Computational and Analytical Tools for Resilient and Secure Power Grids

Saleh Soltan

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2017

©2017
Saleh Soltan
All Rights Reserved

ABSTRACT

Computational and Analytical Tools for Resilient and Secure Power Grids

by

Saleh Soltan

Enhancing power grids' performance and resilience has been one of the greatest challenges in engineering and science over the past decade. A recent report by the National Academies of Sciences, Engineering, and Medicine along with other studies emphasizes the necessity of deploying new ideas and mathematical tools to address the challenges facing the power grids now and in the future. To fulfil this necessity, numerous grid modernization programs have been initiated in recent years.

This thesis focuses on one of the most critical challenges facing power grids which is their vulnerability against failures and attacks. Our approach bridges concepts in power engineering and computer science to improve power grids resilience and security. We *analyze the vulnerability of power grids to cyber and physical attacks and failures*, *design efficient monitoring schemes for robust state estimation*, *develop algorithms to control the grid under tension*, and *introduce methods to generate realistic power grid test cases*. Our contributions can be divided into four major parts:

Power Grid State Prediction: Large scale power outages in Australia (2016), Ukraine (2015), Turkey (2015), India (2013), and the U.S. (2011, 2003) have demonstrated the vulnerability of power grids to cyber and physical attacks and failures. Power grid outages have devastating effects on almost every aspect of modern life as well as on interdependent systems. Despite their inevitability, the effects of failures on power grids' performance can be limited if the system operator can predict and understand the consequences of an initial failure and can immediately detect the problematic failures. To enable these capabilities, we study failures in power grids using computational and analytical tools based on the DC power flow model. We introduce new

metrics to efficiently evaluate the severity of an initial failure and develop efficient algorithms to predict its consequences. We further identify power grids' vulnerabilities using these metrics and algorithms.

Power Grid State Estimation: In order to obtain an accurate prediction of the subsequent effects of an initial failure on the performance of the grid, the system operator needs to exactly know when and where the initial failure has happened. However, due to lack of enough measurement devices or a cyber attack on the grid, such information may not be available directly to the grid operator via measurements. To address this problem, we develop efficient methods to estimate the state of the grid and detect failures (if any) from partial available information.

Power Grid Control: Once an initial failure is detected, prediction methods can be used to predict the subsequent effects of that failure. If the initial failure is causing a cascade of failures in the grid, a control mechanism needs to be applied in order to mitigate its further effects. Power Grid Islanding is an effective method to mitigate cascading failures. The challenge is to partition the network into smaller connected components, called *islands*, so that each island can operate independently for a short period of time. This is to prevent the system to be separated into unbalanced parts due to cascading failures. To address this problem, we introduce and study the Doubly Balanced Connected graph Partitioning (DBCP) problem and provide an efficient algorithm to partition the power grid into two operating islands.

Power Grid Test Cases for Evaluation: In order to evaluate algorithms that are developed for enhancing power grids resilience, one needs to study their performance on the real grid data. However, due to security reasons, such data sets are not publicly available and are very hard to obtain. Therefore, we study the structural properties of the U.S. Western Interconnection grid (WI), and based on the results we present the Network Imitating Method Based on LEarning (NIMBLE) for generating synthetic spatially embedded networks with similar properties to a given grid. We apply NIMBLE to the WI and show that the generated network has similar structural and spatial properties as well as the same level of robustness to cascading failures.

Overall, the results provided in this thesis advance power grids' resilience and security by providing a better understanding of the system and by developing efficient algorithms to protect it at the time of failure.

Table of Contents

List of Figures

List of Tables

Acknowledgments

First and foremost, I would like to thank my Ph.D. advisor, Professor Gil Zussman. I have been extremely lucky and fortunate to have him as my academic advisor in the past 6 years. He inspired me with his unique vision, encouraged me with his genuine support, and enlighten me with his wisdom and knowledge. I am grateful for so many things he thought me, but above all, I will forever be grateful for his friendship.

I owe special thanks to Professor Mihalis Yannakakis for his generous help with my research. His office door has always been open to me whenever I needed his advice and opinion about a research problem. His astounding knowledge in mathematics and his exceptional insight in problem solving have been a true source of inspiration for me.

I would like to extend my special thanks to Professor Daniel Bienstock and Professor Augustin Chaintreau for their guidance and support in many occasions. I would also like to express my appreciation to Professor Debasis Mitra and Professor Matthias Preindl for agreeing to be in my thesis committee.

During my Ph.D. studies, I have been fortunate to collaborate with many distinguished researchers including: Professor Sergey Buldyrev, Professor Fernando Kuipers, Dr. Dorian Mazauric, Hale Cetinay, Russell Spiewak, and Alexander Loh. I am very grateful for the opportunity.

As a member of the Wim.net research lab, although I did not get to collaborate with many of the other talented members, I am sincerely thankful for their help and support. In particular, I would like to thank Dr. Berk Birand, Tingjun Chen, Dr. Jelena Diakonikolas (who has also been a true friend), Dr. Maria Gorlatova, Dr. Varun Gupta, Craig Gutterman, and Dr. Robert Margolis.

I would like to thank Elsa Sanchez, Associate Director of Student Affairs for guiding me through many administrative tasks along these years. She has always been very helpful. I would also like to thank Dr. Janet Kayfetz for her help in editing and writing of many of my research articles. She has been an amazing teacher and mentor.

None of this would have been possible if it was not for my family's supports and sacrifices. I would like to thank my mom for her unconditional support and love, my dad for inspiring me to always do good, my brother for encouraging me to go the distance, and my sister for always taking care of me. Last but absolutely not least, I would like to offer my sincerest thanks to my dear wife. When I first met her, I had no published articles and were nowhere near finishing my Ph.D. studies. She has been in this journey with me through my successes and failures, through my achievements and disappointments, and through my joys and sorrows. She is the kindest, most loving, and most caring partner one could ever wish for. I love her dearly and thank her so very much for loving me with all of my flaws and weaknesses.

Financial Support: This work was supported in part by CIAN NSF ERC under grant EEC-0812072, NSF grant CNS-1018379, DTRA grants HDTRA1-09-1-0057 and HDTRA1-13-1-0021, DARPA RADICS under contract #FA-8750-16-C-0054, and funding from the U.S. DOE OE as part of the DOE Grid Modernization Initiative.

To my parents, my brother Sadegh, my sister Sanam,

and love of my life Ghazal.

Chapter 1

Introduction

Electric power systems are one of the most essential infrastructure systems in the modern world. The National Academy of Engineering named Electrification as the greatest engineering achievement of the 20^{th} century [51]. Despite their importance, power systems are aging infrastructure that are under stress by operational uncertainties and are facing challenges never envisioned when they were first developed.

Power systems consists of four main parts as depicted in Fig. 1.1:

- **Generators:** Convert energy from coal, natural gas, petroleum, nuclear energy, or renewable resources into electricity,
- **Transmission Network:** High Voltage lines connecting generators to the distribution system of a town, city, or a large industrial plant,
- **Distribution Network:** Provides electricity at various levels to residential, commercial, and industrial consumers. It mostly have a loopless topology,
- **Loads or consumers**.

The main focus of this thesis is the transmission network of the power systems, or as we refer to the *power grid*.

Enhancing power grids performance and resilience has been one of the greatest challenges in engineering and science over the past decade. A recent report by the National Academies of Sciences, Engineering, and Medicine [106] along with other studies emphasizes the necessity of

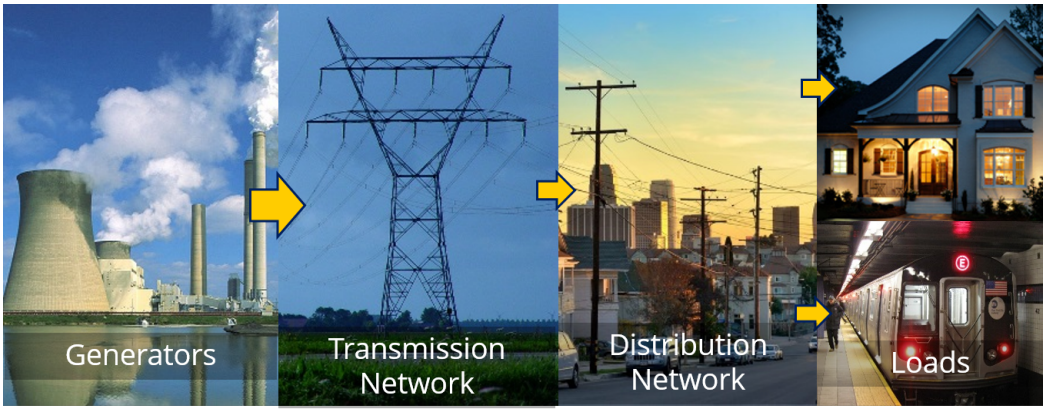


Figure 1.1: The main components of power systems.

deploying new ideas and mathematical tools to address the challenges facing the power grids now and in the future. To fulfil this necessity, numerous grid modernization programs have been initiated in recent years all over the world, including the one by the Department of Energy (DOE) in the U.S.

The DOE envisions that by 2030, the grid will have evolved into an intelligent energy system, a smart grid [106]. Some of the promises of the smart grid are:

- Self-healing from power disturbances
- Resilient operation against both cyber and physical attacks
- Utilizing all generation and energy storage technologies
- Enabling active participation of consumers
- Enabling new energy markets

This thesis focuses on the first two promises listed above which deal with power grids' resilience and security. We study power grids' resilience and security from different angles as described in the following sections. Our approach bridges concepts in power engineering and computer science to improve power grid performance and security. We analyze the vulnerability of power grids to failures, design efficient monitoring schemes for robust state estimation, develop

algorithms to control the grid under tension, and introduce methods to generate realistic power grid test cases.

1.1 Power Flow

Today's power grids deploy the Alternating Current (AC) based systems for power generation and transmission. Electric power in these systems flows from generators to loads based on a set of differential equations which depend on the structure of the network as well as the physical properties of the lines. In the steady state (i.e., fixed frequency), *phasors* are commonly used to compute the AC power flows using a set of nonlinear equations [76].

Under operating conditions, the AC power flow equations can be reduced to a set of linear equations known as the *DC power flow* equations [76]. Due to their linearity, the DC power flow equations are widely used for optimal power flow computation and contingency analysis in power grids [39, 116, 147]. Most of the results provided in this thesis are also based on the DC power flow equations.

In Chapter 2, we provide an overview of the AC and DC power flow equations and study the accuracy of the DC power flows. In particular, in Section 2.4, we analytically derive an upper bound on the difference between the AC and the DC power flows. Results provided in that section appeared in [40].

1.2 Power Grid State Prediction

Large scale power outages in Australia (2016), Turkey (2015), India (2013), and the U.S. (2011, 2003) have demonstrated the vulnerability of power grids to failures caused by natural disasters, such as earthquakes, hurricanes, and solar flares as well as mismanagement and computers malfunctioning. Power grid outages have devastating effects on almost every aspect of modern life as well as on interdependent systems such as telecommunications, gas, water supply, and transportation networks. Follow up reports of these events [7, 8, 9, 12], exposed the insufficient understanding of failures and their consequences in power grids.

Therefore, there is a need to study the vulnerabilities of power grids to failures. The main questions that we seek to answer in this part are:

- How an initial failure may result in a *cascade of failures*?
- How can one predict the evolution of cascading failures?
- What are the most vulnerable parts of the grid to failures?

To answer these questions, in Chapters 3 and 4, we study failures and cascades in power grids using analytical methods.

Analyzing failures and cascades (Chapter 3): We study failures in power grids using computational tools and the DC power flow model. Unlike graph-theoretical network flows, power flows are governed by the laws of physics with no strict capacity bounds on the lines. Nonetheless, there is a rating threshold associated with each line–if the flow exceeds the threshold, the line will eventually experience thermal failure. Such a failure alters network topology, giving rise to a different flow pattern which could cause other line failures. The repetition of this process constitutes a *cascading failure*.

The study of cascading failures in power grids was initiated in [39, 147] which used the linearized DC model and a probabilistic outage rule for overloaded line failures. Similar cascade models have been used to study the properties of the cascades [18, 19, 27, 44, 91], as well as to design control schemes to mitigate the cascade [28, 110] and to detect vulnerable parts of the grid [29, 83]. In Chapter 3, we follow [27, 29] and study cascades due to line overloads in power grids with a *deterministic outage rule*.

First, based on the algebraic properties of the DC power flow equations, we analytically study the impact of a single line failure on the flows on other lines. We illustrate the impact of the distance on the flow increase in lines following a failure and show that unlike in epidemic based models, a failure of a specific line in the grid can affect remote lines, so that the cascade does not necessarily develop contiguously. Then, we introduce metrics to evaluate the robustness of grids to failures, and develop an efficient algorithm to identify the evolution of cascading failures. We compare the severity of the cascade initiated by line failures (selected under different metrics) and proved that finding the set of lines whose removal has the most significant impact is NP-hard.

Our results provides important insights on the consequence of line failures in power grids and can support the development of algorithms for controlling the evolution of a cascade upon failures as well as designing robust power grids.

Our results for this part appeared in [40, 103, 127, 128, 138].

Quantifying the effect of k-line failures (Chapter 4): Contingency analysis in power grids is one of the most effective ways to improve grids' resilience against failures. The main goal of contingency analysis is to detect probable failures in the grid that result in a critical state and deploy preventive measures to avoid such states.

Many great ideas have been developed for contingency analysis in power grids since the advent of the modern power transmission network. *Compensation theorem* and *current injection* methods were used to analyze the effect of line failures in [58, 59, 66, 92, 155]. In particular, [155] introduced the notion of *line outage distribution factors* that inspired many other studies including the work presented in Chapters 3 and 4. [144] used matrix updates to study the effect of two line failures and used the results to introduce an algorithm for the $N-2$ contingency problem. In a follow up work [87], *contingency and Influence graphs* were introduced to study $N-2$ contingency analysis. More optimization-based techniques for contingency analysis of the grids were explored in [30, 47, 63]. A mixed-integer model for the $N-k$ contingency problem was presented and used in [30], while [47] focused on identifying the most probable failure modes in static load distribution using a linear-program. In a more recent innovative paper, probabilistic algorithms were developed to identify collections of multiple contingencies that initiate cascading failure [83].

Due to the large number of possibilities, however, high order contingency analysis is computationally expensive and not fully deployed. In order to circumvent this issue, we analytically compute the redistribution of power flows following a k-line failure (i.e., failures in k distinct lines) using the DC power flow model and based on that introduce the *disturbance value* of a k-line failure. We show that this value can be efficiently computed in $O(1)$ for any set of line failures independently of the size of the grid and can be effectively used to filter out noncritical contingencies. The disturbance value can therefore significantly reduce the time complexity of

contingency analysis by revealing contingencies that are vital for more in depth analysis and pave the way for the deployment of high order contingency analysis in power grids.

Our results for this part appeared in [125, 136].

1.3 Power Grid State Estimation

In order to obtain an accurate prediction of the subsequent effects of an initial failure on the performance of the grid, the system operator needs to exactly know when and where the initial failure has happened. However, due to lack of enough measurement devices or a cyber attack on the grid (e.g., cyber attack on the Ukrainian grid in December 2015 [13]), such information may not be available directly to the grid operator via measurements. Therefore, this information should be estimated from a partial available information. To address this problem, in Chapters 5 and 6, we study cyber-physical attacks on power grids.

Cyber-physical Attacks (DC Model) (Chapter 5): We study an attack where an adversary attacks a zone by disconnecting some of its power lines (*failed lines*) remotely or through a physical attack and blocking the information flow from the zone to the grid control center. We assume that the state of the system is described under the DC power flow model. We use tools from linear algebra and graph theory, and leverage the properties of the DC power flow model to develop methods for information recovery. We prove that by solving a linear program (LP), the failed lines can be detected and the grid's state variables inside the attacked zone can be recovered. We identify sufficient conditions on the zone structure and constraints on the attack characteristics such that the solution to the LP is guaranteed to recover the information.

The considered problem is very similar to the problem of line failure detection using phase angle measurements [73, 142, 143, 163]. Up to two line failures detection, under the DC power flow model, is studied in [142, 143]. Since the provided methods in [142, 143] are greedy-based methods that need to search the entire failure space, the running time of these methods grow exponentially as the number of failures increases. Hence, these methods cannot be generalized to detect higher order failures.

The novelty of our method is the transformation of the line failures detection problem, which is combinatorial in nature, to a convex optimization problem. As a result, our method can be used to detect any set of line failures in a running time that is independent of the size of this set and solely depends on the size of the network.

Our results for this part appeared in [129, 132].

Cyber-physical Attacks (AC Model) (Chapter 6): We extend the results from Chapter 5 to the case when the state of the system is given under the AC power flow model. We present the Convex OPtimization for Statistical State EStimation (COPSSES) Algorithm to detect the failed lines inside the attacked area. The algorithm is based on a convex relaxation of the LP presented in Chapter 5 for information recovery under the DC power flow model. To the best of our knowledge, our work is the first to provide a method for line failures detection under the AC power flow model that can be used to detect any number of line failures.

Our results for this part appeared in [137].

1.4 Power Grid Control

Once an initial failure is detected, the prediction methods mentioned in section 1.2 can be used to predict the subsequent effects of that failure. If the initial failure is causing more subsequent failures in the grid, a control mechanism needs to be applied in order to mitigate its further effects.

One of the most effective of these mechanisms is Power Grid Islanding. The objective is to partition the network into smaller connected components, called *islands*, so that each island can operate independently for a short period of time. This is to prevent the system to be separated into unbalanced parts due to subsequent failures. It also expedites the restoration of the grid after a major failure, by reducing transient stability problems during system reconnection.

In order for an island to operate, it is necessary that the power supply and demand at that island be almost equal (if the supply and demand are not exactly equal but still relatively close, load shedding/generation curtailing can be used in order for the island to operate). Equality of supply and demand in an island, however, may not be sufficient for its independent operation.

It is also important that the infrastructure in that island have the physical capacity to safely carry the power flows. This problem has been studied in the power systems community but almost all the algorithms provided in the literature are heuristic methods that have been shown to be effective only by simulations [71, 107, 121, 139].

In order to provide a theoretical framework for power grid Islanding problem, in Chapter 7, we model and study this problem using graph theory.

Power Grid Islanding (Chapter 7): We introduce and study the Doubly Balanced Connected graph Partitioning (DBCP) problem. The DBCP problem is the problem of partitioning a graph into two parts such that both parts are connected and comparable in size, and supply is almost equal to demand in each part. The idea is that when an island is large enough compared to the initial network, it most likely has enough capacity to carry power flows. In this way, the partitions obtained from solutions to the DBCP problem are operational. We prove that if the graph representing the grid is 2-connected (as it is usually the case), such a partition always exists and can be found in polynomial time. Our method is geometric and used the convex embedding of 3-connected graphs.

Our results for this part appeared in [130, 131].

1.5 Power Grid Test Cases for Evalution

In order to evaluate algorithms that are developed for enhancing power grids resilience, one needs to study their performance on the real grid data. However, due to security reasons, such data sets are not publicly available and are very hard to obtain. There are only very limited test cases and real-world power grid data sets that are publicly and freely available. These include the IEEE test cases [1], the National Grid UK [2], the Polish grid [4], and an approximate model of the European interconnected system [162]. To the best of our knowledge, among these, National Grid UK is the only publicly available dataset with geographical locations. Even if the data was available, it would be unwise to publish vulnerability results which are based on real topologies, due to the enormous cost of grid enhancements. On the other hand, it was recently shown that simple random graph models cannot be used to generate grids with appropriate structural and

spatial characteristics [52]. Therefore, there is a growing interest in generating synthetic power grids [11, 33, 34, 75].

Generating Synthetic Power Grids (Chapter 8): We study the structural properties of the U.S. Western Interconnection grid (WI), and based on the results we present the Network Imitating Method Based on LEarning (NIMBLE) for generating synthetic spatially embedded networks with similar properties to a given grid. We apply NIMBLE to the WI and show that the generated network has similar structural and spatial properties as well as the same level of robustness to cascading failures. NIMBLE uses the Gaussian Mixture Model (GMM) for node density estimation and generates nodes with similar spatial distribution as the nodes in a given network. Then, inspired by the historical evolution of the grids, NIMBLE connects the nodes. This method has several tunable parameters that allow generation of grids similar to any given grid.

Our results for this part appeared in [126, 134, 135].

Chapter 2

Power Flow

In this chapter, we provide an overview of the AC and DC power flow equations as well as some basic definitions and notations that are used in this thesis. Since the DC power flows provide an approximation for the AC power flows, in Section 2.4, we study the accuracy of this approximation by deriving an analytical upper bound on the difference between the AC and the DC power flows.

2.1 Summary of Notations

We represent the power grid by a connected undirected graph $G = (V, E)$ where $V = \{1, 2, \ldots, n\}$ and $E = \{e_1, \ldots, e_m\}$ are the set of nodes and edges corresponding to the buses and transmission lines, respectively. Each edge e_k is a set of two nodes $e_k = \{i, j\}$.

Throughout this thesis, we use bold uppercase characters to denote matrices (e.g., $\mathbf{A}$), italic uppercase characters to denote sets (e.g., V), and italic lowercase characters and overline arrow to denote column vectors (e.g., $\vec{\theta}$). For a matrix $\mathbf{Q}$, q_{ij} denotes its $(i,j)^{\text{th}}$ entry, and $\mathbf{Q}^t$ denotes its transpose. For a column vector $\vec{y}$, y_i denotes its i^{th} entry, $\|\vec{y}\|_1 := \sum_{i=1}^{n} |y_i|$ is its ℓ_1-norm, $\|\vec{y}\|_2 := (\sum_{i=1}^{n} y_i^2)^{1/2}$ is its ℓ_2-norm (also known as Euclidean norm), and $\text{supp}(\vec{y}) := \{i | y_i \neq 0\}$ is its support which indicates the nonzero entries of vector $\vec{y}$.

For a set V, $|V|$ denotes it size; For a real number r, $|r|$ denotes is absolute value; And for a complex number c, $|c|$ denotes its magnitude.

2.2 The AC Power Flows

Today's power grids deploy the Alternating Current (AC) based systems for power generation and transmission. Electric power in these systems flows from generators to loads based on a set of differential equations which depend on the structure of the network as well as the physical properties of the lines. In the steady state (i.e., fixed frequency), *phasors* are commonly used to compute the AC power flows using a set of nonlinear equations [76].

In the phasor domain, the status of each node i is represented by its voltage $\mathrm{v}_i = |\mathrm{v}_i|e^{\mathbf{i}\theta_i}$ in which $|\mathrm{v}_i|$ is the voltage magnitude, θ_i is the phase angle at node i, and $\mathbf{i}$ denotes the imaginary unit. The characteristics of each line $\{i, j\}$ is determined by its *admittance value* $g_{ij} + \mathbf{i}b_{ij} = \frac{1}{r_{ij}+\mathbf{i}x_{ij}}$, where g_{ij}, b_{ij}, r_{ij}, and x_{ij} denote line *conductance, susceptance, resistance,* and *reactance*, respectively. The complex power flow s_{ij} on the line $\{i, j\}$ from node i to j is equal to:

$$s_{ij} = \mathrm{v}_i(\mathrm{v}_i^* - \mathrm{v}_j^*)(g_{ij} + \mathbf{i}b_{ij})^*, \tag{2.1}$$

where * denotes the complex conjugation. The complex power $s_{ij} = p_{ij} + \mathbf{i}q_{ij}$ from node i to node j consists of the active power flow $p_{ij} = \mathrm{Re}\{s_{ij}\}$ and reactive power flow $q_{ij} = \mathrm{Im}\{s_{ij}\}$.

Using 2.1, the injected complex power s_i at node i is:

$$s_i = \sum_{k\in N(i)} s_{ik} = \sum_{k\in N(i)} \mathrm{v}_i(\mathrm{v}_i^* - \mathrm{v}_k^*)(g_{ik} + \mathbf{i}b_{ik})^*, \tag{2.2}$$

where $N(i)$ is the set of nodes that are directly connected with a line to the node i.

Using 2.2, the equations for the active power p_i and the reactive power q_i at each node i can be written as:

$$p_i = -\textstyle\sum_{k\in N(i)\cup\{i\}} |\mathrm{v}_i||\mathrm{v}_k|(g_{ik}\cos\theta_{ik} + b_{ik}\sin\theta_{ik}) \tag{2.3}$$

$$q_i = -\textstyle\sum_{k\in N(i)\cup\{i\}} |\mathrm{v}_i||\mathrm{v}_k|(g_{ik}\sin\theta_{ik} - b_{ik}\cos\theta_{ik}) \tag{2.4}$$

where $\theta_{ik} = \theta_i - \theta_k$, and $g_{ii} + \mathbf{i}b_{ii} = -\sum_{k\in N(i)}(g_{ik} + \mathbf{i}b_{ik})$.

In the AC power flow problem, each node i is categorized into one of the following three types:

1. *Slack node*: The node for which the voltage is typically 1.0. For convenience, it is indexed as node 1. The active power p_1 and the reactive power q_1 need to be computed.

2. *Load node*: The active power p_i and the reactive power q_i at these nodes are known and the voltage v_i needs to be computed.

3. *Voltage controlled node*: The active power p_i and the voltage magnitude $|\mathrm{v}_i|$ at these nodes are known and the reactive power q_i and the phase angle θ_i need to be computed.

The problem is to compute p, q, v at all the nodes.

2.3 The DC Power Flows

Under operating conditions, the AC power flow equations can be reduced to a set of linear equations known as the *DC power flow* equations [76]. The DC power flow equations provide a linearized approximation of the active power flows in the AC model. Due to their linearity, the DC power flow equations are widely used for optimal power flow computation and contingency analysis in power grids [39,116,147]. Linearization is possible under the following conditions [76]:

1. The difference between the voltage phase angles of every couple of neighboring nodes is small such that $\sin\theta_{ik} \approx \theta_{ik}$ and $\cos\theta_{ik} \approx 1$.

2. The active power losses are negligible, and therefore the admittance value of each line can be calculated neglecting the line resistances by $-\mathbf{i}b_{ij} = 1/\mathbf{i}x_{ij}$.

3. The variations in the voltage magnitudes $|\mathrm{v}_i|$ are small and, therefore, it is assumed that $\forall i, |\mathrm{v}_i| = 1$.

Under these assumptions, given the active power vector $\vec{p} \in \mathbb{R}^{|V|\times 1}$ and the reactance values, the *DC power flow* is a solution $\mathbf{P} \in \mathbb{R}^{|V|\times|V|}$ and $\vec{\theta} \in \mathbb{R}^{|V|\times 1}$ of:

$$\sum_{j\in N(i)} p_{ij} = p_i, \ \forall \ i \in V \tag{2.5}$$

$$p_{ij} = -b_{ij}(\theta_i - \theta_j), \ \forall \ \{i,j\} \in E \tag{2.6}$$

where p_{ij} is the active power flow from node i to node j, and θ_i is the phase angle of node i. Eq. (2.5) guarantees (classical) flow conservation and (2.6) captures the dependency of the flow on the reactance values and phase angles. Additionally, (2.6) implies that $p_{ij} = -p_{ji}$. When the total supply equals the total demand in G (i.e., $\sum_{i=1}^{n} p_i = 0$), (2.5)-(2.6) has a unique solution.[1] Eq.(2.5)-(2.6) are equivalent to the following matrix equation:

$$\mathbf{A}\vec{\theta} = \vec{p} \tag{2.7}$$

where $\mathbf{A} \in \mathbb{R}^{|V|\times|V|}$ is the *admittance matrix* of G,[2] defined as follows:

$$a_{ij} = \begin{cases} 0 & \text{if } i \neq j \text{ and } \{i,j\} \notin E, \\ b_{ij} & \text{if } i \neq j \text{ and } \{i,j\} \in E, \\ -\sum_{k\in N(i)} b_{ik} & \text{if } i = j. \end{cases}$$

Since $\mathbf{A}$ is not a full-rank matrix, we use the *Moore-Penrose Pseudo-inverse* of $\mathbf{A}$ [14], denoted by $\mathbf{A}^{+}$[3] to solve (2.7). One can show that if (2.7) has a feasible solution, $\vec{\theta} = \mathbf{A}^{+}\vec{p}$ is a solution for (2.7). Once $\vec{\theta}$ is computed, the power flows, p_{ij}, can be obtained from (2.6).

2.3.1 Incidence Matrix

Under an arbitrary direction assignment to the edges of G, the *incidence matrix* of G is denoted by $\mathbf{D} \in \{-1, 0, 1\}^{n\times m}$ and defined as,

$$d_{ij} = \begin{cases} 0 & \text{if } e_j \text{ is not incident to node } i, \\ 1 & \text{if } e_j \text{ is coming out of node } i, \\ -1 & \text{if } e_j \text{ is going into node } i, \end{cases}$$

and $\mathbf{Y} := \text{diag}([1/x_{e_1}, 1/x_{e_2}, \ldots, 1/x_{e_m}])$ is a diagonal matrix with diagonal entries equal to the inverse of the reactance values. It can be verified that $\mathbf{A} = \mathbf{DYD}^t$.

[1] The uniqueness is in the values of p_{ij}s rather than θ_is (shifting all θ_is by equal amounts does not violate (2.6)).

[2] The admittance matrix $\mathbf{A}$ can also be considered as the *weighted Laplacian matrix* of the graph.

[3] $\mathbf{A}^{+} = \lim_{\delta\to 0} \mathbf{A}^t(\mathbf{AA}^t + \delta^2\mathbf{I})^{-1}$ [14].

The incidence matrix is a very useful matrix in analyzing failures in power grids. We use this matrix in Chapters 3 and 4 to quantify the effects of line failures, and in Chapters 5 and 6 to detect line failures after a cyber-phyical attack on the grid.

The incidence matrix can also be used to compute the power flows on the lines under the DC model. Given the vector of the phase angles, $\vec{f} = \mathbf{D}^t\vec{\theta}$ gives the power flows such that for any $e_k = \{i, j\}$, assuming that e_k is directed from i to j, $f_{e_k} = p_{ij} = -p_{ji}$. We use vector $\vec{f} \in \mathbb{R}^{|E|}$ to efficiently present power flows under the DC model more instead of matrix $\mathbf{P}$ in Chapters 3, 4, and 8.

2.3.2 Matrix of Equivalent Reactance Values

Define matrix $\mathbf{R} \in \mathbb{R}^{m\times m}$ as $\mathbf{R} := \mathbf{D}^t\mathbf{A}^+\mathbf{D}$. It can be shown that for any $\forall 1 \leqslant i \leqslant m : r_{ii}$ is *equivalent reactance* between end buses of the line e_i. Matrix $\mathbf{R}$ is a symmetric matrix and is very useful in quantifying the effect of line failures. In fact, in Chapter 3, we use this matrix to quantify the effect of single line failure when all the reactance values are equal to 1. In Chapter 4, we generalize the idea for k-line failures and arbitrary reactance values.

2.4 The DC Power Flows Accuracy

As mentioned in Section 2.3, the DC power flows provide a good approximation of the AC power flows under three conditions. In the following lemma, we provide an upper bound on the difference between the AC power flows $p_{ij}^{(\mathrm{AC})}$ and the DC power flows $p_{ij}^{(\mathrm{DC})}$ under those conditions. In this section, $\theta_i^{(\mathrm{DC})}$ denotes the phase angle of the nodes under the DC power flows.

Lemma 2.1. *Assume the three conditions for validity of the DC power flows as a linear approximation for the AC power flows hold within following bounds:*

1. $|\theta_i - \theta_j| \leqslant \epsilon_\theta, \forall\{i, j\} \in E$,
2. $|g_{ij}/b_{ij}| \leqslant \epsilon_g, \forall\{i, j\} \in E$,
3. $||\mathrm{v}_i| - 1| \leqslant \epsilon_v, \forall i \in V$,

for $\epsilon_\theta, \epsilon_v, \epsilon_g < 1$. Then, for any $1 \leqslant i, j \leqslant n$:

$$|p_{ij}^{(\mathrm{AC})} - p_{ij}^{(\mathrm{DC})}| \leqslant \epsilon \|\mathbf{A}\|_1 + \|\bar{p}^{(\mathrm{AC})} - \bar{p}^{(\mathrm{DC})}\|_1, \tag{2.8}$$

in which $\epsilon := 2\epsilon_g + 4\epsilon_v\epsilon_g + 2\epsilon_v^2\epsilon_g + 2\epsilon_v\epsilon_\theta + \epsilon_v^2\epsilon_\theta + \epsilon_\theta^3$ and $\|\mathbf{A}\|_1 := \sum_{i=1}^n \sum_{j=1}^n |a_{ij}|$.

Proof. Using the definition of the apparent power $s_{ik} = p_{ik}^{(\mathrm{AC})} + \mathbf{i} q_{ik}^{(\mathrm{AC})}$ in (2.2), we have:

$$\begin{aligned} p_{ij}^{(\mathrm{AC})} &= \mathrm{Re}\big\{ \mathrm{v}_i \big((\mathrm{v}_i - \mathrm{v}_j)(g_{ij} + \mathbf{i} b_{ij}) \big)^* \big\} \\ &= \mathrm{Re}\big\{ (|\mathrm{v}_i|^2 - |\mathrm{v}_i||\mathrm{v}_j| e^{\mathbf{i}(\theta_i - \theta_j)})(g_{ij} - \mathbf{i} b_{ij}) \big\} \\ &= |\mathrm{v}_i|^2 g_{ij} - |\mathrm{v}_i||\mathrm{v}_j| g_{ij} \cos(\theta_i - \theta_j) \\ &\quad - |\mathrm{v}_i||\mathrm{v}_j| b_{ij} \sin(\theta_i - \theta_j). \end{aligned}$$

Define $\theta_i - \theta_j := \alpha_{ij}$ and $|\mathrm{v}_i| := 1 + \beta_i$. Then:

$$\begin{aligned} ||\mathrm{v}_i|^2 g_{ij}| &= |g_{ij} + (2\beta_i + \beta_i^2) g_{ij}| \\ &\leqslant |b_{ij}| (\epsilon_g + 2\epsilon_v\epsilon_g + \epsilon_v^2 \epsilon_g). \end{aligned}$$

Moreover, using $|\cos(x)| \leqslant 1$:

$$\begin{aligned} ||\mathrm{v}_i||\mathrm{v}_j| g_{ij} \cos(\alpha_{ij})| &\leqslant |g_{ij} + (\beta_i + \beta_j + \beta_i\beta_j) g_{ij}| \\ &\leqslant |b_{ij}| (\epsilon_g + 2\epsilon_v\epsilon_g + \epsilon_v^2 \epsilon_g). \end{aligned}$$

Using $\sin(x) = x + F(x)$, for $F(x) := -x^3/6 + O(x^5)$, we also have:

$$\begin{aligned} &- |\mathrm{v}_i||\mathrm{v}_j| b_{ij} \sin(\theta_i - \theta_j) = \\ &= -b_{ij}(\theta_i - \theta_j) \\ &\quad - b_{ij} \big((\beta_i + \beta_j + \beta_i\beta_j)(\theta_i - \theta_j) + F((\theta_i - \theta_j)^3) |\mathrm{v}_i||\mathrm{v}_j| \big), \end{aligned}$$

in which:

$$\begin{aligned} &| - b_{ij} \big((\beta_i + \beta_j + \beta_i\beta_j)(\theta_i - \theta_j) + F((\theta_i - \theta_j)^3) |\mathrm{v}_i||\mathrm{v}_j| \big) | \leqslant \\ &\leqslant |b_{ij}| \big(2\epsilon_v\epsilon_\theta + \epsilon_v^2\epsilon_\theta + \epsilon_\theta^3/6 + \epsilon_\theta^3\epsilon_v/3 + \epsilon_\theta^3\epsilon_v^2/6 \big) \\ &\leqslant |b_{ij}| \big(2\epsilon_v\epsilon_\theta + \epsilon_v^2\epsilon_\theta + \epsilon_\theta^3 \big). \end{aligned}$$

Hence,

$$p_{ij}^{(\mathrm{AC})} = -b_{ij}(\theta_i - \theta_j) + e_{ij}, \tag{2.9}$$

in which:

$$|e_{ij}| \leqslant |b_{ij}|\big(2\epsilon_g + 4\epsilon_v\epsilon_g + 2\epsilon_v^2\epsilon_g + 2\epsilon_v\epsilon_\theta + \epsilon_v^2\epsilon_\theta + \epsilon_\theta^3\big).$$

Notice that $-b_{ij}(\theta_i - \theta_j)$ is not necessarily equal to $p_{ij}^{(\mathrm{DC})} = -b_{ij}(\theta_i^{(\mathrm{DC})} - \theta_j^{(\mathrm{DC})})$ since θ_i and θ_j are obtained from the AC power flows, which are different from the phase angles obtained by the DC power flows. However, we can compute the difference between these two values by writing the power flow equations using (2.9) as follows:

$$\begin{aligned}
&\text{AC: } \mathbf{A}\vec{\theta} + \vec{e} = \vec{p}^{(\mathrm{AC})} \\
&\qquad \mathbf{A}\vec{\theta} = \vec{p}^{(\mathrm{DC})} - \vec{e} + (\vec{p}^{(\mathrm{AC})} - \vec{p}^{(\mathrm{DC})}) \\
&\text{DC: } \mathbf{A}\vec{\theta}^{(\mathrm{DC})} = \vec{p}^{(\mathrm{DC})},
\end{aligned}$$

in which $\vec{e}$ is an $n \times 1$ vector with the i^{th} entry equal to e_i such that $|e_i| \leqslant \sum_{j\in N(i)} |e_{ij}|$. Recall that vectors $\vec{p}^{(\mathrm{AC})}$ and $\vec{p}^{(\mathrm{DC})}$ are equal except (depending on the lossless assumption) in the *slack bus* (first entry). From the flow equations and the superpositions principle, the difference between $-b_{ij}(\theta_i - \theta_j)$ and $-b_{ij}(\theta_i^{(\mathrm{DC})} - \theta_j^{(\mathrm{DC})})$ cannot be greater than the maximum flow that vector $\vec{e} - (\vec{p}^{(\mathrm{AC})} - \vec{p}^{(\mathrm{DC})})$ can produce which is at most $\|\vec{p}^{(\mathrm{AC})} - \vec{p}^{(\mathrm{DC})}\|_1 + (\sum_{i=1}^{n}\sum_{j\in N(i)} |b_{ij}|)(2\epsilon_g + 4\epsilon_v\epsilon_g + 2\epsilon_v^2\epsilon_g + 2\epsilon_v\epsilon_\theta + \epsilon_v^2\epsilon_\theta + \epsilon_\theta^3)$. Hence, a conservative bound for the difference between the AC and DC power flows is:

$$|p_{ij}^{(\mathrm{AC})} - p_{ij}^{(\mathrm{DC})}| \leqslant \epsilon\|\mathbf{A}\|_1 + \|\vec{p}^{(\mathrm{AC})} - \vec{p}^{(\mathrm{DC})}\|_1.$$

■

Corollary 2.1. *If the lines are lossless ($\epsilon_g = 0$), then for any $1 \leqslant i, j \leqslant n$:*

$$|p_{ij}^{(\mathrm{AC})} - p_{ij}^{(\mathrm{DC})}| \leqslant (2\epsilon_v\epsilon_\theta + \epsilon_v^2\epsilon_\theta + \epsilon_\theta^3)\|\mathbf{A}\|_1.$$

Proof. Since lines are lossless, $\|\vec{p}^{(\mathrm{AC})} - \vec{p}^{(\mathrm{DC})}\|_1 = 0$. Using this and $\epsilon_g = 0$ in (2.8) gives the bound. ■

Corollary 2.2. *If the lines are lossless ($\epsilon_g = 0$), and for all the lines $b_{ij} = -1$, then for any $1 \leqslant i, j \leqslant n$:*

$$|p_{ij}^{(\mathrm{AC})} - p_{ij}^{(\mathrm{DC})}| \leqslant 4(2\epsilon_v\epsilon_\theta + \epsilon_v^2\epsilon_\theta + \epsilon_\theta^3)|E|.$$

Proof. In this case, since for all the lines $b_{ij} = -1$, $\|\mathbf{A}\|_1 = 4|E|$. Therefore, using Corollary 2.1 gives the bound. ■

In the following Lemma, we demonstrate that if the power network topology is a tree, we can improve the bound in Corollary 2.1 and prove that the DC power flows are equal to the AC power flows.

Lemma 2.2. *If G is a tree and $\epsilon_g = 0$, then $\forall 1 \leqslant i, j \leqslant n : p_{ij}^{(\mathrm{AC})} = p_{ij}^{(\mathrm{DC})}$*

Proof. We want to show that $p_{ij}^{(\mathrm{DC})} = p_{ij}^{(\mathrm{AC})}, \forall \{i, j\} \in E$ is a DC power flow solution for this instance. As all the lines are purely reactive, the network is lossless and in the AC power flows $|p_{ij}^{(\mathrm{AC})}| = |p_{ji}^{(\mathrm{AC})}|, \forall \{i, j\} \in E$. From Kirchhoff's laws, we also have $\sum_{j \in N(i)} p_{ij}^{(\mathrm{AC})} = p_i$. Hence, setting $p_{ij}^{(\mathrm{DC})} = p_{ij}^{(\mathrm{AC})}$ completely satisfies the active power flow conservation in equation (2.5). It remains to prove that there are phase angles satisfying equation (2.6) with $p_{ij}^{(\mathrm{DC})} = p_{ij}^{(\mathrm{AC})}$. Since G is a tree, $m = n - 1$. Hence, $p_{ij}^{(\mathrm{DC})} = -b_{ij}(\theta_i^{(\mathrm{DC})} - \theta_j^{(\mathrm{DC})})$ for all $\{i, j\} \in E$ consists of $n - 1$ independent linear equations for n variables $\theta_1^{(\mathrm{DC})}, \theta_2^{(\mathrm{DC})}, \ldots, \theta_n^{(\mathrm{DC})}$ (It is known that for a connected graph G, $\mathrm{rank}(\mathbf{A}) = n - 1$ [32]). As a result, by selecting $\theta_1^{(\mathrm{DC})} = 0$, all other phase angles can be found uniquely. Hence, $p_{ij}^{(\mathrm{DC})} = p_{ij}^{(\mathrm{AC})}$ is the DC power flow solution for this instance as well.

■

In this section, we demonstrated that the DC power flows can approximate the AC power flows very well depending on the $\epsilon_\theta, \epsilon_g$, and ϵ_v values. In this thesis, due to the nonlinearity of the AC power flows, we mainly use the DC power flows. Extending the methods provided in this thesis to the AC power flow model should be of particular interest to researchers in power engineering community.

Chapter 3

Analyzing Failures and Cascades

Large scale power outages in Australia (2016), Turkey (2015), India (2013), and the U.S. (2011, 2003) have demonstrated the vulnerability of power grids to failures caused by natural disasters, such as earthquakes, hurricanes, and solar flares as well as mismanagement and computers malfunctioning. These power grid outages have devastating effects on almost every aspect of modern life as well as on interdependent systems such as telecommunications, gas, water supply, and transportation networks. Follow up reports of these events [7,8,9,12], exposed the insufficient understanding of failures and their consequences in power grids. Hence, there is a need for investigation of failures in power grids.

As we discussed in Chapter 2, unlike graph-theoretical network flows, power flows are governed by the laws of physics and there are no strict capacity bounds on the lines [25]. Yet, there is a *rating threshold* associated with each line – if the flow exceeds the threshold, the line will eventually experience thermal failure. Such an outage alters the network topology, giving rise to a different flow pattern which, in turn, could cause other line outages. The repetition of this process constitutes a *cascading failure* [44].

Some previous analytical works (e.g., [42,156] and references therein) assumed that a line/node failure leads, with some probability, to failures of nearby nodes/lines. Such epidemic based modeling allows using percolation-based tools to analyze the cascade's impact. Yet, in real large scale cascades, a failure of a specific line can affect remote lines and the cascade does not neces-

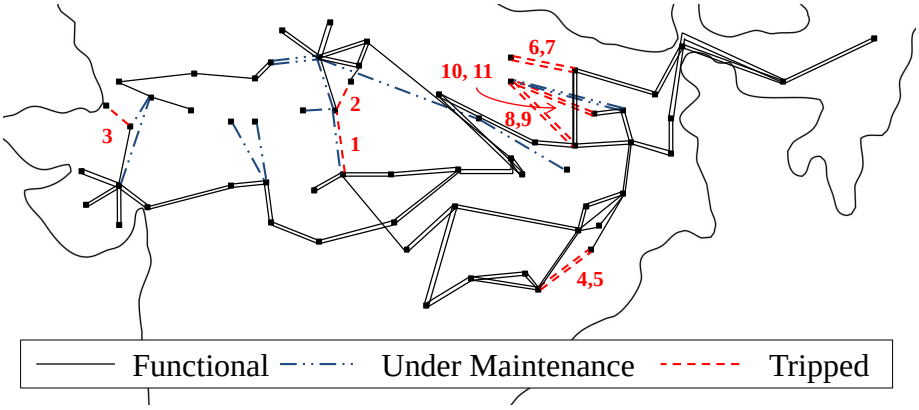


Figure 3.1: The first 11 line outages leading to the India blackout on July 30^{th}, 2012 [8]. The numbers indicate the order in which outages occurred; Lines that were undergoing maintenance at the time of the cascade are marked separately; Note that the failures did not develop contiguously.

sarily develop contiguously. For example, the evolution of the 2012 cascade in India appears in Fig. 3.1. Similar non-contiguous evolution was observed in a 2011 cascade in South California [9] and in simulation studies under more realistic models [27].

Motivated by these observations, we analytically study the properties of failures, detect most vulnerable lines to failures, and introduce algorithms to efficiently predict the cascading failure evolution. We employ the DC power flow model which is commonly used in large-scale contingency analysis of power grids [28, 30, 113], and the cascading failure model of [62] (see also [27, 28, 30, 111]).

First, in order to investigate the impact of a single line failure on the power flow changes, we provide a rank-1 update of the pseudo-inverse of the admittance matrix after a single line failure. We then use that to analytically compute the flows on the lines after a *single line failure.* and obtain an upper bound on the flow changes. We build on these results and develop metrics to study the robustness of graphs to failures.

We also illustrate via simulations the relation between the flow changes after a failure and the distance (in hop count) and *resistance distance* from the failure. In our simulations, we consider the U.S. Western Interconnection and the Texas Interconnection as well as Erdős-Rényi, Watts

and Strogatz [153], and Barábasi and Albert [23] graphs. These simulations show that there are cases in which the flow on a line far away from the initial failure significantly increases. These observations are clearly in contrast to the epidemic-based failure models.

Once a line fails, there is a need for a low complexity algorithm to predict its consequences so that the system operator can react quickly if the initial failure results in a cascading failures. The Cascading Failure Evolution (CFE) Algorithm that has been used to identify the evolution of the cascade [28,30,44] has the time complexity of $O(t|V|^3)$, where $|V|$ is the number of nodes and t is the number of cascade rounds). In order to improve its running time, we develop the low complexity *Cascading Failure Evolution – Pseudo-inverse Based (CFE-PB) Algorithm.*

The algorithm is based on the rank-1 update of the pseudo-inverse of the admittance matrix. We show that its complexity is $O(|V|^3+|F_t^*||V|^2)$ ($|F_t^*|$ is the total number of lines failures during the cascade). Namely, if $t = |F_t^*|$ (one line fails at each round), the complexity of the CFE-PB Algorithm is $O(\min\{|V|, t\})$ times lower than that of the CFE Algorithm. The main advantage of the CFE-PB Algorithm is that it leverages the special structure of the pseudo-inverse to identify properties of the underlying graph and to recompute an instance of the pseudo-inverse from a previous instance.

Finally, we prove that the problem of finding the set of line failures of size at most k that causes a cascade resulting with the minimum possible yield (the fraction of demand satisfied after the cascade), referred to as the *minimum yield problem*, is NP-hard. However, we introduce a very fast heuristic termed the Most Vulnerable Edges Selection – Resistance distance Based (MVES-RB) Algorithm for finding such a set, approximately. We compare the performance of the MVES-RB Algorithm and other heuristic methods on the IEEE 118- and 300-bus benchmark systems [5] and show that it performs relatively well considering its low time complexity.

3.1 Related Work

The study of cascading failures in power grids was initiated in [39,147] which used the linearized DC model and a probabilistic outage rule for overloaded line failures. Similar cascade models have been used to study the properties of the cascades [18,19,26,27,44,91], as well as to design

control schemes to mitigate the cascade [28,110] and to detect vulnerable parts of the grid [29,83]. In this Chapter, we follow [27, 29] and study cascades due to line overloads in power grids with a *deterministic outage rule*.

In Sections 3.3 and 3.5, we use the admittance matrix of the grid to compute flows. This is closely related to *solving Laplacian systems* which can be solved by several techniques, including Gaussian elimination and LU factorization [77]. Recently, [49] used preconditioning, to provide highly precise approximate solutions to Laplacian systems. Yet, this approach is not suitable for analyzing the effects of line failures.

The problem of *identifying the set of failures with the largest impact* was studied in [28, 30, 113]. In particular, [30] studies the $N - k$ problem which focuses on finding a small set of lines whose removal disallows supporting a minimum demand. A broader interdiction problem where all the network components are subject to failure was studied in [120]. A similar problem is studied in [113] using the alternating-current (AC) model. However, none of the previous works consider the cascading failures. Moreover, while the optimal power flow problem has been shown to be NP-hard [31, 93], the complexity of the cascade-related problems was not studied yet.

In simulations, we use graphs that can represent the power grid topology. The structure of power grids was widely studied [23, 42, 52, 153]. Specifically, Watts and Strogatz [153] suggested the small-world graph as a good representative of the power grid. Barabási and Albert [23] showed that scale-free graphs are better representatives. However, [52] indicated that none of these models can represent the U.S. Western Interconnection properly. Hence, we consider these graphs and the Erdős-Rényi graph but we also use the Western Interconnection, the Texas Interconnection, and the IEEE 118- and 300-bus benchmark systems [5] in order to consider realistic systems.

3.2 Model and Preliminaries

In this chapter, we use the DC power flows as described in Section 2.3. We refer to the failed lines and line failures as *failed edges* and *edge failures* in the graph G. Moreover, in this chapter, $\mathbf{A}_i$ denotes the i^{th} row of matrix $\mathbf{A}$.

Algorithm 1 - Cascading Failure Evolution (CFE)

Input: A connected graph $G = (V, E)$ and an initial edge failures event $F_0 \subseteq E$.

1: $F_0^* \leftarrow F_0$ and $i \leftarrow 0$.
2: **while** $F_i \neq \varnothing$ **do**
3: Adjust the total demand (supply) to equal the total supply (demand) within each connected component of $G = (V, E \backslash F_i^*)$.
4: Compute the new flows $f_e(F_i^*) \quad \forall e \in E \backslash F_i^*$.
5: Find the set of new edge failures $F_{i+1} = \{e || f_e(F_i^*)| > c_e,\ e \in E \backslash F_i^*\}$. $F_{i+1}^* \leftarrow F_i^* \cup F_{i+1}$ and $i \leftarrow i + 1$.
6: **return** $t = i - 1$, $(F_0, \ldots, F_t)$, and $f_e(F_t^*)\ \forall e \in E \backslash F_t^*$.

3.2.1 Cascading Failure Model

The Cascading Failure Evolution (CFE) Algorithm described here is a slightly simplified version of the cascade model used in [27,30,62]. For each edge $e = \{i, j\}$, consider an arbitrary direction as in Subsection 2.3.1 and define $f_e := p_{ij}$, if direction of the edge goes from i to j. We assume that each edge has a predetermined capacity $c_e = c_{ij} = c_{ij}$, which bounds the amount of power that it can carry without any risks of failure (that is, $|f_e| \leqslant c_e$). The cascade proceeds in rounds. Denote by $F_i \subseteq E$ the set of edge failures in the i^{th} round and by $F_i^* = F_{i-1}^* \cup F_i$ the set of edge failures until the end of the i^{th} round ($i \geqslant 1$). We assume that before the initial failure event $F_0 \subseteq E$, the power flows satisfy (2.5)-(2.6), and $|f_e| \leqslant c_e\ \forall e \in E$.

Upon a failure, some edges are removed from the graph, implying that it may become disconnected. Thus, within each component, depending on whether demand>supply or supply>demand, the total demand is adjusted to be equal to the total supply by decreasing the demand (supply) by the same factor at all demand (supply) nodes (Line 3). This corresponds to the load shedding/generation curtailing process. Since power grids are operated by a central system, we follow [27, 28, 30] and assume that the demand (supply) are adjusted globally. For any set of failures $F \subseteq E$, we denote by $f_e(F)$ the flow on the edges in $G' = (V, E \backslash F)$ after the shedding/curtailing.

Following an initial failure event F_0, the new flows $f_e(F_0), \forall e \in E \backslash F_0$ are computed (by (2.5)-(2.6)) (Line 4). Then, the set of new edge failures F_1 is identified (Line 5). Following [27,30,62], we use a deterministic outage rule and assume, for simplicity, that an edge e fails once the flow exceeds its capacity: $|f_e(F_0^*)| > c_e$.[1] Therefore, $F_1 = \{e : |f_e(F_0^*)| > c_e, e \in E \backslash F_0^*\}$.

If the set F_1 of new edge failures is empty, the cascade is terminated. Otherwise, the process is repeated while replacing the initial event $F_0^* = F_0$ by the failure event F_1^*, and generally replacing F_i^* by F_{i+1}^* at the i^{th} round (Line 5). The process continues until the system *stabilizes* (i.e., until no edges are removed). Finally, we obtain the sequence $(F_0, F_1, \ldots, F_t)$ of the sets of failures associated with the initial event F_0, and the power flows $f_e(F_t^*)$ at stabilization, where t is the number of rounds until stabilization. Since solving a system of linear equations with n variables, requires $O(n^3)$ time [77], the output can be obtained in $O(t|V|^3)$ time.

When the initial failure event contains a single edge $F_0 = \{e'\}$, for any edges $e = \{i, j\} \in E \backslash \{e'\}$, we denote the flows after the failure by p'_{ij} or $f'_e \equiv f_e(\{e'\})$ and the flow changes by $\Delta p_{ij} = p'_{ij} - p_{ij}$. Moreover $\Delta f_e = \Delta p_{ij}$.

3.2.2 Metrics

We define metrics for evaluating the grid vulnerability (some of which were defined in [27], [155]). To study the effects of a *single edge (e') failure after one round*, we define the ratio between the flow change on an edge e, and its original value or the flow value on the failed edge e':

Edge flow change ratio: $S_{e,e'} := |\Delta f_e / f_e|$.

Mutual edge flow change ratio: $M_{e,e'} := |\Delta f_e / f_{e'}|$.

The mutual edge flow change ratio corresponds to the Line Outage Distribution Factor (LODF) defined in [155] (we used a different term to be consistent with the other metric).

We also define a metric to evaluate the *cascade severity*.

Yield: the ratio between the demand supplied at stabilization and the original demand after an initial failure event.

[1] Note that [27,30,62] maintain moving averages of the f_e values to determine which edges fail.

3.2.3 Graphs and Parameters Used in Simulations

For simulations, we used the *NetworkX* [81] library in Python, as well as *igraph* [54] library in R. The simulation results are presented for graphs described below. The parameters are as indicted below, unless otherwise mentioned.

Western Interconnection (WI): The U.S. Western Interconnection with 13626 nodes and 18089 edges. The data is from the Platts Geographic Information System (GIS) [3]. In some cases a 1374-node connected subgraph of the WI is considered.

Texas Interconnection (TI): The Texas Interconnection with 4544 nodes and 6264 edges (obtained from the Platts GIS [3]).

IEEE benchmark Systems: The IEEE 118-bus and 300-bus benchmark systems with 179 and 409 edges, respectively [5].

Erdős-Rényi graph: A random graph where each edge appears with probability $p = 0.01$.

Watts and Strogatz graph [153]: A small-world random graph where each node connects to $k = 4$ other nodes and the probability of rewiring is $p = 0.1$.

Barábasi and Albert graph [23]: A scale-free random graph where each new node connects to $k = 3$ other nodes at each step following the preferential attachment mechanism.

To maintain consistency when evaluating a metric in generated random networks and in a real-world network (e.g., WI), in this chapter, we assume that the reactance values (or susceptance values) are equal to 1 for all edges ($-1/x_{ij} = b_{ij} = -1 \ \forall\{i,j\} \in E$). However, to provide representative results about real-world networks, we also perform the same evaluations with *estimated reactance* values. The reactance of a line depends on its physical properties (such as its material) and there is a linear relation between its length and reactance: the longer the line, the larger its reactance. Thus, we assume that all lines have the same physical properties and use the length to determine the reactance. It is important to note that flows are scale invariant to the reactance (that is, multiplying the reactance of all lines by the same factor does not change the flow values). Hence, we simply use the length of a line as its estimated reactance.

3.3 Admittance Matrix Properties

Recall that the power flow equations can be solved by using the Pseudo-inverse of the admittance matrix $\mathbf{A}^+$. In this section, we use the Pseudo-inverse of the admittance matrix in order to obtain results that are used throughout the rest of the Chapter. Specifically, it is used in Section 3.4 to study the impact of a single edge failure on the flows on other edges and in Section 3.5 to introduce an efficient algorithm to identify the evolution of the cascade. We prove several properties of the Pseudo-inverse of the admittance matrix $\mathbf{A}^+$.

A *cut-edge* of a graph G is an edge whose removal increases the number of connected components of G. Jointly verifying whether an edge is a cut-edge and finding the connected components of the graph takes $O(|E|)$ (using Depth First Search). The following two Lemmas show that by using the precomputed pseudo-inverse of the admittance matrix, these operations can be done in $O(1)$ and $O(|V|)$, respectively. The algorithm in Section 3.5 uses these results to check if the pseudo-inverse should be recomputed. Moreover, Lemma 3.1 is crucial for the proof of the Theorem 3.1, below. We note that a similar lemma when all the admittance values are equal to 1 appeared in [22].

Lemma 3.1. *Given $G = (V, E)$ and $\boldsymbol{A}^+$, all the cut-edges of the graph G can be found in $O(|E|)$ time. Specifically, an edge $\{i, j\}$ is a cut-edge if, and only if, $a_{ij}^{-1} - 2a_{ij}^+ + a_{ii}^+ + a_{jj}^+ = 0$.*

Proof. Suppose $\{i, j\}$ is a cut-edge. Then, the solution to (2.5)-(2.6) for the power vector $\vec{p}$, in which $p_i = -p_j = 1$ and zero elsewhere is $p_{ij} = -p_{ji} = 1$ and zero elsewhere. On the other hand, $\vec{\theta} = \mathbf{A}^+\vec{p}$ is a solution to the equivalent matrix equation (2.7). Since the solution with respect to the power flows is unique, $1 = p_{ij} = -a_{ij}(\theta_i - \theta_j) = -a_{ij}(\mathbf{A}_i^+\vec{p} - \mathbf{A}_j^+\vec{p}) \Rightarrow 1 = -a_{ij}(a_{ii}^+ - a_{ij}^+ - a_{ji}^+ + a_{jj}^+) \Rightarrow a_{ij}^{-1} = (2a_{ij}^+ - a_{ii}^+ - a_{jj}^+) \Rightarrow a_{ij}^{-1} - 2a_{ij}^+ + a_{ii}^+ + a_{jj}^+ = 0$.

Now suppose $a_{ij}^{-1} - 2a_{ij}^+ + a_{ii}^+ + a_{jj}^+ = 0$. The solution to (2.5)-(2.6) for the power vector $\vec{p}$ (as defined above) for the flow on edge $\{i, j\}$ is $p_{ij} = p_i = -p_j = 1$. Therefore, there are no other paths from i to j. Otherwise, because of the phase angle difference between nodes i and j, part of the flow should be routed through other paths. Thus, edge $\{i, j\}$ is the only path from

node i to j, and is a cut-edge. Hence, using the precomputed $\mathbf{A}^+$, identifying whether an edge is a cut-edge takes $O(1)$ time and finding all the cut-edges of the graph takes $O(|E|)$ time. ■

Lemma 3.2. *Given $G = (V, E)$, $\boldsymbol{A}^+$, and the cut-edge $\{i, j\}$, the connected components of $G\backslash\{i, j\}$ can be found in $O(|V|)$.*

Proof. Suppose that $\{i, j\}$ is a cut-edge of the connected graph G, and $G\backslash\{i, j\} = G_1 \cup G_2$. Assume that $i \in G_1$ and $j \in G_2$. We show below that for any $\{r, s\} \in G\backslash\{i, j\}$, $a^+_{ir} - a^+_{jr} = a^+_{is} - a^+_{js}$. Moreover, for any $r \in G_1$ and $s \in G_2$, $a^+_{ir} - a^+_{jr} \neq a^+_{is} - a^+_{js}$.

Suppose that $\{r, s\} \in G_1$ (the argument is the same for G_2) is an arbitrary edge. Then, the power flows solution to (2.5)-(2.6) in G for the power vector $\vec{p}$ with $p_r = -p_s = 1$ and zero elsewhere, is nonzero in G_1 and zero elsewhere. Therefore, $p_{ij} = 0$. On the other hand, $\vec{\theta} = \mathbf{A}^+\vec{p}$ is a solution to the equivalent matrix equation (2.7). Since the solution with respect to power flows is unique, $0 = f_{ij} = -a_{ij}(\theta_i - \theta_j) = -a_{ij}(\mathbf{A}^+_i\vec{p} - \mathbf{A}^+_j\vec{p}) \Rightarrow 0 = (a^+_{ir} - a^+_{is} - a^+_{jr} + a^+_{js}) \Rightarrow a^+_{ir} - a^+_{jr} = a^+_{is} - a^+_{js}$.

From this and since $a^+_{ii} - a^+_{ji} \neq a^+_{ij} - a^+_{jj}$ (Lemma 3.1), for any $r \in G_1$ and $s \in G_2$, $a^+_{ir} - a^+_{jr} \neq a^+_{is} - a^+_{js}$. Thus, by using the precomputed pseudo-inverse of the admittance matrix, computing $\mathbf{A}^+_i - \mathbf{A}^+_j$, and dividing the entries into two groups with equal values, the connected components of $G\backslash\{i, j\}$ can be identified. This process requires $O(|V|)$ time. ■

In the following, we denote by $\mathbf{A}'$ the admittance matrix of the graph $G' = (V, E\backslash\{i, j\})$ and by $\vec{p}'$ the power vector after removing an arbitrary edge $\{i, j\}$ from G and conducting the corresponding load shedding/generation curtailing. Lemma 3.3 shows that after the removal of a cut-edge, $\mathbf{A}^+$ can be used to solve (2.7) and $\mathbf{A}'^+$ is not required.

