topology Zoo and seven large-scale networks are less good than those for the shifted model. This discrepancy could be attributed to the rescaled model's heightened sensitivity to scaling factors at each point, as opposed to the fixed modifications offered by the shifted model. The results for the rescaled model are reported in Appendix B.

4.4.3. THE EFFICIENCY OF RECOVERY STRATEGIES

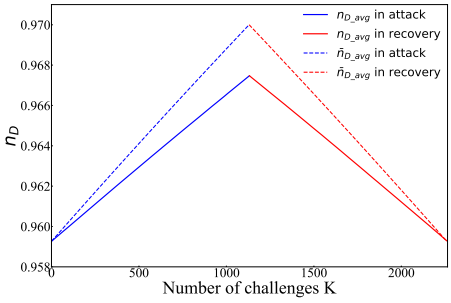
We adopt two recoverability indicators to assess the effectiveness of distinct recovery strategies. For each synthetic network category with different sets of parameters, such as the directed ER network with parameters $N = 500$, $p_{ER} = 0.007$, we generate a total of 10000 networks. For each network instance, we proceed by randomly removing up to 15% of the nodes and subsequently employ diverse recovery strategies to restore the network. During the recovery phase, we reintroduce one node at each step in accordance with the chosen recovery method. We then recalibrate the minimum fraction of driver nodes required for network controllability until all previously eliminated nodes are reinstated.



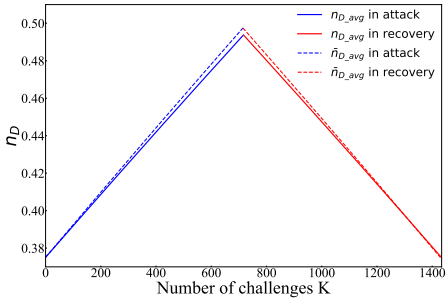
(a) PL(1000, 2.3, 3)



(b) Indochina



(c) Qatif



(d) WebSpam

Figure 4.5: The minimum fraction of driver nodes n_D during random node removals and random node additions in three real-world networks and one PL network after shifting. The blue lines depict node removals, while the red lines represent node additions. Simulation results (solid lines) are obtained by the maximum matching algorithm over 10000 realizations. The blue dashed lines are the shifted analytical approximations under node removals, while the red dashed lines are the shifted analytical approximations under node additions. n_{Davg} denotes the mean minimum fraction of driver nodes in the simulations at each challenge and $\tilde{n}_{Davg}$ denotes the shifted analytical values of the minimum fraction of driver nodes at each challenge.

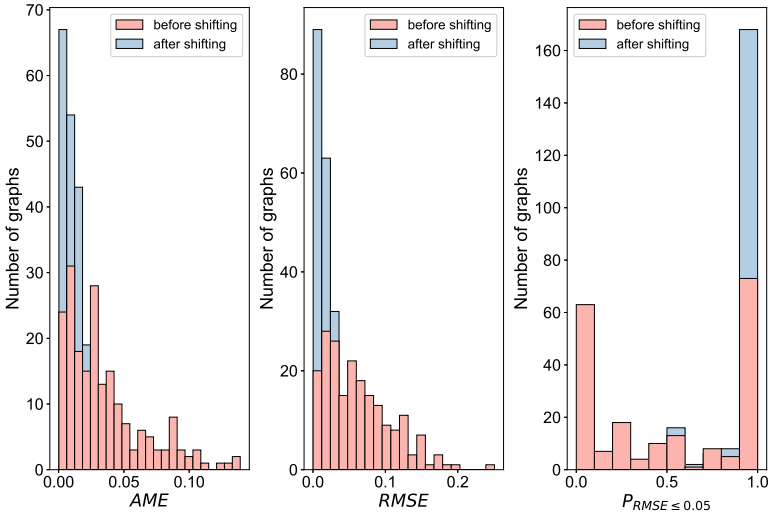
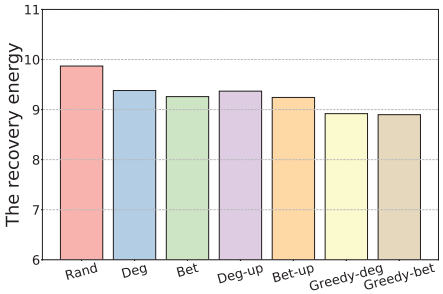


Figure 4.6: Results before and after shifting for 202 networks from the Topology Zoo data set. Clearly the shifted model exhibits better performance for this data set.

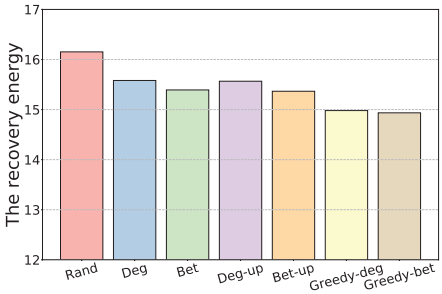
Table 4.2: Results for the shifted model for 7 large scale real-world networks

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

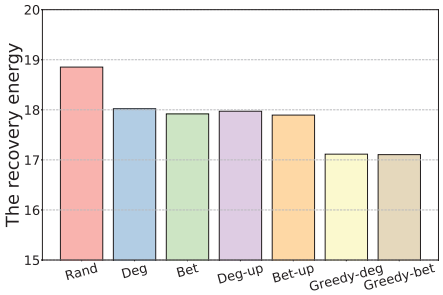
Next, we compute the mean value of the minimum fraction of driver nodes across the 10000 networks for each specific recovery strategy at every step. Subsequently, we sum these mean values to derive the recovery energy associated with the recovery strategy. For the various recovery strategies employed on synthetic networks, we present the corresponding recovery energy in Fig. 4.7. The recoverability metric R is summarized in Table 4.3. In the case of directed ER networks and SSNs, the greedy-betweenness recovery strategy demonstrates the lowest recovery energy, followed by the greedy-degree recovery strategy. The remaining recovery strategies, ranked in order of increasing recovery energy, are updated betweenness recovery strategy, updated degree recovery strategy, betweenness-based recovery strategy, degree-based recovery strategy, and random recovery strategy. Regarding the directed BA network, the degree-related recovery strategies outperform the betweenness-related recovery strategies. The order of recovery energy ranking for different recovery strategies, from lowest to highest, is as follows: greedy-degree recovery strategy, greedy-betweenness recovery strategy, updated degree



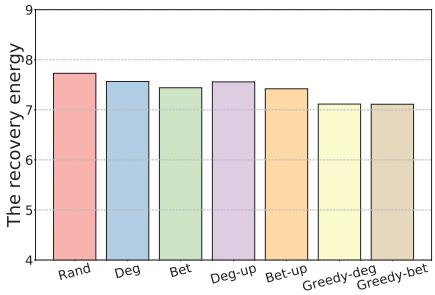
(a) ER(500, 0.007)



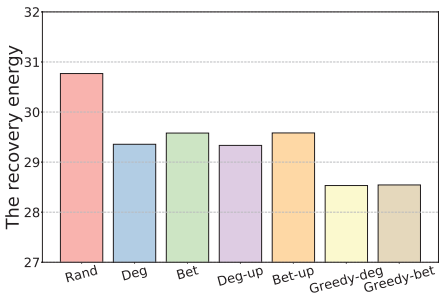
(b) ER(1000, 0.004)



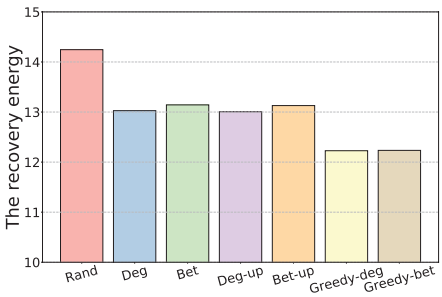
(c) SSN(500, 2)



(d) SSN(500, 4)



(e) BA(500, 2)



(f) BA(500, 4)

Figure 4.7: Recovery energy for different recovery strategies in synthetic networks. Bar "Rand" presents the recovery energy of random recovery strategy; Bar "Deg" presents the recovery energy of degree-based recovery strategy; Bar "Bet" presents the recovery energy of betweenness-based recovery strategy; Bar "Deg-up" presents the recovery energy of updated degree recovery strategy; Bar "Bet-up" presents the recovery energy of updated betweenness recovery strategy; Bar "Greedy-deg" presents the recovery energy of greedy-degree recovery strategy and Bar "Greedy-bet" presents the recovery energy of greedy-betweenness recovery strategy. The numbers above each bar demonstrate different recovery strategies' values of recovery energy.

recovery strategy, updated betweenness recovery strategy, degree-based recovery strategy, betweenness-based recovery strategy, and random recovery strategy. Indeed, it is worth highlighting that the performance improvements brought about by the updated degree (or betweenness) recovery strategy are not substantial when compared to the performance of the corresponding degree-based (or betweenness-based) recovery strategy. On the other hand, the greedy-degree (or greedy-betweenness) recovery strategy significantly enhances performance in comparison to the degree-based (or betweenness-based) recovery strategy. The recovery strategy outcomes for small-sized networks, as presented in Appendix B.3, are consistent with the results discussed here.

Table 4.3: The recoverability metric R for different recovery strategies for different kinds of synthetic networks. "Rand" is an abbreviation of random recovery strategy; "Deg" is an abbreviation of degree based recovery strategy; "Bet" presents betweenness based recovery strategy; "Deg-up" is an abbreviation of updated degree based recovery strategy; "Bet-up" presents updated betweenness recovery strategy; "Greedy-deg" is an abbreviation of greedy-degree recovery strategy; "Greedy-bet" presents greedy-betweenness recovery strategy. The bold values represent the lowest recoverability metric for each type of graph

Name	Rand	Deg	Bet	Deg-up	Bet-up	Greedy-deg	Greedy-bet
ER(500, 0.007)	0.12986	0.12343	0.12183	0.12326	0.12161	0.11735	**0.11708**
ER(1000, 0.004)	0.10697	0.10319	0.10193	0.10309	0.10176	0.09921	**0.09891**
SSN(500, 2)	0.24807	0.23711	0.23576	0.23647	0.23545	0.22519	**0.22506**
SSN(500, 4)	0.10169	0.09955	0.09790	0.09944	0.09763	0.09361	**0.09357**
BA(500, 2)	0.40485	0.38627	0.38921	0.38597	0.38925	**0.37542**	0.37558
BA(500, 4)	0.18744	0.17141	0.17293	0.17113	0.17275	**0.16089**	0.16098

4.5. CHAPTER SUMMARY

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

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

5

RECOVERABILITY OF POWER GRIDS OPTIMIZING THE DC POWER FLOW

We investigated efficient strategies for the recovery of individual links in power grids governed by the direct current (DC) power flow model, under random link failures. Our primary objective was to explore the efficacy of recovering failed links based solely on topological network metrics. In total, we considered 13 recovery strategies, which encompassed 2 strategies based on link centrality values (link betweenness and link flow betweenness), 8 strategies based on the products of node centrality values at link endpoints (degree, eigenvector, weighted eigenvector, closeness, electrical closeness, weighted electrical closeness, zeta vector, and weighted zeta vector), and 2 heuristic strategies (greedy recovery and two-step greedy recovery), in addition to the random recovery strategy. To evaluate the performance of these proposed strategies, we conducted simulations on three distinct power systems: the IEEE 30, IEEE 39, and IEEE 118 systems. Our findings revealed several key insights: Firstly, there were notable variations in the performance of the recovery strategies based on topological network metrics across different power systems. Secondly, all such strategies exhibited inferior performance when compared to the heuristic recovery strategies. Thirdly, the two-step greedy recovery strategy consistently outperformed the others, with the greedy recovery strategy ranking second. Based on our results, we conclude that relying solely on a single metric for the development of a recovery strategy is insufficient when restoring power grids following link failures. By comparison, recovery strategies employing greedy algorithms prove to be more effective choices.

This chapter is based on the published paper [129].

5.1. INTRODUCTION

The power grid system is vulnerable to disruptions caused by manufacturing defects, natural disasters and human actions such as terrorist attacks [130][131][132]. Even minor initial disturbances can lead to severe consequences, including economic and social instability [133][134][135]. To reduce the costs associated with the disruptions, one research direction is to enhance the robustness of the power grid to withstand perturbations. The other direction is to propose efficient restoration strategies.

The power grid can be represented as a complex network, where nodes represent generators, loads and transformers, and links represent transmission lines. Analyzing the power grid's robustness from a network perspective has attracted significant attention. Network robustness is typically evaluated by assessing changes in network performance due to fluctuations such as node removals or link removals [13]. Identifying critical nodes and links within the power grid can lead to strategies for enhancing robustness by protecting the vital components [136][137][138]. Besides, robust failure responses like partitioning grids into islands can mitigate the effects of components disruptions [139][140]

Numerous studies focus on strategies for power grid recovery [39], from black-start [40], like an optimal generator start-up strategy by solving a mixed integer linear programming (MILP) problem [41] or a method that partitions the network to restore parts of the power grid separately and then interconnects them afterward [141]. Additionally, research has been conducted on recovering power grids after partial component failures. Machine learning methods, specifically reinforcement learning, have been developed for restoring networks after node failures [42] and link failures [43]. The machine learning methods have demonstrated better performance to node recovery strategies based on node degrees or node loads and link recovery strategies based on link betweenness. Li *et al.* [142] developed the Q-learning method to find an optimal method to recover the power grids with link failures. From a network perspective, Wu *et al.* [143] develop an effective tool for the sequential recovery graph to recover the nodes in the power grids which perform better than recovery strategies based on node degree and node loads. Forming microgrids can improve the resilience of the system after blackouts, Mosaye *et al.* [144] apply deep reinforcement learning to establish microgrids from black-start after blackouts to restore service in distribution networks. Wei-Chang *et al.* [145] developed an enhanced genetic algorithm to minimize costs while optimizing dispatch in stand-alone microgrid systems.

Despite extensive research on power grid recovery, there is still a lack of investigation into the effectiveness of recovery strategies that rely solely on different network metrics following transmission line failures. In this study, our main contributions are as follows,

1. We examine recovery strategies based on various network metrics, including degree, betweenness, flow betweenness, eigenvector centrality, weighted eigenvector centrality, closeness, electrical closeness, electrical weighted closeness, zeta vector centrality, and weighted zeta vector centrality. Additionally, we compare these strategies with the random recovery, greedy, and two-step greedy strategies.

2. To assess the effectiveness of recovery methods, we utilize the general recoverability framework proposed by He *et al.* [32] to measure power grid recoverability in the context of random link removals, where recoverability signifies a network's ability to return to a predefined desired performance level.

3. Our study does not consider cascading failures after the recovery or removal of a single transmission line. Instead, we use the Direct Current (DC) power flow model to maximize power flow satisfaction for loads after a link removal or addition.

The chapter is structured as follows: Section 5.2 provides an overview of network robustness, laying the foundation for the subsequent analyses. Section 5.3 details the modeling of power grids, including how to transfer a power grid into a network and optimize the DC power flow. Section 5.4 presents the attack and recovery processes with a focus on the strategies employed. The results and summary are presented in Sections 5.5 and 5.6, respectively.

5.2. PRELIMINARY FOR NETWORK ROBUSTNESS

5.2.1. R-VALUE AND CHALLENGES

In a framework for computing network robustness [13], network robustness is interpreted as a measure of the network's response to perturbations such as failures or attacks. The robustness value R was proposed to measure the performance of a network at a certain time, which is related to the function of the network, i.e., the type of service the network is supposed to support, such as road transport, neuron transport or the spreading of news on a social network. The R-value is normalized in the range $[0,1]$. Here, $R = 0$ corresponds to a network completely lacking robustness, while $R = 1$ corresponds to an optimally robust network.

We assume that perturbations are imposed on a network through a number of elementary changes [13][32][146]. We denote the total number of the changes by K. An elementary change in a network is defined as an event that changes the topology of a network, such as a link addition, a link removal or a change in the link weight. An elementary change in the network may result in a change of the R-value. In this paper, we consider elementary changes as the random removal of a link during the attack phase and the recovery of a link previously removed in the attack phase. Additionally, we denote the number of challenges in the attack process as K_a and in the recovery process as K_r.

5.2.2. RECOVERABILITY INDICATOR OF A RECOVERY STRATEGY

The performance of a power grid at any time can be captured by a specific R-value, which possibly changes when an elementary change occurs. The R-value at challenge k is denoted as $R[k]$. The areas under the R-value line during the processes can reflect the effectiveness of the attack strategy or the recovery strategy [38]. We define the attack strength S_a and the recovery strength S_r satisfy the following relations:

$$S_a = \sum_{k=0}^{K_a} R[k], \tag{5.1}$$

and

$$S_r = \sum_{k=K_a}^{K_a+K_r} R[k], \tag{5.2}$$

where K_a is the total number of challenges in the attack process and K_r is the total number of challenges in the recovery process. Further, we define the recoverability energy ratio η

as follows:

$$\eta = \frac{S_r}{S_a}. \tag{5.3}$$

Recoverability energy ratio η presents the efficiency of the recovery strategy with respect to the attack strategy. The larger the recoverability energy ratio η is, the more efficient the recovery strategy is.

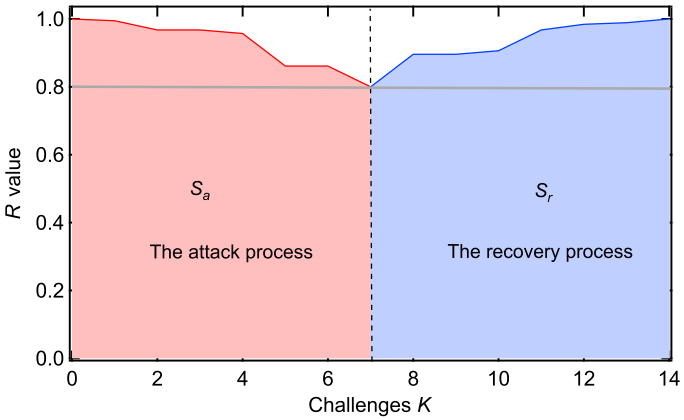


Figure 5.1: Illustration of the attack process and the recovery process in a power grid: one realization with a threshold equal to 0.8.

5.3. MODELING POWER GRIDS

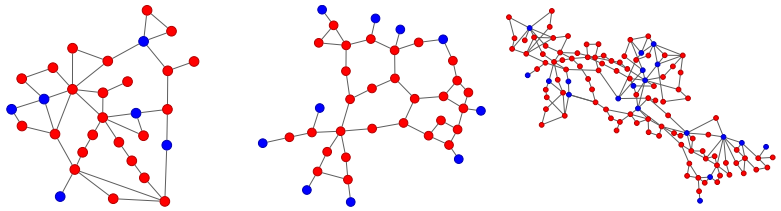
5.3.1. NETWORK MODEL OF POWER GRIDS

A power grid can be represented by a graph $\mathcal{G}$, where the set of N nodes is denoted as $\mathcal{N}$ and the set of L links is denoted as $\mathcal{L}$. For an unweighted and undirected graph, the symmetric adjacency matrix A has elements $a_{ij} = 1$ if the nodes i and j are connected by a link $l_{ij} \in \mathcal{L}$, otherwise $a_{ij} = 0$. Alternatively, we can also model a power grid as a weighted graph. The link weight is related to the impedance of transmission line l_{ij}, which will be denoted as y_{ij}. The weighted symmetric adjacency matrix $\widetilde{A}$ has elements $\widetilde{a}_{ij} = \frac{1}{y_{ij}}$ if a link $l_{ij} \in \mathcal{L}$ exists, otherwise $\widetilde{a}_{ij} = 0$ [147].

In this study, we choose three power grids [148][149], IEEE 30 [150], IEEE 39 [151][152][153] and IEEE 118 [154]. We present the number of nodes N, the number of links L and the average degree d_{av} of the unweighted power grids in Table 5.1.

Table 5.1: Properties of three considered power grids

Name	N	L	d_{av}
IEEE 30	30	41	2.73
IEEE 39	39	46	2.36
IEEE 118	118	179	3.03



(a) IEEE 30 (b) IEEE 39 (c) IEEE 118

Figure 5.2: Three power grids. The red nodes represent loads and the blue nodes represent generators.

