5-9-2016

In-situ Data Analytics In Cyber-Physical Systems

Liang Zhao

IN-SITU DATA ANALYTICS IN CYBER-PHYSICAL SYSTEMS

by

LIANG ZHAO

Under the Direction of WenZhan Song, PhD

ABSTRACT

Cyber-Physical System (CPS) is an engineered system in which sensing, networking, and computing are tightly coupled with the control of the physical entities. To enable security, scalability and resiliency, new data analytics methodologies are required for computing, monitoring and optimization in CPS. This work investigates the data analytics related challenges in CPS through two study cases: Smart Grid and Seismic Imaging System.

For smart grid, this work provides a complete solution for system management based on novel in-situ data analytics designs. We first propose methodologies for two important tasks of power

system monitoring: grid topology change and power-line outage detection. To address the issue of low measurement redundancy in topology identification, particularly in the low-level distribution network, we develop a maximum a posterior based mechanism, which is capable of embedding prior information on the breakers status to enhance the identification accuracy. In power-line outage detection, existing approaches suffer from high computational complexity and security issues raised from centralized implementation. Instead, this work presents a distributed data analytics framework, which carries out in-network processing and invokes low computational complexity, requiring only simple matrix-vector multiplications. To complete the system functionality, we also propose a new power grid restoration strategy involving data analytics for topology reconfiguration and resource planning after faults or changes.

In seismic imaging system, we develop several innovative in-situ seismic imaging schemes in which each sensor node computes the tomography based on its partial information and through gossip with local neighbors. The seismic data are generated in a distributed fashion originally. Different from the conventional approach involving data collection and then processing in order, our proposed in-situ data computing methodology is much more efficient. The underlying mechanisms avoid the bottleneck problem on bandwidth since all the data are processed distributed in nature and only limited decisional information is communicated. Furthermore, the proposed algorithms can deliver quicker insights than the state-of-arts in seismic imaging. Hence they are more promising solutions for real-time in-situ data analytics, which is highly demanded in disaster monitoring related applications. Through extensive experiments, we demonstrate that the proposed data computing methods are able to achieve near-optimal high quality seismic tomography, retain low communication cost, and provide real-time seismic data analytics.

INDEX WORDS: Cyber-Physical System, Smart Grid, Seismic Imaging System, In-Network Processing, In-Situ Analytics, Distributed Computing.

IN-SITU DATA ANALYTICS IN CYBER-PHYSICAL SYSTEMS

by

LIANG ZHAO

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

in the College of Arts and Sciences

Georgia State University

2016

IN-SITU DATA ANALYTICS IN CYBER-PHYSICAL SYSTEMS

by

LIANG ZHAO

Committee Chair: WenZhan Song

Committee: Yanqing Zhang

Zhipeng Cai

Xiaojing Ye

Electronic Version Approved:

Office of Graduate Studies

College of Arts and Sciences

Georgia State University

May 2016

DEDICATION

This dissertation is dedicated to Georgia State University. This dissertation is also dedicated to my

parents and wife for their endless support, sacrifice, hard work, love and passion for academics.

ACKNOWLEDGEMENTS

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

- ADMM - Alternating Direction Method of Multipliers

- CPS - Cyber-Physical System

- SDP - Semidefinite Programming

- RES - Renewable Energy Resources

- PMU - Phasor Measurement Unit

PART 1

INTRODUCTION

In this chapter we introduce the main background of the two topics in the dissertation: the *Smart Grid* and the *Seismic Imaging System*.

1.1 Smart Grid

The evolving modern Smart Grid is devoted to leverage the information and communications technology to enrich the efficiency, reliability and sustainability of the operation of the energy. Phasor measurement units (PMUs) directly obtain the complex voltages and currents; Smart meters implemented between the end-users and the distribution network; Network processors are being equipped over the grid. Those ample measurements of data offer much more powerful potential monitoring capabilities than traditional grid. However, the way of data communication, computation and analytics becomes a key challenge. An overview of smart grid architecture is illustrated in Figure 1.1.
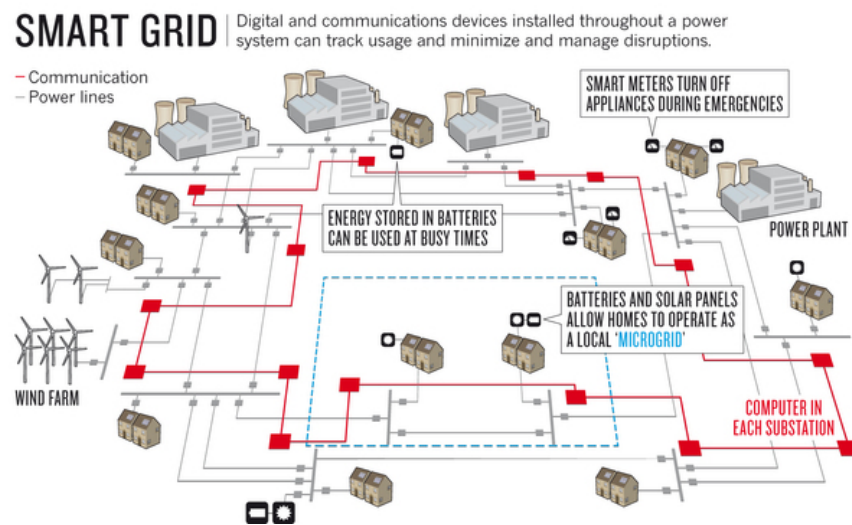
Figure (1.1) Overview of smart grid achitecture [1].

Three important system tasks are considered in this work: topology identification, power-line outage detection, and grid reconfiguration. A important feature of smart grid is the flexibility in network topology. Possible bi-direction energy flows, allowing for various distributed generation from photovoltaic panels, wind turbines, and other sources [3]. In this situation, an accurate identification of grid topology becomes an imperative task since correct topology information consists of a basis for various system analyse functions such as contingency analysis aiming to improve the reliability of the system [4]. Similar as the topology identification, power-line outage detection is also a core functionality. However, this task is extremely difficult, due to its vast scale and the use of distributed energy generation and storage. Thus, a smart power-line outage detection calls for a framework integrating distributed computing, communication and analytics in which local actions can be coordinated for the effective protection of the power grid as a whole. Once the fault has been found and cleared, a strategy to reconfigure the grid is required. The capability of self-reconfiguration is a leading role [6] in smart grid development. It enables smart grids to redirect their power flows in an appropriate way, e.g., via shedding loads, and other control measures, to achieve certain desirable objectives. Considering the great penetration of renewable energy sources (RES) in smart microgrids, the self-reconfiguration of smart microgirds needs a careful design to address the volatility in the RES generations [7], [8].

1.2   Seismic Imaging System

Besides the smart grid study case, this work also investigates another example of CPS: seismic imaging system. Current seismic imaging systems use sensors placed on earth surface to acquire information of the compressional waves generated by underground seismic activities. The acquired data are then used to derive the internal velocity structure of the earth subsurface. However, they are prone to several bottleneck problems. First, for example, in previous volcano monitoring investigations, the number of stations (sensors) is up to 20 due to deployment cost and other issues. The resulting low station coverage inevitably becomes a main constraint on our capability of obtaining high-resolution tomography model [9]. Second, even if thousands of nodes can be incorporated (e.g. petroleum exploration systems), the huge volume of raw seismic data has to be collected into

a central place for post-processing. The underlying time-consuming process prohibits its potential for effective disaster warning in which the time scale can be tens of minutes. We adopt the travel-time based seismic tomography in the presenting work to reveal the velocity model inside the volcano [82]. The basic procedure of seismic tomography is illustrated in Figure 1.2 involving three steps.
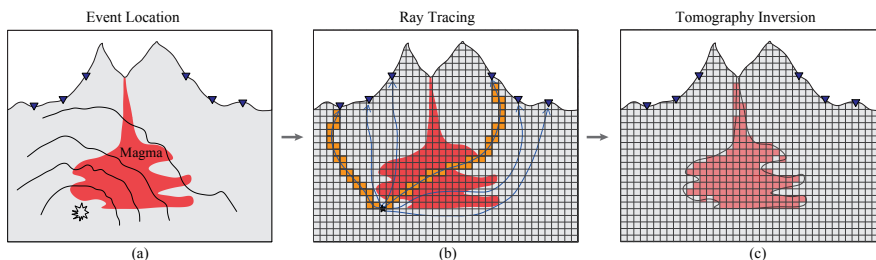
Figure (1.2) Procedure of Seismic Tomography

The first step is "*Event Location*", which means we need estimate the where and when certain seismic event occurs. (Figure 1.2(a)). The second step is named "*Ray tracing*". This process is to estimate the rays coming from the the event location to the receivers. The traces of the rays are affected by the velocity structures of the materials they travel along. In other words, the rays contain information of the internal velocity model that we are interested in (see Figure 1.2(b) as a conceptual view of this process). The final step is "*Tomographic Inversion*". It basically utilizes the traced ray paths to image a 3-D tomography of the velocity model within the volcano (Figure 1.2(c)). In this thesis, we focus on the tomographic inversion process, which can be formulated as a large linear inversion problem. However, it is highly demanded to have a distributed data analytics platform to fill the gap between the "centralized" nature of traditional seismic tomography computing and the "distributed" feature of sensor networks.

1.3 Our Approach

From the previous two subsections, we realize that a paradigm-shifting data computing platform is highly demanded to fit the needs of implementing smart power grid and new seismic imaging system. Our approach is named in-situ data analytics in which in-network computing
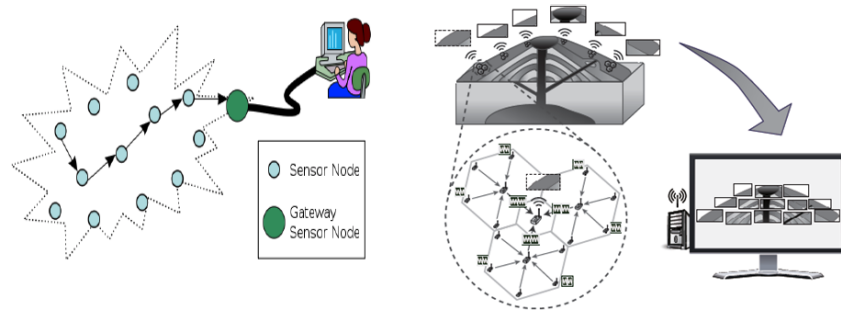
Figure (1.3) Big Data Analytics in Distributed Networks. Left: traditional approach. Right: in-situ computing.

is performed to deliver the insights on the interested quantity (See the right part of Figure 1.3). Traditional approach for data analytics in distributed networks is data collection/aggregation then computing, which is slow and expensive. In contrast, the proposed in-situ computing platform is more efficient and enables real-time data analytics. In the two study cases, we model their challenges as computing problems in distributed systems and develop various algorithms based on the idea of in-situ data analytics.

The rest of thesis is organized as follows. In part 2, we present our in-situ data analytics approach for topology identification, power-line outage detection, and grid reconfiguration in smart grid, respectively. Next in part 3, we propose several decentralized algorithms for in-situ seismic imaging system. Finally in part 4 we conclude this thesis.

PART 2

IN-SITU DATA ANALYTICS IN SMART GRID

In this part, we apply in-situ data analytics approach to address three important challenges in smart grid. In the first stage, we consider two tasks: topology identification and power-line outage detection, which are imperatives for the functionality of fault/change detection in the system. In the second stage, we investigate the problem of designing grid reconfiguration strategy, especially after certain faults/changes have been detected and cleared. The workflow is depicted in Figure 2.1.
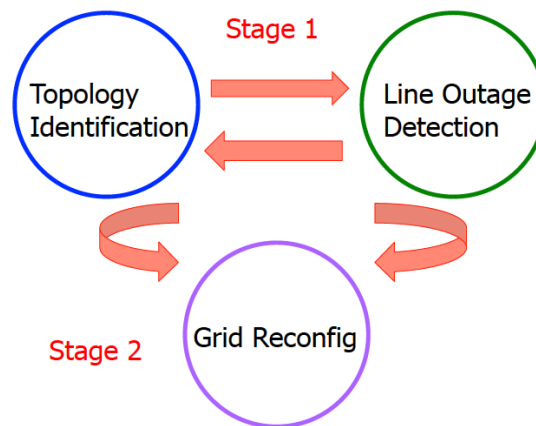
Figure (2.1) Workflow of the smart grid study case.

2.1 Topology Identification with Limited Measurements

2.1.1 Related Work

In grid management, a critical task of a system operator is to take quick actions to restore continuity of electric power supply following forced outages. However, many existing outage management system (OMS) are still based on the process of call aggregation, which can take from tens of minutes to hours to identify the culprit device. Thus, advanced smart grid systems call for au-

tomatic detection schemes, which will dramatically improve the quality of service. Unfortunately, it is said that in particular the distribution networks today may still have only few measurements at the substation. On the other hand, the number of circuit breakers is typically larger than the available measurements [10].

In this thesis, we devise a novel framework capable of successfully infer the true network topology with high probability under limited measurements. Based on the prior information of circuit breaker status, the topology identification can be posed as a maximum a posterior (MAP) estimation problem. Since the formulated MAP is a NP-hard problem which is very difficult to solve exactly, we adopt semidefinite programming (SDP) relaxation to obtain an approximated easier optimization problem. By exploiting compressive sensing advances, the objective function of the aforementioned problem is regularized by the ℓ_1-norm of selected vectors, which explores sparsity of the change of status of circuit breakers.
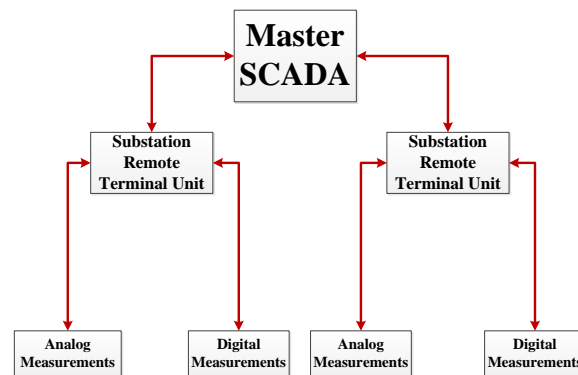
Figure (2.2) SCADA Topology. Analog measurements can be from phasor measurement units (PMUs), power flow meters or power injection meters, etc. Digital measurements contain readings from circuit breaker monitors installed in the substation.