Lemma 3.3. *Given graph $G = (V, E)$, $\boldsymbol{A}^+$, and a cut-edge $\{i, j\}$, then $\vec{\theta}' = \boldsymbol{A}^+\vec{p}'$ is a solution of (2.7) in G'.*

Proof. First, $\vec{\theta}' = \mathbf{A}^+\vec{p}'$ is a solution to (2.7) for the power vector $\vec{p}'$ in the graph G. Since the solution to (2.5)-(2.6) with respect to power flows is unique, if $p'_{ij} = 0$, then $\vec{\theta}' = A^+\vec{p}'$ is also a solution to (2.7) for the power vector $\vec{p}'$ in the graph G'. Therefore, we only need to prove that $\theta'_i = \theta'_j$ in $\vec{\theta}' = \mathbf{A}^+\vec{p}'$.

To prove this, we prove that $\theta'_i - \theta'_j = (\mathbf{A}_i^+ - \mathbf{A}_j^+)\vec{p}' = 0$. From the proof of Lemma 3.2, since $\{i, j\}$ is a cut-edge, the entries of $\mathbf{A}_i^+ - \mathbf{A}_j^+$ have equal values at the entries in the same connected component. On the other hand, since $\vec{p}'$ is the power vector after load shedding, then the sum of the supplies and demands at each connected component is zero. Thus, $(\mathbf{A}_i^+ - \mathbf{A}_j^+)\vec{p}' = 0$. ■

The following theorem gives an analytical rank-1 update of the pseudo-inverse of the admittance matrix. Using Theorem 3.1 and Corollary 3.1, in Section 3.4 we provide upper bounds on the mutual edge flow change ratios ($M_{e,e'}$). We note that the proof could be simplified, if the form of $\mathbf{A'}^+$ was known in advance. However, the proof provides the derivation of $\mathbf{A'}^+$. We also note that a similar result to Theorem 3.1 was independently proved in a technical report [117].

Theorem 3.1. *Given a connected graph $G = (V, E)$, the admittance matrix $\boldsymbol{A}$, and $\boldsymbol{A}^+$, if $\{i, j\}$ is not a cut-edge, then,*

$$\boldsymbol{A'}^+ = (\boldsymbol{A} + a_{ij}\boldsymbol{X}\boldsymbol{X}^t)^+ = \boldsymbol{A}^+ - \frac{1}{a_{ij}^{-1} + \boldsymbol{X}^t\boldsymbol{A}^+\boldsymbol{X}}\boldsymbol{A}^+\boldsymbol{X}\boldsymbol{X}^t\boldsymbol{A}^+$$

in which $\boldsymbol{X}$ is an $n \times 1$ matrix with 1 in i^{th} entry, -1 in j^{th} entry, and 0 elsewhere.

Proof. First, we show that $\mathbf{A}\mathbf{A}^+ = \mathbf{I} - \frac{1}{n}\mathbf{J}$. $\mathbf{A}$ is a real and symmetric matrix. Therefore, there exists an orthogonal and unitary matrix $\mathbf{U}$ such that $\mathbf{A} = \mathbf{U}^t\Lambda\mathbf{U}$, in which $\Lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$ is the diagonal matrix of eigenvalues of $\mathbf{A}$ and $\mathbf{U}_i$ is the normalized eigenvector related to eigenvalue λ_i. It is well-known that when G is connected and unweighted, the multiplicity of eigenvalue 0 of the Laplacian matrix is 1 [32]. Exactly the same result with the same approach can be obtained for a weighted graph. Therefore, we can assume that $\lambda_1 = 0$ and all other eigenvalues are nonzero. In this case $\mathbf{U}_1 = [\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}, \ldots, \frac{1}{\sqrt{n}}]$. On the other hand, $\mathbf{A}^+ = \mathbf{U}^t\Lambda^+\mathbf{U}$, and therefore,

$$\begin{aligned}
\mathbf{A}\mathbf{A}^+ &= \mathbf{U}^t\Lambda\mathbf{U}\mathbf{U}^t\Lambda^+\mathbf{U} = \mathbf{U}^t\Lambda\Lambda^+\mathbf{U} \\
&= \mathbf{U}^t\text{diag}(\lambda_1\lambda_1^+, \lambda_2\lambda_2^+, \ldots, \lambda_n\lambda_n^+)\mathbf{U} \\
&= \mathbf{U}^t(\mathbf{I} - \text{diag}(1, 0, \ldots, 0))\mathbf{U} = \mathbf{I} - \frac{1}{n}\mathbf{J}.
\end{aligned}$$

$\mathbf{A}$ is a real and symmetric matrix, and therefore, there exist an $n \times n$ matrix $\mathbf{B}$ such that $\mathbf{B}\mathbf{B}^t = \mathbf{A}$. Using [14, Theorem 4],

$$(\mathbf{A} + a_{ij}\mathbf{X}\mathbf{X}^t)^+ = (\mathbf{C}\mathbf{C}^t)^+ + [\mathbf{I} - (\sqrt{a_{ij}}\mathbf{X}\mathbf{C}^+)^t]$$
$$\times [\mathbf{A}^+ - a_{ij}\mathbf{A}^+\mathbf{X}(\mathbf{I} - \mathbf{C}^+\mathbf{C})\mathbf{K}\mathbf{X}^t\mathbf{A}^+] \times [1 - \sqrt{a_{ij}}\mathbf{X}\mathbf{C}^+],^2$$

where, $\mathbf{C} = [\mathbf{I} - \mathbf{A}\mathbf{A}^+]\mathbf{X}$ and

$$\mathbf{K} = \{\mathbf{I} + a_{ij}[(\mathbf{I} - \mathbf{C}^+\mathbf{C})\mathbf{X}^t\mathbf{A}^+\mathbf{X}(\mathbf{I} - \mathbf{C}^+\mathbf{C})]\}^{-1}.$$

Therefore, all we need to compute is matrices $\mathbf{C}$ and $\mathbf{K}$. Using the previous part,

$$\mathbf{C} = [\mathbf{I} - \mathbf{A}\mathbf{A}^+]\mathbf{X} = [\mathbf{I} - \mathbf{I} + \frac{1}{n}\mathbf{J}]\mathbf{X} = \frac{1}{n}\mathbf{J}\mathbf{X}.$$

It is easy to see that $\mathbf{J}\mathbf{X} = 0$, and thus, $\mathbf{C} = 0$. Using this,

$$\begin{aligned}\mathbf{K} &= \{\mathbf{I} + a_{ij}[(\mathbf{I} - \mathbf{C}^+\mathbf{C})\mathbf{X}^t\mathbf{A}^+\mathbf{X}(\mathbf{I} - \mathbf{C}^+\mathbf{C})]\}^{-1} \\ &= \{\mathbf{I} + a_{ij}[\mathbf{I}\mathbf{X}^t\mathbf{A}^+\mathbf{X}\mathbf{I}]\}^{-1} = \{1 + a_{ij}\mathbf{X}^t\mathbf{A}^+\mathbf{X}\}^{-1}.\end{aligned}$$

Notice that $\mathbf{X}$ is an $n \times 1$ vector, and therefore, $\mathbf{X}^t\mathbf{A}^+\mathbf{X}$ is a scaler and $\mathbf{I}$ in the second equation is 1×1. Hence, it is written 1 instead of $\mathbf{I}$ in the last equation. Since $\{i, j\}$ is not a cut edge, from Lemma 3.1 we have, $1 + a_{ij}\mathbf{X}^t\mathbf{A}^+\mathbf{X} = a_{ij}[a_{ij}^{-1} - 2(a^+)_{ij} + (a^+)_{ii} + (a^+)_{jj}] \neq 0$, and therefore, $\mathbf{K}$ is well-defined. Replacing $\mathbf{K}$ and $\mathbf{C}$,

$$\begin{aligned}(\mathbf{A} + a_{ij}\mathbf{X}\mathbf{X}^t)^+ &= \mathbf{A}^+ - a_{ij}\mathbf{A}^+\mathbf{X}\{1 + a_{ij}\mathbf{X}^t\mathbf{A}^+\mathbf{X}\}^{-1}\mathbf{X}^t\mathbf{A}^+ \\ &= \mathbf{A}^+ - \frac{1}{a_{ij}^{-1} + \mathbf{X}^t\mathbf{A}^+\mathbf{X}}\mathbf{A}^+\mathbf{X}\mathbf{X}^t\mathbf{A}^+.\end{aligned}$$

■

In the following, we use Theorem 3.1 to derive an equation similar to the one provided in [155, Section 7.4.1] by using the superposition principle. Recall from Section 3.2 that $A^+ = [a_{rs}^+]$.

Corollary 3.1. *The flow on an edge $\{r, s\}$ after a failure in the non-cut-edge $\{i, j\}$ is,*

$$p'_{rs} = p_{rs} - \frac{a_{rs}}{a_{ij}}\frac{(a_{ri}^+ - a_{rj}^+) - (a_{si}^+ - a_{sj}^+)}{a_{ij}^{-1} - 2(a^+)_{ij} + (a^+)_{ii} + (a^+)_{jj}}p_{ij}.$$

[2] $\sqrt{a_{ij}}$ might be an imaginary number.

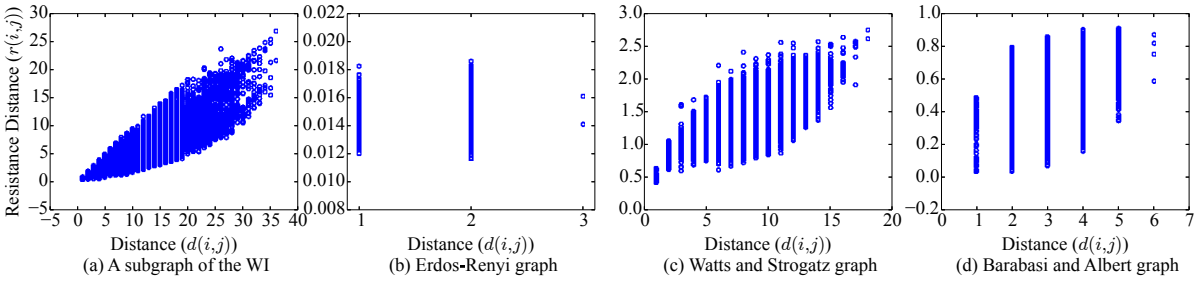


Figure 3.2: Scatter plot showing the distance versus the resistance distance between nodes in the graphs defined in Subsection 5.2.3. All graphs have 1374 nodes.

Finally, Lemma 3.4, gives the complexity of the rank-1 update provided in Theorem 3.1. This is used in the computation of the running time of the algorithm in Section 3.5.

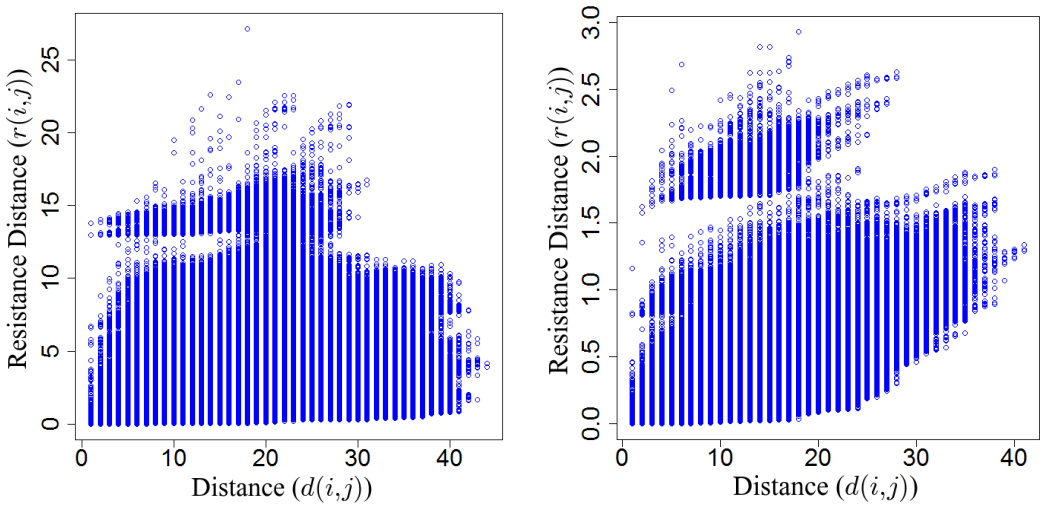
Lemma 3.4. *Given graph* $G = (V, E)$, $\boldsymbol{A}^+$, *and a non-cut-edge* $\{i, j\}$, $\boldsymbol{A}'^+$ *can be computed from* $\boldsymbol{A}^+$ *in* $O(|V|^2)$.

We now define the notion *resistance distance* [100]. In resistive circuits, the resistance distance between two nodes is the equivalent resistance between them. The resistance distance is a measure of distance between nodes of the graph [22]. For any network, this notion can be defined by using the pseudo-inverse of the Laplacian matrix of the network. Specifically, it can be defined in power grid networks by using the pseudo-inverse of the admittance matrix, $\mathbf{A}^+$.

Definition 3.1. *Given* $G = (V, E)$, $\boldsymbol{A}$, *and* $\boldsymbol{A}^+$, *the* resistance distance between two nodes $i, j \in V$ *is* $r(i, j) := a^+_{ii} + a^+_{jj} - 2a^+_{ij}$. *Accordingly, the resistance distance* between two edges $e = \{i, j\}, e' = \{v, u\}$ *is* $r(e, e') = \min\{r(i, v), r(i, u), r(j, v), r(j, u)\}$.

We use the resistance distance in Section 3.4 to obtain upper bounds on the flow changes after a single failure and assess the robustness of specific graphs against such failures. Moreover, using resistance distance, in Section 3.6, we provide a heuristic for the minimum yield problem.

When all edges have the same reactance, $x_{ij} = 1\ \forall\{i, j\} \in E$, the resistance distance between two nodes is a measure of their connectivity. Smaller resistance distance between nodes i and j indicates that they are better connected. Fig. 3.2 shows the relation between the distance



(a) WI (b) TI

Figure 3.3: Scatter plot showing the distance versus the resistance distance between nodes in the WI and TI with estimated reactance values.

and the resistance distance between nodes in the graphs defined in Subsection 5.2.3 (notice that $x_{ij} = 1\ \forall\{i,j\} \in E$).[3] As can be seen, there is no direct relation between these two measures in Erdős-Rényi and Barábasi-Albert graphs. However, in the WI and Watts-Strogatz graph the resistance distance increases with the distance.

Fig. 3.3 also shows the relation between the distance and the resistance distance in the WI and TI when the estimated reactance values (as described in Subsection 5.2.3) are used. As expected, this relation is different from the case when all the edges have equal reactance values.

3.4 Effects of a Single Edge Failure

In this section, we provide upper bounds on the flow changes after a single edge failure and introduce a metric to evaluate the robustness of grids to failures. We evaluate this metric for

[3]While in the WI the reactance values depend on the line characteristics (see values in [27]), for comparison and consistency, we used $x_{ij} = 1\ \forall\{i,j\} \in E$ in all the graphs in Fig. 3.2.

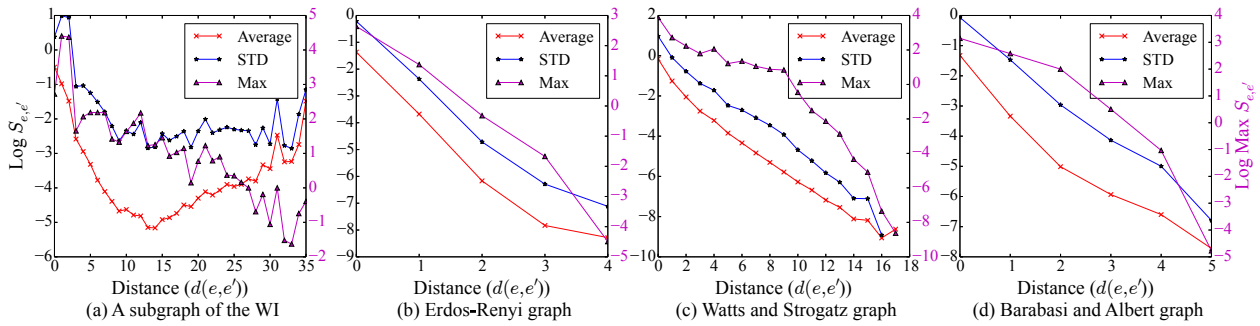


Figure 3.4: The average, standard deviation, and maximum edge flow change ratios ($S_{e,e'}$) as the function of distance ($d(e, e')$) from the failure. The right y-axis shows the values for the maximum edge flow change ratios ($\max S_{e,e'}$). All graphs have 1374 nodes. The data points are obtained for 40 random choices of an initial failure.

Watts and Strogatz graphs and demonstrate that symmetric graphs are the most robust graph structures to edge failures.

For simplicity, in this section, we assume that $x_e = 1\ \forall e \in E$, unless otherwise indicated. However, all the results can be easily generalized.

3.4.1 Flow Changes

3.4.1.1 Edge Flow Change Ratio

To provide insight into the effects of a single edge failure, we first present simulation results. Fig. 3.4 shows the edge flow change ratios ($S_{e,e'}$) as the function of distance ($d(e, e')$) from the failure for over 40 different random choices of an initial edge failure, e'. The power supply/demand in the Western interconnection is based on the actual data. In other graphs, the power supply/demand at nodes are i.i.d. Normal random variables with a slack node to equalize the supply and demand. Notice that if the initial flow in an edge is close to zero, the edge flow change ratio on that edge can be very large. Thus, to focus on the impact of an edge failure on the edges with reasonable initial flows, we do not illustrate the edge flow change ratios for the edges with flow below 1% of the average flow. Yet, we observed that such edges that experience

a flow increase after a single edge failure, are within any arbitrary distance from the initial edge failure.

Fig. 3.4 shows that after a single edge failure, there may be very large flow increases (edge flow change ratios up to 80, 14, 50, and 24 in Figs. 3.4-(a), (b), (c), and (d), respectively). These changes may occur far from the initial edge failure (edge flow change ratio around 10 for edges 11- and 4-hops away from the initial failure in Figs. 3.4-(a) and (c), respectively). Moreover, for all the four graphs, we observed that there are edges (far from the initial edge failure) whose flow changed from zero to a positive value.

These observations motivate us to prove the following result analytically. We show that by choosing specific parameter values, the edge flow change ratio can be arbitrarily large.

Observation 3.1. *For any $w_{e_1}, w_{e_2} \in \mathbb{R}$, there exists a graph $G = (V, E)$ and edges $e_1, e_2 \in E$ such that $S_{e_2,e_1} = w_{e_2}/w_{e_1}$.*

Proof. We construct the graph $G = (V, E)$ as follows: $V = \{s, t\}$, $p_s = -p_t = 1$, and there are two parallel edges e_1 and e_2 between s and t. Assume the reactance values $x_{e_1} = w_{e_1}, x_{e_2} = w_{e_2}$ are such that $0 < x_{e_1} < x_{e_2}$.

By (2.5)-(2.6), we get $f_{e_1} = \frac{x_{e_2}}{x_{e_2}+x_{e_1}}$ and $f_{e_2} = \frac{x_{e_1}}{x_{e_1}+x_{e_2}}$. If $F_0 = \{e_1\}$, then $f_{e_2}(F_0) = 1$ and $S_{e_2,e_1} = x_{e_2}/x_{e_1} = w_{e_2}/w_{e_1}$. ■

3.4.1.2 Mutual Edge Flow Change Ratio

We use the notion of *resistance distance* to find upper bounds on the mutual edge flow change ratios ($M_{e,e'}$). The following Lemma, which is an immediate result of Corollary 3.1, provides a formula for computing the flow changes after a single edge failure based on the resistance distances. It is *independent* of the power supply/demand distribution.

Lemma 3.5. *Given $G = (V, E)$, $\boldsymbol{A}$, and $\boldsymbol{A}^+$, the flow change and the mutual edge flow change ratio for an edge $e = \{i, j\} \in E$ after a failure in a non-cut-edge $e' = \{v, u\} \in E$ are,*

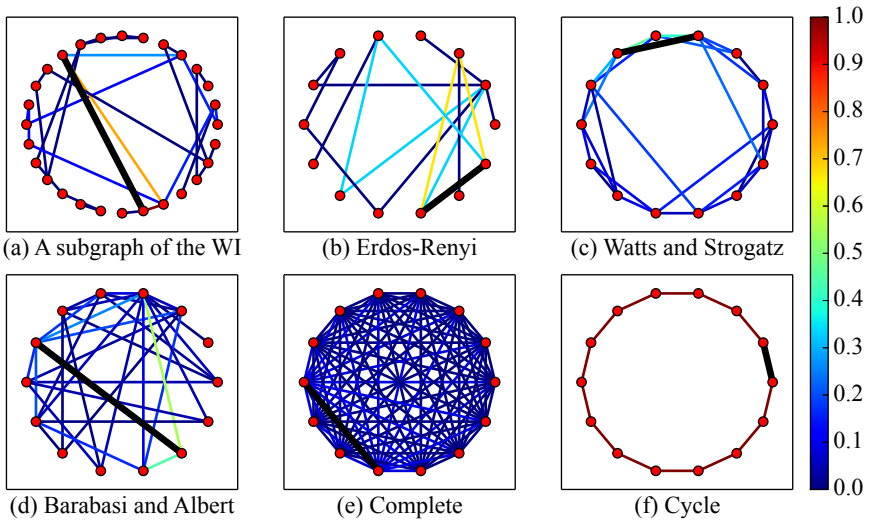


Figure 3.5: Visualization of the mutual edge flow change ratios ($M_{e,e'}$) for edges in different graph classes after a single edge failure (represented by the black wide line).

$$\begin{aligned} \Delta f_e = \Delta p_{ij} &= \frac{1}{2}\frac{-r(i,v)+r(i,u)+r(j,v)-r(j,u)}{1-r(v,u)} f_{vu}, \\ M_{e,e'} &= \left|\frac{1}{2}\frac{-r(i,v)+r(i,u)+r(j,v)-r(j,u)}{1-r(v,u)}\right|. \end{aligned}$$

Proof. It is an immediate result of Corollary 3.1. ■

Lemma 3.5 is similar to the calculation of the Line Outage Distribution Factors (LODF) presented in [155, Appendix 7B.2]. However, here we use Theorem 3.1 to derive similar equations in terms of the resistance distances. This formulation allows us to provide useful equalities and inequalities (e.g., Corollaries 3.2 and 3.6).

Fig. 3.5 shows a visualization of the mutual edge flow change ratios ($M_{e,e'}$) for edges in different graph classes after a single edge failure. These values are differently distributed for different graph classes.

The following Corollary gives an upper bound on the flow changes after a failure in a non-cut-edge $\{v,u\} \in E$ by using the triangle inequality for resistance distance and Lemma 3.5.

Corollary 3.2. *Given $G = (V, E)$, $\boldsymbol{A}$, and $\boldsymbol{A}^+$, the flow changes in any edge $e = \{i, j\} \in E$ after a failure in a non-cut-edge $e' = \{v, u\} \in E$ can be bounded by,*

$$|\Delta p_{ij}| \leqslant \frac{r(v,u)}{1 - r(v,u)} |p_{vu}|, \; M_{e,e'} \leqslant \frac{r(v,u)}{1 - r(v,u)}.$$

Proof. Using triangle inequality for resistance distance, we can write,

$$-r(i,v) + r(i,u) \leqslant r(v,u)$$
$$r(j,v) - r(j,u) \leqslant r(v,u).$$

Apply these to Lemma 3.5 completes the proof. ■

In Observation 3.1, we showed that edge flow change ratios ($S_{e,e'}$) can be arbitrarily large. However, the following observation shows that the mutual edge flow change ratios ($M_{e,e'}$) are always bounded by 1. The proof is intuitive using flow conservation in the power flows.

Observation 3.2. *For any non-cut-edge $e' = \{v, u\}$ and any edge $e \neq e'$, $M_{e,e'} \leqslant 1$.*

Proof. Suppose $\mathbf{X}$ is a $n \times 1$ matrix with 1 in v^{th} entry, -1 in u^{th} entry, and 0 elsewhere. Using Theorem 3.1 or superposition property, it is easy to verify that after a failure in an edge $e' = \{v, u\}$, $\mathbf{A'}^{+}(\vec{p} - p_{vu}\mathbf{X}) = \mathbf{A}^{+}\vec{p}$. Thus, $\vec{\theta'} - \vec{\theta} = p_{vu}\mathbf{A'}^{+}\mathbf{X} \Rightarrow \Delta\vec{\theta} = p_{vu}\mathbf{A'}^{+}\mathbf{X}$. On the other hand, it is easy to see that $\Delta\vec{\theta}$ is the phase angle of the nodes in G' when $\vec{p} = p_{vu}\mathbf{X}$. Thus, the flows should be smaller than the supply value which is equal to $|p_{vu}|$. Hence, for any $e = \{i, j\} \in E'$, $|\Delta p_{ij}| < |p_{vu}| \Rightarrow M_{e,e'} \leqslant 1$. ■

Corollary 3.3. *After a failure in a non-cut-edge $e' = \{u, v\}$, $M_{e,e'} \leqslant \min\{1, \frac{r(u,v)}{1-r(u,v)}\}$.*

Proof. Combining Observation 3.2 and Corollary 3.2 gives the result. ■

Observation 3.2 and Corollary 3.2 provide some initial upper bounds on the mutual edge flow change ratios. In the next subsection, we study the mutual edge flow change ratios in detail and demonstrate how they can be used to evaluate the robustness of graphs against failures.

We present simulations to show the relations between the mutual flow change ratios and the two distance measures. Figs. 3.6 and 3.7 show the mutual edge flow change ratio ($M_{e,e'}$) as

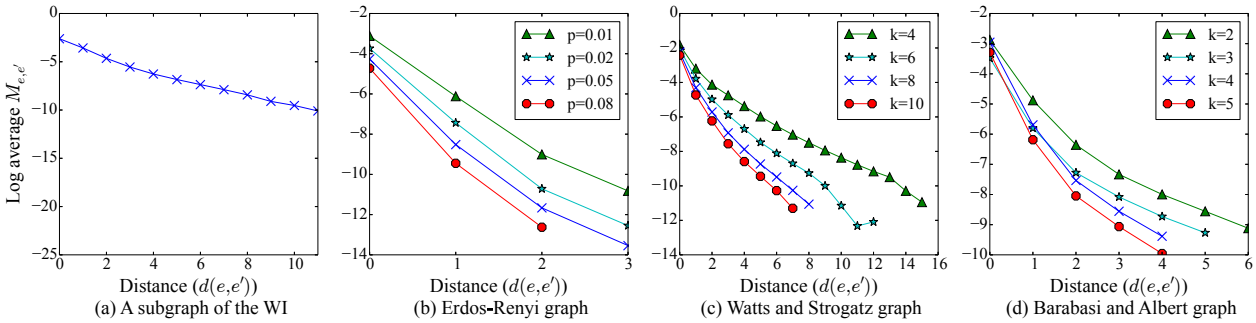


Figure 3.6: The average mutual edge flow change ratios ($M_{e,e'}$) versus the distance from the initial edge failure. All graphs have 1374 nodes. Each point represents the average of 40 different initial single edge failure events.

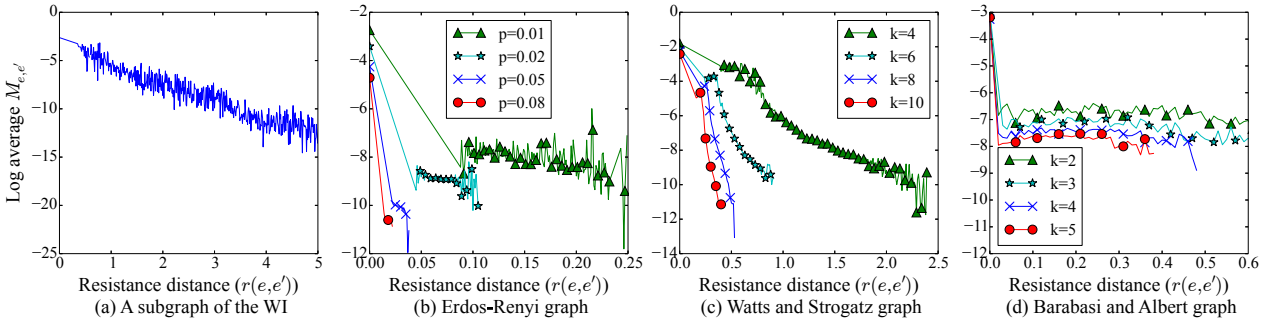
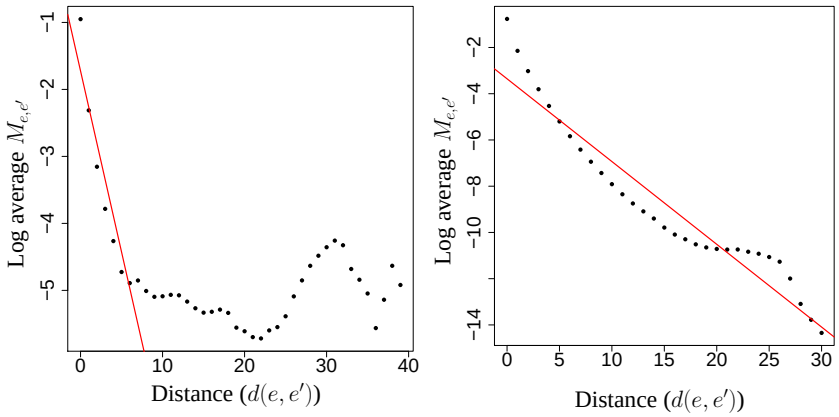


Figure 3.7: The average mutual edge flow change ratios ($M_{e,e'}$) versus the resistance distance from the initial edge failure. All graphs have 1374 nodes. Each point represents the average of 40 different initial single edge failure events. For clarity, the markers appear for every 5 data points.

the function of distance ($d(e,e')$) and resistance distance ($r(e,e')$) from the failure, respectively. The figures show that increasing the number of edges (increasing p in Erdős-Rényi graph and increasing k in Watts and Strogatz, and Barábasi and Albert graphs) affects the $M_{e,e'}$-$r(e,e')$ relation more than the $M_{e,e'}$-$d(e,e')$ relation. This suggests that the resistance distance better captures the structure of a graph. Both figures show a monotone relation between the mutual edge flow change ratios and the distances/resistance distances. However, this monotonicity is smoother in the case of the distance.



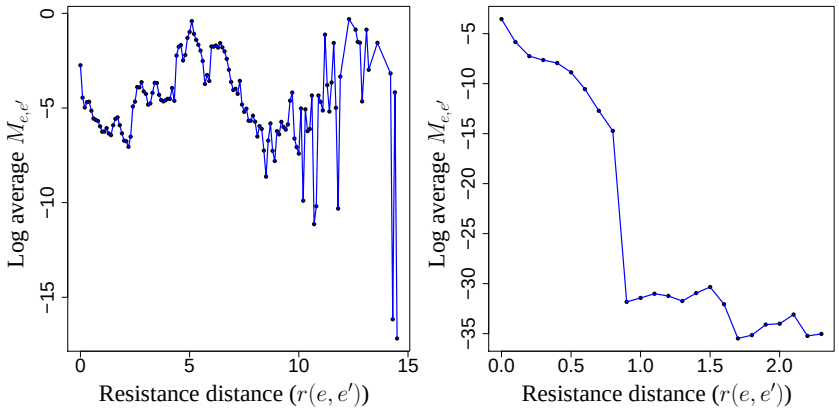
(a) WI (b) TI

Figure 3.8: The average mutual edge flow change ratios ($M_{e,e'}$) versus the distance from the initial edge failure using estimated reactance values in the WI and TI. Each point represents the average over all possible initial single edge failure events.

Moreover, Fig. 3.6, unlike Fig. 3.4, shows that after a single edge failure, the mutual edge flow change ratios decrease exponentially as the distance from the initial failure increases. Thus, it suggests that probabilistic tools may be used to model the mutual edge flow change ratios ($M_{e,e'}$) better than the edge flow change ratios ($S_{e,e'}$).

To show these relations when the estimated reactance values (as described in Subsection 5.2.3) are used, we computed the average mutual edge flow change ratios ($M_{e,e'}$) in the entire WI and the TI. As can be seen in Fig. 3.8, in the TI, the mutual edge flow change ratios decrease exponentially as the distance from the initial failure increases. This is similar to the exponential decrease observed in Fig. 3.6 when all the reactance values are equal. Similarly, in the WI this exponential decrease occurs for the distance up to 7-hops from the initial failure. However, for the longer distances, the changes remain as small as $\approx e^{-5}$.

As can be seen in Fig. 3.9, the $M_{e,e'}$-$r(e, e')$ relation is very similar to Fig. 3.7. Fig. 3.9 shows that the $M_{e,e'}$-$r(e, e')$ relation is not as smooth as the $M_{e,e'}$-$d(e, e')$ relation.



(a) WI (b) TI

Figure 3.9: The average mutual edge flow change ratios ($M_{e,e'}$) versus the resistance distance from the initial edge failure using estimated reactance values in the WI and TI. Each point represents the average over all possible initial single edge failure events.

3.4.2 Graph Robustness

In this subsection, we define the *failure cost of an edge* and the *average edge failure cost in a graph*. Using Corollary 3.1, we analytically compute these costs. We then demonstrate that the results can be used to study the robustness of graphs to single edge failures. Recall that we assume all the lines have reactance values qual to 1.

Definition 3.2. *The* failure cost *of an edge e in G is denoted by FC_e and defined as follows:* $FC_e := \frac{1}{m-1} \sum_{\substack{e' \in E \\ e' \neq e}} (M_{e',e})^2$.

The failure cost of an edge e is a good measure of the average changes that occur in the flows of the other edges to compensate for the failure in an edge e. It can help constructing a reliable power grid in two ways: (i) by designing networks with a minimum maximum failure cost, and (ii) by setting the power supply and demand values such that edges with high failure costs carry small flows. The following Lemma analytically shows the relation between the failure cost of a non-cut-edge and the resistance distance between its ending nodes. It can be considered as a generalization of the upper bound provided in Corollary 3.2.

Lemma 3.6. *In a connected graph G, for any non-cut-edge $e = \{i, j\}$,*

$$FC_e = \frac{1}{m-1}\frac{r(i,j)}{1-r(i,j)}. \tag{3.1}$$

Proof. Recall that $E = \{e_1, e_2, \ldots, e_m\}$. Assume $e := e_w = \{i, j\}$. Denote the set of edges of G with direction (as in Subsection 2.3.1) by $\mathcal{E} = \{\epsilon_1, \epsilon_2, \ldots, \epsilon_m\}$. Recall the definition of matrix $\mathbf{R}$ from Subsection 2.3.2. It is easy to see that, $\forall 1 \leqslant w, z \leqslant m : r_{wz} = a^+_{ip} - a^+_{iq} - a^+_{jp} + a^+_{jq}$ where $\epsilon_w = (i, j), \epsilon_z = (p, q)$.

Based on the definition of the resistance distance, it is easy to see that $\forall 1 \leqslant w \leqslant m : r_{ww} = r(i, j)$ where $\epsilon_w = (i, j)$. Using Corollary 3.1, it is also easy to see that $M_{e_z,e_w} = \frac{|r_{wz}|}{1-r_{ww}}$.

Consider $\mathbf{D} \in \{-1, 0, 1\}^{n\times m}$, the incidence matrix of the graph G as defined in Subsection 2.3.1. From the definition, $\mathbf{R} = \mathbf{D}^t\mathbf{A}^+\mathbf{D}$. On the other hand, $\mathbf{A} = \mathbf{D}\mathbf{D}^t$. Thus, $\mathbf{R} = \mathbf{D}^t(\mathbf{D}\mathbf{D}^t)^+\mathbf{D}$. From [14, eq. 3.11.2], $(\mathbf{D}\mathbf{D}^t)^+ = (\mathbf{D}^t)^+\mathbf{D}^+$. Moreover, from [14, eq. 3.8.2], $(\mathbf{D}^t)^+ = (\mathbf{D}^+)^t$. Thus,

$$\mathbf{R} = \mathbf{D}^t(\mathbf{D}\mathbf{D}^t)^+\mathbf{D} = \mathbf{D}^t(\mathbf{D}^t)^+\mathbf{D}^+\mathbf{D} = (\mathbf{D}^+\mathbf{D})^t(\mathbf{D}^+\mathbf{D}).$$

From the properties of the pseudo-inverse [14, Theorem 3.9], $\mathbf{D}^+\mathbf{D}$ is symmetric and $\mathbf{D}^+\mathbf{D}\mathbf{D}^+ = \mathbf{D}^+$. Hence,

$$\mathbf{R} = (\mathbf{D}^+\mathbf{D})^t(\mathbf{D}^+\mathbf{D}) = \mathbf{D}^+\mathbf{D}\mathbf{D}^+\mathbf{D} = \mathbf{D}^+\mathbf{D}.$$

Hence, from the properties of the pseudo-inverse, $\mathbf{R}^2 = \mathbf{D}^+\mathbf{D}\mathbf{D}^+\mathbf{D} = (\mathbf{D}^+\mathbf{D}\mathbf{D}^+)\mathbf{D} = \mathbf{D}^+\mathbf{D} = \mathbf{R}$. Now we compute the ww^{th} entry at both sides of the equation $\mathbf{R}^2 = \mathbf{R}$,

$$(\mathbf{R}^2)_{ww} = (\mathbf{R})_{ww} \Rightarrow \sum_{z=1}^{m} r^2_{wz} = r_{ww} \Rightarrow \sum_{\substack{z=1\\z\neq w}}^{m} r^2_{wz} = r_{ww} - r^2_{ww}$$
$$\Rightarrow \sum_{\substack{z=1\\z\neq w}}^{m} \frac{r^2_{wz}}{(1-r_{ww})^2} = \frac{r_{ww}}{1-r_{ww}}.$$

As we mentioned at the beginning of the proof, from the definition of the matrix $\mathbf{R}$, it is easy to see that $r_{ww} = r(i, j)$ and $M_{e_z,e_w} = \frac{|r_{wz}|}{1-r_{ww}}$. Hence,

$$\frac{r(i,j)}{1-r(i,j)} = \sum_{\substack{z=1\\z\neq w}}^{m} (M_{e_z,e_w})^2 = (m-1)FC_{e_w}.$$

■

Eq. (3.1) is very insightful. Intuitively, similar to the inequality in Corollary 3.2, it demonstrates that failures in edges with high resistance distance values have a strong effect on the other edges. However, it is more accurate, since it provides an equality instead of an inequality. Moreover, (3.1) allows to obtain a bound on the average edge failure cost, which is defined below as a metric for the robustness of a graph to a single edge failure.

Definition 3.3. *In a graph G with n nodes and m edges, the* average edge failure cost *is defined as,* $\overline{FC}_G := \frac{1}{m}\sum_{e\in E} FC_e$.

Using (3.1), the following Lemma provides a lower bound for the average edge failure cost in a graph.

Lemma 3.7. *In a 2-edge-connected graph G,*

$$\frac{1}{m}\Big(\frac{m-1}{n-1} - \frac{m-1}{m}\Big)^{-1} \leqslant \overline{FC}_G, \tag{3.2}$$

and equality holds if for any two edges $e = \{i, j\}$ and $e' = \{v, u\}$, $r(i, j) = r(v, u)$.

Proof. We use the same notation as in the proof of Lemma 3.6. Since G is 2-edge-connected, it does not have a cut-edge. Thus, from Lemma 3.6, for any edge $e = \{i, j\}$, $(m-1)FC_e = r(i,j)/(1-r(i,j))$. Hence, $\sum_{e\in E} FC_e = \frac{1}{m-1}\sum_{\{i,j\}\in E} \frac{r(i,j)}{1-r(i,j)}$.

From the proof of Lemma 3.6, $\mathbf{R} = \mathbf{D}^+\mathbf{D}$. Thus, $\mathrm{tr}(\mathbf{R}) = \mathrm{tr}(\mathbf{D}^+\mathbf{D}) = \mathrm{tr}(\mathbf{D}\mathbf{D}^+)$. On the other hand, from [14, eq. 3.11.7], $\mathbf{D}\mathbf{D}^+ = (\mathbf{D}\mathbf{D}^t)(\mathbf{D}\mathbf{D}^t)^+$. Since $\mathbf{D}\mathbf{D}^t = \mathbf{A}$, therefore $\mathbf{D}\mathbf{D}^+ = \mathbf{A}\mathbf{A}^+$ and from the proof of Theorem 3.1, $\mathbf{D}\mathbf{D}^+ = \mathbf{A}\mathbf{A}^+ = \mathbf{I} - \frac{1}{n}\mathbf{J}$. Hence, $\mathrm{tr}(\mathbf{R}) = \mathrm{tr}(\mathbf{I} - \frac{1}{n}\mathbf{J}) = n-1$. Therefore, $\sum_{\{i,j\}\in E} r(i,j) = \sum_{w=1}^{m} r_{ww} = \mathrm{tr}(\mathbf{R}) = n-1$.

It is easy to see that $f(x) := \frac{x}{1-x}$ is a convex function in $[0, 1)$. Thus, the minimum of the summation $\sum_{\{i,j\}\in E} \frac{r(i,j)}{1-r(i,j)}$ subject to $\sum_{\{i,j\}\in E} r(i,j) = n-1$ is when all the $r(i,j)$s are equal. In this case, $\forall\{i,j\} \in E : r(i,j)_{\mathrm{opt}} = \frac{n-1}{m}$ and $\sum_{\{i,j\}\in E} \frac{r(i,j)_{\mathrm{opt}}}{1-r(i,j)_{\mathrm{opt}}} = m\frac{n-1}{m-n+1} = (\frac{1}{n-1} - \frac{1}{m})^{-1}$. Hence,

$$\sum_{e\in E} FC_e = \frac{1}{m-1}\sum_{\{i,j\}\in E} \frac{r(i,j)}{1-r(i,j)} \geqslant \Big(\frac{m-1}{n-1} - \frac{m-1}{m}\Big)^{-1}.$$

■

Lemma 3.7 provides a lower bound on the average edge failure cost in a graph ($\overline{FC}_G$). More importantly the proof demonstrates that between all the graphs with n nodes and m edges, the one with equal resistance distance values between all pairs of connected nodes minimizes this metric.

The following Corollary shows that for symmetric graphs the equality holds in Lemma 3.7.

Corollary 3.4. *In a symmetric graph G, $\overline{FC}_G = (\frac{m^2-m}{n-1} - (m-1))^{-1}$. Moreover, for any graph H with the same number of nodes and edges as G, $\overline{FC}_H \geqslant \overline{FC}_G$.*

Proof. Since G is symmetric, for any two edges $\{i,j\},\{v,u\} \in E$, there is an automorphism $\sigma : V \to V$, such that $\sigma(i) = v$ and $\sigma(j) = u$. Suppose Π is the permutation matrix representing σ, then the admittance matrix of $\sigma(G)$ is $\hat{\mathbf{A}} = \Pi\mathbf{A}\Pi^t$. Since σ is an automorphism, $\hat{\mathbf{A}} = \Pi\mathbf{A}\Pi^t = \mathbf{A}$. It is also easy to verify that $\hat{\mathbf{A}}^+ = \Pi\mathbf{A}^+\Pi^t = \mathbf{A}^+$. Thus, $\hat{a}^+_{ii} + \hat{a}^+_{jj} - 2\hat{a}^+_{ij} = a^+_{ii} + a^+_{jj} - 2a^+_{ij}$. On the other hand, from $\hat{\mathbf{A}}^+ = \Pi\mathbf{A}^+\Pi^t$, $\hat{a}^+_{ii} = a^+_{vv}$, $\hat{a}^+_{jj} = a^+_{uu}$, and $\hat{a}^+_{ij} = a^+_{vu}$. Hence, $a^+_{vv} + a^+_{uu} - 2a^+_{vu} = a^+_{ii} + a^+_{jj} - 2a^+_{ij} \Rightarrow r(v,u) = r(i,j)$. Therefore, for any two edges $\{i,j\},\{v,u\} \in E$, $r(v,u) = r(i,j)$. Now by using Lemma 3.7, it is easy to see that $\overline{FC}_G = (\frac{m^2-m}{n-1} - (m-1))^{-1}$. Moreover, from Lemma 3.7, for any graph H with the same number of nodes and edges as G, $\overline{FC}_H \geqslant \overline{FC}_G$. ■

Corollary 3.4 demonstrates that symmetric graphs have the lowest average edge failure cost among all the graphs with the same number of nodes and edges. Moreover, from Lemma 3.7 and Corollary 3.4 it can be concluded that as graphs become more symmetrical, their average edge failure cost ($\overline{FC}_G$) decreases. To show this numerically, we computed the average edge failure cost and the maximum mutual edge flow change ratio ($\max_{e,e' \in E} M_{e,e'}$) versus the rewiring probability (p) in Watts and Strogatz graphs with 30 nodes and 60 edges. We chose the Watts and Strogatz graph, since in this type of graphs, as p increases, the symmetry of the graph decreases.

Fig. 3.10 shows the average edge failure cost of the graph ($\overline{FC}_G$) and the maximum mutual edge flow change ratio ($\max_{e,e' \in E} M_{e,e'}$) versus the probability of rewiring (p). Initially ($p = 0$), G is a 4-regular graph (namely, every node is connected to exactly 4 other nodes). However, as p increases, G tends toward a random graph with no symmetry. Thus, an increase in p in the

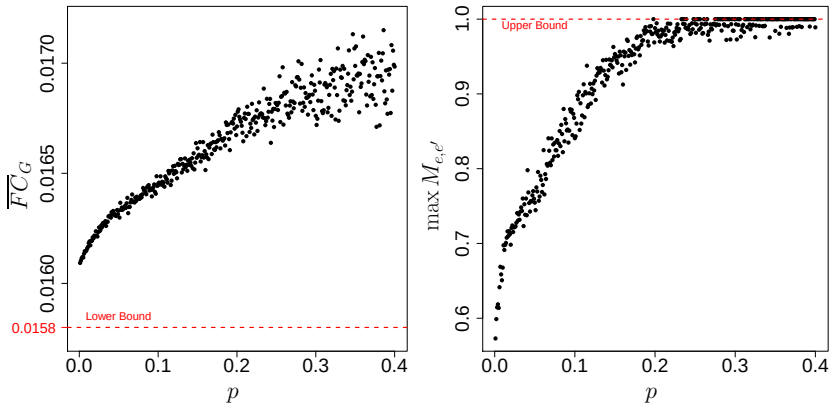


Figure 3.10: The average edge failure cost of the graph ($\overline{FC}_G$) and the maximum mutual edge flow change ratio ($\max_{e,e' \in E} M_{e,e'}$) versus the probability of rewiring (p) in a Watts and Strogatz graph with 30 nodes and 60 edges. Each point is the average over 100 generated graphs with the same parameters.

Watts and Strogatz graph can be considered as decrease in the symmetry of the graph. As we expected, the figure shows that as p increases, both the average edge failure cost of the graph ($\overline{FC}_G$) and the maximum mutual edge flow change ratio ($\max_{e,e' \in E} M_{e,e'}$) increase.

Overall, results suggest that as graphs become more symmetrical, they become more robust against single edge failures.

3.5 Efficient Cascading Failure Evolution Computation

Based on the results we obtained in Section 3.3, we present the Cascading Failure Evolution – Pseudo-inverse Based (CFE-PB) Algorithm which identifies the evolution of the cascade. The CFE-PB Algorithm uses the *Moore-Penrose Pseudo-inverse* of the admittance matrix for solving (2.7). Computing the pseudo-inverse of the admittance matrix requires $O(|V|^3)$ time. However, the algorithm obtains the pseudo-inverse of the admittance matrix in round i from the one obtained in round $(i-1)$, in $O(|F_i||V|^2)$ time. Moreover, in some cases, the algorithm can reuse the pseudo-inverse from the previous round. Since once lines fail, there is a need for low

Algorithm 2 - Cascading Failure Evolution – Pseudo-inverse Based (CFE-PB)

Input: A connected graph $G = (V, E)$ and an initial edge failures event $F_0 \subseteq E$.

1: Compute $\mathbf{A}^+$, $F_0^* \leftarrow F_0$ and $i \leftarrow 0$.
2: **while** $F_i \neq \varnothing$ **do**
3: **for** each $\{r, s\} \in F_i$ **do**
4: **if** $\{r, s\}$ is a cut-edge (see Lemma 3.1) **then**
5: Find the connected components after removing $\{r, s\}$. (see Lemma 3.2)
6: Adjust the total demand (supply) to equal the total supply (demand) within each connected component.
7: **else** update $\mathbf{A}^+$ after removing $\{r, s\}$. (see Lemma 3.4)
8: Compute the phase angles $\vec{\theta} = \mathbf{A}^+\vec{p}$ and compute new flows $f_e(F_i^*)$ from the phase angles.
9: Find the set of new edge failures $F_{i+1} = \{e||f_e| > c_e,\ e \in E \backslash F_i^*\}$. $F_{i+1}^* \leftarrow F_i^* \cup F_{i+1}$ and $i \leftarrow i+1$.
10: **return** $t = i - 1$, $(F_0, \ldots, F_t)$, and $f_e(F_t^*)$ $\forall e \in E \backslash F_t^*$.

complexity algorithms to control and mitigate the cascade, the CFE-PB Algorithm may provide insight into the design of efficient cascade control algorithms.

We now describe the CFE-PB Algorithm. It initially computes the pseudo-inverse of the admittance matrix (in $O(|V|^3)$ time) and this is the only time in which it *computes* $\boldsymbol{A}^+$ *without using a previous version of* $\boldsymbol{A}^+$. Next, starting from F_0, at each round of the cascade, for each $e \in F_i$, it checks whether e is a cut-edge (Line 4). This is done in $O(1)$ (Lemma 3.1). If yes, based on Lemma 3.3, in Lines 5 and 6, the total demand is adjusted to equal the total supply within each connected component (in $O(V)$ time). Else, in Line 7, $\mathbf{A}^+$ after the removal of e is computed in $O(|V|^2)$ time (see Lemma 3.4). After repeating this process for each $e \in F_i$, the phase angles and the flows are computed in $O(|V|^2)$ time (Line 8). The rest of the process is similar to the CFE Algorithm.

The following theorem provides the complexity of the algorithm (the proof is based on the Lemmas 1–4). We show that the algorithm runs in $O(|V|^3 + |F_t^*||V|^2)$ time (compared to the CFE Algorithm which runs in $O(t|V|^3)$). Namely, if $t = |F_t^*|$ (one edge fails at each round), the CFE-PB Algorithm outperforms the CFE Algorithms by $O(\min\{|V|, t\})$.

Theorem 3.2. *CFE-PB Algorithm runs in* $O(|V|^3 + |F_t^*||V|^2)$.

Proof. Finding the pseudo inverse of the matrix requires $O(|V|^3)$ time. Therefore, Line 1 takes $O(|V|^3)$ time. Lines 5 and 6 in the algorithm take $O(|V|)$ time and Line 7 takes $O(|V|^2)$, therefore the whole **for** loop takes at most $O(|F_i||V|^2)$ time at each step. Using $\mathbf{A}^+$ computed in the **for** loop, Lines 8 and 9 take $O(|V|^2)$ time. Thus, the total running time of the algorithm is at most $O(|V|^3) + O((|F_0| + |F_1| + \cdots + |F_t|)|V|^2) = O(|V|^3) + O(|F_t^*||V|^2)$. ■

Note that a similar approach (the step by step rank-1 update) can be applied to other linear equations solution methods (e.g., LU factorization [77]). Yet, using the pseudo-inverse allows developing tools for analyzing the effect of a single edge failure (as showed in Section 3.4) and supports the development of an algorithm for finding the most vulnerable edges (see Section 3.6).

3.6 Hardness and Heuristic

In this section, we establish that deciding if there exists a failure event (of size at most a given value) such that the yield after stabilization is less than a given threshold, is NP-complete. Using the results from Section 3.4, however, we introduce a heuristic for the problem of finding an initial failure that causes a cascade resulting with the minimum possible yield (*minimum yield problem*). We numerically show that in most cases, solutions obtained by the heuristic lead to a yield comparable to the solutions obtained by more numerically complex methods.

Lemma 3.8. *Given* G*, a real number* y*,* $0 \leqslant y \leqslant 1$*, and an integer* $k \geqslant 1$*, deciding if there exists a set of initial edge failures of the size at most* k *resulting in a yield less than* y *is NP-complete.*

Proof. Consider following problem:

1. Suppose $G = (V, E)$ is an instance of the classical flow problem, with a single source node $\{s\}$ and set of sink nodes T. Assume demands are equal to 1 and lines have unbounded capacity ($O(|V|)$). Does a subset of edges $\mathcal{A} \subseteq E$ with $|\mathcal{A}| \leqslant k$ exist such that $|T_{fail}| \geqslant l$? (T_{fail} is the set of sink nodes which get disconnected from the source node s after removing set of edges $\mathcal{A}$.)

It is proved in [20, Theorem 7], that problem above is NP-complete. We want to use this result to proof Lemma 3.8. For this reason we provide a polynomial time reduction from problem above to minimum yield problem.

2. Suppose $G = (V, E)$ is an instance of the power flow problem, with set of supply node $S = \{s\}$ and set of demand nodes T. Assume $p_t = -1$ for all $t \in T$, and $p_s = |T|$. Assume all the lines have capacities equal to $|T|$ and reactance values equal to 1. Is there a set of initial failures with size less than k resulting in yield $Y \leqslant 1 - \frac{l}{|T|}$?

Claim. *Suppose the graphs in problems 1 and 2 are the same, then the answer to problem 1 is* yes *if, and only if, the answer to problem 2 is* yes*.*

Proof. ($\Rightarrow$) Assume the answer to problem 1 is *yes*. It means that there exists a set of edges $\mathcal{A} \subseteq E$ with $|\mathcal{A}| \leqslant k$ such that their removal disconnects at least l of the sink nodes from the source node. Now in problem 2, choose $F_0 = \mathcal{A}$. Since two graphs are the same, at least l of the demand nodes are disconnected from the supply node s. As a result, final yield is at most $|T| - l$. Since initial supplied demand was $|T|$, $Y \leqslant 1 - \frac{l}{|T|}$.
($\Leftarrow$) Now the other way, assume the answer to problem 2 is *yes*. It means that there is an initial set of edge failures $F_0 \subseteq E$ with $|F_0| \leqslant k$ such that $Y \leqslant 1 - \frac{l}{|T|}$. First, since all the edges have capacity equal to $|T|$ which is an upper bound for a flow in an edge, after initial set of failures, there is no cascade. Therefore, there is no further edge failures. Second, with the same reason, as long as a demand node is connected to the supply node, its demand can be satisfied. Now since $Y \leqslant 1 - \frac{l}{|T|}$, with initial set of failure F_0, at least l of the demand nodes are disconnected from supply node s. In problem 1 choose $\mathcal{A} = F_0$, since the graphs in two problems are the same, by removing set of edges $\mathcal{A}$ from G, at least m of the sink nodes are disconnected from source node s. Since $|\mathcal{A}| = |F_0| \leqslant k$, the answer to problem 1 is also *yes*. □

It can be concluded from this claim that problem 1 can be reduced to problem 2. Hence, the minimum yield problem is NP-hard and its decision version is NP-complete. ■

As indicated in Lemma 3.8, the minimum yield problem is NP-hard. We now present a heuristic algorithm for solving this problem when $x_e = 1 \ \forall e \in E$. We refer to it as the Most

Algorithm 3 - Most Vulnerable Edges Selection – Resistance distance Based (MVES-RB)

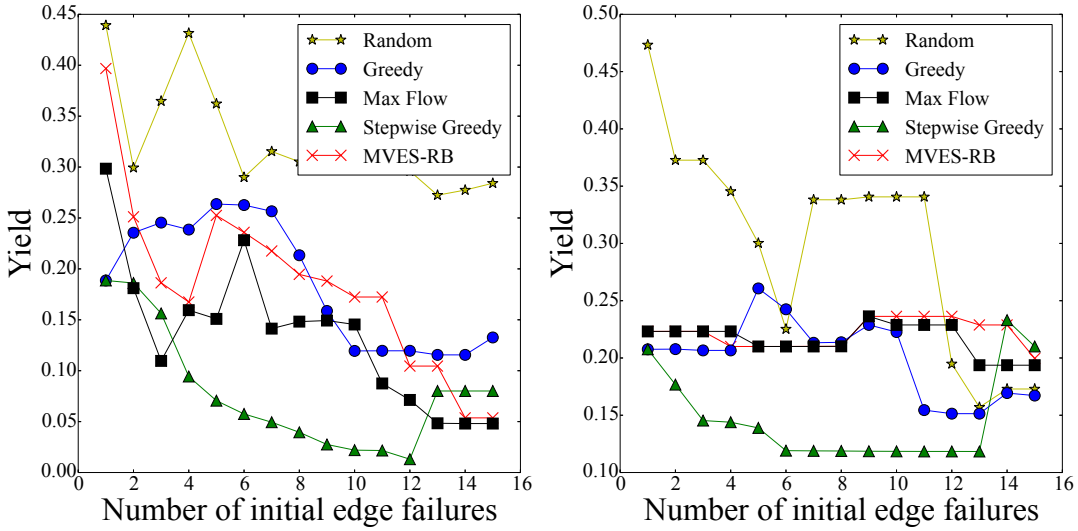
Input: A connected graph $G = (V, E)$ and an integer $k \geqslant 1$.

1: Compute $\mathbf{A}^+$.
2: Compute the phase angles $\vec{\theta} = \mathbf{A}^+\vec{p}$ and compute flows f_e from the phase angles.
3: Compute the resistance distance $r(i, j) = r(e)\ \forall e = \{i, j\} \in E$.
4: Sort edges $e_1, e_2, \ldots, e_{|E|}$ such that $v \leqslant u$ iff $|f_{e_v} r(e_v)| \geqslant |f_{e_u} r(e_u)|$.
5: **return** $e_1, e_2, \ldots, e_k$.

Vulnerable Edge Selection – Resistance distance Based (MVES-RB) Algorithm. From Corollary 3.6, edges with large $r(i, j)$ have larger failure costs. Thus, edges with large $r(i, j) \times |f_{ij}|$ have greater impact on the flow changes on the other edges. Based on this result, the MVES-RB Algorithm selects *the k edges with highest $r(i, j) \times |f_{ij}|$ values as the initial set of failures* in $O(|V|^3)$.

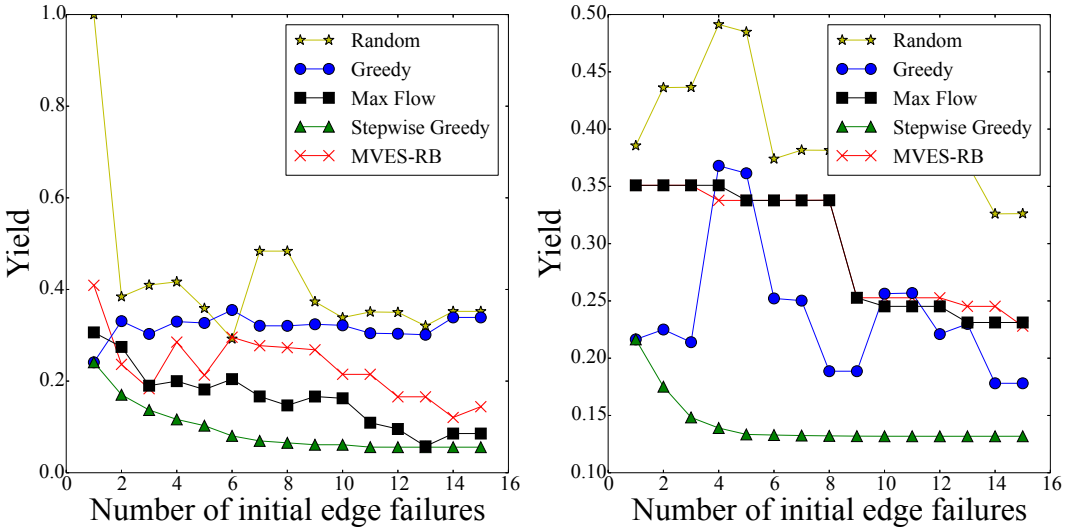
The MVES-RB Algorithm is in the same category as the algorithms that identify the set of failures with the largest impact (i.e., algorithms that solve the $N-k$ *problem* [30,113]). However, none of the previous works focusing on the $N - k$ problem, considers cascading failures. To evaluate the performance of the MVES-RB Algorithm, we compare its performance to that of the four other intuitive methods for selecting the initial set of failures: (i) Random, (ii) Greedy, (iii) Max-flow, and (iv) Stepwise greedy.

In Random selection, k initial edge failures are randomly selected in $O(k)$ time. In Greedy selection, for each edge $e \in E$, the yield is computed after an initial failure in that edge. Then, the k edges that have the lowest resulting yield values are removed. The Greedy selection takes $O(t|V|^3|E|)$ time. In Max-flow selection, the k edges with the maximum amount of initial flow are selected in $O(|V|^3)$ time. Stepwise-greedy selection is a step by step selection method. At each step, an edge e is selected such that if e is removed together with the previously selected edges, the yield is minimized. For $k = 1$, both Stepwise-greedy and Greedy selection select an edge that upon its failure minimizes the yield. The running time of the Stepwise selection method is $O(t|V|^3|E|k)$.



(a) IEEE 118-bus (b) IEEE 300-bus

Figure 3.11: Comparison between the effectiveness of different methods for selecting the most vulnerable edges when the factor of safety is $\alpha = 0.1$.



(a) IEEE 118-bus (b) IEEE 300-bus

Figure 3.12: Comparison between the effectiveness of different methods for selecting the most vulnerable edges when the factor of safety is $\alpha = 0.2$.