5.3.2. PERFORMANCE OF POWER GRIDS

As proposed by Cetinay *et al.* [155], we employ as R-value in a power grid, the ratio of the total satisfied demand to the total initial demand, called yields. At challenge k, we denote the amount of satisfied demand at bus i by $L_i[k]$ and the amount of supply at bus i by $G_i[k]$. Therefore the injected power P_i at node i satisfies $P_i[k] = G_i[k] - L_i[k]$. Since the total demand matches the supply in a power grid, it holds that $\sum_{i=1}^{N} L_i[k] = \sum_{i=1}^{N} G_i[k]$. At challenge $k = 0$, the power grid has not been attacked and the initial demand of bus i is $L_i[0]$. Then yields, the R-value at a challenge k, can be calculated by

$$R[k] = \frac{\sum_{i=1}^{N} L_i[k]}{\sum_{i=1}^{N} L_i[0]} \tag{5.4}$$

5.3.3. OPTIMIZING THE DC POWER FLOW MODEL

Compared with cascading failure models in power grids, we consider a power flow redistribution mechanism to achieve a steady state by optimizing the total satisfied demand value. The mechanism adjusts the supply and demand of buses using a linear programming method to ensure that line flows are below the line capacity. Specifically, the objective of the proposed model is to optimize the total satisfied demand while satisfying several constraints, including (1) the power grid satisfies Kirchhoff's Law and Ohm's Law; (2) the total supply matches the total demand at each challenge; (3) the supply and demand of a bus are not beyond its initial values; (4) the absolute value of a line flow is not beyond the line capacity. The DC power flow model [155] at challenge k obeys the optimization problem:

$$\begin{aligned}
\text{minimize} \quad & -\sum_{i=1}^{N} G_i[k] \\
\text{subject to} \quad & \boldsymbol{G}[k] - \boldsymbol{L}[k] = \widetilde{Q}[k]\boldsymbol{\Theta}[k], \\
& \boldsymbol{F}[k] = \widetilde{B}^T[k]\boldsymbol{\Theta}[k], \\
& \sum_{i}^{N} G_i[k] - \sum_{i}^{N} L_i[k] = 0, \\
& \mathbf{0} \le \boldsymbol{G}[k] \le \boldsymbol{G}[0], \\
& \mathbf{0} \le \boldsymbol{L}[k] \le \boldsymbol{L}[0], \\
& -\boldsymbol{C} \le \boldsymbol{F}[k] \le \boldsymbol{C}.
\end{aligned} \tag{5.5}$$

The weighted *Laplacian* matrix is $\widetilde{Q}[k] = \widetilde{\Delta}[k] - \widetilde{A}[k]$ where $\widetilde{\Delta}[k]$ is the weighted degree matrix and $\widetilde{A}[k]$ is the weighted adjacency matrix; $\boldsymbol{\Theta}$ is the $N \times 1$ vector with elements θ_i, which presents the phase angle of bus i; $\widetilde{B}[k]$ is the $N \times L$ weighted incidence matrix with elements

$$\widetilde{b}_{il} = \begin{cases} \widetilde{a}_{ij} & \text{if link } e_l = i \rightarrow j, \\ -\widetilde{a}_{ij} & \text{if link } e_l = i \leftarrow j, \\ 0 & \text{otherwise,} \end{cases} \tag{5.6}$$

and $\widetilde{Q}[k] = \widetilde{B}[k]\widetilde{B}[k]^T$.

The supply and demand vectors $\boldsymbol{G}[k]$ and $\boldsymbol{L}[k]$ include the supply $G_i[k]$ and demand $L_i[k]$ of each bus i, respectively. The initial supply and demand values of all buses are stored in the vectors $\boldsymbol{G}[0]$ and $\boldsymbol{L}[0]$ at challenge $k = 0$. The active power flow vector $\boldsymbol{F}[k]$ has L elements, each representing the power flow $f_{ij}[k]$ of a transmission line l_{ij} connecting buses i and j. The capacity vector $\boldsymbol{C}$ is a vector with L elements and each element c_{ij} representing the capacity of each transmission line l_{ij}.

For the power grid, the line capacity c_{ij} is defined as $\alpha \boldsymbol{f_{ij}}[0]$ where flow vector $\boldsymbol{F}[0]$ is the active power flow of each line at the initial stage (at challenge $k = 0$) and α is the tolerance level [156].

5.4. THE ATTACK AND RECOVERY PROCESS

5.4.1. THE ATTACK PROCESS

In the attack process, we select links uniformly at random to be removed iteratively until the R value of the power grid falls below a predetermined threshold. The flowchart of the attack process is shown in Fig. 5.3(a).

5.4.2. THE RECOVERY PROCESS

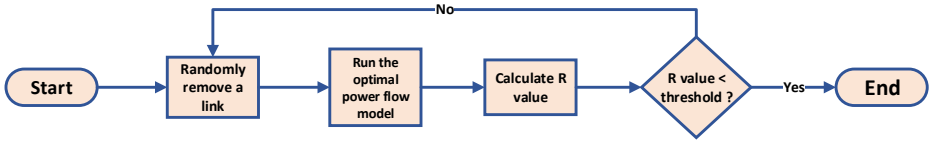
In the recovery process, links previously removed from the power grid are gradually added back individually, with the order of additions determined by a chosen recovery strategy. The flowchart of the recovery process is presented in Fig. 5.3(b). We explore the efficiency of thirteen different recovery strategies introduced in the following section. It is important to note that strategies based on topological network metrics are calculated based on the original network configuration before the attack process. The flow chart of recovery strategies based on network metrics is presented in Fig. 5.4.

RANDOM RECOVERY STRATEGY (RAND)

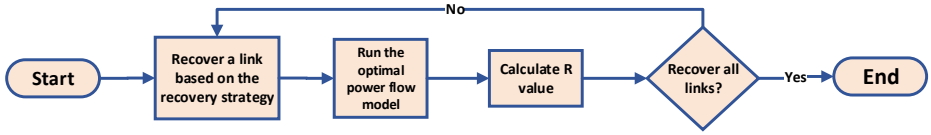
A link is randomly selected to add to the power grid one by one from the set of links that were removed during the attack process.

GREEDY AND TWO-STEP GREEDY RECOVERY STRATEGIES (GREEDY AND TWOGREEDY)

The greedy recovery strategy is to optimize the performance of the power grid at each stage of the recovery process. The strategy selects the link that yields the largest R-value of the power grid in each step and adds the link to the grid. The greedy recovery strategy flow chart is depicted in Fig. 5.5(a).



(a) The attack process



(b) The recovery process

Figure 5.3: The flowcharts of the attack process and the recovery process.

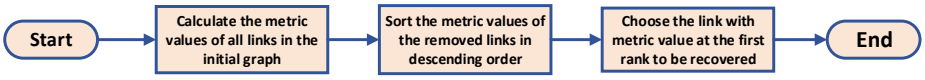
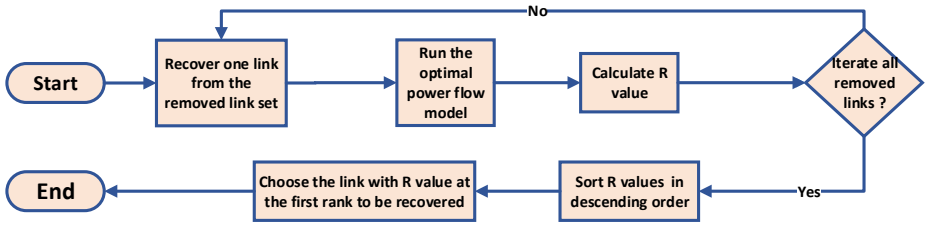


Figure 5.4: The flow chart of recovery strategies based on topological network metrics.

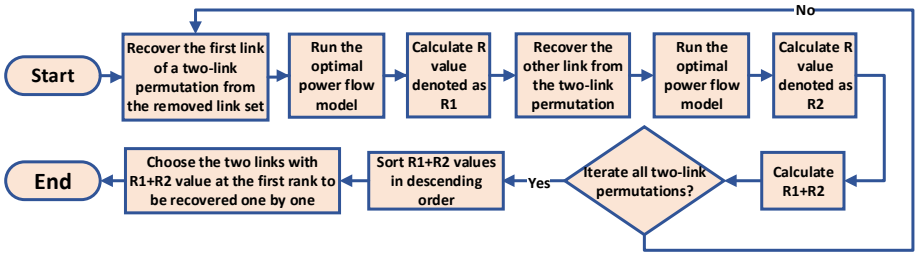
In contrast to the greedy recovery strategy, which focuses on identifying a single link that significantly improves the R-value in one step, an improved approach involves selecting n links that collectively maximize the summation of the R-value in n steps, which can be achieved by exhaustively enumerating all permutations of n links from the set of potential links to be added. By considering multiple steps, more information is incorporated, resulting in potentially better solutions for the n-step greedy recovery strategy. However, if the value of n becomes excessively large, the computational cost becomes prohibitive. Therefore, in this study, we have chosen to investigate the performance of the two-step greedy recovery strategy. Specifically, the two-step greedy strategy aims to determine two links to be added sequentially, optimizing the sum of the R-values achieved in the two steps. We present the flowchart of the two-step greedy strategy in Fig. 5.5(b).

DEGREE RECOVERY STRATEGY (DEGREE)

The degree d_i is the number of neighboring nodes of node i, which can be used to measure the node's importance. By using the adjacency matrix A, the calculation of the



(a) Greedy recovery strategy



(b) Two-step greedy recovery strategy

Figure 5.5: The flowcharts of the greedy and two-step greedy strategies.

degree is $d_i = \sum_{j=1}^{N} a_{ij}$. To evaluate the significance of a link, we can use the product d_{ij} of the degrees of the nodes connected by the link l_{ij} [157], denoted as $d_{ij} = d_i d_j$.

The degree recovery strategy involves a sequence of link additions based on the descending order of the product of the degrees of the end points of each removed link. Consequently, the first link to be added possesses the largest product of the degrees of its end points in the removed link set.

BETWEENNESS AND FLOW BETWEENNESS RECOVERY STRATEGIES (BET AND FLOWBET)

Betweenness of a link is widely used to measure the importance of a link l_{ij} in the network. It is defined as the ratio of the shortest paths through the link l_{ij} to the number of all shortest paths in the network [158]. If we denote the number of shortest paths from node s to node t as $\mathscr{P}_{s\to t}$ and the number of the shortest paths from node s to node t through the link l_{ij} as $\mathscr{P}_{s\to t}(l_{ij})$, the betweenness b_{ij} of link l_{ij} can be calculated by

$$b_{ij} = \sum_{s,t\in\mathcal{N}} \frac{\mathscr{P}_{s\to t}(l_{ij})}{\mathscr{P}_{s\to t}}. \tag{5.7}$$

The shortest paths are the most efficient ways for a flow in a network to travel from one node to another node if the traveling cost of every link is the same and there are no other

limitations in the network like link capacity. However, in a power grid, the distribution of the power flow is determined by the Kirchhoff and Ohm laws, not by the shortest paths. Therefore, Newman [159] proposed flow betweenness to measure the link importance in a power grid. The flow betweenness $\bar{b}_{ij}$ of link l_{ij} is defined as the sum of the power flow through link l_{ij} for any node pair s and t if one unit power is injected to node s and one unit power is extracted from node t. The flow betweenness $\bar{b}_{ij}$ of link l_{ij} can be calculated by

$$\bar{b}_{ij} = \sum_{s,t \in \mathcal{N}} |f_{s \to t}(l_{ij})| \tag{5.8}$$

where $|f_{s \to t}(l_{ij})|$ is the magnitude of flow through the link l_{ij} according to DC flow equations and Ohm's law when we inject one unit active power to node s and extract one unit active power from node t.

The sequence of added links in the betweenness or flow betweenness recovery strategies assumes a descending order of betweenness or flow betweenness of links.

EIGENVECTOR AND WEIGHTED EIGENVECTOR RECOVERY STRATEGIES (EIGEN AND WEIEIGEN)

The eigenvector centrality measures the importance of a node by not only taking into account the number of its connected nodes but also considering the importance of its connected nodes [160][157]. The eigenvector centrality x_i of node i is i-th element of the eigenvector with the largest eigenvalue of the adjacency matrix A. Compared to the eigenvector centrality, the weighted eigenvector centrality $\bar{x}_i$ of node i is the i-th element of the eigenvector with the largest eigenvalue of the weighted adjacency matrix $\widetilde{A}$. The products e_{ij} and $\bar{e}_{ij}$ of the eigenvector centrality and the weighted eigenvector centrality of link l_{ij}'s end points are calculated by $e_{ij} = x_i x_j$, $\bar{e}_{ij} = \bar{x}_i \bar{x}_j$.

The sequences of links are added in descending order of products e_{ij} or $\bar{e}_{ij}$ of links in the eigenvector or weighted eigenvector recovery strategies.

CLOSENESS, ELECTRICAL CLOSENESS AND ELECTRICAL WEIGHTED CLOSENESS RECOVERY STRATEGIES (CLOSE, ELECLOSE AND ELEWEICLOSE)

The closeness centrality of a node relates to the node's distance towards all other nodes [161]. If the distance of the shortest path between node i and node j is denoted by $\mathscr{H}_{ij}$, then the closeness $\bar{c}_i$ of node i is defined as

$$\bar{c}_i = \frac{1}{\sum_{j=1}^{N} \mathscr{H}_{ij}}. \tag{5.9}$$

We calculate the product $\bar{c}_{ij}$ of the closeness of link l_{ij}'s end points, $\bar{c}_{ij} = \bar{c}_i \bar{c}_j$, which is used in the closeness recovery strategy to measure the importance of a link.

As the flow in a power grid follows the Kirchhoff law, not only the shortest path, using the so-called effective resistance is a more appropriate way to quantify the distance between a pair of nodes in a power grid [162]. The effective resistance between node i and node j is denoted as Ω_{ij}, which can be calculated by using the pseudo-inverse matrix $Q^{\dagger}$ of the Laplacian matrix Q: $\Omega_{ij} = (Q^{\dagger})_{ii} + (Q^{\dagger})_{jj} - 2(Q^{\dagger})_{ij}$ [163]. In analogy with closeness, the electrical closeness $\widetilde{c}_i$ of node i is given by

$$\widetilde{c}_i = \frac{1}{\sum_{j=1}^{N} \Omega_{ij}}. \tag{5.10}$$

We use the product $\widetilde{c}_{ij}$ of electrical closeness values of the end points of link l_{ij} in the electrical closeness recovery strategy, where $\widetilde{c}_{ij} = \widetilde{c}_i \widetilde{c}_j$.

Compared with the electrical closeness of a link, the difference in calculating the electrical weighted closeness of a link is that we use the weighted Laplacian matrix $\widetilde{Q}$. Analogously, we first obtain the effective resistance $\widetilde{\Omega}$ using $\widetilde{\Omega}_{ij} = (\widetilde{Q}^{\dagger})_{ii} + (\widetilde{Q}^{\dagger})_{jj} - 2(\widetilde{Q}^{\dagger})_{ij}$ where $\widetilde{Q}^{\dagger}$ is the pseudo-inverse matrix of the Laplacian matrix $\widetilde{Q}$. Then the electrical weighted closeness w_i of node i is calculated by

$$w_i = \frac{1}{\sum_{j=1}^{N} \widetilde{\Omega}_{ij}}. \tag{5.11}$$

The product w_{ij} of electrical weighted closeness values of the end points of link l_{ij}, $w_{ij} = w_i w_j$, is the measurement of a link in the electrical weighted closeness strategy.

In the closeness, electrical closeness, and electrical weighted closeness recovery strategies, we prioritize the recovery of links based on the descending order of the products $\bar{c}_{ij}$, $\widetilde{c}_{ij}$, or w_{ij}.

5

ZETA VECTOR AND WEIGHTED ZETA RECOVERY STRATEGIES (ZETA AND WEIZETA)

The zeta vector ζ, inspired by the electrical flow in a resistant network, serves as a representation of nodal spread capacity. The minimization of the zeta vector leads to the identification of the best spreader [147]. Specifically, the zeta vector consists of the diagonal elements of the pseudo-inverse matrix $Q^{\dagger}$, derived from the Laplacian matrix Q. Thus, $\zeta = ((Q^{\dagger})_{11}, (Q^{\dagger})_{22}, ..., (Q^{\dagger})_{ii})$, where $(Q^{\dagger})_{ii}$ represents the zeta vector value ζ_i associated with node i.

When employing the weighted adjacency matrix $\widetilde{A}$, the weighted zeta vector $\widetilde{\zeta}$ is given by the diagonal elements of the pseudo-inverse matrix of the weighted Laplacian matrix $\widetilde{Q}$, denoted as $\widetilde{\zeta} = ((\widetilde{Q}^{\dagger})_{11}, (\widetilde{Q}^{\dagger})_{22}, ..., (\widetilde{Q}^{\dagger})_{ii})$. Here, $(\widetilde{Q}^{\dagger})_{ii}$ represents the weighted zeta vector value $\widetilde{\zeta}_i$ associated with node i.

To assess the significance of a link based on the zeta vector values of its end points, we introduce the zeta vector metric $\zeta_{ij} = \zeta_i \zeta_j$ and the weighted zeta vector metric $\widetilde{\zeta}_{ij} = \widetilde{\zeta}_i \widetilde{\zeta}_j$ for a given link l_{ij}. The metrics are calculated by multiplying the zeta vector values or the weighted zeta vector values of the link's end points.

To restore the removed links in the zeta vector and weighted zeta vector recovery strategies, we adopt a ranking scheme based on the descending order of the links' zeta vector metric values or weighted zeta vector metric values.

5.5. RESULTS

To investigate the effectiveness of recovery strategies in power grids, we conducted simulations on three different power grids, the IEEE 30 bus system, IEEE 39 bus system and IEEE 118 bus system. In each realization, we randomly removed links until the R-value of the system fell below a specified threshold. We then used various strategies to restore the system until all the removed links were added. Finally, we computed the recoverability energy ratio of each recovery strategy given the same random removal process. To explore the impact of thresholds and tolerance levels, we selected two thresholds (0.8 and 0.5) and four tolerance levels (1, 2, 2.5 and 3) and performed 1000 realizations for a power

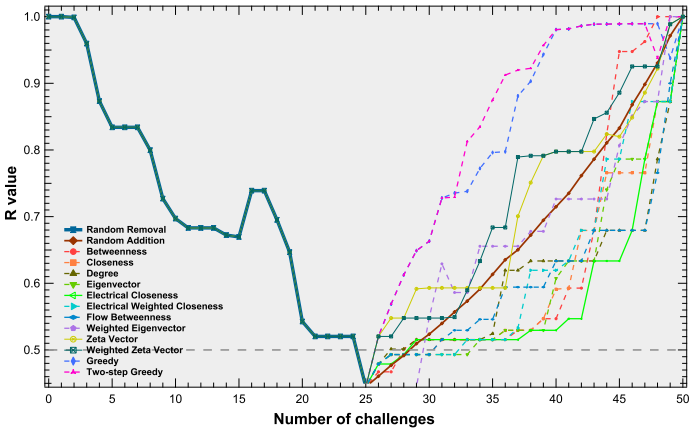


Figure 5.6: An example of R values during one random attack realization and different recovery processes with a threshold equal to 0.5 and a tolerance level equal to 2 for the IEEE 39 bus system. The results of the random recovery strategy are the average values of 100 realizations.

5

Table 5.2: The abbreviations of the strategies used in figures and tables in the paper

Abbreviation	Full name
TwoGreedy	Two step greedy recovery strategy
Greedy	Greedy recovery strategy
Bet	Betweenness recovery strategy
FlowBet	Flow betweenness recovery strategy
EleWeiClose	Electrical weighted closeness recovery strategy
WeiEigen	Weighted eigenvector recovery strategy
Close	Closeness recovery strategy
EleClose	Electrical closeness recovery strategy
Rand	Random recovery strategy
Degree	Degree recovery strategy
Zeta	Zeta vector recovery strategy
WeiZeta	Weighted zeta vector recovery strategy
Degree	Degree recovery strategy
Eigen	Eigenvector recovery strategy

grid with each setting. In Fig. 5.6, we demonstrate how the R-value varies for one attack realization with different recovery strategies for the IEEE 39 bus system.