2.1.2 Problem Formulation

In this section we describe the model in detail, and the basic maximum a posteriori probability (MAP) method for estimating the topology pattern.

Topology Pattern Model and Prior Distribution We consider a system with N circuit breakers. We represent a topology pattern, i.e., as a vector $\mathbf{s} \in \{0, 1\}^N$, where $\mathbf{s}_i = 1$ means that circuit breaker i is open and $\mathbf{s}_i = 0$ means breaker is closed. Since circuit breaker is normally-closed or open (circuit breaker operates rarely), leveraging this prior knowledge will improve the accuracy of topology identification from an information theory perspective [51]. For instance, consider the case that the probability of a circuit breaker is open equals to 0.5, then if no measurement is available, we can only guess the status of breaker by tossing a coin. However, in another case when the probability of a breaker is open equals to 0.99, we can always decide the breaker is open with a very high accuracy even without any measurement. In other words, the prior knowledge in the later case is more "*informative*" than the first case. We assume that circuit breakers act independently, and the open state for circuit breaker i occurs with known probability p_i based on the historical data. That is:

$$p(\mathbf{s}_i) = \begin{cases} p_i & \mathbf{s}_i = 1; \\ 1 - p_i & \mathbf{s}_i = 0; \end{cases} \tag{2.1}$$

Thus, the (prior) probability of topology pattern $\mathbf{s}$ occurring is:

$$p(\mathbf{s}) = \prod_{i=1}^{N} p_i^{\mathbf{s}_i}(1 - p_i)^{1-\mathbf{s}_i} \tag{2.2}$$

Measurement Model We combine two types of measurements: Analog measurements and Digital measurements to infer the true topology and state of the system. $\mathbf{z}_a$ denotes the vector of analog measurements including readings from PMUs, power flow meters and power injection meters. $\mathbf{z}_d$ represents the vector of digital measurements from circuit breaker monitors with values 1 or 0 indicating the open/closed status of the breaker. We will explain the relation between state variables and analog measurements, circuit breaker status and digital measurements and the constraint between state variables and breaker status.

We define the state of the system $\mathbf{x} \in \mathbb{R}^K$ as the voltage phasor angles in all the K buses (substations) and assume M analog measurements are available. These measurements depend on

the state $\mathbf{x}$ by the following linear model:

$$\mathbf{z}_a = \mathbf{H}_a\mathbf{x} + \mathbf{v} \qquad (2.3)$$

where $\mathbf{H}_a \in \mathbb{R}^{M \times K}$ is the analog measurement matrix. $\mathbf{v}$ is the measurement noise vector which is assumed to be Gaussian distributed, i.e., $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Lambda})$ and each element of $\mathbf{v}$ is independent of each other and $\mathbf{x}$, which implies that the covariance matrix $\mathbf{\Lambda}$ is diagonal. Explicitly, the diagonal element $\mathbf{\Lambda}_{ii}$ equals to σ^2 for $i = 1, 2, \ldots, M$.

Similar as the analog measurement model, we assume L digital measurements can be obtained, thus, the equation relating the digital measurements to the breaker status $\mathbf{s}$ can be denoted as follows.

$$\mathbf{z}_d = \mathbf{H}_d\mathbf{s} \qquad (2.4)$$

where $\mathbf{z}_d \in \{0, 1\}^L$, $\mathbf{H}_d \in \mathbb{R}^{L \times N}$ and each row of $\mathbf{H}_d$ is with values of zeros except one entry of 1 that corresponds to the measurement in that breaker. We assume digital measurements $\mathbf{z}_d$ are communicated through reliable channels which implies (2.4) is a noise-free model. In network topology identification scenario which we focus on in this paper, the above set of linear equations is typically assumed to be under-determined ($L < N$), which means that the number of digital measurements is smaller than the number of circuit breakers.

Since both the analog and digital measurements will be used to identify the topology with the state estimation stage, the relationship between state $\mathbf{x}$ and topology pattern $\mathbf{s}$ can be modeled in the following way [52]: Define the switch-bus incident matrix $\mathbf{A} \in \mathbb{R}^{L \times K}$ with the properties of: 1) its i-th row $\mathbf{a}_i^T$ corresponds to the (m, n) switch between buses m and n with $m < n$; 2) the m-th (n-th) entry of the vector a_i^T equals to $+1(-1)$ and zero elsewhere. When the (m, n) switch is closed, the voltage phasor angle difference between its two ends is zero, and this can be expressed as: $\mathbf{a}_i^T\mathbf{x} = 0$.

Posterior Probability   Let $p(\mathbf{x}, \mathbf{s}|\mathbf{z}_a, \mathbf{z}_d)$ to be the posterior probability of state $\mathbf{x}$ and topology pattern $\mathbf{s}$ given the measurement $\mathbf{z}_a$ and $\mathbf{z}_d$. Based on Bayes' rule we can have:

$$
\begin{aligned}
p(\mathbf{x}, \mathbf{s}|\mathbf{z}_a, \mathbf{z}_d) &= \frac{p(\mathbf{z}_a, \mathbf{z}_d|\mathbf{x}, \mathbf{s})p(\mathbf{x}, \mathbf{s})}{p(\mathbf{z}_a, \mathbf{z}_d)} \\
&\propto p(\mathbf{z}_a, \mathbf{z}_d|\mathbf{x}, \mathbf{s})p(\mathbf{x}, \mathbf{s}) \\
&\propto p(\mathbf{z}_a, \mathbf{z}_d, \mathbf{x}, \mathbf{s})
\end{aligned}
\tag{2.5}
$$

where $p(\mathbf{z}_a, \mathbf{z}_d, \mathbf{x}, \mathbf{s})$ is the joint probability of $(\mathbf{z}_a, \mathbf{z}_d, \mathbf{x}, \mathbf{s})$. We assume that they are mutually independent, and by using (2.2) and (2.3) we can derive:

$$
\begin{aligned}
p(\mathbf{z}_a, \mathbf{z}_d, \mathbf{x}, \mathbf{s}) &= p(\mathbf{z}_a)p(\mathbf{s})p(\mathbf{z}_d)p(\mathbf{x}) \\
&= \mathcal{N}(\mathbf{H}_a\mathbf{x}, \mathbf{\Lambda}) \prod_{i=1}^{N} p_i^{\mathbf{s}_i}(1 - p_i)^{1-\mathbf{s}_i} p(\mathbf{z}_d)p(\mathbf{x})
\end{aligned}
\tag{2.6}
$$

where $\mathcal{N}(\mathbf{H}_a\mathbf{x}, \mathbf{\Lambda})$ means the probability of analog measurements $\mathbf{z}_a$. It is Gaussian distributed with mean $\mathbf{H}_a\mathbf{x}$ and covariance matrix $\mathbf{\Lambda}$. Note that (2.4) is noise-free, thus $p(\mathbf{z}_d) = 1$. With respect to $p(\mathbf{x})$, we assume no "*informative*" prior information on $\mathbf{x}$ is available. In consequence, $\mathbf{x}$ can be assumed uniform distributed which implies $p(\mathbf{x}) = c$, where c is a constant. Then (2.6) is simplified as:

$$
p(\mathbf{z}_a, \mathbf{z}_d, \mathbf{x}, \mathbf{s}) = c\mathcal{N}(\mathbf{H}_a\mathbf{x}, \mathbf{\Lambda}) \prod_{i=1}^{N} p_i^{\mathbf{s}_i}(1 - p_i)^{1-\mathbf{s}_i}
\tag{2.7}
$$

In next section, we will see "prior information" on $\mathbf{x}$ will not influence our MAP estimation under this setting.

MAP Estimation   The maximum a posteriori estimation can be rewritten as:

$$
\max \; p(\mathbf{z}_a, \mathbf{z}_d, \mathbf{x}, \mathbf{s}) = \max_{\mathbf{x}, \mathbf{s}} \; \mathcal{N}(\mathbf{H}_a\mathbf{x}, \mathbf{\Lambda}) \prod_{i=1}^{N} p_i^{\mathbf{s}_i}(1 - p_i)^{1-\mathbf{s}_i}
\tag{2.8}
$$

Define the log-loss function as the negative log probability $l_{\mathbf{z}_a, \mathbf{z}_d}(\mathbf{x}, \mathbf{s}) = -\log p(\mathbf{z}_a, \mathbf{z}_d, \mathbf{x}, \mathbf{s})$, then it is equivalent to reformulate the objective as follows:

$$
\begin{aligned}
\min_{\mathbf{x}, \mathbf{s}} \; l_{\mathbf{z}_a, \mathbf{z}_d}(\mathbf{x}, \mathbf{s}) &= -\log \left(\mathcal{N}(\mathbf{H}_a \mathbf{x}, \mathbf{\Lambda}) \prod_{i=1}^{N} p_i^{\mathbf{s}_i} (1 - p_i)^{1 - \mathbf{s}_i} \right) \\
&= -\log \left(\mathcal{N}(\mathbf{H}_a \mathbf{x}, \mathbf{\Lambda}) \right) - \log \left(\prod_{i=1}^{N} p_i^{\mathbf{s}_i} (1 - p_i)^{1 - \mathbf{s}_i} \right) \\
&= \frac{1}{2} (\mathbf{z}_a - \mathbf{H}_a \mathbf{x})^T \mathbf{\Lambda}^{-1} (\mathbf{z}_a - \mathbf{H}_a \mathbf{x}) + \mathbf{b}^T \mathbf{s} + \mathbf{C} \\
&= \frac{1}{2\sigma^2} (\mathbf{z}_a - \mathbf{H}_a \mathbf{x})^T (\mathbf{z}_a - \mathbf{H}_a \mathbf{x}) + \mathbf{b}^T \mathbf{s} + \mathbf{C} \\
&= \frac{1}{2\sigma^2} \mathbf{x}^T \mathbf{H}_a^T \mathbf{H}_a \mathbf{x} - 2\mathbf{z}_a^T \mathbf{H}_a \mathbf{x} + \mathbf{b}^T \mathbf{s} + \mathbf{C}'
\end{aligned}
\tag{2.9}
$$

where $\mathbf{\Lambda}$ is replaced by $\sigma^2 \mathbf{I}$, and $\mathbf{b} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_N]$ with $\mathbf{b}_i = \log((1 - p_i)/p_i)$. $\mathbf{C}$ and $\mathbf{C}'$ are constants that are independent of the optimization variables $\mathbf{x}$ and $\mathbf{s}$.

To ease the presentation, we define:

$$
\begin{aligned}
\mathbf{Q} &= \tfrac{1}{2\sigma^2} \mathbf{H}_a^T \mathbf{H}_a \\
\mathbf{q}^T &= -2\mathbf{z}_a^T \mathbf{H}_a
\end{aligned}
\tag{2.10}
$$

Then (2.9) can be reposed as:

$$
\underset{\mathbf{x}, \mathbf{s}}{\text{minimize}} \; \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{q}^T \mathbf{x} + \mathbf{b}^T \mathbf{s}
\tag{2.11}
$$

Now we add the aforementioned constraints of the problem into (2.11), the complete MAP estimation problem becomes as follows:

$$
\begin{aligned}
\underset{\mathbf{x}, \mathbf{s}}{\text{minimize}} \quad & \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{q}^T \mathbf{x} + \mathbf{b}^T \mathbf{s} \\
\text{subject to} \quad & \mathbf{z}_d = \mathbf{H}_d \mathbf{s} \\
& \mathbf{A}_c \mathbf{x} = \mathbf{0} \\
& \mathbf{s} \in \{0, 1\}^N
\end{aligned}
\tag{2.12}
$$

where $\mathbf{A}_c$ is a sub-matrix of $\mathbf{A}$ (switch-bus matrix) corresponding to the set of circuit breakers which are measured to be in the closed status. Notice that (2.12) is a standard form of mixed $0 - 1$ integer quadratic optimization problem which is known to be NP-hard. We can analyze why (2.12) is very hard to solve from an optimization point of view. From the definition of $\mathbf{H}_a$, it can be deduced that matrix $\mathbf{Q}$ is symmetric positive semidefinite. Consequently, the objective of (2.12) is a convex quadratic function of $\mathbf{x}$ and $\mathbf{s}$. In other words, the difficulty of this problem results from the constraint. The constraint $\mathbf{s} \in \{0, 1\}^N$ is non-convex which makes (2.12) to be a non-convex problem as a whole.

We present in this section a heuristic for solving (2.12) approximately. Our method is based on relaxing the original problem (2.12) into an *easier* convex program which can be solved efficiently and globally.

2.1.3 SDP Relaxation of the MAP Problem

In this section, we describe a type of convex relaxation: SDP relaxation. In recent years, it has been at the center of some very exciting developments in the area of signal processing and communications, and it has shown great significance and relevance on a variety of applications. Roughly speaking, SDP relaxation is a powerful, computationally efficient approximation technique for a host of very difficult optimization problems [53].

We now derive the SDP relaxation for the MAP problem. The relevant notations are listed as follows for reference:

- $\mathbb{S}^N = \{\mathbf{A} \in \mathbb{R}^{N \times N} \mid \mathbf{A} = \mathbf{A}^T\}$ denotes the set of symmetric matrices.

- $\mathbf{A} \succeq \mathbf{B}$ means matrix $\mathbf{A} - \mathbf{B}$ is positive semidefinite with $\mathbf{A}, \mathbf{B} \in \mathbb{S}^N$.