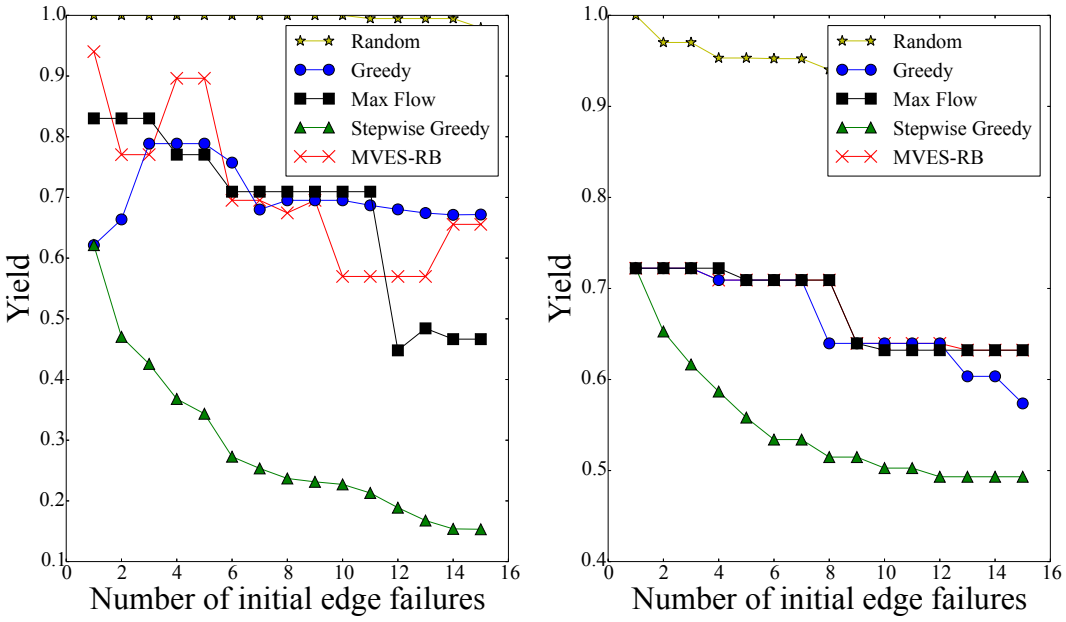
To compare the performance of different selection methods, we computed the yield after selecting $k = 1, \ldots, 15$ edges as the initial failed edges in the IEEE 118- and 300-bus benchmark systems. For all the edges the reactance values are $x_e = 1$. For capacities, we consider two options: (i) for each edge $c_e = (1 + \alpha)|f_e|$,[4] where f_e is the initial flow on the edge, and (ii) all edges have equal capacities equal to $c_e = 1.2 \max_{e'} |f_{e'}|$ (thereby removing the effect of asymmetry in the capacities).

Figs. 3.11 and 3.12 illustrate the results when capacities are chosen based on the initial flows with $\alpha = 1.1$ and $\alpha = 1.2$, respectively. As can be seen, the Stepwise-greedy method that has the worst running time outperforms the other methods in most of the cases. However, the Max-flow method and the MVES-RB Algorithm that have much lower running time than the Stepwise-greedy method, perform relatively well and in most cases better than the Greedy method. The Random selection method that has the lowest running time also seems to perform well in some cases.

Fig. 3.13 illustrates the results when all the edges have equal capacities. As can be seen, in that case, the Random selection is not comparable to other methods. As it was previously the case, the Stepwise-greedy method outperforms other methods here as well. The MVES-RB Algorithm, the Max-flow, and Greedy methods perform equally good and none of them outperforms the other. The running time of the MVES-RB Algorithm and Max-flow method, however, is much less than the Greedy method.

Overall, the Stepwise-greedy method outperforms others in most of the cases. However, its high running time makes this method impractical in larger networks. Thus, since both the MVES-RB Algorithm and the Max-flow method have a much lower running time and perform relatively good in the most cases, they seem to be better options.

[4]Following [27], we assume that the capacities are $1 + \alpha$ times the initial flows. α is often referred to as the *Factor of Safety (FoS)* of the grid.



(a) IEEE 118-bus

(b) IEEE 300-bus

Figure 3.13: Comparison between the effectiveness of different methods for selecting the most vulnerable edges when capacities are all equal to $1.2 \max_e |f_e|$.

3.7 Conclusion

In order to better understand failures and the evolution of cascades in the power grid, we studied properties of the admittance matrix of the grid and provided analytical tools for computing the impact of a single edge failure on the flows on the other edges. Based on these tools, we derived upper bounds on the flow changes after a single edge failure and introduced a metric to assess the robustness of graphs to single edge failures. We illustrated via simulations the impact of such failures. Then, we showed the unique properties of the cascading failure model and introduced a pseudo-inverse based efficient algorithm to identify the cascade evolution. Finally, we proved that the minimum yield problem is NP-hard and introduced a simple heuristic to detect the most vulnerable edges.

In a follow up to our work presented in this chapter, in [138], we extensively studied cascading failures' spatial and temporal properties in the U.S. Western Interconnection (WI) power grid with collaboration with researchers at Yeshiva University. Based on simulations, we observed that the yield has a bimodal distribution (i.e., the failure of a randomly selected line leads either to an insignificant power loss or to a major blackout). We found that there is a latent period in the development of major blackouts where few lines are overloaded and the yield remains high. The duration of this latent period is proportional to the line tolerance (the ratio of the maximum load a line can carry to its initial load). The existence of the latent period suggests that intervention during the early stages of a cascade can significantly reduce the risk of a major blackout. Our recent observations as well as the results provided in this Chapter show that due to their unique properties, cascades in power grids require more studies to be fully understood and controlled.

In another recent collaborative work [40], we studied cascades under the more detailed AC power flow model and compared the severity of the cascade under the AC and DC models. We observed that despite the strong correlation between the results under the two models, the DC cascade model can lead to an inaccurate and overly optimistic cascade predictions in large power grids. Hence, more studies under the AC power flows is needed to obtain a more accurate understanding of the cascades in power grids.

Chapter 4

Quantifying the Effects of k-line Failures

Power grids are required to withstand a single component failure, known as the $N-1$ contingency standard. However, one or more lines can often be out of service for various reasons such as maintenance and construction works. This can result in the violation of the $N-1$ standard and vulnerability of the power grid to a single failure, as was the case in Turkey in 2015 [12]. Therefore, higher order contingency analysis is necessary to detect critical events caused by more than a single failure.

Due to the large number of possibilities, high order contingency analysis is computationally expensive. Therefore, most of the previous tools are effective for analyzing contingencies caused by failures in one or two components of the grid [30, 47, 58, 59, 66, 67, 92, 144]. In order to circumvent this issue, in this chapter, we introduce the *disturbance value* of a failure. We show that this value can be efficiently computed for any set of line failures independently of the size of the grid and can be effectively used to filter out less crucial contingencies. The disturbance value can therefore significantly reduce the time complexity of contingency analysis by revealing contingencies that are vital for more in depth analysis.

First, we extend and build on the results in Section 3.4 which focused on single line failures to analytically compute the effect of *multiple line failures* on redistribution of power flows. We

refer to an event resulting in the failure of k distinct lines, a *k-line failure event*. Similar to the most of the previous work on contingency analysis, we use the linearized DC approximation of the power flows, due to the complexities associated with the AC power flow model. We present an analytical update of the pseudo-inverse of the admittance matrix after a k-line failure event. Our approach is similar to [79], but we use the pseudo-inverse instead of the truncated inverse of the admittance matrix which requires selecting a node as the slack bus. An advantage of using pseudo-inverse is that it allows having a unified formula for all k-line failures regardless of their location and their connectivity to the slack bus.[1] Using this result, we define and analytically compute the *k-line outage distribution matrix* which generalizes the definition of the line outage distribution factors for single line failures [155] or as we called it, mutual edge flow change ratio in Chapter 3.

While the k-line outage distribution matrix captures all effects of a k-line failure on flow changes, it is not efficient to compute and store this matrix for contingency analysis in large power grids. To overcome this challenge, we use the *matrix of equivalent reactance values* as we did in Subsection 3.4.2 to efficiently compute the sum of changes in the power flows after a k-line failure and to provide a metric that captures the essence of the flow changes after failures. In particular, we define and analytically compute the *disturbance value* of a failure (the weighted sum of squares of the flow changes) and show that this computation can be done for a k-line failure in $O(1)$ as long as k is much smaller than the total number of lines, the case in contingency analysis of the power grids. Hence, the disturbance value of a k-line failure can be computed independently of the size of the grid.

To show that the disturbance values provide a separation between failures with higher impact and lower impact, we compute the disturbance values for all possible choices of 3-line failures in the IEEE 118-bus and 300-bus systems. We demonstrate that by ranking cases based on their disturbance values and considering only cases with high disturbance value, we are able to

[1] Another advantage of using pseudo-inverse is that our results can be tied to the related notions of *Laplacian* and *resistance distances* in graph theory [22]. This enables further use of existing tools in graph theory for contingency analysis in power grids as in [128].

decrease the total number of cases needed to be analyzed for contingency analysis by more than 90%.

To analyze all the k-line failure scenarios, one needs to compute the disturbance values for all possible $\binom{m}{k}$ cases which requires $O(m^k)$ time. To alleviate this issue, we provide an approximation for the disturbance values. We show that by approximating the disturbance value of a k-line failure, one can only focus on the $\widetilde{m} << m$ lines with the highest 1-line disturbance values and analyze only $\binom{\widetilde{m}}{k}$ cases instead of $\binom{m}{k}$ cases. Moreover, we numerically show that the approximation error is below 10% in most of the cases in the 118- and 300-bus systems.

To show that disturbance values effectively rank k-line failures, we numerically compute the relationship between disturbance value of all 3-line failures and the severity of the initiated cascades in the IEEE 118- and 300-bus systems. We show that although the disturbance value does not take into account the capacities of the lines, it can predict the average severity of the cascade initiated by a failure.

Finally, we numerically show the usefulness of the disturbance values in predicting changes under the more detailed AC power flows after a k-line failure. In particular, we compute the disturbance values under the AC power flows and show that the disturbance values under the AC and DC differ only in scaling. We also study the correlation between the voltage changes after a k-line failure and disturbance values, and show that k-line failures with higher disturbance values cause more voltage changes as well.

Since the AC power flow equations are nonlinear, there are k-line failures after which the AC power solver does not converge to a solution. We show that almost all such failures in the IEEE 118- and 300-bus systems have the disturbance values that are between the top 10% of all the k-line failures in terms of the disturbance values.

4.1 Related Work

Many great ideas have been developed for contingency analysis in power grids since the advent of the modern power transmission network. Detecting most important lines and nodes solely

based on the topology of the power grid was studied in [151] using network centrality measures. *Compensation theorem* and *current injection* methods were used to analyze the effect of line failures in [58,59,66,92,155]. In particular, [155] introduced the notion of *line outage distribution factors* that inspired many other studies including the work presented in this and previous chapter. [86,144] used matrix updates to study the effect of two line failures and used the results to introduce an algorithm for the $N-2$ contingency problem (contingencies caused by failures in at most 2 components of the grid). In a follow up work [87], *contingency and Influence graphs* were introduced to study $N-2$ contingency analysis.

More optimization-based techniques for contingency analysis of the grids were explored in [47, 63]. In particular, [47] focused on identifying the most probable failure modes in static load distribution using a linear-program. In a recent innovative paper, probabilistic algorithms were developed to identify collections of multiple contingencies that initiate cascading failure [83].

High order contingency analysis was studied in [30, 45, 114]. In [45], the contingencies were ordered based on their empirical occurrence probabilities and only contingencies with high probability were considered. A mixed-integer model for the $N-k$ contingency problem was presented and used in [30]. However, this method does not scale well as k increases. In [114], the resistance distances are used as in [128] to identify most important lines and to prune the graph. The contingency analysis was then performed on the reduced graph instead of the entire graph in order to reduce the total number of contingencies. However, since resistance distances are computed independently of supply and demand values, such an approach may miss contingencies that are caused by failures in lines carrying large power flows.

The main advantages of using disturbance value for contingency analysis, as proposed in this chapter, over previous work is that: (i) using techniques from linear algebra, the disturbance value can be efficiently computed for any number of line failures independently of the size of a grid, and therefore it scales well for high order contingency analysis, (ii) due to its simplicity, it is easy to implement, and (iii) since the disturbance value is based on the power flows as well as the admittance matrix, it captures both the topological and operational properties of the system.

4.2 Model

We consider contingencies caused by the set of line failures of size k denoted by $L \subseteq E$. We refer to these failures as *k-line failures*. Without loss of generality, for convenience we assume $L = \{e_1, e_2, \ldots, e_k\}$. We denote the graph after failures by $G' = (V', E')$, in which $E' = E - L$ and $V' = V$. We also assume that removing edges in L from G does not disconnect the graph. Notice that failures that disconnect the grid have the highest priority, since they most likely divide the grid into unbalanced islands in terms of supply/demand values that may result in further failures. Therefore, such failures should always be considered for more in depth contingency analysis. In Section 4.4, we show that some of our results can be used to approximately rank k-line failures that disconnect the grid based on their criticality among themselves.

Hence, here we assume G' is connected. Upon failures, the power flows redistributed in G' based on the equation $\mathbf{A}'\vec{\theta'} = \vec{p}$, in which $\mathbf{A}'$ is the admittance matrix of G'. In Section 4.3, we use linear algebra to compute $\mathbf{A}'^{+}$ and quantify the effect of k-line failures.

Notation. In this Chapter, $\vec{y}_k$ denotes the subvector of $\vec{y}$ with its first k entries. We use $\bar{k}$ to show the indices other that 1 to k (e.g., $\vec{f}_{\bar{k}}$ denotes the subvector of $\vec{f}$ with its $k+1$ to m entries). We denote the submatrix of $\mathbf{Q}$ limited to the first k columns by $\mathbf{Q}_k$ and the submatrix of $\mathbf{Q}$ limited to the first k rows and columns by $\mathbf{Q}_{k|k}$. We define $\Delta\vec{f}_{\bar{k}} = \vec{f}_{\bar{k}} - \vec{f}'_{\bar{k}}$ to show the flow changes on the lines in $E \backslash L$ after the failure in lines in L.

4.3 Failure Analysis

In this section, we study the effect of k-line failures on the flow changes on the other lines using the DC power flows as described in Section 2.3.

The following lemma is the generalization of Lemma 3.1 and is similar to the idea used in [79, Theorem 2] to detect the connected components after multiple line failures. Our proof is different from the proof of a similar Theorem in [79, Theorem 2].

Lemma 4.1. *Matrix* $\boldsymbol{I} - \boldsymbol{Y}_{k|k}^{1/2}\boldsymbol{D}_k^t\boldsymbol{A}^{+}\boldsymbol{D}_k\boldsymbol{Y}_{k|k}^{1/2}$ *is invertible if, and only if* G' *is connected.*

Proof. First, it is easy to see that $\mathbf{I} - \mathbf{Y}_{k|k}^{1/2}\mathbf{D}_k^t\mathbf{A}^+\mathbf{D}_k\mathbf{Y}_{k|k}^{1/2}$ is invertible if, and only if $\mathbf{I} - \mathbf{Y}_{k|k}\mathbf{D}_k^t\mathbf{A}^+\mathbf{D}_k$ is invertible. Now assume, G' is disconnected. Without loss of generality, assume $C = \{e_1, e_2, \ldots, e_r\}$ is a minimal subset of $\{e_1, e_2, \ldots, e_k\}$ such that $G\backslash C$ is disconnected. Since, C is a minimal subset, $G\backslash C$ has only two connected components G_1 and G_2 and each $e_i \in C$ has one end in G_1 and the other end in G_2. Again with out loss of generality, assume that all the edges in C are directed from G_1 to G_2. We prove that vector $\vec{v} \in \mathbb{R}^k$ defined as $v_i = y_{ii}$ for $i \leqslant r$ and $v_i = 0$ for $i > r$ is an eigenvector of $\mathbf{Y}_{k|k}\mathbf{D}_k^t\mathbf{A}^+\mathbf{D}_k$ associated with the eigenvalue 1. Notice that if we set $\vec{p} = \mathbf{D}_k\vec{v}$, then $\theta_i = 1$ for all $i \in G_1$ and $\theta_i = 0$ for all $i \in G_2$ gives a solution to DC power flow problem in G. It is easy to see that in this setting $\vec{f_k} = \vec{v}$. On the other hand, $\vec{f_k} = \mathbf{Y}_{k|k}\mathbf{D}_k^t\mathbf{A}^+\vec{p}$, and since $\vec{f_k} = \vec{v}$ and $\vec{p} = \mathbf{D}_k\vec{v}$, therefore $\mathbf{Y}_{k|k}\mathbf{D}_k^t\mathbf{A}^+\mathbf{D}_k\vec{v} = \vec{v}$. Hence, $\mathbf{Y}_{k|k}\mathbf{D}_k^t\mathbf{A}^+\mathbf{D}_k$ has eigenvalue 1 and $\mathbf{I} - \mathbf{Y}_{k|k}\mathbf{D}_k^t\mathbf{A}^+\mathbf{D}_k$ is not invertible.

Now assume, $\mathbf{I} - \mathbf{Y}_{k|k}\mathbf{D}_k^t\mathbf{A}^+\mathbf{D}_k$ is not invertible. Then, $\mathbf{I} - \mathbf{Y}_{k|k}\mathbf{D}_k^t\mathbf{A}^+\mathbf{D}_k$ has an eigenvalue 0 and $\mathbf{Y}_{k|k}\mathbf{D}_k^t\mathbf{A}^+\mathbf{D}_k$ has an eigenvalue 1. Assume $\vec{v}$ is the eigenvector associated with the eigenvalue 1 of $\mathbf{Y}_{k|k}\mathbf{D}_k^t\mathbf{A}^+\mathbf{D}_k$. It is again easy to see that since $\mathbf{Y}_{k|k}\mathbf{D}_k^t\mathbf{A}^+\mathbf{D}_k\vec{v} = \vec{v}$, if we set $\vec{p} = \mathbf{D}_k\vec{v}$, then $\vec{f_k} = \vec{v}$ is the solution to the power flow problem in G. From the flow conservation equations, it is also easy to verify that $\vec{f_{\bar{k}}} = 0$. Now, by contradiction assume G' is connected. Then, there should be a path in G' from a node i to node j such that $\theta_i \neq \theta_j$. Therefore, there should be an edge $e = (w, z)$ in this path such that $\theta_w \neq \theta_z$ and thus $f_e \neq 0$. However, since $e \in G'$ and $\vec{f_{\bar{k}}} = 0$ we know that $f_e = 0$ which is a contradiction. Therefore, G' is not connected.

From the proof it can be seen that if $\vec{v}$ is an eigenvector associated with the eigenvalue 1 of $\mathbf{Y}_{k|k}\mathbf{D}_k^t\mathbf{A}^+\mathbf{D}_k$, then nodes with the same phase angle values in the solution of the power flow problem in G with $\vec{p} = \mathbf{D}_k\vec{v}$ form a connected component in G'. This can be used to efficiently find the connected components of G' after a k-line failure. ■

In the following lemma, we generalize Theorem 3.1 for single line failures and provide an analytical update of the pseudo-inverse of the admittance matrix following a k-line failure.

Lemma 4.2. *If G' is connected,*

$$\boldsymbol{A'}^{+} = \boldsymbol{A}^{+} + \boldsymbol{A}^{+}\boldsymbol{D}_k\,\boldsymbol{Y}_{k|k}^{1/2}[\boldsymbol{I} - \boldsymbol{Y}_{k|k}^{1/2}\boldsymbol{D}_k^t\boldsymbol{A}^{+}\boldsymbol{D}_k\,\boldsymbol{Y}_{k|k}^{1/2}]^{-1}\,\boldsymbol{Y}_{k|k}^{1/2}\boldsymbol{D}_k^t\boldsymbol{A}^{+}.$$

Proof. First, from Lemma 4.1, since G' is connected, $[\mathbf{I}-\mathbf{Y}_{k|k}^{1/2}\mathbf{D}_k^t\mathbf{A}^+\mathbf{D}_k\mathbf{Y}_{k|k}^{1/2}]^{-1}$ is defined. Now to show the equality, it is easy to see that $\mathbf{A}\mathbf{A}^+ = \mathbf{I}-\frac{1}{n}\mathbf{J}$, in which $\mathbf{I}$ is the identity matrix and $\mathbf{J}$ is all 1 matrix (For more details see the proof of Theorem 3.1). Hence, from [14, Theorem 4.8], since $\mathbf{A}' = \mathbf{A}-\mathbf{D}_k\mathbf{Y}_{k|k}\mathbf{D}_k^t = \mathbf{A}-\mathbf{D}_k\mathbf{Y}_{k|k}^{1/2}(\mathbf{D}_k\mathbf{Y}_{k|k}^{1/2})^t$, the pseudo inverse of $\mathbf{A}'$ can be computed as,

$$\mathbf{A}'^{+} = \mathbf{A}^+ + \mathbf{A}^+\mathbf{D}_k\mathbf{Y}_{k|k}^{1/2}[\mathbf{I}-\mathbf{Y}_{k|k}^{1/2}\mathbf{D}_k^t\mathbf{A}^+\mathbf{D}_k\mathbf{Y}_{k|k}^{1/2}]^{-1}\mathbf{Y}_{k|k}^{1/2}\mathbf{D}_k^t\mathbf{A}^+.$$

■

From Lemma 4.2, the changes in phase angles after k-line failures can be computed as,

$$\begin{aligned}\vec{\theta}'-\vec{\theta} &= (\mathbf{A}'^{+}-\mathbf{A}^+)\vec{p}\\ &= \mathbf{A}^+\mathbf{D}_k\mathbf{Y}_{k|k}^{1/2}[\mathbf{I}-\mathbf{Y}_{k|k}^{1/2}\mathbf{D}_k^t\mathbf{A}^+\mathbf{D}_k\mathbf{Y}_{k|k}^{1/2}]^{-1}\mathbf{Y}_{k|k}^{1/2}\mathbf{D}_k^t\mathbf{A}^+\vec{p}\\ &= \mathbf{A}^+\mathbf{D}_k\mathbf{Y}_{k|k}^{1/2}[\mathbf{I}-\mathbf{Y}_{k|k}^{1/2}\mathbf{D}_k^t\mathbf{A}^+\mathbf{D}_k\mathbf{Y}_{k|k}^{1/2}]^{-1}\mathbf{Y}_{k|k}^{-1/2}\vec{f}_k\\ &= \mathbf{A}^+\mathbf{D}_k\mathbf{Y}_{k|k}^{1/2}[\mathbf{I}-\mathbf{Y}_{k|k}^{1/2}\mathbf{R}_{k|k}\mathbf{Y}_{k|k}^{1/2}]^{-1}\mathbf{Y}_{k|k}^{-1/2}\vec{f}_k. \qquad (4.1)\end{aligned}$$

Recall from Subsection 2.3.2 that matrix $\mathbf{R}$ is the matrix of equivalent reactance values. Using (4.1), we can compute the changes in the flows as,

$$\begin{aligned}\Delta\vec{f}_{\bar{k}} &= \mathbf{Y}_{\bar{k}|\bar{k}}\mathbf{D}_{\bar{k}}^t\mathbf{A}^+\mathbf{D}_k\mathbf{Y}_{k|k}^{1/2}[\mathbf{I}-\mathbf{Y}_{k|k}^{1/2}\mathbf{R}_{k|k}\mathbf{Y}_{k|k}^{1/2}]^{-1}\mathbf{Y}_{k|k}^{-1/2}\vec{f}_k\\ &= \mathbf{Y}_{\bar{k}|\bar{k}}\mathbf{R}_{\bar{k}|k}\mathbf{Y}_{k|k}^{1/2}[\mathbf{I}-\mathbf{Y}_{k|k}^{1/2}\mathbf{R}_{k|k}\mathbf{Y}_{k|k}^{1/2}]^{-1}\mathbf{Y}_{k|k}^{-1/2}\vec{f}_k. \qquad (4.2)\end{aligned}$$

It is important to see that $\mathbf{Y}_{\bar{k}|\bar{k}}\mathbf{R}_{\bar{k}|k}\mathbf{Y}_{k|k}^{1/2}[\mathbf{I}-\mathbf{Y}_{k|k}^{1/2}\mathbf{R}_{k|k}\mathbf{Y}_{k|k}^{1/2}]^{-1}\mathbf{Y}_{k|k}^{-1/2}$ is independent of $\vec{p}$ and solely depends on the structure properties of the network. Hence, following a similar definition in [155] for single line failures, we define this matrix as *k-line outage distribution matrix* and denote it by $\boldsymbol{\mathcal{L}} := \mathbf{Y}_{\bar{k}|\bar{k}}\mathbf{R}_{\bar{k}|k}\mathbf{Y}_{k|k}^{1/2}[\mathbf{I}-\mathbf{Y}_{k|k}^{1/2}\mathbf{R}_{k|k}\mathbf{Y}_{k|k}^{1/2}]^{-1}\mathbf{Y}_{k|k}^{-1/2}$. Hence, $\Delta\vec{f}_{\bar{k}} = \boldsymbol{\mathcal{L}}\vec{f}_k$.

While k-line outage distribution matrix captures all effects of a k-line failure on the flow changes, it is not efficient to compute and store it for contingency analysis in large power grids. In order to overcome this problem, we use the matrix of equivalent reactance values to efficiently compute the sum of changes in the power flows after k-line failures and to provide a metric to capture the essence of the flow changes after failures. The following lemma is the main step towards this goal. It demonstrates that $\mathbf{Y}^{1/2}\mathbf{R}\mathbf{Y}^{1/2}$ is an idempotent matrix. Recall that we

used a similar idea in the proof of Lemma 3.6. We use this property, to provide the results in Corollaries 4.1 and 4.2.

Lemma 4.3. $\boldsymbol{Y}^{1/2}\boldsymbol{R}\boldsymbol{Y}^{1/2} = \boldsymbol{Y}^{-1/2}\boldsymbol{D}^{+}\boldsymbol{D}\boldsymbol{Y}^{1/2}$, *and therefore* $(\boldsymbol{Y}^{1/2}\boldsymbol{R}\boldsymbol{Y}^{1/2})^2 = \boldsymbol{Y}^{1/2}\boldsymbol{R}\boldsymbol{Y}^{1/2}$.

Proof. We know from before that $\mathbf{R} = \mathbf{D}^t\mathbf{A}^{+}\mathbf{D}$ and $\mathbf{A} = \mathbf{D}\mathbf{Y}\mathbf{D}^t = (\mathbf{D}\mathbf{Y}^{1/2})(\mathbf{D}\mathbf{Y}^{1/2})^t$. Hence,

$$\begin{aligned}\mathbf{Y}^{1/2}\mathbf{R}\mathbf{Y}^{1/2} &= \mathbf{Y}^{1/2}\mathbf{D}^t\mathbf{A}^{+}\mathbf{D}\mathbf{Y}^{1/2} = (\mathbf{D}\mathbf{Y}^{1/2})^t\mathbf{A}^{+}(\mathbf{D}\mathbf{Y}^{1/2}) \\ &= (\mathbf{D}\mathbf{Y}^{1/2})^t\big((\mathbf{D}\mathbf{Y}^{1/2})(\mathbf{D}\mathbf{Y}^{1/2})^t\big)^{+}(\mathbf{D}\mathbf{Y}^{1/2}) \\ &= (\mathbf{D}\mathbf{Y}^{1/2})^t\big((\mathbf{D}\mathbf{Y}^{1/2})^t\big)^{+}(\mathbf{D}\mathbf{Y}^{1/2})^{+}(\mathbf{D}\mathbf{Y}^{1/2}) \\ &= \big((\mathbf{D}\mathbf{Y}^{1/2})^{+}(\mathbf{D}\mathbf{Y}^{1/2})\big)^t(\mathbf{D}\mathbf{Y}^{1/2})^{+}(\mathbf{D}\mathbf{Y}^{1/2}).\end{aligned}$$

From the properties of the pseudo-inverse, $(\mathbf{D}\mathbf{Y}^{1/2})^{+}(\mathbf{D}\mathbf{Y}^{1/2})$ is a symmetric matrix. Moreover, $(\mathbf{D}\mathbf{Y}^{1/2})^{+}(\mathbf{D}\mathbf{Y}^{1/2})(\mathbf{D}\mathbf{Y}^{1/2})^{+} = (\mathbf{D}\mathbf{Y}^{1/2})^{+}$. Therefore,

$$\begin{aligned}\mathbf{Y}^{1/2}\mathbf{R}\mathbf{Y}^{1/2} &= (\mathbf{D}\mathbf{Y}^{1/2})^{+}(\mathbf{D}\mathbf{Y}^{1/2})(\mathbf{D}\mathbf{Y}^{1/2})^{+}(\mathbf{D}\mathbf{Y}^{1/2}) \\ &= (\mathbf{D}\mathbf{Y}^{1/2})^{+}(\mathbf{D}\mathbf{Y}^{1/2}) = \mathbf{Y}^{-1/2}\mathbf{D}^{+}\mathbf{D}\mathbf{Y}^{1/2}.\end{aligned}$$

From this,

$$\begin{aligned}(\mathbf{Y}^{1/2}\mathbf{R}\mathbf{Y}^{1/2})^2 &= (\mathbf{Y}^{-1/2}\mathbf{D}^{+}\mathbf{D}\mathbf{Y}^{1/2})^2 \\ &= \mathbf{Y}^{-1/2}\mathbf{D}^{+}\mathbf{D}\mathbf{D}^{+}\mathbf{D}\mathbf{Y}^{1/2} \\ &= \mathbf{Y}^{-1/2}\mathbf{D}^{+}\mathbf{D}\mathbf{Y}^{1/2} = \mathbf{Y}^{1/2}\mathbf{R}\mathbf{Y}^{1/2}.\end{aligned}$$

■

Corollary 4.1. $\boldsymbol{R}_{k|\bar{k}}\Delta\vec{f}_{\bar{k}} = \boldsymbol{R}_{k|k}\vec{f}_{k}$.

Proof. To make equations cleaner in the proof, define $\mathbf{H} := \mathbf{Y}^{1/2}\mathbf{R}\mathbf{Y}^{1/2}$. From Lemma 4.3, $\mathbf{H}^2 = \mathbf{H}$. Hence, if we use block multiplication, then $\mathbf{H}_{k|k} = \mathbf{H}_{k|k}^2 + \mathbf{H}_{k|\bar{k}}\mathbf{H}_{\bar{k}|k}$. Using this

equation,

$$\begin{aligned}
&\Delta\vec{f}_{\bar{k}} = \mathbf{Y}_{\bar{k}|\bar{k}}\mathbf{R}_{\bar{k}|k}\mathbf{Y}_{k|k}^{1/2}[\mathbf{I}-\mathbf{Y}_{k|k}^{1/2}\mathbf{R}_{k|k}\mathbf{Y}_{k|k}^{1/2}]^{-1}\mathbf{Y}_{k|k}^{-1/2}\vec{f}_k\\
\Rightarrow&\mathbf{Y}_{k|k}^{1/2}\mathbf{R}_{k|\bar{k}}\Delta\vec{f}_{\bar{k}} = \mathbf{H}_{k|\bar{k}}\mathbf{H}_{\bar{k}|k}[\mathbf{I}-\mathbf{H}_{k|k}]^{-1}\mathbf{Y}_{k|k}^{-1/2}\vec{f}_k\\
\Rightarrow&\mathbf{Y}_{k|k}^{1/2}\mathbf{R}_{k|\bar{k}}\Delta\vec{f}_{\bar{k}} = \mathbf{H}_{k|k}\mathbf{Y}_{k|k}^{-1/2}\vec{f}_k\\
\Rightarrow&\mathbf{Y}_{k|k}^{1/2}\mathbf{R}_{k|\bar{k}}\Delta\vec{f}_{\bar{k}} = \mathbf{Y}_{k|k}^{1/2}\mathbf{R}_{k|k}\vec{f}_k \Rightarrow \mathbf{R}_{k|\bar{k}}\Delta\vec{f}_{\bar{k}} = \mathbf{R}_{k|k}\vec{f}_k.
\end{aligned}$$

■

Corollary 4.1 shows the use of matrix $\mathbf{R}$ in evaluating the effect of k-line failures without computing the flows directly. This equation can be used to estimate the effect of k-line failures. Since the matrix of equivalent reactance values needs to be computed only once, the matrix equation in Corollary 4.1 can be written for any k-line failures without further computations.

To quantify the effect of k-line failures more efficiently, in the following, we define a metric that captures the effect of k-line failures by a single value and show that it can be computed in $O(1)$. Inspired by the notion of energy in resistive networks, we define $\delta_k(1,2,\ldots,k) := \Delta\vec{f}_{\bar{k}}^{t}\mathbf{Y}_{\bar{k}|\bar{k}}^{-1}\Delta\vec{f}_{\bar{k}}$ as the *disturbance value* of a k-line failure. It is easy to see that $y_{ii}^{-1}\Delta f_i$ captures the changes in the phase angle differences between the end buses of a single line. Hence, the disturbance value $\Delta\vec{f}_{\bar{k}}^{t}\mathbf{Y}_{\bar{k}|\bar{k}}^{-1}\Delta\vec{f}_{\bar{k}} = \sum_{i=k+1}^{m} y_{ii}^{-1}\Delta f_i^2$ reflects both the big phase difference changes (which is important for the stability of the system) and the big flow changes (which is important for thermal safety of a line). Notice that the disturbance values can be defined also based on the phase angle of the nodes instead of the power flows using (2.6).

In the following lemma, we provide the key step in computing the disturbance value of a failure analytically and efficiently in Corollary 4.2.

For convenience in equations, define $\mathbf{C} := [\mathbf{I}-\mathbf{Y}_{k|k}^{1/2}\mathbf{R}_{k|k}\mathbf{Y}_{k|k}^{1/2}]^{-1}$ and $\mathbf{\Phi} := \mathbf{Y}_{\bar{k}|\bar{k}}^{1/2}\mathbf{R}_{\bar{k}|k}\mathbf{Y}_{k|k}^{1/2}[\mathbf{I}-\mathbf{Y}_{k|k}^{1/2}\mathbf{R}_{k|k}\mathbf{Y}_{k|k}^{1/2}]^{-1}$. From (4.2), we know $\mathbf{Y}_{\bar{k}|\bar{k}}^{-1/2}\Delta\vec{f}_{\bar{k}} = \mathbf{\Phi}\mathbf{Y}_{k|k}^{-1/2}\vec{f}_k$.

Lemma 4.4. $\mathbf{\Phi}^t\mathbf{\Phi} = -\boldsymbol{I} + \boldsymbol{C}$.

Proof. To make equations cleaner in the proof, define $\mathbf{H} := \mathbf{Y}^{1/2}\mathbf{R}\mathbf{Y}^{1/2}$. From Lemma 4.3, $\mathbf{H}^2 = \mathbf{H}$. Hence, if we use block multiplication, then $\mathbf{H}_{k|k} = \mathbf{H}_{k|k}^2 + \mathbf{H}_{k|\bar{k}}\mathbf{H}_{\bar{k}|k}$. Thus,

$$\begin{aligned}\mathbf{H}_{k|k} = \mathbf{H}_{k|k}^2 + \mathbf{H}_{k|\bar{k}}\mathbf{H}_{\bar{k}|k} &\Rightarrow \mathbf{H}_{k|k}[\mathbf{I} - \mathbf{H}_{k|k}] = \mathbf{H}_{k|\bar{k}}\mathbf{H}_{\bar{k}|k} \\ &\Rightarrow \mathbf{H}_{k|k} = \mathbf{H}_{k|\bar{k}}\mathbf{H}_{\bar{k}|k}[\mathbf{I} - \mathbf{H}_{k|k}]^{-1}.\end{aligned}$$

It is easy to see that $\mathbf{\Phi} = \mathbf{H}_{\bar{k}|k}[\mathbf{I} - \mathbf{H}_{k|k}]^{-1}$. Hence, using equation above,

$$\begin{aligned}\mathbf{\Phi}^t\mathbf{\Phi} &= [\mathbf{I} - \mathbf{H}_{k|k}]^{-1}\mathbf{H}_{k|\bar{k}}\mathbf{H}_{\bar{k}|k}[\mathbf{I} - \mathbf{H}_{k|k}]^{-1} \\ &= [\mathbf{I} - \mathbf{H}_{k|k}]^{-1}\mathbf{H}_{k|k} \\ &= [\mathbf{I} - \mathbf{H}_{k|k}]^{-1}\mathbf{H}_{k|k} - [\mathbf{I} - \mathbf{H}_{k|k}]^{-1} + [\mathbf{I} - \mathbf{H}_{k|k}]^{-1} \\ &= -[\mathbf{I} - \mathbf{H}_{k|k}]^{-1}[-\mathbf{H}_{k|k} + \mathbf{I}] + [\mathbf{I} - \mathbf{H}_{k|k}]^{-1} \\ &= -\mathbf{I} + [\mathbf{I} - \mathbf{H}_{k|k}]^{-1} = -\mathbf{I} + \mathbf{C}.\end{aligned}$$

■

Corollary 4.2. $\Delta\vec{f}_{\bar{k}}^{\,t}\,\boldsymbol{Y}_{\bar{k}|\bar{k}}^{-1}\Delta\vec{f}_{\bar{k}} = \vec{f}_k^{\,t}\,\boldsymbol{Y}_{k|k}^{-1/2}[-\boldsymbol{I} + \boldsymbol{C}]\,\boldsymbol{Y}_{k|k}^{-1/2}\,\vec{f}_k$.

Corollary 4.2 provides a very important tool for contingency analysis in the power grids. It shows that the disturbance value of a k-line failure can be computed in $O(k^3)$ time which is independent of the size of the network and only depends on the size of the initial failures. Since most of the times in contingency analysis $k << m$, $O(k^3) \approx O(1)$. Corollary 4.2 can be used for fast ranking of the contingencies based on the disturbance values and pruning most of the cases based on this value. This can significantly reduce the time complexity of the contingency analysis for large k.

Moreover, notice that following Lemma 4.1, matrix $\mathbf{C}$ is well defined if, and only if, the k-line failure does not disconnect the grid. Therefore, Corollary 4.2 can also indicate in $O(1)$ whether a k-line failure disconnects the grid.

We computed the disturbance values for all possible choices of 3-line failures in the IEEE 118-bus and 300-bus systems. Since, the IEEE 118-bus system has 186 lines, it is easy to see that there are $\binom{186}{3} = 1,055,240$ possible choices for the initial set of failures. Using Corollary 4.2,

we could compute the disturbance values for all set of failures (and detect cases that disconnect the grid) in less than a minute. Out of those, 159, 591 of them make the graph disconnected. The cumulative distribution function of the disturbance values for the rest of 895,649 cases are shown in Fig. 4.1. As can be seen, most of the cases do not result in a high (more than 5,000) disturbance value. Only 10% of the cases have a significant disturbance value. The figure suggests that disturbance values can provide a separation between the failures with higher impact and lower impact.

Fig. 4.1 also shows the cumulative distribution function of the disturbance values for all possible 3-line failures in IEEE 300-bus system that do not disconnect the grid (5,473,725 cases). As can be seen, again most of the cases do not result in a high disturbance value and only less than 10% of the total cases have a significant disturbance value (more than 35,000).[2]

The numerical results support our previous statement that the disturbance values can significantly reduce the total number of cases that are needed to be considered in the contingency analysis of the grids. Moreover, since Corollary 4.2 provides a very fast way of computing the disturbance values, it can significantly decrease the time complexity of the contingency analysis in power grids.

4.4 Approximating the Disturbance Values

In the previous section, we demonstrated that using Corollary 4.2, the disturbance value of a k-line failure can be computed efficiently in $O(1)$. However, to analyze all the k-line failure scenarios, one needs to compute the disturbance values for all possible $\binom{m}{k}$ cases which requires $O(m^k)$ time. To alleviate this issue, in this section we provide an approximation for the disturbance values that allows one to focus on the lines with high 1-line disturbance values instead of all the lines.

[2]Notice that the threshold level for "high disturbance value" depends on the level of safety that a system operator likes to maintain. As can be seen in Fig. 4.1, more than 80% of the cases result in a very small disturbance value. Hence, 1% to 20% of the cases with the highest disturbance values can be selected for deeper analysis depending on the level of safety.

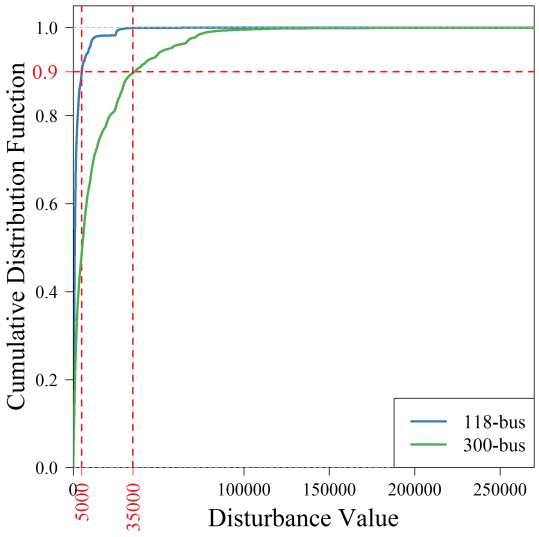


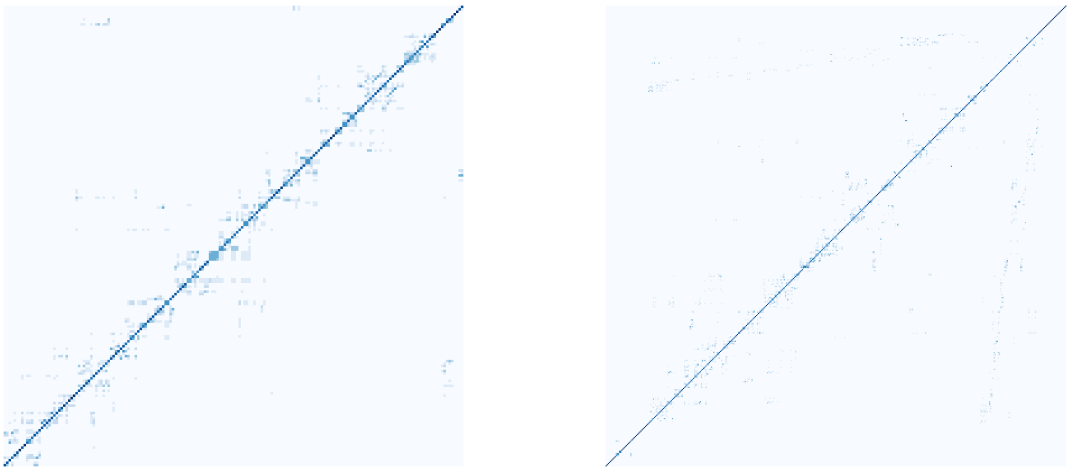
Figure 4.1: The cumulative distribution function of the disturbance values for all 3-line failures in the IEEE 118- and 300-bus systems that do not disconnect the grid.

The idea of our approximation lies in the structure of the matrix $\mathbf{R}$. As shown in Fig. 4.2, most of the entries of matrix $\mathbf{R}$ in the IEEE 118- and 300-bus systems have very small values compare to its diagonal entries. Hence, we can approximate $\mathbf{R}$ by its diagonal entries as $\mathbf{R} \approx \text{diag}(\mathbf{R}) := \widetilde{\mathbf{R}}$. Using this approximation and Corollary 4.2, the disturbance values can be computed as:

$$\begin{aligned}
\Delta\vec{f}_{\bar{k}}^{t}\mathbf{Y}_{\bar{k}|\bar{k}}^{-1}\Delta\vec{f}_{\bar{k}} &= \vec{f}_{k}^{t}\mathbf{Y}_{k|k}^{-1/2}[-\mathbf{I}+[\mathbf{I}-\mathbf{Y}_{k|k}^{1/2}\mathbf{R}_{k|k}\mathbf{Y}_{k|k}^{1/2}]^{-1}]\mathbf{Y}_{k|k}^{-1/2}\vec{f}_{k} \\
&\approx \vec{f}_{k}^{t}\mathbf{Y}_{k|k}^{-1/2}[-\mathbf{I}+[\mathbf{I}-\mathbf{Y}_{k|k}^{1/2}\widetilde{\mathbf{R}}_{k|k}\mathbf{Y}_{k|k}^{1/2}]^{-1}]\mathbf{Y}_{k|k}^{-1/2}\vec{f}_{k} \\
&= \sum_{i=1}^{k} f_i y_{ii}^{-1/2}(-1+\frac{1}{(1-y_{ii}^{1/2}r_{ii}y_{ii}^{1/2})})y_{ii}^{-1/2}f_i \\
&= \sum_{i=1}^{k} f_i^2\frac{r_{ii}}{(1-y_{ii}r_{ii})}.
\end{aligned}$$

In another words, we approximate the disturbance value of a k-line failure as the sum of the disturbance values of the k single line failures:

$$\delta_k(1,2,\ldots,k) \approx \sum_{i=1}^{k}\delta_1(i). \tag{4.3}$$



(a) IEEE 118-bus (b) IEEE 300-bus

Figure 4.2: Heatmap for the absolute value of the entries of matrix $\mathbf{R}$ for the IEEE 118- and 300-bus systems. Darker points have higher values.

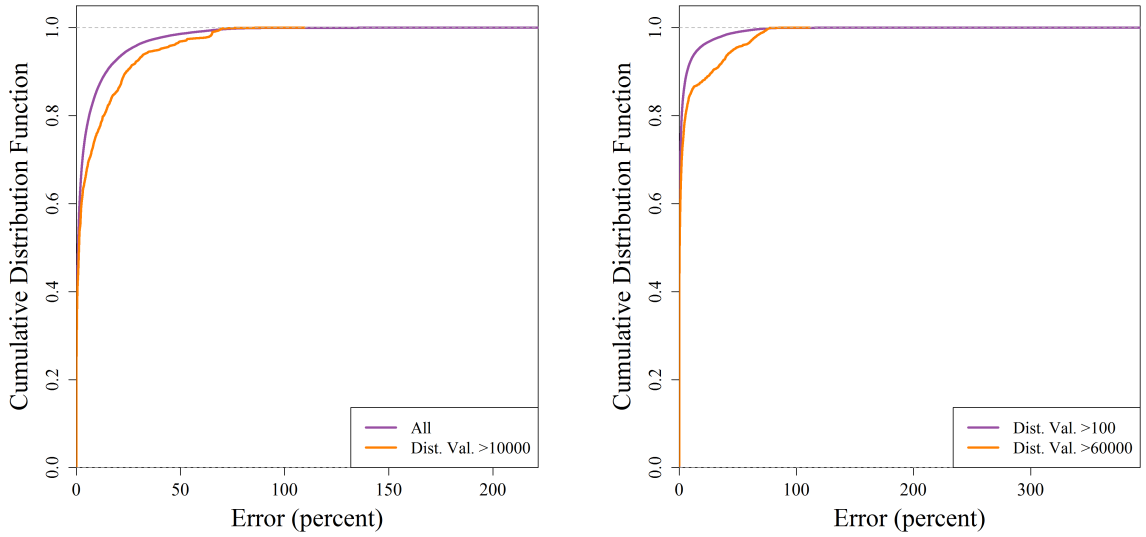
The advantage of this approximation is that to detect k-line failures with high disturbance values, one can focus only on the $\widetilde{m} << m$ lines with the highest 1-line disturbance values. So instead of analyzing $\binom{m}{k}$ cases, one can analyze only $\binom{\widetilde{m}}{k}$ cases.

Another advantage of using the approximation (4.3) for disturbance values is that it can used to quantify the effect of k-line failures that disconnect the grid as well. Although failures that disconnect the grid may have more devastating effect because of the imbalance they may cause between supply and demand, we believe that this approximation can be used to rank these failures among themselves. Validating the usefulness of approximation (4.3) for the failures that disconnect the grid is part of the future work.

To quantify the quality of the approximation (4.3), we define the approximation error percentage (ε):

$$\varepsilon = \frac{\left|\delta_k(i_1, i_2, \ldots, i_k) - \sum_{j=1}^{k} \delta_1(i_j)\right|}{\delta_k(i_1, i_2, \ldots, i_k)} \times 100.$$

Fig. 4.3 shows the approximation error in the IEEE 118- and 300-bus systems. As can be seen, in both systems, for most of the cases the error is less than 10%. In particular, the



(a) IEEE 118-bus

(b) IEEE 300-bus

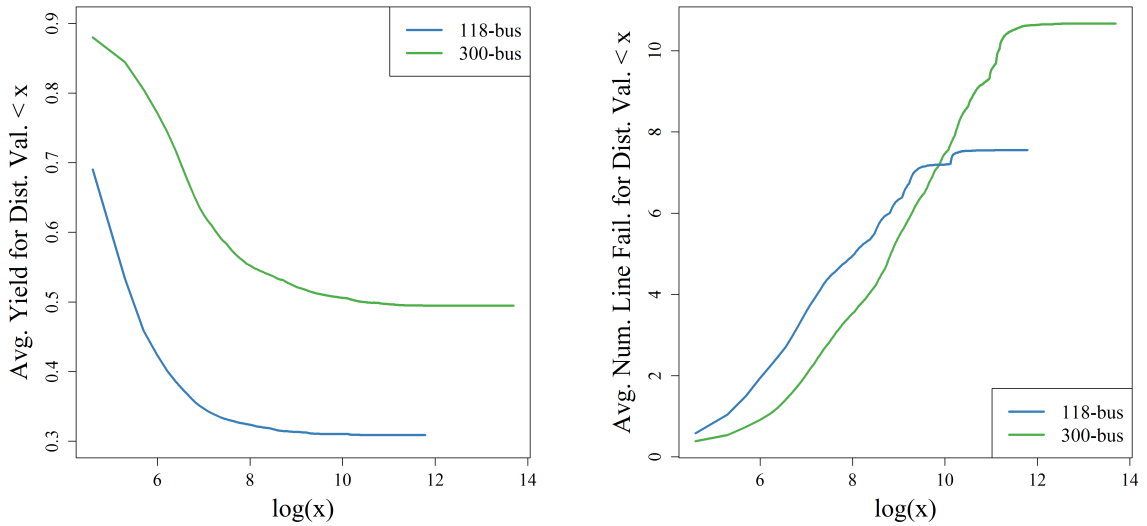
Figure 4.3: Approximation error percentage of the approximate disturbance values for all the 3-line failures than do not disconnect the grid in the IEEE 118- and 300-bus systems.

approximation error in the important cases (the ones with high disturbance value) is very low (see orange lines in Fig. 4.3). Moreover, the correlation between the approximation and actual disturbance values is about 0.98 in both the IEEE 118- and 300-bus systems.

4.5 Cascading Failures

To show that disturbance values effectively rank k-line failures, we numerically compute the relationship between disturbance value of all 3-line failures and the severity of the initiated cascades in the IEEE 118- and 300-bus systems.

We for cascade model as described in Subsection 3.2.1 with *deterministic outage rule*: namely, a line $e = \{i, j\}$ fails when the magnitude $|f_e|$ of the flow on that line exceeds its capacity c_e. The line flow capacities are estimated as $c_e = (1 + \alpha) \max\{|f_{ij}|, \overline{f}\}$, where $\overline{f}$ is the *median* of the initial magnitude of line flows and $\alpha = 0.2$ is the lines' *factor of safety*. To measure the severity



(a) Yield

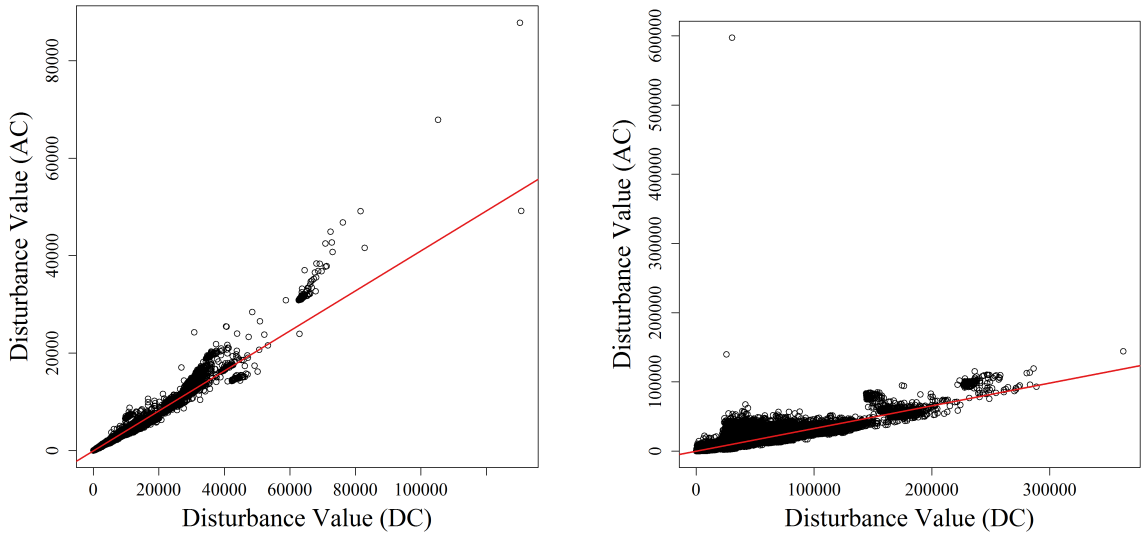
(b) Number of Line Failures

Figure 4.4: The relationship between the disturbance values and the average severity of cascades initiated by all the 3-line failures than do not disconnect the grid in the IEEE 118- and 300-bus systems.

of a cascade, we compute *yield (the ratio between the demand supplied at the end of a cascade and the original demand) and total number of line failures* at the end of the cascade.

Fig. 4.4 depicts the relationship between the disturbance values and the average severity of cascades initiated by all the 3-line failures than do not disconnect the grid in the IEEE 118- and 300-bus systems. As can been seen in Fig. 4.4a, in both systems, the average yield decreases as the disturbance value increases. Moreover, Fig. 4.4b shows that the average number of line failures at the end of the cascade increases on average, as the disturbance value increases.

The results in this section suggest that although the disturbance value does not take into account the capacities of the lines, it can predict the average severity of the cascade initiated by a failure.



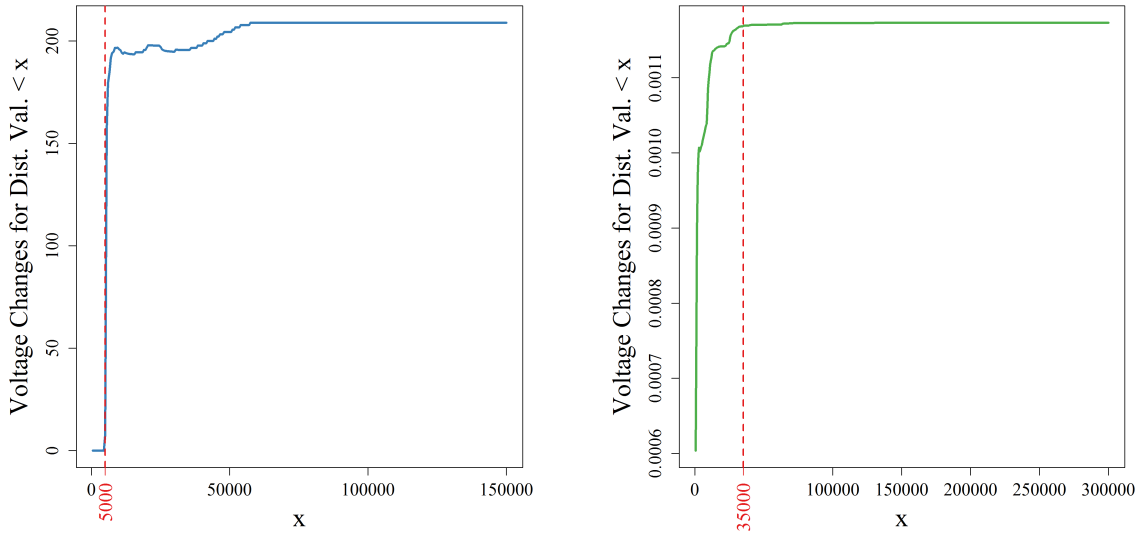
(a) IEEE 118-bus (b) IEEE 300-bus

Figure 4.5: Scatter plot of the disturbance values under the AC and DC power flows for all the 3-line failures that do not disconnect the grid in the IEEE 118- and 300-bus systems.

4.6 Disturbance Values under the AC Power Flow Model

In this section, we numerically show the usefulness of the disturbance values in predicting the changes under the more detailed AC power flows after a k-line failure. To solve the AC power flows, we used the MATPOWER AC power flow solver [164].

We first computed the disturbance values under the AC power flows. To do so, we used the phase angles of the nodes computed before and after a k-line failure under the AC model (recall that the disturbance values can be defined also based on the phase angle). Fig. 4.5 shows the relationship between the disturbance values computed under DC (using corollary 4.2) and AC power flows. As can be seen, the disturbance values under the AC and DC are only different in scale. The correlation between these two values is 0.99 and 0.93 in the IEEE 118- and 300-bus systems, respectively. Hence, detecting the important cases based on the disturbance values in the DC is almost similar to the AC model.



(a) IEEE 118-bus (b) IEEE 300-bus

Figure 4.6: Mean sum squared of the voltage changes for the 3-line failures that do not disconnect the grid with the disturbance value less than a certain value in the IEEE 118- and 300-bus systems.

Second, we studied the correlation between the voltage changes after a k-line failure and disturbance values. For this purpose, we computed the mean sum squared of the voltage changes after all 3-line failures that do not disconnect the grid in the IEEE 118- and 300-bus systems. As can be seen in Fig. 4.6, on average, 3-line failures with higher disturbance values cause more voltage changes as well. This demonstrates that the disturbance values which can be computed very efficiently using corollary 4.2, can also predict the disturbances in the node voltages whose computation is of higher complexity.

Finally, since the AC power flow equations are nonlinear (see Appendix 2.2), there are 3-line failures after which the AC power solver does not converge to a solution. The CDF of the disturbance values for these 3-line failures is depicted in Fig. 4.7. As can be seen, for all such failures in the IEEE 118-bus system the disturbance values are above 5000, meaning that they are between the top 10% of all the 3-line failures in terms of the disturbance values (see Fig. 4.1).

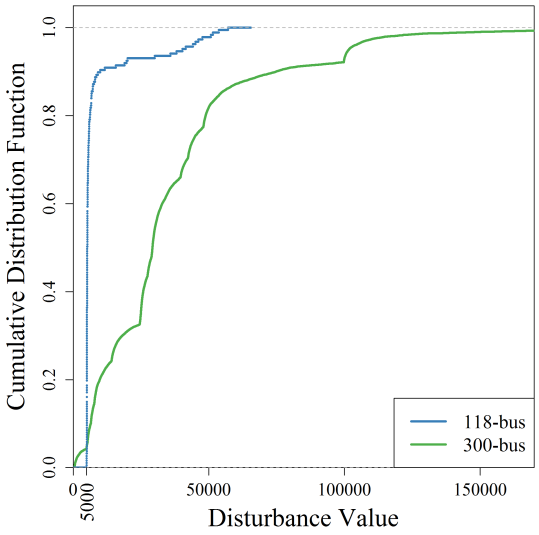


Figure 4.7: The CDF of the disturbance values for the 3-line failures that do not disconnect the grid but produce enough instability such that the AC power solver does not converge to a solution.

The same is true in the IEEE 300-bus system. Most of the 3-line failures that result in the divergence of the AC power flow solver, are between the top 10-20% cases in terms of the disturbance values.

Overall, the results in this section demonstrate that the disturbance values can be useful for predicting the changes under the AC power flow model. Since following corollary 4.2 the disturbance values can be computed much faster compared to the computation of steady state changes under the AC power flows (which requires solving a nonlinear set of equations several times), this significantly reduces the computational complexity associated with the contingency analysis in the grid.

4.7 Conclusion

The results in this chapter provide efficient tools for quantifying the effect of k-line failures. The most unique aspect of our approach is the use of the matrix of equivalent reactance values to efficiently capture the effect of k-line failures. We defined the disturbance value of a failure and

show that this metric can be computed for any set of failures in $O(1)$. Moreover, based on the approximation of the disturbance values, the total number of cases that need to be considered can be significantly reduced as well. Our numerical results showed that disturbance values provide a clear separation between the failures with higher impact and lower impact under both the AC and DC power flows.

Despite providing a fast and useful measure for quantifying the severity of a k-line failure, the disturbance value does not capture the thermal capacity of the lines as well as further instabilities caused by separation of the grid into several islands. Hence, the disturbance values alone cannot be used to detect important contingencies. Since thorough contingency analysis for all high order scenarios is intractable, however, disturbance values can be used to reduce the total number of cases needed to be analyzed in contingency analysis and can significantly reduce the computational complexity associated with this analysis.

Chapter 5

Cyber-physical Attacks (DC Model)

Power grids are comprised of two components: (i) the physical infrastructure of the power transmission system (power lines, substations, power stations), and (ii) the Supervisory Control and Data Acquisition (SCADA) system that monitors and controls the grid (the control network) (Fig. 5.1). The physical infrastructure is the target of physical attacks and SCADA is the target of cyber attacks.

In the case of a physical attack, the system's stability can be maintained if SCADA receives precise information about the location of the attack and takes proper action accordingly. If however, the flow of information is obstructed by a cyber attack, the SCADA is prevented from taking necessary and appropriate actions. This problem, the joint cyber and physical attacks on power grids, is the focus of our work. We develop methods to *estimate the state of the power grid following a joint cyber and physical attack, and study the resilience of different topologies as well as the resilience to different kinds of attacks.*

We use the DC power flow model to describe the state of the grid. We also use a modified version of the *control network model* [85] that includes Phasor Measurement Units (PMU), Phasor Data Concentrators (PDC), and a control center (Fig. 5.1). We define a *zone* as a set of buses (nodes), power lines (edges), PMUs, and an associated PDC. We analyze an attack that disconnects lines within a zone (physical attack) and obstructs the flow of information from the PMUs within the zone to the control center (cyber attack). For example, an adversary can

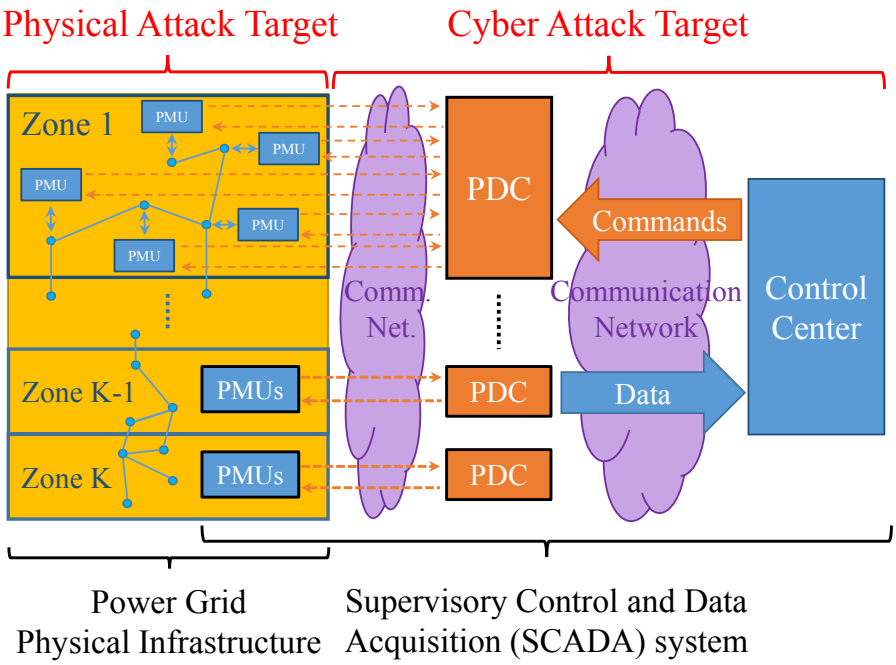


Figure 5.1: Components of the power grid and potential attacks: physical attacks target the physical infrastructure (lines, substations, etc.); Cyber attacks target the SCADA system – an adversary can obstruct the flow of information from the PMUs within the zone to the control center.

perform the cyber attack by disabling the zone's associated PDC. Alternatively, the adversary can attack the communication network between the PMUs and the PDC, or between the PDC and the control center. Because our control network model is a generic model of SCADA that monitors the status of the grid, most of the results and methods provided in this chapter can be interpreted and used for more complicated control systems and scenarios.