We present the recoverability energy ratios of various recovery strategies concerning two thresholds and the tolerance level equal to 1 for the considered power grids, as shown in Tables 5.3, 5.4 and 5.5. The tables illustrate that the greedy two-step method exhibits the highest mean value of recoverability energy ratio and the lowest standard deviation value for all three power grids in both threshold cases. Additionally, the greedy method consistently maintains the second-best performance in terms of recoverability energy

Table 5.3: The mean value and standard deviation values of different recovery strategies for the IEEE 30 bus system with the tolerance level α equal to 1 while we set the threshold values as 0.8 and 0.5. The results of the random recovery strategy have a gray background color

Rank	Threshold = 0.8			Threshold = 0.5		
	Strategy	**Mean**	**std**	**Strategy**	**Mean**	**std**
1	TwoGreedy	1.0292	0.0241	TwoGreedy	1.1710	0.0656
2	Greedy	1.0285	0.0240	Greedy	1.1595	0.0654
3	Eigen	0.9877	0.0400	Zeta	0.9854	0.0679
4	EleWeiClose	0.9852	0.0420	Eigen	0.9837	0.0714
5	Close	0.9823	0.0427	WeiZeta	0.9796	0.0676
6	Rand	0.9808	0.0265	Rand	0.9733	0.0526
7	EleClose	0.9807	0.0429	EleWeiClose	0.9687	0.0762
8	Degree	0.9804	0.0423	Close	0.9629	0.0722
9	Zeta	0.9803	0.0358	EleClose	0.9576	0.0713
10	WeiEigen	0.9772	0.0440	Degree	0.9571	0.0727
11	WeiZeta	0.9772	0.0354	WeiEigen	0.9509	0.0761
12	Bet	0.9765	0.0404	Bet	0.9442	0.0667
13	FlowBet	0.9709	0.0447	FlowBet	0.9126	0.0693

Table 5.4: The mean value and standard deviation values of different recovery strategies' recoverability energy ratio for IEEE 39 bus system with the tolerance level α equal to 1 while we set the threshold values as 0.8 and 0.5. The results of the random recovery strategy have a gray background color

Rank	Threshold = 0.8			Threshold = 0.5		
	Strategy	**Mean**	**std**	**Strategy**	**Mean**	**std**
1	TwoGreedy	1.0299	0.0221	TwoGreedy	1.1665	0.0605
2	Greedy	1.0292	0.0219	Greedy	1.1543	0.0585
3	Zeta	1.0065	0.0240	Zeta	1.0709	0.0571
4	WeiZeta	1.0000	0.0260	WeiZeta	1.0582	0.0594
5	Rand	0.9858	0.0217	Rand	0.9741	0.0474
6	WeiEigen	0.9800	0.0347	WeiEigen	0.9430	0.0618
7	Bet	0.9794	0.0333	Bet	0.9384	0.0673
8	EleWeiClose	0.9730	0.0347	EleWeiClose	0.8945	0.0625
9	Eigen	0.9703	0.0351	Eigen	0.8871	0.0653
10	Degree	0.9701	0.0363	Degree	0.8848	0.0706
11	FlowBet	0.9697	0.0331	FlowBet	0.8827	0.0618
12	Close	0.9675	0.0358	Close	0.8809	0.0629
13	EleClose	0.9646	0.0348	EleClose	0.8639	0.0621

ratio mean value for all three power grids, albeit its performance is slightly inferior to that of the greedy two-step method. Notably, the difference in performance between the greedy two-step method and the greedy method is more pronounced when the threshold is equal to 0.5, compared to the case when the threshold is 0.8. The observation suggests that the greedy two-step method outperforms the greedy method when the removal link set is relatively large. Another noteworthy finding concerns the results of the random recovery method. The mean recoverability energy ratios of all power grids with different

Table 5.5: The mean value and standard deviation values of different recovery strategies' recoverability energy ratio for IEEE 118 bus system with the tolerance level α equal to 1 while we set the threshold values as 0.8 and 0.5. The results of the random recovery strategy have a gray background color

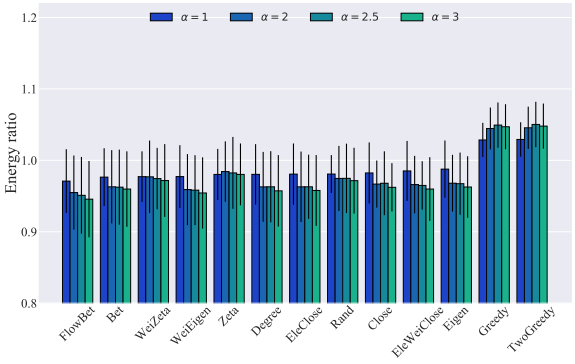
Rank	Threshold = 0.8			Threshold = 0.5		
	Strategy	**Mean**	**std**	**Strategy**	**Mean**	**std**
1	TwoGreedy	1.0458	0.0270	TwoGreedy	1.2611	0.0715
2	Greedy	1.0441	0.0266	Greedy	1.2294	0.0717
3	Bet	0.9959	0.0409	Bet	1.0855	0.0683
4	FlowBet	0.9824	0.0685	WeiEigen	1.0225	0.0782
5	EleWeiClose	0.9759	0.0639	EleWeiClose	1.0206	0.0732
6	WeiEigen	0.9726	0.0648	Close	1.0051	0.0734
7	Zeta	0.9705	0.0388	FlowBet	0.9989	0.0988
8	Close	0.9703	0.0677	Zeta	0.9961	0.0735
9	Rand	0.9666	0.0400	Eigen	0.9909	0.0701
10	EleClose	0.9649	0.0799	Rand	0.9819	0.0545
11	WeiZeta	0.9611	0.0435	Degree	0.9646	0.0846
12	Degree	0.9599	0.0774	WeiZeta	0.9624	0.0715
13	Eigen	0.9523	0.0731	EleClose	0.9589	0.0855

thresholds of the random recovery method are less than one, indicating that the cost of recovery outweighs the cost of attacks.

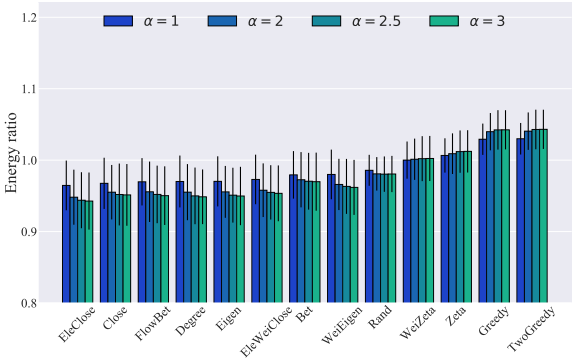
For the recovery strategies utilizing link metrics or the product of node centralities of the link's end points, we have observed noteworthy variations in performance across different power grids. Specifically, the recovery method based on link betweenness exhibits diverse rankings of mean recoverability energy ratio when applied to the three power grids under consideration. In the IEEE 30 bus system, the mean recoverability energy ratio rank of the betweenness recovery method falls within the bottom two. Conversely, within the IEEE 39 bus system, the betweenness recovery method achieves an intermediate rank among all methods based on the mean recoverability energy ratio. Remarkably, the recovery method based on betweenness centrality yields the highest mean recoverability energy ratio among all recovery strategies based on ink metrics or the product of node centrality values of the link's end points in the IEEE 118 bus system.

In addition to the betweenness recovery strategy, the zeta vector recovery strategy exhibits a similar performance pattern. The strategy demonstrates effectiveness in the IEEE 30 bus system with a threshold value of 0.5, as well as in the IEEE 39 bus system. However, the efficiency diminishes when applied to the IEEE 118 bus system. The results indicate that the efficacy of recovery methods based on link metrics or the product of node centralities of the link's end points is contingent upon the underlying network topology and dynamics on the networks.

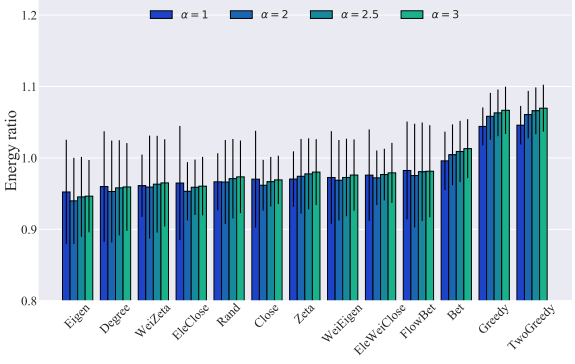
The observation can be attributed to the distinct allocation of generators in the IEEE 39 bus system, as illustrated in Fig. 5.2. Specifically, the majority of generators in the IEEE 39 bus system have only one neighbor and the proportion of links attached to generators is 22.74%, which is significantly higher than the corresponding proportions in the other two systems. In contrast, the IEEE 30 bus system has only one generator with one neighbor and the proportion of links connected to generators is 14.63%, while the IEEE 118 bus



(a) IEEE 30 Threshold = 0.8



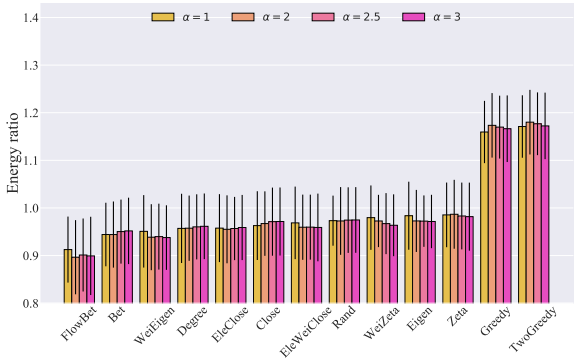
(b) IEEE 39 Threshold = 0.8



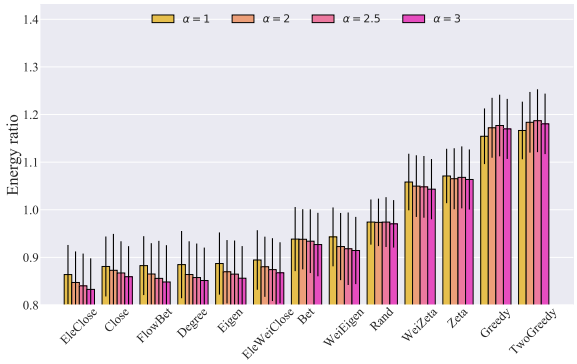
(c) IEEE 118 Threshold = 0.8

Figure 5.7: Bar graph of different recovery strategies' recoverability energy ratios with thresholds equal to 0.8 for the three considered bus systems with different tolerance levels. The number of realizations is 1000.

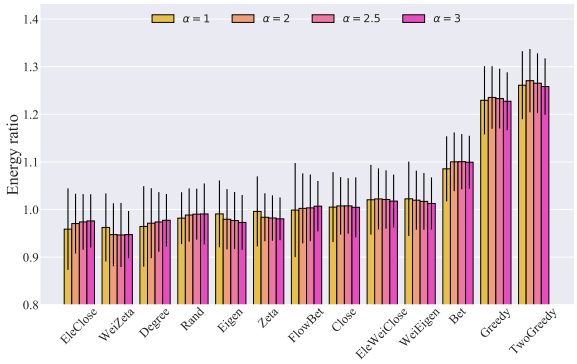
system has three generators with one neighbor and the proportion of links connected



(a) IEEE 30 Threshold = 0.5



(b) IEEE 39 Threshold = 0.5



(c) IEEE 118 Threshold = 0.5

Figure 5.8: Bar graph of different recovery strategies' recoverability energy ratios with thresholds equal to 0.5 for the three considered bus systems with different tolerance levels. The number of realizations is 1000.

to generators is 10.62%. Therefore, the location of generators in the IEEE 39 system

makes the links connected to generators more susceptible to attacks and less likely to be recovered than the other two systems. Based on calculating the average values of links' metrics used for the different recovery strategies of two kinds of links: links connecting generators with loads and links connecting loads with loads in Appendix C.2 Table C.2, we find that in the IEEE 39 system, the link metrics of links connecting generators and loads are larger than the link metrics of links connecting loads with loads in zeta recovery strategy and weighted zeta recovery strategy, indicating recovering the links connecting generators and loads matter in the recovery process.

Compared to the IEEE 30 bus system and the IEEE 39 bus system, the recovery methods based on link metrics and the product of node centrality values of the link's end points in the IEEE 118 system demonstrate substantially better performance. Specifically, most recovery strategies based on link metrics and the product of node centrality values of the link's end points exhibit higher average recoverability energy ratios than the random recovery strategy for both thresholds.

The link capacity is a crucial factor in causing blackouts in power grids. Although our method does not employ the cascading failure model, we investigated the impact of link capacity on recovery strategies. To this end, we conducted simulations with different tolerance levels ($\alpha = 1, \alpha = 2, \alpha = 2.5, \alpha = 3$) and analyzed the results, which are presented in Figs. 5.7 and 5.8 for the IEEE 30 bus system, IEEE 39 bus system and IEEE 118 bus system. The simulation results show that the mean values of the recoverability energy ratio with respect to different tolerance levels do not change monotonically for the same recovery method, which could inspire the study on the optimal tolerance level. The ranking of recovery strategies based on network metrics may vary slightly, but the two-step greedy recovery strategy consistently performs the best, with the greedy recovery strategy following closely behind. Moreover, the results indicate that the recovery strategy based on betweenness outperforms the recovery strategy based on flow betweenness, while the electrical weighted closeness recovery strategy is the best among the electrical closeness recovery strategy, closeness recovery strategy and electrical weighted closeness recovery strategy.

5.6. CHAPTER SUMMARY

In this chapter, we observe significant variations in the performances of different strategies for recovering power grids. Firstly, recovery strategies based on link metrics and the product of node centrality values of the link's endpoints do not achieve the highest performance. Secondly, the performances of these recovery strategies vary notably across different power systems, emphasizing the significance of network topology and generator placements in power grids. Thirdly, the two-step greedy recovery strategy consistently outperforms all other network metric-based recovery methods across all power systems, thresholds, and tolerance levels. The greedy recovery strategy, which is slightly less effective than the two-step greedy recovery strategy, also shows promising results. The findings confirm that implementing an effective recovery strategy can significantly improve the recoverability of power grids and the two-step greedy strategy is particularly efficient while relying solely on one network metric for recovery is less effective.

PART III: SPREADING ON NETWORKS

6

TIME DEPENDENCE OF SIS EPIDEMICS ON NETWORKS WITH NODAL SELF-INFECTIONS

The average fraction of infected nodes, in short the prevalence, of the Markovian ϵ-SIS (susceptible-infected-susceptible) process with small self-infection rate $\epsilon>0$ exhibits, as a function of time, a typical "two-plateau" behavior, which was first discovered in the complete graph K_N. Although the complete graph is often dismissed as an unacceptably simplistic approximation, its analytic tractability allows to unravel deeper details, that are surprisingly also observed in other graphs as demonstrated by simulations. The time-dependent mean-field approximation for K_N performs only reasonably well for relatively large self-infection rates, but completely fails to mimic the typical Markovian ϵ-SIS process with small self-infection rates. While self-infections, particularly when their rate is small, are usually ignored, the interplay of nodal self-infection and spread over links may explain why absorbing processes are hardly observed in reality, even over long time intervals.

This chapter is based on the published paper [164].

6.1. INTRODUCTION

The ϵ-susceptible-infected-susceptible (ϵ−SIS) model on a network describes that each node has two states: susceptible and infected. A susceptible node can be infected by an infected neighbor with an infection rate or with its independent self-infection rate ϵ. An infected node will recover with a specific recovery rate. Self-infections occur naturally in the metapopulation, where each metapopulation group consists of items (individuals, computers, etc.) that are indistinguishable and exchangeable. Each metapopulation group can be represented by a node, and iterations can take place between groups. Additionally, items within the same group can infect each other, which can be modeled as a self-infection process. Using the ϵ−SIS model, emotions spreading on social networks can be modeled [50]. Furthermore, the self-infection process can also be used to model the "imminent" infection in reality [165]. Viruses usually do not fully die out in the environment, which keeps challenging people's immunity in case people lose entirely immunity. In addition, as the ϵ−SIS model does not have an absorbing state, the model with a small self-infection rate can be used to measure the performance of different mean-field approximation methods in the susceptible-infected-susceptible (SIS) model [49].

In this chapter, we focus on a simple and stochastic epidemic model with self-infections. In a graph G with N nodes, the viral state of a node i at time t is specified by a Bernoulli random variable $X_i(t) \in \{0,1\}$: if $X_i(t) = 0$, the node is in a healthy (susceptible) state; if $X_i(t) = 1$, the node is in an infected state. Thus, at any time t, a node can only be in of two possible states: infected, with probability $v_i(t) = \Pr[X_i(t) = 1] = E[X_i(t)]$ or healthy, with probability $1 - v_i(t)$. The continuous-time description of the homogeneous self-infectious Markovian susceptible-infected-susceptible (ϵ−SIS) process on a contact graph is as follows. The curing process per infected node is a Poisson process with rate δ and the infection process per link is a Poisson process with rate β. The self-infection process per node is also a Poisson process with rate ϵ, which means, on average, every $\frac{1}{\epsilon}$ time units, a self-infection event occurs. All Poisson processes are independent. The $\epsilon - SIS$ model reduces to the "classical" SIS model when the self-infection rate is zero. In the ϵ−SIS heterogeneous setting, the curing rate δ_i and the self-infection rate ϵ_i are coupled to a node i and β_{ij} specifies the infection rate on the link from node i to node j.

Inspired by the observation that the ϵ−SIS model with special self-infection rates is very close to the mean-field steady state in the classical SIS model [49], Van Mieghem [51] investigated the ϵ−SIS model with arbitrary small but nonzero self-infection rate on complete graphs and found that the corresponding model exhibits a phase transition in the average steady-state fraction of infected nodes, with the steepness of the transition increasing as the size N of the graph grows. In this chapter, we explore the time dependence of the average fraction $y_N(\tau, \epsilon^*; t)$ of infected nodes, briefly called the prevalence, in a graph with N nodes where $\tau = \frac{\beta}{\delta}$ is the effective infection rate and $\epsilon^* = \frac{\epsilon}{\delta}$ is the effective self-infection rate. We emphasize that the Markovian setting is important and that a mean-field analysis is unable to describe the "typical ϵ−SIS temporal behavior".

6.2. TIME-DEPENDENT ϵ-SIS PREVALENCE IN THE COMPLETE GRAPH

6.2.1. MARKOVIAN SETTING

On the complete graph K_N, we can obtain an exact expression for the steady-state fraction of infected nodes in the $\epsilon - SIS$ epidemic process. The number of infected nodes $M(t)$ at time t is a continuous-time Markov process on {0, 1, ..., N} states with the following rates:

$$\begin{aligned} M &\mapsto M+1 \quad \text{at rate} \quad (\beta M+\epsilon)(N-M), \\ M &\mapsto M-1 \quad \text{at rate} \quad \delta M. \end{aligned}$$

Every infected node is recovered with rate δ and every susceptible node (in total $N-M$) is infected with rate βM from its M infected neighbors and its self-infection rate ϵ. The Markov process is indeed the birth and death process with birth rate $\lambda_j = (\beta j+\epsilon)(N-j)$ and death rate $\mu_j = \delta j$. We denote the probability that the number of infected nodes $M(t)$ at time t equals k by

$$s_k(t) = \Pr[M(t) = k]. \tag{6.1}$$

Applying the differential equations of a general birth and death process to the ϵ−SIS process yields the set of equations

$$s_0'(t) = \delta s_1(t) - N\epsilon s_0(t), \tag{6.2}$$

$$\begin{aligned} s_k'(t) = \{\beta K^2 - (n\beta+\delta-\epsilon)k - N\epsilon\} s_k(t) + \{-\beta(k-1)^2 + (N\beta-\epsilon)(k-1) + N\epsilon\} \\ \times s_{k-1}(t) + \delta(k+1) s_{k+1}(t), \end{aligned} \tag{6.3}$$

where all involved rates β, δ, and ϵ can depend on time t.

We denote the probability of steady state i as $\pi_j = \lim_{t\to\infty} \Pr[M(t) = j]$, which can be computed in the birth and death process as follows [166],

$$\pi_0 = \frac{1}{1 + \sum_{k=1}^{N} \prod_{m=0}^{k-1} \frac{(\beta m+\epsilon)(N-m)}{(m+1)\delta}}, \tag{6.4}$$

$$\pi_j = \pi_0 \prod_{m=0}^{j-1} \frac{(\beta m+\epsilon)(N-m)}{(m+1)\delta}, 1 \le j \le N. \tag{6.5}$$

Using the normalized self-infection rate $\epsilon^* = \frac{\epsilon}{\delta}$, the above steady-state probabilities are

$$\pi_0 = \frac{1}{\sum_{k=0}^{N} \binom{N}{k} \tau^k \frac{\Gamma(\frac{\epsilon^*}{\tau}+k)}{\Gamma(\frac{\epsilon^*}{\tau})}}, \tag{6.6}$$

$$\pi_j = \pi_0 \binom{N}{j} \epsilon^* \tau^{j-1} \frac{\Gamma(\frac{\epsilon^*}{\tau}+j)}{\Gamma(\frac{\epsilon^*}{\tau}+1)}. \tag{6.7}$$

The average steady-state fraction of infected nodes (the steady-state prevalence) is

$$y_{\infty;N}(\tau,\epsilon^*) = \frac{1}{N}\sum_{j=0}^{N} j\pi_{j;N} = \frac{\pi_{0;N}}{N}\sum_{j=1}^{N} j\binom{N}{j}\tau^j \frac{\Gamma(\frac{\epsilon^*}{\tau}+j)}{\Gamma(\frac{\epsilon^*}{\tau})}. \tag{6.8}$$

For the classical SIS model with effective infection rate τ on the complete graph K_N, using the first order $N-$intertwined mean-field (NIMFA) prevalence model [167], the average steady-state fraction $y^{(1)}_{\infty;N}$ of infected nodes follows,

$$y^{(1)}_{\infty;N} = 1 - \frac{1}{(N-1)\tau}, \qquad \tau > \frac{1}{N-1} = \tau_c^{(1)}, \tag{6.9}$$

where $\tau_c^{(1)}$ is the first-order mean-field epidemic threshold. The mean-field SIS epidemic threshold in any graph is $\tau_c^{(1)} = \frac{1}{\lambda_1}$, where λ_1 is the largest eigenvalue of the adjacency matrix of the contact graph and lower bounds the Markovian SIS epidemic threshold τ_c [168].