- $\mathbf{A} \bullet \mathbf{B} = \sum_{i=1}^{N} \sum_{j=1}^{N} \mathbf{A}_{ij} \mathbf{B}_{ij}$ is the inner product of $\mathbf{A}, \mathbf{B} \in \mathbb{S}^N$.

Before the SDP relaxation, we first raise an equivalent problem to (2.12). We introduce

$\mathbf{X} = \mathbf{x}\mathbf{x}^T$, $\mathbf{S} = \mathbf{s}\mathbf{s}^T$ and rewrite (2.12) as follows:

$$
\begin{aligned}
\underset{\mathbf{x},\mathbf{X},\mathbf{s},\mathbf{S}}{\text{minimize}} \quad & \mathbf{Q} \bullet \mathbf{X} + \mathbf{q}^T\mathbf{x} + \mathbf{b}^T\mathbf{s} \\
\text{subject to} \quad & \mathbf{z}_d = \mathbf{H}_d\mathbf{s} \\
& \mathbf{A}_c\mathbf{x} = \mathbf{0} \\
& \mathbf{S}_{ii} - \mathbf{s}_i = 0 \quad i = 1, 2, \ldots, N \\
& \begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & \mathbf{X} \end{pmatrix} = \begin{pmatrix} 1 \\ \mathbf{x} \end{pmatrix} \cdot \begin{pmatrix} 1 \\ \mathbf{x} \end{pmatrix}^T \\
& \begin{pmatrix} 1 & \mathbf{s}^T \\ \mathbf{s} & \mathbf{S} \end{pmatrix} = \begin{pmatrix} 1 \\ \mathbf{s} \end{pmatrix} \cdot \begin{pmatrix} 1 \\ \mathbf{s} \end{pmatrix}^T
\end{aligned}
\tag{2.13}
$$

Reformulation can be done based on the following theorem [53].

Theorem 1 $\begin{pmatrix} 1 & \mathbf{s}^T \\ \mathbf{s} & \mathbf{S} \end{pmatrix} \geq \mathbf{0}$ and **rank** $\mathbf{S} = 1 \Leftrightarrow \mathbf{S} = \mathbf{s}\mathbf{s}^T$.

Notice that the constraint **rank** $\mathbf{S} = 1$ is non-convex, by removing it we have the SDP relaxation as follows.

$$
\begin{aligned}
\underset{\mathbf{x},\mathbf{X},\mathbf{s},\mathbf{S}}{\text{minimize}} \quad & \mathbf{Q} \bullet \mathbf{X} + \mathbf{q}^T\mathbf{x} + \mathbf{b}^T\mathbf{s} \\
\text{subject to} \quad & \mathbf{z}_d = \mathbf{H}_d\mathbf{s} \\
& \mathbf{A}_c\mathbf{x} = \mathbf{0} \\
& \mathbf{S}_{ii} - \mathbf{s}_i = 0 \quad i = 1, 2, \ldots, N \\
& \begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & \mathbf{X} \end{pmatrix} \geq \mathbf{0} \\
& \begin{pmatrix} 1 & \mathbf{s}^T \\ \mathbf{s} & \mathbf{S} \end{pmatrix} \geq \mathbf{0}
\end{aligned}
\tag{2.14}
$$

(2.14) is a standard form of SDP which can be solved efficiently using interior-point algorithms in various solver packages [54], [55]. Note that the known Linear Programming (LP) relaxation

technique can also be applied here, however, SDP relaxation meets our requirement because it is superior to LP relaxation in terms of approximation quality. In fact, a small error in topology identification of power grid can mislead the system operations significantly. Detail derivation can be found in [56]. The comparison of LP and SDP relaxation is summarized in [57].

The optimal objective value of LP, SDP relaxation and original integer program are represented by $v(\mathbf{LP})$, $v(\mathbf{SDP})$, $v(\mathbf{IP})$ respectively.

Theorem 2 *The SDP relaxation gives a tighter lower bound than LP relaxation with respect to the original interger program such as: $v(\mathbf{LP}) \leq v(\mathbf{SDP}) \leq v(\mathbf{IP})$.*

Another point needs attention is that the result of the SDP relaxation problem (2.14) offers a *soft decision* on the topology pattern, i.e., a continues value between 0 and 1 for each circuit breaker. Hence, we need to recover the *exact* topology pattern from this soft decision. This problem is investigated in the next section.

2.1.4 Combining Sparsity with SDP Relaxation

Inspired by the compressive sensing idea [58], we shall add the ℓ_1-norm term in the SDP relaxation in order to further improve the accuracy of our topology identification approach. Recall that we assume every breaker has a much higher probability staying in one state than the other. We consider the state with higher probability to be common state, and the other one is uncommon state for each breaker. It implies that in each topology identification process, the solved topology pattern should not change too much from the expected "*common*" topology pattern that every breaker is in its common state. This property enables us to employ ℓ_1-norm regularization to achieve better performance. We explain this idea in a formal way as follows. We first add the term $\lambda \|\mathbf{s} - \mathbf{s}_u\|_1$ into (2.14) with the constrains unchanged (in (2.15)). $\mathbf{s}_u$ is the vector of common status of the breakers. $\|\mathbf{s} - \mathbf{s}_u\|_1$ is the regularization function used to incorporate the rarity of event that circuit breakers change from common status to uncommon one. λ is the regularization parameter, which controls the sparsity of the status change vector $\mathbf{s} - \mathbf{s}_u$. The above idea is characterized in the following optimization problem:

$$\underset{\mathbf{x},\mathbf{X},\mathbf{s},\mathbf{S}}{\text{minimize}} \quad \mathbf{Q} \bullet \mathbf{X} + \mathbf{q}^T \mathbf{x} + \mathbf{b}^T \mathbf{s} + \lambda \left\| \mathbf{s} - \mathbf{s}_u \right\|_1$$

$$\text{subject to} \quad \mathbf{z}_d = \mathbf{H}_d \mathbf{s}$$

$$\mathbf{A}_c \mathbf{x} = \mathbf{0}$$

$$\mathbf{S}_{ii} - \mathbf{s}_i = 0 \quad i = 1, 2, \ldots, N \tag{2.15}$$

$$\begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & \mathbf{X} \end{pmatrix} \succeq \mathbf{0}$$

$$\begin{pmatrix} 1 & \mathbf{s}^T \\ \mathbf{s} & \mathbf{S} \end{pmatrix} \succeq \mathbf{0}$$

(2.15) is not a standard SDP due to the non-linearity of $\left\| \mathbf{s} - \mathbf{s}_u \right\|_1$. However, we can convert it into a standard SDP with the introduction of extra variable vector $\mathbf{t}$. $\left\| \mathbf{s} - \mathbf{s}_u \right\|_1$ is replaced by $\mathbf{1}^T \mathbf{t}$ and two more constraints are added in (2.15): $\mathbf{diag}(\mathbf{t} - \mathbf{s} + \mathbf{s_u}) \succeq \mathbf{0}$ and $\mathbf{diag}(\mathbf{t} + \mathbf{s} - \mathbf{s_u}) \succeq \mathbf{0}$.

Numerical Tests   In this section, the novel topology identification approach is tested on IEEE 14-bus system in Fig. 2.3. The network model parameters and system state are obtained using MATPOWER [59]. The measurements are from PMUs and circuit breaker monitors. They are assumed to be independent and the analog PMU measurement error is modeled as zero Gaussian with standard deviation $\sigma = 0.05$. We consider three scenarios that 1/3, 1/2 and 2/3 number of all states and breaker status are randomly selected as our measurement pool. The probability of uncommon status for each circuit breaker is set to be $0.05, 0.1, 0.3, 0.5$. 50 different topologies are simulated with each topology having a certain number of uncommon breaker status according to different prior distributions. Our proposed method in is solved by CVX, an optimization package targeted for solving convex optimization problems [55]. The evaluation performance with different prior distributions and measurement ratios is illustrated in Table. 2.1.

The simulation results in Table 2.1 are compliance with our expectation. Notice that when the probability of uncommon status occurring is equal to 0.5, it is the situation that we have no prior information on the breaker status, which is equivalent to the case that only the measurement
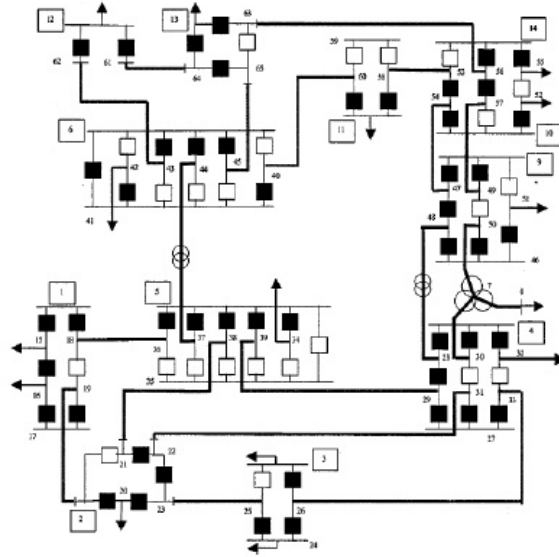
Figure (2.3) The IEEE 14-bus system modeled at the substation level [2]. Solid (hollow) squares indicate closed (open) circuit breakers, and thick (thin) lines correspond to transmission lines (breaker connections).

Table (2.1) Average Success Rate for Topology Identification

Prob of uncommon status	Ratio = $\frac{1}{3}$	Ratio = $\frac{1}{2}$	Ratio = $\frac{2}{3}$
0.05	96%	98%	98%
0.1	91%	92%	95%
0.3	76%	82%	90%
0.5	63%	71%	80%

data is available for topology inference. However, incorporating prior information in our designed framework is shown to be able to improve the accuracy of topology identification. Studying the proposed scheme for practical power network dimensions is currently under investigation.

Summary   A novel circuit breaker identification scheme was proposed in this section. By exploiting the prior distribution on breaker status which can potentially enhance the identification performance, we formulated the topology detection problem as a MAP estimation problem. To efficiently solve the optimization problem thus obtained, a SDP relaxed problem was further proposed. Leveraging the premises of compressive sampling, the aforementioned objective was regularized by the ℓ_1-norm of selected vectors to further account for prior information on breakers. Numerical tests on the IEEE 14-bus model verified the effectiveness of the novel scheme.

2.2 Distributed Power-Line Outage Detection

A key aspect of situational awareness in the power grid is the knowledge of transmission line status. Lessons learned from the 2003 northeastern blackout in United States reveal that accurate line monitoring in real-time is required throughout the whole power grid [60]. Fortunately, the development of real-time synchronized PMUs enables the direct usage of PMU-provided measurements to detect events within the power grid. At present, PMU-based line outage detection has been considered as a promising approach to facilitate effective fault identification.

2.2.1 Related Work

Existing PMU-based line outage detection methods typically use the internal-external network model for the whole interconnected system in which the goal is to identify external line outages using only measurements within the internal system [11], [12], [13], [14], [15]. Specifically, [13] formulates line outage detection as a best match problem, which contains an exhaustive searching process for the most likely outage line. Thus, it can only handle the single-line outage scenario. Building upon the work of [13], double-line outage detection is considered in [14], while it restricts to the case with exactly a double-line outage in the system. A similar exhaustive search is also applied in [14], but the searching space is much larger than that of the single-line case, which, thus, is very computationally expensive. Another method for line outage identification employs a Gauss–Markov graphical model of the power network and is capable of dealing with multiple outages at a moderate complexity [16] despite requiring a grid-wise measurement. An alternative sparse overcomplete representation-based algorithm was proposed in [15], which can also handle multiple line outages. Then, Chen *et al.* developed a global stochastic optimization technique based on cross-entropy optimization [17]. The algorithm in [17] does not require prior knowledge of the parameters used in [15], whose selection can significantly affect solution accuracy. Wu *et al.* proposed an ambiguity group-based location recognition algorithm, which claims to be faster and shows higher accuracy than the algorithm in [15] for multiple line outage detection [18]. Banerjee *et al.* exploited the fact that the line outage is persistent and studied the problem of

line outage detection and identification in the framework of the theory of quickest change detection [19]. However, the aforementioned methods all carry out the processing in a centralized manner, which is vulnerable in practice. Further, these existing approaches need to transmit raw data in the system and, thus, may raise privacy issues. Huge recent interest in research and applications fall into distributed methods for diagnosing faults in complex distributed systems. In [20], a distributed fault detection method was devised for rail vehicle suspension systems in which the observers are co-operated mainly by the state estimation errors. A hidden Markov random field-based distributed fault detection algorithm was invented for wireless sensor networks [21].

In this section, we aim at proposing a scheme to detect power-line outage in a distributed manner. The proposed scheme relies on a wide area measurement system (WAMS), which can be seen as a network of sensors that cooperatively measure the status of the grid. The proposed scheme is expected to work based on WAMS as follows. First, the raw measurements from different PMUs are collected in the corresponding phasor data concentrators (PDCs) for processing; second, the line outage detection is performed among the PDCs in a distributed fashion; finally, the results after detection (instead of the raw data) are transmitted to the WAMS center, which provides critical information to the system operators.
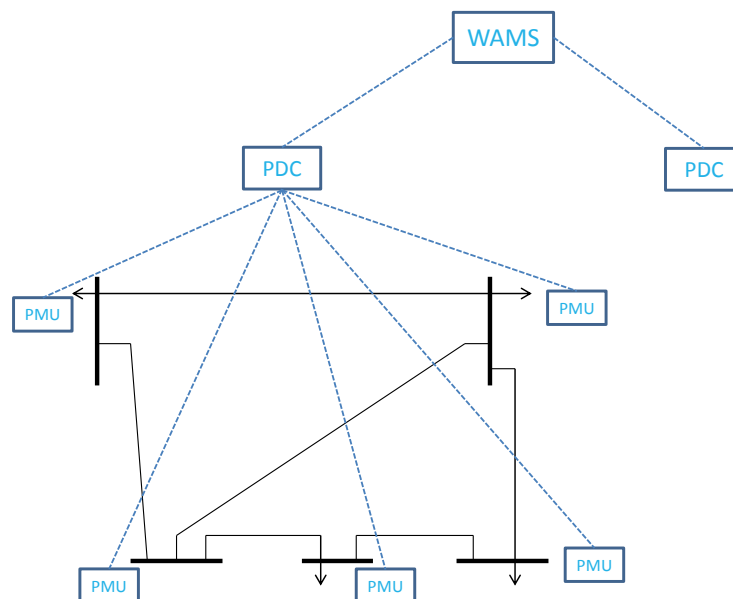
Figure (2.4) Hierarchical architecture of a wide area measurement system (WAMS) in a smart grid.