As a result of an attack, some lines get disconnected, and the phase angles and the status of the lines within the *attacked zone* $H = (V_H, E_H)$ become unavailable (Fig. 5.2). Our objective is to recover the phase angles and detect the disconnected lines using the information available outside of the attacked zone.

Recall from Chapter 2 that power flows are governed by the laws of physics, where a line failure results in changes to flows and node phase angles throughout the power grid. We use

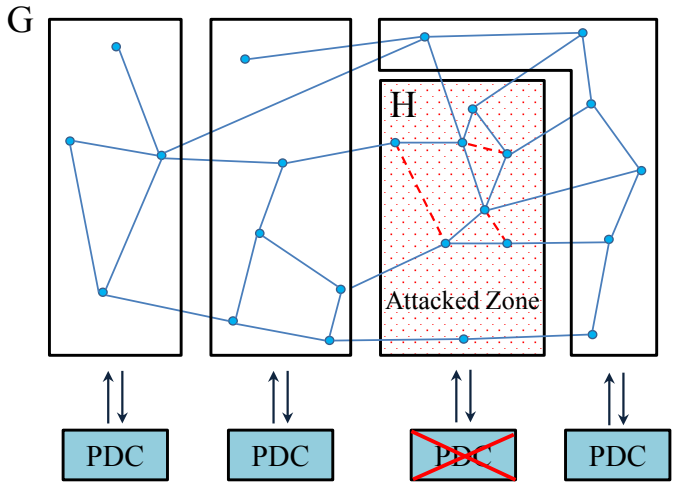


Figure 5.2: The attack model. G is the power grid graph and H is an induced subgraph of G that represents the attacked zone. An adversary attacks a zone by disconnecting some of its power lines (red dashed lines) and disallowing the information from the PMUs within the zone to reach the control center.

this property and show that it is possible to estimate the state in the attacked zone using the information available outside of the zone. Specifically, we *develop methods for retrieving information from the attacked zone* by applying matrix analysis and graph theoretical tools to the matrix representation of the DC equations.

First, we prove that it is NP-hard to detect the set of line failures given the phase angle of the nodes before and after the attack in general case. However, we present *necessary and sufficient conditions on the structure of a zone such that our methods are guaranteed to recover the state of the grid inside the attacked zone.* We prove that if there is a *matching* between the nodes inside and outside the attacked zone that covers the inside nodes (V_H), then the phase angles of the nodes in the attacked zone are recoverable by solving a set of linear equations of size $|V_H|$. We also prove that if H is *acyclic*, the disconnected lines in H are detectable by solving a set of linear equations of size $|E_H|$. Moreover, we show that if H is *planar*, under some constraints, the disconnected lines are detectable by solving a Linear Programming (LP) problem.

We develop another method for simultaneous recovery of phase angles and detection of disconnected lines by solving a single LP problem. We show that this method is guaranteed to recover the information under certain constraints on the attack (i.e., on the disconnected lines) if there is a *partial matching* between the nodes inside and outside of H, and if H is *planar*. Based on these results, we present the Post-Attack Recovery and Detection (PARD) Algorithm. We propose that our methods can be generalized to the case where multiple zones are attacked simultaneously. We show that if the attacked zones are relatively distant from each other, any of the methods provided in this chapter can be applied to recover the information and detect the failures in the attacked zones.

We briefly study the problem of information recovery in the presence of measurement noise. By relaxing some of the constraints introduced in developing the methods used in the PARD Algorithm, we provide a method for information recovery in the noisy scenarios as well. We numerically evaluate the performance of the method and show that if the Signal to Noise Ratio (SNR) is high enough, it can successfully recover the information.

We study the problem of partitioning power grids into the minimum number of attack-resilient zones (i.e., zones in which the information can be recovered by the methods mentioned above). We show that this problem is not approximable to within $n^{1-\epsilon}$ for all $\epsilon > 0$, unless P=NP. However, since power grids are often represented by planar graphs, we introduce our Zone Selection (ZS) Algorithm and demonstrate that the ZS Algorithm provides a constant approximation ratio for these graphs. We present numerical results to demonstrate the operation of the ZS Algorithm on several power grids. This algorithm can also be used for designing a secure control network for smart grids. Finally, we present numerical results to assess the relationship between the structural properties of the power grid and its resilience to joint attacks.

5.1 Related Work

The vulnerability of general networks to attacks has been studied extensively (e.g., [16, 90, 112] and references therein). In particular, attacks and failures in power grids has been studied using probabilistic failure propagation models (e.g., [38, 42, 156], and references therein) as well as

using deterministic DC power flows [27, 30, 62, 96]. Malicious data attacks on the power grid control network have also been studied [55, 89, 98, 148]. To the best of our knowledge however, no previous work has focused on vulnerability of power grids to joint cyber and physical attacks.

In Section 5.4, we study the problem of recovering the phase angles and detecting disconnected lines after a joint cyber and physical attack, a problem related to line outage identification from changes in phase angles [73,142,143]. These studies however, were based on complete knowledge of phase angle measurements and in the case of [142,143] were limited to two line failures. The problem of line failure identification in an internal system using the information from an external system was studied in [163], where a heuristic algorithm was proposed for only one and two line failures.

In Section 5.7, we disucss the problem of partitioning the power grid into the minimum number of attack-resilient zones. This problem is similar to PMU placement problems [88,102, 161]. Recently, PMU placemnet problem has attracted much attention in India after the major blackouts of 2013 [88]. In [161] the problem of PMU placement for line outage detection was studied. However, none of these previous works addressed the problem of PMU placement from the security point of view where both the PDC/PMUs and the physical network are under attack.

In Section 5.7, we reduce the attack-resilient zone partitioning problem to the problem of partitioning a graph into subgraphs where each subgraph is (i) acyclic, and (ii) there is a matching between nodes inside and outside the subgraph that covers all the subgraph nodes. This problem is closely related to the problems of *vertex arboricity* (which is known to be NP-hard to be determined [74, p.193]) and *k-matching cover* of a graph (which can be found in $O(n^3)$ time [150]). However, to the best of our knowledge, the joint problem ((i) and (ii) above) was not studied before.

5.2 Model and Preliminaries

In this Chapter, we use the DC power flows as described in Section 2.3.

5.2.1 Control Network

We use a modified version of the model described in [85] to model the SCADA system to which we refer as the control network. Fig. 5.1 illustrates the components of the control network. We assume that there is a Phasor Measurement Unit (PMU) at each node of G. The PMU at node i reports the phase angle θ_i as well as the status of the lines (either *operational* or *failed*) adjacent to node i. Phasor Data Concentrators (PDC) gather the data collected by PMUs. The data gathered by PDCs is sent to a control center which monitors and controls the entire grid. A *zone* is a subgraph induced by a subset of nodes with a single associated PDC.

5.2.2 Attack Model

We study attacks on power grids that affect both the physical infrastructure and the control network. We assume that an adversary attacks a zone by: (i) disconnecting some edges within the attacked zone (physical attack), and (ii) obstructing the flow of information from the PMUs within the zone to the control center (cyber attack). An adversary can perform the cyber attack by, for example, disabling the zone's associated PDC. Alternatively, the communication network between the PMUs and the PDC or between the PDC and the control center can be attacked. We assume that disconnecting edges within a zone does not make G disconnected.

Fig. 5.2 shows an example of an attack on the zone represented by H. Due to the attack, some edges are disconnected (we refer to these edges as *failed lines*) and the phase angles and the status of the lines within the *attacked zone* become unavailable. We denote the set of failed lines in zone H by $F \subseteq E_H$. Upon failure, the failed lines are removed from the graph and the flows are redistributed according to (2.5)-(2.6).

Notation. Throughout this and next chapter, we denote an attacked zone by $H = (V_H, E_H)$. Without loss of generality we assume that the indices are such that $V_H = \{1, 2, \ldots, |V_H|\}$ and $E_H = \{e_1, e_2, \ldots, e_{|E_H|}\}$. We denote the complement of the zone H by $\bar{H} = G \backslash H$. If X, Y are two subgraphs of G, $\mathbf{A}_{X|Y}$ and $\mathbf{A}_{V_X|V_Y}$ both denote the submatrix of the admittance matrix of G with rows from V_X and columns from V_Y. For instance, $\mathbf{A}$ can be written in any of the

following forms,

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{H|H} & \mathbf{A}_{H|\bar{H}} \\ \mathbf{A}_{\bar{H}|H} & \mathbf{A}_{\bar{H}|\bar{H}} \end{bmatrix}, \mathbf{A} = \begin{bmatrix} \mathbf{A}_{G|H} & \mathbf{A}_{G|\bar{H}} \end{bmatrix}, \mathbf{A} = \begin{bmatrix} \mathbf{A}_{H|G} \\ \mathbf{A}_{\bar{H}|G} \end{bmatrix}.$$

We use the very same notation for the vectors. For instance $\vec{\theta}_H$ and $\vec{\theta}_{\bar{H}}$ are the vectors of phase angle of the nodes in H and $\bar{H}$, respectively. We use the prime symbol ($'$) to denote the values after an attack. For instance, G', $\mathbf{A}'$, and $\vec{\theta}'$ are used to represent the graph, the admittance matrix of the graph, and the phase angles of the nodes after an attack. $\mathbf{D}_H \in \{-1, 0, 1\}^{|V_H| \times |E_H|}$ is the submatrix of $\mathbf{D}$ as defined in Subsection 2.3.1, with rows from V_H and columns from E_H.

5.2.3 Graph Theoretical Terms

In this chapter, we use several graph theoretical terms and theorems mostly borrowed from [36]. We briefly review some of the important definitions in this subsection.

Subgraphs, Cuts, and Cycles: Let X and Y be subsets of the nodes of a graph G. $G[X]$ denotes the subgraph of G *induced* by X. We denote by $E[X, Y]$ the set of edges of G with one end in X and the other end in Y. We denote the complement of a set X by $\bar{X} = V \backslash X$. The *coboundary* of X is the set $E[X, \bar{X}]$ and is denoted by $\partial(X)$. $\partial(v)$ denotes the coboundary of $X = \{v\}$. $G[X, \bar{X}]$ denotes the subgraph of G induced by the edges from $E[X, \bar{X}]$. $N(X)$ is the set of neighbors of the nodes in X excluding X itself, and $N_c(X) = X \cup N(X)$. We say that $Q \subseteq E$ is *G-separable*, if there are pairwise edge-disjoint cycles $C_q (q \in Q)$, such that $\forall q \in Q, q \in C_q$ [123].

Planar Graphs: A graph G is *planar*, if it can be drawn in the plane so that its edges intersect only at their ends. A planar graph G partitions the rest of the plane into a number of edgewise-connected open sets called the *faces* of G.

Given a planar graph G, its dual graph G^* is defined as follows. Corresponding to each face c of G there is a node c^* of G^*, and corresponding to each edge e of G there is an edge e^* of G^*. Two nodes c_1^* and c_2^* are joined by the edge e^* in G^*, if and only if their corresponding faces c_1 and c_2 are separated by the edge e in G. It is easy to see that the dual G^* of a planar graph G is itself a planar graph [36].

5.3 Hardness

Using the notation provided in the previous section, the problem can be stated as follows: Given $\mathbf{A}, \vec{\theta}, H$, and $\vec{\theta}'_{\bar{H}}$, recover phase angles $\vec{\theta}'_H$ and detect the line failures F. In this section, we study the computational complexity of this and related problems.

First, we prove that the problem of finding the set of line failures given the phase angle of the nodes before and after the attack is NP-hard. We prove this by reduction from the *3-partition problem*.

Definition 5.1. *Given set* $S = \{s_1, s_2, \ldots, s_{3k}\}$ *of* $3k$ *elements and a bound* B*, such that* $\sum_{i=1}^{3k} s_i = kB$ *and for* $1 \leqslant i \leqslant 3k$*,* $B/4 < s_i < B/2$*, the* 3-partition problem *is the problem of whether* S *can be partitioned into* k *disjoint sets* $S_1, \ldots, S_k$ *such that for* $1 \leqslant i \leqslant k$*,* $\sum_{s_j \in S_i} s_j = B$ *(note that each* S_i *must therefore contain exactly 3 elements from* S*).*

Lemma 5.1 (Garey and Johnson, 1975 [74])**.** *The 3-partition problem is strongly NP-complete.*

Lemma 5.2. *Given* $\boldsymbol{A}, \vec{\theta}$*, and* $\vec{\theta}'$*, it is strongly NP-hard to determine the set of line failures* F*.*

Proof. We reduce the 3-partition problem to this problem. Assume S is a given set as described in Def. 5.1, we form a bipartite graph $G = (V, E)$ such that $V = X \cup Y$, $E = \{\{x, y\} | x \in X, y \in Y\}$, $X = \{1, \ldots, k\}$, and $Y = \{k+1, \ldots, 4k\}$. For all edges in G, we set the reactance values equal to 1. For each $i \in X$, we set $p_i = B$ and for each $j \in Y$ we set $p_j = -s_{j-k}$. Define the vector of phase angles $\vec{\theta}$ as follows:

$$\theta_i = \begin{cases} 0 & i \leqslant k \\ -s_{i-k}/k & i > k. \end{cases}$$

If $\mathbf{A}$ is the admittance matrix of G, it is easy to check that $\mathbf{A}\vec{\theta} = \vec{p}$. Now define $\vec{\theta}'$ as follows:

$$\theta'_i = \begin{cases} 0 & i \leqslant k \\ -s_{i-k} & i > k. \end{cases}$$

We prove that there exist a set of line failures F such that $\mathbf{A}'\vec{\theta}' = \vec{p}$ if, and only if, there exist a solution to the 3-partition problem.

First, lets assume that there exist a solution to the 3-partition problem such as $S_1, \ldots, S_k$. Set $E_S = \{\{i,j\}|s_{j-k} \in S_i\}$. We show that $F = E \backslash E_S$ implies $\mathbf{A}'\vec{\theta}' = \vec{p}$. Notice that $F = E \backslash E_S$ means that $G' = (V, E_S)$. Given the p_i and the reactance values, it is easy to check that the defined $\vec{\theta}'$ satisfies the DC power flow equations (2.5)-(2.6) in G'. Hence, $\mathbf{A}'\vec{\theta}' = \vec{p}$.

Now, lets assume there exist a set of line failures F such that $\mathbf{A}'\vec{\theta}' = \vec{p}$. Set $E_S = E \backslash F$ and $G' = (V, E_S)$. Given the phase angles $\vec{\theta}'$, it is easy to see that for any $\{i,j\} \in E_S$, $p_{ij} = s_{j-k}$. This implies that for $j \in Y$, at most one edge in E_S is incident to j. On the other hand, using (2.5), for any $i \in X$, $\sum_{j \in N(i)'} s_{j-k} = B$ in which by $N(i)'$ we mean the set of neighbors of node i in G'. Given that each node $j \in Y$ is incident to at most one edge in E_S, defining $S_i = \{s_{j-k} | j \in N(i)'\}$ for $1 \leqslant i \leqslant k$ gives a good solution to the 3-partition problem.

Hence the problem of finding the set of line failures from changes in the phase angle measurements is at least as hard as finding a solution to the 3-partition problem and therefore it is an NP-hard problem in the strong sense. ■

In Lemma 5.2, we proved that given the phase angle of the nodes before and after the attack, it is NP-hard to detect the set of line failures F. In the following lemma, we show that even if the attack area H is known (since $\vec{\theta}'_H$ is not given) the problem remains NP-hard.

Lemma 5.3. *Given $\boldsymbol{A}, \vec{\theta}$, H, and $\vec{\theta}'_{\bar{H}}$, it is strongly NP-hard to determine the set of line failures F.*

Proof. The idea of the proof is very similar to the proof of Lemma 5.2. Again we reduce the 3-partition problem with a given set S as described in Def. 5.1 to this problem. Consider sets $X_1 = \{1, \ldots, k\}$, $X_2 = \{k+1, \ldots, 2k\}$, $Y_2 = \{2k+1, \ldots, 5k\}$, $Y_1 = \{5k+1, \ldots, 8k\}$. We form a bipartite graph $G = (V, E)$ such that $V = X_1 \cup X_2 \cup Y_2 \cup Y_1$ and $E = \{\{i, k+i\} | 1 \leqslant i \leqslant k\} \cup \{\{x,y\} | x \in X_2, y \in Y_2\} \cup \{\{j, j+3k\} | 2k+1 \leqslant j \leqslant 5k\}$. Notice that the defined bipartite graph here is very similar to the one defined in the proof of Lemma 5.2 except that here for each node in X_2 and Y_2 there exist a dummy node in X_1 and Y_1, accordingly, that is directly connected to its counterpart. We set $H = G[X_2 \cup Y_2]$, it is easy that H has exactly the same topology as the graph G in the proof of Lemma 5.2. Again for all edges in G, we set the reactance values equal to 1. For each $i \in X_2 \cup Y_2$ we set $p_i = 0$, for each $i \in X_1$, we set $p_i = B$, and for

each $j \in Y_1$ we set $p_j = -s_{j-5k}$. Define the vector of phase angles $\vec{\theta}$ as follows:

$$\theta_i = \begin{cases} B & 1 \leqslant i \leqslant k \\ 0 & k+1 \leqslant i \leqslant 2k \\ -s_{i-2k}/k & 2k+1 \leqslant i \leqslant 5k \\ -s_{i-5k}/k - s_{i-5k} & 5k+1 \leqslant i \leqslant 8k \end{cases}$$

If $\mathbf{A}$ is the admittance matrix of G, it is easy to check that $\mathbf{A}\vec{\theta} = \vec{p}$. Now define $\vec{\theta'}$ as follows:

$$\theta'_i = \begin{cases} B & 1 \leqslant i \leqslant k \\ 0 & k+1 \leqslant i \leqslant 2k \\ -s_{i-2k} & 2k+1 \leqslant i \leqslant 5k \\ -2s_{i-5k} & 5k+1 \leqslant i \leqslant 8k \end{cases}$$

Now given $\vec{\theta'}_{\bar{H}}$, we later show in Section 5.4 that $\vec{\theta'}_H$ can be uniquely recovered by solving a set of linear equations in polynomial time. So without loss of generality, for now we can assume that $\vec{\theta'}$ is given. We prove that there exist a set of line failures F in H such that $\mathbf{A}'\vec{\theta'} = \vec{p}$ if, and only if, there exist a solution to the 3-partition problem. Given the way we build the graph G and since the set of failures should be in H, the rest of the proof is exactly similar to the proof of Lemma 5.2. ■

Lemma 5.3 clearly indicates that the problem of detecting the line failures after an attack as described in Subsection 5.2.2 is NP-hard. However, in the following sections, we provide sufficient conditions on the structural properties of the attacked area H and the set of line failures F such that the phase angles can be recovered and the line failures can be detected efficiently after the attack.

5.4 Attack Analysis

In this section, we study the effects of an attack and provide analytical methods for recovering the phase angles and detecting failed lines in the attacked zone H. We find conditions on

the structural properties of a zone and constraints on the failed lines for which these methods successfully recover the phase angles and detect the failed lines. These conditions depend on the connections between V_H and $\bar{V}_H$ as well as the inner connections of the nodes in H. Therefore, we refer to them as *external* and *internal* conditions on H, respectively. Finally, we briefly study the case in which multiple zones are attacked simultaneously. Table 5.1 summarizes the results regarding the resilience of a zone based on its internal and external conditions, and the constraints on the set of failed lines F.

In this section, when we describe our methods, we assume that there are no edges $\{i, j\} \in E_H$ for which $\theta'_i = \theta'_j$ (we refer to these edges as *null-edges*). Following (2.6), a null-edge does not carry any flow. Thus, we cannot detect the status of those edges since they cannot be distinguished from failed lines. However, we can detect the null-edges and treat them separately (we consider this in the PARD Algorithm provided in the next section).

5.4.1 Recovery of Phase Angles

In this subsection, we introduce a method to recover the phase angles of the nodes in an attacked zone H. We provide sufficient conditions on $G[V_H, \bar{V}_H]$ such that the method recovers the phase angles of the nodes in V_H successfully. As we mentioned, since these conditions depend only on the connections between V_H and $\bar{V}_H$, we refer to them as the *external conditions* on H.

The following lemma is the first step towards designing the method for recovering the phase angles and for detecting the failed lines (see Subsection 5.4.2).

Lemma 5.4. $\text{supp}(\mathbf{A}(\vec{\theta} - \vec{\theta'})) \subseteq V_H$.

Proof. Suppose $F = \{e_{i_1}, e_{i_2}, \ldots, e_{i_k}\} \subseteq E_H$ are the edges that are disconnected from the grid after the attack on the zone H. Define the column vectors $\vec{x_1}, \vec{x_2} \ldots \vec{x_k} \in \{-1, 0, 1\}^n$ associated with the failed lines as follows. If $e_{i_j} = \{s_j, t_j\}$ then $\vec{x_j}$ is 1 in its $s_j{}^{th}$ entry, -1 in its t_j^{th} entry, and 0 everywhere else. It is easy to see that $\mathbf{A'}$ is related to $\mathbf{A}$ as $\mathbf{A'} = \mathbf{A} - \sum_{j=1}^{k} a_{s_j t_j} \vec{x_j} \vec{x_j}^t$. Since the graph G does not get disconnected after an attack, the flow equations in G' are $\mathbf{A'}\vec{\theta'} = \vec{p}$.

On the other hand, $\mathbf{A}\vec{\theta} = \vec{p}$, therefore $\mathbf{A}\vec{\theta} - \mathbf{A}'\vec{\theta}' = 0$. Thus,

$$0 = \mathbf{A}\vec{\theta} - \mathbf{A}'\vec{\theta}' = \mathbf{A}\vec{\theta} - \mathbf{A}\vec{\theta}' + \sum_{j=1}^{k} a_{s_j t_j} \vec{x_j} \vec{x_j}^t \vec{\theta}'$$

$$\Rightarrow \text{supp}(\mathbf{A}(\vec{\theta} - \vec{\theta}')) \subseteq \bigcup_{i=1}^{k} \{s_j, t_j\} \subseteq V_H.$$

■

One of the immediate results of Lemma 5.4 is the following corollary. This corollary gives a true statement about $\vec{\theta}'$ (recall that $\vec{\theta}'$ is partly unknown). It states that $\vec{\theta}'$ is in the solution space of the matrix equation (5.1).

Corollary 5.1. *For any $U \subseteq \bar{H}$, $\boldsymbol{A}_{U|N_c(U)}(\vec{\theta}_{N_c(U)} - \vec{\theta}'_{N_c(U)}) = 0$. In particular, when $U = \bar{H}$,*

$$\mathbf{A}_{\bar{H}|G}(\vec{\theta} - \vec{\theta}') = 0. \tag{5.1}$$

For simplicity of the notations and equations, through the most of this chapter, we consider the case in which $U = \bar{H}$. However, as we briefly describe in Subsection 5.4.4, using a smaller U allows the recovery of the phase angles after an attack on multiple zones.

We find sufficient conditions such that the solution $\vec{\theta}'_H$ to (5.1) is unique (given $\vec{\theta}$ and $\vec{\theta}'_{\bar{H}}$), and consequently $\vec{\theta}'_H$ can be recovered after any attack on H. We first define a *well-supported* zone.

Definition 5.2. *A zone H is called* well-supported*, if $\vec{\theta}'_H$ can be recovered after any attack on H.*

Using Corollary 5.1, the following theorem gives sufficient condition for a zone H to be *well-supported*.

Theorem 5.1. *A zone H is* well-supported*, if $\mathbf{A}_{\bar{H}|H}$ has linearly independent columns.*

Proof. From Corollary 5.1 we know that $\mathbf{A}_{\bar{H}|G}(\vec{\theta} - \vec{\theta}') = 0$, therefore $\mathbf{A}_{\bar{H}|H}\vec{\theta}'_H = \mathbf{A}_{\bar{H}|\bar{H}}(\vec{\theta}_{\bar{H}} - \vec{\theta}'_{\bar{H}}) + \mathbf{A}_{\bar{H}|H}\vec{\theta}_H$. The only unknown in this equation is $\vec{\theta}'_H$. Now since $\mathbf{A}_{\bar{H}|H}$ has linearly independent columns, this equation has a unique solution $\vec{\theta}'_H$ which can be computed in polynomial time. Thus, $\vec{\theta}'_H$ can be recovered in this case and zone H is well-supported. ■

It can be seen that the sufficient condition in Theorem 5.1 depends on the reactance values. However, the following corollary relaxes the condition in Theorem 5.1. It shows that if $G[V_H, \bar{V}_H]$ has a matching that covers V_H, then for almost any reactance values for the edges in $E[V_H, \bar{V}_H]$, H is *well-supported*. The idea is that the set of reactance values for the edges in $E[V_H, \bar{V}_H]$ for which $\mathbf{A}_{\bar{H}|H}$ does not have linearly independent columns is a measure zero set in the real space [119].

Corollary 5.2. *If there is a matching in $G[V_H, \bar{V}_H]$ that covers V_H, then H is* well-supported *almost surely.*[1]

Proof. Suppose $M = (U, V_H)$ is the matching for $G[V_H, \bar{V}_H]$ that covers V_H, and suppose $U \subseteq \bar{V}_H$ are the matched nodes which are in $\bar{V}_H$. Since M is the matching in $G[V_H, \bar{V}_H]$ that covers H, thus $|U| = |V_H|$. Regarding Theorem 5.1, to show that H is well-supported almost surely, we need to show that the columns of the matrix $\mathbf{A}_{\bar{H}|H}$ are linearly independent almost surely. For this reason, we show that $\det(\mathbf{A}_{U|V_H}) \neq 0$ almost surely. $\det(\mathbf{A}_{U|V_H})$ can be considered as a polynomial of the nonzero entries of the admittance matrix using Leibniz formula. Now assume $U = \{u_1, u_2, \ldots, u_{|V_H|}\}$ are matched to $V_H = \{v_1, v_2, \ldots, v_{|V_H|}\}$ in order. It can be seen that $\prod_{i=1}^{|V_H|} a_{u_i v_i}$ is a term with nonzero coefficient in $\det(\mathbf{A}_{U|V_H})$. Therefore, $\det(\mathbf{A}_{U|V_H})$ is not a zero polynomial in terms of its nonzero entries. Now since the set of reactance values for the edges in $E[V_H, \bar{V}_H]$ such that $\det(\mathbf{A}_{U|V_H}) = 0$ is a measure zero set in the real space, thus $\det(\mathbf{A}_{U|V_H}) \neq 0$ almost surely. ■

In reality, since the reactance values are derived by the physical properties of the lines, we expect that these values are relatively random around a mean value. Thus, following Corollary 5.2, the existence of a matching that covers every node in V_H is enough for a zone to be well-supported (see Fig. 5.3 for an example of a graph in which every node in a zone is covered by a matching). Hence, in the following sections we consider the existence of a matching as a sufficient external condition on H to be well-supported.

[1] In probability theory, one says that an event happens almost surely, if it happens with probability one.

Table 5.1: Summary of the results in Section 5.4. The external/internal conditions on the structural properties of a zone H such that after an attack with certain constraints, the phase angles can be recovered and the failed lines can be detected by solving (5.5). Matching and partial matching refer to matchings in $G[V_H, \bar{V}_H]$ that cover V_H and $V_H \backslash (V_H^{\text{in}} \cup V_H^{\text{out}})$, respectively.

External conditions	Internal conditions	Attack constraints	Resilience	Results
Matching	Acyclic	None	attack-resilient	Corollary 5.2/Lemma 5.6
Matching	Planar	$\forall$ cycle C, $\lvert C \cap F\rvert < \lvert C \backslash F\rvert$ F^* is H^*-separable	weakly-attack-resilient	Corollary 5.2/Theorem 5.2
Partial matching	Acyclic	$\forall v \in V_H^{\text{in}}$, $\lvert\partial(v) \cap F\rvert < \lvert\partial(v) \backslash F\rvert$	weakly-attack-resilient	Lemmas 5.6,5.9/Corollary 5.5
Partial matching	Planar No cycle contains an inner-connected-node	$\forall$ cycle C, $\lvert C \cap F\rvert < \lvert C \backslash F\rvert$ $\forall v \in V_H^{\text{in}}$, $\lvert\partial(v) \cap F\rvert_b < \lvert\partial(v) \backslash F\rvert_b$ F^* is H^*-separable	weakly-attack-resilient	Theorem 5.3/Corollary 5.5

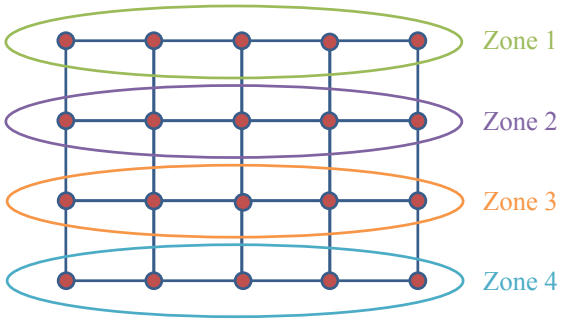


Figure 5.3: An example of a graph and set of zones such that each zone is both well-supported and acyclic.

5.4.2 Detecting Failed Lines

In this subsection, we assume that after an attack, the phase angles are recovered using the method in Subsection 5.4.1 (i.e., by solving (5.1)). We introduce methods to detect the failed lines using $\vec{\theta}'$. We provide sufficient conditions on H such that these methods detect the failed lines successfully. As we mentioned, since these conditions depend only on the connections between the nodes in H, we refer to them as *internal conditions* on H.

The following Lemma is the foundation for our approach to find the failed lines. It limits the set of failed lines to the solution space of the matrix equation (5.2). It can be considered as the complement of Corollary 5.1.

Lemma 5.5. *There exists a vector* $\vec{x} \in \mathbb{R}^{|E_H|}$ *such that* $\text{supp}(\vec{x}) = \{i | e_i \in F\}$ *and*

$$\mathbf{D}_H \vec{x} = \mathbf{A}_{H|G}(\vec{\theta} - \vec{\theta'}). \tag{5.2}$$

Moreover, for any $W \subseteq G$ *such that* $N_c(H) \subseteq W$, $\boldsymbol{D}_H \vec{x} = \boldsymbol{A}_{H|W}(\vec{\theta}_W - \vec{\theta}'_W)$.

Proof. We use the notation that we used in proof of Lemma 5.4. Recall from the proof of Lemma 5.4 that $\mathbf{A}(\vec{\theta} - \vec{\theta'}) = -\sum_{j=1}^{k} a_{s_j t_j} \vec{x_j} \vec{x_j}^t \vec{\theta'}$. It is easy to see that if $\vec{d_1}, \vec{d_2}, \ldots, \vec{d_m}$ are the columns of the incidence matrix $\mathbf{D}$, then $\forall j (1 \leqslant j \leqslant k)$, there exists $b_j \in \mathbb{R}$ such that $b_j \vec{d_{i_j}} = -a_{s_j t_j} \vec{x_j} \vec{x_j}^t \vec{\theta'}$. Therefore, $\mathbf{A}(\vec{\theta} - \vec{\theta'}) = \sum_{j=1}^{k} b_j \vec{d_{i_j}}$. Thus, if we define $\vec{y} \in \mathbb{R}^m$ such that $\forall e_{i_j} \in F, y_{i_j} = b_j$ and 0 elsewhere, then $\mathbf{A}(\vec{\theta} - \vec{\theta'}) = \mathbf{D}\vec{y}$ and $\text{supp}(\vec{y}) \subseteq \{i_1, i_2, \ldots, i_k\}$. However, from the Corollary 5.1 we know that $\mathbf{A}_{\bar{H}|G}(\vec{\theta} - \vec{\theta'}) = 0$. Moreover, since $F \subseteq E_H$, $\vec{y}_{\bar{H}} = 0$. Thus, we can restrict the equation only to the components of the zone H, which means that $\mathbf{A}_{H|G}(\vec{\theta} - \vec{\theta'}) = \mathbf{D}_H \vec{y}_H$. Now it is easy to see that since we assumed that no null-edges are in F, all the b_is are nonzero and $\text{supp}(\vec{y}_H) = \{i_1, i_2, \ldots, i_k\}$. Therefore, $\vec{x} = \vec{y}_H$ is a solution to (5.2) and $\text{supp}(\vec{x}) = \{i | e_i \in F\}$. Now, since for any $i \in H$ and $j \notin N_c(H)$ we have $a_{ij} = 0$, it is easy to see that for any $W \subseteq G$ such that $N_c(H) \subseteq W$, $\mathbf{D}_H \vec{x} = \mathbf{A}_{H|W}(\vec{\theta}_W - \vec{\theta}'_W)$. ■

Lemma 5.5 provides important information regarding the failed lines. It states that there exists a solution $\vec{x}$ to (5.2) such that $\text{supp}(\vec{x})$ reveals the set of failed lines. However, the solution to (5.2) may not be unique. Again, for simplicity of the notations and equations, through the most of this chapter, we consider the case in which $W = G$. However, as we briefly describe in Subsection 5.4.4, using a smaller W allows the failed lines detection after an attack on multiple zones.

The lemma below provides a necessary and sufficient condition on H such that the solution to (5.2) is unique.

Lemma 5.6. *The solution to (5.2) is unique and* $\text{supp}(\vec{x}) = \{i | e_i \in F\}$*, if and only if* H *is acyclic.*

Proof. It is easy to see that the solution to (5.2) is unique if and only if $\mathbf{D}_H$ has linearly independent columns. It is known that $\text{rank}(\mathbf{D}_H) = |V_H| - c$ in which c is the number of

connected components of H [22, Theorem 2.3]. Therefore, $\mathbf{D}_H$ has linearly independent columns if and only if each connected component of $\mathbf{D}_H$ is a tree, which means that $\mathbf{D}_H$ should be acyclic. ■

According to Lemma 5.6 the set of failed lines for any attack can be detected, if and only if H is acyclic. Fig. 5.3 shows an example of a graph and set of zones such that each zone is both well-supported and acyclic (case I in Table 5.1).

Although Lemma 5.6 requires H to be an acyclic graph in order for the solution of (5.2) to be unique, by setting some constraints on the failed lines F, we provide a method to detect the failed lines in broader class of graphs. The underlying idea is that the set of failed lines is expected to be relatively sparse compared to the overall set of edges within a zone. Thus, we are interested in the solutions of (5.2) that are relatively sparse. The ℓ_0-norm should be used to capture the sparseness of a vector. However, since minimizing ℓ_0-norm is a combinatorial problem in general cases, we prefer to use ℓ_1-norm which is known to be a good approximation of the ℓ_0-norm. Thus, we consider the following minimization problem,

$$\min \|\vec{x}\|_1 \text{ s.t. } \mathbf{D}_H\vec{x} = \mathbf{A}_{H|G}(\vec{\theta} - \vec{\theta}'). \tag{5.3}$$

Notice that (5.3) is still linear and can be solved using Linear Programming. Moreover, when the solution to (5.3) also appears to be sparse, which is usually the case in the considered scenario, there are very fast algorithms to solve it [64].

The Lemma below states that by solving (5.3), the failed lines can be detected in more cases than by solving (5.2). The idea that we use in proof of Lemma 5.7 is the core idea in proofs of Theorems 5.2 and 5.3, as well. Namely, the null space of D_H is in one-to-one correspondence with the cycle space of the graph H. Therefore, there are graph theoretical interpretations to the solution space of (5.2). Hence, by using tools from graph theory and linear algebra, we find the solution to (5.2) with the minimum ℓ_1-norm.

Lemma 5.7. *If H is a cycle and $|E_H \cap F| < |E_H \backslash F|$, the solution to (5.3) is unique and* $\text{supp}(\vec{x}) = \{i | e_i \in F\}$.

Proof. Here without loss of generality, we assume that $\mathbf{D}_H$ is the incidence matrix of H when edges of H has been oriented clockwise. Since H is connected, it is known that $\text{rank}(\mathbf{D}_H) = |V_H| - 1$ [22, Theorem 2.2]. Therefore, $\dim(\text{Null}(\mathbf{D}_H)) = 1$. Suppose $\vec{e} \in \mathbb{R}^{|E_H|}$ is the all one vector. It is easy to see that $\mathbf{D}_H\vec{e} = 0$. Since $\dim(\text{Null}(\mathbf{D}_H)) = 1$, $\vec{e}$ is the basis for the null space of $\mathbf{D}$. Suppose $\vec{x}$ is a solution to (5.2) such that $\text{supp}(\vec{x}) = \{i|e_i \in F\}$ (from Lemma 5.5 we know that such a solution exists). To prove that $\vec{x}$ is the unique solution for (5.3), we only need to prove that $\forall c \in \mathbb{R}\backslash\{0\}$, $\|\vec{x}\|_1 < \|\vec{x} - c\vec{e}\|_1$. Without loss of generality we can assume that $x_1, x_2, \ldots, x_k$ are the nonzero elements of $\vec{x}$, in which $k = |F|$. From the assumption we know that $|E_H \cap F| < |E_H\backslash F|$, therefore $k < |E_H|/2$. Hence, we have

$$\begin{aligned}\|\vec{x} - c\vec{e}\|_1 &= \sum_{i=1}^{k} |x_i - c| + (|E_H| - k)|c| \\ &= \sum_{i=1}^{k} (|x_i - c| + |c|) + (|E_H| - 2k)|c| \\ &\geqslant \sum_{i=1}^{k} |x_i| + (|E_H| - 2k)|c| > \sum_{i=1}^{k} |x_i| = \|\vec{x}\|_1.\end{aligned}$$

Thus, the solution to (5.3) is unique. ■

Corollary 5.3. *If all the cycles in H are edge-disjoint and for any cycle C in H, $|C \cap F| < |C\backslash F|$, then the solution to (5.3) is unique and* $\text{supp}(\vec{x}) = \{i|e_i \in F\}$.

The following Theorem extends the idea in the proof of Lemma 5.7 and provides sufficient conditions for failed lines in a planar graph H to be detected by solving (5.3) (recall from subsection 5.2.3 that H^* is the dual of the planar graph H and F^* is the dual of the set of failed lines). For the proof details see Section 5.B.

Theorem 5.2. *In a planar graph H, the solution to (5.3) is unique and* $\text{supp}(\vec{x}) = \{i|e_i \in F\}$, *if:* (i) *for any cycle C in H, $|C \cap F| < |C\backslash F|$, and* (ii) *$F^*$ is H^*-separable.*

Fig. 5.4 shows an example of a zone H for which the set of failed lines can be detected by solving (5.3) based on Theorem 5.2 (case II in Table 5.1).

The Corollary below states that in planar bipartite graphs, condition (ii) in Theorem 5.2 immediately holds, if condition (i) holds. For the proof details see Section 5.B.

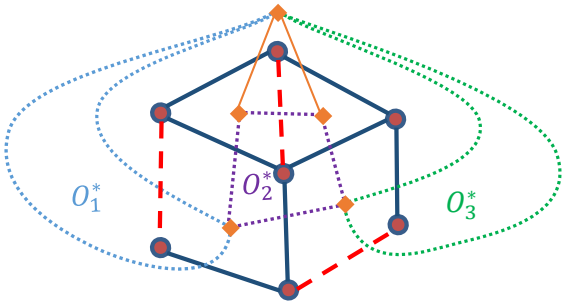


Figure 5.4: An example of a zone H and a set of failed lines (shown by red dashed lines) that can be detected by solving (5.3) based on Theorem 5.2. The diamond orange nodes are the nodes of the dual graph H^*. As can be seen, the dual of the failed lines can be covered by three edge disjoint cycles O_1^*, O_2^*, O_3^* (shown by dotted lines) in H^*. Thus, as Theorem 5.2 requires, F^* is H^*-separable.

Corollary 5.4. *In a planar bipartite graph H, the solution to (5.3) is unique and* $\text{supp}(\vec{x}) = \{i|e_i \in F\}$, *if for any cycle C in H, $|C \cap F| < |C \backslash F|$.*

Theorem 5.2 and Corollary 5.4 are important since power grids are usually considered to be planar. For instance, lattice graphs are planar bipartite.

5.4.3 Simultaneous Phase Angles Recovery and Failed Lines Detection

In Subsection 5.4.1 we showed that the phase angles of the zone H are recoverable, if there is a matching in $G[V_H, \bar{V}_H]$ that covers V_H. However, in reality, this condition might be very difficult and costly to maintain (i.e., it may require to increase the number of zones). Therefore, in this subsection, using similar ideas as in subsection 5.4.2, we relax the external conditions on H.

The key idea which is summarized in the following Lemma, is to combine Corollary 5.1 and Lemma 5.5.

Lemma 5.8. *There exist vectors $\vec{x} \in \mathbb{R}^{|E_H|}$ and $\vec{\delta}_H \in \mathbb{R}^{|V_H|}$ such that* $\text{supp}(\vec{x}) = \{i|e_i \in F\}$, $\vec{\delta}_H = \vec{\theta}_H - \vec{\theta}'_H$, *and*

$$\begin{aligned} \mathbf{D}_H \vec{x} &= \mathbf{A}_{H|H} \vec{\delta}_H + \mathbf{A}_{H|\bar{H}} \vec{\delta}_{\bar{H}} \\ \mathbf{A}_{\bar{H}|H} \vec{\delta}_H &+ \mathbf{A}_{\bar{H}|\bar{H}} \vec{\delta}_{\bar{H}} = 0 \end{aligned} \tag{5.4}$$

where $\vec{\delta}_{\bar{H}} = \vec{\theta}_{\bar{H}} - \vec{\theta}'_{\bar{H}}$ *and is known.*

From Subsections 5.4.1 and 5.4.2 we know that the solution to (5.4) is unique, if and only if H is acyclic and $\mathbf{A}_{\bar{H}|H}$ has linearly independent columns. Therefore, to deal with cases for which $\mathbf{A}_{\bar{H}|H}$ does not have linearly independent columns, we consider a similar optimization problem as in (5.3) but with more constraints. For this reason, as we mentioned in Subsection 5.4.2, since the set of failed lines is expected to be relatively sparse compared to the overall set of edges, we consider the following optimization problem,

$$\begin{aligned} &\min \|\vec{x}\|_1 \text{ s.t.} \\ &\mathbf{D}_H \vec{x} = \mathbf{A}_{H|H}\vec{\delta}_H + \mathbf{A}_{H|\bar{H}}\vec{\delta}_{\bar{H}} \\ &\mathbf{A}_{\bar{H}|H}\vec{\delta}_H + \mathbf{A}_{\bar{H}|\bar{H}}\vec{\delta}_{\bar{H}} = 0. \end{aligned} \tag{5.5}$$

The following Lemma states that if there is an independent set of nodes in H with no neighbors in $\bar{H}$, then under some conditions on F, we can recover F and $\vec{\theta}'_H$ by solving (5.5) even when $\mathbf{A}_{\bar{H}|H}$ does not have linearly independent columns (case III in Table 5.1). First, we define *inner-connected* nodes.

Definition 5.3. *A node* $v \in V_H$ *is called* H-inner-connected *if* $N(v) \subseteq V_H$*. It is called* H-outer-connected *if* $N(v) \subseteq V_{\bar{H}}$*. We denote the set of* H-inner-connected *and* H-outer-connected *nodes by* V_H^{in} *and* V_H^{out}*, respectively.*

For any $L \subseteq E$, define its size weighted by the susceptance values, as $|L|_b := \sum_{e\in L} |b_e|$, in which b_e is the susceptance of the line e. Notice that when the susceptance values for all the lines equal to -1, then $|L|_b = |L|$.

Lemma 5.9. *Suppose* H*-inner-connected nodes form an independent set in* H*. If* H *is acyclic,* $\text{rank}(\boldsymbol{A}_{\bar{H}|H}) = |V_H| - |V_H^{in}|$*, and* $\forall v \in V_H^{in}, |\partial(v) \cap F|_b < |\partial(v)\backslash F|_b$*, then the solution* $\vec{x}, \vec{\delta}$ *to (5.5) is unique. Moreover,* $\text{supp}(\vec{x}) = \{i | e_i \in F\}$ *and* $\vec{\delta}_H = \vec{\theta}_H - \vec{\theta}'_H$*.*

Proof. The idea of the proof is very similar to the proof of Lemma 5.7. Suppose $\vec{x}, \vec{\delta}_H$ is the solution to (5.4) such that $\text{supp}(\vec{x}) = \{i | e_i \in F\}$ and $\vec{\delta}_H = \vec{\theta}_H - \vec{\theta}'_H$. From Lemma 5.8 we know

that such a solution exists. We show that this solution is the unique solution to (5.5) in this setting.

Without loss of generality in addition to assuming $V_H = \{1, 2, 3, \ldots, |V_H|\}$ and $E_H = \{e_1, e_2, \ldots, e_{|E_H|}\}$, we can assume the labeling of the nodes in G is such that $V_H^{\text{in}} = \{1, 2, \ldots, t\}$ is the set of H-inner-connected nodes. Suppose $\vec{\alpha}_1, \vec{\alpha}_2, \ldots, \vec{\alpha}_t \in \mathbb{R}^{|V_H|}$ are the coordinate vectors, in other words $\vec{\alpha}_i$ is 1 at its i^{th} entry and 0 everywhere else. It is easy to see that $\forall i \in V_H^{\text{in}} : A_{\bar{H}|H}\vec{\alpha}_i = 0$. On the other hand, since $\text{rank}(\mathbf{A}_{\bar{H}|H}) = |V_H| - t$ and $\vec{\alpha}_i$s are linearly independent, $\vec{\alpha}_1, \vec{\alpha}_2, \ldots, \vec{\alpha}_t$ form a basis for $\text{Null}(\mathbf{A}_{\bar{H}|H})$.

Assume $\mathbf{D}_H$ is the incidence matrix of H when its edges are oriented such that for each $i \in V_H^{\text{in}}$, the edges are coming out of i. Now suppose $\vec{z}$ is another solution to (5.5), it is easy to see that $\mathbf{D}_H(\vec{z} - \vec{x}) = \mathbf{A}_{H|H}\vec{\alpha}$ for a vector $\vec{\alpha} \in \text{Null}(\mathbf{A}_{\bar{H}|H})$. Since $\vec{\alpha} \in \text{Null}(\mathbf{A}_{\bar{H}|H})$, there are unique coefficients $c_1, c_2, \ldots, c_t \in \mathbb{R}$ such that $\vec{\alpha} = c_1\vec{\alpha_1} + c_2\vec{\alpha_2} + \cdots + c_t\vec{\alpha_t}$. Thus,

$$\begin{aligned}\mathbf{D}_H(\vec{z} - \vec{x}) &= \mathbf{A}_{H|H}\vec{\alpha} = \mathbf{A}_{H|H}(c_1\vec{\alpha}_1 + c_2\vec{\alpha}_2 + \cdots + c_t\vec{\alpha}_t) \\ &= c_1\mathbf{A}_{H|H}\vec{\alpha}_1 + c_2\mathbf{A}_{H|H}\vec{\alpha}_2 + \cdots + c_t\mathbf{A}_{H|H}\vec{\alpha}_t.\end{aligned}$$

Suppose $\vec{d_j}$ is the column associated with edge e_j in D_H. Notice that for each $i \in V_H^{\text{in}}$, $\partial(i) \subseteq E_H$. Therefore, $\forall i \in V_H^{\text{in}}$ and $\forall e_j \in \partial(i)$, $\vec{d_j}$ is a column of $\mathbf{D}_H$. It can be seen that for any $i \in V_H^{\text{in}}$, $\sum_{j:e_j \in \partial(i)} -b_{e_j}\vec{d_j} = \mathbf{A}_{H|H}\vec{\alpha}_i$. If for any $i \in V_H^{\text{in}}$, we define vector $\vec{\beta}_i \in \mathbb{R}^{|E_H|}$ as follows,

$$\beta_{ij} := \begin{cases} -b_{e_j} & \text{if } e_j \in \partial(i) \\ 0 & \text{otherwise,} \end{cases}$$

then $\mathbf{D}_H\vec{\beta}_i = \mathbf{A}_{H|H}\vec{\alpha}_i$ for any $i \in V_H^{\text{in}}$. Thus,

$$\begin{aligned}\mathbf{D}_H(c_1\vec{\beta}_1 + \cdots + c_t\vec{\beta}_t) &= c_1\mathbf{A}_{H|H}\vec{\alpha}_1 + \cdots + c_t\mathbf{A}_{H|H}\vec{\alpha}_t \\ \Rightarrow \mathbf{D}_H(\vec{z} - \vec{x}) &= \mathbf{D}_H(c_1\vec{\beta}_1 + c_2\vec{\beta}_2 + \cdots + c_t\vec{\beta}_t).\end{aligned}$$

Now since H is acyclic, $\mathbf{D}_H$ has linearly independent columns. Thus, from the equation above we can conclude that,

$$\begin{aligned}\vec{z} - \vec{x} &= c_1\vec{\beta}_1 + c_2\vec{\beta}_2 + \cdots + c_t\vec{\beta}_t \\ \Rightarrow \vec{z} &= \vec{x} + c_1\vec{\beta}_1 + c_2\vec{\beta}_2 + \cdots + c_t\vec{\beta}_t.\end{aligned}$$

Using equation above, we show that $\|\vec{z}\|_1 > \|\vec{x}\|_1$ unless $c_1 = c_2 = \cdots = c_t = 0$. First, notice that since V_H^{in} is an independent set, $\forall i \neq j \in V_H^{\text{in}}$, $\partial(i) \cap \partial(j) = \varnothing$. Suppose $\forall i \in V_H^{\text{in}}, |\partial(i) \cap F|_b = k_i$, we have

$$\begin{aligned}
\|\vec{z}\|_1 &= \|\vec{x} + c_1\vec{\beta}_1 + c_2\vec{\beta}_2 + \cdots + c_t\vec{\beta}_t\|_1 \\
&= \sum_{i\in V_H^{\text{in}}}\Big((|\partial(i)|_b - k_i)|c_i| + \sum_{j\in F\cap\partial(i)} |x_j + b_{e_j}c_i|\Big) + \sum_{i\in F\setminus\partial(V_H^{\text{in}})}|x_i| \\
&= \sum_{i\in V_H^{\text{in}}}\Big((|\partial(i)|_b - 2k_i)|c_i| + \sum_{j\in F\cap\partial(i)} \big(|x_j + b_{e_j}c_i| + |b_{e_j}c_i|\big)\Big) + \sum_{i\in F\setminus\partial(V_H^{\text{in}})}|x_i| \\
&\geqslant \sum_{i\in V_H^{\text{in}}}\Big((|\partial(i)|_b - 2k_i)|c_i| + \sum_{j\in F\cap\partial(i)} |x_j|\Big) + \sum_{i\in F\setminus\partial(V_H^{\text{in}})}|x_i| \\
&= \sum_{i\in V_H^{\text{in}}}\big((|\partial(i)|_b - 2k_i)|c_i|\big) + \sum_{i\in V_H^{\text{in}}}\sum_{j\in F\cap\partial(i)} |x_j| + \sum_{i\in F\setminus\partial(V_H^{\text{in}})}|x_i| \\
&= \sum_{i\in V_H^{\text{in}}}\big((|\partial(i)|_b - 2k_i)|c_i|\big) + \|\vec{x}\|_1.
\end{aligned}$$

Now, since from the assumptions $\forall i \in V_H^{\text{in}}, k_i < |\partial(i)|_b/2$, it is easy to see that $\sum_{i\in V_H^{\text{in}}}((|\partial(i)|_b - 2k_i)|c_i|) + \|\vec{x}\|_1 > \|\vec{x}\|_1$, unless $c_1 = c_2 = \cdots = c_t = 0$. Since $\vec{z}$ is a solution to (5.5), we should have $c_1 = c_2 = \cdots = c_t = 0$, and $\vec{z} = \vec{x}$. Thus, $\vec{x}$ is the unique solution to (5.5) and $\text{supp}(\vec{x}) = \{i | e_i \in F\}$. From the proof, it is easy to see that $\vec{\delta}_H$ is also unique. ■

To generalize Lemma 5.9, first let us consider cases in which H contains H-outer-connected nodes. The Lemma below shows that the value of δ for these nodes is unique.

Lemma 5.10. *If v is H-outer-connected and $\vec{\delta}_H$ is a solution to (5.4), then δ_v is unique and equal to $\delta_v = 1/a_{vv}\sum_{u\in N(v)} -a_{vu}\delta_u$.*

Proof. First, notice that since v is H-outer-connected, $N(v) \subseteq \bar{V}_H$. Now, let us compute the v^{th} entry of the vectors on the both side of the equation $\mathbf{D}_H\vec{x} = \mathbf{A}_{H|H}\vec{\delta}_H + \mathbf{A}_{H|\bar{H}}\vec{\delta}_{\bar{H}}$. Since v is H-outer-connected, the v^{th} row of $\mathbf{D}_H$ is a zero vector. Thus, $(\mathbf{D}_H\vec{x})_v = 0$ for any $\vec{x}$. One the other hand, from the definition of admittance matrix $\mathbf{A}$, $(\mathbf{A}_{H|H}\vec{\delta}_H)_v = a_{vv}\delta_v$ and $(A_{H|\bar{H}}\vec{\delta}_{\bar{H}})_v = \sum_{u\in N(v)} a_{vu}\delta_u$. Since $(\mathbf{D}_H\vec{x})_v = (\mathbf{A}_{H|H}\vec{\delta}_H)_v + (A_{H|\bar{H}}\vec{\delta}_{\bar{H}})_v$, we can conclude that $\delta_v = 1/a_{vv}\sum_{u\in N(v)} -a_{vu}\delta_u$. Thus, the proof is complete. ■

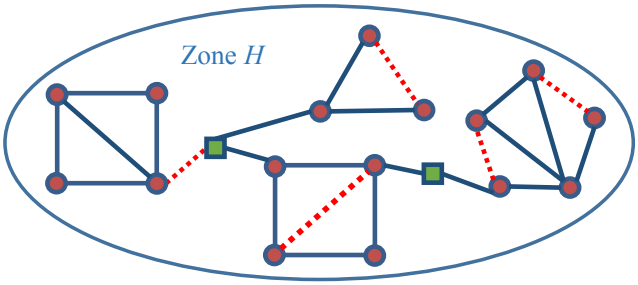


Figure 5.5: An example of a zone H and an attack such that the phase angles can be recovered and the failed lines can be detected by solving (5.5) based on Theorem 5.3. The squared green nodes are the H-inner-connected nodes. The failed lines are shown by red dashed edges.

In the following theorem, we generalize Lemma 5.9. This theorem combines Lemma 5.9 and Theorem 5.2, and provides a broader class of graphs in which solving (5.5) recovers phase angles and detects the failed lines after an attack. For the proof details see Section 5.B.

Theorem 5.3. *In a planar graph H, the solution $\vec{x}, \vec{\delta}_H$ to (5.5) is unique with* $\text{supp}(\vec{x}) = \{i | e_i \in F\}$ *and $\vec{\delta}_H = \vec{\theta}_H - \vec{\theta}'_H$, if the following conditions hold:* (i) $\forall v \in V_H^{\text{in}}$, $|\partial(v) \cap F|_b < |\partial(v) \backslash F|_b$, (ii) *for any cycle C in H, $|C \cap F| < |C \backslash F|$,* (iii) *$F^*$ is H^*-separable,* (iv) *in $\boldsymbol{A}_{\bar{H}|H}$, columns associated with nodes that are neither H-inner-connected nor H-outer-connected are linearly independent,* (v) *no cycle in H contains a H-inner-connected node, and* (vi) *H-inner-connected nodes form an independent set.*

Note that when H is well-supported, there are no H-inner-connected or H-outer-connected nodes. Thus, conditions (i), (iv), (v), and (vi) immediately hold and Theorem 5.3 reduces to Theorem 5.2.

Fig. 5.5 shows an example of a zone H and an attack such that the phase angles can be recovered and the failed lines can be detected by solving (5.5) using Theorem 5.3 (case IV in Table 5.1). As it can be seen, this theorem covers a broad set of graphs and attacks for which we can recover the phase angles and detect the failed lines. Notice that here, with similar argument as in Corollary 5.2 we can replace condition (iv) in Theorem 5.3 with a simpler matching condition as follows.

Corollary 5.5. *If there is a matching in $G[V_H, \bar{V}_H]$ that covers $V_H \backslash (V_H^{\text{in}} \cup V_H^{\text{out}})$, then condition* (iv) *in Theorem 5.3 holds almost surely.*

To conclude, we define the *attack-resilient* and *weakly-attack-resilient* notions to summarize the resilience of a zone to joint cyber and physical attacks.

Definition 5.4. *A zone H is called* attack-resilient*, if it is both well-supported and acyclic.*

Definition 5.5. *A zone H is called* weakly-attack-resilient*, if $\vec{\theta}'_H$ and F can be uniquely found after a constrained attack on the zone H by solving (5.5).*

It is easy to see that an attack-resilient zone is also weakly-attack-resilient.

5.4.4 Recovery and Detection After Attacks on Multiple Zones

In this subsection, we study the case in which multiple zones are attacked simultaneously. When the attacked zones are close to each other, it may not always be possible to recover information. However, if the attacked zones are relatively distant from each other, any of the methods provided in the previous subsections (depending on the conditions on the zones and attacks) can be applied to recover the information and detect the failures in the attacked zones.

The idea is to use Corollary 5.1 and Lemma 5.5 for sets U and W much smaller than $\bar{H}$ and G, respectively. Assume H_1 and H_2 are two attacked zones. Let U_1 and U_2 be two sets with the minimum size such that $U_1 \subseteq \bar{H}_1$, $H_1 \subseteq N_c(U_1)$, $U_2 \subseteq \bar{H}_2$, and $H_2 \subseteq N_c(U_2)$. Following Corollary 5.1, $\mathbf{A}_{U_1|N_c(U_1)}(\vec{\theta}_{N_c(U_1)} - \vec{\theta}'_{N_c(U_1)}) = 0$ and $\mathbf{A}_{U_2|N_c(U_2)}(\vec{\theta}_{N_c(U_2)} - \vec{\theta}'_{N_c(U_2)}) = 0$. Now if $N_c(U_1) \cap H_2 = N_c(U_2) \cap H_1 = \varnothing$ (i.e., H_1 and H_2 are distant enough), and both $\mathbf{A}_{U_1|H_1}$ and $\mathbf{A}_{U_2|H_2}$ have linearly independent columns, then similar to the proof of Theorem 5.1, the phase-angles of the nodes in H_1 and H_2 can be recovered by solving a set of linear equations.

To detect the failed lines, let W_1 and W_2 be two sets with the minimum size such that $W_1, W_2 \subseteq G$, $N_c(H_1) \subseteq W_1$, and $N_c(H_2) \subseteq W_2$. Following Lemma 5.5, there exist vectors $\vec{x_1}, \vec{x_2} \in \mathbb{R}^{|E_H|}$ such that $\text{supp}(\vec{x_1})$ and $\text{supp}(\vec{x_2})$ give the failed lines in H_1 and H_2, and also $\mathbf{D}_{H_1}\vec{x_1} = \mathbf{A}_{H_1|W_1}(\vec{\theta}_{W_1} - \vec{\theta}'_{W_1})$ and $\mathbf{D}_{H_2}\vec{x_2} = \mathbf{A}_{H_2|W_2}(\vec{\theta}_{W_2} - \vec{\theta}'_{W_2})$. Now, if H_1 and H_2 are acyclic and $W_1 \cap H_2 = W_2 \cap H_1 = \varnothing$, then similar to the Lemma 5.6, the solutions to $\mathbf{D}_{H_1}\vec{x_1} =$

$\mathbf{A}_{H_1|W_1}(\vec{\theta}_{W_1} - \vec{\theta}'_{W_1})$ and $\mathbf{D}_{H_2}\vec{x_2} = \mathbf{A}_{H_2|W_2}(\vec{\theta}_{W_2} - \vec{\theta}'_{W_2})$ are unique and the failed lines can be detected by $\text{supp}(\vec{x_1})$ and $\text{supp}(\vec{x_2})$.

Notice that the methods in subsection 5.4.3 can also be simply used to recover the phase angles and detect the failed lines in the attacked zones that are distant enough. The following corollary summarizes our discussion in this subsection.

Corollary 5.6. *The phase angles and the failed lines can be recovered/detected after a simultaneous attack on zones $H_1, H_2, \ldots, H_k$, if followings hold: (i) for any $1 \leqslant i \leqslant k$, if H_i was the only attacked zone, then the phase angle of the nodes and the failed lines could be recovered/detected using the methods in subsections 5.4.1,5.4.2, and 5.4.3, (ii) there exist $U_1, U_2, \ldots, U_k \subseteq G$ and $W_1, W_2, \ldots W_k \subseteq G$ such that:*

1. *For any $1 \leqslant i \leqslant k$, $U_i \subseteq \bar{H}_i$, $H_i \subseteq N_c(U_i)$, and $N_c(H_i) \subseteq W_i$.*
2. *For any $1 \leqslant i \neq j \leqslant k$, $N_c(U_i) \cap H_j = \varnothing$ and $W_i \cap H_j = \varnothing$.*

Proof. For any $1 \leqslant i \leqslant k$, consider equations $\mathbf{A}_{U_i|N_c(U_i)}(\vec{\theta}_{N_c(U_i)} - \vec{\theta}'_{N_c(U_i)}) = 0$ and $\mathbf{D}_{H_i}\vec{x_i} = \mathbf{A}_{H_i|W_i}(\vec{\theta}_{W_i} - \vec{\theta}'_{W_i})$ instead of (5.1) and (5.2). Then, recover the phase angle of the nodes and detect the failed lines at each H_i separately using any of the methods provided in subsections 5.4.1,5.4.2, and 5.4.3. ■

5.5 Post-Attack Recovery and Detection Algorithm

In this section, we present the Post-Attack Recovery and Detection (PARD) Algorithm for recovering the phase angles and detecting the failed lines after an attack on a zone H. Based on the results provided in previous subsections, if a zone H is weakly-attack-resilient, the PARD Algorithm will recover the phase angles and detect the failed lines after a constrained attack.

Notice that if there are some failed lines but no data is missing, then from the data that is gathered by the PDCs from the PMUs, all the information regarding the status of the lines and phase angles is available and there is no need for the algorithm. Thus, as the first step, the PARD Algorithm detects the attacked zone H by checking the missing data (line 1). Then, it solves (5.5) by Linear Programming to obtain $\vec{x}, \vec{\delta}_H$. If H is weakly-attack-resilient, from the

Algorithm 4 - Post-Attack Recovery & Detection (PARD)

Input: A connected graph G, phase angles before the attack $\vec{\theta}$, and partial phase angles after the attack $\vec{\theta}'_{\bar{H}}$.