In [51], it was shown that there always exists a phase transition around τ_c^ϵ in the $\epsilon-SIS$ process on the complete graph K_N, above which the effective infection rate $\tau > \tau_c^\epsilon$ causes the prevalence $y_{\infty;N}(\tau,\epsilon^*)$ to approach the prevalence $y^{(1)}_{\infty;N}$ of the classical SIS model, no matter how small, but still non-zero, the self-infection rate ϵ^* is. The phase transition τ_c^ϵ can be bounded by

$$\frac{1}{e}\left(\frac{10^{-s}}{\epsilon*(N-1)!}\right)^{\frac{1}{N-1}} < \tau_c^\epsilon < \left(\frac{10^{-s}}{\epsilon*(N-1)!}\right)^{\frac{1}{N-1}}, \tag{6.10}$$

where s specifies an agreed level for the onset of the phase transition at which $y_{\infty;N}(\tau,\epsilon^*) = 10^{-s}$ is first reached, when τ is gradually increased from $\tau = 0$ at $y_{\infty;N}(0,\epsilon^*) = \frac{\epsilon^*}{1+\epsilon^*}$ on.

The observation of a phase transition, increasingly more abrupt with size N of the graph was only possible by a purely analytical study of the complete graph K_N in [51]. Unfortunately, the phase transition is difficult to simulate as the absorbing time increases exponentially with N in the SIS model. Therefore, we show simulations for small graphs of $N = 30$ nodes. For most graphs with $N = 30$ nodes, the absorbing time at which the virus dies out from the network is within 10^5 time units, which is within an interval equal to 10^5 times the average curing time. The simulation is performed in our simulator introduced in [49]. The unit of time equals the average curing time. We have set the curing rate in simulations to $\delta = 1$ (so that $\epsilon^* = \epsilon$ here).

We present the prevalence $y_N(\tau,\epsilon^*;t)$ of the $\epsilon-$SIS process on the complete graph K_{30} with effective infection rate $\tau = 2\tau_c^{(1)} = \frac{2}{29}$ and different values of the effective self-infection rate ϵ in Fig. 6.1. The simulation results are over 100 realizations. For a small self-infection rate $\epsilon = 10^{-5}$, the prevalence $y_{30}(2\tau_c^{(1),10^{-5}};t)$ starts with one infected node, increases rapidly, almost exponentially fast, towards about 20% of infected nodes between 1 and 10 time units, after which the prevalence decreases slowly towards its steady state around 4%. The $\epsilon-$SIS with small self-infection rates generalizes the classical SIS where eventually the prevalence tends to zero for any effective infection rate τ in any graph with the finite number of N nodes. When the self-infection rate is relatively high ($\epsilon = 10^{-1}$, or $\epsilon = 10^{-2}$), the prevalence curves resemble the mean-field typical SIS behavior; when the

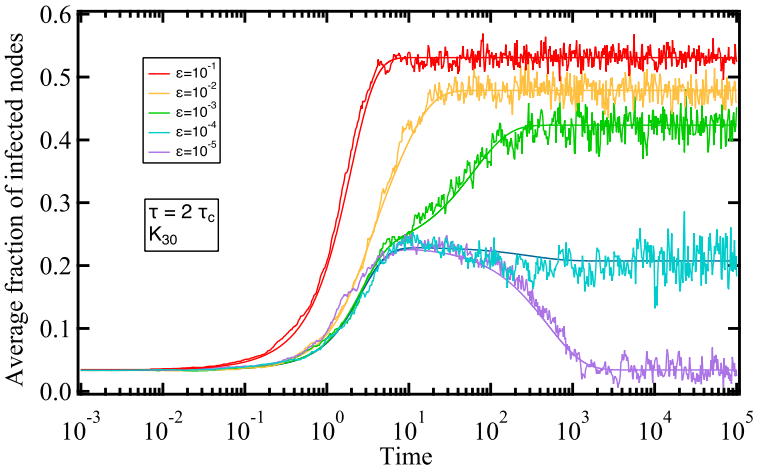


Figure 6.1: The prevalence $y_N(\tau,\epsilon^*;t)$ of the ϵ−SIS process on the complete graph K_N on $N=30$ nodes with effective infection rate values of the self-infection rate ϵ. The epidemic started with one infected node. Both simulations and numerical solutions of the differential equations Eqs. (6.2) and (6.3). are presented for each self-infection rate ϵ.

self-infection rate is between $\epsilon = 10^{-3}$ and $\epsilon = 10^{-4}$ in Fig. 6.1, the typical ϵ−SIS behavior in time is shown, which we will explain below.

Fig. 6.2 presents the prevalence $y_{\infty;N}(\tau,\epsilon^*)$ of the ϵ−SIS process and the NIMFA steady state prevalence $y^{(1)}_{\infty;N}(\tau)$ in a Markovian SIS process. When the number of nodes N increases, the transition of the prevalence from small to NIMFA values is increasingly steep, resulting in a step function if $N \to \infty$ [51]. Here, the number of nodes is 30, so the transition is not very steep. Although we present the results of the complete graph in Figs. 6.1 and 6.2, for other kinds of graphs, the prevalence $y_{\infty;N}(\tau,\epsilon^*)$ behavior is the same as we show in Fig. 6.1. We will demonstrate the corresponding results of other graphs later.

Based on the numerical solution of the differential equations Eqs. (6.2) and (6.3) of the complete graph K_{30} with self-infection rate $\epsilon^* = 10^{-5}$, we present the prevalence versus time with various effective infection rate τ in Fig. 6.3. We can find the characteristic ϵ−SIS prevalence with two plateaus. For sufficiently high effective infection rates τ, the first plateau is rapidly reached and is due to link infections. The second, higher plateau is attained roughly around time $\frac{1}{\epsilon}$ (in units of the average curing time). The interarrival time of self-infection events generated at each node is exponentially distributed with mean $\frac{1}{\epsilon}$ based on the property of the Markov process. The minimum of N−interdependent exponential random variables, each with rate ϵ_i, is again an exponential random variable with a rate equal to their sum $\sum_{j=1}^{N}\epsilon_i$. Here, we have N nodes, each with a self-infection rate ϵ. Thus, the first appearance of self-infections is expected around time $\frac{1}{N\epsilon}$, which is here about 3×10^3 time units. Fig. 6.3 indeed illustrates that the onset towards the second plateau occurs around time 3×10^3. The precise time changes also depend on the effective infection rate τ over links because the onset starts for different τ, in particular for small τ,

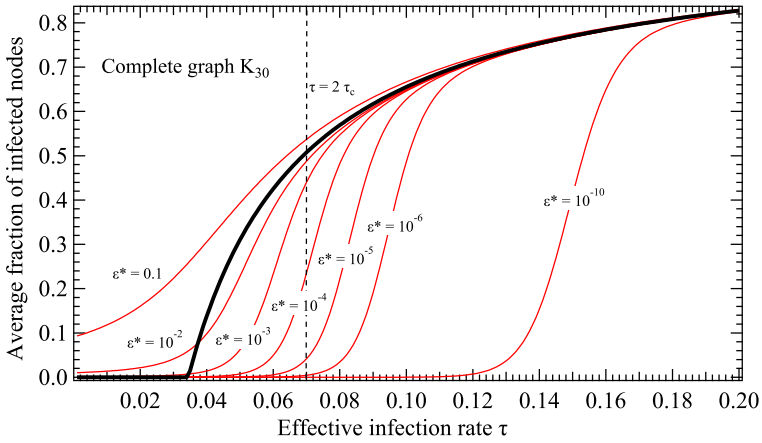


Figure 6.2: The steady-state prevalence $y_{\infty;N}(\tau,\epsilon^*)$ in K_{30} versus the effective infection rate τ, for various self-infection rates varying from $\epsilon^* = 10^{-1}$ down to $\epsilon^* = 10^{-10}$. The dotted line at $\tau = 2\tau_c^{(1)}$ corresponds to the setting in Fig. 6.1. The bold line is the NIMFA steady-state prevalence $y_{\infty;N}(\tau)$ in the Markovian SIS process.

6

at slightly different times.

For lower effective infection rates τ, the second plateau in the prevalence, which corresponds to its steady state value, is lower than the first plateau caused by spreading over links in the graph. The lower value is due to a curious equilibrium between absorption (due to a series of rate curing events as explained in [51]) and self-infection. When $\epsilon = 0$, the ϵ−SIS process is the classical Markovian SIS process where the prevalence tends to zero in any finite graph after a sufficiently long time. In other words, the epidemics are eradicated from the graph because of a sequence of curing events that happens with very low, though positive, probability. Hence, the observation interval must be long enough to observe such a rare event of successive curing events, which explains the long absorption time in the classical SIS process.

The onset of the second plateau, due to the self-infection process, is shown in Fig. 6.4 for several self-infection rates and a relatively high effective infection rate $\tau = 3\tau_c^{(1)} \approx 0.1$. A self-infection in each node is created on average every $\frac{1}{\epsilon}$ time units and as mentioned before the onset toward the second plateau is observed earlier at about $\frac{1}{N\epsilon}$. Compared with Fig. 6.1, Fig. 6.4 shows a narrow range of prevalence, which follows the result presented in Fig. 6.2: for higher effective infection rate τ and a fixed $\epsilon^* \in [10^{-5}, 10^{-1}]$ band, the corresponding prevalence band decreases. Also, the other two results in Fig. 6.4 satisfy the results with the effective infection rate $\tau = 0.1$ in Fig. 6.2 as well: the steady-state prevalence of self-infection rate $\epsilon = 0.1$ is slightly higher than the steady-state prevalence of self-infection rates from $\epsilon = 10^{-4}$ to $\epsilon = 10^{-2}$; the steady-state prevalence of self-infection rate $\epsilon = 10^{-5}$ is slightly lower than the steady-state prevalence of self-infection rates from $\epsilon = 10^{-4}$ to $\epsilon = 10^{-2}$.

Fig. 6.5 showing the steady-state prevalence with self-infection rate $\epsilon = 0.1$, is the companion of Fig. 6.3. In the ϵ−SIS process with high self-infection rate $\epsilon = 0.1$, both the

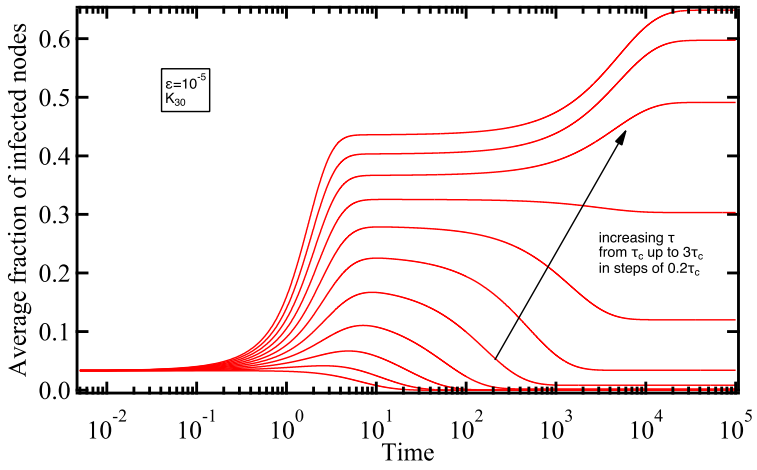


Figure 6.3: The prevalence $y_N(\tau,\epsilon^*;t)$ in the complete graph K_{30} versus time for self-infection rate $\epsilon = 10^{-5}$ and effective infection rates $\tau_c^{(1)} \le \tau \le 3\tau_c^{(1)}$ and each curve differs $0.2\tau_c^{(1)}$ from its neighboring curves.

link and the node infection events occur roughly at the same timescale and the combined effect results in "traditional mean-field" curves. In summary, only for small self-infection rate ϵ, the anomalous behavior in the ϵ−SIS, with respect to the mean-field theory is observed.

6.2.2. N-INTERTWINED MEAN-FIELD APPROXIMATION FOR ϵ-SIS

The governing differential equations of the N−intertwined mean-field approximation (NIMFA) for heterogeneous $\epsilon - SIS$ epidemics on a graph with adjacency matrix A are

$$\frac{dv_i(t)}{dt} = -\delta_i v_i(t) + [1 - v_i(t)]\left(\sum_{j=1}^{N} \alpha_{ij}\beta_{ij} v_j + \epsilon_i\right). \tag{6.11}$$

Eq. (6.11) shows that the change of the infected probability $v_i(t)$ of node i at time t depends on two parts: one is that when the node state is infected with probability $v_i(t)$, it will be healthy with its cure rate $\delta_i(t)$; another is that when the node state is healthy with probability $1 - v_i(t)$, it can be infected by its infected neighbors or get self-infected with self-infection rate ϵ_i. The probability of an infected neighbor j is $\alpha_{ij}v_j(t)$ and the infected rate of the specific link is β_{ij}. The matrix form of Eq. (6.11) is as follows.

$$\frac{dv(t)}{dt} = Bv + \eta - \text{diag}(v)(Bv + \eta + c), \tag{6.12}$$

where the vector $v = (v_1, v_2, \ldots, v_N)$, the self-infection vector $\eta = (\epsilon_1, \epsilon_2, \epsilon_3, \ldots, \epsilon_N)$, the curing vector $c = (\delta_1, \delta_2, \ldots, \delta_N)$ and the $N \times N$ matrix B has elements $\alpha_{ij}\beta_{ij}$. Using the $N \times 1$ all one vector, the average fraction of infected nodes (prevalence) is calculated by $y = \frac{u^T v}{N} = \frac{1}{N}\sum_{i=1}^{N} v_i(t)$, whose changes follow,

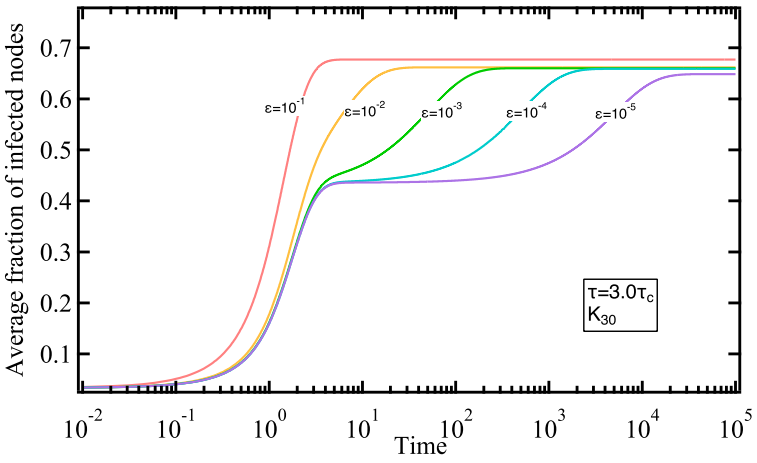


Figure 6.4: The prevalence on K_{30} over time for a high effective infection rate $\tau = 3\tau_c^{(1)} = \frac{3}{29}$ and several self-infection rates ϵ. The curves are numerical solutions of the differential equations.

6

$$N\frac{dy(t)}{dt} = u^T\eta - v^T(c+\eta) + (u-v)^T Bv. \tag{6.13}$$

For the complete graph K_N, we can analytically solve the above differential equation. When the initial condition $v_i(0)$ is the same for every node i, we have $y_N^{(1)}(\tau,\epsilon^*;t) = v(t) = v_i(t)$. By normalizing time $t^* = \delta t$, Eq. (6.11) can be denoted as

$$\frac{dv(t)}{dt^*} = \epsilon^* + [t(N-1) - (1+\epsilon^*)]v(t) - \tau(N-1)v^2(t). \tag{6.14}$$

We recognize that the above equation is a Ricatti differential equation with solution,

$$\begin{aligned} y_N^{(1)}(\tau,\epsilon^*;t) = {} & \frac{\tau(N-1)-(1+\epsilon^*)}{2\tau(N_1)} + \frac{\gamma}{2\tau(N-1)} \\ & \times \tanh\left[\frac{t}{2}\gamma + \operatorname{arctanh}\left(\frac{\tau(N-1)(2y_0-1)+(1+\epsilon^*)}{\gamma}\right)\right], \end{aligned} \tag{6.15}$$

where

$$\gamma = \sqrt{[\tau(N-1)-(1+\epsilon^*)]^2 + 4\tau(N-1)\epsilon^*}. \tag{6.16}$$

The mean-field approximation is the upper bound of the prevalence of the Markovian SIS process [168], which is manifested in Figs. 6.5 and 6.6. The mean-field approximation deviates considerably for small self-infection rates in Fig. 6.6 compared to the mean-field approximation for a large self-infection rate in Fig. 6.5.

6.3. BEYOND THE COMPLETE GRAPH

To investigate whether other graphs exhibit the same characteristic ϵ−SIS prevalence time behavior as complete graphs, we have done simulations on five other kinds of graphs:

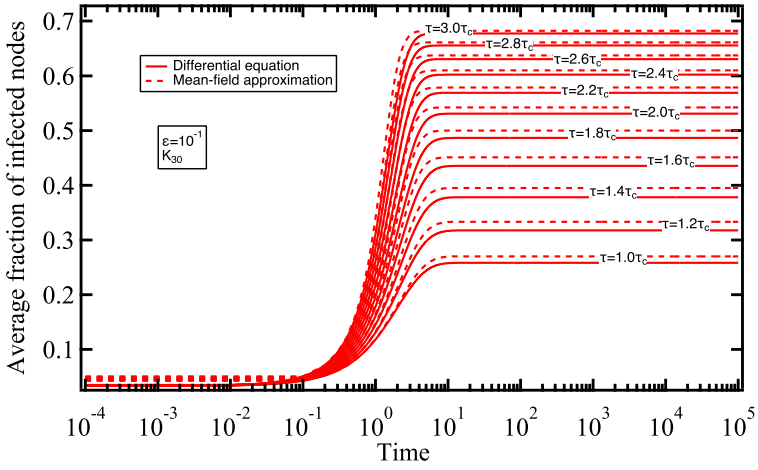


Figure 6.5: The prevalence $y_N(\tau,\epsilon^*;t)$ in the complete graph K_{30} versus time for the self-infection rate $\epsilon = 0.1$ with the same values of effective infection rates τ as in Fig. 6.1. The mean-field approximation (dotted lines) lies above ϵ−SIS Markovian prevalence which is computed by solving the differential equations Eq. (6.2) and (6.3).

6

square lattice, ring graph, Erdős-Rényi (ER) graph and Barabási-Albert power law (BA) graph. Fig. 6.7 confirms that different graphs, in addition to complete graphs, also exhibit the existence of two plateaus. A surprising fact concerns the BA graph with 500 nodes. For large N and $\epsilon = 0$, we expect that the absorbing state is reached only after $O(e^{Nc})$ time units, where the constant c depends on the effective infection rate and the topology of the graph. However, the green curve of the BA graph with $\tau = \frac{2.6}{\lambda_1}$, initially increases to show that the effective infection rate lies above the epidemic threshold, but decreases relatively fast.

Fig. 6.8 for an ER graph with 100 nodes, on the other hand, does not depict such a decrease; the green curve stays much longer in the metastable state, as expected because the absorbing time increases exponentially in the number of nodes N. Fig. 6.9 for a star graph with 30 nodes looks amazingly similar to Fig. 6.3. For the star graph with N nodes, the spectral radius is $\lambda_1 = \sqrt{N-1}$ and the mean-field epidemic threshold satisfies $\tau_c^{(1)} = \frac{1}{\lambda_1} = \frac{1}{\sqrt{N-1}}$. From [169], the actual epidemic threshold for sufficiently large N is $\tau_c = \frac{\alpha}{\sqrt{N-1}}$ with $\alpha = \sqrt{\frac{\log N}{2} + \frac{3}{2}\log\log N}$. Although $N = 30$ is not large, the estimated actual threshold is $\tau_c \approx \frac{1.872}{\sqrt{29}}$, which explains why the lowest curve of Fig. 6.9 corresponding to $\tau = 2\tau_c^{(1)}$ just exceeds the epidemic threshold τ_c. In contrast to the classical SIS process where all prevalence curves eventually end up at zero, Figs. 6.9 and 6.7 indicate that in the steady state (not the metastable state), all possible values of the prevalence between zero and the mean-field steady state are possible as claimed in [51].