Specifications for the Proposed Framework Our main idea is to devise a distributed and robust protocol that can be performed in WAMS for smart grid monitoring application. In this section, the assumptions and problem settings in the proposed method will be described.

Sensor Network Model Our proposed method is based on the hierarchical network of WAMS (as shown in Figure 2.4), which consists of a hierarchical structure, as follows. In each area, a certain number of PMUs are installed in the bus substations of the power grid. In the middle level, there is a set of phasor data concentrators (PDCs). Each PDC can share information with the PDCs in neighborhoods. In the top level, there is a WAMS center, which collects information from PDCs supporting the system-wide monitoring task. As a result, we can naturally see that in each area with a PDC, it is a local control area or sub-system [61].

Sensor Measurement Settings We consider a linear physical equation describing the relation between the measurable quantity and the set of unknown variables. The set of unknown variables is related to the sensor reading through the measurement matrix. In this paper, the branch currents are considered as the unknown quantities, and the measurements we use are bus voltage phasors and all the branch-current phasors that are incident to the bus if a PMU is installed in the bus substation. Our algorithm recognizes faulty/normal lines by determining whether their linear physical measurement equations are valid or not. Furthermore, an additional assumptions is made:

- For our purpose of detecting possible faulty lines, the number of measurements we have is relatively smaller than the number of unknown variables, which implies that the measurement matrix is under-determined.

2.2.2 Problem Formulation

In this section, we describe the detailed measurement equation and centralized line outage detection solution adopted in this paper. The proposed novel algorithm will be built upon them.

PMU Measurement Equation In a typical power transmission system, the synchrophasor measurements at the **n**-th PDC area, expressed in rectangular coordinates, are collected in a vector

$\bar{\mathbf{y}}_\mathbf{n}$, and they satisfy the following linear model:

$$\bar{\mathbf{y}}_\mathbf{n} = \bar{\mathbf{H}}_\mathbf{n}\mathbf{x} + \bar{\mathbf{g}}_\mathbf{n} \tag{2.16}$$

where $\mathbf{x}$ is the unknown vector to be estimated containing all branch currents, $\bar{\mathbf{H}}_\mathbf{n} \in \mathbb{R}^{M_n \times 2N_l}$ is the measurement matrix, M_n is the number of measurements within the $\mathbf{n}$-th PDC area, N_l is the number of transmission lines in the whole system and $\bar{\mathbf{g}}_\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Lambda}_\mathbf{n})$ denotes the additive Gaussian noise vector. For notational convenience, we multiply with $\mathbf{\Lambda}_\mathbf{n}^{-1/2}$ on both sides of Equation (2.16) to yield:

$$\mathbf{y}_\mathbf{n} = \mathbf{H}_\mathbf{n}\mathbf{x} + \mathbf{g}_\mathbf{n} \tag{2.17}$$

where $\mathbf{y}_\mathbf{n} = \mathbf{\Lambda}_\mathbf{n}^{-1/2}\bar{\mathbf{y}}_\mathbf{n}$, and the other terms are manipulated similarly. Using Equation (2.17), the weighted least squares form:

$$\left\|\mathbf{\Lambda}_\mathbf{n}^{-1/2}(\bar{\mathbf{y}}_\mathbf{n} - \bar{\mathbf{H}}_\mathbf{n}\mathbf{x})\right\|_2^2$$

is replaced by the regular least squares $\|\mathbf{y}_\mathbf{n} - \mathbf{H}_\mathbf{n}\mathbf{x}\|_2^2$. We will use this notation in the following sections.

Now, we first introduce some basic concepts on electrical circuits:

- Kirchhoff's current law: at any node (junction) in an electrical circuit, the sum of currents flowing into that node is equal to the sum of currents flowing out of that node.

- Kirchhoff's voltage law: the sum of all voltage drops and rises in a closed loop equals zero.

The laws above are two approximate equalities that deal with the current and voltage difference in electrical circuits [62].

Let $\mathbf{v} = Re(\mathbf{v}) + Im(\mathbf{v})$ be the $N_b \times 1$ vector of complex nodal voltages with N_b the number of buses in the system. By writing down the node equations of Kirchhoff's current law (KCL) and Kirchhoff's voltage law (KVL) at each node, we can derive the vector of complex currents injected on each line as follows:

$$\mathbf{i_{fl}} = \mathbf{x} = \mathbf{Y_{fl}v} \tag{2.18}$$

where $\mathbf{Y_{fl}}$ describes the line-to-bus admittance matrix. The matrices $\bar{\mathbf{H}}_\mathbf{n}$ in Equation (2.16) can be expressed as:

$$\bar{\mathbf{H}}_\mathbf{n} = \begin{pmatrix} \mathbf{Q_n}Re(\mathbf{Y_{fl}^{-1}}) & -\mathbf{Q_n}Im(\mathbf{Y_{fl}^{-1}}) \\ \mathbf{Q_n}Im(\mathbf{Y_{fl}^{-1}}) & \mathbf{Q_n}Re(\mathbf{Y_{fl}^{-1}}) \\ \mathbf{e_n^T} & \mathbf{0^T} \\ \mathbf{0^T} & \mathbf{e_n^T} \end{pmatrix} \tag{2.19}$$

where $\mathbf{Q_n}$ is the selection matrix according to the $\mathbf{n}$-th PDC.

At this point, our problem is equivalent to using a distributed method to determine whether the linear model in Equation (2.16) is valid. A conventional and straightforward way to solve this problem would be:

(1) In each PDC area, estimate the unknown variables locally.

(2) Communicate and share the estimates with other PDCs.

(3) Perform a fusion of estimates in each PDC.

(4) Apply a likelihood ratio test to detect faulty lines.

This above method will work well when there are sufficient measurements (more than the number of unknown variables) available in each PDC [63]. However, in some scenarios, for example in the smart grid system that we focus on in this paper, fetching sufficiently-sized measurements may be infeasible or costly. Consequently, a framework that can make accurate decisions with fewer data sets will be of practical importance. From the next section, we are going to describe our solution for this purpose.

2.2.3 Possible Centralized Solution for Line Outage Detection

In this paper, we combine the measurements and the prior information on the branch currents to do the line outage detection. We consider a Bayesian framework, where the branch current variables are random vectors with Gaussian distribution $\mathcal{N}(\mathbf{x_p}, \mathbf{\Lambda_p})$. We assume that, in practice, the mean vector $\mathbf{x_p}$ and covariance matrix $\mathbf{\Lambda_p}$ can be estimated from historical data [64656465]. The variables are assumed to be independent, and thus, the covariance matrix $\mathbf{\Lambda_p}$ is diagonal. Inspired by the idea of compressive sensing, we can have a sparse solution for a certain under-determined system by adding the ℓ_1-norm regularization [58]. Since most of the components of the item in the ℓ_1-norm term is pushed into zero, we make the unknown vector $\mathbf{x}$ to compare with its nominal model in the ℓ_1-norm term in order to create "sparse" faulty branches. Now, suppose that there are k transmission line outages in the system. Then, the maximum likelihood (ML) estimation in a single control center can be formulated as:

$$
\begin{aligned}
&\underset{\mathbf{x}}{\text{minimize}} && \frac{1}{2} \|\mathbf{y} - \mathbf{Hx}\|_2^2 \\
&\text{subject to:} && \left\| \mathbf{\Lambda_p}^{-1/2}(\mathbf{x} - \mathbf{x_p}) \right\|_0 = k
\end{aligned}
\tag{2.20}
$$

where $\mathbf{x}$ is the unknown vector of the system defined in Equation (2.16). $\mathbf{y}$ denotes the measurements collected in the single center, and $\mathbf{H}$ is the corresponding measurement matrix of the system. It contains the global topology and impedance information. Note that $\|\cdot\|_p$ means p-norm. Here, the faulty lines can be identified by non-zero components in the vector $\mathbf{x} - \mathbf{x_p}$. Based on the optimization theory [54], there exists a λ that makes the following equation equivalent to the problem formulation (2.20):

$$
\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{y} - \mathbf{Hx}\|_2^2 + \lambda \left\| \mathbf{\Lambda_p}^{-1/2}(\mathbf{x} - \mathbf{x_p}) \right\|_0
\tag{2.21}
$$

where $\lambda > 0$ is an application-dependent pre-defined parameter. It quantizes the tradeoff of effects between the two objectives in Equation (2.21). The selection of λ will be discussed in a later section.

Both Equations (2.20) and (2.21) are non-convex, which means it is hard to solve them exactly in a reasonable time. We employ the ℓ_1-norm approximation in [58] to replace the zero-norm term

in Equation (2.21), which leads to the convex optimization problem shown below:

$$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \lambda \left\|\mathbf{\Lambda_p}^{-1/2}(\mathbf{x} - \mathbf{x_p})\right\|_1 \tag{2.22}$$

Remark 1 *The centralized grid-wise measurement data collection the computation in implementing Equation (2.22) are inefficient due to bandwidth and time constraints or infeasible because of data privacy concerns; thus, distributed computations are strongly preferred or demanded.*

2.2.4   Distributed Line Outage Detection

In this section, we striveto solve the optimization problem in Equation (2.22) in a distributed manner. Note that if we decompose Equation (2.22) into *N* PDC areas, then Equation (2.22) can be expressed in the following:

$$\min_{\mathbf{x_n}} \sum_{n=1}^{N} f_n(\mathbf{x_n}) \tag{2.23}$$

in which function $f_n(\mathbf{x_n})$ denotes the "cost function" for each PDC, and it is given by:

$$f_n(\mathbf{x_n}) = \frac{1}{2}\|\mathbf{y_n} - \mathbf{H_n}\mathbf{x_n}\|_2^2 + \lambda \left\|\mathbf{\Lambda_{pn}}^{-1/2}(\mathbf{x_n} - \mathbf{x_{pn}})\right\|_1 \tag{2.24}$$

where $\mathbf{x_n}$, $\mathbf{H_n}$, $\mathbf{x_{pn}}$ and $\mathbf{\Lambda_{pn}}$ correspond to the unknown variables associated with the **n**-th PDC. Each PDC in the area can choose to minimize Equation (2.24) individually, but this method is clearly sub-optimal, since the overlapping variables are not taken into account.

Remark 2 *The criterion in Equation (2.24) will force some entries of the vector of branch currents ($\mathbf{x_n}$) equal to their mean values (corresponding entries of $\mathbf{x_{pn}}$), which implies that they are consistent with their statistical distribution, and thus, these branches are recognized as in the normal condition. On the other hand, if certain entries of the branch currents fail to be equal to their mean values, then the associated branches are considered to be possibly faulty or abnormal.*

Distributed Power-Line Change Detection Solution   Denote $\mathbf{x_n}$ as the sub-vector of $\mathbf{x}$, which contains the unknown variables involved in the **n**-th PDC. Furthermore, denote $\mathbf{x_{nm}}$ as the
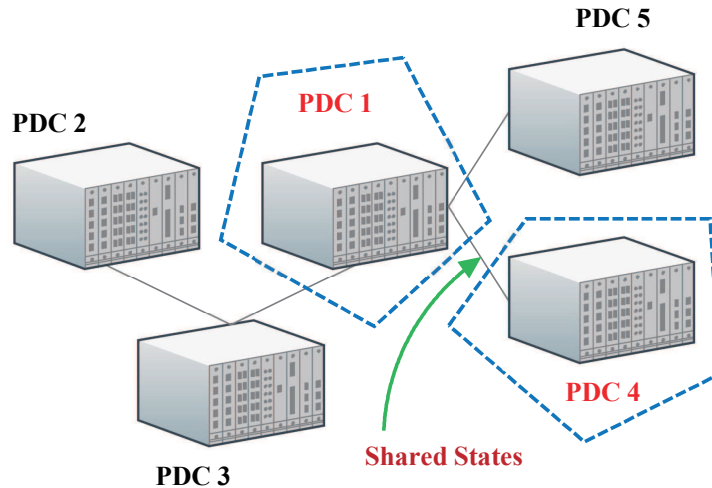
Figure (2.5) An example of a phasor data concentrator (PDC) network.

value of the shared variables between neighboring $\mathbf{n}$-th and $\mathbf{m}$-th PDC (a sub-vector of $\mathbf{x_n}$ or $\mathbf{x_m}$). Then, the estimate of overlapping unknown variables by neighboring PDCs should be same. Then, Equation (2.23) can be reformulated as:

$$\underset{\mathbf{x_n}}{\text{minimize}} \quad \sum_{\mathbf{n}=1}^{N} f_{\mathbf{n}}(\mathbf{x_n})$$
$$\text{subject to:} \quad \mathbf{x_{nm}} = \mathbf{x_{mn}}, \ \mathbf{m} \in \mathcal{N}_{\mathbf{n}}; \ \mathbf{n}, \mathbf{m} \in P \tag{2.25}$$

where $\mathcal{N}_{\mathbf{n}}$ is the set of neighboring PDCs of the $\mathbf{n}$-th PDC and P is the set of PDCs. For instance, in Figure 2.5, Node 1 and Node 4 share the edge (1,4). This means that these two PDC areas have overlapping unknown variables. As a result, Node 1's estimate of the branch current on (1,4) should be the same as Node 4's estimate on (1,4).