1: Detect the attacked zone H by checking for missing data.
2: Compute $\vec{x}, \vec{\delta}_H$ the solution to (5.5) by Linear Programming.
3: Compute $\vec{\theta}'_H = \vec{\theta}_H - \vec{\delta}_H$.
4: Compute $F = \{e_i | i \in \text{supp}(\vec{x})\}$.
5: Detect the set of null-edges that appear after the attack as $N = \{\{i,j\} \in E_H | \theta'_i = \theta'_j\}$.
6: **return** N, F, $\vec{\theta}'_H$.

results in previous subsections, we know that $\vec{x}, \vec{\delta}_H$ are unique, $\vec{\theta}'_H = \vec{\theta}_H - \vec{\delta}_H$ (line 3), and $F = \{e_i | i \in \text{supp}(\vec{x})\}$ (line 4). Finally, using $\vec{\theta}'$ computed in previous line, the PARD Algorithm detects the set of null-edges N (line 5), and returns N, F, and $\vec{\theta}'_H$.

5.6 Attack Analysis in the Presence of Measurement Noise and Uncertainty

In this section, we briefly discuss the problem of information recovery after an attack in the presence of a measurement noise and uncertainty. We follow [89] and model the measurement noise by changing (2.7) to $\mathbf{A}(\vec{\theta} - \vec{e}) = \vec{p}$ where $\vec{e} \in \mathbb{R}^{|V|\times 1}$ is a Gaussian measurement noise with a diagonal covariance matrix Σ. Following [163], $\vec{e}$ can also account for the perturbations in $\vec{p}$ after failures. It is obvious that in the presence of noise, the optimization problem (5.5) has no feasible solution. However, since the ℓ_1-norm is relatively robust against noise, one possible approach to generalize the optimization problem (5.5) to the noisy case is to relax the conditions as follows:

$$\begin{aligned} &\min \|\vec{x}\|_1 \text{ s.t.} \\ &\|\mathbf{D}_H \vec{x} - \mathbf{A}_{H|H}\vec{\delta}_H - \mathbf{A}_{H|\bar{H}}\vec{\delta}_{\bar{H}}\|_2 < \epsilon \\ &\|\mathbf{A}_{\bar{H}|H}\vec{\delta}_H + \mathbf{A}_{\bar{H}|\bar{H}}\vec{\delta}_{\bar{H}}\|_2 < \epsilon. \end{aligned} \tag{5.6}$$

Algorithm 5 - 3-Acyclic Partition of Planar (3APP)

Input: A non-empty planar graph G.

1: Find a node $v \in V$ such that $\deg(v) \leqslant 5$.
2: **if** $G \backslash v = \varnothing$ **then** set $Q_1 = Q_2 = Q_3 = \varnothing$.
3: **else** Find 3-partition of $G \backslash v$ using 3APP Algorithm as
Q_1, Q_2, Q_3.
4: Add v to the partition that $|N(v) \cap Q_i|$ is minimum.
5: **return** Q_1, Q_2, Q_3.

It is easy to see that the optimization problem (5.6) is a *second-order cone program* that can be solved using gradient decent methods. After solving (5.6), the line failures can then be detected as before by $F = \{e_i | i \in \text{supp}(\vec{x})\}$.

Generalizing Theorem 5.3 to take into account the noisy case modeled by (5.6) is part of the future work. However, in Section 5.8, we show via simulation that solving the optimization problem (5.6) can correctly recover the phase angles and detect the failed lines depending on the level of the Signal to Noise Ratio (SNR).[2]

5.7 Zone Selection Algorithm

In this section we use the results from Section 5.4 to provide an algorithm for partitioning the power grid into the minimum number of attack-resilient zones. From Lemma 5.6 and Corollary 5.2, for a zone H to be attack-resilient, it is sufficient that H is acyclic and there is a matching in $G[V_H, \bar{V_H}]$ that covers every node in V_H. Fig. 5.3 shows an example of a partitioning such that each zone is attack-resilient. Thus, we define a *matched-forest* partition of a graph G as follows.

Definition 5.6. *A* matched-forest *partition of a graph G into $H_1, H_2, \ldots, H_k$ is a partition such that for any i, H_i is acyclic and $G[V_{H_i}, \bar{V}_{H_i}]$ has a matching that covers V_{H_i}.*

[2] We define the SNR (in dB) as $20 \log_{10}(\|\vec{\theta}\|_2 / \|\vec{e}\|_2)$.

Algorithm 6 - Zone Selection (ZS)

Input: A connected graph G.

1: Find an optimal matching cover $M_1, M_2, \ldots, M_t$ of G [150].
2: For each M_i, separate the matched nodes into two set of nodes V_{2i-1}, V_{2i} such that $\forall\{v,u\} \in M$, $v \in V_{2i-1}$ and $u \in V_{2i}$.
3: For any $1 \leqslant i \leqslant 2t$, $Q_i = V_i \backslash \bigcup_{j=1}^{i-1} Q_j$.
4: **for** each Q_i **do**
5: **if** $G[Q_i]$ is acyclic **then continue**
6: **if** $G[Q_i]$ is a planar graph **then**
7: Use 3APP Algorithm to partition $G[Q_i]$.
8: **else**
9: Use any greedy algorithm to partition $G[Q_i]$ into acyclic subgraphs.
10: Name the resulted partitions $P_1, \ldots, P_k$.
11: **return** $P_1, \ldots, P_k$.

The problem of finding a matched-forest partition of G is closely related to two previously known problems of *vertex arboricity* and *k-matching cover* of a graph. The vertex arboricity $a(G)$ of a graph G is the minimum number of subsets into which the nodes of G can be partitioned so that each subset induces an acyclic graph. It is known that determining $a(G)$ is NP-hard [74, p.193].

A k-matching cover of a graph G is a union of k matchings of G which covers V. The matching cover number of G, denoted by $mc(G)$, is the minimum number k such that G has a k-matching cover. An optimal matching cover of a graph on n nodes can be found in $O(n^3)$ time [150].

Using these results, we study the time complexity of the *minimum matched-forest* partition problem.[3] The following Lemma shows that it is hard to find the minimum matched-forest partition of a graph. For the proof details see Section 5.B.

[3]To the best of our knowledge, this is the first time that the problem is studied.

Lemma 5.11. *The problem of finding the minimum matched-forest partition of a graph G is NP-hard.*

Moreover, we show that finding the minimum matched-forest partition is even hard to approximate. We use the well-known result by Zuckerman [165] that for all $\epsilon > 0$, it is NP-hard to approximate *chromatic number* to within $n^{1-\epsilon}$.

Lemma 5.12. *For all $\epsilon > 0$, it is NP-hard to approximate the minimum matched-forest partition of a graph G to within $n^{1-\epsilon}$.*

Proof. For a graph G, assume $\chi(G)$ is its chromatic number. Since each color gives an independent set of G, induced subgraph by the nodes with the same color is acyclic with no edges. Thus, it is easy to see that $a(G) \leqslant \chi(G)$. Suppose there is an α-approximation algorithm for the minimum matched-forest problem. Define $\hat{G}$ as in proof of Lemma 5.11. Assume this algorithm partitions $\hat{G}$ into k subsets. From the proof of Lemma 5.11, it is easy to see that $k \leqslant \alpha a(G)$. On the other hand, since each acyclic graph has the chromatic number of at most 2, this algorithm gives the $2k$ coloring of graph G. However, $2k \leqslant 2\alpha a(G) \leqslant 2\alpha\chi(G)$. Thus, this algorithm gives a 2α-approximation of the chromatic number of G. However, the result by Zuckerman [165] states that for all $\epsilon > 0$, it is NP-hard to approximate chromatic number to within $n^{1-\epsilon}$. Therefore, for all $\epsilon > 0$, it is NP-hard to approximate the minimum matched-forest problem to within $n^{1-\epsilon}$ as well. ■

Despite these hardness results, we provide the polynomial-time Zone Selection (ZS) Algorithm to find a matched-forest partition of a graph. We prove that the ZS Algorithm provides a constant approximation for the minimum matched-forest partition of a graph G when G is planar.

Before describing the ZS Algorithm in detail, we first describe an algorithm that is used in the ZS Algorithm, when G is planar. It is known that for a planar graph G, $a(G) \leqslant 3$ [41]. Based on the proof provided in [41], we introduce a recursive 3-Acyclic Partition of Planar (3APP) Algorithm. The Lemma below shows the correctness of this Algorithm.

Lemma 5.13. *The 3APP Algorithm partitions the nodes of a planar graph G into 3 subsets such that each subset induces an acyclic graph.*

Proof. It is known that every planar graph has a node of degree less than or equal to 5 [154]. Therefore, line 1 of the algorithm can always find v. At line 4, recursively we know that subgraphs induced by Q_1, Q_2, Q_3 in $G \backslash v$ are acyclic. Now since $\deg(v) \leqslant 5$, there exists a partition such that $|N(v) \cap Q_i| \leqslant 1$. Without loss of generality we can assume that $|N(v) \cap Q_1| \leqslant 1$. Hence, adding v to Q_1 does not produces any cycles. Thus, subgraphs induced by $Q_1 \cup \{v\}, Q_2, Q_3$ in G are acyclic. ■

We now present the ZS Algorithm. The ZS Algorithm first finds an optimal matching cover $M_1, M_2, \ldots, M_t$ of G using an $O(n^3)$ algorithm introduced in [150] (line 1). Then, in lines 2 and 3, it uses this matching cover to partition V into $Q_1, Q_2, \ldots, Q_{2t}$. It is easy to see that for each Q_i, $M_{\lceil i/2 \rceil} \cap E[Q_i, \bar{Q}_i]$ is the matching in $G[Q_i, \bar{Q}_i]$ that covers nodes in Q_i. Then, in order to satisfy the acyclicity condition on the partitions, it partitions Q_is that do not induce an acyclic graph, into subsets so that each subset induces an acyclic graph. When $G[Q_i]$ is a planar graph, it uses 3APP Algorithm to partition $G[Q_i]$. When it is not, it uses any *greedy algorithm* to do so. Thus, the resulted partition $P_1, P_2, \ldots, P_k$ satisfies the conditions of a matched-forest partition.

The lemma below states that when G is planar, the ZS Algorithm provides a constant approximation of the optimal matched-forest partition. We demonstrate the results obtained by the algorithm in the following section. For the proof details see Section 5.B.

Lemma 5.14. *If G is planar, the ZS Algorithm provides a 6-approximation of the minimum* matched-forest *partition of G in $O(n^3)$.*

Notice that the planarity of G is a sufficient but not a necessary condition for the successful execution of the 3APP Algorithm. Hence, as we show in Section 5.8, the ZS algorithm can be applied to almost any power grid network without checking its planarity as long as the 3APP algorithm is executed successfully.

5.8 Numerical Results

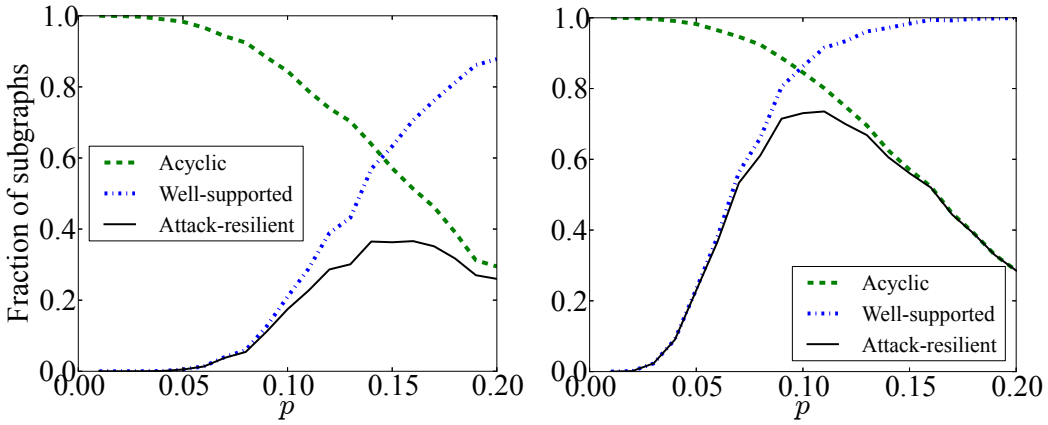
5.8.1 Evaluating grid structural robustness and the performance of the ZS Algorithm

In this section, we (i) numerically study the relationship between the structural properties of a grid and its resilience to joint cyber and physical attacks, and (ii) demonstrate the results obtained by the ZS Algorithm in several known power grid networks.

To assess the relationship between the structural properties of a grid and its resilience to attacks, we quantify the resilience by the fraction of the node induced subgraphs of G that form attack-resilient zones. Recall from Section 5.4 that a zone H is attack-resilient, if it is well-supported and acyclic. We first study the relationships between the number of edges and the fraction of induced subgraphs that form well-supported, acyclic, and attack-resilient zones. These relationships can be best demonstrated in *random graphs* [68], since by increasing p (the probability that two nodes are connected), the total number of edges increases.

We generated random graphs with $n = 30$ and $n = 50$ nodes with various p values. For each p, we generated 100 random graphs and in each graph randomly selected 100 subgraphs of size 10. We then computed the fraction of subgraphs that form well-supported, acyclic, and attack-resilient zones. As can be seen in Fig. 5.6, as p increases, the fraction of subgraphs that form acyclic zones decreases and the fraction that form well-supported zones increases. In Fig. 5.6(a), when $p \approx 0.14$ these two fractions are equal and the fraction of attack-resilient subgraph is maximized. As can be seen in Fig. 5.6(b), the value of p for which these two fractions are equal (i.e., the fraction of attack-resilient subgraphs is maximized) decreases as n increases.

We also illustrate the relationships between the size of the zone $|V_H|$ and the fraction of induced subgraphs that form well-supported, acyclic, and attack-resilient zones. It is known that *scale-free graphs* are relatively good representatives of power grid networks [23], and therefore, we focus here on such graphs. Fig. 5.7 shows the relationships in scale-free graphs with $n = 30$ nodes and $m = 56$ or 104 edges. As can be seen in 5.7(a), as the size of the zone increases, the fraction of zones that are well-supported decreases faster than the fraction of zones that



(a) $n = 30$ (b) $n = 50$

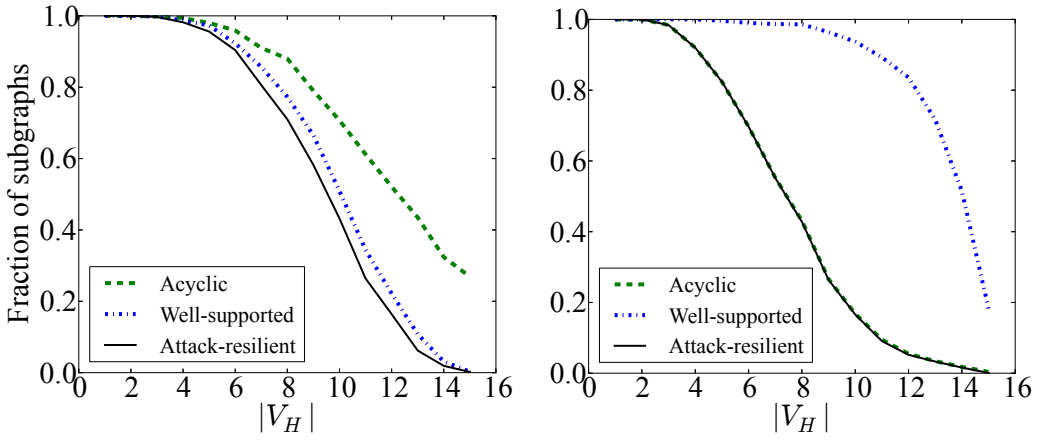
Figure 5.6: The relationships between the number of edges in random graphs (or equivalently p), and the fraction of induced subgraphs that form well-supported and acyclic zones of size $|V_H| = 10$. For each p, 100 random graphs were generated and in each graph, 100 subgraphs were chosen randomly.

are acyclic. Thus, the well-supportedness is the restricting factor for attack-resilience. However, this trend is different in 5.7(b). Since the graph has more edges, as the size of the zone increases, the fraction of zones that are acyclic decreases faster than the fraction of zones that are well-supported. Here, the acyclicity is the restricting factor.

Overall, these results show that the structure of the grid plays an important role in its resilience and should be considered when designing a control network.[4]

We now demonstrate results obtained by the ZS Algorithm in several known power networks. Table 5.2 lists the considered grids and number of resulting partitions. For example, Fig. 5.8 shows the partitions obtained by ZS Algorithm in the IEEE 14-Bus and 30-Bus benchmark systems [5]. As can be seen, in both cases the graphs can be partitioned into two attack-resilient zones. We also evaluated the ZS Algorithm on the IEEE 118 and 300-bus systems, the

[4]Clearly, random and scale free graphs do not model power grids perfectly. However, we used them to gain insight into the relations between structural properties and resilience. In future work, we will study more realistic network topologies [133].



(a) $n = 30, m = 56$ (b) $n = 30, m = 104$

Figure 5.7: The relationship between the size of the zone $|V_H|$ in scale-free graphs, and the fraction of induced subgraphs that form well-supported and acyclic zones of size $|V_H|$. For each case, 1000 subgraphs were chosen randomly.

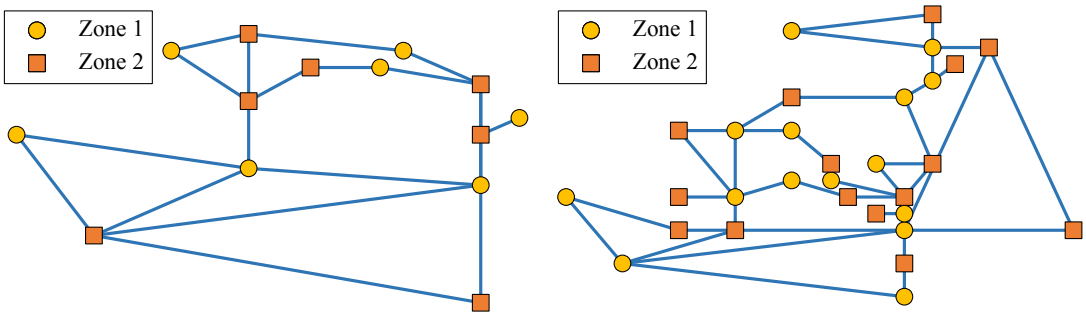
Polish grid (available with MATPOWER [164]), the Colorado state grid, and the U.S. Western Interconnection network.[5] For example, the 6 zones into which the Colorado grid is partitioned appear in Fig. 5.9. Recall form Section 5.7 that when G is planar, the ZS Algorithm is a 6-approximation algorithm for the minimum matched-forest problem. However, as can be seen from the examples above, in practice, it partitions the networks into few zones.

We note that the ZS Algorithm does not take the geographical constraints into account. Thus, when partitioning very large networks such as the Western Interconnection (see Fig. 5.10), the nodes in the same partition may be geographically distant from each other. This is impractical, since the PMUs from the same zone should send the data to a single PDC. However, it is easy to see that if a zone is attack-resilient, any of its subgraphs is also attack-resilient. Therefore, the partitions obtained by the ZS Algorithm can be further divided into smaller zones based on geographical constraints (e.g., into zones within different states in Fig. 5.10). This

[5]The data of the Western Interconnection (and of Colorado) was obtained from the Platts Geographic Information System (GIS) [3].

Table 5.2: Number of partitions into which the ZS Algorithm divides different networks.

Network	Nodes	Edges	Partitions
IEEE 14-Bus	14	20	2
IEEE 30-Bus	30	41	2
IEEE 118-Bus	118	179	5
IEEE 300-Bus	300	409	14
Polish grid	3120	3684	10
Colorado state grid	662	864	6
Western interconnection	13135	16860	9



(a) IEEE 14 bus (b) IEEE 30 bus

Figure 5.8: Partitioning of the IEEE 14 and IEEE 30 bus systems into 2 attack resilient zones (using the ZS Algorithm).

approach does not result in an optimal partitioning. Hence, obtaining an efficient partitioning with geographical constraints is a subject of future work.

5.8.2 Recovering the Information in the Presence of a Measurement Noise

In this subsection, we show via simulation that solving the optimization problem (5.6) can correctly recover the phase angles and detect the failed lines in the presence of the measurement noise depending on the SNR level. To evaluate the results, we count number of *false negatives* and *false positives*. False negatives are the failed lines that are not detected in the solution of (5.6). False positives are the edges that are detected as failed lines in the solution of (5.6)

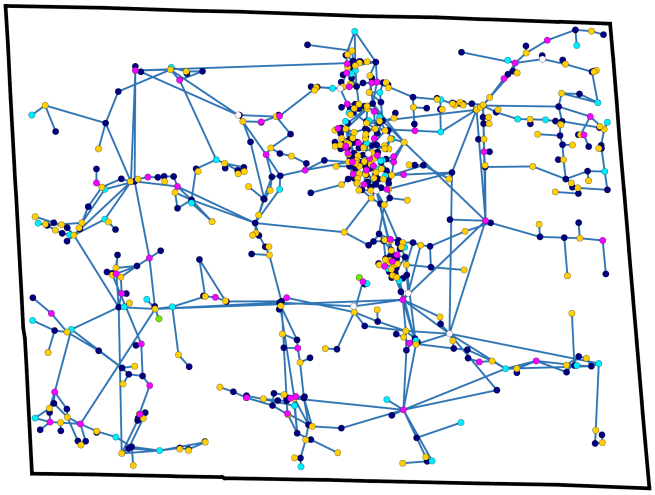


Figure 5.9: Partitioning of the Colorado state grid into 6 attack-resilient zones (using the ZS Algorithm). Nodes with the same color are in the same zone.

despite the fact that they were not failed. We use the Matlab-based solver *CVX* [78] for solving the optimization problem (5.6).

We provide simulation results with the graph and zone H shown in Fig. 5.11 (it is easy to see that H is attack-resilient). Notice that the graph in Fig. 5.11 can be part of a much bigger graph, however following Corollary 5.1 and Lemma 5.5, only the local information is needed to recover the information inside the attacked zone. As we mentioned in Section 5.6, in the simulations, we assume that the readings from the PMUs somewhat differ from the solution of (2.7) (i.e., to the DC power flow). Hence, if $\vec{\theta}$ and $\vec{\theta}'$ are the phase angles obtained from the PMUs (before and after the attack, respectively), then $\mathbf{A}(\vec{\theta} - \vec{e}) = \vec{p}$ and $\mathbf{A}'(\vec{\theta}' - \vec{e}') = \vec{p}$ for unknown Gaussian noise vectors $\vec{e}$ and $\vec{e}'$ with equal covariance matrices.

Figs. 5.12 and 5.13 show two attack scenarios with different SNR values and the information recovered by solving (5.6). Fig. 5.14 shows the average number of false negatives and positives in detecting line failures by solving (5.6) versus the SNR level for different numbers of line failures. As can be seen, for any number of line failures, when the SNR is above a certain level (e.g., 40 dB) the solution to (5.6) can detect the line failures with acceptable accuracy (less than one false negative and zero false positives on average). Using the *CVX* solver, the solution to

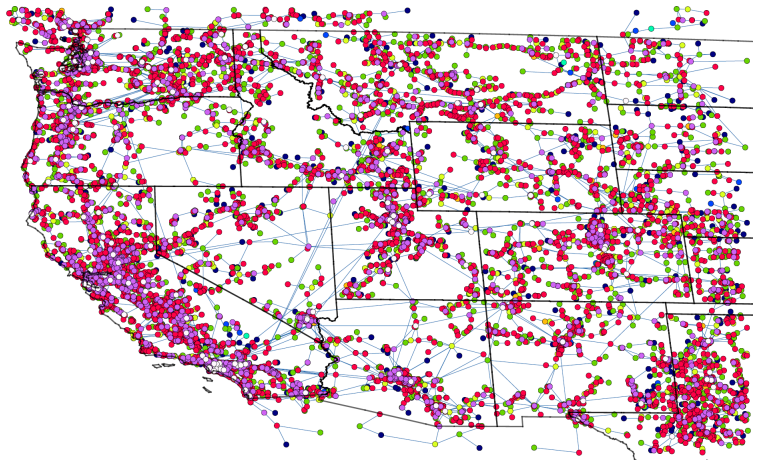


Figure 5.10: Partitioning of the U.S. Western Interconnection into 9 attack-resilient zones (using the ZS Algorithm). Nodes with the same color are in the same zone.

the optimization problem (5.6) can be found in 0.07 sec in our system with Intel Core i7-2600 @3.40GHz CPU and 16GB RAM for the graph depicted in Fig. 5.11.

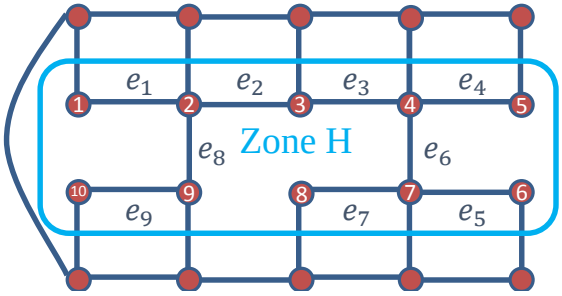
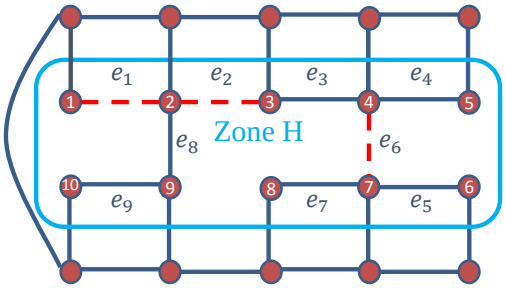


Figure 5.11: The graph and the zone H that are used in the simulations in Subsection 5.8.2. All the edges in the graph have admittance value equal 1. The supply/demand values are chosen randomly.

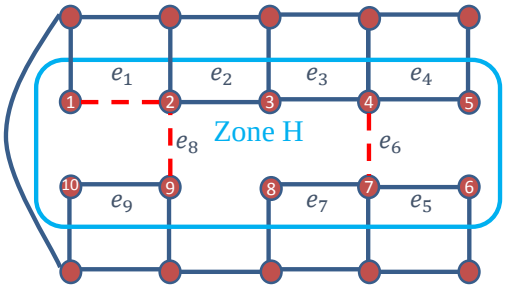


SNR = 50dB

Actual $\vec{\theta}'_H$	Recovered $\vec{\theta}'_H$
0.1062	0.1030
0.0882	0.0872
0.0042	0.0075
0.0062	0.0036
0.0342	0.0334
-0.0758	-0.0752
-0.0971	-0.0937
-0.0925	-0.0919
0.0442	0.0441
0.0322	0.0321

$\vec{x}$	-0.0074	-0.0068	0	0	0	-0.0857	0	0	0
supp($\vec{x}$)	1	1	0	0	0	1	0	0	0

Figure 5.12: An example of an attack and recovered information in the presence of the measurement noise for SNR= 50dB. Red dashed lines show the attacked lines. As can be seen, the attacked lines can be detected successfully in this case.

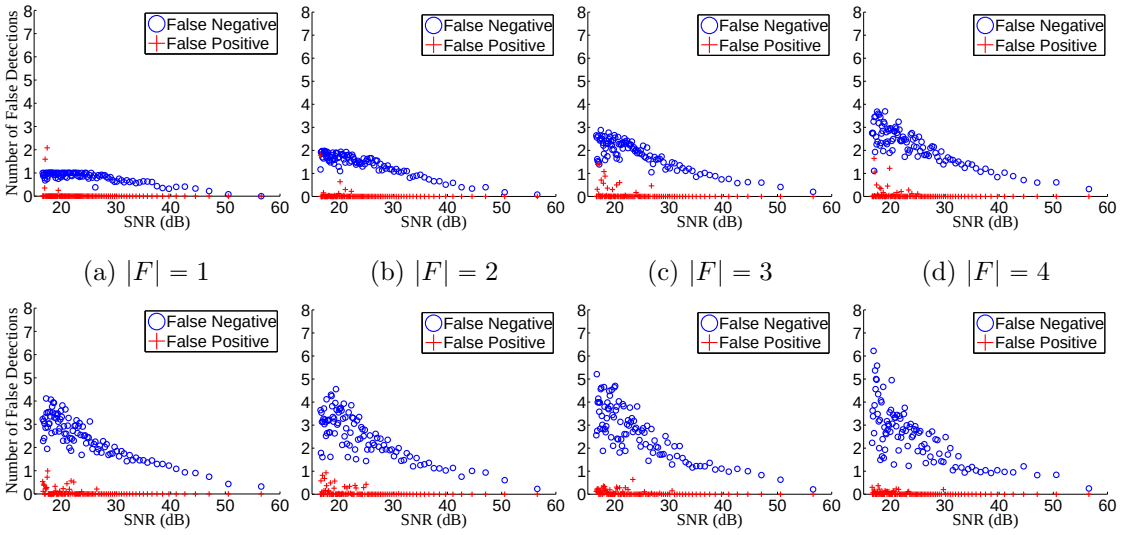


SNR = 30dB

Actual $\vec{\theta}'_H$	Recovered $\vec{\theta}'_H$
0.1071	0.1203
0.1428	0.1242
0.1085	0.0997
0.1013	0.0884
0.1271	0.1273
-0.1409	-0.1381
-0.1623	-0.1496
-0.1576	-0.1563
-0.0429	-0.0299
-0.0329	0.0321

$\vec{x}$	0	0	0	0	0	-0.1938	0	-0.1164	0
supp($\vec{x}$)	0	0	0	0	0	1	0	1	0

Figure 5.13: An example of an attack and recovered information in the presence of the measurement noise for SNR= 30dB. Red dashed lines show the attacked lines. As can be seen, 2 out of the 3 attacked lines can be detected in this case.



(a) $|F| = 1$ (b) $|F| = 2$ (c) $|F| = 3$ (d) $|F| = 4$

(e) $|F| = 5$ (f) $|F| = 6$ (g) $|F| = 7$ (h) $|F| = 8$

Figure 5.14: The average number of false negatives and positives in detecting line failures by solving (5.6) in the presence of the measurement noise versus the SNR. Each data point is the average over 100 trials. (a)-(h) Show this relationship for different number of line failures ($|F|$). Figs. 5.12 and 5.13 provide the detailed information for two of the points in (c).

5.9 Conclusion

We studied joint cyber and physical attacks on power grids. We developed methods to estimate the state of the grid inside the attacked zone using only the information available outside of the attacked zone. We identified graph topologies and constraints on the attacked edges for which these methods are guaranteed to recover the state information. We briefly studied the problem of information recovery in the presence of measurement noise and showed that by relaxing some of the constraints the same methods can be used for information recovery in noisy scenarios. Moreover, we showed that the problem of partitioning the grid into the minimum number of attack-resilient zones is not approximable to within $n^{1-\epsilon}$ for all $\epsilon > 0$ unless P=NP. However, for planar graphs, we developed an approximation algorithm for the partitioning problem and numerically illustrated the operation of the algorithm.

This is one of the first steps towards understanding the vulnerabilities of power grids to joint cyber and physical attacks and developing methods to mitigate their effects. In the next Chapter, we evaluate the performance of the recovery method presented in Section 5.6 when the phase angles are obtained using the AC power flow model.

5.A Preliminaries from Graph Theory

In this appendix, we provide some of the known theorems and definitions used in the proofs provided in Section 5.B.

A *bond* of a graph is a minimal nonempty edge cut, that is, a nonempty edge cut none of whose nonempty proper subsets is an edge cut.

A graph in which each node has even degree is called an *even* graph. A *circuit* is the union of edge-disjoint *cycles*. It is easy to see that a *cycle* is also a *circuit*.

Theorem 5.4 (Theorem 5.2 [123]). *Let $G = (V, E)$ be a planar Eulerian graph, and let $F \subseteq E$. Then the following are equivalent:* (i) *F is G-separable,* (ii) *for each bond D, $|D \cap F| \leqslant |D - F|$.*

Theorem 5.5 (Theorem 10.16 [37]). *Let G be a connected planar graph, and let G^* be a planar dual of G.* (i) *If C is a cycle of G, then C^* is a bond of G^*.* (ii) *If B is a bond of G, then B^* is a cycle of G^*.*

Theorem 5.6 (Theorem 6.1.16 [154]). *The followings are equivalent for a planar graph G,* (i) *G is bipartite,* (ii) *every face of G has even length, and* (iii) *the dual graph G^* is Eulerian.*

Theorem 5.7 (Euler's formula [36]). *For any planar graph G with n nodes, m edges, r faces, and c connected components, the following formula holds, $n + r - m = c$.*

Lemma 5.15 (Corollary 2.16 [36]). *The symmetric difference of two even subgraphs is an even subgraph.*

5.B Proofs

In this appendix, we provide the omitted proofs.

Proof of Theorem 5.2. Recall that we can assume $V_H = \{1, 2, \ldots, |V_H|\}$ and $E_H = \{e_1, e_2, \ldots, e_{|E_H|}\}$. We assign an arbitrary orientation to the edges of H and fix the embedding of H on the plane. We show the set of oriented edges by $\mathcal{E}_{\mathcal{H}} = \{\epsilon_1, \epsilon_2, \ldots, \epsilon_{|E_H|}\}$. Suppose H has r faces and

$C_1, C_2, \ldots, C_r$ are the cycles that surrounded those faces. For each cycle C_i, define vector $\vec{c_i} \in \{-1, 0, 1\}^{|E_H|}$ as follows,

$$c_{ij} = \begin{cases} 0 & \text{if } \epsilon_j \notin C_i, \\ 1 & \text{if } \epsilon_j \in C_i \text{ and } \epsilon_j \text{ traverse } C_i \text{ clockwise}, \\ -1 & \text{if } \epsilon_j \in C_i \text{ and } \epsilon_j \text{ traverse } C_i \text{ counterclockwise}. \end{cases}$$

It is easy to see that $\forall i, \mathbf{D}_H \vec{c_i} = 0$. Therefore, $\vec{c_i} \in \text{Null}(\mathbf{D})$. On the other hand it is easy to see that $\vec{c_i}$s are linearly independent. If c is the number of connected components of H, then $\dim(\text{Null}(\mathbf{D})) = |E_H| - |V_H| + c$ which from *Euler's formula* is equal to r. Thus, $\vec{c_i}$s form a basis for the null space of the incidence matrix $\mathbf{D}$. Now suppose $\vec{x}$ is the solution to (5.2) such that $\text{supp}(\vec{x}) = \{i | e_i \in F\}$ (from Lemma 5.5 we know that such a solution exists). We need to prove that for any $\vec{z} = \vec{x} + y_1\vec{c_1} + \cdots + y_r\vec{c_r}$, $\|\vec{z}\|_1 \leqslant \|\vec{x}\|_1$ iff $y_1 = y_2 = \cdots = y_r = 0$. Please notice that since $\vec{c_i}$s are the cycles associated with the faces of the planar graph H, each edge e_j appears in at most two cycles C_t and C_s. Thus, the entries of $\vec{z}$, z_js are in one of the following forms, (i) $z_j = 0$, (ii) $z_j = x_j$, (iii) $z_j = x_j \pm y_t$, (iv) $z_j = x_j \pm (y_t - y_s)$, or (v) $z_j = \pm(y_t - y_s)$.

First, we show that $\|\vec{z}\|_1 \geqslant \|\vec{x}\|_1$. Suppose $F = \{e_{i_1}, e_{i_2}, \ldots, e_{i_k}\}$. From the assumption, we know that F^* is H^*-separable. Thus, there are pairwise edge-disjoint cycles $O_j^* (1 \leqslant j \leqslant k)$ in H^*, such that $\forall e_{i_j} \in F, e_{i_j}^* \in O_j^*$. For each $j (1 \leqslant j \leqslant k)$ define $N_O(j) = \{p | e_p^* \in O_j^*\}$. From the definitions it is easy to see that $\forall p, q (1 \leqslant p \neq q \leqslant k), N_O(p) \cap N_O(q) = \varnothing$. Thus, $\|\vec{z}\|_1 \geqslant \sum_{j=1}^{k} \sum_{p \in N_O(j)} |z_p|$. On the other hand, from the triangle inequality, it is easy to see that for all $1 \leqslant j \leqslant k$, $\sum_{p \in N_O(j)} |z_p| \geqslant |x_{i_j}|$. Thus, $\|\vec{z}\|_1 \geqslant \sum_{j=1}^{k} \sum_{p \in N_O(j)} |z_p| \geqslant \sum_{j=1}^{k} |x_{i_j}| = \|\vec{x}\|_1$.

Now we show that $\|\vec{z}\|_1 = \|\vec{x}\|_1$ is not possible unless $y_1 = y_2 = \cdots = y_r = 0$. If $\{j_1, j_2, \ldots, j_t\}$ are the indices for which $y_{j_1} = y_{j_2} = \cdots = y_{j_t} = w$, then $y_{j_1}\vec{c_{j_1}} + \cdots + y_{j_t}\vec{c_{j_t}} = w\vec{c''}$ for which from Lemma 5.15, $\vec{c''}$ is the vector associated with the circuit $C'' = C_{j_1} \Delta C_{j_2} \Delta \ldots \Delta C_{j_t}$ (Δ is the symmetric difference). Therefore we can rewrite $\vec{z}$ as $\vec{x} + w_1\vec{c_1''} + w_2\vec{c_2''} + \cdots + w_q\vec{c_q''}$ for which $w_1 > w_2 > \cdots > w_q$. If $q = 0$, then there is nothing left to prove. If $q = 1$, then it can be easily concluded from Lemma 5.7 that it is not possible to have $\|\vec{z}\|_1 = \|\vec{x}\|_1$ unless $w_1 = 0 \Rightarrow y_1 = y_2 = \cdots = y_r = 0$. Now assume $q > 1$, from what we showed previously, we know that in order to have $\|\vec{z}\|_1 = \|\vec{x}\|_1$, we should have $\|\vec{z}\|_1 = \sum_{j=1}^{k} \sum_{p \in N_O(j)} |z_p| = \sum_{j=1}^{k} |x_{i_j}| = \|\vec{x}\|_1$. Equality

$\|\vec{z}\|_1 = \sum_{j=1}^{k} \sum_{p \in N_O(j)} |z_p|$ shows that $\text{supp}(\vec{z}) \subseteq \bigcup_{j=1}^{k} N_O(j)$. Equality $\sum_{j=1}^{k} \sum_{p \in N_O(j)} |z_p| = \sum_{j=1}^{k} |x_{i_j}|$ shows that w_1 should appear in $\|\vec{z}\|_1$, half of the time positive and half of the time negative. However, since w_1 has the largest value, it appears always as $w_1 - w_i$ in the instances like $|w_1 - w_i|$. Now from assumption we know that $|\text{supp}(\vec{x}) \cap C_1''| < |C_1''|/2$. Thus, there are more than $|C_1''|/2$ instances like $|w_1 - w_i|$ or $|w_1|$ in $\|\vec{z}\|_1$. Therefore, it is not possible that w_1 appears half of the time positive and half of the time negative in $\|\vec{z}\|_1$, which shows that if $q > 1$ then $\|\vec{z}\|_1 \neq \|\vec{x}\|_1$. Thus, the proof is complete. ■

Proof of Corollary 5.4. H is bipartite, therefore from Theorem 5.6, H^* is Eulerian. For any C in H, $|C \cap F| < |C \backslash F|$, therefore from Theorem 5.5 for any bond C^* in H^*, $|C^* \cap F^*| < |C^* \backslash F^*|$. Combining these two, from Theorem 5.4 we can conclude that F^* is H^*-separable. Thus, we can simply apply Theorem 5.2. ■

Proof of Lemma 5.8. From Corollary 5.1 we know that $\vec{\delta}_H = \vec{\theta}_H - \vec{\theta}'_H$ is a solution to $\mathbf{A}_{\bar{H}|H} \vec{\delta}_H + \mathbf{A}_{\bar{H}|\bar{H}} \vec{\delta}_{\bar{H}} = 0$. Now if $\vec{\delta} = \vec{\theta} - \vec{\theta}'$, from Lemma 5.5 we know that there exists a vector $\vec{x} \in \mathbb{R}^{|E_H|}$ such that $\text{supp}(\vec{x}) = \{i | e_i \in F\}$ and $\mathbf{D}_H \vec{x} = \mathbf{A}_{H|H} \vec{\delta}_H + \mathbf{A}_{H|\bar{H}} \vec{\delta}_{\bar{H}}$. Thus, the proof is complete. ■

Proof of Theorem 5.3. First, using Lemma 5.10, since the phase angles for the H-outer-connected nodes can be computed uniquely, without loss of generality and for simplicity, we can assume that H does not contain any H-outer-connected nodes. Suppose $\vec{c}_1, \dots \vec{c}_r$ are the vectors associated with the faces in H as we defined in the proof of Theorem 5.2. Suppose $\vec{\beta}_1, \dots, \vec{\beta}_t$ are the vectors associated with the coboundry of the H-inner-connected nodes as we defined in the proof of Lemma 5.9. All we need to prove is that if $\vec{x}$ is the solution to (5.4) such that $\text{supp}(\vec{x}) = \{i | e_i \in F\}$ then for any solution $\vec{z}$ for (5.4), $\|\vec{z}\|_1 > \|\vec{x} + y_1 \vec{c}_1 + \cdots + y_r \vec{c}_1 + w_1 \vec{\beta}_1 + \cdots + w_t \vec{\beta}_t\|_1$ unless $y_1 = \cdots = y_r = w_1 = \cdots = w_t = 0$. The rest of the proof is exactly similar to proofs of Lemma 5.9 and Theorem 5.2. The key is Condition (v) implies that the cycles ($\vec{c}_i$s) and coboundries ($\vec{b}_i$s) are edge disjoint. Thus, since all the conditions for Lemma 5.9 and Theorem 5.2 also hold here, with exactly the same approach as in the proofs of Lemma 5.9 and Theorem 5.2 we can conclude this Theorem. ■

Proof of Lemma 5.11. We show that finding the minimum matched-forest partition is at least as hard as vertex arboricity. Suppose $G = (V, E)$ is given and we are interested in $a(G)$. We build a new graph $\hat{G} = (\hat{V}, \hat{E})$ by getting a copy of graph G, $G'' = (V'', E'')$ and connect every node in G to its counterpart in G''. Thus, $\hat{V} = V \cup V''$ and $\hat{E} = E \cup E'' \cup \{\{v_i, v_i''\} | v_i \in V\}$. We prove that $\hat{G}$ has a matched-forest partition of size k if and only if $a(G) \leqslant k$. First, let us assume $\hat{G}$ has a matched-forest partition of size k, namely $\hat{H}_1, \hat{H}_2, \ldots, \hat{H}_k$. Since subgraphs induced by $\hat{H}_1, \hat{H}_2, \ldots, \hat{H}_k$ in $\hat{G}$ are acyclic and partition $\hat{V}$, it is easy to see that subgraphs induced by $\hat{H}_1 \cap V, \hat{H}_2 \cap V, \ldots, \hat{H}_k \cap V$ in G are acyclic and partition V. Thus, $a(G) \leqslant k$. Now, let us assume $a(G) \leqslant k$. There should exists a partition of nodes of G into subsets $H_1, H_2, \ldots, H_k$ such that each subset induces an acyclic graph. Assume $H_1'', H_2'', \ldots, H_k''$ are the counterparts of these subsets in G''. For any $i, 1 \leqslant i \leqslant k-1$ define $\hat{H}_i = H_i \cup H_{i+1}''$, and $\hat{H}_k = H_k \cup H_1''$. It is easy to see that $\hat{H}_1, \hat{H}_2, \ldots, \hat{H}_k$ is a matched-forest partition of size k for $\hat{G}$. Thus, we proved that $\hat{G}$ has a matched-forest partition of size k if and only if $a(G) \leqslant k$. It means that the minimum matched-forest partition of $\hat{G}$ is equal to $a(G)$, which shows that the minimum matched-forest partition is at least as hard as vertex arboricity and therefore it is NP-hard. ■

Proof of Lemma 5.14. Suppose OPT is the minimum number of matched-forest partitions of G and OPT_m is the number of optimal matching cover of G. Since for any subset V_H in the partition of $G = (V, E)$ into matched-forest partitions there exists a matching that covers V_H, we can cover V with OPT matchings. Thus, $OPT_m \leqslant OPT$. Now since G is planar, ZS Algorithm uses $3APP$ to partition Q_is into atmost 3 subsets. Hence, if k is the number of subsets returned by ZS Algorithm then $k \leqslant 3 \times 2 \times OPT_m \leqslant 6 \times OPT$. Thus, ZS Algorithm provides a 6-approximation of the minimum *matched-forest* partition of G.

As for the running time, ZS algorithm takes $O(n^3)$ time to find the optimal matching cover of G and $O(|Q_i|)$ to partition Q_i. Now since $k \leqslant n$ and $|Q_i| \leqslant n$, line 4 does not take more than $O(n^2)$. Thus, the total running time is $O(n^3)$. ■

Chapter 6

Cyber-physical Attacks (AC Model)

We follow the model that was introduced in the Chapter 5 to study cyber-physical attacks on power grids. Under this model, we assume that an adversary attacks an area by: (i) disconnecting some lines within that area (*failed lines*), and (ii) obstructing the information from within the area to reach the control center. Given the phase angles of the buses outside the attacked area (before and after the attack), our objective is to estimate the phase angles and detect the failed lines in the attacked area. Unlike the previous chapter which was based on the DC power flow model, in this chapter, we assume that the phase angles are given under the more accurate AC power flow model.

As we proved in Section 5.3, the problem of the line failures detection using phase angle measurments is combinatorial in nature, since the solution space is the discrete set of all possible line failures. Despite the complexities, we present the Convex OPtimization for Statistical State EStimation (COPSSES) Algorithm to estimate the phase angles of the buses and detect the failed lines inside the attacked area. The algorithm is based on a variation of the convex relaxation that was introduced in Section 5.6 for information recovery under the noisy DC power flow model. Here, we adapt a similar idea in the COPSSES Algorithm and show that it can estimates the phase angles and detects the line failures accurately under the AC power flow model. The novelty of our approach is the transformation of the line failures detection problem to a convex

optimization problem. Therefore, the COPSSES Algorithm can be used to detect any number of line failures without affecting its running time.

We evaluate the performance of the COPSSES Algorithm in the IEEE 118- and 300-bus systems, and show that it estimates the phase angles of the buses with less that 1% error, and can detect the line failures with 80% accuracy for single, double, and triple line failures. The algorithm can detect failures beyond triple line failures.

In a recent work [73], a linear multinomial regression model is proposed as a classifier for a single line failure detection using transient PMU data. Due to the time complexity of the learning process for more than a single line failure, this method is impractical for detecting higher order failures. To the best of our knowledge, our work is the first to provide a method for line failures detection under the AC power flow model that can be used to detect any number of line failures.

6.1 State Estimation

We can formulate the state estimation problem after a cyber-physical attack similar to the previous Chapter as follows: Given $\mathbf{A}$, $\vec{\theta}$, and $\vec{\theta}'_{\bar{H}}$, the objective is to estimate $\vec{\theta}'_{H}$ and detect F. To address this problem, we use and build on the idea that we introduced in the previous Chapter. We proved in Chapter 5 that if the phase angles of the nodes are given under the DC power flow equations, there exist vectors $\vec{x} \in \mathbb{R}^{|E_H|}$ and $\vec{\delta}_H \in \mathbb{R}^{|V_H|}$ satisfying following optimization problem for $\epsilon_1 = \epsilon_2 = 0$, and that $\text{supp}(\vec{x}) = \{i | e_i \in F\}$ and $\vec{\delta}_H := \vec{\theta}_H - \vec{\theta}'_H$:

$$\begin{aligned} &\min \|\vec{x}\|_1 \text{ s.t.} \\ &\|\mathbf{D}_H \vec{x} - \mathbf{A}_{H|H}\vec{\delta}_H - \mathbf{A}_{H|\bar{H}}\vec{\delta}_{\bar{H}}\|_2 \leqslant \epsilon_1 \\ &\|\mathbf{A}_{\bar{H}|H}\vec{\delta}_H + \mathbf{A}_{\bar{H}|\bar{H}}\vec{\delta}_{\bar{H}}\|_2 \leqslant \epsilon_2. \end{aligned} \tag{6.1}$$

where $\vec{\delta}_{\bar{H}} := \vec{\theta}_{\bar{H}} - \vec{\theta}'_{\bar{H}}$.

The novelty of this method is that it provides a convex relaxation for the line failures detection problem which is combinatorial in nature. Notice that for $\epsilon_1 = \epsilon_2 = 0$, the optimization problem (6.1) is a Linear Program (LP). We proved that under several conditions on H, the

solution to (6.1) is unique, therefore the relaxation is exact and the state of the grid can be recovered by solving (6.1) (supp($\vec{x}$) gives the failed lines and the phase angles can be computed as $\vec{\theta}'_H = \vec{\theta}_H - \vec{\delta}_H$). In particular, when H is acyclic and there is a matching between the nodes in H and $\bar{H}$ that covers H, the solution to (6.1) is unique for any set of line failures.[1]

Since the DC power flows only provides an approximation of the phase angles, it is obvious that if the phase angles of the nodes are given under the AC power flows, the optimization problem (6.1) for $\epsilon_1 = \epsilon_2 = 0$ is no longer feasible. One way to overcome this challenge is to relax the exact conditions by selecting $\epsilon_1, \epsilon_2 > 0$.

It is easy to see that if $\epsilon_1, \epsilon_2 > 0$, the optimization problem (6.1) becomes a *second-order cone program* that can still be efficiently solved using gradient decent methods.

The only challenge in using (6.1) for state estimation is that ϵ_1 and ϵ_2 need to be determined. To overcome this challenge, we present the Convex OPtimization for Statistical State EStimation (COPSSES) Algorithm. The idea is to change ϵ_i from s_i to t_i and compute the solution to (6.1) for each setup. If $\mathcal{F}$ is an array that contains all the detected line failures for each setup, then the appearance frequency of each line in $\mathcal{F}$ gives a rough probability that the line is failed. P_F denotes the appearance frequency table of the lines in $\mathcal{F}$. Moreover, the computed vector $\vec{\theta}_H - \vec{\delta}_H$ in each iteration is an estimate of the phase angles inside the attacked zone. By computing the mean and variance of all the estimated phase angle vectors in each iteration, it can improve this estimation. We refer to the mean and variance of the estimated phase angles in H as $\vec{\mu}'_H, \vec{\sigma}'_H$. The COPSSES Algorithm is summarized in Algorithm 7.

6.2 Numerical Results

In this section, we evaluate the performance of the COPSSES Algorithm. We use the *CVX* [78] for solving the optimization problem (6.1) and use MATPOWER [164] to compute $\vec{\theta}$ and $\vec{\theta}'$ under the AC power flow model.

[1]We proved in Chapter 5 that the solution to (6.1) is unique under less restricted conditions. However, for simplicity, we focus on the simplest case.

Algorithm 7 Convex OPtimization for Statistical State EStimation (COPSSES)

Input: A connected graph G, attacked zone H, $\vec{\theta}$, and $\vec{\theta}'_{\bar{H}}$

for $\epsilon_1 = s_1$ **to** t_1 **do**
  for $\epsilon_2 = s_2$ **to** t_2 **do**
    Compute $\vec{x}, \vec{\delta}_H$ the solution to (6.1) by second order cone programming;
    Compute $\vec{\theta}_H - \vec{\delta}_H$ as the estimated phase angles;
    Compute $F = \{e_i | i \in \text{supp}(\vec{x})\}$;
    $\mathcal{F} = [\mathcal{F}, F]$;
    $\Theta'_H = [\Theta'_H, \vec{\theta}_H - \vec{\delta}_H]$
Compute the appearance frequency of each line in $\mathcal{F}$ to form an appearance frequency table P_F;
Compute the row mean and variance of Θ'_H as $\vec{\mu}_H, \vec{\sigma}_H$;
return $P_F, \vec{\mu}'_H, \vec{\sigma}'_H$;

We use the IEEE 118- and 300-bus benchmark systems as the test networks [1] and consider the attacked zones H_1 and H_2 within these networks, repectively. Fig. 6.1 shows the topology of the attacked zone H_1 within the 118-bus system. Fig. 6.2 also shows the topology of the attacked zone H_2 within the 300-bus system. It is easy to see from Figs. 6.1 and 6.2 that H_1 and H_2 are both acyclic. For both of the attacked zones there is also a matching between the nodes in H_i and $\bar{H}_i$ that covers H_i.

To compute the error in the estimated phase angles, we compute $\|\vec{\mu}'_{H_i} - \vec{\theta}'_{H_i}\|_2 / \|\vec{\theta}'_{H_i}\|_2 \times 100$. Recall that $\vec{\mu}'_{H_i}$ is the vector of average estimated phase angles of the nodes in the attacked zone obtained by the COPSSES Algorithm and $\vec{\theta}'_{H_i}$ is the vector of the actual phase angles of the nodes.

To quantify the performance of the COPSSES Algorithm in detecting the failed lines, we can use the appearance frequency table P_F to detect the most likely failed lines using a threshold value t. The idea is that if a line is detected as failed in at least $(t \times 100)\%$ of the settings, then we consider that line as a line that is most likely failed. If $t = 0.5$, then the solution is similar to the *maximum likelihood* set of failures based on P_F. We consider $t = 0.2, 0.5, 0.8$.

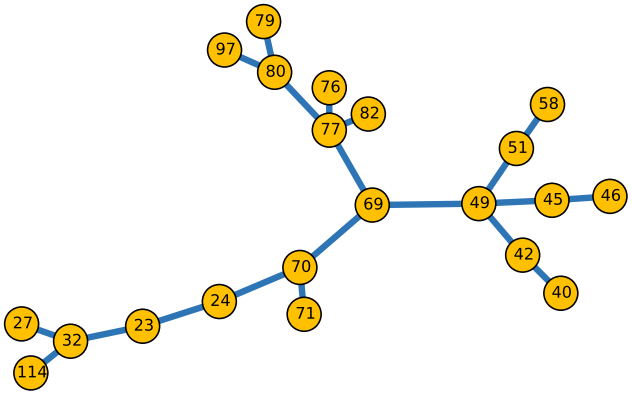


Figure 6.1: Topology of the attacked zone in the 118-bus system with 21 nodes and 22 lines (referred to as H_1).

In this section, in the COPSSES Algorithm, we use $s_1 = 3$, $t_1 = 7$, $s_2 = 1$, and $s_2 = 20$. Therefore, the COPSSES Algorithm estimates the state under 100 different settings for ϵ_1 and ϵ_2. Notice that increasing the intervals $[s_i, t_i]$ increases the accuracy at the expense of the running time.

6.2.1 Single line failures

In this subsection, we consider all possible single line failure scenarios in zones H_1 and H_2 as the failed lines.

6.2.1.1 118-bus System

For all single line failure scenarios, the error in the estimated phase angles using the COPSSES Algorithm is below 1%.

To show the results for the detected line failures, we use a heatmap matrix as in Fig. 6.3. As can be seen, in most of the cases, the correct line is detected as the most probable failed line by the COPSSES Algorithm. For example:

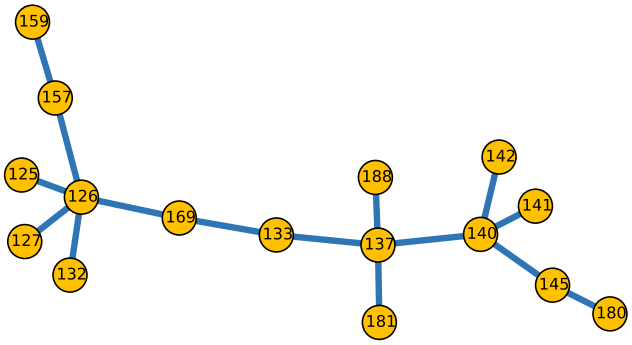


Figure 6.2: Topology of the attacked zone in the 300-bus system with 16 nodes and 15 lines (referred to as H_2).

- For $i = 1$, in 95% of the settings, line 1 is detected as the only failed line. In 5% of the settings, however, no failure is detected.

- For $i = 15$, in 100% of the settings, line 15 is detected as the failed line. However, in 3, 12, 16, 20, 20, 23, and 20% of the times, lines 12, 14, 16, 17, 18, 19, and 20 are also detected as the failed lines.

Fig. 6.4 also shows the number of false negatives and positives if we use the appearance frequency table P_F and a threshold value t to detect the most likely failed lines. As can be seen, for $t = 0.5$, for almost 80% of the cases there are no false negatives or false positives. For $t = 0.2$, for almost 95% of the cases, there are no false negatives while for almost 80% of the cases there are no false positives either.

6.2.1.2 300-bus System

In this case, the phase angle estimation is also very accurate. For all single line failure scenarios, the error in the estimated phase angles using the COPSSES Algorithm is below 2%.

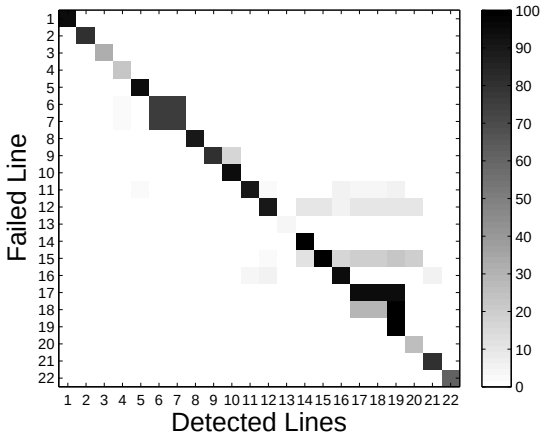


Figure 6.3: Detected line failures after all single line failures in zone H_1 within the IEEE 118-bus system. The color intensity of each (i, j) square shows the number of times line j is detected as failed when only line i is actually failed.

Fig. 6.5 shows the heatmap matrix of the failed line and detected failed lines. As can be seen, in this case also the actual failed line is detected as the most probable failed line by the COPSSES Algorithm, most of the time. For example:

- For $i = 5$, in 95% of the settings, line 5 is detected as the only failed line. In 5% of the settings, however, the optimization problem (6.1) is infeasible.

- For $i = 7$, in 75% of the settings, line 7 is detected as the failed line. However, in 40 and 13% of the settings, lines 3 and 5 are also detected as the failed lines, respectively. In 25% of the settings, (6.1) is infeasible.

As in the 118-bus system case, Fig. 6.6 shows the number of false negatives and positives using the appearance frequency table P_F and a threshold value t. As can be seen, for $t = 0.2$ and $t = 0.5$, the number of false positive and negatives is zero for most of the cases.

6.2.2 Double line failures

In this subsection, we consider all double line failure scenarios in zones H_1 and H_2 as the failed lines.

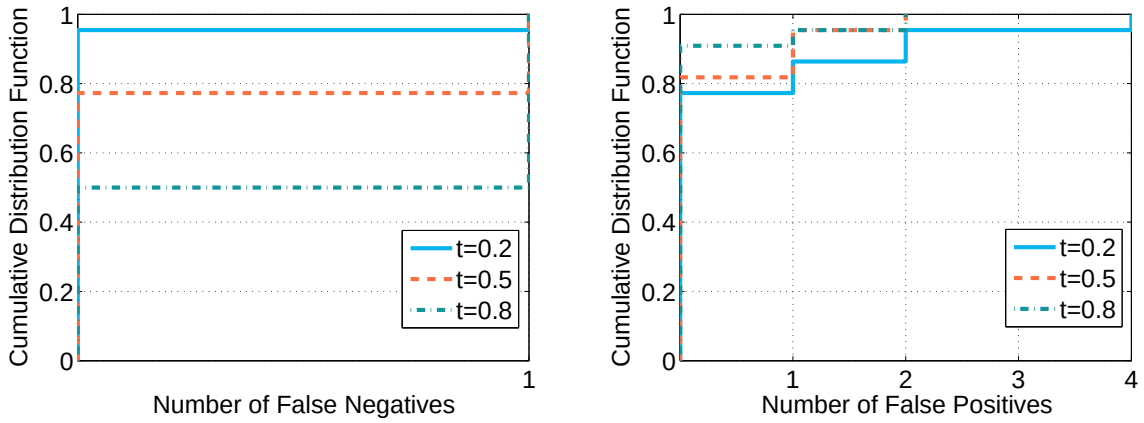


Figure 6.4: The CDF of the number of false negatives and positives in detecting single line failures in H_1 within the 118-bus system using the COPSSES Algorithm and a threshold value t.

6.2.2.1 118-bus System

In this case, as in the single line failure, the phase angle estimation is very accurate. For all double line failure scenarios, the error in the estimated phase angles using the COPSSES Algorithm is below 2%.

Since there are many double line failure cases, we cannot show the failed lines detection results as a matrix heatmap. However, as in the previous subsection, we can show the number of false negatives and positives if we use the appearance frequency table P_F and a threshold value t to detect the most likely failed lines. As can be seen in Fig. 6.7, for $t = 0.2$ for more than 80% of the cases there is no false negative. Moreover, for more than 80% of the cases there is less than a single false positive line detection.

6.2.2.2 300-bus System

In this case, as in the single line failure scenario and the 118-bus system, the phase angle estimation is very accurate. For all double line failure scenarios, the error in the estimated phase angles is below 2%.

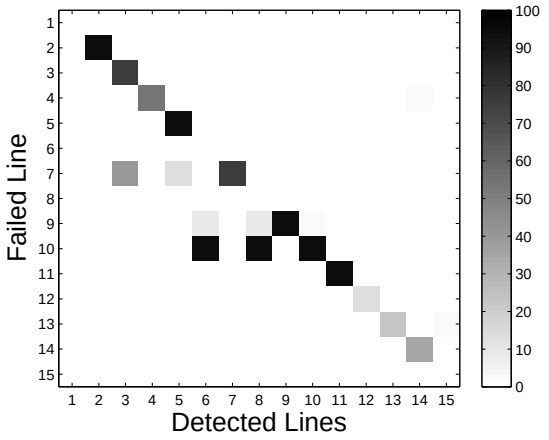


Figure 6.5: Detected line failures after all single line failures in zone H_2 within the IEEE 300-bus system. The color intensity of each (i, j) square shows the number of times line j is detected as failed when only line i is actually failed.

To show the performance of the COPSSES Algorithm in detecting failures, as in the 118-bus case, we compute the number of false negative and positive failure detections using P_F and a threshold value t. As can be seen in Fig. 6.8, in this case also for $t = 0.2$, the detection is relatively accurate. In almost 70% of the cases, there are no false negatives while in 80% of the cases, there is no false positives either.