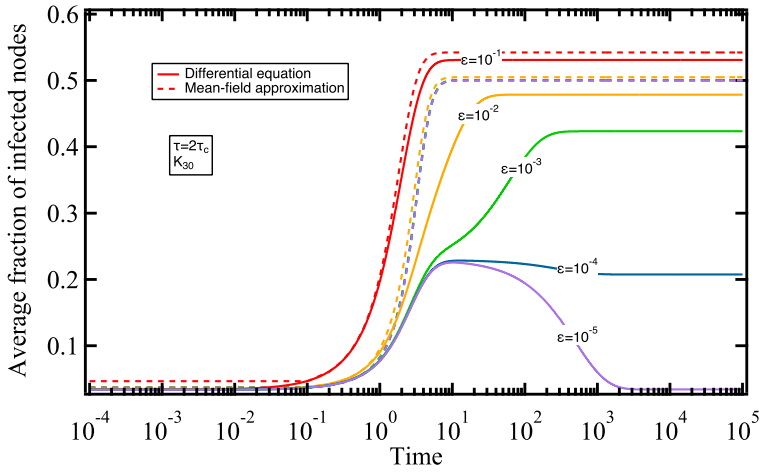


Figure 6.6: The prevalence $y_N(\tau,\epsilon^*;t)$ of the ϵ−SIS with the same parameters as in Fig. 6.1 to which the mean-field approximation is added. The same color curve presents the model with the same self-infection rate ϵ. The mean-field approximations of the prevalence with $\epsilon = 10^{-3}, 10^{-4}$ and 10^{-5} overlap, which is represented by the purple dashed line.

6

6.4. CHAPTER SUMMARY

The prevalence $y_N(\tau,\epsilon^*;t)$ of the Markovian ϵ−SIS process with small self-infection rate $\epsilon > 0$ has been studied as a function of time. The typical "two-plateau" behavior, first explained for the complete graph K_N based on analytic computations, is demonstrated to hold generally for other graphs other than complete graphs. In fact, we claim this behavior may hold for any finite graph. In reality, if a disease or the spread process has a small self-infection rate ϵ, the second plateau may play a significant role: after the epidemics arrive at the first plateau (in a relatively short time), the epidemics will not disappear like the classical SIS process nor stay constant as in the mean-field approximation but will tend to reach the second plateau which can be either higher of lower than the first plateau depending on its self-infection rate ϵ.

The time-dependent mean-field approximation for K_N only performs reasonably well for relatively large self-infection rates but completely fails to describe the typical Markovian ϵ−SIS process with small self-infection rates. While self-infections, in particular when their rates are small, are usually ignored, we have shown that the interplay of nodal infection and spread over links may explain why the absorbing processes are less often observed in real life.

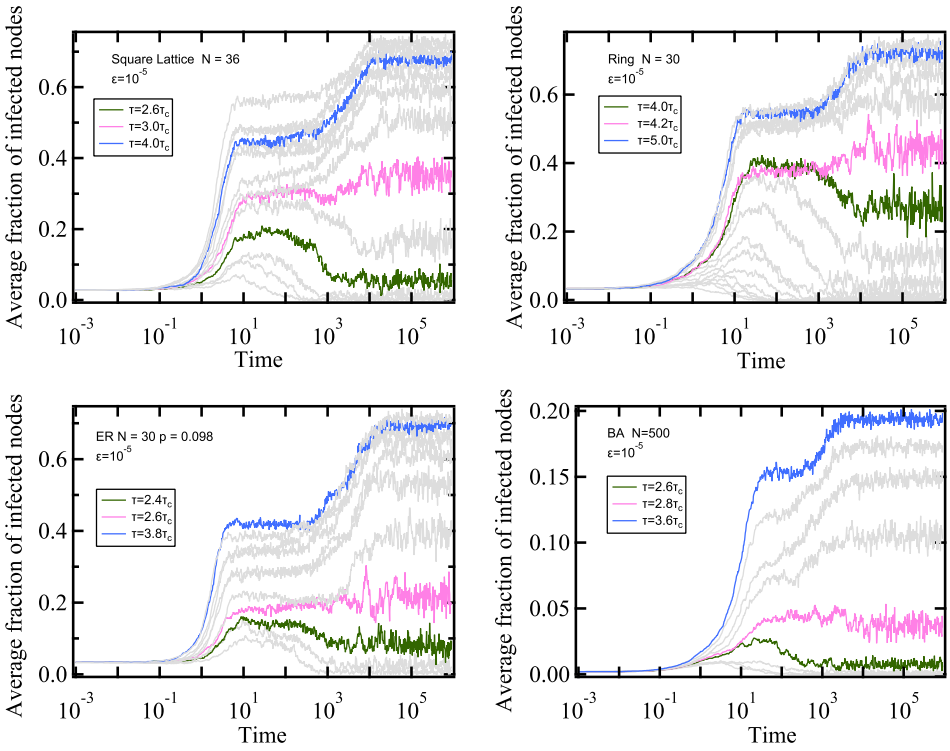


Figure 6.7: The prevalence $y_N(\tau, \epsilon^*; t)$ with self-infection rate $\epsilon = 10^{-5}$ versus time for four different kinds of graphs; the square lattice with 36 nodes, the ring graph with 30 nodes, the Erdős-Rényi random graph with link density $p = 0.098$ and the Barabási-Albert power law graph with $N = 500$. The average degree of the Barabási-Albert power law graph is two. Initially, there is one infected node and the figure is based on the average results of 100 realizations of each $\epsilon-$SIS process. The prevalence curves shown increase with a step of $\tau = 0.2\tau_c^{(1)}$. The three curves in color illustrate the three different behaviors: blue shows two plateaus and upward bending, red presents the almost constant plateau and green depicts downward bending.

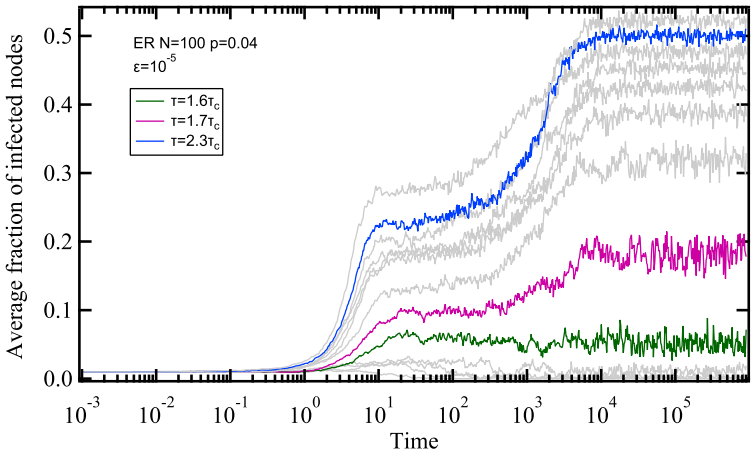


Figure 6.8: The "two-plateau" $\epsilon-$SIS prevalence curve versus time for an ER graph with 100 nodes and link density $p = 2p_c = 0.04$ and $\epsilon = 10^{-5}$. The downward bending of the green curve only occurs for considerably larger times.

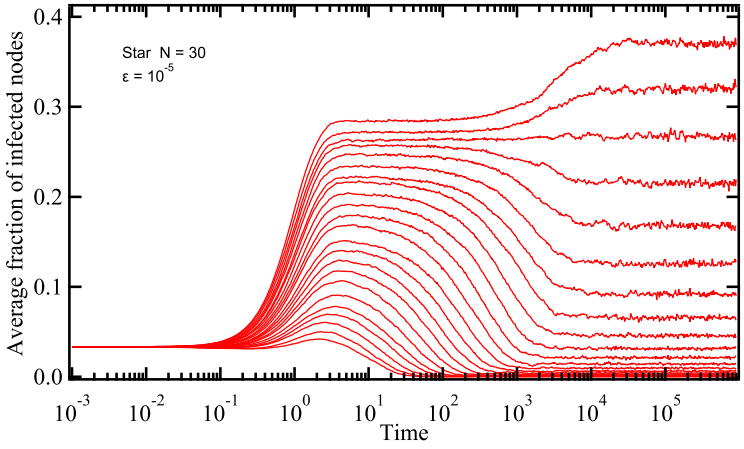


Figure 6.9: The $\epsilon-$SIS prevalence curve versus time in the star graph with 30 nodes. Initially, one node is infected and the prevalence is averaged over 10^4 realizations. The curves presented increase with a step of effective infection rate $\tau = 0.2\tau_c^{(1)}$ and the values of effective infection rates are from the lowest $\tau = 2\tau_c^{(1)}$ to the highest $\tau = 6\tau_c^{(1)}$.

7

CONCLUSION

7.1. MAIN CONTRIBUTIONS

In this dissertation, we have discussed research in network science on three topics: network robustness and reliability, network recoverability and spreading.

In Chapter 2, we deduce the analytical methods based on generating functions of in-degree and out-degree distributions to approximate the minimum fraction of the number of driver nodes in directed networks, subject to random node removals and targeted node removals based on in-degree, out-degree and total degree. We find that the analytical methods fit simulation results very well for random node removals; the proposed analytical methods for targeted node removal fit the simulation results reasonably well, in particular for small values of the fraction of removed nodes. Furthermore, our investigation indicates that random node removal may also serve as a reliable predictor of the results of various targeted node removals, particularly when the fraction of removed nodes is minimal (below 10%). Further, we find that for a real-world network, the minimum fraction of driver nodes calculated by the proposed analytical method utilizing generating functions does not always coincide with the results obtained using the maximum matching algorithm before node removals. As such, our proposed methods are inadequate for predicting the minimum fraction of driver nodes under node removal for these networks.

In Chapter 3, based on network reliability, we adopt controller reachability as the performance metric for addressing the controller placement problem. Controller reachability is the probability that each node can reach at least one controller. We employ the path decomposition algorithm to evaluate controller reachability, commonly used for computing all-terminal reliability. By applying this algorithm, we can calculate the controller reachability and quantify the impact of different placements on network performance. Our analysis reveals that different controller placements can significantly influence controller reachability, highlighting the importance of developing efficient placement strategies. Moreover, we investigate how many controllers are necessary to ensure satisfactory performance. In the case of small-size networks, introducing a second

controller yields substantial improvements in controller reachability for most networks. Upon analyzing the optimal placements for various graphs, we have determined that two topological graph metrics, namely degree and distance, heavily influence controller reachability. Building upon this insight, we propose a controller placement strategy that leverages degree and distance as key factors. Additionally, we develop three heuristic algorithms: the greedy algorithm, the classic genetic algorithm, and the heuristic genetic algorithm, to facilitate controller placement. To evaluate the effectiveness of these strategies, we conducted experiments using real-world graphs from the Topology Zoo dataset. All algorithms perform reasonably well. Considering both performance and algorithm complexity, we conclude that the strategy based on graph metrics and the greedy algorithm are suitable approaches for addressing the controller placement problem with respect to controller reachability.

In Chapter 4, we have introduced an analytical approach based on generating functions of in-degree and out-degree distributions to estimate the minimum fraction of driver nodes needed for achieving network controllability through random node additions. Besides, we have employed two recoverability indicators to assess the efficiency of seven recovery strategies after random node removals. These strategies include the random recovery strategy, degree-based recovery strategy, betweenness-based recovery strategy, updated degree recovery strategy, updated betweenness recovery strategy, greedy degree recovery strategy, and greedy betweenness recovery strategy. Upon analysis, we have observed a difference between our initial analytical predictions and simulation results in both synthetic and real-world networks. To address this inconsistency, we propose an adjustment to the original method to more closely align the results. Regarding the seven recovery strategies, we have determined that the greedy-betweenness recovery strategy demonstrates superior efficiency in directed ER networks and SSNs, while the greedy-degree recovery strategy proves most efficient in directed BA networks.

In Chapter 5, we investigate the efficiency of different link recovery strategies with respect to random link removals based on the optimal power flow model on power grids. We observe significant variations in the performance of different recovery strategies. First, recovery strategies based on link metrics and the product of node centrality values of the link's endpoints do not achieve the highest performance. Secondly, the performance of these recovery strategies varies significantly between different power systems, emphasizing the importance of network topology and generator placements in power grids. Third, the two-step greedy recovery strategy consistently outperforms all other network metrics-based recovery methods for all power systems, thresholds, and tolerance levels. Additionally, the greedy recovery strategy, although slightly less effective than the two-step greedy recovery strategy, demonstrates promising results. The findings confirm that implementing an effective recovery strategy can significantly improve the recoverability of power grids and that the two-step greedy strategy is particularly efficient while relying solely on one network metric for recovery is less effective.

In Chapter 6, the prevalence $y_N(\tau, \epsilon^*; t)$ of the Markovian ϵ−SIS process with small self-infection rate $\epsilon > 0$ has been studied as a function of time. The typical "two-plateau" behavior, first explained for the complete graph K_N based on analytic computations, is demonstrated to hold for other graphs other than complete graphs. Indeed, we argue that this behavior may hold for any finite graph. In reality, if a disease or the spread

process exhibits a small self-infection rate ϵ, the second plateau may play a significant role: after the epidemics reach the first plateau (in a relatively short time), the epidemics will not disappear like the classical SIS process nor stay constant as in the mean-field approximation but will tend to reach the second plateau which can be either higher of lower than the first plateau depending on its self-infection rate ϵ. The time-dependent mean-field approximation for K_N only performs reasonably well for relatively large self-infection rates but completely fails to describe the typical Markovian $\epsilon-$SIS process with small self-infection rates. While self-infections, in particular when their rates are small, are usually ignored, we have shown that the interplay of nodal infection and spread over links may explain why the absorbing processes are less observed in real life.

7.2. DIRECTIONS FOR FUTURE WORK

The approximations in Chapter 2 for node removals based on in- or out-degree, involve the assumption that the in-degree and out-degree distributions evolve independently. However, the assumption requires further investigation to ensure its validity. Another direction is to broaden the scope of our findings by including other types of node attacks, specifically localized node attacks, as documented in [77]. Furthermore, we intend to verify our conclusions on a more comprehensive collection of real-world networks and various types of networks, such as interdependent networks. We also plan to apply additional prediction techniques, such as machine learning methods, to assess network controllability under node removals concerning in-degree and out-degree.

In relation to Chapter 3, we can also consider the impact of nodes being non-operational with a certain probability or links between network devices and controllers being non-operational with a certain probability. Second, other performance metrics like the worst or the average node-controller reachability, where node-controller reachability is defined as the probability of a node being able to establish communication with at least one controller. Third, the strategy based on graph metrics might be improved by incorporating additional topological properties, such as the existence of multiple paths between network devices and controllers. Besides, the study of multi-objective optimization can be done while placing controllers, such as considering both latency and controller reachability.

With the investigation into approximating network controllability under random node additions, future research in line with Chapter 4 could delve into the development of analytical techniques for estimating network controllability under various recovery strategies. For instance, Wang *et al.* [52] have laid the groundwork for potential analytical methods to approximate network controllability under targeted node additions based on degree. Furthermore, considering the additional computation cost associated with the shifted model, there is potential for enhancing its effectiveness. One promising avenue is the exploration of algorithms with lower complexity to calculate the initial minimum number of driver nodes, thereby optimizing the performance of the shifted model. Moreover, considering that cycles play a critical role in network controllability [170], we can consider the method proposed by Fan *et al.* [171] to measure node centrality based on cycles, which could lead to the development of a cycle ratio recovery strategy, potentially offering improved recovery efficiency. In addition, the concept of the l-shell of a given node, defined as the set of nodes at a distance l from the focal node [78], presents an intriguing avenue for further research. Exploring localized attacks and subsequent recovery strategies based

on the shell distance l could offer insights into strategies that leverage localized information. These investigations hold the potential to deepen our understanding of the efficacy of diverse recovery methods and contribute to the evolution of more efficient network recovery techniques in the context of network controllability.

To enhance the results of Chapter 5, we can develop hierarchical recovery strategies by classifying links into two groups: those connected to generators and those not connected to generators. Then, we prioritize restoring links connected to generators. Second, we could investigate whether allowing cascading failures after link recovery or removal impacts the conclusions of this study. Thirdly, we could introduce additional constraints, such as generator costs, and incorporate cost minimization as an objective within the DC power flow model. Finally, given that distributed energy systems enhance power system resilience, it is promising to research how to establish and integrate such systems after a blackout using the discussed strategies.

To extend the research reported in Chapter 6, we can first consider the non-Markovian $\epsilon-$SIS process and investigate whether the time-dependent behavior is the same as the Markovian $\epsilon-$SIS process. Secondly, we can employ the $\epsilon-$SIS process to model failure spreading in real-world systems. Thirdly, we can extend the research by calculating the threshold for the spreading process on time-varying networks.

ACKNOWLEDGEMENTS

I am delighted to have reached the end of my Ph.D., although saying goodbye to many people I've come to know in the Netherlands is sad.

Firstly, my heartfelt thanks go to my promoters, Piet Van Mieghem and Robert E. Kooij. I am grateful to Piet for offering me the opportunity to become a Ph.D. candidate at NAS and for providing me with a free atmosphere to explore the research topics I was passionate about. He emphasized the importance of novel and attractive research, and his virtues, such as kindness and a rigorous attitude towards research, have left a lasting impression. Rob has been both inspiring and wise; working and conversing with him have been enjoyable experiences. Initially, he was my mentor, and later, he became my promoter as well. Rob has the magic to simplify and clarify things. He taught me presentation skills, guided me on supervising master's students, and offered valuable suggestions on job choices. He encouraged me to attend conferences and express my ideas. His boundless energy and youthful spirit are inspiring. I hope to meet him in China in the future.

I would also like to express my gratitude to NAS senior colleagues Edgar, Remco, Eric, Maksim, and Rogier. I enjoyed their presentations during NAS meetings and cherished the time spent in NAS drinks and lunches, where we shared views and experiences about life. Prior to my arrival in Delft and during staying in the Netherlands, the secretaries in the QCE department provided me with substantial support. Special thanks to Paul, Trisha, Laura, Francis, and Zenequy.

During my time at NAS, collaborating with researchers was a pleasure. Thanks to Hanna, Mattia, Rogier and David, as well as the master students with whom I had the opportunity to work, Yufei, Brian, Ran, and Jinyi.

I appreciate the friendly working atmosphere at NAS. Upon my arrival in Delft with my friend Li, the senior colleagues in the NAS group and the multimedia group extended help in both research and life perspectives. Thanks to Long, Zhidong, Peng, Qiang, Xiuxiu, Hale, and Bastian. Having nice and friendly colleagues during my Ph.D. was wonderful. I enjoyed our lunch time, making coffee, coffee breaks, discussions on various topics, conference trips, and casual hangouts. Special thanks to Ivan, Gaby, Liza, Xinhan, Massimo, Maria, Zhihao, Yingyue, Qingfeng, Robin, Brian, Matteo, David, Sergey, Yuxuan, and Scott. I am also grateful to Huijuan and colleagues from Huijuan's group, Alberto, Omar, Li, Shilun and Tianrui.

Life is not always easy, but I feel fortunate to have met dear friends who have brought me so much joy. I cherish the many joyful memories I have had with them. My heartfelt thanks to all, including Jing, Biyue, Li, Cheng, Jinke, Yawei, Sitong, Shenglan, Fengqiao, Qian, Tianqi, Ming, etc. Special thanks to Shilun for accompanying me during times of difficulty. I also extend my thanks to my friends in China, Lei, Xiaozhen, Rong, and Yanmeng.

Last but certainly not least, my deepest appreciation goes to my family. Thanks to my parents for their unwavering support and encouragement in allowing me to become the person I aspire to be. I am also grateful to my grandparents and relatives, who have encouraged me to face difficulties. Special thanks to CSC and the Dutch government for their financial support. Thanks to ChatGPT. To everyone who has offered help, support, and encouragement, I sincerely thank you. May all of you have a happy life and a successful future.

A

APPENDIX TO CHAPTER 2

A.1. THE SIMULATION RESULTS BASED ON DIFFERENT α VALUES

In Figs. A.1 and A.2, we demonstrate for $\alpha = 0$, $\alpha = 1$, $\alpha = 10$ and $\alpha = 100$, how the minimum fraction of driver nodes changes under targeted attacks based on degree, in-degree and out-degree for ER networks and SSNs. We find that the results of $\alpha = 10$ and $\alpha = 100$ overlap.