In this paper, the proposed formulation for line outage detection in Equation (2.25) is solved by resorting to the so-called ADMM. To briefly illustrate the general ADMM algorithm [66], consider the prototype problem:

$$\text{minimize} \quad f(x) + g(z)$$
$$\text{subject to:} \quad Ax + Bz = c \tag{2.26}$$

with variables $x \in \mathbf{R}^n$ and $z \in \mathbf{R}^m$, where $A \in \mathbf{R}^{p \times n}$, $B \in \mathbf{R}^{p \times m}$ and $c \in \mathbf{R}^p$. Functions f and g are assumed to be convex. As in the method of multipliers, the augmented Lagrangian can be formed:

$$L_\rho(x, z, y) = f(x) + g(z) + y^T(Ax + Bz - c) + (\rho/2) \|Ax + Bz - c\|_2^2 \,.$$

ADMM consists of the iterations:

$$x^{k+1} := \underset{x}{\operatorname{argmin}} \, L_\rho(x, z^k, y^k) \tag{2.27a}$$

$$z^{k+1} := \underset{z}{\operatorname{argmin}} \, L_\rho(x^{k+1}, z, y) \tag{2.27b}$$

$$y^{k+1} := y^k + \rho(Ax^{k+1} + Bz^{k+1} - c), \tag{2.27c}$$

where $\rho > 0$ is the predefined augmented Lagrangian parameter and y is the Lagrangian multiplier (dual variable) of the constraint in Equation (2.26). The ADMM algorithm is considered to have three steps: an x-minimization Equation (2.27a), a z-minimization step Equation (2.27b) and a dual variable update Equation (2.27c).

Let us now apply the method of ADMM in [66] to solve the line outage detection problem formulated in Equation (2.25) using a distributed mechanism. We introduce auxiliary variables $\vartheta_{\mathbf{nm}}$ and $\mathbf{z_n}$ in order to fit the ADMM framework. Then, Equation (2.25) can be alternatively expressed as:

$$\begin{aligned}
\underset{\mathbf{x_n}, \vartheta_{\mathbf{nm}}, \mathbf{z_n}}{\text{minimize}} \quad & \sum_{\mathbf{n}=1}^{N} f_{\mathbf{n}}(\mathbf{x_n}) \\
\text{subject to:} \quad & \mathbf{x_{nm}} = \vartheta_{\mathbf{nm}}, \; \mathbf{m} \in \mathcal{N}_{\mathbf{n}}; \; \mathbf{n}, \mathbf{m} \in P \\
& \mathbf{x_n} - \mathbf{x_{pn}} = \mathbf{z_n}
\end{aligned} \tag{2.28}$$

We also introduce variable $v_{\mathbf{nm}}$ to denote the Lagrangian multiplier for the first constraint in Equation (2.28) and $\mathbf{s_n}$ to denote the multiplier for the second constraint in Equation (2.28). Note that by using ADMM in our problem, there are three primal variables: $\mathbf{x_n}$, $\vartheta_{\mathbf{nm}}$ and $\mathbf{z_n}$; two dual

variable: $v_{\mathbf{nm}}$ and $\mathbf{s_n}$. The augmented Lagrangian function can be obtained as:

$$
L_\rho(\mathbf{x_n}, \vartheta_{\mathbf{nm}}, \mathbf{z_n}, v_{\mathbf{nm}}, \mathbf{s_n})
$$

$$
= \sum_{n=1}^{N} \{ f_n(\mathbf{x_n}) + \sum_{m \in \mathcal{N}_n} (v_{\mathbf{nm}}^T(\mathbf{x_{nm}} - \vartheta_{\mathbf{nm}})
$$

$$
+ (\rho/2) \|\mathbf{x_{nm}} - \vartheta_{\mathbf{nm}}\|_2^2) + \mathbf{s_n}^T(\mathbf{x_n} - \mathbf{x_{pn}} - \mathbf{z_n})
$$

$$
+ (\rho/2) \|\mathbf{x_n} - \mathbf{x_{pn}} - \mathbf{z_n}\|_2^2 \}
$$

(2.29)

Let k be the iteration index; then, the ADMM algorithm consists of the following update rules:

$$
\mathbf{x_n}^{k+1} = \underset{\mathbf{x_n}}{\operatorname{argmin}} \; L_\rho(\mathbf{x_n}, \vartheta_{\mathbf{nm}}^k, \mathbf{z_n}^k, v_{\mathbf{nm}}^k, \mathbf{s_n}^k) \tag{2.30a}
$$

$$
(\vartheta_{\mathbf{nm}}^{k+1}, \mathbf{z_n}^{k+1}) = \underset{\vartheta_{\mathbf{nm}}, \mathbf{z_n}}{\operatorname{argmin}} \; L_\rho(\mathbf{x_n}^{k+1}, \vartheta_{\mathbf{nm}}, \mathbf{z_n}, v_{\mathbf{nm}}^k, \mathbf{s_n}^k) \tag{2.30b}
$$

$$
v_{\mathbf{nm}}^{k+1} = v_{\mathbf{nm}}^k + \rho(\mathbf{x_{nm}}^{k+1} - \vartheta_{\mathbf{nm}}^{k+1}) \; \text{ for all } \mathbf{n}, \mathbf{m}. \tag{2.30c}
$$

$$
\mathbf{s_n}^{k+1} = \mathbf{s_n}^k + \rho(\mathbf{x_n}^{k+1} - \mathbf{x_{pn}} - \mathbf{z_n}^{k+1}) \tag{2.30d}
$$

To simplify the presentation, we combine the linear and quadratic terms in the augmented Lagrangian in Equation (2.29) that can be applied in Equations (2.30a) and (2.30b) by ignoring the terms independent of the decision variables:

$$
L_\rho(\mathbf{x_n}, \vartheta_{\mathbf{nm}}, \mathbf{z_n}, v_{\mathbf{nm}}^k, \mathbf{s_n}^k) =
$$

$$
\sum_{n=1}^{N} (f_n(\mathbf{x_n}) + \sum_{m \in \mathcal{N}_n} (\rho/2) \|\mathbf{x_{nm}} - \vartheta_{\mathbf{nm}} + (1/\rho)v_{\mathbf{nm}}^k\|_2^2 \tag{2.31}
$$

$$
+ (\rho/2) \|\mathbf{x_n} - \mathbf{x_{pn}} - \mathbf{z_n} + (1/\rho)\mathbf{s_n}^k\|_2^2)
$$

Now, we are concerned about how to implement the updates in Equations (2.30a)–(2.30d) efficiently. Since Equations (2.30c) and (2.30d) are simple linear updating equations, we only need to focus on the deduction of Equations (2.30a) and (2.30b). To solve Equation (2.30a), several algebraic manipulations are used to enable the simplification of the analysis. We define:

(1) $\mathbf{D_n}$ as a diagonal matrix with its $(\mathbf{m}, \mathbf{m})$-th entry being 1;

(2) $\mathbf{r}_\mathbf{n}^k = \vartheta_\mathbf{n}^k - (1/\rho)v_\mathbf{n}^k;$

(3) $\mathbf{I_n}$ denotes an identity matrix with its dimension being the number of states in $\mathbf{n}$-th area.

As a result, the term $\sum_{\mathbf{m} \in \mathcal{N}_\mathbf{n}} (\frac{\rho}{2}) \left\| \mathbf{x_{nm}} - \vartheta_\mathbf{nm} + (\frac{1}{\rho})v_\mathbf{nm}^k \right\|_2^2$ in Equation (2.30a) can be expressed as:
$(\rho/2) \left\| \mathbf{D_n}(\mathbf{x_n} - \mathbf{r}_\mathbf{n}^k) \right\|_2^2$. Then, after manipulating via matrix calculus, we obtain the minimizer of Equation (2.30a) as follows:

$$
\begin{aligned}
\mathbf{x}_\mathbf{n}^{k+1} = &\left(\mathbf{H}_\mathbf{n}^T \mathbf{H_n} + \rho \mathbf{D_n} + \rho \mathbf{I_n} \right)^{-1} \\
&\times \left(\mathbf{H}_\mathbf{n}^T \mathbf{y_n} + \rho(\mathbf{D_n}\mathbf{r}_\mathbf{n}^k + \mathbf{x_{pn}} + \mathbf{z}_\mathbf{n}^k - (1/\rho)\mathbf{s}_\mathbf{n}^k) \right)
\end{aligned}
\tag{2.32}
$$

Regarding solving Equation (2.30b), it is known that the optimality conditions satisfy when the zero vector belongs to subdifferentials of Equation (2.30b) with respect to variable $\vartheta_\mathbf{nm}$ and $\mathbf{z_n}$ [67]. We first consider the minimization with $\vartheta_\mathbf{nm}$; the following Theorem is derived in order to conclude the updates of $\vartheta_\mathbf{nm}$.

Theorem 3 *For each pair of* $\mathbf{n}, \mathbf{m}$ *in Equation* (2.30c)*, the following holds for the updating Lagrange multipliers:* $v_\mathbf{nm}^k + v_\mathbf{mn}^k = 0$

Proof 1 *In Equation* (2.30b)*, we note that the optimization task will be performed in* $\mathbf{n}$-*th and* $\mathbf{m}$-*th PDC in parallel for each adjacent pair* $(\mathbf{n}, \mathbf{m})$*. Thus, we can obtain the following result by solving Equation* (2.30b) *for* $(\mathbf{n}, \mathbf{m})$ *and* $(\mathbf{m}, \mathbf{n})$*, respectively:*

$$
\begin{aligned}
\vartheta_\mathbf{nm}^{k+1} &= \mathbf{x}_\mathbf{nm}^{k+1} + (1/\rho)v_\mathbf{nm}^k \\
\vartheta_\mathbf{mn}^{k+1} &= \mathbf{x}_\mathbf{mn}^{k+1} + (1/\rho)v_\mathbf{mn}^k
\end{aligned}
\tag{2.33}
$$

where $\vartheta_\mathbf{nm}$ *and* $\vartheta_\mathbf{mn}$ *are the same variable; then, averaging the both sides of the two equations in Equation* (2.33) *implies:*

$$
\vartheta_\mathbf{nm}^{k+1} = \left(\frac{\mathbf{x}_\mathbf{nm}^{k+1} + \mathbf{x}_\mathbf{mn}^{k+1}}{2} \right) + \left(\frac{v_\mathbf{nm}^k + v_\mathbf{mn}^k}{2\rho} \right)
\tag{2.34}
$$

In a similar manner, we can express ϑ_{nm}^{k+1} and ϑ_{mn}^{k+1} by using Equation (2.30c). The calculations are:

$$\vartheta_{nm}^{k+1} = \mathbf{x}_{nm}^{k+1} + (1/\rho)v_{nm}^k - (1/\rho)v_{nm}^{k+1}$$
$$\vartheta_{mn}^{k+1} = \mathbf{x}_{mn}^{k+1} + (1/\rho)v_{mn}^k - (1/\rho)v_{mn}^{k+1}$$

(2.35)

Finally, averaging both sides of Equation (2.35) yields:

$$
\begin{aligned}
\vartheta_{nm}^{k+1} &= \left(\frac{\vartheta_{nm}^{k+1} + \vartheta_{mn}^{k+1}}{2}\right) \\
&= \left(\frac{\mathbf{x}_{nm}^{k+1} + \mathbf{x}_{mn}^{k+1}}{2}\right) + \left(\frac{v_{nm}^k + v_{mn}^k}{2\rho}\right) \\
&- \left(\frac{v_{nm}^{k+1} + v_{mn}^{k+1}}{2\rho}\right)
\end{aligned}
$$

(2.36)

By comparing the right side of Equations (2.34) and (2.36), we find that the only different part is the last item in Equation (2.36), which turns out to be zero. Theorem 3 is then proven.

At this point, it is clear to see that by using Theorem 3, Equation (2.34) can be reduced to:

$$\vartheta_{nm}^{k+1} = \frac{(\mathbf{x}_{nm}^{k+1} + \mathbf{x}_{mn}^{k+1})}{2}$$

(2.37)

Next, we are concerned about how to address the updates of $\mathbf{z_n}$. Note that due to the ℓ_1-norm term, Equation (2.30b) is not differentiable everywhere, but sub-differentiable with respect to $\mathbf{z_n}$ [67]. As mentioned previously, we take the sub-differential over Equation (2.30b) with respect to $\mathbf{z_n}$ and the optimality condition becomes:

$$\mathbf{0} \in \partial\lambda\left\|\mathbf{\Lambda}_{pn}^{-1/2}\mathbf{z_n}\right\|_1 + \rho(\mathbf{z_n} - (\mathbf{x}_n^{k+1} - \mathbf{x_{pn}} + (1/\rho)\mathbf{s}_n^k))$$

By using the soft thresholding operator defined in [66], for instance, the i-th component $\mathbf{z}_n^{k+1}[i]$ (scalar) is updated as:

$$\mathbf{z}_n^{k+1}[i] = S_{(\lambda/\rho)\mathbf{\Lambda}_{pn}^{-1/2}[i][i]}(\mathbf{x}_n^{k+1}[i] - \mathbf{x_{pn}}[i] + (1/\rho)\mathbf{s}_n^k[i])$$

In a similar way, a closed-form solution for the updates of $\mathbf{z_n}$ is obtained as follows:

$$\mathbf{z}_\mathbf{n}^{k+1} = S_{(\lambda/\rho)\mathbf{\Lambda}_{\mathbf{pn}}^{-1/2}}(\mathbf{x}_\mathbf{n}^{k+1} - \mathbf{x}_{\mathbf{pn}} + (1/\rho)\mathbf{s}_\mathbf{n}^k) \tag{2.38}$$

where:

$$S_b(a) = \begin{cases} a - b, & a > b; \\ 0, & |a| \le b; \\ a + b, & a < -b. \end{cases} \tag{2.39}$$

Note, here, component-wise updating is applied, such that the i-th component of $\mathbf{z}_\mathbf{n}$ is updated according to the i-th entry of the rest of the vectors in Equation (2.38) and the (i, i)-th entry of the diagonal matrix $\mathbf{\Lambda}_{\mathbf{pn}}^{-1/2}$.