6.2.3 Triple line failures and beyond

Here, due to the page limit, we only consider up to 3 line failures in our numerical results. However, the COPSSES Algorithm can be used to estimate the state in the attacked zone for any number of line failures.

In this subsection, we consider 100 randomly sampled triple line failures from all possible triple line failures in H_1 and H_2 as the failed lines.

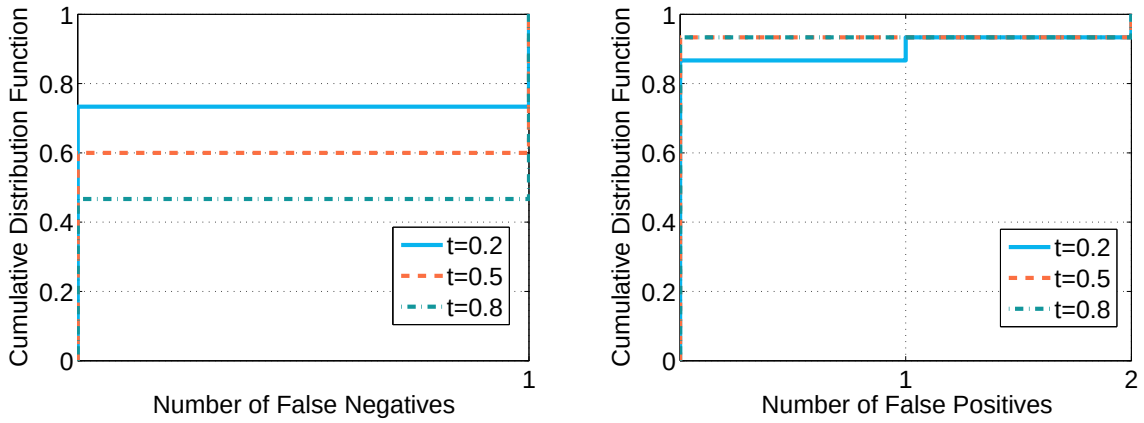


Figure 6.6: The CDF of the number of false negatives and positives in detecting single line failures in H_2 within the 300-bus system using the COPSSES Algorithm and a threshold value t.

6.2.3.1 118-bus System

The phase angle estimation is nearly perfect for the triple line failure scenarios as in the previous cases. The error for the estimated phase angles is less that 1% for all the sampled triple line failure cases.

Fig. 6.9 shows the number of false negatives and positives, if we use the appearance frequency table P_F and a threshold value t to detect the most likely failed lines. As can be seen, for $t = 0.2$, in 80% of the times there is no false negatives while in 80% of the times there is at most 1 false positive.

6.2.3.2 300-bus System

The phase angle estimation of the COPSSES Algorithm is surprisingly perfect for the triple line failures scenarios in the H_2. The error for the estimated phase angles is 0% for all the sampled triple line failure cases.

As in the previous cases, to evaluate the performance of the COPSSES Algorithm in detecting failures, we compute the number of false positives and negatives by selecting different threshold values t. As can be seen in Fig. 6.10, for $t = 0.2$, the Algorithm performs relatively well. In 70%

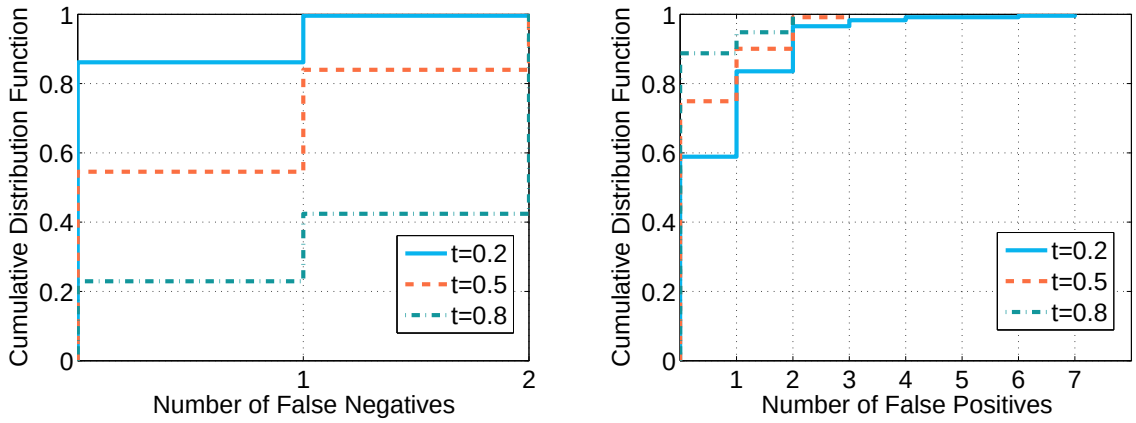


Figure 6.7: The CDF of the number of false negatives and positives in detecting double line failures in H_1 within the 118-bus system using the COPSSES Algorithm and a threshold value t.

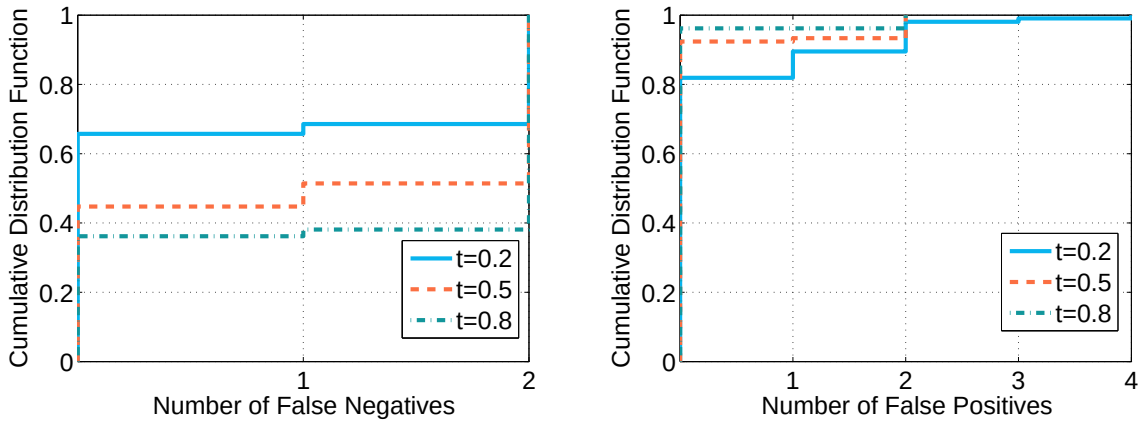


Figure 6.8: The CDF of the number of false negatives and positives in detecting double line failures in H_2 within the 300-bus system using the COPSSES Algorithm and a threshold value t.

of the cases, there is no false negatives while for more than 80% of the cases, there is no false positives either.

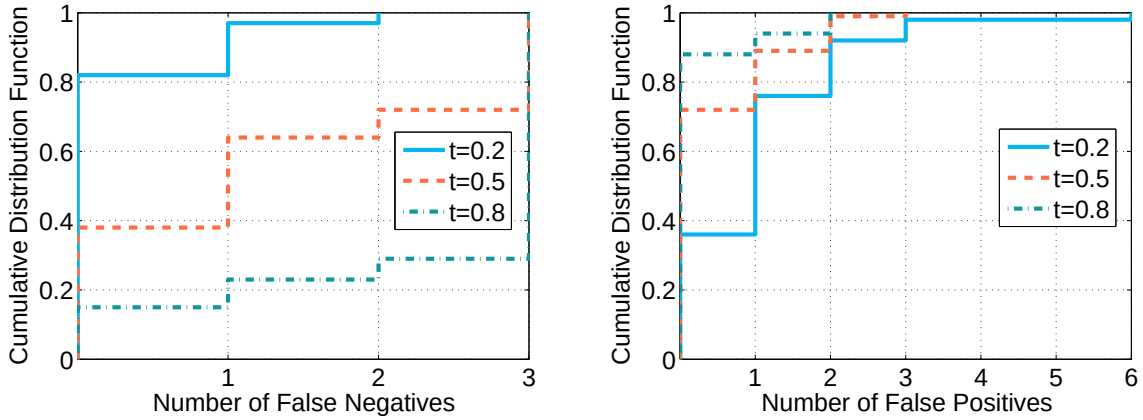


Figure 6.9: The CDF of the number of false negatives and positives in detecting triple line failures in H_1 within the 118-bus system using the COPSSES Algorithm and a threshold value t. 100 randomly sampled triple line failures are considered.

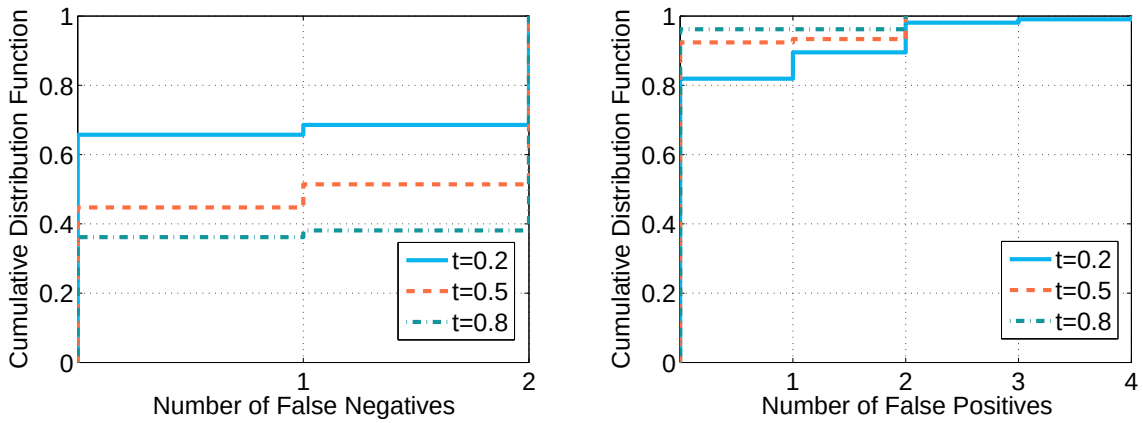


Figure 6.10: The CDF of the number of false negatives and positives in detecting triple line failures in H_2 within the 300-bus system using the COPSSES Algorithm and a threshold value t. 100 randomly sampled triple line failures are considered.

6.3 Conclusion

We provided an algorithm to estimate the state of the grid following a cyber-physical attack under the AC power flow model. We studied its performance under different scenarios (single, double, and triple line failures) in IEEE 118- and 300-bus systems and showed that it can estimate the phase angles almost perfectly (with less that 1% error) in these scenarios. Moreover, we showed that our algorithm can detect line failures with less that 20% chance of producing false positives and negatives.

We believe that the COPSSES Algorithm can accurately estimate the state of the grid for less constrained attacked zones as well as in different scenarios such as false data injection. Exploring these directions can be of interest to researchers.

Chapter 7

Power Grid Islanding

Power Grid Islanding is an effective method to mitigate cascading failures in power grids [140]. The objective is to partition the network into smaller connected components, called *islands*, such that each island can operate independently for a while. Power gird islanding is performed in the early stages of a cascading failure to prevent the system to be separated into unbalanced parts. It also expedites the restoration of the grid after a major failure, by reducing transient stability problems during system reconnection.

In order for an island to operate, it is necessary that the power supply and demand at that island are almost equal.[1] Equality of supply and demand in an island, however, may not be sufficient for its independent operation. It is also important that the infrastructure in that island has the physical capacity to safely transfer the power from the supply nodes to the demand nodes. When the island is large enough compared to the initial network, it is more likely that it has enough capacity. This problem has been studied in the power systems community but almost all the algorithms provided in the literature are heuristic methods that have been shown to be effective only by simulations [71, 107, 121, 139].

Motivated by this application, we formally introduce and study the Doubly Balanced Connected graph Partitioning (DBCP) problem: Let $G = (V, E)$ be a connected graph with a

[1] If the supply and demand are not exactly equal but still relatively close, load shedding/generation curtailing can be used in order for the island to operate.

weight (supply/demand) function $p : V \to \mathbb{Z}$ satisfying $p(V) = \sum_{j \in V} p(j) = 0$. The objective is to partition V into (V_1, V_2) such that $G[V_1]$ and $G[V_2]$ are connected, $|p(V_1)|, |p(V_2)| \leqslant c_p$, and $\max\{\frac{|V_1|}{|V_2|}, \frac{|V_2|}{|V_1|}\} \leqslant c_s$, for some constants c_p and c_s. We also consider the case that $p(V) \neq 0$, in which the excess supply/demand should be split roughly evenly.

The problem calls for a partition into two connected subgraphs that simultaneously balances two objectives, (1) the supply/demand within each part, and (2) the sizes of the parts. The connected partitioning problem with only the size objective has been studied previously. In the most well-known result, Lováz and Gyori [80, 99] independently proved, using different methods, that every k-connected graph can be partitioned into k arbitrarily sized connected subgraphs. However, neither of the proofs is constructive, and there are no known polynomial-time algorithms to find such a partition for $k > 3$. For $k = 2$, a linear time algorithm is provided in [141] and for $k = 3$ an $O(|V|^2)$ algorithm is provided in [149].[2] The complexity of the problem with the size objective and related optimization problems have been studied in [43, 48, 65] and there are various NP-hardness and inapproximability results. Note that the size of the cut is not of any relevance here (so the extensive literature on finding balanced partitions, not necessarily connected, that minimize the cut is not relevant.)

The objective of balancing the supply/demand alone, when all $p(i)$ are ± 1, can also be seen as an extension for the objective of balancing the size (which corresponds to $p(i) = 1$). Our bi-objective problem of balancing both supply/demand and size, can be seen also as an extension of the problem of finding a partition that balances the size for two types of nodes simultaneously: Suppose the nodes of a graph are partitioned into red and blue nodes. Find a partition of the graph into two large connected subgraphs that splits approximately evenly both the red and the blue nodes.

We now summarize our results and techniques. Since the power grids are designed to withstand a single failure ("$N - 1$" standard) [29], and therefore 2-connected, our focus is mainly on the graphs that are at least 2-connected. We first, in Section 7.4, study the connected partitioning problem with only the supply/demand balancing objective, and show results that parallel

[2]For $k = 2$, a much simpler approach than the one in [141] is to use the *st*-numbering [94] for 2-connected graphs.

the results for balancing size alone, using similar techniques: The problem is NP-hard in general. For 2-connected graphs and weights $p(i) = \pm 1$, there is always a perfectly balanced partition and we can find it easily using an st-numbering. For 3-connected graphs and weights $p(i) = \pm 1$, there is a perfectly balanced partition into three connected graphs, and we can find it using a nonseparating ear decomposition of 3-connected graphs [46] and similar ideas as in [149].

The problem is more challenging when we deal with both balancing objectives, supply/demand and size. This is the main focus and occupies the bulk of this chapter. Our main results are existence results and algorithms for 2- and 3-connected graphs. It is easy to observe that we cannot achieve perfection in one objective ($c_p = 0$ or $c_s = 1$) without sacrificing completely the other objective. We show that allowing the supply/demand of the parts to be off balance by at most the weight of one node suffices to get a partition that is roughly balanced also with respect to size.

First, in Section 7.4.1, we study the case of 3-connected graphs since we use this later as the basis of handling 2-connected graphs. We show that if $\forall i,\ p(i) = \pm 1$, there is a partition that is perfectly balanced with respect to both objectives, if $|V| \equiv 0 (\text{mod } 4)$ (otherwise the sizes are slightly off for parity reasons); for general p, the partition is perfect in both objectives up to the weight of a single node. Furthermore, the partition can be constructed in polynomial time. Our approach uses the convex embedding characterization of k-connectivity studied by Linial, Lováz, and Wigderson [95]. We need to adapt it for our purposes so that the convex embedding also has certain desired geometric properties, and for this purpose we use the nonseparating ear decomposition of 3-connected graphs of [46] to obtain a suitable embedding.

Then, in Section 7.4.2, we analyze the case of 2-connected graphs. We reduce it to two subcases: either (1) there is a separation pair that splits the graph into components that are not very large, or (2) we can perform a series of contractions to achieve a 3-connected graph whose edges represent contracted subgraphs that are not too large. We provide a good partitioning algorithm for case (1), and for case (2) we extend the algorithms for 3-connected graphs to handle also the complications arising from edges representing contracted subgraphs. Finally, in Section 7.5, we briefly discuss the problem of finding a connected partitioning of a graph with two types of nodes that splits roughly evenly both types.

7.1 Related Work

Power grid islanding is widely studied in power engineering community [71, 121, 140, 146]. Efficient search algorithms for splitting the grid into two balanced (in terms of supply and demand values) parts is studied in [140, 160]. These algorithm are shown to be effective for small grids. However, since they depend on searching the entire solution space they are impractical in larger grids.

Islanding methods using mixed integer programming were studied in [71, 107]. These methods consider operational constraints as well as balancing conditions on the islands to partition the network into two or more islands. Therefore, from engineering view point, these methods consider the most realistic constraints between all previous works. However, since these methods rely on integer variables, they do not guarantee to find the optimal solution in polynomial time. Moreover, no approximation bounds are provided for these algorithms. These algorithms are shown to be effective for small networks by simulations.

In a recent series of works, spectral clustering techniques in graph theory are used for power grid islanding [69,109,121]. Although spectral clustering methods can be implemented efficiently, since these methods are not bounded by any balancing or operational constraints, the solutions obtained by these methods are not guaranteed to results in balanced or operable islands. These methods are shown to perform well by simulations on small networks.

Finally, power grid islanding is also used to group synchronized generators together at the time of frequency disturbance events. Effective heuristic methods with no guarantee of performance are provided to partition the grid during these events in [97, 146, 158].

7.2 Preliminaries

In this section, we provide a short overview of the definitions and tools used in our work. Most of the graph theoretical terms used in this chapter are relatively standard and borrowed from [36] and [154].

7.2.1 Terms from Graph Theory

Cutpoints and Subgraphs: A *cutpoint* of a connected graph G is a node whose deletion results in a disconnected graph. Let X and Y be subsets of the nodes of a graph G. $G[X]$ denotes the subgraph of G *induced* by X. We denote by $E[X, Y]$ the set of edges of G with one end in X and the other end in Y.

Connectivity: The connectivity of a graph $G = (V, E)$ is the minimum size of a set $S \subset V$ such that $G \backslash S$ is not connected. A graph is k-connected if its connectivity is at least k.

7.2.2 *st*-numbering of a Graph

Given any edge $\{s, t\}$ in a 2-connected graph G, an *st-numbering* for G is a numbering for the nodes in G defined as follows [94]: the nodes of G are numbered from 1 to n so that s receives number 1, node t receives number n, and every node except s and t is adjacent both to a lower-numbered and to a higher-numbered node. It is shown in [70] that such a numbering can be found in $O(|V| + |E|)$.

7.2.3 Series-Parallel Graphs

A Graph G is *series-parallel*, with terminals s and t, if it can be produced by a sequence of the following operations:

1. Create a new graph, consisting of a single edge between s and t.
2. Given two series parallel graphs, X and Y with terminals s_X, t_X and s_Y, t_Y respectively, form a new graph $G = P(X, Y)$ by identifying $s = s_X = s_Y$ and $t = t_X = t_Y$. This is known as the *parallel composition* of X and Y.
3. Given two series parallel graphs X and Y, with terminals s_X, t_X and s_Y, t_Y respectively, form a new graph $G = S(X, Y)$ by identifying $s = s_X, t_X = s_Y$ and $t = t_Y$. This is known as the *series composition* of X and Y.

It is easy to see that a series-parallel graph is 2-connected if, and only if, the last operation is a parallel composition.

7.2.4 Nonseparating Induced Cycles and Ear Decomposition

Let H be a subgraph of a graph G. An *ear* of H in G is a nontrivial path in G whose ends lie in H but whose internal vertices do not. An ear decomposition of G is a decomposition $G = P_0 \cup \cdots \cup P_k$ of the edges of G such that P_0 is a cycle and P_i for $i \geqslant 1$ is an ear of $P_0 \cup P_1 \cup \cdots \cup P_{i-1}$ in G. It is known that every 2-connected graph has an ear decomposition (and vice-versa), and such a decomposition can be found in linear time.

A cycle C is a *nonseparating induced cycle* of G if $G \backslash C$ is connected and C has no chords. We say a cycle C avoids a node u, if $u \notin C$.

Theorem 7.1 (Tutte [145]). *Given a 3-connected graph $G(V, E)$ let $\{t, r\}$ be any edge of G and let u be any node of G, $r \neq u \neq t$. Then there is a nonseparating induced cycle of G through $\{t, r\}$ and avoiding u.*

Notice that since G is 3-connected in the previous theorem, every node in C has a neighbor in $G \backslash C$. Cheriyan and Maheshwari showed that the cycle in Theorem 7.1 can be found in $O(E)$ [46]. Moreover, they showed that any 3-connected graph G has a nonseparating ear decomposition $G = P_0 \cup \cdots \cup P_k$ defined as follows: Let $V_i = V(P_0) \cup V(P_1) \cdots \cup V(P_i)$, let $G_i = G[V_i]$ and $\overline{G}_i = G[V \backslash V_i]$. We say that $G = P_0 \cup P_1 \cup \cdots \cup P_k$ is an ear decomposition through edge $\{t, r\}$ and avoiding vertex u if the cycle P_0 contains edge $\{t, r\}$, and the last ear of length greater than one, say P_m, has u as its only internal vertex. An ear decomposition $P_0 \cup P_1 \cdots \cup P_k$ of graph G through edge $\{t, r\}$ and avoiding vertex u is a *nonseparating ear decomposition* if for all i, $0 \leqslant i < m$, graph $\overline{G}_i$ is connected and each internal vertex of ear P_i has a neighbor in $\overline{G}_i$.

Theorem 7.2 (Cheriyan and Maheshwari [46]). *Given an edge $\{t, r\}$ and a vertex u of a 3-connected graph G, a nonseparating induced cycle of G through $\{t, r\}$ and avoiding u, and a nonseparating ear decomposition can be found in time $O(|V| + |E|)$.*

7.2.5 Partitioning of Graphs to Connected Subgraphs

The following theorem is the main existing result in partitioning of graphs into connected subgraphs and is proved independently by Lováz and Gyori [80, 99] by different methods.

Theorem 7.3 (Lováz and Gyori [80, 99]). *Let $G = (V, E)$ be a k-connected graph. Let $n = |V|$, $v_1, v_2, \ldots, v_k \in V$ and let $n_1, n_2, \ldots, n_k$ be positive integers satisfying $n_1 + n_2 + \cdots + n_k = n$. Then, there exists a partition of V into $(V_1, V_2 \ldots, V_k)$ satisfying $v_i \in V_i, |V_i| = n_i$, and $G[V_i]$ is connected for $i = 1, 2, \ldots, k$.*

Although the existence of such a partition has long been proved, there is no polynomial-time algorithm to find such a partition for $k > 3$. For $k = 2$, it is easy to find such partition using *st*-numbering. For $k = 3$, Wada and Kawaguchi [149] provided an $O(n^2)$ algorithm using the nonseparating ear decomposition of 3-connected graph.

7.2.6 Convex Embedding of Graphs

In this subsection, we provide a short overview of the beautiful work by Linial, Lováz, and Wigderson [95] on convex embedding of the k-connected graphs. Let $Q = \{q_1, q_2, \ldots, q_m\}$ be a finite set of points in $\mathbb{R}^d$. The convex hull $\text{conv}(Q)$ of Q is the set of all points $\sum_{i=1}^{m} \lambda_i q_i$ with $\sum_{i=1} \lambda_i = 1$. The rank of Q is defined by $\text{rank}(Q) = 1 + \dim(\text{conv}(Q))$. Q is in general position if $\text{rank}(S) = d + 1$ for every $(d + 1)$-subset $S \subseteq Q$. Let G be a graph and $X \subset V$. A convex X-embedding of G is any mapping $f : V \to \mathbb{R}^{|X|-1}$ such that for each $v \in V \backslash X$, $f(v) \in \text{conv}(f(N(v)))$. We say that the convex embedding is in general position if the set $f(V)$ of the points is in general position.

Theorem 7.4 (Linial, Lováz, and Wigderson [95]). *Let G be a graph on n vertices and $1 < k < n$. Then the following two conditions are equivalent:*

1. *G is k-connected*

2. *For every $X \subset V$ with $|X| = k$, G has a convex X-embedding in general position.*

Notice that the special case of the Theorem for $k = 2$ asserts the existence of an *st*-numbering of a 2-connected graph. The proof of this theorem is inspired by physics. The embedding is found by letting the edges of the graph behave like ideal springs and letting its vertices settle. A formal summary of the proof $(1 \to 2)$ is as follows (for more details see [95]). For each $v_i \in X$, define $f(v_i)$ arbitrary in $\mathbb{R}^{k-1}$ such that $f(X)$ is in general position. Assign to every

edge $(u, v) \in E$ a positive elasticity coefficient c_{uv} and let $c \in \mathbb{R}^{|E|}$ be the vector of coefficients. It is proved in [95] that for almost any coefficient vector c, an embedding f that minimizes the potential function $\mathrm{P} = \sum_{\{u,v\}\in E} c_{uv}\|f(u) - f(v)\|_2^2$ provides a convex X-embedding in general position. Moreover, the embedding that minimizes P can be computed as follows,

$$f(v) = \frac{1}{c_v} \sum_{u\in N(v)} c_{uv} f(u) \text{ for all } v \in V\backslash X,$$

in which $c_v = \sum_{u\in N(v)} c_{uv}$. Hence, the embedding can be found by solving a set of linear equations in at most $O(|V|^3)$ time (or matrix multiplication time).

7.3 Balancing the Supply/Demand Only

In this section, we study the single objective problem of finding a partition of the graph into connected subgraphs that balances (approximately) the supply and demand in each part of the partition, without any regard to the sizes of the parts. We can state the optimization problem as follows, and will refer to it as the Balanced Connected Partitioning with Integer weights (BCPI) problem.

Definition 7.1. *Given a graph $G = (V, E)$ with a weight (supply/demand) function $p : V \to \mathbb{Z}$ satisfying $\sum_{j\in V} p(j) = 0$. The BCPI problem is the problem of partitioning V into (V_1, V_2) such that*

1. *$V_1 \cap V_2 = \varnothing$ and $V_1 \cup V_2 = V$,*
2. *$G[V_1]$ and $G[V_2]$ are connected,*
3. *$|p(V_1)| + |p(V_2)|$ is minimized, where $p(V_i) = \sum_{j\in V_i} p(j)$.*

Clearly, the minimum possible value for $|p(V_1)| + |p(V_2)|$ that we can hope for is 0, which occurs iff $p(V_1) = p(V_2) = 0$. It is easy to show that the problem of determining whether there exists such a 'perfect' partition (and hence the BCPI problem) is strongly NP-hard. The proof is very similar to analogous results concerning the partition of a graph into two connected subgraphs with equal sizes (or weights, when nodes have positive weights) [43, 65]

Proposition 7.1. *(1) It is strongly NP-hard to determine whether there is a solution to the BCPI problem with value 0, even when G is 2-connected.*
(2) If G is not 2-connected, then this problem is NP-hard even when $\forall i, p(i) = \pm 1$.

Proof. We use the proof of [43, Theorem 2] with a modest change. The reduction is from the X3C problem [108], which is a variant of the *Exact Cover by 3-sets* and defined as follows: Given a set X with $|X| = 3q$ and a family C of 3-element subsets of X such that $|C| = 3q$ and each element of X appears in exactly 3 sets of C, decide whether C contains an exact cover for X. Given an instance (X, C) of $X3C$, let $G = (V, E)$ be the graph with the vertex set $V = X \cup C \cup \{a, b\}$ and edge set $E = \bigcup_{j=1}^{3q}[\{C_j x_i | x_i \in C_j\} \cup \{C_j a\} \cup \{C_j b\}]$. Set $p(a) = 2q$, $p(b) = 9q^2 + q$, $p(C_j) = -1$, and $p(x_i) = -3q$. It is easy to verify that C contains an exact cover for X if and only if the BCPI problem has a solution such that $p(V_1) = p(V_2) = 0$. This shows the first claim.

For the second claim, attach to nodes a, b, and the x_is, paths of length $2q$, $9q^2 + q$, and $3q$, respectively, and set the supply/demand values of a, b, the x_i's and the new nodes equal to $+1$ (for the paths for a and b) or -1 (for the x_i's). ■

Although it is NP-hard to tell whether there is a solution satisfying $p(V_1) = p(V_2) = 0$, even when $\forall i, p(i) = \pm 1$, in this case, if the graph G is 2-connected there is always such a solution. For general weights p, there is a solution such that $|p(V_1)|, |p(V_2)| \leqslant \max_{j \in V} |p(j)|/2$ and it can be found easily in linear time using the st-numbering between two nodes.

Proposition 7.2. *Let G be a 2-connected graph and u, v any two nodes in V such that $p(u)p(v) > 0$.*
(1) There is a solution such that $u \in V_1$, $v \in V_2$, and $|p(V_1)| = |p(V_2)| \leqslant \max_{j \in V} |p(j)|/2$.
(2) If $\forall i, p(i) = \pm 1$, we can find a solution such that $u \in V_1$, $v \in V_2$, and $p(V_1) = p(V_2) = 0$.
In both cases, the solution can be found in $O(|E|)$ time.

Proof. Clearly, part (2) follows immediately from part (1) because in this case, $p(V_1), p(V_2)$ are integer and $\max_{j \in V} |p(j)|/2 = 1/2$. To show part (1), pick two arbitrary nodes $u, v \in V$ with $p(u)p(v) > 0$. Since we want to separate u and v, we can assume wlog that initially $\{u, v\} \in G$.

Since G is 2-connected, an st-numbering between nodes u and v as $u = v_1, v_2, \ldots, v_n = v$ can be found in $O(|V|+|E|)$ [70]. Define $V_1^{(i)} := \{v_1, v_2, \ldots, v_i\}$. It is easy to see that $p(V_1^{(1)}) = p(u) > 0$ and $p(V_1^{(n-1)}) = -p(v) < 0$. Hence, there must exist an index $1 \leqslant i^* < n$ such that $|p(V_1^{(i^*)})| > 0$ and $|p(V_1^{(i^*+1)})| \leqslant 0$. Since $|p(V_1^{(i)}) - p(V_1^{(i+1)})| = |p(i^*+1)|$, either $|p(V_1^{(i^*)})| \leqslant |p(i^*+1)|/2$ or $|p(V_1^{(i^*+1)})| \leqslant |p(i^*+1)|/2$; Accordingly set $V_1 = V_1^{(i^*)}$ or $V_1 = V_1^{(i^*+1)}$. Let $V_2 = V \backslash V_1$. Hence, (V_1, V_2) is a solution with $|p(V_1)| = |p(V_2)| \leqslant |p(i^*+1)/2| \leqslant \max_{j \in V} |p(j)|/2$. It is easy to see that i^* can be found in $O(|V|)$. ■

Remark 7.1. *The bound in Proposition 7.2 (1) is tight. A simple example is a cycle of length 4 like (v_1, v_2, v_3, v_4) with $p(v_1) = -p$, $p(v_2) = -p/2$, $p(v_3) = p$, and $p(v_4) = p/2$. It is easy to see that in this example $|p(V_1)| + |p(V_2)| = \max_{j \in V} |p(j)| = p$ is the best that one can do.*

7.3.1 Connected Partitioning into Many Parts

The BCPI problem can be extended to partitioning a graph into $k = 3$ or more parts. Let $G = (V, E)$ be a graph with a weight function $p : V \to \mathbb{Z}$ satisfying $\sum_{j \in V} p(j) = 0$. The BCPI$_k$ problem is the problem of partitioning G into $(V_1, V_2, \ldots, V_k)$ such that for any $1 \leqslant i \leqslant k$, $G[V_i]$ is connected and $\sum_{i=1}^{k} |p(V_i)|$ is minimized.

In the following proposition, we show that for $k = 3$, if $p(i) = \pm 1, \forall i$, then there is always a perfect partition (i.e., with $p(V_1) = p(V_2) = p(V_3) = 0$) and it can be found efficiently. For general p, we can find a partition such that $|p(V_1)| + |p(V_2)| + |p(V_3)| \leqslant 2 \max_{j \in V} |p(j)|$. The proof and the algorithm use a similar approach as the algorithm in [149] for partitioning a 3-connected graph to three connected parts with prescribed sizes, using the nonseparating ear decomposition of 3-connected graphs as described in Subsection 7.2.4.

Proposition 7.3. *Let G be a 3-connected graph and u, v, w three nodes in V such that $p(u), p(v), p(w) > 0$ or $p(u), p(v), p(w) < 0$.*

(1) There is a solution such that $u \in V_1$, $v \in V_2$, $w \in V_3$, and $|p(V_1)| + |p(V_2)| + |p(V_3)| \leqslant 2 \max_{j \in V} |p(j)|$.

(2) If $\forall i, p(i) = \pm 1$, then there is a solution such that $u \in V_1$, $v \in V_2$, $w \in V_3$, and $|p(V_1)| =$

$|p(V_2)| = |p(V_3)| = 0$.

In both cases, the solution can be found in $O(|E|)$ time.

Proof. Consider the case of general function p, and let $p_{\max} = \max_{j \in V} |p(j)|$. We will show that we can find a solution such that $u \in V_1, v \in V_2, w \in V_3$ with $|p(V_1)|, |p(V_2)| \leqslant p_{\max}/2$. Since $|p(V_3)| = |p(V_1) + p(V_2)|$ (recall $p(V) = 0$), this implies that $|p(V_3)| \leqslant p_{\max}$, and hence $|p(V_1)| + |p(V_2)| + |p(V_3)| \leqslant 2p_{\max}$. Furthermore, if $p(i) = \pm 1$ for all $i \in V$, hence $p_{\max} = 1$, then $|p(V_1)|, |p(V_2)| \leqslant p_{\max}/2$ implies that $p(V_1) = p(V_2) = 0$, and therefore also $p(V_3) = 0$. Thus, both claims will follow.

Assume $u, v, w \in V$ and $p(u), p(v), p(w) > 0$ (the case of negative $p(u), p(v), p(w)$ is symmetric). Since we want to separate u from v, we can assume without loss of generality that $\{u, v\} \in E$. Using [46], there is a non-separating ear decomposition through the edge $\{u, v\}$ and avoiding node w. Ignore the ears that do not contain any internal nodes, and let $Q_0 \cup Q_1 \cup \cdots \cup Q_r$ be the decomposition consisting of the ears with nodes; we have $w \in Q_r$. Let $V_i = V(Q_0) \cup V(Q_1) \cdots \cup V(Q_i)$, let $G_i = G[V_i]$ and $\overline{G}_i = G[V \backslash V_i]$. We distinguish two cases, depending on whether $p(V_0) \leqslant 0$ or $p(V_0) > 0$.

(i) If $p(V_0) \leqslant 0$, then consider an *st*-numbering between u and v in V_0, say $u = v_1, v_2, \ldots, v_s = v$. Define $V_0^{(i)} = \{v_1, v_2, \ldots, v_i\}$. Since $p(u), p(v) > 0$ and $p(V_0) \leqslant 0$, there must exist indices $1 \leqslant i^* \leqslant j^* < s$ such that $p(V_0^{(i^*)}) > 0$, $p(V_0^{(i^*+1)}) \leqslant 0$ and $p(V_0 \backslash V_0^{(j^*+1)}) > 0$, $p(V_0 \backslash V_0^{(j^*)}) \leqslant 0$.

(a) If $i^* = j^*$, since $p(V_0^{(i^*)}) + p(v_{i^*+1}) + p(V_0 \backslash V_0^{(i^*+1)}) = p(V_0) < 0$, we have $p(V_0^{(i^*)}) + p(V_0 \backslash V_0^{(i^*+1)}) \leqslant |p(v_{i^*+1})|$. Now, one of the following three cases happens:

- If $p(V_0^{(i^*)}) \leqslant |p(v_{i^*+1})|/2$ and $p(V_0 \backslash V_0^{(i^*+1)}) \leqslant |p(v_{i^*+1})|/2$, then it is easy to see that $V_1 = V_0^{(i^*)}$, $V_2 = V_0 \backslash V_0^{(i^*+1)}$, and $V_3 = V \backslash (V_1 \cup V_2)$ is a good partition.
- If $p(V_0^{(i^*)}) > |p(v_{i^*+1})|/2$ and $p(V_0 \backslash V_0^{(i^*+1)}) \leqslant |p(v_{i^*+1})|/2$, then $p(V_0^{(i^*)}) + p(v_{i^*+1}) = p(V_0^{(i^*+1)}) \leqslant |p(v_{i^*+1})|/2$. Hence, $V_1 = V_0^{(i^*+1)}$, $V_2 = V_0 \backslash V_0^{(i^*+1)}$, and $V_3 = V \backslash V_0$ is a good partition.

- If $p(V_0^{(i^*)}) \leqslant |p(v_{i^*+1})|/2$ and $p(V_0 \backslash V_0^{(i^*+1)}) > |p(v_{i^*+1})|/2$, then $p(V_0 \backslash V_0^{(i^*+1)}) + p(v_{i^*+1}) = p(V_0 \backslash V_0^{(i^*)}) \leqslant |p(v_{i^*+1})|/2$. Hence, $V_1 = V_0^{(i^*)}$, $V_2 = V_0 \backslash V_0^{(i^*)}$, and $V_3 = V \backslash V_0$ is a good partition.

(b) If $i^* < j^*$, then either $p(V_0^{(i^*)}) \leqslant |p(v_{i^*+1})|/2$ or $|p(V_0^{(i^*+1)})| \leqslant |p(v_{i^*+1})|/2$, accordingly set $V_1 = V_0^{(i^*)}$ or $V_1 = V_0^{(i^*+1)}$. Similarly, either $p(V_0 \backslash V_0^{(j^*+1)}) \leqslant |p(v_{j^*+1})|/2$ or $|p(V_0 \backslash V_0^{(j^*)})| \leqslant |p(v_{j^*+1})|/2$, so accordingly set $V_2 = V_0 \backslash V_0^{(j^*+1)}$ or $V_2 = V_0 \backslash V_0^{(j^*)}$. Set $V_3 = V \backslash (V_1 \cup V_2)$. It is easy to check that (V_1, V_2, V_3) is a good partition.

(ii) If $p(V_0) > 0$, then since $p(w) > 0$ and therefore $p(V_{r-1}) < 0$, there must exist an index $0 \leqslant j < r-1$ such that $p(V_j) > 0$ and $p(V_{j+1}) \leqslant 0$. Consider an st-numbering between u and v in $G[V_j]$ as $u = v_1, v_2, \ldots, v_s = v$ and define $V_j^{(i)} = \{v_1, v_2, \ldots, v_i\}$. The ear Q_{j+1} is a path of new nodes $q_1, q_2, \ldots, q_t$ attached to two (distinct) nodes v_x, v_y of $G[V_j]$ through edges $\{v_x, q_1\}, \{q_t, v_y\} \in E$; assume wlog that $1 \leqslant x < y \leqslant s$. For simplicity, we will use below Q_{j+1} to denote also the set $\{q_1, q_2, \ldots, q_t\}$ of internal (new) nodes of the ear. Also define $Q_{j+1}^{(i)} = \{q_1, q_2, \ldots, q_i\}$ and $Q_{j+1}^{(0)} = \varnothing$. One of the following cases must happen:

(a) Suppose there is an index $1 \leqslant i^* < (y-1)$ such that $p(V_j^{(i^*)}) > 0$ and $p(V_j^{(i^*+1)}) \leqslant 0$ or there is an index $x+1 < i^* < s$ such that $p(V_j \backslash V_j^{(i^*-1)}) > 0$ and $p(V_j \backslash V_j^{(i^*)}) \leqslant 0$. Let's assume there is an index $1 \leqslant i^* < (y-1)$, such that $p(V_j^{(i^*)}) > 0$ and $p(V_j^{(i^*+1)}) \leqslant 0$ (the other case is exactly similar). Then either $p(V_j^{(i^*)}) \leqslant |p(v_{i^*+1})|/2$ or $|p(V_j^{(i^*+1)})| \leqslant |p(v_{i^*+1})|/2$, accordingly set either $V_1 = V_j^{(i^*)}$ or $V_1 = V_j^{(i^*+1)}$. Set $V_2' = V_j \backslash V_1$. One of the following cases happens:

- If $V_1 = V_j^{(i^*)}$ and $p(V_2') \leqslant 0$, then since $p(V_j^{(i^*+1)}) \leqslant 0$, we have $p(V_j \backslash V_j^{(i^*+1)}) > 0$. Hence, $p(V_2' \backslash \{v_{i^*+1}\}) > 0$. So, it is either $|p(V_2')| \leqslant |p(v_{i^*+1})|/2$ or $p(V_2' \backslash \{v_{i^*+1}\}) \leqslant |p(v_{i^*+1})|/2$. Now if $p(V_2' \backslash \{v_{i^*+1}\}) \leqslant |p(v_{i^*+1})|/2$, since also $p(V_1) \leqslant |p(v_{i^*+1})|/2$, $p(V_j) \leqslant 0$ which contradicts with the assumption. Therefore, $|p(V_2')| \leqslant |p(v_{i^*+1})|/2$. Set $V_2 = V_2'$ and $V_3 = V \backslash (V_1 \cup V_2)$. It is easy to check that (V_1, V_2, V_3) is a good partition.
- If $V_1 = V_j^{(i^*)}$ and $p(V_2') > 0$, then since $p(V_j \cup Q_{j+1}) < 0$, there is an index $0 < t^* \leqslant t$, such that $p(V_2' \cup (Q_{j+1} \backslash Q_{j+1}^{(t^*)})) > 0$ and $p(V_2' \cup (Q_{j+1} \backslash Q_{j+1}^{(t^*-1)})) \leqslant 0$.

Hence, either $p(V_2' \cup (Q_{j+1} \backslash Q_{j+1}^{(t^*)})) \leqslant |p(q_{t^*})|/2$ or $|p(V_2' \cup (Q_{j+1} \backslash Q_{j+1}^{(t^*-1)}))| \leqslant |p(q_{t^*})|/2$, accordingly set $V_2 = V_2' \cup (Q_{j+1} \backslash Q_{j+1}^{(t^*)})$ or $V_2 = V_2' \cup (Q_{j+1} \backslash Q_{j+1}^{(t^*-1)})$. Set $V_3 = V \backslash (V_1 \cup V_2)$. It is easy to see that (V_1, V_2, V_3) is a good partition.

- If $V_1 = V_j^{(i^*+1)}$, then since $p(V_1) \leqslant 0$, we have $p(V_2') > 0$. The rest is exactly like the previous case when $V_1 = V_j^{(i^*)}$ and $p(V_2') > 0$.

(b) Suppose that for every $1 \leqslant i < y$, $p(V_j^{(i)}) > 0$ and for every $x < i < s$, $p(V_j \backslash V_j^{(i)}) > 0$. Set $V_1' = V_j^{(y-1)}$ and $V_2' = V_j \backslash V_1'$. Based on the assumption $p(V_1'), p(V_2') > 0$. Since $p(V_{j+1}) \leqslant 0$, there are indices $0 \leqslant i^* \leqslant j^* < t$ such that $p(V_1' \cup Q_{j+1}^{(i^*)}) > 0$, $p(V_1' \cup Q_{j+1}^{(i^*+1)}) \leqslant 0$ and $p(V_2' \cup (Q_{j+1} \backslash Q_{j+1}^{(j^*+1)})) > 0$, $p(V_2' \cup (Q_{j+1} \backslash Q_{j+1}^{(j^*)})) \leqslant 0$. The rest of the proof is similar to case (i) when $p(V_0) \leqslant 0$.

■

7.4 Balancing Both Objectives

In this section, we formally define and study the Doubly Balanced Connected graph Partitioning (DBCP) problem.

Definition 7.2. *Given a graph $G = (V, E)$ with a weight (supply/demand) function $p : V \to \mathbb{Z}$ satisfying $p(V) = \sum_{j \in V} p(j) = 0$ and constants $c_p \geqslant 0$, $c_s \geqslant 1$. The* DBCP problem *is the problem of partitioning V into (V_1, V_2) such that*

1. *$V_1 \cap V_2 = \varnothing$ and $V_1 \cup V_2 = V$,*
2. *$G[V_1]$ and $G[V_2]$ are connected,*
3. *$|p(V_1)|, |p(V_2)| \leqslant c_p$ and $\max\{\frac{|V_1|}{|V_2|}, \frac{|V_2|}{|V_1|}\} \leqslant c_s$, where $p(V_i) = \sum_{j \in V_i} p(j)$.*

Remark 7.2. *Our techniques apply also to the case that $p(V) \neq 0$. In this case, the requirement 3 on $p(V_1)$ and $p(V_2)$ is $|p(V_1) - p(V)/2|, |p(V_2) - p(V)/2| \leqslant c_p$, i.e., the excess supply/demand is split approximately evenly between the two parts.*

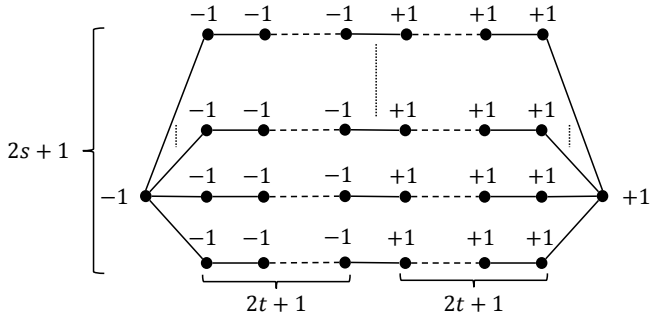


Figure 7.1: Series-parallel graphs with $2s + 1$ paths of length $4t + 2$ used in Observations 7.1 and 7.2.

We will concentrate on 2-connected and 3-connected graphs and show that we can construct efficiently good partitions. For most of the section we will focus on the case that $p(i) = \pm 1, \forall i \in V$. This case contains all the essential ideas. All the techniques generalize to the case of arbitrary p, and we will state the corresponding theorems.

We observed in Section 2 that if the graph is 2-connected and $p(i) = \pm 1, \forall i \in V$ then there is always a connected partition that is perfect with respect to the weight objective, $p(V_1) = p(V_2) = 0$, i.e., (3) is satisfied with $c_p = 0$. We know also from [80, 99] that there is always a connected partition that is perfect with respect to the size objective, $|V_1| = |V_2|$, i.e., condition 3 is satisfied with $c_s = 1$. The following observations show that combining the two objectives makes the problem more challenging. If we insist on $c_p = 0$, then c_s cannot be bounded in general, (it will be $\Omega(|V|)$), and if we insist on $c_s = 1$, then c_p cannot be bounded. The series-parallel graphs of Figure 7.1 provide simple counterexamples.

Observation 7.1. *If $c_p = 0$, then for any $c_s < |V|/2 - 1$, there exist a 2-connected graph G such that the DBCP problem does not have a solution even when $\forall i, p(i) = \pm 1$.*

Proof. In the graph depicted in Figure 7.1, set $t = 0$. ■

Observation 7.2. *If $c_s = 1$, then for any $c_p < |V|/6$, there exist a 2-connected graph G such that the DBCP problem does not have a solution even when $\forall i, p(i) = \pm 1$.*

Proof. In the graph depicted in Figure 7.1, set $s = 1$. ■

Thus, c_p has to be at least 1 to have any hope for a bounded c_s. We show in this section that $c_p = 1$ suffices for all 2-connected graphs. We first treat 3-connected graphs.

7.4.1 3-Connected Graphs

Let $G = (V, E)$ be a 3-connected graph. Assume for the most of this section that $\forall i, p(i) = \pm 1$ and $p(V) = 0$ (we will state the results for general p at the end). We show that G has a partition that is essentially perfect with respect to both objectives, i.e., with $c_p = 0$ and $c_s = 1$. We say "essentially", because $p(V_1) = p(V_2) = 0$ and $|V_1| = |V_2|$ imply that $|V_1| = |V_2|$ are even, and hence V must be a multiple of 4. If this is the case, then indeed we can find such a perfect partition. If $|V| \equiv 2(\text{mod } 4)$ ($|V|$ has to be even since $p(V) = 0$), then we can find an 'almost perfect' partition, one in which $|p(V_1)| = |p(V_2)| = 1$ and $|V_1| = |V_2|$ (or one in which $p(V_1) = p(V_2) = 0$ and $|V_1| = |V_2| + 2$).

We first treat the case that G contains a triangle (i.e., cycle of length 3). In the following Lemma, we use the embedding for k-connected graphs introduced in [95] and as described in Subsection 7.2.6, to show that if G is 3-connected with a triangle and all weights are ± 1, then the DBCP problem has a perfect solution.

Lemma 7.1. *If G is 3-connected with a triangle, $\forall i, p(i) = \pm 1$, and $|V| \equiv 0(\text{mod } 4)$, then there exists a solution to the DBCP problem with $p(V_1) = p(V_2) = 0$ and $|V_1| = |V_2|$. If $|V| \equiv 2(\text{mod } 4)$, then there is a solution with $p(V_1) = p(V_2) = 0$ and $|V_1| = |V_2| + 2$. Moreover, this partition can be found in polynomial time.*

Proof. Assume that $|V| \equiv 0(\text{mod } 4)$; the proof for the case $|V| \equiv 2(\text{mod } 4)$ is similar. In [95] as described in Subsection 7.2.6, it is proved that if G is a k-connected graph, then for every $X \subset V$ with $|X| = k$, G has a convex X-embedding in general position. Moreover, this embedding can be found by solving a set of linear equations of size $|V|$. Now, assume $v, u, w \in V$ form a triangle in G. Set $X = \{v, u, w\}$. Using the theorem, G has a convex X-embedding $f : V \to \mathbb{R}^2$ in general position. Consider a circle $\mathcal{C}$ around the triangle $f(u), f(v), f(w)$ in $\mathbb{R}^2$ as shown in an example in Fig. 7.2. Also consider a directed line $\mathcal{L}$ tangent to the circle $\mathcal{C}$ at point A. If we project the nodes of G onto the line $\mathcal{L}$, since the embedding is convex and also $\{u, v\}, \{u, w\}, \{w, v\} \in E$, the

order of the nodes' projection gives an st-numbering between the first and the last node (notice that the first and last nodes are always from the set X). For instance in Fig. 7.2, the order of projections give an st-numbering between the nodes u and v in G. Hence, if we set V_1 to be the $|V|/2$ nodes whose projections come first and V_2 are the $|V|/2$ nodes whose projections come last, then $G[V_1]$ and $G[V_2]$ are both connected and $|V_1| = |V_2| = |V|/2$. The only thing that may not match is $p(V_1)$ and $p(V_2)$. Notice that for each directed line tangent to the circle $\mathcal{C}$, we can similarly get a partition such that $|V_1| = |V_2| = |V|/2$. So all we need is a point D on the circle $\mathcal{C}$ such that if we partition based on the directed line tangent to $\mathcal{C}$ at point D, then $p(V_1) = p(V_2) = 0$. To find such a point, we move $\mathcal{L}$ from being tangent at point A to point B (AB is a diameter of the circle $\mathcal{C}$) and consider the resulting partition. Notice that if at point A, $p(V_1) > 0$, then at point B since V_1 and V_2 completely switch places compared to the partition at point A, $p(V_1) < 0$. Hence, as we move $\mathcal{L}$ from being tangent at point A to point B and keep it tangent to the circle, in the resulting partitions, $p(V_1)$ goes from some positive value to a non-positive value. Notice that the partition (V_1, V_2) changes only if $\mathcal{L}$ passes a point D on the circle such that at D, $\mathcal{L}$ is perpendicular to a line that connects $f(i)$ to $f(j)$ for some $i, j \in V$. Now, since the embedding is in general position, there are exactly two points on every line that connects two points $f(i)$ and $f(j)$, so V_1 changes at most by one node leaving V_1 and one node entering V_1 at each step as we move $\mathcal{L}$. Hence, $p(V_1)$ changes by either ± 2 or 0 value at each change. Now, since $|V| \equiv 0 (\text{mod } 4)$, $p(V_1)$ has an even value in all the resulting partitions. Therefore, as we move $\mathcal{L}$ from being tangent at point A to point B, there must be a point D such that in the resulting partition $p(V_1) = p(V_2) = 0$.

Note that the order of the projected points and V_1 may change only when a line that passes through 2 nodes of graph G is perpendicular to $\mathcal{L}$. We can sort first the slopes of all the lines connecting two nodes of G (or alternatively we can use a priority queue) and then rotate the line $\mathcal{L}$ from the initial position A until we find the point D that yields a partition with $p(V_1) = p(V_2) = 0$ in $O(|V|^2 \log |V|)$ time. ■

When G is a triangle-free 3-connected graph, however, the proof in Lemma 7.1 cannot be directly used anymore. The reason is if for example $\{u, v\} \notin E$ and we project the nodes of G onto

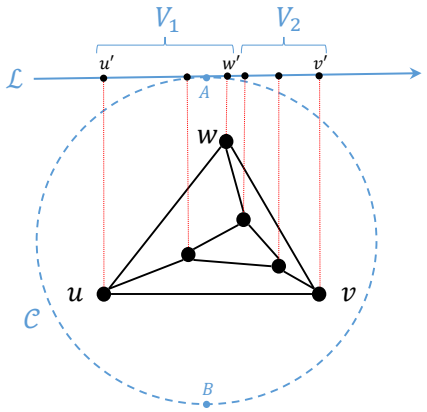


Figure 7.2: Proof of Lemma 7.1.

the line $\mathcal{L}$, this time the order of the nodes projection does not give an st-numbering between the first and the last node if for example u and w are the first and last node, since some of the middle nodes may only be connected to v. To prove a similar result for triangle-free 3-connected case, we first provide the following two Lemmas. The main purpose of the following two Lemmas are to compensate for the triangle-freeness of G in the proof of Lemma 7.1. The idea is to show that in every 3-connected graph, there is a triple $\{u, w, v\} \in V$, such that $\{u, w\}, \{w, v\} \in E$ and in every partition that we get by the approach used in the proof of Lemma 7.1, if u and v are in V_i, so is a path between u and v.

Lemma 7.2. *If G is 3-connected, then there exists a set $\{u, v, w\} \in V$ and a partition of V into (V_1', V_2') such that:*

1. *$V_1' \cap V_2' = \varnothing$ and $V_1' \cup V_2' = V$,*
2. *$G[V_1']$ and $G[V_2']$ are connected,*
3. *$\{u, w\}, \{v, w\} \in E$,*
4. *$w \in V_1'$, $u, v \in V_2'$, and u, v are not cutpoints of $G[V_2']$,*
5. *$|V_2'| \leqslant |V|/2$.*

Moreover, such a partition and $\{u, v, w\}$ can be found in $O(|E|)$ time.

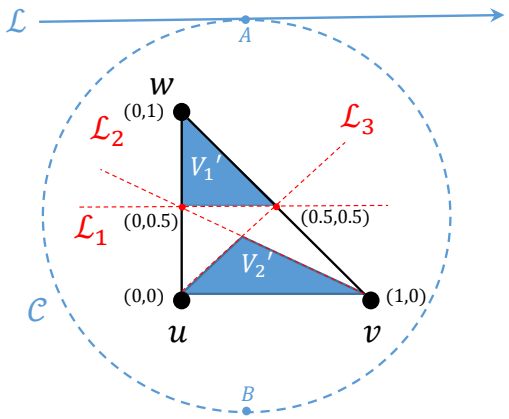


Figure 7.3: Proof of Lemma 7.3 and Theorem 7.5.

Proof. Using the algorithm presented in [46], we can find a non-separating cycle C_0 in G such that every node in C_0 has a neighbor in $G \backslash C_0$ in $O(|E|)$ time. Now, we consider two cases:

(i) If $|C_0| \leqslant |V|/2 + 1$, then select any three consecutive nodes (u, w, v) of C_0 and set $V_2' = C_0 \backslash \{w\}$ and $V_1' = V \backslash V_2'$.

(ii) If $|C_0| > |V|/2 + 1$, since every node in C_0 has a neighbor in $G \backslash C_0$, there exists a node $w \in V \backslash C_0$ such that $|N(w) \cap C_0| \geqslant 2$. Select two nodes $u, v \in N(w) \cap C_0$. There exists a path P in C_0 from u to v such that $|P| < |V|/2 + 1$. Set $V_2' = P$ and $V_1' = V \backslash V_2'$. ■

Lemma 7.3. *Given a partition (V_1', V_2') of a 3-connected graph G with following properties:*

1. *$V_1' \cap V_2' = \varnothing$ and $V_1' \cup V_2' = V$,*
2. *$G[V_1']$ and $G[V_2']$ are connected,*
3. *$w \in V_1'$, $u, v \in V_2'$, and u, v are not cutpoints of $G[V_2']$,*

G has a convex X-embedding in general position with mapping $f : V \to \mathbb{R}^2$ such that:

1. *$X = \{u, w, v\}$, $f(u) = (0, 0)$, $f(v) = (1, 0)$, and $f(w) = (0, 1)$,*
2. *Every node i in V_1' is mapped to a point $(f_1(i), f_2(i))$ with $f_2(i) \geqslant 1/2$,*

3. *Every node i in V_2' is mapped to a point $(f_1(i), f_2(i))$ with $f_1(i) \geqslant f_2(i)$ and $f_1(i)+2f_2(i) \leqslant 1$.*

Moreover, such an embedding can be found in in polynomial time.

Proof. Set $X = \{v, u, w\}$. Using [95], G has a convex X-embedding in $\mathbb{R}^2$ in general position with mapping $f : V \to \mathbb{R}^2$ such that $f(u) = (0, 0)$, $f(v) = (1, 0)$, and $f(w) = (0, 1)$. In the X-embedding of the nodes, we have a freedom to set the elasticity coefficient vector $\vec{c}$ to anything that we want (except a measure zero set of vectors). So for any edge $\{i, j\} \in G[V_1'] \cup G[V_2']$, set $c_{ij} = g$; and for any $\{i, j\} \in E[V_1', V_2']$, set $c_{ij} = 1$. Assume $\mathcal{L}_1$ is the line $y = 0.5$, $\mathcal{L}_2$ is the line $x + 2y = 1$, and $\mathcal{L}_3$ is the line $x = y$.

First, we show that there exist a g for which all the nodes in V_1' will be embedded above the line $\mathcal{L}_1$. To show this, from [95], we know the embedding is such that it minimizes the total potential $\mathrm{P}(f, \vec{c}) = \sum_{\{i,j\}\in E} c_{ij}\|f(i) - f(j)\|^2$. Notice that we can independently minimize P on x-axis values and y-axis values as below:

$$\begin{aligned}\min_f \mathrm{P} &= \min_{f_1} \mathrm{P}_x + \min_{f_2} \mathrm{P}_y \\ &= \min_{f_1} \sum_{\{i,j\}\in E} c_{ij}(f_1(i) - f_1(j))^2 + \min_{f_2} \sum_{\{i,j\}\in E} c_{ij}(f_2(i) - f_2(j))^2\end{aligned}$$

Now, notice that if we place all the nodes in V_1' at point (0,1) and all the nodes in V_2' on the line uv, then $\mathrm{P}_y \leqslant |E|$. Hence, if f_2 minimizes P_y, then $\mathrm{P}_y(f_2, c) \leqslant |E|$. Set $g \geqslant 4|V|^2|E|$. We show that if f_2 minimizes P_y, then for all edges $\{i, j\} \in G[V_1'] \cup G[V_2']$, $(f_2(i) - f_2(j))^2 \leqslant 1/(4|V|^2)$. By contradiction, assume there is an edge $\{i, j\} \in G[V_1'] \cup G[V_2']$ such that $(f_2(i) - f_2(j))^2 > 1/(4|V|^2)$. Then, $c_{ij}(f_2(i) - f_2(j))^2 = g(f_2(i) - f_2(j))^2 > |E|$. Hence, $\mathrm{P}_y(f_2, c) > |E|$ which contradicts with the fact the f_2 minimizes P_y. Therefore, if $g \geqslant 4|V|^2|E|$, then for all $\{i, j\} \in G[V_1'] \cup G[V_2']$, $|f_2(i) - f_2(j)| \leqslant 1/(2|V|)$. Now, since $G[V_1']$ is connected, all the nodes in V_1' are connected to w with a path of length (in number of hops) less than $|V| - 1$. Hence, using the triangle inequality, for all $i \in V_1'$:

$$|f_2(w) - f_2(i)| \leqslant (|V| - 1)/(2|V|) < 1/2 \Rightarrow |1 - f_2(i)| < 1/2,$$

which means that all the nodes in V_1' are above $\mathcal{L}_1$.

With the very same argument, if $g \geqslant t^2|V|^2|E|$, then for all $i \in V_2'$, $f_2(i) < 1/t$.

Now, we want to prove that there is a g such that all the nodes in V_2' will be embedded below the lines $\mathcal{L}_2$ and $\mathcal{L}_3$. Define $n_1(i) := |N(i) \cap V_1'|$ and $n_2(i) := |N(i) \cap V_2'|$. From [95], we know the embedding is such that for all $i \in V \backslash \{u, v, w\}$, $f(i) = 1/c_i \sum_{j \in N(i)} c_{ij} f(j)$, where $c_j = \sum_{j \in N(i)} c_{ij} f(j)$. Since $G[V_2']$ is connected and u is not a cutpoint of $G[V_2']$, for every $i \in V_2' \backslash \{u, v\}$ there is a path $i = v_1, v_2, \ldots, v_r = v$ in V_2' not containing node u. Using this ordering:

$$\begin{cases} f_1(v_j) \geqslant \frac{1}{n_2(v_j)g + n_1(v_j)} g f_1(v_{j+1}) \geqslant (1/|V|) f_1(v_{j+1}), & \forall j \in \{1, \ldots, r-1\} \\ f_1(v_r) = f_1(v) = 1 \end{cases}$$
$$\Rightarrow \forall i \in V_2' \backslash \{u, v\} : \ f_1(i) \geqslant (1/|V|)^r \geqslant (1/|V|)^{|V|}.$$

On the other hand, from the previous part, if we set $g \geqslant |V|^{2|V|+2}|E|$, then for all $i \in V_2'$, $f_2(i) \leqslant (1/|V|)^{|V|}$. Hence, for all $i \in V_2'$, $f_2(i) \leqslant f_1(i)$, which means that all the nodes in V_2' will be placed below the line $\mathcal{L}_3$.