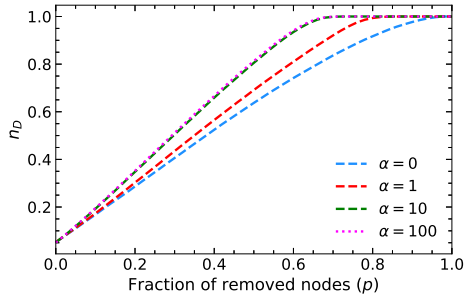
A.2. COMPARISON WITH NODE REMOVAL BASED ON DEGREE WITH $\alpha = 1$

We present the results of four types of node removal strategies: random removal, targeted node removal based on the total degree with $\alpha = 1$, targeted node removal based on in-degree with $\alpha = 1$, and targeted node removal based on out-degree with $\alpha = 1$ for three networks in Fig. A.3. We find that the three targeted node removal strategies are more disruptive than random removal. However, the effectiveness of the targeted node removal strategies varies depending on the network structure. For instance, in ER(100,0.04), all three targeted node removal strategies show similar performance. In SSN(10^4,2), the targeted node removal based on in-degree is the most disruptive, whereas, in HinerniaGlobal, the targeted node removal based on out-degree is the most disruptive.

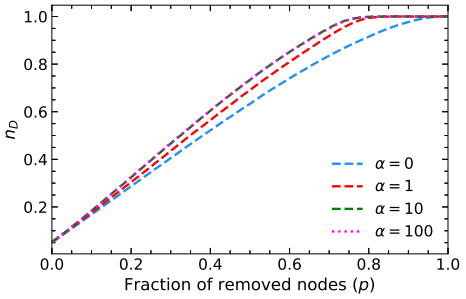
A.3. RESULTS FOR ANOTHER REAL-WORLD NETWORK

In Chapter 2, the real-world graphs utilized have an average degree ranging from 2 to 3. To further evaluate the efficacy of the proposed techniques, we selected a network from the Topology Zoo dataset, namely BtNorthAmerica, which possesses an average total degree of 4.22. The network under consideration comprises 36 nodes and 76 links. We analyzed the controllability of the network under node removals concerning node in-degree and out-degree with different α. The results are presented in Fig. A.4 and Tables A.1 and A.2. Our findings suggest that the predicted values of the proposed methods are valid. It is worth mentioning when $\alpha = 10$, attacks based on out-degree at the onset are not as

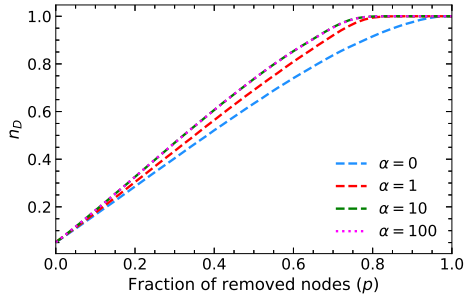
A



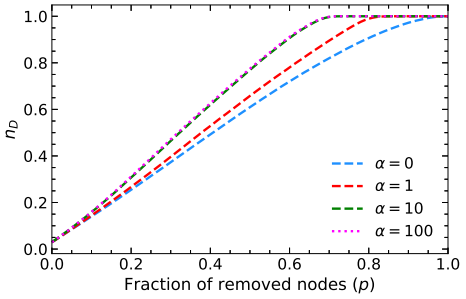
(a) Total−ER(50, 0.07)



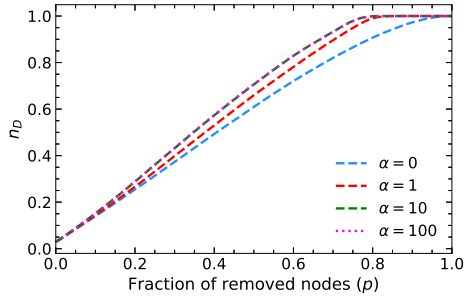
(b) In−ER(50, 0.07)



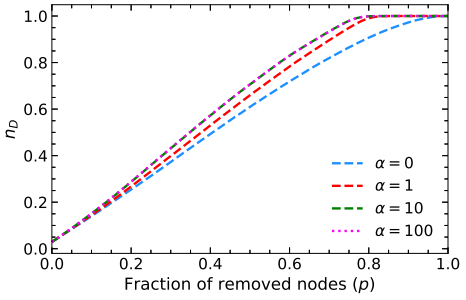
(c) Out−ER(50, 0.07)



(d) Total−ER(100, 0.04)

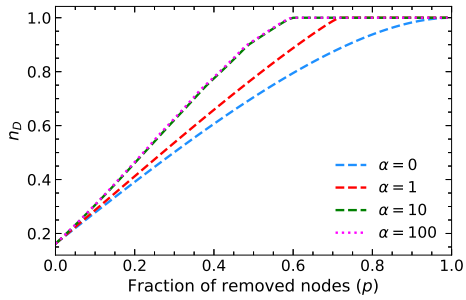


(e) In−ER(100, 0.04)

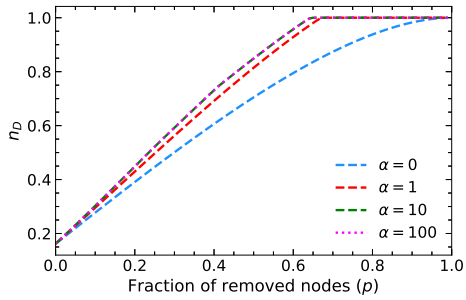


(f) Out−ER(100, 0.04)

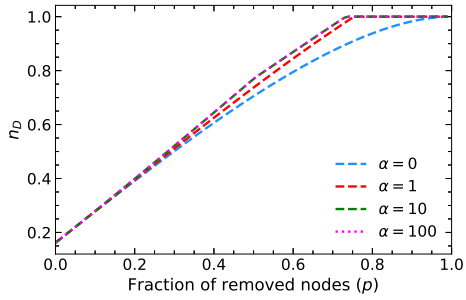
Figure A.1: The minimum fraction of driver nodes n_D during targeted node removal based on total degree, in-degree and out-degree with $\alpha = 0$, $\alpha = 1$, $\alpha = 10$ and $\alpha = 100$ for ER networks. The results are the average n_D calculated by the maximum matching algorithm over 10000 realizations of real-world networks and 1000 realizations of model networks. The blue, orange, and green dashed lines are the results of simulations with $\alpha = 0$, $\alpha = 1$ and $\alpha = 10$, respectively. The pink dotted lines are obtained by the simulation results with $\alpha = 100$.



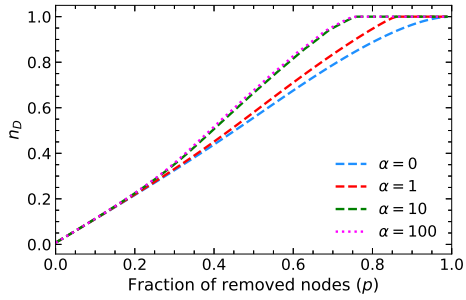
(a) Total−SSN(10^4,2)



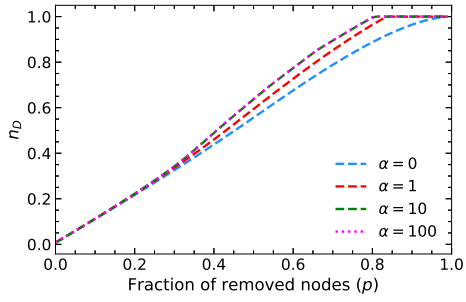
(b) In−SSN(10^4,2)



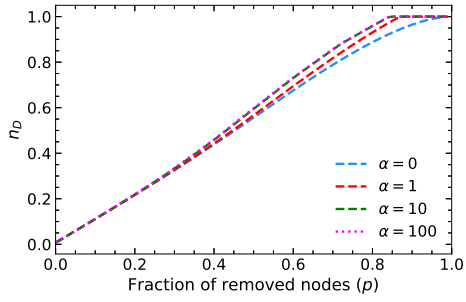
(c) Out−SSN(10^4,2)



(d) Total−SSN(10^4,5)



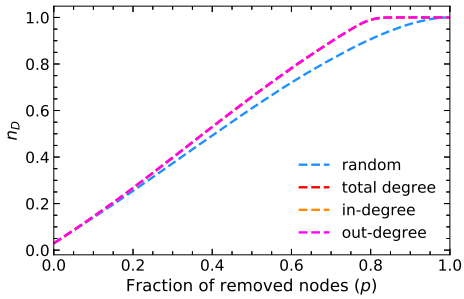
(e) In−SSN(10^4,5)



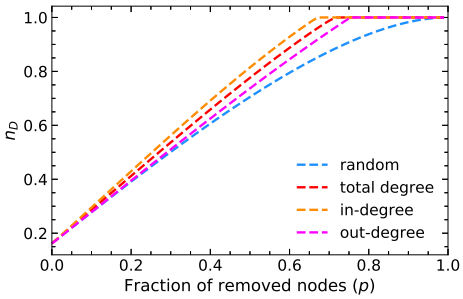
(f) Out−SSN(10^4,5)

Figure A.2: The minimum fraction of driver nodes n_D during targeted node removal based on total degree, in-degree and out-degree with $\alpha = 0$, $\alpha = 1$, $\alpha = 10$ and $\alpha = 100$ for SSN networks. The results are the average n_D calculated by the maximum matching algorithm over 10000 realizations of real-world networks and 1000 realizations of model networks. The blue, orange, and green dashed lines are the results of simulations with $\alpha = 0$, $\alpha = 1$ and $\alpha = 10$, respectively. The pink dotted lines are obtained by the simulation results with $\alpha = 100$.

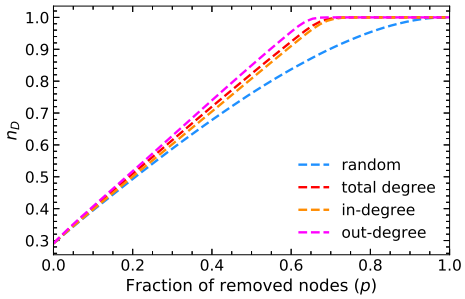
A



(a) ER(100, 0.04)



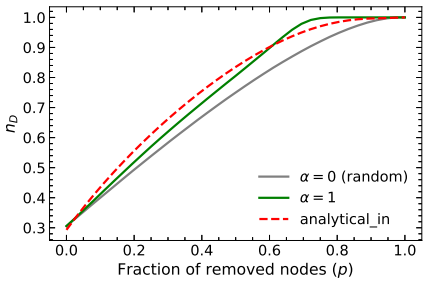
(b) SSN(10^4, 2)



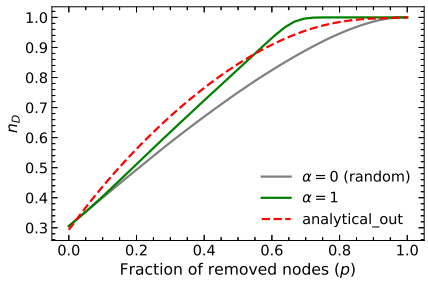
(c) HinerniaGlobal

Figure A.3: The minimum fraction of driver nodes n_D during random removal and targeted node removal based on total degree, in-degree and out-degree with $\alpha = 1$ for three networks. The results are the average n_D calculated by the maximum matching algorithm over 10000 realizations of HinerniaGlobal and 1000 realizations of ER(100, 0.04) and SSN(10^4, 2). The blue, red, orange and pink dashed lines are the results of simulations with random removal, targeted removal based on the total degree with $\alpha = 1$, targeted removal based on in-degree with $\alpha = 1$ and targeted removal based on out-degree with $\alpha = 1$.

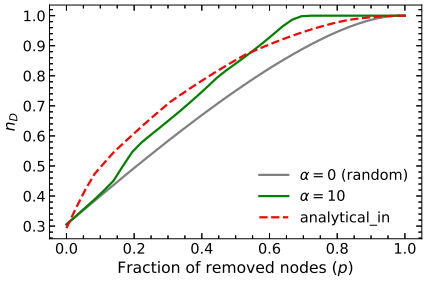
deleterious as random removals. Removing the node with the highest out-degree in the initial steps results in a lower average number of driver nodes than removing other nodes, on average.



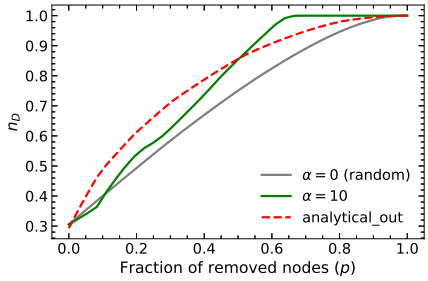
(a) In-degree with $\alpha = 1$



(b) Out-degree with $\alpha = 1$



(c) In-degree with $\alpha = 10$



(d) Out-degree with $\alpha = 10$

Figure A.4: The minimum fraction of driver nodes n_D during targeted node removal based on in-degree and out-degree with $\alpha = 1$ and $\alpha = 10$, respectively, for the network BtNorthAmerica.

Table A.1: The RMSE between the analytical results for random removals and the simulation results under random removals, targeted removals with $\alpha = 1$ and $\alpha = 10$, respectively, while removing up to 10% of the nodes.

network	random	$\alpha = 1$			$\alpha = 10$		
		in-degree	out-degree	degree	in-degree	out-degree	degreee
BtNorthAmerica	0.0097	0.0140	0.0104	0.0121	0.0117	0.0096	0.0101

Table A.2: The RMSE between the analytical results of the proposed analytical methods and the simulation results under different kinds of removals while removing up to 10% of the nodes.

.

network	random	$\alpha = 1$			$\alpha = 10$		
		in-degree	out-degree	degree	in-degree	out-degree	degreee
BtNorthAmerica	0.0097	0.0126	0.0175	0.0091	0.0538	0.0612	0.0527

B

Appendix to Chapter 4

B.1. Fitting Power-Law Exponents of the In-Degree and Out-Degree Distributions for Directed BA Networks

To illustrate the power-law degree distributions in directed BA networks, we present the mean and standard deviation values of the fitted power-law exponents for both in-degree and out-degree distributions across 10,000 networks. This analysis is conducted for each pair of parameters N and m outlined in Table B.1.

B.2. Analytical method based upon rescaling

Besides using the difference between the simulation result $n_D[0]'$ and the analytical approximation $n_D[0]$, we also constructed a model using a rescaling factor γ, defined as $\gamma = \frac{n_D[0]'}{n_D[0]}$. Consequently, if the analytical result of the minimum fraction of driver nodes at a particular challenge k is denoted as $n_D[k]$, the rescaled result $n_D[k]'$ can be computed as follows:

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

We present the results of the rescaled model for the Topology Zoo dataset in a histogram Fig. B.1, where the results obtained before and after rescaling are depicted in orange and blue, respectively. We observe a noticeable improvement in the approximations for the seven large real-world graphs by comparing the results obtained before and after rescaling (Table B.2). The results for the rescaled model exhibit smaller AME and RMSE values, and higher $P_{RMSE \leq 0.05}$ values, thus indicating the effectiveness of the rescaling method. However, compared to the results using the shifted model, the prediction improvements are slightly less.

B

Table B.1: The mean value γ_{in} and γ_{out}, and standard deviation value in_{std} and out_{std}, of fitted power-law exponents of the in-degree distribution and the out-degree distribution for 10000 directed BA networks for each pair of parameters N and m

Name	γ_{in}	γ_{out}	in_{std}	out_{std}
BA(50,2)	3.53	3.50	1.78	1.78
BA(50,4)	4.72	4.61	3.06	2.96
BA(100,2)	3.29	3.34	2.04	2.13
BA(100,4)	3.19	3.19	1.29	1.29
BA(500,2)	2.81	2.81	0.33	0.28
BA(500,4)	2.87	2.87	0.15	0.14
BA(1000,2)	2.83	2.83	0.12	0.15
BA(1000,4)	2.89	2.89	0.07	0.07
BA(5000,2)	2.86	2.86	0.06	0.06
BA(5000,4)	2.92	2.92	0.04	0.04
BA(10000,2)	2.87	2.87	0.05	0.05
BA(10000,4)	2.92	2.92	0.04	0.03

Table B.2: The results of the rescaled model for 7 large scale real-world networks

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

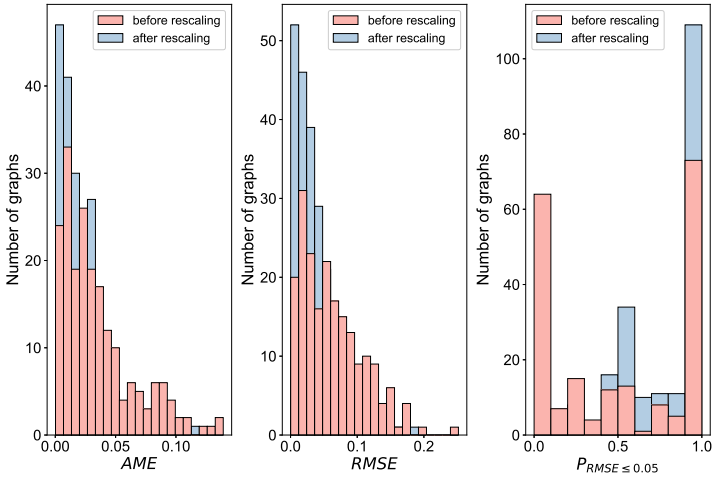


Figure B.1: The method based upon rescaling has better approximation performance for the Topology Zoo data set than the original analytical model.

B.3. THE PERFORMANCE OF DIFFERENT RECOVERY STRATEGIES IN SMALL SIZE SYNTHETIC NETWORKS

Table B.3: The recovery energy E of different recovery strategies for different kinds of synthetic networks. "Rand" is an abbreviation of random recovery strategy; "Deg" is an abbreviation of degree based recovery strategy; "Bet" presents betweenness based recovery strategy; "Deg-up" is an abbreviation of updated degree based recovery strategy; "Bet-up" presents updated betweenness recovery strategy; "Greedy-deg" is an abbreviation of greedy-degree recovery strategy; "Greedy-bet" presents greedy-betweenness recovery strategy. The bold values represent the lowest recovery energy for each type of graph

Name	Rand	Deg	Bet	Deg-up	Bet-up	Greedy-deg	Greedy-bet
ER(50, 0.07)	1.29925	1.24832	1.23136	1.24708	1.22850	1.19586	**1.19462**
ER(100, 0.04)	1.79139	1.73317	1.70636	1.73172	1.70256	1.65979	**1.65714**
SSN(50, 2)	2.23936	2.15538	2.13779	2.15146	2.13314	2.06563	**2.06426**
SSN(50, 4)	0.98394	0.97422	0.96245	0.97383	0.96052	0.94980	**0.94963**
SSN(100, 2)	3.92723	3.76935	3.74061	3.7614	3.73418	3.59817	**3.59557**
SSN(100, 4)	1.61484	1.58919	1.56030	1.58803	1.55644	1.51731	**1.51674**
BA(50, 2)	3.39699	3.25094	3.26215	3.24886	3.26049	**3.17485**	3.17540
BA(50, 4)	1.44235	1.34377	1.33573	1.34272	1.33353	1.28932	**1.28862**
BA(100, 2)	6.20176	5.92666	5.95579	5.92276	5.95477	**5.77948**	5.78079
BA(100, 4)	2.63404	2.42488	2.42039	2.42208	2.41671	2.30261	**2.30216**

Table B.4: The recoverability metric R of different recovery strategies for different kinds of synthetic networks. "Rand" is an abbreviation of random recovery strategy; "Deg" is an abbreviation of degree based recovery strategy; "Bet" presents betweenness based recovery strategy; "Deg-up" is an abbreviation of updated degree based recovery strategy; "Bet-up" presents updated betweenness recovery strategy; "Greedy-deg" is an abbreviation of greedy-degree recovery strategy; "Greedy-bet" presents greedy-betweenness recovery strategy. The bold values represent the lowest recoverability metric for each type of graph

Name	Rand	Deg	Bet	Deg-up	Bet-up	Greedy-deg	Greedy-bet
ER(50, 0.07)	0.14436	0.13870	0.13682	0.13856	0.13650	0.13287	**0.13274**
ER(100, 0.04)	0.11196	0.10832	0.10665	0.10823	0.10641	0.10374	**0.10357**
SSN(50, 2)	0.24882	0.23949	0.23753	0.23905	0.23702	0.22951	**0.22936**
SSN(50, 4)	0.10933	0.10825	0.10694	0.10820	0.10672	0.10553	**0.10551**
SSN(100, 2)	0.24545	0.23558	0.23379	0.23509	0.23339	0.22489	**0.22472**
SSN(100, 4)	0.10093	0.09932	0.09752	0.09925	0.09728	0.09483	**0.09480**
BA(50, 2)	0.37744	0.36122	0.36246	0.36098	0.36228	**0.35276**	0.35282
BA(50, 4)	0.16026	0.14931	0.14841	0.14919	0.14817	0.14326	**0.14318**
BA(100, 2)	0.38761	0.37042	0.37224	0.37017	0.37217	**0.36122**	0.36130
BA(100, 4)	0.16463	0.15156	0.15127	0.15138	0.15104	0.14391	**0.14389**