Now, the ADMM updating in Equations (2.30a)–(2.30d) for each processor can be summarized in Algorithm 1.

Algorithm 1 Distributed line change detection (D-LCD).

1: **Input:** $\mathbf{y}_\mathbf{n}, \mathbf{H}_\mathbf{n}, \mathbf{\Lambda}_\mathbf{n}, \mathbf{\Lambda}_{\mathbf{pn}}, \mathbf{x}_{\mathbf{pn}}, \mathbf{D}_\mathbf{n}, \lambda > 0, \rho > 0, k = 0$.
2: Initialize: $\mathbf{x}_\mathbf{n}, \vartheta_{\mathbf{nm}}, \mathbf{z}_\mathbf{n}, \nu_{\mathbf{nm}}, \mathbf{s}_\mathbf{n}$.
3: **while** not converged or stopping criterion not reached **do**
4:     $k \leftarrow k + 1$.
5:     Update $\mathbf{x}_\mathbf{n}^{k+1}$ based on Equation (2.32).
6:     Exchange $\mathbf{x}_{\mathbf{nm}}^{k+1}$ with its neighbors.
7:     Update $\vartheta_{\mathbf{nm}}^{k+1}, \mathbf{z}_\mathbf{n}^{k+1}$ via Equations (2.37) and (2.38), respectively.
8:     Update $\nu_{\mathbf{nm}}^{k+1}$ and $\mathbf{s}_\mathbf{n}^{k+1}$ through Equations (2.30c) and (2.30d).

2.2.5 Distributed Line Change Detection with Warm Start

The most computational-intensive step in Algorithm 1 is the update of $\mathbf{x}_\mathbf{n}$ given in Equation (2.32), which, in essence, requires matrix inversion and multiplication for each PDC in every iteration. Nevertheless, a detailed look shows that the variables in Equation (2.32) may not change significantly within two consecutive iterations. The previous ADMM iteration $\mathbf{x}_\mathbf{n}^k$ often provides a good approximation to the results, which can be used as a warm start to update $\mathbf{x}_\mathbf{n}^k$. The warm start process reduces the complexity in computing $\mathbf{x}_\mathbf{n}^{k+1}$, since the computation starts from a more appropriate initialization instead of from zero (or some other fixed and default initialization) [66].

Now, if we look at the the minimization step in Equation (2.30a) along with its minimizer in Equation (2.32), it actually can be regarded as solving a system of linear equations:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{2.40}$$

The least squares solution of Equation (2.40) is [68]:

$$\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} \tag{2.41}$$

We observe that Equation (2.32) is equivalent to finding the least squares solution with matrix $\mathbf{A}$ and vector $\mathbf{b}$ formed in the following:

$$\mathbf{A} = \begin{pmatrix} \mathbf{H_n} \\ \sqrt{\rho}\mathbf{I_n} \\ \sqrt{\rho}\mathbf{D_n} \end{pmatrix} \tag{2.42}$$

$$\mathbf{b} = \begin{pmatrix} \mathbf{y_n} \\ \sqrt{\rho}(\mathbf{x_{pn}} + \mathbf{z_n^k} - \frac{1}{\rho}\mathbf{s_n^k}) \\ \sqrt{\rho}\mathbf{r_n^k} \end{pmatrix} \tag{2.43}$$

At this point, we have changed the problem of $\mathbf{x_n}$-update in Equation (2.32) into finding a method to solve linear equations in Equation (2.40) with $\mathbf{A}$ and $\mathbf{b}$ defined in Equations (2.42) and (2.43), respectively. To this end, we adopt the LSQR algorithm in this paper. Recall that $\mathbf{I_n}$, $\mathbf{D_n}$ are diagonal matrices and that $\mathbf{H_n}$ is sparse in general. Thus, matrix $\mathbf{A}$ is also sparse. LSQR thus fits our need, since it is very efficient for solving sparse linear equations [69]. Interested readers, please refer to [69] for the details. We omit its introduction, here due to space limitation. In short, the modified distributed line detection algorithm with a warm start is described in Algorithm 4.

Selection of the Tuning Parameter   In our proposed centralized and distributed algorithms stated in Equation (2.22) and Algorithm 1, we have to choose the parameter λ first. As discussed

---

Algorithm 2 D-LCD with a warm start.

1: **Input:** $\mathbf{y_n}, \mathbf{H_n}, \mathbf{\Lambda_n}, \mathbf{\Lambda_{pn}}, \mathbf{x_{pn}}, \mathbf{D_n}, \lambda > 0, \rho > 0, k = 0.$
2: Initialize: $\mathbf{x_n}, \vartheta_{\mathbf{nm}}, \mathbf{z_n}, \nu_{\mathbf{nm}}, \mathbf{s_n}.$
3: **while** not converged or stopping criterion not reached **do**
4:     Assemble $\mathbf{A}$ and $\mathbf{b}$ according to (2.42) and (2.43).
5:     Solve linear equations $\mathbf{Ax} = \mathbf{b}$ using LSQR procedure with initial value $\mathbf{x_n}^k$.
6:     $k \leftarrow k + 1.$
7:     Update $\mathbf{x_n}^{k+1}$ based on the solution in Step 5.
8:     Exchange $\mathbf{x_{nm}}^{k+1}$ with its neighbors.
9:     Update $\vartheta_{\mathbf{nm}}^{k+1}, \mathbf{z_n}^{k+1}$ via (2.37) and (2.38), respectively.
10:    Update $\nu_{\mathbf{nm}}^{k+1}$ and $\mathbf{s_n}^{k+1}$ through (2.30c) and (2.30d).
---

in Section 4.2, ℓ_1-norm term in Equation (2.22) will force the item in the norm to be sparse, and λ determines the importance of this objective. If λ is very large, most of the components in the ℓ_1-norm would be zeros. In other words, the tuning parameter λ specifies the sparsity level of the solution. In addition, the selection of λ depends on the specific application we are working on. Thanks to the help of the cross-validation technique, we can have some portion of data for model validation. The optimized λ is then derived in terms of prediction accuracy. By using the "one-standard-error" rule, one can also have the largest value of λ, such that the error is within one standard-error of the minimum [70].

Numerical Tests To evaluate the proposed centralized and distributed line change detection algorithms, we use an Intel Duo Core at 1.8 GHz (1.5 GB RAM) computer with MATLAB for numerical testing. The branch current phasors and the PMU measurements are obtained from MATPOWER [59]. To solve the centralized algorithm in Equation (2.22), we used CVX, a package for specifying and solving convex optimization problems [55]. The PMU measurement noise is simulated as an independent zero-mean Gaussian with its covariance matrix $\mathbf{\Lambda_n} = 0.002\mathbf{I_n}$. The covariance matrix $\mathbf{\Lambda_p}$ is set to $0.003\mathbf{I_p}$, where $\mathbf{I_p}$ is an identity matrix with the same dimension as the unknown vector.

WSCC Nine-Bus Test Case In this section, the WSCC nine-bus test case system was used for our simulation. The diagram of the system is demonstrated in Figure 2.6. There are three

generators (G1,G2,G3), three transformers (T1,T2,T3) and nine lines in which the line parameter

information is listed in Table 2.2.

Table (2.2) Line parameters of the WSCC nine-bus system.

Line	Resistance (p.u)	Reactance (p.u)
1–4	0.0000	0.0576
4–5	0.0170	0.0920
5–6	0.0390	0.1700
3–6	0.0000	0.0586
6–7	0.0119	0.1008
7–8	0.0085	0.0720
8–2	0.0000	0.0625
8–9	0.0320	0.1610
9–4	0.0100	0.0850
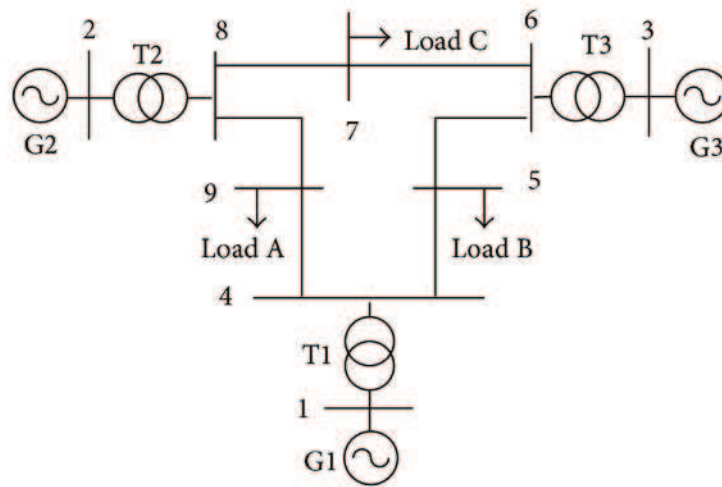
Figure (2.6) WSCC nine-bus test case system.

From Table 2.2, the line-to-bus admittance matrix $\mathbf{Y_{fl}}$ can be formed, which is used for con-

structing the measurement matrix $\mathbf{H}$ in Equation (2.22). In this case, the size of the unknown vector

is nine by one, and we place two PMUs at Bus 4 and Bus 6 with their line current measurements

in (1–4), (4–5), (9–4), (5–6), (3–6), (6–7). The system is assumed to be at steady state before and

after the line change. We made the line change on the reactance of line (1–4), which was altered
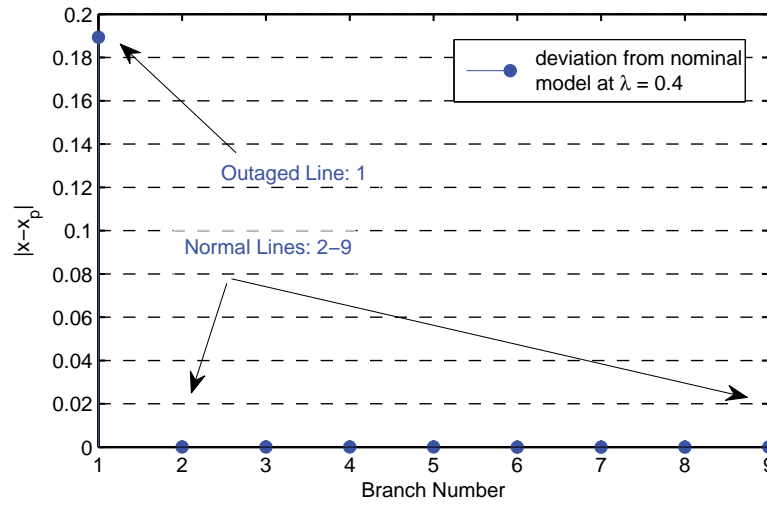
Figure (2.7) Centralized line outage detection.
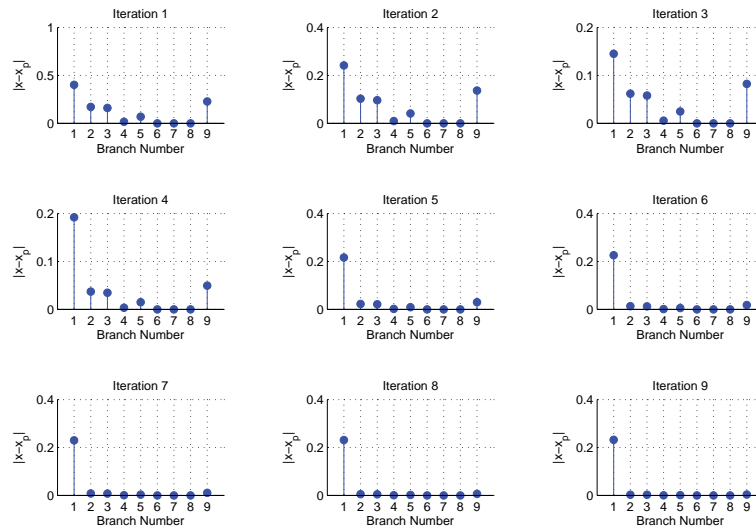
Figure (2.8) Distributed line outage detection.

Line Outage Detection
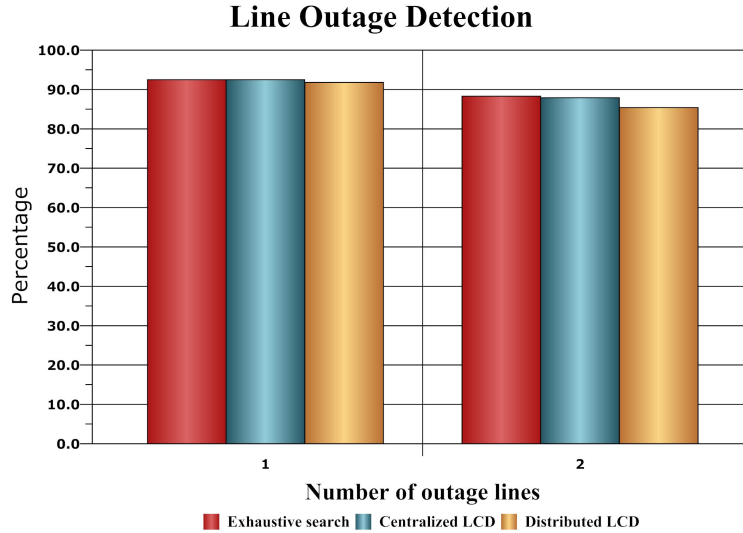
Figure (2.9) Comparison of detection performance for the IEEE 118-bus system.

from 0.0576 to infinity. Then, we ran a DC power flow in MATPOWER to obtain the branch currents in normal conditions and the measurements after change. The above are all of the quantities considered as the input to our centralized line change detection algorithm. The result in Figure 2.7 shows that the faulty line (1–4) has been correctly detected by the algorithm. Note that here, λ from $0.35 - -0.45$ can guarantee the accurate decision in this case.