With the very same idea, we show that there exist a g for which all the nodes in V_2' will be placed below the line $\mathcal{L}_2$. Since $G[V_2']$ is connected and v is not a cutpoint of $G[V_2']$, for every $i \in V_2' \backslash \{u, v\}$ there is a path $u = u_1, u_2, \ldots, u_t = i$ in V_2' not containing node v. Notice that for all $i \in V \backslash \{u, v, w\}$, $1 - f_1(i) = 1/c_i \sum_{j \in N(i)} c_{ij}(1 - f_1(j))$. Hence, since $\forall j \in V : f_1(j) \leqslant 1$, we have,

$$\begin{cases} 1 - f_1(u_j) \geqslant \frac{1}{n_2(u_j)g + n_1(u_j)} g(1 - f_1(u_{i-1})) \geqslant (1/|V|)(1 - f_1(u_{i-1})), & \forall j \in \{2, \ldots, t\} \\ 1 - f_1(u) = 1 - f_1(u_1) = 1 \end{cases}$$
$$\Rightarrow \forall i \in V_2' \backslash \{u, v\} : \ 1 - f_1(i) \geqslant (1/|V|)^t \geqslant (1/|V|)^{|V|}.$$

From the previous part, if we set $g \geqslant 4|V|^{2|V|+2}|E|$, then for all $i \in V_2'$, $f_2(i) \leqslant 1/2(1/|V|)^{|V|}$. Hence, for $i \in V_2'$, $f_1(i) + 2f_2(i) \leqslant 1$, which means that all the nodes in V_2' will be placed below the line $\mathcal{L}_3$. Therefore, if we set $g \geqslant 4|V|^{2|V|+2}|E|$, then we will get an embedding as depicted in Fig. 7.3. Note that a polynomial number of bits suffices for g.

Notice that if $\vec{c}$ is a "good" vector, then so is $\vec{c} + \vec{\epsilon}$ in which $\vec{\epsilon}$ is a vector with very small Euclidean norm. Hence, we can always find a "good" vector $\vec{c}$ which results in a X-embedding in general position. ■

Using Lemmas 7.2 and 7.3, we are now able to prove that for any 3-connected graph G such that all the weights are ± 1, the DBCP problem has a solution for $c_p = 0$ and $c_s = 1$. The idea of the proof is similar to the proof of Lemma 7.1, however, we need to use Lemma 7.2 to find a desirable partition (V_1', V_2') and then use this partition to find an embedding with properties as described in Lemma 7.3. By using this embedding, we can show that in every partition that we obtain by the approach in the proof of Lemma 7.1, if u and v are in V_i, so is a path between u and v. This implies then that $G[V_1]$ and $G[V_2]$ are connected. So we can use similar arguments as in the proof of Lemma 7.1 to prove the following theorem.

Theorem 7.5. *If G is 3-connected, $\forall i, p(i) = \pm 1$, and $|V| \equiv 0 (\text{mod } 4)$, then there exists a solution to the DBCP problem with $p(V_1) = p(V_2) = 0$ and $|V_1| = |V_2|$. If $|V| \equiv 2 (\text{mod } 4)$, then there is a solution with $p(V_1) = p(V_2) = 0$ and $|V_1| = |V_2| + 2$. Moreover, this partition can be found in polynomial time.*

Proof. Assume that $|V| \equiv 0 (\text{mod } 4)$; the proof for the case $|V| \equiv 2 (\text{mod } 4)$ is similar. Using Lemma 7.2, we can find $\{u, v, w\} \in V$ and a partition (V_1', V_2') of V with properties described in the Lemma. Set $X = \{u, v, w\}$. Using Lemma 7.3, we can find a convex X-embedding of G in general position with properties described in the Lemma as depicted in Fig. 7.3. The rest of the proof is very similar to the proof of Lemma 7.1. We consider again a circle $\mathcal{C}$ around $f(u), f(v), f(w)$ in $\mathbb{R}^2$ as shown in Fig. 7.3. Also consider a directed line $\mathcal{L}$ tangent to the circle C at point A. If we project the nodes of G onto the line $\mathcal{L}$, this time the order of the nodes projection gives an st-numbering between the first and the last node only if u and v are the first and last node. However, if we set V_1 to be the $|V|/2$ nodes whose projections come first and V_2 are the $|V|/2$ nodes whose projections come last, then $G[V_1]$ and $G[V_2]$ are both connected even when u and v are not the first and last nodes. The reason lies on the special embedding that we considered here. Assume for example w and v are the first and the last projected nodes, and V_1

and V_2 are set of the $|V|/2$ nodes which projections come first and last, respectively. Two cases might happen:

(i) If $u, w \in V_1$ and $v \in V_2$, then since $\{u, w\} \in E$, both $G[V_1]$ and $G[V_2]$ are connected because of the properties of the embedding.

(ii) If $w \in V_1$ and $u, v \in V_2$, since $|V_2'| \leqslant |V|/2$ and $|V_2| = |V|/2$, then either $V_2 = V_2'$ or $V_2 \cap V_1' \neq \varnothing$. If $V_2 = V_2'$, and hence $V_1 = V_1'$ then there is nothing to prove. So assume there is a node $z \in V_2 \cap V_1'$. From the properties of the embedding, the triangle $\{z, u, v\}$ contains all the nodes of V_2'. Since $\{z, u, v\} \in V_2$, and V_2 contains all the nodes that are on a same side of a halfplane, we should also have $V_2' \subset V_2$. Now, from the properties of the embedding, it is easy to see that every node in V_2 has a path either to u or v. Since $V_2' \subset V_2$, there is also a path between u and v. Thus, $G[V_2]$ is connected. From the properties of the embedding, $G[V_1]$ is connected as before.

The rest of the proof is exactly the same as the proof of Lemma 7.1. We move $\mathcal{L}$ from being tangent at point A to point B (AB is a diameter of the circle $\mathcal{C}$) and consider the resulting partition. Notice that if at point A, $p(V_1) > 0$, then at point B since V_1 and V_2 completely switch places compared to the partition at point A, $p(V_1) < 0$. Hence, as we move $\mathcal{L}$ from being tangent at point A to point B and keep it tangent to the circle, in the resulted partitions, $p(V_1)$ goes from some positive value to a negative value. Notice that the partition (V_1, V_2) changes only if $\mathcal{L}$ passes a point D on the circle such that at D, $\mathcal{L}$ is perpendicular to a line that connects $f(i)$ to $f(j)$ for $i, j \in V$. Now, since the embedding is in general position, there are exactly two points on every line that connects two points $f(i)$ and $f(j)$, so V_1 changes at most by one node leaving V_1 and one node entering V_1. Hence, $p(V_1)$ changes by either ± 2 or 0 value at each change. Now, since $|V| \equiv 0 (\text{mod } 4)$, $p(V_1)$ has an even value in all the resulting partitions. Therefore, as we move $\mathcal{L}$ from being tangent at point A to point B, there should be a point D such that in the resulted partition $p(V_1) = p(V_2) = 0$.

■

It is easy to check for a 3-connected graph G, by using the same approach as in the proof of Lemma 7.1 and Theorem 7.5, that even when the weights are arbitrary (not necessarily ± 1)

and also $p(V) \neq 0$, we can still find a connected partition (V_1, V_2) for G such that $|p(V_1) - p(V)/2|, |p(V_1) - p(V)/2| \leqslant \max_{i \in V} |p(i)|$ and $|V_1| = |V_2|$.

Corollary 7.1. *If G is 3-connected, then the DBCP problem (with arbitrary p, and not necessarily satisfying $p(V) = 0$) has a solution for $c_p = \max_{i \in V} |p(i)|$ and $c_s = 1$. Moreover, this solution can be found in polynomial time.*

7.4.2 2-Connected Graphs

We first define a *pseudo-path* between two nodes in a graph as below. The definition is inspired by the definition of the *st*-numbering.

Definition 7.3. *A* pseudo-path *between nodes u and v in $G = (V, E)$, is a sequence of nodes $v_1, \ldots, v_t$ such that if $v_0 = u$ and $v_{t+1} = v$, then for any $1 \leqslant i \leqslant t$, v_i has neighbors v_j and v_k such that $j < i < k$. Note that the pseudo-path does not include the ending points u and v.*

Using the pseudo-path notion, in the following lemma we show that if G is 2-connected and has a separation pair such that none of the resulting components are too large, then the DBCP problem always has a solution for some $c_p = c_s = O(1)$. The idea used in the proof of this lemma is one of the building blocks of the proof for the general 2-connected graph case.

Lemma 7.4. *Given a 2-connected graph G, if $\forall i : p(i) = \pm 1$ and G has a separation pair $\{u, v\} \subset V$ such that for every connected component $H_i = (V_{H_i}, E_{H_i})$ of $G[V \backslash \{u, v\}]$, $|V_{H_i}| < \lfloor 2|V|/3 \rfloor$, then the DBCP problem has a solution for $c_p = 1$, $c_s = 2$, and it can be found in $O(|E|)$ time.*

Proof. There is a separation pair $\{u, v\} \in V$ such that if $H_1, \ldots, H_k$ are the connected components of $G \backslash \{u, v\}$, for every i, $|V_{H_i}| < \lfloor 2|V|/3 \rfloor$. Since G is 2-connected, $H_1, \ldots, H_k$ can be represented by pseudo-paths $P_1, \ldots, P_k$ between u and v. Assume $P_1, \ldots, P_k$ are in increasing order based on their lengths. We can partition the pseudo-paths into two subsets S_1 and S_2 such that $S_1 \cap S_2 = \varnothing$, $S_1 \cup S_2 = \{P_1, \ldots, P_k\}$ and $\sum_{P_j \in S_i} |P_j| \geqslant \lceil |V|/3 \rceil - 1$ for $i = 1, 2$. The proof is very simple. Add greedily pseudo-paths in order to S_1 until $\sum_{P_j \in S_1} |P_j|$ becomes at least $\lceil |V|/3 \rceil - 1$. Let $S_1 = \{P_1, \ldots, P_i\}$, and $S_2 = \{P_{i+1}, \ldots, P_k\}$. Since $|P_k| < \lfloor 2|V|/3 \rfloor$, we have $i < k$

and $S_2 \neq \varnothing$. We have to show that $\sum_{P_j \in S_2} |P_j| \geqslant \lceil |V|/3 \rceil - 1$. If $|P_k| \geqslant \lceil |V|/3 \rceil - 1$, then the claim really holds. If $|P_k| < \lceil |V|/3 \rceil - 1$, then $|P_i| \leqslant |P_k| < \lceil |V|/3 \rceil - 1$, and $|P_1| + \cdots + |P_{i-1}| < \lceil |V|/3 \rceil - 1$ implies that $|P_{i+1}| + \cdots + |P_k| \geqslant \lceil |V|/3 \rceil - 1$.

Now, if we put all the pseudo-paths in S_1 back to back, they will form a longer pseudo-path Q_1 between u and v. Similarly, we can form another pseudo-path Q_2 from the pseudo-paths in S_2 (Fig. 7.4a). Without loss of generality we can assume $|Q_1| \geqslant |Q_2|$. From u, including u itself, we count $\lceil |V|/3 \rceil$ of the nodes in Q_1 towards v and put them in a set V'. Without loss of generality, we can assume $p(V') \geqslant 0$. If $p(V') = 0$, then $(V', V \backslash V')$ is a good partition and we are done. Hence, assume $p(V') > 0$. We keep V' fixed and make a new set V'' by adding nodes from Q_1 to V' one by one before we get to v. If $p(V'')$ hits 0 as we add nodes one by one, we stop and let $V_1 = V''$ and $V_2 = V \backslash V''$, then (V_1, V_2) is a good partition and we are done (Fig. 7.4b).

So, assume $V'' = Q_1 \cup \{u\}$ and $p(V'') > 0$. Since $|Q_2 \cup \{v\}| \geqslant \lceil |V|/3 \rceil$, $|V''| \leqslant \lfloor 2|V|/3 \rfloor$. If $|V''| < \lceil 2|V|/3 \rceil$, we add nodes from $Q_2 \cup \{v\}$ one by one toward u until either $p(V'') = 0$ or $|V''| = \lceil 2|V|/3 \rceil$. If we hit 0 first, or at the same time, i.e., $p(V'') = 0$ and $|V''| \leqslant \lceil 2|V|/3 \rceil$), define $V_1 = V'' \backslash \{u\}$; then $(V_1, V \backslash V_1)$ is a good partition (Fig. 7.4c). So assume $|V''| = \lceil 2|V|/3 \rceil$ and $p(V'') > 0$. Define $V''' = V \backslash V''$. Since $p(V'') > 0$ and $|V''| = \lceil 2|V|/3 \rceil$, then $p(V''') < 0$ and $|V'''| = \lfloor |V|/3 \rfloor$. Also notice that $V''' \subseteq Q_2$. We consider two cases. Either $|p(V')| \geqslant |p(V''')|$ or $|p(V')| < |p(V''')|$.

If $|p(V')| \geqslant |p(V''')|$ we start from u and pick nodes one by one from Q_1 in order until we get a subset $V_1' \neq \varnothing$ of V' such that $p(V_1') = |p(V''')|$ (i.e., $p(V_1' \cup V''') = 0$). Define $V_1 = V_1' \cup V'''$. Then $p(V_1) = 0$, $|V_1| \geqslant |V'''| + 1 \geqslant \lceil |V|/3 \rceil$ and $|V_1| \leqslant 2\lfloor |V|/3 \rfloor$ (note that $|V_1|$ is even since $p(V_1) = 0$ and $|V_1| \leqslant |V' \cup V'''| = \lfloor |V|/3 \rfloor + \lceil |V|/3 \rceil$). Hence, $(V_1, V \backslash V_1)$ is a good partition (Fig. 7.4d).

If $|p(V')| < |p(V''')|$, the argument is similar. We can build a new set V_1 by adding nodes one by one from V''' to V' until $p(V_1) = 0$; then $|V_1| \leqslant \lfloor 2|V|/3 \rfloor$. Hence, $(V_1, V \backslash V_1)$ is a good partition (Fig. 7.4e). ■

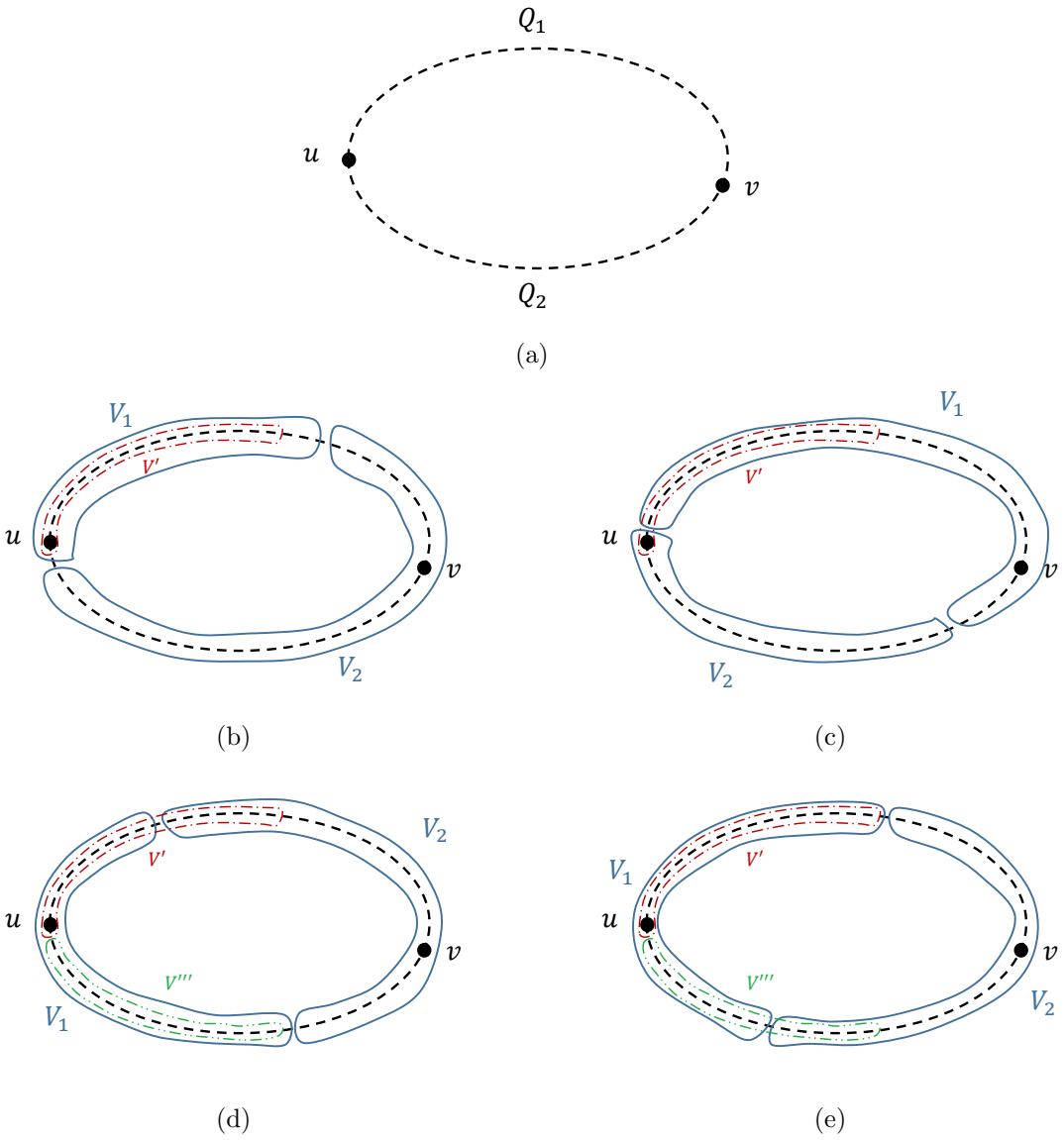


Figure 7.4: Proof of Lemma 7.4.

Corollary 7.2. *If G is a 2-connected series-parallel graph and $\forall i : p(i) = \pm 1$, then the DBCP problem has a solution for $c_p = 1,\ c_s = 2$, and the solution can be found in $O(|E|)$ time.*

Proof. Every series-parallel graph G has a separation pair $\{u, v\}$ such that every connected component of $G[V \backslash \{u, v\}]$ has less than $\lfloor 2|V|/3 \rfloor$ nodes, and furthermore, such a separation pair can be found in linear time. To see this, consider the derivation tree T of the construction

of G. The root of T corresponds to G, the leaves correspond to the edges, and every internal node i corresponds to a subgraph $G_i = (V_i, E_i)$ that is the series or parallel composition of the subgraphs corresponding to its children. Starting at the root of T, walk down the tree following always the edge to the child corresponding to a subgraph with the maximum number of nodes until the number of nodes becomes $\leqslant \lceil 2|V|/3 \rceil$. Thus, we arrive at a node i of the tree such that $|V_i| > \lceil 2|V|/3 \rceil$ and $|V_j| \leqslant \lceil 2|V|/3 \rceil$ for all children j of i. Let u_i, v_i be the terminals of G_i. Note that u_i, v_i separate all the nodes of G_i from all the nodes that are not in G_i. Since $|V_i| > \lceil 2|V|/3 \rceil$, we have $|V \backslash V_i| < |V|/3$. If G_i is the parallel composition of the graphs corresponding to the children of i, then the separation pair $\{u_i, v_i\}$ has the desired property, i.e. all the components of $G[V \backslash \{u, v\}]$ have less than $\lfloor 2|V|/3 \rfloor$ nodes.

Suppose G_i is the series composition of the graphs G_j, G_k corresponding to the children j, k of i, and let w be the common terminal of G_j, G_k; thus, G_i has terminals u_i, w, and G_k has terminals w, v_i. Assume wlog that $|V_j| \geqslant |V_k|$. Then $\lceil |V|/3 \rceil < |V_j| \leqslant \lceil 2|V|/3 \rceil$. The pair $\{u_i, w\}$ of terminals of G_j separates all the nodes of $V_j \backslash \{u_i, w\}$ from all the nodes of $V \backslash V_j$, and both these sets have less than $\lfloor 2|V|/3 \rfloor$ nodes. Thus, $\{u_i, w\}$ has the required property. ■

The graph in Figure 7.1 with $s = 1$ shows that these parameters are the best possible for series parallel graphs: if $c_p = O(1)$ then c_s must be at least 2.

To generalize Lemma 7.4 to all 2-connected graphs, we need to define the *contractible* subgraph and the *contraction* of a given graph as below.

Definition 7.4. *We say an induced subgraph H of a 2-connected graph G is* contractible*, if there is a separating pair $\{u, v\} \subset V$ such that $H = (V_H, E_H)$ is a connected component of $G[V \backslash \{u, v\}]$. Moreover, if we replace H by a weighted edge e' with weight $\mathrm{w}(e') = |V_H|$ between the nodes u and v in G to obtain a smaller graph G', we say G is* contracted *to G'.*

Remark 7.3. *Notice that every contractible subgraph of a 2-connected graph G can also be represented by a pseudo-path between its associated separating pair. We use this property in the proof of Theorem 7.6.*

Using the notion of the graph contraction, in the following lemma, we show that to partition a 2-connected graph, we can reduce it to one of two cases: either G can be considered as a graph

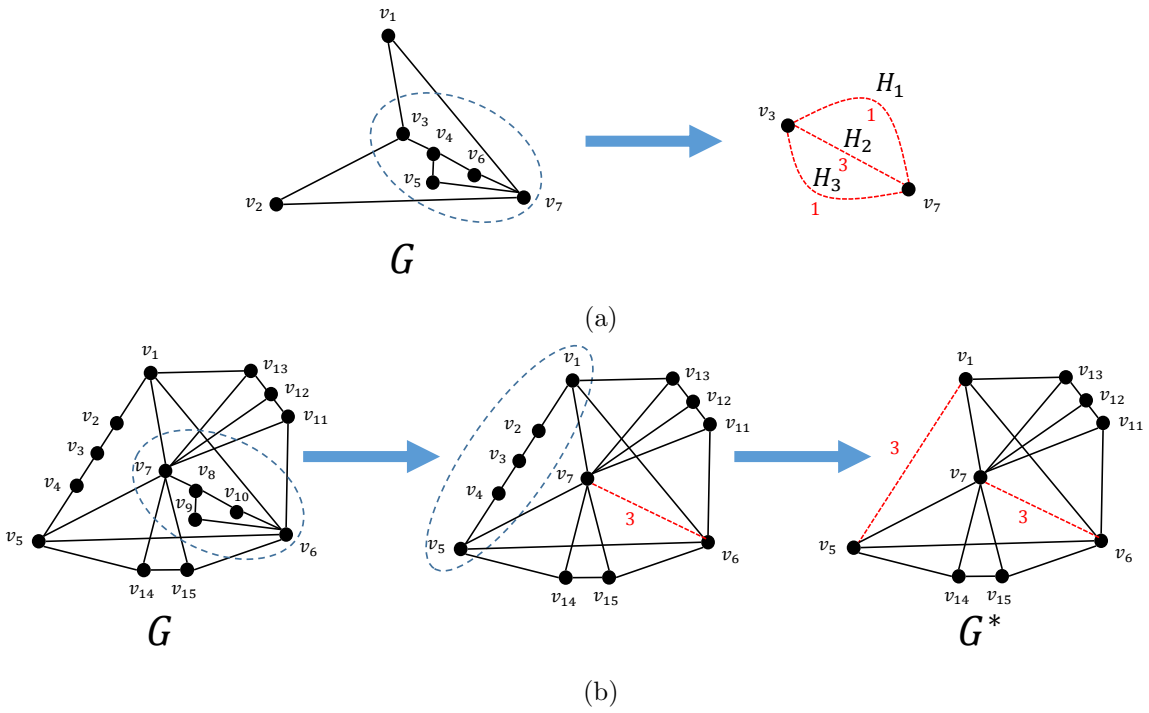


Figure 7.5: Lemma 7.5.

with a set of short pseudo-paths between two nodes, or it can be contracted into a 3-connected graph as illustrated in Fig. 7.5.

Lemma 7.5. *In every 2-connected graph $G = (V, E)$, one of the following cases holds, and we can determine which in $O(|E|)$ time:*

1. *There is a separation pair $\{u, v\} \subset V$ such that if $H_1, \ldots, H_k$ are the connected components of $G[V \backslash \{u, v\}]$, for all i, $|V_{H_i}| < \lfloor 2|V|/3 \rfloor$.*

2. *After a set of contractions, G can be transformed into a 3-connected graph $G^* = (V^*, E^*)$ with weighted edges representing contracted subgraphs such that for every $e^* \in E^*$, $\mathrm{w}(e^*) \leqslant \lceil |V|/3 \rceil - 2$.*

Proof. If there is no separation pairs in G, then G is 3-connected and there is nothing left to prove. So assume $\{u, v\} \subset V$ is a separation pair and $H_1, \ldots, H_k$ are the connected components of $G[V \backslash \{u, v\}]$. If $\forall i, |V_{H_i}| < \lfloor 2|V|/3 \rfloor$, we are done. So let's assume there is a connected

component H_j such that $|V_{H_j}| \geqslant \lfloor 2|V|/3 \rfloor$. Then all the other components H_i, $i \neq j$, can be contracted and represented by an edge of weight at most $\lceil |V|/3 \rceil - 2$ between u and v. Now, we repeat the process by considering the weight of the edges in the size of each connected component (a weighted edge can be contracted again as part of a new connected component and its weight will be added to the total number of nodes in that connected component). An example for each case is shown in Fig. 7.5 for $q = 3$. We can find either a suitable separation pair as in case 1 or a suitable contracted graph G^* as in case 2 in linear time using the Hopcroft-Tarjan algorithm for finding the 3-connected components [84]. ■

Definition 7.5. *In a graph G^* with weighted edges representing contracted subgraphs, define the weight for a subset of nodes $U^* \subset V^*$ as* $\mathrm{w}(U^*) = \mathrm{w}(G^*[U^*]) := |U^*| + \sum_{e \in G^*[U^*]} \mathrm{w}(e)$.

The following lemma strengthens and extends Lemma 7.2 to weighted graphs.

Lemma 7.6. *If $G^* = (V^*, E^*)$ is a 3-connected graph obtained after a set of contractions on G such that for every $e^* \in E^*$, $\mathrm{w}(e^*) \leqslant \lceil |V|/3 \rceil - 2$, then there exists a set $\{u, v, w\} \in V^*$ and a partition of V^* into (V_1^*, V_2^*) such that:*

1. *$V_1^* \cap V_2^* = \varnothing$ and $V_1^* \cup V_2^* = V^*$,*
2. *$G^*[V_1^*]$ and $G^*[V_2^*]$ are connected,*
3. *$\{u, w\}, \{v, w\} \in E^*$,*
4. *$w \in V_1^*$, $u, v \in V_2^*$, and u, v are not cutpoints of $G^*[V_2^*]$,*
5. *$\mathrm{w}(V_2^*) \leqslant \lfloor |V|/3 \rfloor + 1$.*

Moreover, such a partition and $\{u, v, w\}$ can be found in $O(|E|)$ time.

Proof. If G^* has a triangle $\{u, v, w\}$ then since G^* is 3-connected, $V_2^* = \{u, v\}$ and $V_1^* = V^* \backslash V_2^*$ is a good partition. So assume G^* is a triangle-free graph and therefore $|V^*| > 3$. Using the algorithm presented in [46], we can find a non-separating ear decomposition $G^* = P_0^* \cup P_1^* \cdots \cup P_k^*$. A property of their construction that we will use is that if $|P_i^*| > 3$, then each internal node of P_i^* has degree two in G_i^*.

Consider a (triangle-free) counterexample graph G^* and its non-separating ear decomposition. We will show a series of properties for the graph and its decomposition, leading eventually to a contradiction.

Claim 1. For any $0 \leqslant i < k$, if G_i^* has two adjacent degree-2 nodes, then $\mathrm{w}(G_i^*) \leqslant \lfloor 2|V|/3 \rfloor$. In particular, $\mathrm{w}(P_0^*) \leqslant \lfloor 2|V|/3 \rfloor$, and if $|P_i^*| > 3$ then $\mathrm{w}(G_i^*) \leqslant \lfloor 2|V|/3 \rfloor$.

Proof. Suppose that $\mathrm{w}(G_i^*) \geqslant \lfloor 2|V|/3 \rfloor + 1$ and that G_i^* has two adjacent degree-2 nodes w, v. Since G^* is 3-connected, w and v are adjacent respectively to some nodes u, z of $\overline{G_i^*}$. Note that $u \neq z$ because G^* does not have a triangle. If u is not a cutpoint of $\overline{G_i^*}$, set $V_1^* = G_i^* \backslash \{v\}$ and $V_2^* = V^* \backslash V_1^*$. From the properties of the non-separating ear decomposition, it is easy to see that (V_1^*, V_2^*) is a good partition: $G[V_1^*]$ is connected, since G_i^* is biconnected, $G[V_2^*]$ is connected since $\overline{G_i^*}$ is connected, u, v are not cutpoints of $G[V_2^*]$, and $|V_2^*| \leqslant |V| - \lfloor 2|V|/3 \rfloor = \lceil |V|/3 \rceil$. Otherwise, if u is a cutpoint of $\overline{G_i^*}$, let H^* be the connected component of $\overline{G_i^*} \backslash \{u\}$ that contains node z. Set $V_2^* = H^* \cup \{u, v\}$ and $V_1^* = V^* \backslash V_1^*$. It is easy to see again that this is a good partition. Notice that since $G^* \backslash \{u, v\}$ is connected, it follows that $G^*[V_1^*]$ is connected.

The nodes of P_0^* have degree 2 in G_0^*, hence the claim holds for P_0^*, and more generally for any i such that $|P_i^*| > 3$, since the internal nodes of P_i^* have degree 2 in G_i^*. □

We consider now the first ear P_1^* and the corresponding graph G_1^*.

Claim 2. G_1^* consists of three parallel disjoint paths of length at least 3 between two nodes, and $\mathrm{w}(G_1^*) \leqslant \lfloor 2|V|/3 \rfloor$.

Proof. Suppose first that $|P_1^*| = 3$, and let $P_1^* = u, w, v$. At least one of the two $u - v$ paths of the cycle P_0^* has weight at most $\mathrm{w}(P_0^*)/2 + 1$. Let V_2^* be this path and $V_1^* = V^* \backslash V_2^*$. Since $\mathrm{w}(P_0^*) \leqslant \lfloor 2|V|/3 \rfloor$, it follows that $\mathrm{w}(V_2^*) \leqslant \lfloor |V|/3 \rfloor + 1$. Clearly $G[V_1^*]$ and $G[V_2^*]$ are connected, and all the desired properties are satisfied, so this is a good partition.

Therefore, $|P_1^*| > 3$. All the internal nodes of P_1^* have degree 2. By Claim 1, $\mathrm{w}(G_1^*) \leqslant \lfloor 2|V|/3 \rfloor$, and G_1^* consists of three disjoint paths between two nodes x, y, the endpoints of the ear P_1^*. If one of the paths is an edge (x, y), then let Q be the lighter of the other two paths. Set $w = x$, $u = y$, let v be the node of Q adjacent to x, $V_2^* = Q \backslash \{x\}$, and $V_1^* = V^* \backslash V_2^*$; it is

easy to see that this is a good partition. If one of the paths has length 2, then let Q again be the lighter of the other two paths. Setting $u = x$, $v = y$, w the internal node of Q, $V_2^* = Q$, $V_1^* = V^* \backslash V_2^*$ gives a good partition. Note that $\mathrm{w}(Q) \leqslant (\mathrm{w}(G_1^*) + 1)/2 \leqslant (\lfloor 2|V|/3 \rfloor + 1)/2$, hence $\mathrm{w}(Q) \leqslant \lceil |V|/3 \rceil$.

We conclude that G_1^* satisfies the claim. □

Thus G_1^* consists of three paths of length at least 3 between two nodes x, y, and $\mathrm{w}(G_1^*) \leqslant \lfloor 2|V|/3 \rfloor$. We consider now the next ear P_2^*.

Claim 3. $|P_2^*| > 3$ and $\mathrm{w}(G_2^*) \leqslant \lfloor 2|V|/3 \rfloor$.

Proof. Suppose that $|P_2^*| = 3$ and let $P_2^* = u, w, v$. Let R_1, R_2 be two disjoint paths in G_1^* connecting u and v, and assume without loss of generality that $\mathrm{w}(R_2) \leqslant \mathrm{w}(R_1)$. Set $V_2^* = R_2$ and $V_1^* = V^* \backslash V_2^*$. Note that every degree-2 node of $G_1^* - R_2$ has an edge to $\overline{G_1^*}$, and if x or y is not in R_2 then it is adjacent to at least one degree-2 node of $G_1^* - R_2$. Hence $G[V_2^*]$ is connected. Also, $\mathrm{w}(R_2) \leqslant \mathrm{w}(G_1^*)/2 \leqslant |V|/3$.

We conclude that $|P_2^*| > 3$, and hence by Claim 1, $\mathrm{w}(G_2^*) \leqslant \lfloor 2|V|/3 \rfloor$. □

We can finish now the proof of the lemma. G_1^* consists of three paths of length at least 3 between two nodes x, y. Since $|P_2^*| > 3$, all internal nodes of P_2^* have degree 2 in G_2^*. The endpoints of the path P_2^* are either internal nodes of different paths of G_1^* (in which case G_2^* is homeomorphic to K_4, the complete graph on 4 nodes), or they both lie on one of the three paths of G_1^* (either or both endpoints may coincide with the degree-3 nodes x, y of G_1^*). The graph G_2^* is a planar graph in either case. Consider a planar embedding of G_2^*. It has four faces. The sum of the weights of the four faces is $2\mathrm{w}(G_2^*) + 4$ (every edge is counted twice and every node is counted as many times as its degree). Therefore at least one of the faces has weight at most $\mathrm{w}(G_2^*)/2 + 1 \leqslant (|V|/3) + 1$. Let C be the bounding cycle of such a face. If the cycle C has a chord (the chord of course must be embedded outside the face), then let (w, u) be a chord such that one of the paths of C connecting w to u is chordless, let R be this path,and let v be its node adjacent to w. Set $V_2^* = R \backslash \{w\}$ and $V_1^* = V^* \backslash V_2^*$. Then $\mathrm{w}(V_2^*) \leqslant \lfloor |V|/3 \rfloor$, and it is easy to check also that $G[V_1^*]$ is connected. If C is chordless, then let u, w, v be three consecutive

nodes of C. Set $V_2^* = C\backslash\{w\}$ and $V_1^* = V^*\backslash V_2^*$. Again, it is easy to check that the partition satisfies the required properties of the lemma.

This concludes the proof of the lemma. ■

Using Lemma 7.5, then Lemma 7.4, and the idea of the proof for Theorem 7.5, we can prove that when G is 2-connected and all $p(i) = \pm 1$, the DBCP problem has a solution for $c_p = 1$ and $c_s = 2$. We find a suitable convex embedding of the 3-connected graph G^* using Lemma 7.6 and Lemma 7.3, and then embed the nodes of the contracted pseudo-paths appropriately along the segments corresponding to the weighted edges. Some care is needed to carry out the argument as in the proof for Theorem 7.5, since as the line tangent to the circle rotates, the order of the projections of many nodes may change at once, namely the nodes on an edge perpendicular to the rotating line.

Theorem 7.6. *If G is 2-connected, $\forall i, p(i) = \pm 1$, then the DBCP problem has a solution for $c_p = 1$ and $c_s = 2$. Moreover, this solution can be found in polynomial time.*

Proof. Using Lemma 7.5, we consider two cases:

(i) There is a separation pair $\{u, v\} \in V$ such that if $H_1, \ldots, H_k$ are the connected components of $G\backslash\{u, v\}$, for all i, $|V_{H_i}| < \lfloor 2|V|/3 \rfloor$. In this case Lemma 7.4 proves the theorem.

(ii) After a set of contractions, G can be transformed into a 3-connected graph $G^* = (V^*, E^*)$ with weighted edges such that for any edge $e^* \in E^*$, $\mathrm{w}(e^*) \leqslant \lceil |V|/3 \rceil - 2$. In this case the proof is similar to the proof of Theorem 7.5. Notice that if G^* contains a triangle then the proof is much simpler as in the proof of Lemma 7.1 but here to avoid repetition, we use the approach in the proof of Theorem 7.5 and prove the theorem once for all cases of G^*.

Using Lemma 7.6, we can find $\{u, v, w\} \in V^*$ and a partition (V_1^*, V_2^*) of V^* with properties described in the lemma. Set $X = \{u, v, w\}$. Using Lemma 7.3, G^* has a convex X-embedding in general position, $f^* : V^* \to \mathbb{R}^2$, as described in the lemma and depicted in Fig. 7.3. Now, from this embedding, we get a convex X-embedding $f : V \to \mathbb{R}^2$ for G as follows. For any $i \in V \cap V^*$, $f(i) = f^*(i)$. For any edge $\{i, j\} \in E^*$ such that $\{i, j\}$ represents an induced subgraph of G, we represent it by a pseudo-path P of G between

i and j and place the nodes of P in order on random places on the line segment that connects $f(i)$ to $f(j)$. If the edge $\{i,j\} \in E^*$ is between a node in V_1^* and a node in V_2^* and represents a pseudo-path P in G , we place the nodes in P in order on random places on the segment that connects $f(i)$ to $f(j)$ but above the line $\mathcal{L}_1$. Hence, by this process, we get a convex X-embedding for G which is in general position (almost surely) except for the nodes that are part of a pseudo-path. From Lemma 7.3, the embedding has the following property. Consider any line on the plane, and the subset of nodes whose points lie on the same side of the line. If the subset has size at least $\lfloor |V|/3 \rfloor + 1$ then it induces a connected subgraph of G.

The rest of the proof is similar to the proof of Theorem 7.5. We consider again a circle $\mathcal{C}$ around $f(u), f(v), f(w)$ in $\mathbb{R}^2$ as shown in Fig. 7.3. Also consider a directed line $\mathcal{L}$ tangent to the circle C at point A and project the nodes of G onto the line $\mathcal{L}$ (we consider a line such that the projections are distinct). We label nodes based on their projection order on the line $\mathcal{L}$ from left to right from $1^{(\mathcal{L})}$ to $|V|^{(\mathcal{L})}$. For each $t = 1, \ldots, |V|$, let $V^{(\mathcal{L})}(t) = \{1^{(\mathcal{L})}, \ldots, t^{(\mathcal{L})}\}$ denote the set of the first t nodes in this ordering. Since the embedding f has the properties described in Lemma 7.6 and 7.3, for all $\lfloor |V|/3 \rfloor + 1 \leqslant t \leqslant \lceil 2|V|/3 \rceil - 1$, if we set $V_1 = V^{(\mathcal{L})}(t)$ and $V_2 = V \backslash V_1$, then $G[V_1]$ and $G[V_2]$ are both connected. Define $V_1' = V^{(\mathcal{L})}(\lfloor |V|/3 \rfloor)$ and $V_2' = V \backslash V^{(\mathcal{L})}(\lceil 2|V|/3 \rceil)$, i.e., V_1' contains the first $\lfloor |V|/3 \rfloor$ nodes and V_2' the last $\lfloor |V|/3 \rfloor$ nodes in the ordering. If $p(V_1')p(V_2') \geqslant 0$, then there must exist an index j with $\lfloor |V|/3 \rfloor \leqslant j \leqslant \lceil 2|V|/3 \rceil$ such that $p(V^{(\mathcal{L})}(j)) = \sum_{i=1}^{j} p(i^{(\mathcal{L})}) = 0$. Consequently, there is an index t such that $\lfloor |V|/3 \rfloor + 1 \leqslant t \leqslant \lceil 2|V|/3 \rceil - 1$ and $|p(V^{(\mathcal{L})}(t))| \leqslant 1$: if $j = \lfloor |V|/3 \rfloor$ then let $t = j + 1$, if $j = \lceil 2|V|/3 \rceil$ then $t = j - 1$, and otherwise let $t = j$. Hence, $V_1 = V^{(\mathcal{L})}(t) = \{1^{(\mathcal{L})}, \ldots, t^{(\mathcal{L})}\}$ and $V_2 = V \backslash V_1$ is a good partition. Therefore, if $p(V_1')p(V_2') \geqslant 0$ then we can obtain a good partition. We will show that there is a line such that $p(V_1')p(V_2') \geqslant 0$.

Assume without loss of generality that in the initial position of the line, $p(V_1') > 0$ and $p(V_2') < 0$. As we move $\mathcal{L}$ from being tangent at point A to point B where AB is a diameter of the circle $\mathcal{C}$, and consider the resulting partition at point B, since V_1' and V_2' completely

switch places compared to the partition at point A, at point B we have $p(V_1') < 0$ and $p(V_2') > 0$. Hence, as we move $\mathcal{L}$ from being tangent at point A to point B and keep it tangent to the circle, in the resulting partitions, $p(V_1')$ goes at some point from some positive value to a nonpositive value. Notice that the ordering of the projections on the line changes only if $\mathcal{L}$ passes a point D on the circle such that at D, $\mathcal{L}$ is perpendicular to a line that connects $f(i)$ to $f(j)$ for $i, j \in V^*$; then the order of i and j is switched, and if (i, j) is an edge of G^* then also the ordering of the nodes in the corresponding pseudopath is reversed. Note that all of these affected nodes are consecutive in the ordering. If $p(V_1')$ changes then V_1' must contain at least one of the affected nodes, and similarly for $p(V_2')$.

So assume $p(V_1') > 0$ in a projection on a line $\mathcal{L}_b$ slightly before a perpendicular point and $p(V_1') \leqslant 0$ in a resulting projection on a line $\mathcal{L}_a$ slightly after a perpendicular point. Define $V_{1b}' := \{1^{(\mathcal{L}_b)}, \ldots, \lfloor |V|/3 \rfloor^{(\mathcal{L}_b)}\}$, $V_{2b}' := \{(|V| - \lfloor |V|/3 \rfloor + 1)^{(\mathcal{L}_b)}, \ldots, |V|^{(\mathcal{L}_b)}\}$, $V_{1a}' := \{1^{(\mathcal{L}_a)}, \ldots, \lfloor |V|/3 \rfloor^{(\mathcal{L}_a)}\}$, and $V_{2a}' := \{(|V| - \lfloor |V|/3 \rfloor + 1)^{(\mathcal{L}_a)}, \ldots, |V|^{(\mathcal{L}_a)}\}$. So $p(V_{1b}') > 0$ and $p(V_{1a}') \leqslant 0$.

If $p(V_{1b}')p(V_{2b}') \geqslant 0$, as we argued previously, we can find a good partition (V_1, V_2) such that $V_{1b}' \subseteq V_1$ and $V_{2b}' \subseteq V_2$, and we are done. So assume $p(V_{1b}') > 0$ and $p(V_{2b}') < 0$. Since for any $e^* \in E^*$, $\mathrm{w}(e^*) \leqslant \lceil |V|/3 \rceil - 2$, the ordering of the nodes based on the projections on lines $\mathcal{L}_b$ and $\mathcal{L}_a$ is different for at most $\lceil |V|/3 \rceil$ consecutive nodes (the ordering of at most $\lceil |V|/3 \rceil$ consecutive nodes reverses as we move from $\mathcal{L}_b$ to $\mathcal{L}_a$). Since $V_{1b}' \neq V_{1a}'$ (recall $p(V_{1b}') > 0$ and $p(V_{1a}') \leqslant 0$), the set of reversed nodes includes one of the first $\lfloor |V|/3 \rfloor$ nodes in the ordering, hence it cannot extend to include also one of the last $\lfloor |V|/3 \rfloor$ nodes; therefore $p(V_{2a}') = p(V_{2b}') < 0$. Thus, $p(V_{1a}')p(V_{2a}') \geqslant 0$ and hence, as we argued before, there is a good partition (V_1, V_2) such that $V_{1a}' \subseteq V_1$ and $V_{2a}' \subseteq V_2$.

Regarding the computation of a good partition, after we compute the contracted graph G^* and its convex embedding, the rest of the computation can be easily carried out in $O(|V|^2 \log |V|)$ time.

■

Similar to Corollary 7.1, the approach used in the proof of Theorem 7.6, can also be used for the case when the weights are arbitrary (not necessarily ± 1) and $p(V) \neq 0$. It is easy to verify that in this case, if G is 2-connected, the DBCP problem has a connected partition (V_1, V_2) such that $|p(V_1) - p(V)/2|, |p(V_2) - p(V)/2| \leqslant \max_{j \in V} |p(j)|$ and $|V_1|, |V_2| \geqslant \lceil |V|/3 \rceil$.

Corollary 7.3. *If G is 2-connected, then the DBCP problem (with general p and not necessarily satisfying $p(V) = 0$) has a solution for $c_p = \max_{j \in V} |p(j)|$ and $c_s = 2$. Moreover, this solution can be found in polynomial time.*

7.5 Graphs with Two Types of Nodes

Assume G is a connected graph with nodes colored either red ($R \subseteq V$) or blue ($B \subseteq V$). Let $|V| = n$, $|R| = n_r$, and $|B| = n_b$. If G is 3-connected, set $p(i) = 1$ if $i \in R$ and $p(i) = -1$ if $i \in B$. Corollary 7.1 implies then that there is always a connected partition (V_1, V_2) of V that splits both the blue and the red nodes evenly (assuming n_r and n_b are both even), i.e., such that $|V_1| = |V_2|$, $|R \cap V_1| = |R \cap V_2|$, and $|B \cap V_1| = |B \cap V_2|$. (If n_r and/or n_b are not even, then one side will contain one more red or blue node.)

Corollary 7.4. *Given a 3-connected graph G with nodes colored either red ($R \subseteq V$) or blue ($B \subseteq V$). There is always a partition (V_1, V_2) of V such that $G[V_1]$ and $G[V_2]$ are connected, $|V_1| = |V_2|$, $|R \cap V_1| = |R \cap V_2|$, and $|B \cap V_1| = |B \cap V_2|$ (assuming $|R|$ and $|B|$ are both even). Such a partition can be computed in polynomial time.*

Proof. Suppose without loss of generality that $n_r \geqslant n_b$ and let $n_r - n_b = 2t$ and $n_r + n_b = n = 2m$. Set $p(i) = 1$ for $i \in R$ and $p(i) = -1$ for $i \in B$. Then $p(V) = 2t$. From the equations, we have $n_r = m + t$ and $n_b = m - t$.

From Corollary 7.1 we can find a partition (V_1, V_2) such that $|V_1| = |V_2|$ and $|p(V_1) - p(V)/2|, |p(V_1) - p(V)/2| \leqslant 1$. Let $r_1 = |R \cap V_1|$ and $b_1 = |B \cap V_1|$. We have $r_1 + b_1 = n/2 = m$ and $t - 1 \leqslant r_1 - b_1 \leqslant t + 1$. Therefore, $(m + t)/2 - (1/2) \leqslant r_1 \leqslant (m + t)/2 + (1/2)$. Since r_1 is an integer and $n_r = m + t$ is even, it follows that $r_1 = (m + t)/2 = n_r/2$. Hence, $b_1 = (m - t)/2 = n_b/2$. Therefore, V_2 also contains $n_r/2$ red nodes and $n_b/2$ blue nodes. ■

If G is only 2-connected, we may not always get a perfect partition. Assume wlog that $n_r \leqslant n_b$. If for every $v \in R$ and $u \in B$, we set $p(v) = 1$ and $p(u) = -n_r/n_b$, Corollary 7.3 implies that there is always a connected partition (V_1, V_2) of V such that both $|(|R \cap V_1| - n_r/n_b|B \cap V_1|)| \leqslant 1$ and $|(|R \cap V_2| - n_r/n_b|B \cap V_2|)| \leqslant 1$, and also $\max\{\frac{|V_1|}{|V_2|}, \frac{|V_2|}{|V_1|}\} \leqslant 2$. Thus, the ratio of red to blue nodes in each side V_i differs from the ratio n_r/n_b in the whole graph by $O(1/n)$. Hence if the numbers of red and blue nodes are $\omega(1)$, then the two types are presented in both sides of the partition in approximately the same proportion as in the whole graph.

Corollary 7.5. *Given a 2-connected graph G with nodes colored either red ($R \subseteq V$) or blue ($B \subseteq V$), and assume wlog $|R| \leqslant |B|$. We can always find in polynomial time a partition (V_1, V_2) of V such that $G[V_1]$ and $G[V_2]$ are connected, $|V_1|, |V_2| \geqslant \lceil |V|/3 \rceil$, and the ratio of red to blue nodes in each side V_i differs from the ratio $|R|/|B|$ in the whole graph by $O(1/n)$.*

7.6 Conclusion

In this chapter, we studied power grid Islanding problem. We modeled it as the problem of partitioning a graph into two connected subgraphs that satisfy simultaneouly two objectives: (1) they balance the supply and demand within each side of the partition (or more generally, for the case of $p(V) \neq 0$, they split approximately equally the excess supply/demand between the two sides), and (2) the two sides are large and have roughly comparable size (they are both $\Omega(|V|)$). We showed that for 2-connected graphs it is always possible to achieve both objectives at the same time, and for 3-connected graphs there is a partition that is essentially perfectly balanced in both objectives. Furthermore, these partitions can be computed in polynomial time. This is a paradigmatic bi-objective balancing problem. We observed how it can be easily used to find a connected partition of a graph with two types of nodes that is balanced with respect to the sizes of both types. Overall, we believe that the novel techniques used in this chapter can be applied to partitioning heterogeneous networks in various contexts.

There are several interesting further directions that suggest themselves. First, extend the theory and algorithms to find doubly balanced connected partitions to more than two parts. Second, even considering only the supply/demand objective, does the analogue of the results of Lováz and Gyori [80, 99] for the connected k-way partitioning of k-connected graphs with respect to size (which corresponds to $p(i) = 1$) extend to the supply/demand case ($p(i) = \pm 1$) for $k > 3$? And is there a polynomial algorithm that constructs such a partition? Finally, extend the results of Section 7.5 to graphs with more than two types of nodes, that is, can we partition (under suitable conditions) a graph with several types of nodes to two (or more) large connected subgraphs that preserve approximately the diversity (the proportions of the types) of the whole population?

Chapter 8

Generating Synthetic Power Grids

Addressing most of the challenges facing the power grids requires real grid topologies with real geographical coordinates. For example, to study the vulnerabilities of the grid to natural disasters one needs to match the grid map with the paths of hurricanes or water flood maps. Similarly, incorporating renewable energy resources in the grid requires the approximate locations of the grid lines and buses to be matched with the wind maps. However, in order to avoid exposing vulnerabilities, topologies of the power transmission networks and particularly the locations of the substations and the lines are usually not publicly available or are hard to obtain.

There are only very limited test cases and real-world power grid data sets that are publicly and freely available. These include the IEEE test cases [1], the National Grid UK [2], the Polish grid [4], and an approximate model of the European interconnected system [162]. To the best of our knowledge, among these, National Grid UK is the only publicly available dataset with geographical locations. Even if the data was available, it would be unwise to publish vulnerability results which are based on real topologies, due to the enormous cost of grid enhancements. On the other hand, it was recently shown that simple random graph models cannot be used to generate grids with appropriate structural and spatial characteristics [52]. Therefore, there is a growing interest in generating synthetic power grids [11, 33, 34, 75].

Motivated by this need, we thoroughly study the structural properties of real power grids and introduce the Network Imitating Method Based on LEarning (NIMBLE) for generating

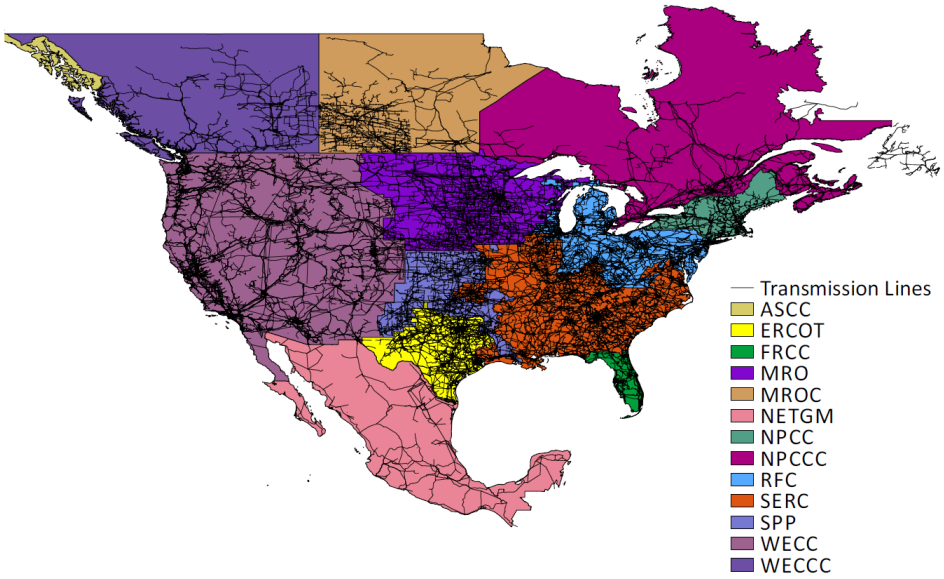


Figure 8.1: The North American Electric Reliability Corporation (NERC) regional entities and the National Electricity Transmission Grid of Mexico (NETGM). Different reliability corporations/councils are marked with different colors.

synthetic networks with similar structural and spatial properties. We focus on the transmission networks of the North American and Mexican power grids (see Fig. 8.1) using data that we obtained from the Platts Geographic Information System (GIS) [3] (Similar techniques could be applied to industry-grade data sets, if they become publicly available). In particular, we consider the Western Interconnection (WI), one of the two major interconnections in North America, which includes the Western Electricity Coordinating Council in the United States (WECC) and Canada (WECCC).

We evaluate the networks' structural properties under five metrics: *average path length (L), clustering coefficient (C), degree distribution of the nodes, number of line intersections ($\mathcal{X}$), and the length distribution of the lines.* For each node i with degree d_i, at most $d_i(d_i-1)/2$ lines can exist between its neighbors. The clustering coefficient is the fraction of these allowable lines that actually exist, averaging over all the nodes. The average path length is defined as the number of lines in the shortest path between two nodes, averaged over all pairs of vertices. The

first three metrics are very common [15, 17, 23, 42, 52, 53, 153]. The importance of the number of line intersections in generating synthetic power grids was first discussed by Birchfield, et al. [34]. However, to the best of our knowledge, the length distribution of the lines has not been thoroughly studied before in power grids. The line lengths are particularly important in power grids, since the physical properties of a line (e.g., admittance and type) are directly correlated with its length [76], and hence, directly impact the grid's structural properties.

To compare the robustness of the WI and the generated network to failures, we simulate cascading failures initiated by double line failures as well as circular area failures (as failures caused by natural disasters), and compute the *yield (the ratio between the demand supplied at the end of a cascade and the original demand), number of failed lines, and number of connected components* at the end of the cascade in these networks. Cascading failures in networks and power grids and their robustness have been widely studied [21, 29, 38, 42, 53, 57, 60, 61, 82, 105, 156, 159]. In this chapter, we use the cascade model due to line overloads in power grids with a *deterministic outage rule* as provided in Chapter 3. We show that the generated networks have similar structural and spatial properties as well the same level of robustness to failures to the WI.

8.1 Related work

The structural properties of various power grids (e.g., in North America, some European countries, and Iran) were studied before [53, 56, 104, 118, 124, 153]. Most of these studies considered one or two properties (e.g., average degree, degree distribution, average path length, and clustering coefficient) and computed it in a given power grid. In some cases a certain class of graphs was suggested as a good representative of a power grid network, based on one or two structural properties [15, 17, 23, 42, 52, 53, 153]. For example, Watts and Strogatz [153] suggested the small-world graph as a good representative, based on the shortest path lengths between nodes and the clustering coefficient of the nodes. Barabási and Albert [23] showed that scale-free graphs are better representatives based on the degree distribution. However, by comparing the WI

with these models, Cotilla-Sanchez, et al. [52] showed that none of them can represent the WI properly.

More detailed models that are specifically tailored to the power grid characteristics were proposed by Wang, et al. [152] and also by Schultz, et al. [122] but they did not consider the *spatial distribution* of the nodes and the length distribution of the lines. The spatial distribution of the nodes is correlated with the length of the lines, and as mentioned above, it is important to consider line lengths when designing a method for synthetic power grid generation.

While there are several models for generating spatial networks [24, 101, 157], most of them were not designed to generate networks with properties similar to power grid networks'. In very recent novel papers [34,75], a synthetic network based on the locations of the cities and the power plants in Texas was generated. Despite using the geographical locations, no comparisons to the real grid in Texas (neither topological nor performance wise) were provided by the authors. The generated networks, however, was shown to have similar topological (e.g., degree distribution) properties to the WI. In a follow up work [33], more engineering properties (e.g., transformer and generator parameters) of power grids were studied in detail.

The novel methods in [34, 75] consider more engineering details than our work but do not provide a general framework to generate multiple synthetic power grids. Our work is the first to consider the spatial distribution of the nodes (buses) in power grids and to provide a general framework for generating multiple synthetic networks with realistic structural properties. Moreover, this is the first time that *the power flows and vulnerability against cascading failures* are considered in evaluating a synthetic power grid. In our future work, we plan to incorporate more engineering details [33] to improve the quality of our generated power grids to the level of industrial grade grids.

8.2 Preliminaries

8.2.1 Data

We obtained the topological network of the WI from the Platts GIS [3] and conducted longitude-latitude to planar (x, y) coordinate transformation, using the great-circle distance method. We

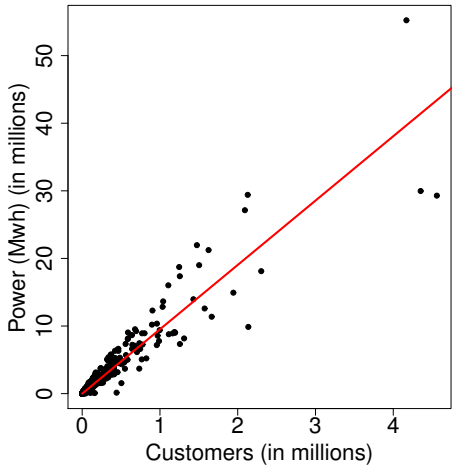
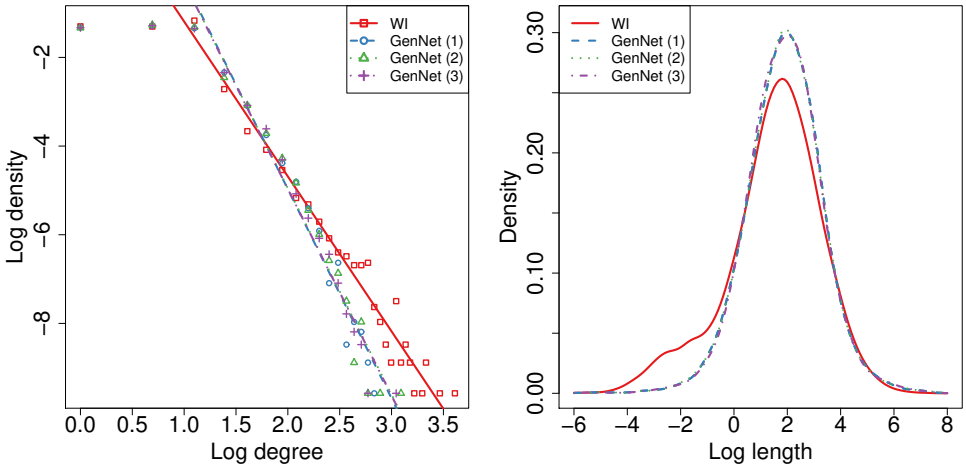


Figure 8.2: The average residential power usage versus the number of customers in the US. We observed that average residential power usage (obtained from EIA website [6]) is directly related to the population (the slope of the regression line is 9.51). We used this relationship to estimate the demand at each city based on its population.

extracted the coordinates of the buses/substations from the endpoint coordinates of the lines. We then used the geographical coordinates of the substations and the lines to construct the graph with nodes and edges that represent substations and lines, respectively. We used the map of reliability coorporations/councils boundaries to divide the graph into regional entities (See Fig. 8.1).

The GIS does not provide the reactance values of the lines. The reactance of a line depends on its geometrical properties and there is a linear relation between the line's length and reactance: the longer the line is, the larger its reactance. Thus, we assumed that all lines have the same physical properties (other than length) and used the length to determine the reactance. Moreover, since the power flow solution is scale invariant of the reactance values, we simply use the length of each line as its reactance. We consider the same for the lines in the generated networks.

To estimate the demands and supplies, we used the cities' populations and the power plants' capacities, as well as their locations. The locations and populations of the cities in the U.S. and Canada are publicly available. We obtained the information about the locations of the power

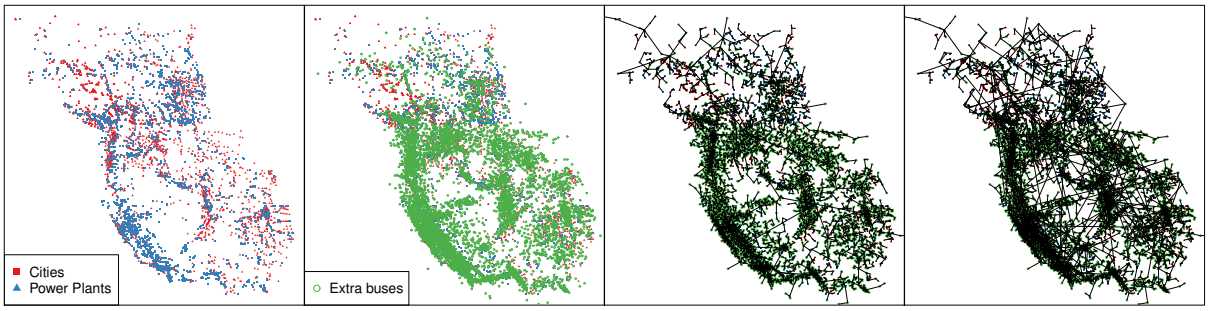


(a) Degree distributions (b) Length distributions

Figure 8.3: The degree distributions of the nodes and length distributions of the lines in the WI and three generated networks. **(a)** The degree distributions of the nodes in log-log scale. In this figure, the slope of the fitted linear regression line to the distributions' tail is $\zeta = -3.49$ for the WI and $\zeta \approx -4.7$ for all the generated networks. The Kolmogrov-Smirnov statistic between the degree distributions in the WI and the generated networks is $D_{KS} \approx 0.05$. **(b)** The log length distributions of the lines (in km). In this figure, $D_{KL} \approx 0.1$ between the length distributions of the lines in the WI and the generated networks.

plants and their capacities from U.S. Energy Information Administration (EIA) website [6]. We observed that the average residential power usage (obtained from EIA website [6]) is directly related to the population (see Fig. 8.2). By modeling this relationship using linear regression, we estimated the power demand at each city based on its population. Once we computed the total demand, we assigned the power generation level to each power plant according to its capacity to supply the demand.

In the WI, we assigned each city (along with its demand) or power plant (along with its supply) to its closest node.



(a) (b) (c) (d)

Figure 8.4: NIMBLE's steps for generating a synthetic grid similar to the WI. **(a)** In the first step, it picks the locations of the cities and power plants as a subset of the nodes (buses) in the generated network. **(b)** In the second step, by estimating the spatial density of the nodes in the WI using GMM, it adds more nodes to the generated network to make the total number of nodes equal to the one in the WI. **(c)** In the third step, it finds an spanning tree of the nodes by connecting each node to its closest node with a higher index to ensure the connectivity of the network. **(d)** In the last step, it adds more lines to the network to increase the robustness of the generated network and adjust its properties (e.g., total number of lines and degree distribution) to resemble those of the WI. It does it by repeatedly selecting a low degree node in a dense area and connecting it to a high degree node which is also nearby.