C

APPENDIX TO CHAPTER 5

C.1. ABBREVIATIONS

K_a	the number of challenges in the attack process
K_r	the number of challenges in the recovery process
$R[k]$	the value of R at challenge k
S_a	the attack strength
S_r	the recovery strength
η	the recoverability energy ratio
$\mathcal{G}$	a graph
N	the number of nodes in a graph
$\mathcal{N}$	the set of N nodes
L	the number of links in a graph
$\mathcal{L}$	the set of L links
A	the adjacency matrix
a_{ij}	the element of the adjacency matrix
l_{ij}	the link connecting node i and node j
y_{ij}	the impedance of transmission line l_{ij}
$\widetilde{A}$	the weighted adjacency matrix
$\widetilde{a}_{ij}$	the element of the weighted adjacency matrix
d_{av}	the average degree
$L_i[k]$	the amount of satisfied demand at bus i at challenge k
$G_i[k]$	the amount of supply at bus i at challenge k
P_i	the injected power at node i
$\widetilde{Q}$	the weighted Laplacian matrix
$\widetilde{\Delta}$	the weighted degree matrix
θ_i	the phase angle of bus i
$\boldsymbol{\Theta}$	the $N \times 1$ vector with elements θ
$\widetilde{B}$	the weighted incidence matrix
$\widetilde{b}_{ij}$	the element of the weighted incidence matrix

C

$\boldsymbol{G}[k]$	the supply vector including the supply $G_i[k]$ of each bus i at challenge k
$\boldsymbol{L}[k]$	the demand vector including the demand $L_i[k]$ of each bus i at challenge k
$\boldsymbol{F}[k]$	the active power flow vector at challenge k
$f_{ij}[k]$	the power flow of line l_{ij} at challenge k
$\boldsymbol{C}$	the capacity vector including the capacity of all transmission lines
c_{ij}	the capacity of each transmission line l_{ij}
α	the tolerance level
d_i	the degree of node i
d_{ij}	the degree of link l_{ij}
b_{ij}	the betweenness of link l_{ij}
$\mathscr{P}_{s\to t}$	the number of shortest paths from node s to node t
$\mathscr{P}_{s\to t}(l_{ij})$	the number of shortest paths from node s to node t through link l_{ij}
$\bar{b}_{ij}$	the flow betweenness
$\lvert f_{s\to t}(l_{ij})\rvert$	the magnitude of flow through the link l_{ij} when we inject one unit active power to node s and extract one unit active power from node t
x_i	the eigenvector centrality of node i
e_{ij}	the product of eigenvector centrality values of end points of link l_{ij}
$\bar{x}_i$	the weighted eigenvector centrality of node i
$\bar{e}_{ij}$	the product of weighted eigenvector centrality values of end points of link l_{ij}
$\mathscr{H}_{ij}$	the distance of the shortest path between node i and node j
$\bar{c}_i$	the closeness of node i
$\bar{c}_{ij}$	the product of closeness values of end points of link l_{ij}
Q	the Laplacian matrix of adjacency matrix A
$Q^\dagger$	the pseudo-inverse Laplacian matrix of the Laplacian matrix Q
Ω_{ij}	the effective resistance between node i and node j calculated by using the pseudo-inverse matrix $Q^\dagger$
$\widetilde{c}_i$	the electrical closeness of node i
$\widetilde{c}_{ij}$	the product of electrical closeness values of end points of link l_{ij}
$\widetilde{Q}^\dagger$	the pseudo-inverse matrix of the weighted Laplacian matrix $\widetilde{Q}$
$\widetilde{\Omega}_{ij}$	the effective resistance between node i and node j calculated by using the pseudo-inverse matrix $\widetilde{Q}^\dagger$
w_i	the electrical weighted closeness of node i
w_{ij}	the product of electrical weighted closeness values of end points of link l_{ij}
ζ	the zeta vector
$\widetilde{\zeta}$	the weighted vector
ζ_{ij}	the product of zeta vector values of end points of link l_{ij}
$\widetilde{\zeta}_{ij}$	the product of weighted zeta vector values of end points of link l_{ij}

C.2. THE STATISTICAL PROPERTIES OF LINK METRICS FOR TWO KINDS OF LINKS BASED ON WHETHER OR NOT LINKS ARE CONNECTED WITH GENERATORS

Table C.1: The mean value and standard deviation of various link metrics are calculated for two categories of links, classified based on whether or not links are connected with generators for the IEEE 30 bus system

Metrics	Connected to generators		Not connected to generators	
	Mean	Std	Mean	Std
FlowBet	94.770551	44.594022	114.742752	47.264862
Degree	10.400000	6.231258	12.269231	9.101902
Zeta	0.528584	0.296456	0.492346	0.434582
Bet	0.081226	0.044353	0.080283	0.063794
Eigen	0.032307	0.042318	0.050533	0.055398
WeiZeta	0.025984	0.031659	0.024918	0.050187
EleWeiClose	0.012889	0.004615	0.014670	0.005090
WeiEigen	0.045759	0.122430	0.009824	0.037966
EleClose	0.000488	0.000132	0.000535	0.000167
Close	0.000123	0.000029	0.000136	0.000038

Table C.2: The mean value and standard deviation of various link metrics are calculated for two categories of links, classified based on whether or not links are connected with generators for the IEEE 39 bus system

Metrics	Connected to generators		Not connected to generators	
	Mean	Std	Mean	Std
FlowBet	96.370201	45.321198	240.779292	107.181289
Degree	3.272727	0.646670	8.800000	2.654630
Zeta	3.114607	1.747125	1.138656	0.685581
EleWeiClose	0.229349	0.068645	0.437775	0.126349
Bet	0.050301	0.005854	0.119877	0.072109
Eigen	0.008464	0.004923	0.037058	0.016150
WeiEigen	0.003469	0.010297	0.030148	0.074798
WeiZeta	0.001033	0.000709	0.000310	0.000363
EleClose	0.000070	0.000018	0.000115	0.000027
Close	0.000026	0.000005	0.000039	0.000010

Table C.3: The mean value and standard deviation of various link metrics are calculated for two categories of links, classified based on whether or not links are connected with generators for the IEEE 118 bus system

Metrics	Connected to generators		Not connected to generators	
	Mean	Std	Mean	Std
FlowBet	832.375534	764.6106	778.816749	542.1641
Degree	19.189189	10.90265	10.761905	5.989612
Zeta	1.091884	1.30306	1.338246	0.9381184
Bet	0.046504	0.0607467	0.027309	0.0374265
WeiZeta	0.009382	0.0138922	0.011388	0.0088103
WeiEigen	0.002517	0.0154307	0.005404	0.0468745
Eigen	0.020586	0.022895	0.005041	0.008672
EleWeiClose	0.002065	0.0006941	0.001772	0.0005885
EleClose	0.000017	0.0000051	0.000014	0.0000038
Close	0.000002	0.0000008	0.000002	0.0000006

BIBLIOGRAPHY

[1] M. E. J. Newman. "Resource Letter CS–1: Complex Systems". In: *American Journal of Physics* 79.8 (July 2011), pp. 800–810.

[2] Marko Gosak et al. "Network science of biological systems at different scales: A review". In: *Physics of Life Reviews* 24 (2018), pp. 118–135. ISSN: 1571-0645.

[3] Jingyi Lin and Yifang Ban. "Complex Network Topology of Transportation Systems". In: *Transport Reviews* 33.6 (2013), pp. 658–685.

[4] Fabio Caccioli, Paolo Barucca, and Teruyoshi Kobayashi. "Network models of financial systemic risk: a review". In: *Journal of Computational Social Science* 1 (2018), pp. 81–114.

[5] S. Boccaletti et al. "Complex networks: Structure and dynamics". In: *Physics Reports* 424.4 (2006), pp. 175–308.

[6] Purna Chandra Biswal. *Discrete mathematics and graph theory*. PHI Learning Pvt. Ltd., 2015.

[7] Leonhard Euler. "Solutio problematis ad geometriam situs pertinentis". In: *Commentarii academiae scientiarum Petropolitanae* (1741), pp. 128–140.

[8] Taha Mostafaie, Farzin Modarres Khiyabani, and Nima Jafari Navimipour. "A systematic study on meta-heuristic approaches for solving the graph coloring problem". In: *Computers & Operations Research* 120 (2020), p. 104850.

[9] Christian Sommer. "Shortest-Path Queries in Static Networks". In: *ACM Comput. Surv.* 46.4 (Mar. 2014).

[10] Stanley Wasserman and Katherine Faust. "Social network analysis: Methods and applications". In: (1994).

[11] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. "Network Flows: Theory, Algorithms, and Applications". In: 1993. URL: https://api.semanticscholar.org/CorpusID:12577796.

[12] Mark Newman, Albert-László Barabási, and Duncan J. Watts. *The Structure and Dynamics of Networks*. Princeton: Princeton University Press, 2006.

[13] P Van Mieghem et al. "A framework for computing topological network robustness". In: *Delft University of Technology, Report20101218* (2010).

[14] Milena Oehlers and Benjamin Fabian. "Graph metrics for network robustness—A survey". In: *Mathematics* 9.8 (2021), p. 895.

[15] Wendy Ellens and Robert E. Kooij. "Graph measures and network robustness". In: *CoRR* abs/1311.5064 (2013). arXiv: 1311.5064. URL: http://arxiv.org/abs/1311.5064.

[16] A. N. Bishop and I. Shames. “Link operations for slowing the spread of disease in complex networks”. In: *Europhysics Letters* 95.1 (June 2011), p. 18005.

[17] Hau Chan and Leman Akoglu. “Optimizing Network Robustness by Edge Rewiring: A General Framework”. In: *Data Min. Knowl. Discov.* 30.5 (Sept. 2016), pp. 1395–1425.

[18] Wu Jun et al. “Natural Connectivity of Complex Networks”. In: *Chinese Physics Letters* 27.7 (July 2010), p. 078902.

[19] Miroslav Fiedler. “Algebraic connectivity of graphs”. In: *Czechoslovak mathematical journal* 23.2 (1973), pp. 298–305.

[20] John S. Baras and Pedram Hovareshti. “Efficient and robust communication topologies for distributed decision making in networked systems”. In: *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*. 2009, pp. 3751–3756.

[21] Scott Freitas et al. “Graph Vulnerability and Robustness: A Survey”. In: *IEEE Transactions on Knowledge and Data Engineering* 35.6 (2023), pp. 5915–5934.

[22] Yang Lou, Lin Wang, and Guanrong Chen. “Structural Robustness of Complex Networks: A Survey of A Posteriori Measures[Feature]”. In: *IEEE Circuits and Systems Magazine* 23.1 (2023), pp. 12–35.

[23] Christian M. Schneider et al. “Mitigation of malicious attacks on networks”. In: *Proceedings of the National Academy of Sciences* 108.10 (2011), pp. 3838–3841.

[24] Yang-Yu Liu, Jean-Jacques Slotine, and Albert-László Barabási. “Controllability of complex networks”. In: *Nature* 473.7346 (2011), pp. 167–173.

[25] Zhengzhong Yuan et al. “Exact controllability of complex networks”. In: *Nature Communications* 4.1 (2013), pp. 1–9.

[26] Yong Lu et al. “Measuring and Improving Communication Robustness of Networks”. In: *IEEE Communications Letters* 23.12 (2019), pp. 2168–2171.

[27] Petter Holme et al. “Attack vulnerability of complex networks”. In: *Phys. Rev. E* 65 (5 May 2002), p. 056109.

[28] Anqi Chen, Peng Sun, and Robert E. Kooij. “The Recoverability of Network Controllability”. In: *2021 5th International Conference on System Reliability and Safety (ICSRS)*. 2021, pp. 198–208.

[29] Brandon Heller, Rob Sherwood, and Nick McKeown. “The controller placement problem”. In: *ACM SIGCOMM Computer Communication Review* 42.4 (2012), pp. 473–478.

[30] V. Li and J. Silvester. “Performance Analysis of Networks with Unreliable Components”. In: *IEEE Transactions on Communications* 32.10 (1984), pp. 1105–1110.

[31] Vaibhav Gaur et al. “A literature review on network reliability analysis and its engineering applications”. In: *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability* 235.2 (2021), pp. 167–181.

[32] Zhidong He, Peng Sun, and Piet Van Mieghem. “Topological Approach to Measure Network Recoverability”. In: *2019 11th International Workshop on Resilient Networks Design and Modeling (RNDM)*. 2019, pp. 1–7.

[33] Tanzina Afrin. “Resilience Assessment for Complex Networks Based on Recovery Strategies”. In: (2019).

[34] Xing Pan and Huixiong Wang. “Resilience of and recovery strategies for weighted networks”. In: *PloS one* 13.9 (2018).

[35] Ze Wang et al. “Target recovery of the economic system based on the target reinforcement path method”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 32.9 (2022).

[36] Weiman Sun and An Zeng. “Target recovery in complex networks”. In: *The European Physical Journal B* 90.1 (2017), pp. 1–6.

[37] Sheng Hong et al. “Cascading failure analysis and restoration strategy in an interdependent network”. In: *Journal of Physics A: Mathematical and Theoretical* 49.19 (2016), p. 195101.

[38] Peng Sun et al. “Topological approach to measure the recoverability of optical networks”. In: *Optical Switching and Networking* 41 (2021), p. 100617.

[39] Abdollah Younesi et al. “Trends in modern power systems resilience: State-of-the-art review”. In: *Renewable and Sustainable Energy Reviews* 162 (2022), p. 112397. ISSN: 1364-0321.

[40] Yutian Liu, Rui Fan, and Vladimir Terzija. “Power system restoration: a literature review from 2006 to 2016”. In: *Journal of Modern Power Systems and Clean Energy* 4.3 (2016), pp. 332–341.

[41] Wei Sun, Chen-Ching Liu, and Li Zhang. “Optimal generator start-up strategy for bulk power system restoration”. In: *IEEE Transactions on Power Systems* 26.3 (2010), pp. 1357–1366.

[42] Jiajing Wu et al. “Sequential topology recovery of complex power systems based on reinforcement learning”. In: *Physica A: Statistical Mechanics and its Applications* 535 (2019), p. 122487.

[43] Yihan Zhang et al. “Sequential Node/Link Recovery Strategy of Power Grids Based on Q-Learning Approach”. In: *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*. 2019, pp. 1–5.

[44] Fred Brauer, Carlos Castillo-Chavez, and Carlos Castillo-Chavez. *Mathematical models in population biology and epidemiology*. Vol. 2. 40. Springer, 2012.

[45] Romualdo Pastor-Satorras et al. “Epidemic processes in complex networks”. In: *Rev. Mod. Phys.* 87 (3 Aug. 2015), pp. 925–979.

[46] M. Nekovee et al. “Theory of rumour spreading in complex social networks”. In: *Physica A: Statistical Mechanics and its Applications* 374.1 (2007), pp. 457–470.

[47] M. Garetto, W. Gong, and D. Towsley. “Modeling malware spreading dynamics”. In: *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428)*. Vol. 3. 2003, 1869–1879 vol.3.

[48] Lorenzo Zino and Ming Cao. “Analysis, Prediction, and Control of Epidemics: A Survey from Scalar to Dynamic Network Models”. In: *IEEE Circuits and Systems Magazine* 21.4 (2021), pp. 4–23.

[49] Cong Li, Ruud van de Bovenkamp, and Piet Van Mieghem. “Susceptible-infected-susceptible model: A comparison of N-intertwined and heterogeneous mean-field approximations”. In: *Phys. Rev. E* 86 (2 Aug. 2012), p. 026116.

[50] Alison L Hill et al. “Emotions as infectious diseases in a large social network: the SISa model”. In: *Proceedings of the Royal Society B: Biological Sciences* 277.1701 (2010), pp. 3827–3835.

[51] Piet Van Mieghem. “Explosive phase transition in susceptible-infected-susceptible epidemics with arbitrary small but nonzero self-infection rate”. In: *Phys. Rev. E* 101 (3 Mar. 2020), p. 032303.

[52] Fenghua Wang and Robert E. Kooij. “Robustness of Network Controllability with Respect to Node Removals”. In: *Complex Networks and Their Applications XI*. Ed. by Hocine Cherifi et al. Cham: Springer International Publishing, 2023, pp. 383–394. ISBN: 978-3-031-21131-7.

[53] Fenghua Wang and Robert E. Kooij. “Robustness of Network Controllability with Respect to Node Removals Based on In-Degree and Out-Degree”. In: *Entropy* 25.4 (2023), p. 656.

[54] Lin Wu et al. “Controllability and its applications to biological networks”. In: *Journal of Computer Science and Technology* 34.1 (2019), pp. 16–34.

[55] Marco Rinaldi. “Controllability of transportation networks”. In: *Transportation Research Part B: Methodological* 118 (2018), pp. 381–406.

[56] Philip C Solimine. “Network Controllability Metrics for Corruption Research”. In: *Corruption Networks*. Springer, 2021, pp. 29–50.

[57] R. E. Kalman. “Mathematical Description of Linear Dynamical Systems”. In: *Journal of the Society for Industrial and Applied Mathematics Series A Control* 1.2 (1963), pp. 152–192.

[58] Ching-Tai Lin. “Structural controllability”. In: *IEEE Transactions on Automatic Control* 19.3 (1974), pp. 201–208.

[59] Jiajia Jia et al. “A Unifying Framework for Strong Structural Controllability”. In: *IEEE Transactions on Automatic Control* 66.1 (2021), pp. 391–398.

[60] Alex Olshevsky. “Minimal controllability problems”. In: *IEEE Transactions on Control of Network Systems* 1.3 (2014), pp. 249–258.

[61] Noah J. Cowan et al. “Nodal Dynamics, Not Degree Distributions, Determine the Structural Controllability of Complex Networks”. In: *PloS ONE* 7.6 (June 2012), pp. 1–5.

[62] Cun-Lai Pu, Wen-Jiang Pei, and Andrew Michaelson. “Robustness analysis of network controllability”. In: *Physica A: Statistical Mechanics and its Applications* 391.18 (2012), pp. 4420–4425.

[63] Zhe-Ming Lu and Xin-Feng Li. “Attack Vulnerability of Network Controllability”. In: *PLOS ONE* 11.9 (Sept. 2016), pp. 1–27.

[64] Lifu Wang et al. “Controllability and Optimization of Complex Networks Based on Bridges”. In: *Complexity* 2020 (Dec. 2020), pp. 1–10.

[65] Peng Sun, Robert E. Kooij, and Piet Van Mieghem. “Reachability-Based Robustness of Controllability in Sparse Communication Networks”. In: *IEEE Transactions on Network and Service Management* 18.3 (2021), pp. 2764–2775.

[66] Yang Lou, Lin Wang, and Guanrong Chen. “A framework of hierarchical attacks to network controllability”. In: *Communications in Nonlinear Science and Numerical Simulation* 98 (2021), p. 105780.

[67] Giulia Menichetti, Luca Dall’Asta, and Ginestra Bianconi. “Network Controllability Is Determined by the Density of Low In-Degree and Out-Degree Nodes”. In: *Phys. Rev. Lett.* 113 (7 Aug. 2014), p. 078701.

[68] Yang Lou et al. “Controllability Robustness of Henneberg-Growth Complex Networks”. In: *IEEE Access* 10 (2022), pp. 5103–5114.

[69] Zenghu Zhang et al. “Optimization of robustness of interdependent network controllability by redundant design”. In: *PLOS ONE* 13.2 (Feb. 2018), pp. 1–17.

[70] Ashish Dhiman, Peng Sun, and Robert E. Kooij. “Using machine learning to quantify the robustness of network controllability”. In: *Machine Learning for Networking: Third International Conference, MLN 2020, Paris, France, November 24–26, 2020, Revised Selected Papers 3*. Springer. 2021, pp. 19–39.

[71] Mohammad Komareji and Roland Bouffanais. “Resilience and Controllability of Dynamic Collective Behaviors”. In: *PLOS ONE* 8.12 (Dec. 2013), pp. 1–15.

[72] Simon Knight et al. “The Internet Topology Zoo”. In: *IEEE Journal on Selected Areas in Communications* 29.9 (2011), pp. 1765–1775.

[73] Rudolf E Kalman. “On the general theory of control systems”. In: *Proceedings First International Conference on Automatic Control, Moscow, USSR*. 1960, pp. 481–492.

[74] Yang-Yu Liu and Albert-László Barabási. “Control principles of complex systems”. In: *Rev. Mod. Phys.* 88 (3 Sept. 2016), p. 035006.

[75] Ching-Tai Lin. “Structural controllability”. In: *IEEE Transactions on Automatic Control* 19.3 (1974), pp. 201–208.

[76] John E Hopcroft and Richard M Karp. “An nˆ5/2 algorithm for maximum matchings in bipartite graphs”. In: *SIAM Journal on Computing* 2.4 (1973), pp. 225–231.

[77] Dror Y Kenett et al. “Network of interdependent networks: overview of theory and applications”. In: *Networks of Networks: The Last Frontier of Complexity* (2014), pp. 3–36.