We also tested our D-LCD algorithm on this nine-bus system, and the results of first nine ADMM iterations are captured in Figure 2.8. Note that initially, Branches 1–3, 5 and 9 have positive values, which means that they are all seen as a group of possible faulty lines. During Iteration 2–4, the values of Branches 1–3, 5 and 9 are actually decreasing, while an interesting point is that the decreasing speed of Branches 2, 3, 5 and 9 is much faster than Branch 1's. This observation is conformed with the theory part discussed previously, that the most likely set of branches should survive for the next iteration. From Iteration 5, Branch 1 is almost the only one standing out. This implies that Branch 1 is considered to be faulty by our distributed line change detection algorithm. In other words, the distributed algorithm almost converges to the centralized version result (we assume it as a benchmark) in Figure 2.7 in just five iterations.

IEEE 118-Bus System   The IEEE 118-bus system is tested here for evaluating our algo-rithms in the case of a large network. There are 186 branches in the test system, which will result in over 17, 200 possible faulty topologies in just a double-line outage scenario. All of the single line outage possibilities and 300 double-line outage cases are randomly chosen for testing. We adopt the method in [71] as our pool of measurements and randomly select two thirds the number of measurements from it. The exhaustive search algorithm in [13141314] is compared with our proposed methods in Figure 2.9 in terms of the percentage of the correctly detected outage pattern.

Note that the exhaustive search scheme is considered as the benchmark here since it is "op-timal" in the statistical sense. It is impressive that both the centralized and distributed line outage detection methods perform very close to this optimal criterion.

IEEE 300-Bus System   The running times of the developed algorithms are also tested on the IEEE 300-bus test system. Following a Monte Carlo simulation method, the results for single and double-line outages are listed in Table 2.3.

In both the single- and double-line outage cases, D-LCD and D-LCD with warm start outper-form the rest of algorithms, which is within expectations. It is found that as the system size and the number of line outages increases, the advantage of the warm-started D-LCD over distributed LCD becomes more sharper in terms of computational time. However, the exhaustive search approach does not scale well, as its running time jumps up in an order much higher than the others.

Table (2.3) Running time comparison for the IEEE 300-bus system.

Algorithm	Single-Line Outage	Double-Line Outage
Exhaustive Search	0.50 s	28 s
Centralized LCD	0.37 s	0.95 s
Distributed LCD	0.12 s	0.31 s
Warm-started D-LCD	0.053 s	0.14 s

Discussion   In this paper, the proposed algorithms are assumed to work in transmission net-works. Nevertheless, theoretically, they can also apply to distribution networks. The current dis-

tribution networks usually lack measurements and have a low level of monitoring capabilities. As smart grids develop, the proposed algorithms have the potential to work in distribution networks once the the infrastructure of "smart" sensor networks has been deployed.

Our proposed distributed algorithms involve the communication of neighboring PDCs. Each PDC only communicates with its neighbors by its estimates of the shared unknown variables. Hence, if the PDC is unable to collect the neighbors's information, it will keep its value of estimates unchanged. In this paper, the proposed distributed algorithms are more robust than the centralized ones in the following sense: for the centralized algorithms, if the sole processing center is attacked or fails, all of the information will be lost, and the system cannot obtain a solution for the outage detection function. However, for the proposed distributed algorithms, the probability of having similar serious conditions is much smaller.

Summary A novel distributed line outage detection algorithm was developed based on WAMS, which has been an important component of smart grids. The proposed approach allows multiple line outage identification using limited PMU measurements. The feature of low-complexity distributed processing in the proposed framework can enhance the efficiency, security and privacy level in smart grid monitoring. Numerical tests demonstrated the merits of the proposed schemes in coordinating the discovery of multiple line outages in a power grid.

2.3 Microgrid Reconfiguration Strategy

As an important content of smart grid, the concept of smart microgrid has been proposed recently. Unlike traditional power grid, microgrid has three important features. First, it owns a small-scale group of generation resources and local customers itself. Second, microgrid is able to manage its energy system in a cooperative fashion based on two-way communications and bidirectional electricity paths. Third, microgrid is a promising way to integrate distributed generations (DGs) (i.e., renewable energy resources (RES)) on the local community in order to diversify energy supply and improve efficiency [72].

Microgrids belong to an area where the cyber and physical worlds meet. It is an application

of the Cyber-Physical System (CPS) in which sensing, networking, and computing are tightly coupled with the control of the physical power grid. In the cyber-layer, one of the challenges is distributed resource management. Fortunately, the advances in information infrastructure such as "smart sensors" provide opportunities to better cope with this issue [73]. Based on the collected information on loads, generators and transmission lines, etc., system operator can have the potential of continuous decision-making and monitoring over the grid.

2.3.1 Related Work

At present, a lot of research efforts have been conducted on conventional distribution system and microgrid reconfiguration. In [22], [23], [24], [25], minimizing the active power loss (or total power loss) was considered as the sole objective. Multi-objective formulation of the reconfiguration problem was investigated in [26]. Besides the objective of minimization of the total power loss, [26] also considers minimizing both the nodes voltage deviation and the violation of line current limit. Consequently, it is a more advanced and realistic formulation comparing to the single-objective ones. However, in case of an emergency, power accessibility is a more important issue than economical reasons. Thus, the main goal in such scenarios should be to serve as much critical total loads as possible in the system. Recent work of reconfiguration for quick service restoration in distribution systems or microgrids were discussed in [27], [28], [29]. However, like all the aforementioned frameworks, they are not suitable for microgrids since the presence of RES is not considered, which is an important feature of future smart microgrids. Considering all the issues above, we are thus motivated to design a new microgrid restoration framework.

2.3.2 System Architecture and Problem Statement

System Architecture Fig. 2.10 depicts the system architecture for the microgrid reconfiguration. The houses can be residential households or other consumers (i.e., hospitals). The green arrows are bidirectional energy paths between two neighboring units. There are conventional DERs and renewable energy for the power supply. More importantly, microgrid "controllers" (e.g. various smart sensors) are installed in the physical units across the grid. For example, the microgrid
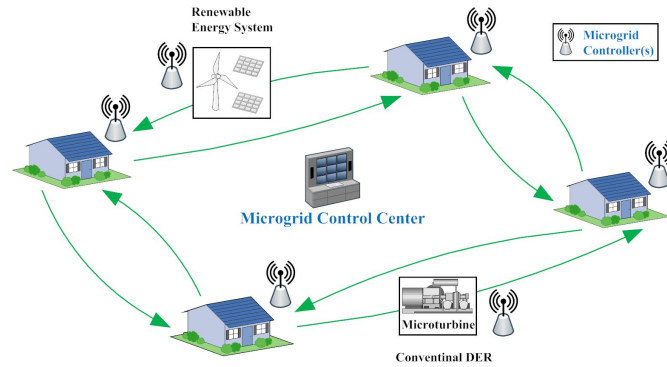
Figure (2.10) An example microgrid CPS required for self-reconfiguration.

control center can adjust the wind turbine based on the wind profile (captured by its controller) to achieve higher efficiency. In short, the control center can perform grid resource management (such as grid reconfiguration, load balancing) based on the timely information provided by this cyber system.

Problem Description   We consider a microgrid as a graph in this paper. The vertices represent the buses in the power grid. Each bus has load and generation units (i.e., conventional distributed generation units and renewable energy generators). The set of edges denotes the branches in the microgrid. It is assumed that all branches have circuit breakers. The load in each bus is also assumed to have circuit breaker attached modeling the possible action of load shedding. The loads in the system have different priorities. This means that some loads (i.e., medical services) are more critical than others (i.e., residential households). Our reconfiguration scheme is expected to guarantee these critical loads served first. The reconfiguration is modeled as a decision problem simultaneously maximizing the loads served and the operation reliability. Relevant notations are listed in table 2.4.

Objectives in the Problem Formulation   We first describe the two different objectives adopted in the presenting paper.

Total Loads Served   We consider the microgrid reconfiguration for service restoration. It is usually taken place after the fault occurrence. Hence an important goal is to restore the power

Table (2.4) Notations

P_{G_i}	conventional DG units power generation on i-th bus.
P_{R_i}	RES power generation on i-th bus (random variable).
P_{L_i}	actual power consumption on i-th bus.
$P_{t_{ij}}$	power flow between i-th bus and j-th bus.
S_{l_i}	circuit breaker status at the load of i-th bus.
λ_{M_i}	weighting factor for the load of i-th bus.
N	the set of buses in the microgrid.
B	the set of branches in the microgrid.
$\mathcal{N}_i$	the set of neighboring buses to i-th bus.
c_{ij}	the power flow limit in branch (i, j).
$P_{G_i}^{min}$	minimum amount of DG units power generation on i-th bus.
$P_{G_i}^{max}$	maximum amount of DG units power generation on i-th bus.
$P_{L_i}^{min}$	minimum required actual power consumption on i-th bus.
$P_{L_i}^{max}$	maximum actual power consumption on i-th bus.

supply service to the loads in the system [25], [28]. Specifically, the objective of total loads served can be expressed as:

$$\sum_{i=1}^{|N|} \lambda_{M_i} S_{l_i} P_{L_i} \tag{2.44}$$

where the parameter λ_{M_i} accounts for the priority of load i. Specifically, in objective (2.44), the set of λ_{M_i} helps put the loads with high priority to be supplied first.

Reliability of Reconfiguration Operation   In the reconfiguration process, it is reasonable and meaningful to take the operating reliability into account since it is highly required to avoid service interruption again in a short period. We define the reliability of reconfiguration operation using the mathematical expression as follows.

$$\min_{(i,j)\in B}\left\{c_{ij} - \left|P_{t_{ij}}\right|\right\} \tag{2.45}$$

Equation (2.45) is to first compute the distance of actual power flow to its limit for each branch, and find the minimum distance among them. This measure is important since operating power flow close to its limit will greatly increase the probability of failure and be more vulnerable to contingencies. A simple example in Fig. 2.11 shows cases with different values of operation
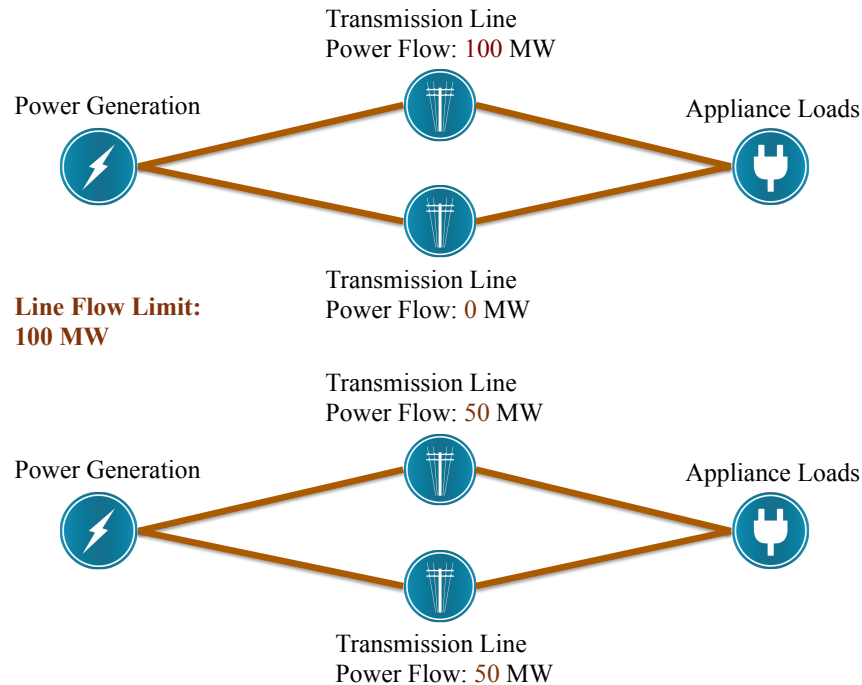
Figure (2.11) Illustration of measuring the reliability of reconfiguration operation under two different cases. The power generation source needs to supply 100 MW to the appliance loads and there are two transmission lines between them. The upper case solution has reliability value 0 MW while the bottom one has reliability value 50 MW based on the definition in (2.45). It is stated that the upper one is more vulnerable than the bottom one since one of the lines in upper case is reaching the line flow limit.

reliability.

Uncertainty in Renewable Energy Generation Another distinct assumption made in our reconfiguration problem is that renewable energy generators are involved in supplying power to the grid. They are considered to be widely used and installed in each bus. Unfortunately, the energy output of renewable generators (denoted as random variables P_{R_i} in this paper) will fluctuate around its forecasted values due to e.g., fast-varying weather conditions [7], [8]. Thus, at some time, the power supply available at some buses might not satisfy their power required. As a result, it is essential to perform a grid reconfiguration that is capable of limiting this kind of risk.

Microgrid Reconfiguration Formulation Based on the system model described above, we formulate the microgrid reconfiguration as the optimization problem (P1) as follows.

$$\text{(P1)} \max_{\mathcal{V}} \left[\sum_{i=1}^{|N|} \lambda_{M_i} S_{l_i} P_{L_i} \, , \, \min_{(i,j) \in B} \left\{ c_{ij} - \left| P_{t_{ij}} \right| \right\} \right]^T \tag{2.46a}$$

$$\text{s.t.} \quad -c_{ij} \leq P_{t_{ij}} \leq c_{ij}, \; (i,j) \in B \tag{2.46b}$$

$$-S_{t_{ij}} \times c_{ij} \leq P_{t_{ij}} \leq S_{t_{ij}} \times c_{ij}, \; (i,j) \in B \tag{2.46c}$$

$$P_{G_i}^{min} \leq P_{G_i} \leq P_{G_i}^{max}, \; \forall i \in N \tag{2.46d}$$

$$S_{l_i} \times P_{L_i}^{min} \leq P_{L_i} \leq S_{l_i} \times P_{L_i}^{max}, \; \forall i \in N \tag{2.46e}$$

$$S_{l_i} \in \{0, 1\}, \; \forall i \in N \tag{2.46f}$$

$$S_{t_{ij}} \in \{0, 1\}, \; (i,j) \in B \tag{2.46g}$$

$$\Pr \left(P_{G_i} + P_{R_i} - \sum_{j \in N_i} P_{t_{ij}} \geq P_{L_i}, \forall i \in N \right) \geq 1 - \epsilon \tag{2.46h}$$

where $\mathcal{V} := \{P_{G_i}, P_{L_i}, P_{t_{ij}}, S_{l_i}, S_{t_{ij}}\}$ contains all the decision variables. Specifically, the objective function (2.46a) takes accounts of two different aspects, namely, the total power served and the index of operational reliability; Constraint (2.46b) illustrates that the power flow along the lines should conform with their line limits; Constraint (2.46c) ensures that the power will flow through the bus pair (i, j) only when the breaker on this branch is closed; Constraint (2.46d) represents the generator capacity in each bus; Constraint (2.46e) models the possible action of load curtailment by the system operator. Furthermore, constraints (2.46f) and (2.46g) represent the binary state of circuit breakers in the buses (for the loads) and the branches respectively; Finally, constraint (2.46h) requires that the actual power supply at each bus satisfies its demand with probability no less than $1 - \epsilon$, where ϵ is a pre-selected threshold.