8.2.2 Degree and line length distributions comparison

To compare the degree distributions of the nodes in the WI and the generated networks, we use the Kolmogrov-Smirnov (D_{KS}) statistic [115]. If $P(x)$ and $Q(x)$ are two Cumulative Distribution Functions (CDFs), the KS statistic between these two is $D_{KS} = \max_x |P(x) - Q(x)|$.

To measure the similarity between the length distributions of the lines in a given network and a generated network, we use the Kullback-Leibler (D_{KL}) divergence. Specifically, the KL-divergence of distribution q from p, denoted by $D_{KL}(p\|q)$, is a measure of the information lost when q is used to approximate p: $D_{KL}(p\|q) = \int_{-\infty}^{\infty} p(x) \ln \frac{p(x)}{q(x)} dx$.

8.2.3 Cascading failures model

We follow the cascade model due to line overloads in power grids with a *deterministic outage rule* as described in Chapter 3: namely, a line $e = \{u, v\}$ fails when the magnitude $|f_e|$ of the flow on that line exceeds its capacity c_e. The line flow capacities are estimated as $c_e = (1 + \alpha) \max\{|f_e|, \overline{f}\}$, where $\overline{f}$ is the median of the initial magnitude of line flows and α is the lines' factor of safety. For comparison purposes, we select the median of the initial magnitude of line flows in the WI as the minimum capacity for the lines in the generated networks as well.

We use *yield (the ratio between the demand supplied at the end of a cascade and the original demand), number of failed lines, and number of connected components* at the end of the cascade to evaluate its severity.

8.2.4 Computation

For analyzing the power grid topological properties, we used the `igraph` library in `R` [10]. This library provides a collection of network analysis tools.

To estimate the KL-divergence between distributions, we used the `FNN` library in `R` which utilizes the method introduced by Boltz, et al. [35] for estimating the KL-divergence between two distributions using their samples.

For fitting a Gaussian Mixture Model (GMM) (see Section 8.3), we used the `mclust` library in `R` [72]. This library uses the Expectation Maximization (EM) algorithm to fit a GMM and provides the Bayesian Information Criterion (BIC) for the selected number of clusters.

8.3 NIMBLE

The structural properties of the WI are shown in Fig. 8.3 and Table 8.1. The main observations are: (i) the degree distributions of power grids are very similar to those of scale-free networks, but grids have less degree 1 and 2 nodes and do not have very high degree nodes (e.g., Fig. 8.3a), (ii) it is inefficient and unsafe for the power grids to include very long lines (e.g., Fig. 8.3b), and

Table 8.1: Comparison between the structural properties of the WI and the generated networks. Three instances of GenNet are shown to illustrate that the metric values are similar in various generated networks. All networks have 14,430 nodes and 18,884 lines.

Networks	L	C	ζ	D_{KS}	$\mathcal{X}$	D_{KL}
WI	17.44	0.048	-3.49	0	7,358	0
GenNet (1)	16.28	0.048	-4.72	0.051	12,108	0.10
GenNet (2)	16.02	0.045	-4.65	0.043	12,132	0.10
GenNet (3)	16.28	0.050	-4.66	0.052	11,145	0.10

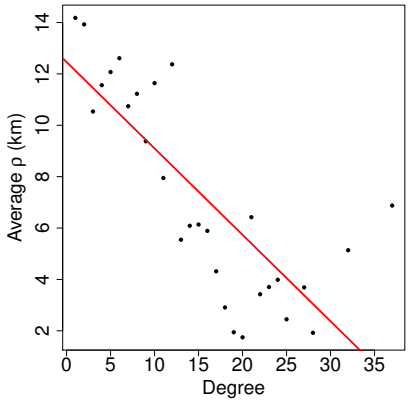
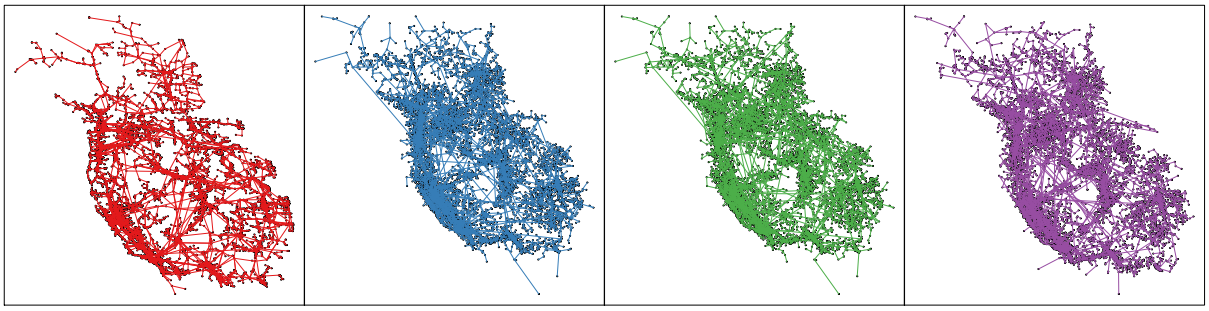


Figure 8.5: The relationship between the degree of a node and its average ρ with $k = 10$, for the nodes in the WI (the red line is the linear regression fit to the data points).

(iii) the average path length and the clustering coefficient demonstrate the *small-world* property of power grids [153].

Based on these characteristics, we introduce the Network Imitating Method Based on LEarning (NIMBLE) for generating synthetic networks similar to the real power grids. The NIMBLE steps are summarized in Fig. 8.4.

In the first step, the NIMBLE picks the locations of the cities and power plants as a subset of the nodes (buses) in the generated network. However, since the cities and the power plants are usually the endpoints of the network, more nodes need to be added to the generated network to make the total number of nodes equal to the number of nodes in a given grid (here the



(a) WI (b) GenNet (1) (c) Gen Net (2) (d) GenNet (3)

Figure 8.6: The Western Interconnection (WI) power grid and the three generated networks with 14,430 buses (nodes) and 18,884 lines.

WI). Notice that if the locations of the cities and power plants are not available and only the topological properties of the given network need to be imitated, step 1 of the method can be skipped (see the arXiv report [134], for a more general algorithm for this case).

The node (buses/substations) positions are correlated with the population and geographical properties. Thus, the nodes can be clustered into groups based on their geographical proximity using mixture models and in particular the Gaussian Mixture Models (GMM). Hence, in the second step, NIMBLE uses a GMM for clustering the positions and uses the Bayesian Information Criterion (BIC) to find the best number of clusters (c). It obtains the mean and covariance matrix (μ_j, Σ_j) of the points in clusters $j = 1, \ldots, c$ along with the categorical probability of the clusters $\pi = (\pi_1, \ldots, \pi_c)$. Then, it uses these parameters to generate more nodes with similar spatial distribution as the nodes in a given network to make the total number of nodes equal in the given and generated networks. For the WI, we select $c = 55$ based on the BIC.

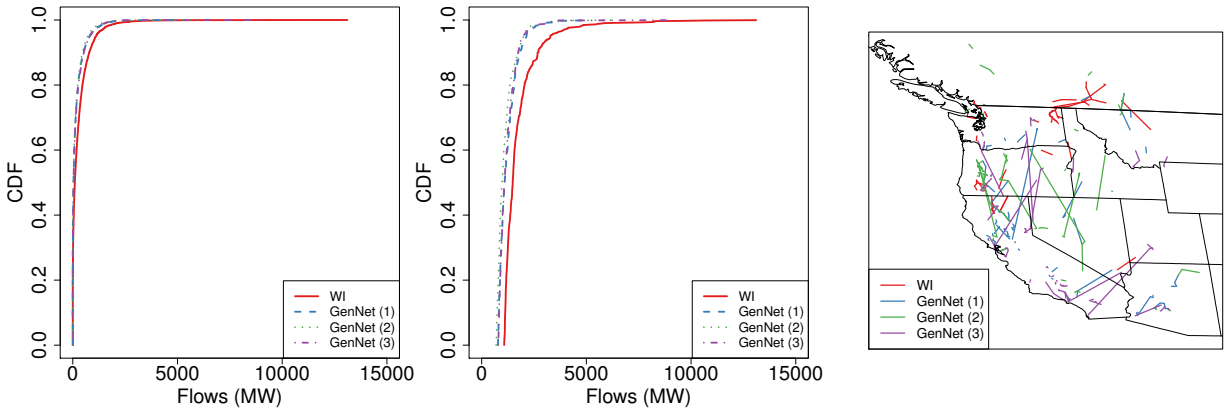
To connect the nodes, the NIMBLE takes two steps inspired by the historical evolution of power grids. The two main design consideration of the grid are (i) connectivity and (ii) robustness.

In order for the power grid to operate, the substations (nodes) should be connected. To satisfy the connectivity of the generated network, in the third step, the NIMBLE connects each node i to its closest (Euclidean distance) node j such that $i < j$ in order to form a spanning

tree of the nodes. This is inspired by the way the power grids are evolved. Each new bus or substation is connected to the closest bus and substation that already exist in the grid. Since the cities and the power plants are usually the end points in the power grid, in the NIMBLE, these nodes have the lowest index. At the end of this step, the network gets connected. Notice that the resulting tree differs from the minimum spanning of the nodes since the obtained tree depends on the order of the nodes (for more details see the arXiv report [134]).

In the last step, the NIMBLE repeatedly adds lines to the generated network to increase its robustness and adjust its properties (e.g., total number of lines, L, and C) to resemble those of a given network. This step is based on three observations: (i) the degree distributions of power grids are very similar to those of scale-free networks, but grids have less degree 1 and 2 nodes and do not have very high degree nodes (e.g., Fig. 8.3a), (ii) it is inefficient and unsafe for the power grids to include very long lines (e.g., Fig. 8.3b), and (iii) nodes in denser areas are more likely to have higher degrees (see Fig. 8.5). To compute the density around node i, we define ρ_i as the average Euclidean distance of node i from its k nearest nodes. We select $k = 10$ in this work.

Hence, the NIMBLE repeats the following steps until the number of lines is equal to the number of lines in the given network: (1) select a low degree node in a dense area (observations (i) and (iii)), and (2) connect it to a high degree node which is also nearby (observations (i) and (ii)). *To select a low degree node in a dense area*, the NIMBLE samples a node i from all the degree 1 and 2 nodes with probability $\propto \rho_i^{-\eta}$, where η is a tunable parameter. *To connect the sampled node to a high degree but nearby node*, the NIMBLE connects node i to node j sampled from all other nodes with probability $\propto \|\vec{z}_i - \vec{z}_j\|_2^{-\beta} d_j^{\gamma}$, where vector $\vec{z}_i$ is the (x, y) coordinate of the node i, and β, γ are tunable parameters. This implies that node i preferentially connects to a high-degree node, unless the high-degree node is too far in which case it is desirable to connect to a low-degree but nearby node (as in the preferential attachment model [23], however distance was not considered there). For the generated networks in this chapter, we empirically select $\eta = 0.5$, $\beta = -2.5$, and $\gamma = 1.5$.



(a) (b) (c)

Figure 8.7: Power flows in the WI and the generated networks. **(a)** The CDF of the flows on the lines. **(b)** The CDF of the top 1,000 flows on the lines. **(c)** The geographical locations of the lines carrying the top 100 flows.

8.4 Topological evaluation of the generated networks

We demonstrate the performance of the NIMBLE in generating synthetic grids similar to the WI power grid. Since our method is probabilistic, we generate three networks for evaluation purposes to show the consistency of the generated networks' properties. We refer to the generated networks as: *GenNet (1), GenNet (2),* and *GenNet (3).* The generated networks visually resemble the WI very well (see Fig. 8.6).

Fig. 8.3 compares the degree distributions of the nodes and length distributions of the lines in the WI and the generated networks. As can be seen in Fig. 8.3a, the tail of the degree distribution in the WI follows a power-law distribution. However, following the work by Clauset, et al. [50] and since these networks are finite, there is not enough statistical evidence to support the power-law hypothesis. Therefore, we only use the slope (ζ) of the fitted linear regression line to the tail distribution for comparison purposes. Table 8.1 provides a summary of the topological properties of the WI and the generated networks. The provided results demonstrate

Table 8.2: Statistics of the flows (MW). The *backup lines* are the lines that do not initially carry any significant flow.

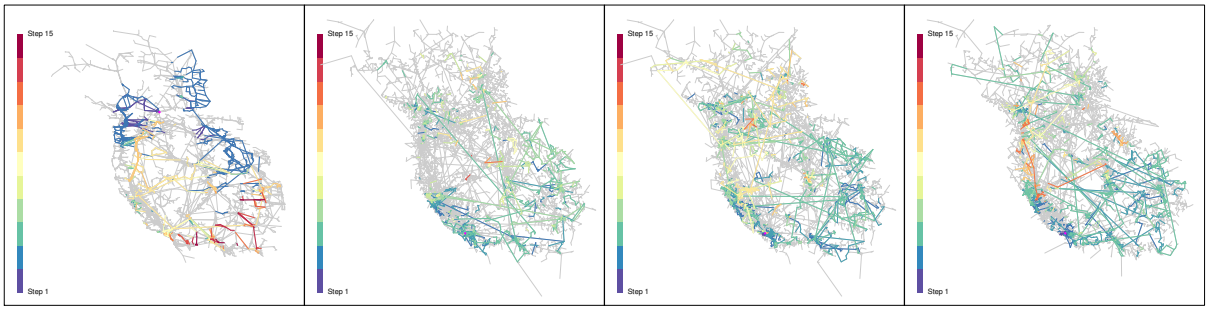
Networks	Mean	Median	SD	Max	Backup lines
WI	282.37	98.54	488.79	13,111.68	3,558
GenNet (1)	168.93	33.32	320.93	4,777.07	3,419
GenNet (2)	153.78	29.53	295.83	6,171.67	3,464
GenNet (3)	172.27	34.71	321.12	8,867.31	3,289

that the generated networks resemble the topological properties of the WI very well. However, the number of line intersections ($\mathcal{X}$) is about 50% more in the generated networks.

8.5 Robustness evaluation of the generated networks

We use the DC power flow equations and compare the flow distribution as well as the robustness against cascading failures in the WI and the generated networks. The power demands and supplies are estimated based on the population of the cities and power plants' capacities as described in Subsection 8.2.1. The flow statistics are very similar in the generated networks and the WI (see Table 8.2). Despite the close similarity in the flow CDFs between the WI and the generated networks, the locations of the lines that carry the top 100 flows are uncorrelated in all the four networks (see Fig. 8.7). This demonstrates that NIMBLE generates networks that not only have very similar flow characteristics to the real network, but also do not reveal the locations of the potentially vulnerable lines in the real network.

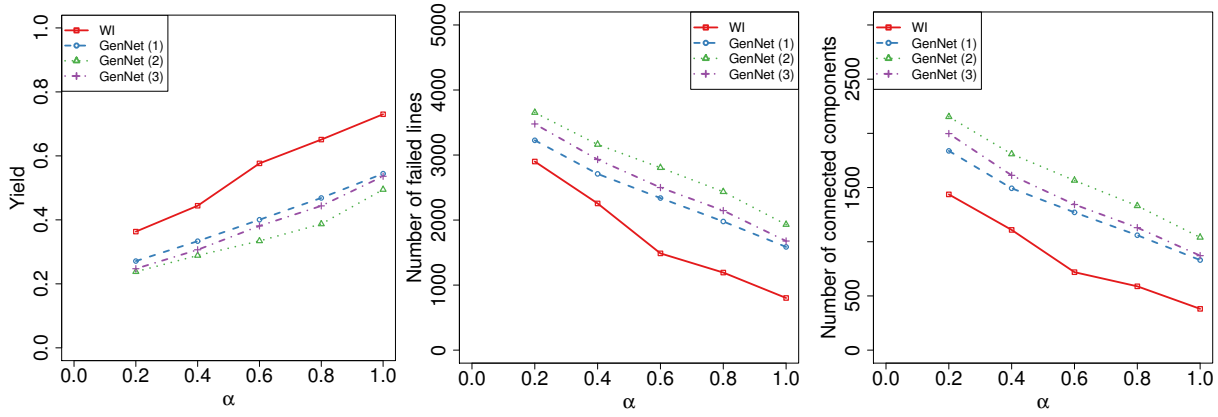
To evaluate the robustness of the generated networks compared to the WI, we study cascading failures in these networks. As an example, the evolution of the cascade in the WI and the generated networks is depicted in Fig. 8.8. We consider cascades initiated by double line failures (see Fig. 8.9) and removal of all the nodes and lines in a circular area (see Figs. 8.10 and 8.11 and also Figs. 8.12 and 8.13).



(a) WI (b) GenNet (1) (c) GenNet (2) (d) GenNet (3)

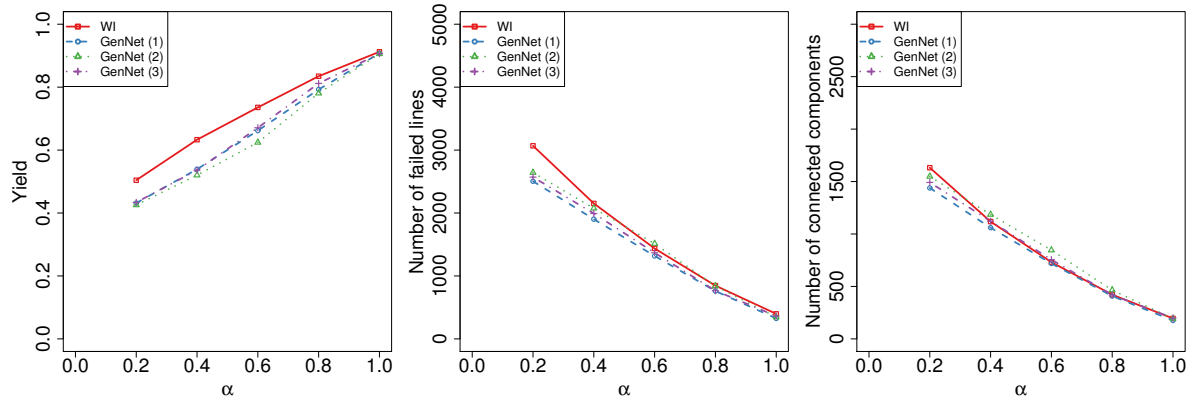
Figure 8.8: The first 15 steps of the cascades in the WI and the three generated networks. In all the networks $\alpha = 0.2$ and the cascade is initiated by a failure in the line with the highest flow, indicated with a magenta colored point. **(a)** Evolution of the cascade in the WI, which continues over 25 steps, ending with the yield = .338, number of failed lines = 3,041, and number of connected components = 1,558. **(b)** Evolution of the cascade in the GenNet (1), which continues over 15 steps, ending with the yield = .357, number of failed lines = 2,621, and number of connected components = 1,366. **(c)** Evolution of the cascade in the GenNet (2), which continues over 15 steps, ending with the yield = .257, number of failed lines = 3,573, and number of connected components = 1,946. **(d)** Evolution of the cascade in the GenNet (3), which continues over 15 steps, ending with the yield = .277, number of failed lines = 3,434, and number of connected components = 1,889.

In Fig. 8.9, we observe that the generated networks and the WI have similar level of robustness against failures in the lines that carry the highest flows (notice that as we mentioned in Fig. 8.7 these lines are in very different locations in the four networks). Figs. 8.10 and 8.11 also suggest that if a big area in the networks fails, the WI and the generated networks on *average* behave very similarly. Figs. 8.12 and 8.13 show the heatmaps of the yield for cascades initiated by area failures in the WI and the generated networks. As can be seen, despite having a very similar level of robustness, the generated networks do not reveal the vulnerabilities of the WI since the vulnerable areas in the generated networks and the WI are not in one to one correspondence.



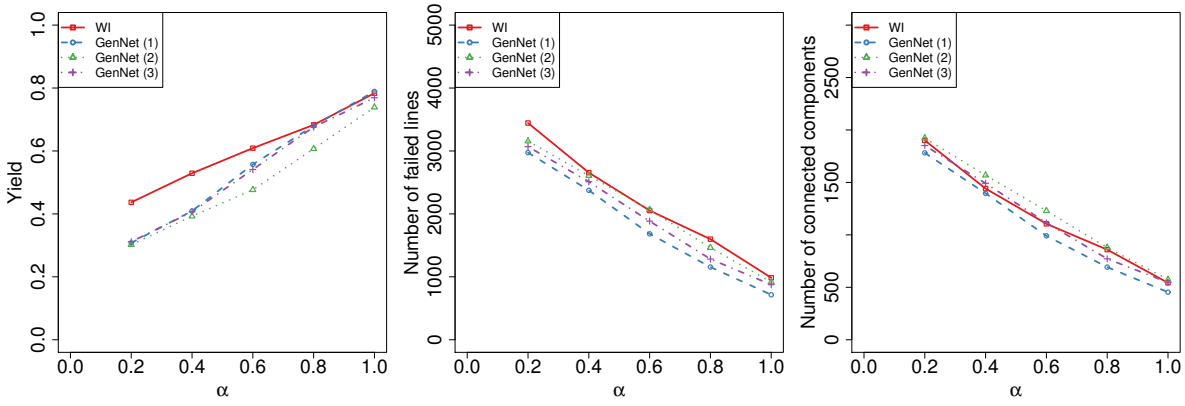
(a) (b) (c)

Figure 8.9: The severity of the cascades initiated by all possible double line failures selected from the lines that carry the top 25 largest flows in the networks as a function of the lines' factor of safety (α). **(b)** The total number of failed lines at the end of the cascade. **(c)** Number of connected components at the end of the cascade. As the cascade proceeds, the networks may get partitioned into several parts.



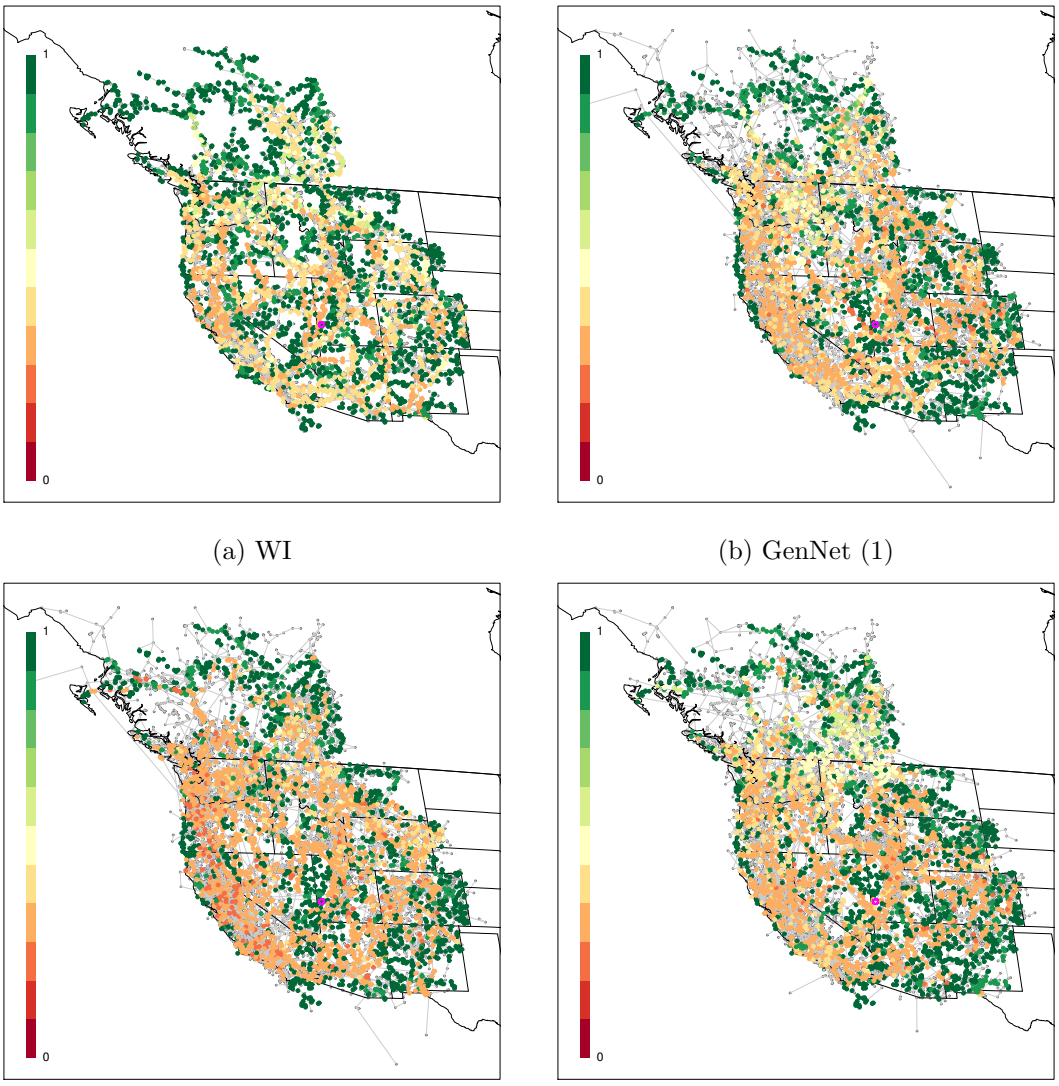
(a) (b) (c)

Figure 8.10: The average severity of the cascades initiated by failures in 10,000 uniformly distributed regions of radius 20 km as a function of lines' factor of safety (α). All the nodes and lines are removed from the initial failed area. **(a)** Yield. **(b)** The total number of failed lines at the end of the cascade. **(c)** Number of connected components at the end of the cascade.



(a) (b) (c)

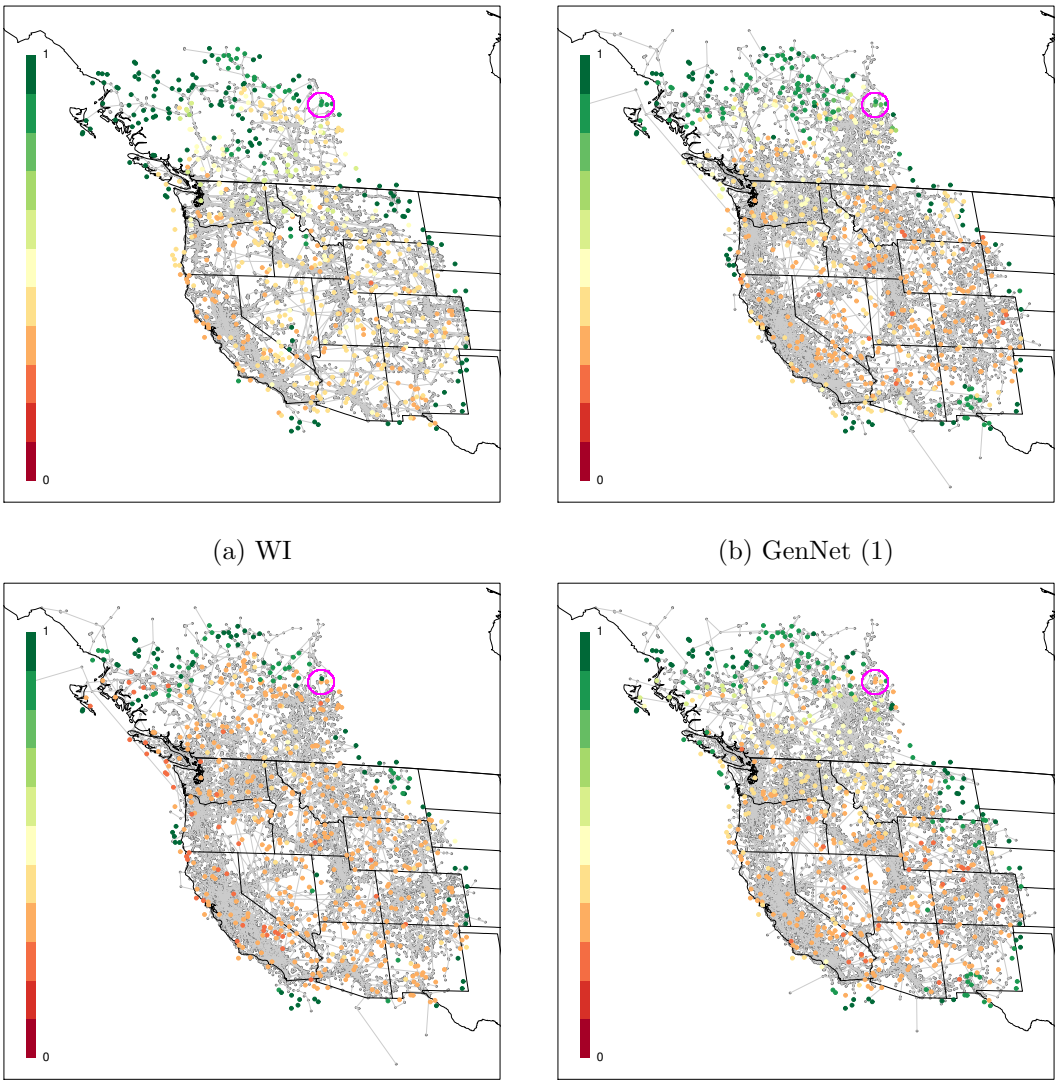
Figure 8.11: The average severity of the cascades initiated by failures in 1,000 uniformly distributed regions of radius 100 km as a function of lines' factor of safety (α). All the nodes and lines are removed from the initial failed area. The locations of failed areas are exactly the same in all the four networks. **(a)** Yield. **(b)** The total number of failed lines at the end of the cascade. **(c)** Number of connected components at the end of the cascade.



(a) WI (b) GenNet (1)

(c) GenNet (2) (d) GenNet (3)

Figure 8.12: Yield of the cascades initiated by failures in 10000 regions of radius 20 *km* uniformly distributed when $\alpha = 0.6$. The color of each point represents the yield of the cascade initiated by the failure in the area centered at that point. The size of the failed area in the map scale is shown by a magenta circle.



(a) WI (b) GenNet (1)

(c) GenNet (2) (d) GenNet (3)

Figure 8.13: Yield of the cascades initiated by failures in 1000 regions of radius 100 km uniformly distributed when $\alpha = 0.6$. The color of each point represents the yield of the cascade initiated by the failure in the area centered at that point. The size of the failed area in the map scale is shown by a magenta circle.

8.6 Conclusion

We studied the structural and spatial properties of the WI and developed the NIMBLE for generating synthetic power grid networks. We showed that the generated networks have similar structural properties as well as the same level of robustness against failures to the WI. Since the generated grids are embedded on approximately the same coordinates as the real grid, they can provide realistic test networks to the researchers.

There are several engineering aspects of power grid networks such as line nominal voltages, voltage and frequency stabilities, and transformer characteristics that were not considered in this work. In a recent paper [33], statistical properties of some of the related engineering parameters were studied in details. Incorporate these findings in our model can improve the quality of the generated networks to the level of industrial grade data sets. Moreover, we believe that if industrial grade data sets become available to the broad research community, the methods developed in this chapter can be used in order to develop synthetic grids based on them.

We also believe that our approach can be extended for generating various other types of spatially distributed networks (e.g., water and gas pipe networks).

Chapter 9

Future Directions

In this thesis, we have explored some of the critical challenges facing power grids and introduced new and unconventional approaches to address these challenges. Most of our results have a theoretical core with applications to power grids. Some algorithms we developed have applications beyond these networks. Exploring these applications and extending some of the results in this thesis can be of interest to researchers in power engineering as well as various other technical fields. We summarize some of the future research ideas and directions below.

Power Grid Monitoring and Cyber Security: Optimal and robust grid monitoring and state estimation is essential for its reliable operation and control. Almost all the aftermath reports of recent major blackouts indicated the "lack of exact grid state awareness" as a key factor in the inability of the control center to maintain the stability of the grid. Moreover, recent cyber attacks on the Ukrainian grid have revealed the importance of robust grid monitoring in the event of a cyber attack. Despite a significant amount of work that has been done to address these issues, they require more research for development of effective state estimation methods under critical scenarios. Some of the high level research plans in this direction are: (i) extend the results in Chapters 5 and 6 to address challenges in optimal measurement units placement, measurement consistency verification and false data detection, and state estimation under the AC power flow model, (ii) develop probabilistic state estimation methods using Bayesian data analysis techniques where the exact state of the grid is hard to estimate, and (iii) develop

situational awareness tools based on machine learning and data analysis algorithms capable of spatio-temporal monitoring of frequency dynamics within control areas of transmission grids for both normal and emergency operations.

Power Grid Contingency Analysis and Control: Recent blackouts have exposed the insufficiency of current protection measures and control mechanisms against failures and attacks. Studying power grid vulnerabilities and development of sophisticated control mechanisms to maintain its stability should be an essential part of the future research in this direction. In particular, one can: (i) build on the work in Chapter 4 and employ both graph theory and machine learning techniques to provide efficient and tractable methods for high order contingency analysis in power grids, (ii) extend the work in Chapter 7 for partitioning the grid into several islands as well as adapt our work to applications beyond power grid networks such as protein-protein interaction networks, social networks, and sensor networks, (iii) study new challenges introduced by the energy market to power grids operation both from economical and security points of view, (iv) study the vulnerabilities and control challenges introduced by the growth of renewable energy resources and facilitate their deployment in power grids to reduce fossil fuels consumption, and (v) study interdependencies between failures and cascades in power grids and other networks (e.g., telecommunications, gas, water, and transportation network), as well as cascades in other networked cyber-physical systems whose complexities requires interdisciplinary collaborations.

Smart Grid: The Department of Energy (DOE) envisions that by 2030, the grid will have evolved into an intelligent energy system, a smart grid [106]. This development will provide a set of challenges that are completely new to the way power grids have been traditionally operating. One of the most important promises of the smart grid–which can significantly change the picture of the grid as we know it–is the ability of consumers to sell excess electric power (produced by local wind turbines or solar cells) back to the grid. This ability introduces several security, control, and pricing challenges. Moreover, sales growth of Electric Vehicles can soon change the dynamics of the demand and supply in power grid operation. While increasing the total number of Electric Vehicles will result in a huge load on the grid, if the charging and discharging of these vehicles is done smartly by efficient algorithms, batteries within these vehicles can help to reduce

the operation cost of the grid in load peak times and also facilitate incorporating of renewable energy resources. Developing such algorithms requires more interdisciplinary research.

Bibliography

[1] IEEE benchmark systems. Available at `http://www.ee.washington.edu/research/pstca/`.

[2] National Grid UK. Available at `http://www2.nationalgrid.com/uk/services/land-and-development/planning-authority/shape-files/`.

[3] Platts GIS Data. `http://www.platts.com/Products/gisdata`.

[4] Polish grid. Available at `http://www.pserc.cornell.edu/matpower/`.

[5] Power systems test case archive. Available at: `http://www.ee.washington.edu/research/pstca/`.

[6] U.S. Energy Information Administration. Available at `http://www.eia.gov/`.

[7] U.S.-Canada Power System Outage Task Force. Report on the August 14, 2003 blackout in the United States and Canada: Causes and recommendations. `https://reports.energy.gov`, 2004.

[8] Report of the enquiry committee on grid disturbance in Northern region on 30th July 2012 and in Northern, Eastern and North-Eastern region on 31st July 2012, Aug. 2012. `http://www.powermin.nic.in/pdf/GRID_ENQ_REP_16_8_12.pdf`.

[9] The Federal Energy Regulatory Commission (FERC) and the North American Electric Reliability Corporation (NERC). Arizona-Southern California Outages on September 8, 2011,

2012. http://www.ferc.gov/legal/staff-reports/04-27-2012-ferc-nerc-report.pdf.

[10] *R: A Language and Environment for Statistical Computing*, 2014. Available at https://www.R-project.org.

[11] Avanced Research Projects Agency Energy (ARPA-E). Generating Realistic Information for the Development of Distribution and Transmission Algorithms (GRID DATA), 2015. Available at http://arpa-e.energy.gov/?q=arpa-e-programs/grid-data.

[12] Project Group Turkey. Report on Blackout in Turkey on 31st March 2015, Sept. 2015. Avilable at: https://www.entsoe.eu/Documents/SOC%20documents/Regional_Groups_Continental_Europe/20150921_Black_Out_Report_v10_w.pdf.

[13] Analysis of the cyber attack on the Ukrainian power grid, Mar. 2016. http://www.nerc.com/pa/CI/ESISAC/Documents/E-ISAC_SANS_Ukraine_DUC_18Mar2016.pdf.

[14] A. Albert. *Regression and the Moore-Penrose pseudoinverse*, volume 3. Academic Press, 1972.

[15] R. Albert, I. Albert, and G. L. Nakarado. Structural vulnerability of the North American power grid. *Phys. Rev. E*, 69(2):025103, 2004.

[16] R. Albert, H. Jeong, and A.-L. Barabási. Error and attack tolerance of complex networks. *Nature*, 406(6794):378–382, 2000.

[17] L. A. N. Amaral, A. Scala, M. Barthélémy, and H. E. Stanley. Classes of small-world networks. *PNAS*, 97(21):11149–11152, 2000.

[18] M. Anghel, K. A. Werley, and A. E. Motter. Stochastic model for power grid dynamics. In *Proc. HICSS'07*, Jan. 2007.

[19] A. Asztalos, S. Sreenivasan, B. K. Szymanski, and G. Korniss. Cascading failures in spatially-embedded random networks. *PloS one*, 9(1):e84563, 2014.

[20] T. Aura, M. Bishop, and D. Sniegowski. Analyzing single-server network inhibition. In *Proc. IEEE CSFW'00*, 2000.

[21] R. Baldick, B. Chowdhury, I. Dobson, Z. Dong, B. Gou, D. Hawkins, H. Huang, M. Joung, D. Kirschen, F. Li, et al. Initial review of methods for cascading failure analysis in electric power transmission systems IEEE PES CAMS task force on understanding, prediction, mitigation and restoration of cascading failures. In *Proc. IEEE PES-GM'08*, July 2008.

[22] R. Bapat. *Graphs and matrices*. Springer, 2010.

[23] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[24] M. Barthélemy. Spatial networks. *Physics Reports*, 499(1):1–101, 2011.

[25] A. R. Bergen and V. Vittal. *Power Systems Analysis*. Prentice-Hall, 1999.

[26] A. Bernstein, D. Bienstock, D. Hay, M. Uzunoglu, and G. Zussman. Sensitivity analysis of the power grid vulnerability to large-scale cascading failures. *ACM SIGMETRICS Perform. Eval. Rev.*, 40(3):33–37, 2012.

[27] A. Bernstein, D. Bienstock, D. Hay, M. Uzunoglu, and G. Zussman. Power grid vulnerability to geographically correlated failures - analysis and control implications. In *Proc. IEEE INFOCOM'14*, Apr. 2014.

[28] D. Bienstock. Optimal control of cascading power grid failures. *Proc. IEEE CDC-ECC*, Dec. 2011.

[29] D. Bienstock. *Electrical Transmission System Cascades and Vulnerability: An Operations Research Viewpoint*, volume 22. SIAM, 2016.

[30] D. Bienstock and A. Verma. The $N-k$ problem in power grids: New models, formulations, and numerical experiments. *SIAM J. Optimiz.*, 20(5):2352–2380, 2010.

[31] D. Bienstock and A. Verma. Strong NP–hardness of AC power flows feasibility. *arXiv:1512.07315 [math.OC]*, 2015.

[32] N. Biggs. *Algebraic graph theory*. Cambridge university press, 1994.

[33] A. B. Birchfield, K. M. Gegner, T. Xu, K. S. Shetye, and T. J. Overbye. Statistical considerations in the creation of realistic synthetic power grids for geomagnetic disturbance studies. *to appear in IEEE Trans. Power Syst.*, 2017.

[34] A. B. Birchfield, T. Xu, K. M. Gegner, K. S. Shetye, and T. J. Overbye. Grid structural characteristics as validation criteria for synthetic networks. *to appear in IEEE Trans. Power Syst.*, 2017.

[35] S. Boltz, E. Debreuve, and M. Barlaud. High-dimensional statistical measure for region-of-interest tracking. *IEEE Trans. Image Process.*, 18(6):1266–1283, 2009.

[36] J. A. Bondy and U. Murty. Graph theory, volume 244 of graduate texts in mathematics, 2008.

[37] J. A. Bondy and U. S. R. Murty. *Graph theory with applications*. Macmillan London, 1976.

[38] S. Buldyrev, R. Parshani, G. Paul, H. Stanley, and S. Havlin. Catastrophic cascade of failures in interdependent networks. *Nature*, 464(7291):1025–1028, 2010.

[39] B. Carreras, V. Lynch, I. Dobson, and D. Newman. Critical points and transitions in an electric power transmission model for cascading failure blackouts. *Chaos*, 12(4):985–994, 2002.

[40] H. Cetinay, S. Soltan, F. A. Kuipers, G. Zussman, and P. Van Mieghem. Comparing the effects of failures in power grids under the AC and DC power flow models. *Submitted for journal publication*, 2017.

[41] G. Chartrand, D. Geller, and S. Hedetniemi. Graphs with forbidden subgraphs. *J. Combin. Theory Ser. B*, 10(1):12–41, 1971.

[42] D. P. Chassin and C. Posse. Evaluating North American electric grid reliability using the barabási–albert network model. *Phys. A*, 355(2):667–677, 2005.

[43] F. Chataigner, L. R. Salgado, and Y. Wakabayashi. Approximation and inapproximability results on balanced connected partitions of graphs. *DMTCS*, 9(1):177–192, 2007.

[44] J. Chen, J. S. Thorp, and I. Dobson. Cascading dynamics and mitigation assessment in power system disturbances via a hidden failure model. *Int. J. Elec. Power and Ener. Sys.*, 27(4):318 – 326, 2005.

[45] Q. Chen and J. D. McCalley. Identifying high risk N-k contingencies for online security assessment. *IEEE Trans. on Power Syst.*, 20(2):823–834, 2005.

[46] J. Cheriyan and S. Maheshwari. Finding nonseparating induced cycles and independent spanning trees in 3-connected graphs. *J. Algorithms*, 9(4):507–537, 1988.

[47] M. Chertkov, F. Pan, and M. G. Stepanov. Predicting failures in power grids: The case of static overloads. *IEEE Trans. Smart Grid*, 2(1):162–172, 2011.

[48] J. Chlebíková. Approximating the maximally balanced connected partition problem in graphs. *Inform. Process. Lett.*, 60(5):225–230, 1996.

[49] P. Christiano, J. A. Kelner, A. Madry, D. A. Spielman, and S.-H. Teng. Electrical flows, Laplacian systems, and faster approximation of maximum flow in undirected graphs. In *Proc. ACM STOC'11*, 2011.

[50] A. Clauset, C. R. Shalizi, and M. E. Newman. Power-law distributions in empirical data. *SIAM review*, 51(4):661–703, 2009.

[51] G. Constable and B. Somerville. *A century of innovation: Twenty engineering achievements that transformed our lives*. Joseph Henry Press, 2003.

[52] E. Cotilla-Sanchez, P. D. Hines, C. Barrows, and S. Blumsack. Comparing the topological and electrical structure of the North American electric power infrastructure. *IEEE Syst. J.*, 6(4):616–626, 2012.

[53] P. Crucitti, V. Latora, and M. Marchiori. A topological analysis of the Italian electric power grid. *Phys. A*, 338(1):92–97, 2004.

[54] G. Csardi and T. Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006.

[55] G. Dán and H. Sandberg. Stealth attacks and protection schemes for state estimators in power systems. In *Proc. IEEE SmartGridComm'10*, 2010.

[56] M. M. Danziger, L. M. Shekhtman, Y. Berezin, and S. Havlin. Two distinct transitions in spatially embedded multiplex networks. *arXiv:1505.01688*, 2015.

[57] L. Daqing, J. Yinan, K. Rui, and S. Havlin. Spatial correlation analysis of cascading failures: congestions and blackouts. *Scientific reports*, 4, 2014.

[58] C. Davis and T. Overbye. Linear analysis of multiple outage interaction. In *Proc. IEEE HICSS'09*, 2009.

[59] C. Davis and T. J. Overbye. Multiple element contingency screening. *IEEE Trans. Power Syst.*, 26(3):1294–1301, 2011.

[60] I. Dobson. *Encyclopedia of Systems and Control.* Cascading network failure in power grid blackouts. Springer, 2015.

[61] I. Dobson. Electricity grid: When the lights go out. *Nature Energy*, 1:16059, 2016.

[62] I. Dobson, B. Carreras, V. Lynch, and D. Newman. Complex systems analysis of series of blackouts: cascading failure, critical points, and self-organization. *Chaos*, 17(2):026103, 2007.

[63] V. Donde, V. López, B. Lesieutre, A. Pinar, C. Yang, and J. Meza. Severe multiple contingency screening in electric power systems. *IEEE Trans. Power Syst.*, 23(2):406–417, 2008.

[64] D. L. Donoho and Y. Tsaig. Fast solution of ℓ_1-norm minimization problems when the solution may be sparse. *IEEE Trans. Inf. Theory*, 54(11):4789–4812, 2008.

[65] M. Dyer and A. Frieze. On the complexity of partitioning graphs into connected subgraphs. *Discrete Applied Math.*, 10:139–153, 1985.

[66] M. K. Enns, J. J. Quada, and B. Sackett. Fast linear contingency analysis. *IEEE Trans. Power App. Syst*, (4):783–791, 1982.

[67] M. J. Eppstein and P. D. Hines. A "random chemistry" algorithm for identifying collections of multiple contingencies that initiate cascading failure. *IEEE Trans. Power Syst.*, 27(3):1698–1705, 2012.

[68] P. Erdős and A. Rényi. On random graphs. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.

[69] A. Esmaeilian and M. Kezunovic. Prevention of power grid blackouts using intentional islanding scheme. *IEEE Trans. Ind. Appl.*, 53(1):622–629, 2017.

[70] S. Even and R. E. Tarjan. Computing an st-numbering. *Theoretical Computer Science*, 2(3):339–344, 1976.

[71] N. Fan, D. Izraelevitz, F. Pan, P. M. Pardalos, and J. Wang. A mixed integer programming approach for optimal power grid intentional islanding. *Energy Systems*, 3(1):77–93, 2012.

[72] C. Fraley, A. E. Raftery, T. B. Murphy, and L. Scrucca. mclust version 4 for R: Normal mixture modeling for model-based clustering, classification, and density estimation. Technical Report 597, Department of Statistics, University of Washington, 2012.

[73] M. Garcia, T. Catanach, S. Vander Wiel, R. Bent, and E. Lawrence. Line outage localization using phasor measurement data in transient state. *IEEE Trans. Power Syst.*, 31(4):3019–3027, 2016.

[74] M. R. Garey and D. S. Johnson. Computers and intractability: a guide to the theory of np-completeness. 1979.

[75] K. M. Gegner, A. B. Birchfield, T. Xu, K. S. Shetye, and T. J. Overbye. A methodology for the creation of geographically realistic synthetic power flow models. In *Proc. PECI'16*, Feb. 2016.

[76] J. D. Glover, M. Sarma, and T. Overbye. *Power System Analysis & Design, 4th Edition*. Cengage Learning, 2011.

[77] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in Mathematical Sciences, 4th edition, 2012.

[78] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, Mar. 2014.

[79] T. Güler and G. Gross. Detection of island formation and identification of causal factors under multiple line outages. *IEEE Trans. Power Syst.*, 22(2):505–513, 2007.

[80] E. Gyori. On division of graphs to connected subgraphs. In *Combinatorics (Proc. Fifth Hungarian Colloq., Keszthely, 1976)*, volume 1, pages 485–494, 1976.

[81] A. A. Hagberg, D. A. Schult, and P. J. Swart. Exploring network structure, dynamics, and function using NetworkX. In *Proc. SciPy'08*, Aug. 2008.

[82] P. Hines, K. Balasubramaniam, and E. C. Sanchez. Cascading failures in power grids. *IEEE Potentials*, 28(5):24–30, 2009.

[83] P. D. Hines, I. Dobson, E. Cotilla-Sanchez, and M. Eppstein. "dual graph" and "random chemistry" methods for cascading failure analysis. In *Proc. IEEE HICSS'13*, 2013.

[84] J. E. Hopcroft and R. E. Tarjan. Dividing a graph into triconnected components. *SIAM J. Comput.*, 2(3):135–158, 1973.

[85] Y.-F. Huang, S. Werner, J. Huang, N. Kashyap, and V. Gupta. State estimation in electric power grids: Meeting new challenges presented by the requirements of the future grid. *IEEE Signal Process. Mag.*, 29(5):33–43, 2012.

[86] P. Kaplunovich and K. Turitsyn. Fast and reliable screening of $N-2$ contingencies. *IEEE Trans. Power Syst.*, 31(6):4243–4252, 2016.

[87] P. Kaplunovich and K. S. Turitsyn. Statistical properties and classification of $N-2$ contingencies in large scale power grids. In *Proc. IEEE HICSS'14*, 2014.

[88] K. Khandeparkar, P. Patre, S. Jain, K. Ramamritham, and R. Gupta. Efficient PMU data dissemination in smart grid. In *Proc. ACM e-Energy'14 (poster description)*, June 2014.

[89] J. Kim and L. Tong. On topology attack of a smart grid: undetectable attacks and countermeasures. *IEEE J. Sel. Areas Commun*, 31(7):1294–1305, 2013.

[90] J. Kleinberg, M. Sandler, and A. Slivkins. Network failure detection and graph connectivity. In *Proc. ACM-SIAM SODA'04*, Jan. 2004.

[91] Y. Koç, M. Warnier, P. Van Mieghem, R. E. Kooij, and F. M. Brazier. The impact of the topology on cascading failures in a power grid model. *Phys. A*, 402:169–179, 2014.

[92] C. Lai and S. Low. The redistribution of power flow in cascading failures. In *Proc. 51st Annual Allerton Conf.*, Oct. 2013.

[93] J. Lavaei and S. Low. Zero duality gap in optimal power flow problem. *IEEE Trans. Power Syst.*, 27(1):92–107, 2012.

[94] A. Lempel, S. Even, and I. Cederbaum. An algorithm for planarity testing of graphs. In *Theory of graphs: International symposium*, volume 67, pages 215–232. Gordon and Breach, New York, 1967.

[95] N. Linial, L. Lovasz, and A. Wigderson. Rubber bands, convex embeddings and graph connectivity. *Combinatorica*, 8(1):91–102, 1988.

[96] J. Liu, C. H. Xia, N. B. Shroff, and H. D. Sherali. Distributed optimal load shedding for disaster recovery in smart electric power grids: A second-order approach. In *Proc. ACM SIGMETRICS'14 (poster description)*, June 2014.

[97] L. Liu, W. Liu, D. A. Cartes, and I.-Y. Chung. Slow coherency and angle modulated particle swarm optimization based islanding of large-scale power systems. *Adv. Eng. Inform.*, 23(1):45–56, 2009.

[98] Y. Liu, P. Ning, and M. K. Reiter. False data injection attacks against state estimation in electric power grids. *ACM Trans. Inf. Syst. Secur.*, 14(1):13, 2011.

[99] L. Lovász. A homology theory for spanning tress of a graph. *Acta Mathematica Hungarica*, 30(3-4):241–251, 1977.

[100] I. Lukovits, S. Nikolić, and N. Trinajstić. Resistance distance in regular graphs. *Int. J. of Quantum Chem.*, 71(3):217–225, 1999.

[101] S. S. Manna and P. Sen. Modulated scale-free network in Euclidean space. *Phys. Rev. E*, 66(6):066114, 2002.

[102] N. M. Manousakis, G. N. Korres, and P. S. Georgilakis. Taxonomy of PMU placement methodologies. *IEEE Trans. Power Syst.*, 27(2):1070–1077, 2012.

[103] D. Mazauric, S. Soltan, and G. Zussman. Computational analysis of cascading failures in power networks. In *Proc. ACM SIGMETRICS'13 (poster description)*, July 2013.

[104] M. A. S. Monfared, M. Jalili, and Z. Alipour. Topology and vulnerability of the Iranian power grid. *Phys. A*, 406:24–33, 2014.

[105] A. E. Motter and Y.-C. Lai. Cascade-based attacks on complex networks. *Phys. Rev. E*, 66(6):065102, 2002.

[106] National Academies of Sciences, Engineering, and Medicine. *Analytic Research Foundations for the Next-Generation Electric Grid*. The National Academies Press, 2016.

[107] S. Pahwa, M. Youssef, P. Schumm, C. Scoglio, and N. Schulz. Optimal intentional islanding to enhance the robustness of power grid networks. *Phys. A*, 392(17):3741–3754, 2013.

[108] C. H. Papadimitriou and M. Yannakakis. The complexity of restricted spanning tree problems. *JACM*, 29(2):285–309, 1982.

[109] A. Peiravi and R. Ildarabadi. A fast algorithm for intentional islanding of power systems using the multilevel kernel k-means approach. *J. Appl. Sci.*, 9(12):2247–2255, 2009.

[110] R. Pfitzner, K. Turitsyn, and M. Chertkov. Controlled tripping of overheated lines mitigates power outages. *ArXiv e-prints*, Oct. 2011.

[111] R. Pfitzner, K. Turitsyn, and M. Chertkov. Statistical classification of cascading failures in power grids. In *Proc. IEEE PES-GM'11*, 2011.

[112] C. Phillips. The network inhibition problem. In *Proc. ACM STOC'93*, May 1993.

[113] A. Pinar, J. Meza, V. Donde, and B. Lesieutre. Optimization strategies for the vulnerability analysis of the electric power grid. *SIAM J. Optimiz.*, 20(4):1786–1810, 2010.

[114] S. Poudel, Z. Ni, and W. Sun. Electrical distance approach for searching vulnerable branches during contingencies. *IEEE Trans. Smart Grid*, 2016.

[115] W. H. Press. *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, 2007.

[116] K. Purchala, L. Meeus, D. Van Dommelen, and R. Belmans. Usefulness of DC power flow for active power flow analysis. In *IEEE PES-GM'05*, June 2005.

[117] G. Ranjan, Z.-L. Zhang, and D. Boley. Incremental computation of pseudo-inverse of Laplacian: Theory and applications. *arXiv:1304.2300*, Apr. 2013.

[118] M. Rosas-Casals, S. Valverde, and R. V. Solé. Topological vulnerability of the European power grid under errors and attacks. *Int. J. Bifurcat. Chaos*, 17(07):2465–2475, 2007.

[119] W. Rudin. *Real and complex analysis*. McGraw-Hill Education, 1987.

[120] J. Salmeron, K. Wood, and R. Baldick. Analysis of electric grid security under terrorist threat. *IEEE Trans. Power Syst.*, 19(2), 2004.

[121] R. J. Sánchez-García, M. Fennelly, S. Norris, N. Wright, G. Niblo, J. Brodzki, and J. W. Bialek. Hierarchical spectral clustering of power grids. *IEEE Trans. Power Syst.*, 29(5):2229–2237, 2014.

[122] P. Schultz, J. Heitzig, and J. Kurths. A random growth model for power grids and other spatially embedded infrastructure networks. *Eur. Phys. J. Spec. Top.*, 223(12):2593–2610, 2014.

[123] P. D. Seymour. On odd cuts and plane multicommodity flows. *P. Lond. Math. Soc.*, 3(1):178–192, 1981.

[124] R. V. Solé, M. Rosas-Casals, B. Corominas-Murtra, and S. Valverde. Robustness of the European power grids under intentional attack. *Phys. Rev. E*, 77(2):026102, 2008.

[125] S. Soltan, A. Loh, and G. Zussman. Analyzing and quantifying the effect of k-line failures in power grids. *to appear in IEEE Trans. Control Netw. Syst.*, 2017.

[126] S. Soltan, A. Loh, and G. Zussman. A learning-based method for generating synthetic power grids. *Submitted for journal publication*, 2017.

[127] S. Soltan, D. Mazauric, and G. Zussman. Cascading failures in power grids: analysis and algorithms. In *Proc. ACM e-Energy'14*, June 2014.

[128] S. Soltan, D. Mazauric, and G. Zussman. Analysis of failures in power grids. *in IEEE Trans. Control Netw. Syst.*, 4(3):288–300, 2017.

[129] S. Soltan, M. Yannakakis, and G. Zussman. Joint cyber and physical attacks on power grids: Graph theoretical approaches for information recovery. In *Proc. ACM SIGMETRICS'15*, July 2015.

[130] S. Soltan, M. Yannakakis, and G. Zussman. Doubly balanced connected graph partitioning. In *Proc. ACM-SIAM SODA'17*, Jan. 2017.

[131] S. Soltan, M. Yannakakis, and G. Zussman. Doubly balanced connected graph partitioning. *Submitted for journal publication*, 2017.

[132] S. Soltan, M. Yannakakis, and G. Zussman. Power grid state estimation following a joint cyber and physical attack. *to appear in IEEE Trans. Control Netw. Syst. (available on IEEE Xplore Digital Library)*, 2017.

[133] S. Soltan and G. Zussman. A statistical method for synthetic power grid generation based on the U.S. Western Interconnection. Abstract presented at *SIAM NS'15*, May 2015.

[134] S. Soltan and G. Zussman. Generation of synthetic spatially embedded power grid networks. *arXiv:1508.04447*, Aug. 2015.

[135] S. Soltan and G. Zussman. Generation of synthetic spatially embedded power grid networks. In *Proc. IEEE PES-GM'16*, July 2016.

[136] S. Soltan and G. Zussman. Quantifying the effect of k-line failures in power grids. In *Proc. IEEE PES-GM'16*, July 2016.

[137] S. Soltan and G. Zussman. Power grid state estimation after a cyber-physical attack under the AC power flow model. In *Proc. IEEE PES-GM'17*, July 2017.

[138] R. Spiewak, S. V. Buldyrev, Y. Forman, S. Soltan, and G. Zussman. A study of cascading failures in real and synthetic power grid topologies using DC power flows. *arXiv preprint arXiv:1609.07395*, 2016.

[139] K. Sun, Q. Zhao, D.-Z. Zheng, J. Ma, and Q. Lu. A two-phase method based on obdd for searching for splitting strategies of large-scale power systems. In *Proc. IEEE PowerCon'02*, 2002.

[140] K. Sun, D.-Z. Zheng, and Q. Lu. Splitting strategies for islanding operation of large-scale power systems using obdd-based methods. *IEEE Trans. Power Syst.*, 18(2):912–923, 2003.

[141] H. Suzuki, N. Takahashi, and T. Nishizeki. A linear algorithm for bipartition of biconnected graphs. *Inform. Process. Lett.*, 33(5):227–231, 1990.

[142] J. E. Tate and T. J. Overbye. Line outage detection using phasor angle measurements. *IEEE Trans. Power Syst.*, 23(4):1644–1652, 2008.

[143] J. E. Tate and T. J. Overbye. Double line outage detection using phasor angle measurements. In *Proc. IEEE PES-GM'09*, July 2009.

[144] K. Turitsyn and P. Kaplunovich. Fast algorithm for $N-2$ contingency problem. In *Proc. IEEE HICSS'13*, 2013.

[145] W. T. Tutte. How to draw a graph. *Proc. London Math. Soc*, 13(3):743–768, 1963.

[146] V. Vittal, W. Kliemann, Y.-X. Ni, D. Chapman, A. Silk, and D. Sobajic. Determination of generator groupings for an islanding scheme in the manitoba hydro system using the method of normal forms. *IEEE Trans. Power Syst.*, 13(4):1345–1351, 1998.

[147] B. A. Carreras, V. E. Lynch, I. Dobson, and D. E. Newman. Complex dynamics of blackouts in power transmission systems. *Chaos*, 14(3):643–652, 2004.

[148] O. Vukovic, K. C. Sou, G. Dán, and H. Sandberg. Network-layer protection schemes against stealth attacks on state estimators in power systems. In *Proc. IEEE SmartGridComm'11*, 2011.

[149] K. Wada and K. Kawaguchi. Efficient algorithms for tripartitioning triconnected graphs and 3-edge-connected graphs. In *Graph-Theoretic Concepts in Computer Science*, pages 132–143. Springer, 1994.

[150] X. Wang, X. Song, and J. Yuan. On matching cover of graphs. *Math. Program.*, 147(1-2):499–518, 2014.

[151] Z. Wang, A. Scaglione, and R. J. Thomas. Electrical centrality measures for electric power grid vulnerability analysis. In *Proc. IEEE CDC'10*, 2010.

[152] Z. Wang, A. Scaglione, and R. J. Thomas. Generating statistically correct random topologies for testing smart grid communication and control networks. *IEEE Trans. Smart Grid*, 1(1):28–39, 2010.

[153] D. J. Watts and S. H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393(6684):440–442, 1998.

[154] D. B. West et al. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, 2001.

[155] A. J. Wood and B. F. Wollenberg. *Power generation, operation, and control*. John Wiley & Sons, 3rd edition, 2012.

[156] H. Xiao and E. M. Yeh. Cascading link failure in the power grid: A percolation-based analysis. In *Proc. IEEE Int. Work. on Smart Grid Commun.*, June 2011.

[157] R. Xulvi-Brunet and I. M. Sokolov. Evolving networks with disadvantaged long-range connections. *Phys. Rev. E*, 66(2):026118, 2002.

[158] H. You, V. Vittal, and Z. Yang. Self-healing in power systems: an approach using islanding and rate of frequency decline-based load shedding. *IEEE Trans. Power Syst.*, 18(1):174–181, 2003.

[159] J. Zhao, D. Li, H. Sanhedrai, R. Cohen, and S. Havlin. Spatio-temporal propagation of cascading overload failures in spatially embedded networks. *Nat. Commun.*, 7, 2016.

[160] Q. Zhao, K. Sun, D.-Z. Zheng, J. Ma, and Q. Lu. A study of system splitting strategies for island operation of power system: a two-phase method based on obdds. *IEEE Trans. Power Systems*, 18(4):1556–1565, 2003.

[161] Y. Zhao, A. Goldsmith, and H. V. Poor. On PMU location selection for line outage detection in wide-area transmission networks. In *Proc. IEEE PES-GM'12*, July 2012.

[162] Q. Zhou and J. W. Bialek. Approximate model of European interconnected system as a benchmark system to study effects of cross-border trades. *IEEE Trans. Power Syst.*, 20(2):782–788, 2005.

[163] H. Zhu and G. B. Giannakis. Sparse overcomplete representations for efficient identification of power line outages. *IEEE Trans. Power Syst.*, 27(4):2215–2224, 2012.

[164] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas. Matpower: Steady-state operations, planning, and analysis tools for power systems research and education. *IEEE Trans. Power Syst.*, 26(1):12–19, 2011.

[165] D. Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. In *Proc. ACM STOC'06*, May 2006.