[78] Jia Shao et al. “Structure of shells in complex networks”. In: *Phys. Rev. E* 80 (3 Sept. 2009), p. 036105.

[79] Fenghua Wang and Robert E. Kooij. “The recoverability of network controllability with respect to node additions”. In: *New Journal of Physics* 25.10 (2023), p. 103034.

[80] Jie Lu et al. “A survey of controller placement problem in software-defined networking”. In: *IEEE Access* 7 (2019), pp. 24290–24307.

[81] Yuan Zhang et al. “A survey on software defined networking with multiple controllers”. In: *Journal of Network and Computer Applications* 103 (2018), pp. 101–118.

[82] Mili Dhar et al. “A comprehensive study of different objectives and solutions of controller placement problem in software-defined networks”. In: *Transactions on Emerging Telecommunications Technologies* 33.5 (2022), e4440.

[83] Tao Hu et al. “Multi-controller based software-defined networking: A survey”. In: *IEEE Access* 6 (2018), pp. 15980–15996.

[84] Mili Dhar et al. “A comprehensive study of different objectives and solutions of controller placement problem in software-defined networks”. In: *Transactions on Emerging Telecommunications Technologies* 33.5 (2022), e4440.

[85] Tianshu Li et al. “Approximation Algorithms for Controller Placement Problems in Software Defined Networks”. In: *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*. 2018, pp. 250–257. DOI: 10.1109/DSC.2018.00043.

[86] Mu He et al. “Modeling flow setup time for controller placement in SDN: Evaluation for dynamic flows”. In: *2017 IEEE International Conference on Communications (ICC)*. IEEE. 2017, pp. 1–7.

[87] Yannan Hu et al. “On reliability-optimized controller placement for software-defined networks”. In: *China Communications* 11.2 (2014), pp. 38–54.

[88] Francisco Javier Ros and Pedro Miguel Ruiz. “Five nines of southbound reliability in software-defined networks”. In: *Proceedings of the third workshop on Hot topics in software defined networking*. 2014, pp. 31–36.

[89] Qinghong Zhong et al. “A min-cover based controller placement approach to build reliable control network in SDN”. In: *NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium*. IEEE. 2016, pp. 481–487.

[90] James L Beck and Konstantin M Zuev. “Rare event simulation”. In: *arXiv preprint arXiv:1508.05047* (2015).

[91] Jacques Carlier and Corinne Lucet. “A decomposition algorithm for network reliability evaluation”. In: *Discrete Applied Mathematics* 65.1-3 (1996), pp. 141–156.

[92] Hans L. Bodlaender. “A partial k-arboretum of graphs with bounded treewidth”. In: *Theoretical Computer Science* 209.1 (1998), pp. 1–45. ISSN: 0304-3975. DOI: https://doi.org/10.1016/S0304-3975(97)00228-4.

[93] Hans L. Bodlaender and Ton Kloks. “Efficient and Constructive Algorithms for the Pathwidth and Treewidth of Graphs”. In: *Journal of Algorithms* 21.2 (1996), pp. 358–402. ISSN: 0196-6774. DOI: https://doi.org/10.1006/jagm.1996.0049. URL: https://www.sciencedirect.com/science/article/pii/S0196677496900498.

[94] Michael O. Ball, Charles J. Colbourn, and J. Scott Provan. “Chapter 11 Network reliability”. In: *Network Models*. Vol. 7. Handbooks in Operations Research and Management Science. Elsevier, 1995, pp. 673–762. DOI: https://doi.org/10.1016/S0927-0507(05)80128-8.

[95] Sourabh Katoch, Sumit Singh Chauhan, and Vijay Kumar. “A review on genetic algorithm: past, present, and future”. In: *Multimedia Tools and Applications* 80 (2021), pp. 8091–8126.

[96] Osman Alp, Erhan Erkut, and Zvi Drezner. “An efficient genetic algorithm for the p-median problem”. In: *Annals of Operations research* 122 (2003), pp. 21–42.

[97] Jinghui Zhong et al. “Comparison of performance between different selection strategies on simple genetic algorithms”. In: *International conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce (CIMCA-IAWTIC’06)*. Vol. 2. IEEE. 2005, pp. 1115–1121.

[98] Raissa M D’Souza, Mario di Bernardo, and Yang-Yu Liu. “Controlling complex networks with complex nodes”. In: *Nature Reviews Physics* 5.4 (2023), pp. 250–262.

[99] Lucas Cuadra et al. “A Critical Review of Robustness in Power Grids Using Complex Networks Concepts”. In: *Energies* 8.9 (2015), pp. 9211–9265.

[100] Jingyi Lin and Yifang Ban. “Complex Network Topology of Transportation Systems”. In: *Transport Reviews* 33.6 (2013), pp. 658–685.

[101] Dan Zhang et al. “Analysis and synthesis of networked control systems: A survey of recent advances and challenges”. In: *ISA Transactions* 66 (2017), pp. 376–392.

[102] Guanrong Chen. “Pinning control and controllability of complex dynamical networks”. In: *International Journal of Automation and Computing* 14 (2017), pp. 1–9.

[103] Peng Sun et al. “Quantifying the robustness of network controllability”. In: *2019 4th International Conference on System Reliability and Safety (ICSRS)*. IEEE. 2019, pp. 66–76.

[104] Yilun Shang. “Consensus recovery from intentional attacks in directed nonlinear multi-agent systems”. In: *International Journal of Nonlinear Sciences and Numerical Simulation* 14.6 (2013), pp. 355–361.

[105] Rudolf E Kalman. “On the general theory of control systems”. In: *Proceedings First International Conference on Automatic Control, Moscow, USSR*. 1960, pp. 481–492.

[106] Alex Arenas et al. “Synchronization in complex networks”. In: *Physics Reports* 469.3 (2008), pp. 93–153.

[107] Abdollah Amirkhani and Amir Hossein Barshooi. “Consensus in multi-agent systems: a review”. In: *Artificial Intelligence Review* 55.5 (2022), pp. 3897–3935.

[108] Réka Albert, Hawoong Jeong, and Albert-László Barabási. “Error and attack tolerance of complex networks”. In: *Nature* 406.6794 (2000), pp. 378–382.

[109] Seyedmohsen Hosseini, Kash Barker, and Jose E. Ramirez-Marquez. “A review of definitions and measures of system resilience”. In: *Reliability Engineering & System Safety* 145 (2016), pp. 47–61.

[110] Yilun Shang. “Localized recovery of complex networks against failure”. In: *Scientific Reports* 6.1 (2016), p. 30521.

[111] Yang Lou, Lin Wang, and Guanrong Chen. “A framework of hierarchical attacks to network controllability”. In: *Communications in Nonlinear Science and Numerical Simulation* 98 (2021), p. 105780.

[112] Ashish Dhiman, Peng Sun, and Robert E. Kooij. “Using Machine Learning to Quantify the Robustness of Network Controllability”. In: *Machine Learning for Networking*. Ed. by Éric Renault, Selma Boumerdassi, and Paul Mühlethaler. Cham: Springer International Publishing, 2021, pp. 19–39. ISBN: 978-3-030-70866-5.

[113] Yang Lou et al. “Predicting Network Controllability Robustness: A Convolutional Neural Network Approach”. In: *IEEE Transactions on Cybernetics* 52.5 (2022), pp. 4052–4063.

[114] Yilun Shang and Roland Bouffanais. “Influence of the number of topologically interacting neighbors on swarm dynamics”. In: *Scientific Reports* 4.1 (2014), p. 4184.

[115] Jeff Alstott, Ed Bullmore, and Dietmar Plenz. “Powerlaw: A Python Package for Analysis of Heavy-Tailed Distributions”. In: *PLOS ONE* 9.1 (Jan. 2014), pp. 1–11.

[116] M. E. J. Newman, S. H. Strogatz, and D. J. Watts. “Random graphs with arbitrary degree distributions and their applications”. In: *Phys. Rev. E* 64 (2 July 2001), p. 026118.

[117] Albert-László Barabási and Réka Albert. “Emergence of scaling in random networks”. In: *science* 286.5439 (1999), pp. 509–512.

[118] Ryan A. Rossi and Nesreen K. Ahmed. “The Network Data Repository with Interactive Graph Analytics and Visualization”. In: *AAAI*. 2015. URL: https://networkrepository.com.

[119] Jure Leskovec and Andrej Krevl. *SNAP Datasets: Stanford Large Network Dataset Collection*. http://snap.stanford.edu/data. June 2014.

[120] Carlos Castillo, Kumar Chellapilla, and Ludovic Denoyer. “Web spam challenge 2008”. In: *Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*. 2008.

[121] Paolo Boldi et al. “UbiCrawler: A Scalable Fully Distributed Web Crawler”. In: *Software: Practice & Experience* 34.8 (2004), pp. 711–726.

[122] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. “Signed networks in social media”. In: *Proceedings of the SIGCHI conference on human factors in computing systems*. 2010, pp. 1361–1370.

[123] Ryan A. Rossi et al. “Fast Maximum Clique Algorithms for Large Graphs”. In: *Proceedings of the 23rd International Conference on World Wide Web (WWW)*. 2014.

[124] Hao Yin et al. “Local higher-order graph clustering”. In: *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 2017, pp. 555–564.

[125] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. “Graph evolution: Densification and shrinking diameters”. In: *ACM transactions on Knowledge Discovery from Data (TKDD)* 1.1 (2007), 2–es.

[126] Matei Ripeanu and Ian Foster. “Mapping the gnutella network: Macroscopic properties of large-scale peer-to-peer systems”. In: *Peer-to-Peer Systems: First InternationalWorkshop, IPTPS 2002 Cambridge, MA, USA, March 7–8, 2002 Revised Papers 1*. Springer. 2002, pp. 85–93.

[127] Christian M. Schneider et al. “Mitigation of malicious attacks on networks”. In: *Proceedings of the National Academy of Sciences* 108.10 (2011), pp. 3838–3841.

[128] M. E. J. Newman, S. H. Strogatz, and D. J. Watts. “Random graphs with arbitrary degree distributions and their applications”. In: *Phys. Rev. E* 64 (2 July 2001), p. 026118.

[129] Fenghua Wang et al. “Recovering Power Grids Using Strategies Based on Network Metrics and Greedy Algorithms”. In: *Entropy* 25.10 (2023), p. 1455.

[130] Camillo Nuti, Alessandro Rasulo, and Ivo Vanzi. “Seismic safety evaluation of electric power supply at urban level”. In: *Earthquake engineering & structural dynamics* 36.2 (2007), pp. 245–263.

[131] Xiaoxin Zhou and Changyou Yan. “A blackout in Hainan Island power system: Causes and restoration procedure”. In: *2008 IEEE Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century*. IEEE. 2008, pp. 1–5.

[132] Chong Wang et al. “A systematic review on power system resilience from the perspective of generation, network, and load”. In: *Renewable and Sustainable Energy Reviews* 167 (2022), p. 112567. ISSN: 1364-0321.

[133] A Muir and J Lopatto. *Final Report on the August 14, 2003 Blackout in the United States and Canada: Causes and Recommendations*. Tech. rep. U.S.-Canada Power System Outage Task Force, Apr. 2004.

[134] Kristina Hamachi LaCommare and Joseph H Eto. *Understanding the cost of power interruptions to US electricity consumers*. Tech. rep. Lawrence Berkeley National Lab.(LBNL), Berkeley, CA (United States), 2004.

[135] Paul Hines, Jay Apt, and Sarosh Talukdar. “Large blackouts in North America: Historical trends and policy implications”. In: *Energy Policy* 37.12 (2009), pp. 5249–5259.

[136] Hale Cetinay, Karel Devriendt, and Piet Van Mieghem. “Nodal vulnerability to targeted attacks in power grids”. In: *Applied network science* 3.1 (2018), p. 34.

C

[137] Benjamin Schäfer et al. “Dynamically induced cascading failures in power grids”. In: *Nature communications* 9.1 (2018), pp. 1–13.

[138] Giuliano Andrea Pagani and Marco Aiello. “The power grid as a complex network: a survey”. In: *Physica A: Statistical Mechanics and its Applications* 392.11 (2013), pp. 2688–2700.

[139] Roberto Rocchetta. “Enhancing the resilience of critical infrastructures: Statistical analysis of power grid spectral clustering and post-contingency vulnerability metrics”. In: *Renewable and Sustainable Energy Reviews* 159 (2022), p. 112185. ISSN: 1364-0321.

[140] Alexis Kyriacou et al. “Controlled Islanding Solution for Large-Scale Power Systems”. In: *IEEE Transactions on Power Systems* 33.2 (2018), pp. 1591–1602.

[141] S Arash Nezam Sarmadi et al. “A sectionalizing method in power system restoration based on WAMS”. In: *IEEE Transactions on Smart Grid* 2.1 (2011), pp. 190–197.

[142] Qingming Li et al. “Integrating Reinforcement Learning and Optimal Power Dispatch to Enhance Power Grid Resilience”. In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 69.3 (2022), pp. 1402–1406.

[143] Jiajing Wu et al. “Sequential Recovery of Complex Networks Suffering From Cascading Failure Blackouts”. In: *IEEE Transactions on Network Science and Engineering* 7.4 (2020), pp. 2997–3007.

[144] Mosayeb Afshari Igder and Xiaodong Liang. “Service Restoration Using Deep Reinforcement Learning and Dynamic Microgrid Formation in Distribution Networks”. In: *IEEE Transactions on Industry Applications* 59.5 (2023), pp. 5453–5472.

[145] Wei-Chang Yeh et al. “New genetic algorithm for economic dispatch of standalone three-modular microgrid in DongAo Island”. In: *Applied Energy* 263 (2020), p. 114508. ISSN: 0306-2619.

[146] Stojan Trajanovski et al. “Robustness envelopes of networks”. In: *Journal of Complex Networks* 1.1 (Mar. 2013), pp. 44–62.

[147] Piet Van Mieghem, Karel Devriendt, and H Cetinay. “Pseudoinverse of the Laplacian and best spreader node in a network”. In: *Physical Review E* 96.3 (2017), p. 032311.

[148] Yakup Koç et al. “MATCASC: A tool to analyse cascading line outages in power grids”. In: *2013 IEEE International Workshop on Inteligent Energy Systems (IWIES)*. 2013, pp. 143–148.

[149] Ray Daniel Zimmerman, Carlos Edmundo Murillo-Sánchez, and Robert John Thomas. “MATPOWER: Steady-State Operations, Planning, and Analysis Tools for Power Systems Research and Education”. In: *IEEE Transactions on Power Systems* 26.1 (2011), pp. 12–19. DOI: 10.1109/TPWRS.2010.2051168.

[150] O. Alsac and B. Stott. “Optimal Load Flow with Steady-State Security”. In: *IEEE Transactions on Power Apparatus and Systems* PAS-93.3 (1974), pp. 745–751. DOI: 10.1109/TPAS.1974.293972.

[151] G W Bills. "On-line stability analysis study, RP 90-1". In: (Oct. 1970).

[152] Anantha Pai. *Energy function analysis for power system stability*. Springer Science & Business Media, 1989.

[153] T. Athay, R. Podmore, and S. Virmani. "A Practical Method for the Direct Analysis of Transient Stability". In: *IEEE Transactions on Power Apparatus and Systems* PAS-98.2 (1979), pp. 573–584. DOI: 10.1109/TPAS.1979.319407.

[154] Richard D. Christie. *The UW Power System Test Case Archive*. Aug. 1999. URL: https://labs.ece.uw.edu/pstca/pg_tcaintro.htm.

[155] Hale Cetinay et al. "Comparing the effects of failures in power grids under the AC and DC power flow models". In: *IEEE Transactions on Network Science and Engineering* 5.4 (2017), pp. 301–312.

[156] Adilson E. Motter and Ying-Cheng Lai. "Cascade-based attacks on complex networks". In: *Phys. Rev. E* 66 (6 Dec. 2002), p. 065102.

[157] Piet Van Mieghem et al. "Decreasing the spectral radius of a graph by link removals". In: *Phys. Rev. E* 84 (1 July 2011), p. 016101.

[158] Mark EJ Newman and Michelle Girvan. "Finding and evaluating community structure in networks". In: *Physical review E* 69.2 (2004), p. 026113.

[159] Mark EJ Newman. "A measure of betweenness centrality based on random walks". In: *Social networks* 27.1 (2005), pp. 39–54.

[160] Mark EJ Newman. "The mathematics of networks". In: *The new palgrave encyclopedia of economics* 2.2008 (2008), pp. 1–12.

[161] Linton C Freeman. "Centrality in social networks conceptual clarification". In: *Social networks* 1.3 (1978), pp. 215–239.

[162] Hale Cetinay, Fernando A Kuipers, and Piet Van Mieghem. "A topological investigation of power flow". In: *IEEE Systems Journal* 12.3 (2016), pp. 2524–2532.

[163] Wendy Ellens et al. "Effective graph resistance". In: *Linear algebra and its applications* 435.10 (2011), pp. 2491–2506.

[164] Piet Van Mieghem and Fenghua Wang. "Time dependence of susceptible-infected-susceptible epidemics on networks with nodal self-infections". In: *Physical Review E* 101.5 (2020), p. 052310.

[165] F. Altarelli et al. "Containing Epidemic Outbreaks by Message-Passing Techniques". In: *Phys. Rev. X* 4 (2 May 2014), p. 021024.

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

[167] Piet Van Mieghem. "The N-intertwined SIS epidemic network model". In: *Computing* 93.2-4 (2011), pp. 147–169.

[168] P. Van Mieghem and R. van de Bovenkamp. "Accuracy criterion for the mean-field approximation in susceptible-infected-susceptible epidemics on networks". In: *Phys. Rev. E* 91 (3 Mar. 2015), p. 032812.

[169] E. Cator and P. Van Mieghem. “Susceptible-infected-susceptible epidemics on the complete graph and the star graph: Exact analysis”. In: *Phys. Rev. E* 87 (1 Jan. 2013), p. 012811.

[170] Siyang Jiang et al. “Searching for Key Cycles in a Complex Network”. In: *Phys. Rev. Lett.* 130 (18 May 2023), p. 187402.

[171] Tianlong Fan et al. “Characterizing cycle structure in complex networks”. In: *Communications Physics* 4.1 (2021), p. 272.

Curriculum Vitæ

Fenghua Wang was born in Hubei, China, on November 11, 1994, in the Chinese lunar calendar. She received her Bachelor's degree in Management Science and her Master's degree in System Theory from Beijing Normal University in 2016 and 2019. Since September 2019, she has been a Ph.D. candidate in the Architectures and Services (NAS) group, faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, under the supervision of Prof.dr.ir. Robert E. Kooij and Prof.dr.ir. Piet Van Mieghem.

Her research interests are in the robustness and recoverability of complex networks, the dynamics of complex networks and network modeling.

LIST OF PUBLICATIONS

9. **F. Wang**, J. Zou and R. E. Kooij, *Analytical approximation for ATTR with respect to node removals*, In 2024 20th International Conference on the Design of Reliable Communication Networks (DRCN), pp. 24-31. IEEE, 2024.

8. D. Martínez, S.i Bergillos, L. Corominas, J. Comas, **F. Wang**, R. E. Kooij and E. Calle, *Enhancing water distribution network resilience with cost-effective meshing*, Science of the Total Environment 938 (2024): 173051.

7. R. Xu, **F. Wang** and R. E. Kooij, *Controller placement with respect to controller reachability*, 7th International Conference on System Reliability and Safety (ICSRS 2023), 22-24 November, Bologna, Italy.

6. **F. Wang** and R. E. Kooij, *The recoverability of network controllability with respect to node additions*, New Journal of Physics **25**, no. 10 (2023): 103034.

5. **F. Wang**, H. Cetinay, Z. He, L. Liu, P. Van Mieghem and R. E. Kooij, *Recovering Power Grids Using Strategies Based on Network Metrics and Greedy Algorithms*, Entropy **25**, no. 10 (2023): 1455.

4. **F. Wang** and R. E. Kooij, *Robustness of Network Controllability with Respect to Node Removals Based on In-Degree and Out-Degree*, Entroy **25**, no. 4 (2023): 656.

3. **F. Wang** and R. E. Kooij, *Robustness of Network Controllability with Respect to Node Removals*, In International Conference on Complex Networks and Their Applications pp. 383-394. Cham: Springer International Publishing, 2022.

2. B. Chang, L. Yang, M. Sensi, M. A. Achterberg, **F. Wang**, M. Rinaldi and P. Van Mieghem. *Markov Modulated Process to model human mobility*, In International Conference on Complex Networks and Their Applications, pp. 607-618. Cham: Springer International Publishing, 2021.

1. P. Van Mieghem and **F. Wang**, *Time dependence of susceptible-infected-susceptible epidemics on networks with nodal self-infections*, Physical Review E **101**, no. 5 (2020): 052310.