Unfortunately, solving (P1) is computationally difficult for the following reasons:

r1: The bilinear term $S_{l_i} P_{L_i}$ in (2.46a) renders (P1) nonconvex, which implies that (P1) is difficult to solve optimally and efficiently in general;

r2: Due to the binary variables $\{S_{l_i}, S_{t_{ij}}\}$, solving (P1) is NP-hard;

r3: The probabilistic constraint (2.46h) is generally in a computationally intractable form.

We propose an approach to cope with **r1**-**r3** in the ensuing section with a computationally tractable reformulation of (P1).

2.3.3 Computationally Tractable Reformulation

We first derive an alternative problem formulation of (P1) that has linear objectives as follows.

$$(\text{P2}) \quad \underset{\mathcal{V}'}{\text{minimize}} \qquad \begin{bmatrix} t_1 , & t_2 \end{bmatrix}^T \tag{2.47a}$$

$$\text{subject to:} \quad (2.46\text{b}) \text{ - } (2.46\text{h}) \text{ and} \tag{2.47b}$$

$$\sum_{i=1}^{|N|} \lambda_{M_i} S_{l_i} P_{L_i} \geq -t_1 \tag{2.47c}$$

$$-(c_{ij} + t_2) \leq P_{t_{ij}} \leq (c_{ij} + t_2), \ (i, j) \in B \tag{2.47d}$$

where $\mathcal{V}' := \{P_{G_i}, P_{L_i}, P_{t_{ij}}, S_{l_i}, S_{t_{ij}}, t_1, t_2\}$ collects the reconfiguration decision variables.

A closer look into (P1) and (P2) reveals the following theorem.

Theorem 4 *The problems in (P1) and (P2) are equivalent.*

Proof 2 *In the first place, the objective in* (2.46a) *is equivalent to the following:*

$$min \ \begin{bmatrix} -\sum_{i=1}^{|N|} \lambda_{M_i} S_{l_i} P_{L_i} , & -\min_{(i,j) \in B}\left\{c_{ij} - \left|P_{t_{ij}}\right|\right\} \end{bmatrix}^T$$

Second, we start with the trick of introducing two auxiliary variables t_1 and t_2 that serve as the upper bounds on the two objectives respectively. Thus, the above objective can be cast as (2.47a) *with two extra constraints as follows:*

$$-\sum_{i=1}^{|N|} \lambda_{M_i} S_{l_i} P_{L_i} \ \leq \ t_1$$

$$-\min_{(i,j) \in B}\left\{c_{ij} - \left|P_{t_{ij}}\right|\right\} \ \leq \ t_2$$

Here, the first constraint is the same as (2.47c). *The second constraint means that the minimum value of* $c_{ij} - |P_{t_{ij}}|$ *among all branches* $(i, j) \in B$ *is at least* $-t_2$. *In other words, every branch* $(i, j) \in$ *B satisfies the condition that* $c_{ij} - |P_{t_{ij}}|$ *is greater or equal to* $-t_2$, *that is* $c_{ij} - |P_{t_{ij}}| \geq -t_2,\ (i, j) \in B$. *It can become* (2.47d) *after several steps of algebraic manipulations.*

Note that the vector optimization problem (P2) can be decomposed into two single-objective problems. The first single-objective problem in (P2) has objective function t_1 and the second problem has objective function t_2, which are linear objectives. Besides, the two problems have the same constraints. Thus, in this section, which is dealing with difficult constraints arising from **r1**-**r3** of the optimization program, all the involving implicit objective functions are linear.

Convex Relaxation of Bilinear Inequalities (for r1)   Bilinear inequalities such as (2.47c) can be categoized into the form of so-called *Bilinear Matrix Inequalities* (BMIs). Problems involving BMIs have been a focus in mathematical programming as wells as the robust control theory. Unfortunately, BMIs are known to be nonconvex constraints. They are computationally complex and hard to be solve [74]. The BMI Feasibility Problem is even shown to be NP-hard [75]. To relax this problem, one way is to search for a convex set which includes BMIs and closely approximates the BMIs. Replacing BMIs by such a set, the optimization problem becomes convex and therefore tractable [76].

To address **r1**, we adopt the convex relaxation method in [74] to approximate the non-convex set specified by the constraint in (2.47c).

We first define a new variable $Q_i = P_{L_i} S_{l_i},\ \forall i \in N$. Then (2.47c) is equivalent to the following:

$$\sum_{i=1}^{|N|} \lambda_{M_i} Q_i \geq -t_1 \tag{2.48a}$$

$$Q_i = P_{L_i} S_{l_i},\ \forall i \in N \tag{2.48b}$$

By applying the convex relaxation advance in [74], (2.48b) is replaced by:

$$Q_i \leq P_{L_i}\overline{S}_{l_i} + \underline{P_{L_i}}S_{l_i} - \underline{P_{L_i}}\overline{S}_{l_i}, \ \forall i \in N \tag{2.49a}$$

$$Q_i \leq P_{L_i}\underline{S}_{l_i} + \overline{P_{L_i}}S_{l_i} - \overline{P_{L_i}}\underline{S}_{l_i}, \ \forall i \in N \tag{2.49b}$$

$$Q_i \geq P_{L_i}\underline{S}_{l_i} + \underline{P_{L_i}}S_{l_i} - \underline{P_{L_i}}\,\underline{S}_{l_i}, \ \forall i \in N \tag{2.49c}$$

$$Q_i \geq P_{L_i}\overline{S}_{l_i} + \overline{P_{L_i}}S_{l_i} - \overline{P_{L_i}}\,\overline{S}_{l_i}, \ \forall i \in N \tag{2.49d}$$

where $\underline{P_{L_i}}, \overline{P_{L_i}}$ are the minimum and maximum values of variable P_{L_i} respectively. Similarly, $\underline{S}_{l_i}, \overline{S}_{l_i}$ are the minimum and maximum values of variable S_{l_i}, which are known constants. In our problem, according to the formulation in previous, $\underline{P_{L_i}} = 0$, $\overline{P_{L_i}} = P_{L_i}^{max}$, $\underline{S}_{l_i} = 0$, $\overline{S}_{l_i} = 1$. Putting those facts into constraints (2.49a)-(2.49d) yields:

$$Q_i \leq P_{L_i}, \ \forall i \in N \tag{2.50a}$$

$$Q_i \leq P_{L_i}^{max}S_{l_i}, \ \forall i \in N \tag{2.50b}$$

$$Q_i \geq 0, \ \forall i \in N \tag{2.50c}$$

$$Q_i \geq P_{L_i} + P_{L_i}^{max}(S_{l_i} - 1), \ \forall i \in N \tag{2.50d}$$

Now, the constraint in (2.47c) containing bilinear term is replaced by (2.48a) and (2.50a)-(2.50d). Note that (2.48a) and (2.50a)-(2.50d) are linear constraints with respect to decision variables $\{Q_i, P_{L_i}, S_{l_i}\}$.

SDP Relaxation of 0-1 Integer Constraint (for r2)   Regarding the issues in **r2**, we use the semidefinite relaxation technique in [53] to obtain a convex-relaxed version of binary variable constraints in (2.46f) and (2.46g). In the following, we will present the detailed derivation of the aforementioned conversion.

Background on SDP Relaxations   In general, SDP relaxation is a subfield of convex relaxation techniques. Particularly, it has been applied to many difficult problems in combinatorial optimization, signal processing, and control theory. SDP relaxation is proven to provide a very

tight bound for several classes of nonconvex problems [53].

To briefly illustrate the SDP relaxation technique, we consider the binary quadratic problem as follows.

$$\text{minimize} \quad \mathbf{s}^T \mathbf{W} \mathbf{s}$$
$$\text{subject to} \quad \mathbf{s} = \{0, 1\}^n \tag{2.51}$$

where matrix $\mathbf{W}$ is semidefinite. The 0-1 quadratic program is a well-known difficult optimization problem since it is shown to be equivalent to Max-Cut [77], which is in the class of NP-hard problems (many of its problem instances would be intractable).

The cost function in (2.51) can be rewritten as:

$$\mathbf{s}^T \mathbf{W} \mathbf{s} = \langle \mathbf{W}, \mathbf{s}\mathbf{s}^T \rangle \tag{2.52}$$

where $\langle \cdot \rangle$ indicates inner product operation. Following the idea of linearization, we introduce a matrix variable $\mathbf{S}$ to take the role of $\mathbf{s}\mathbf{s}^T$. The binary constraint in (2.51) is equivalent to:

$$s_i^2 = s_i, \ \forall i \in N \tag{2.53}$$

It implies that the main diagonal of matrix $\mathbf{S}$ is equal to $\mathbf{s}$. Then the binary constraint can be expressed as follows:

$$\text{diag}(\mathbf{S}) = \mathbf{s} \tag{2.54a}$$
$$\mathbf{S} = \mathbf{s}\mathbf{s}^T \tag{2.54b}$$

Further, it can be proved that (2.54b) is equivalent to the following [53]:

$$\mathbf{S} \succeq \mathbf{0}, \ \text{rank}(\mathbf{S}) = 1 \tag{2.55}$$

Here, $\mathbf{A} \succeq \mathbf{B}$ is meant that matrix $\mathbf{A} - \mathbf{B}$ is positive semidefinite. The first constraint in (2.55) is linear with respect to variable $\mathbf{S}$ while the rank constraint is not. In fact, the difficulty in **r2** arises

from the nonconvexity of the rank constraint. By dropping it, we obtain the following relaxed version of the original quadratic program:

$$
\begin{aligned}
\underset{\mathbf{s}, \mathbf{S}}{\text{minimize}} \quad & \langle \mathbf{W}, \mathbf{S} \rangle \\
\text{subject to} \quad & \text{diag}(\mathbf{S}) = \mathbf{s} \\
& \mathbf{S} \geq \mathbf{0}
\end{aligned}
\tag{2.56}
$$

Relaxation for r2   Now, we go back to our problem. Define $\mathbf{s} = \left[S_{l_1}, \cdots, S_{l_i}, \cdots, S_{l_{|N|}} \right]^T$ as a $|N| \times 1$ vector and introduce a new matrix $\mathbf{S}^L = \mathbf{s}\mathbf{s}^T$. Then the binary constraint in (2.46f) can be relaxed to:

$$
\begin{aligned}
& \mathbf{S}_{ii}^L - \mathbf{s}_i = 0, \ \ \forall i \in N \\
& \mathbf{S}^L \geq \mathbf{0}
\end{aligned}
\tag{2.57}
$$

Constraint (2.46g) can be addressed in a similar manner. We order the transmission lines and define vector $\mathbf{t} = \left[t^1, \cdots, t^i, \cdots, t^{|B|} \right]^T$, where t^i is the circuit breaker's status on i-th transmission line, and $|B|$ is the number of transmission lines in the system. Let matrix variable $\mathbf{S}^t = \mathbf{t}\mathbf{t}^T$. Then constraint (2.46g) can be replaced by:

$$
\begin{aligned}
& \mathbf{S}_{ii}^t - \mathbf{t}_i = 0, \ \ i = 1, 2, \ldots, |B| \\
& \mathbf{S}^t \geq \mathbf{0}
\end{aligned}
\tag{2.58}
$$

Notice that (2.46f) and (2.46g) are replaced by (2.57) and (2.58) respectively which contain only linear and semidefinite cone constraints. These constraints are known to be convex and are conformed with the formulation of SDP [54]. For now, the only issue left is **r3** (since we have approximated the constraints in **r1, r2** with convex ones). As long as we have a tight approximation of probability constraint in (2.46d) by using linear constraint, the original problem formulation (P1) can be relaxed into a SDP [1]. In addition, it is known that several off-the-shelf efficient interior point methods can be used to solve SDPs [78].

---

[1]Please note that all the other constraints are linear.

Table (2.5) Matrix-wise Definitions

$\mathbf{P^G}$	$|N|$-dimensional vector where P_{G_i} is the i-th element.
$\mathbf{P^L}$	$|N|$-dimensional vector where P_{L_i} is the i-th element.
$\mathbf{P^t}$	$|B|$-dimensional vector collecting power flows on all lines.
$\mathbf{A}$	$|N| \times |B|$ oriented incidence matrix of the grid graph.
$\mathbf{c}$	$|B|$-dimensional vector collects power flow limits on all lines.
$\mathbf{C}$	$|B| \times |B|$ matrix equals diag($\mathbf{c}$).
$\mathbf{P_G^{min}}$	$|N|$-dimensional vector where $P_{G_i}^{min}$ is the i-th element.
$\mathbf{P_G^{max}}$	$|N|$-dimensional vector where $P_{G_i}^{max}$ is the i-th element.
$\mathbf{P_L^{min}}$	$|N|$-dimensional vector where $P_{L_i}^{min}$ is the i-th element.
$\mathbf{P_L^{max}}$	$|N|$-dimensional vector where $P_{L_i}^{max}$ is the i-th element.
$\mathbf{P_{L_{min}}}$	$|N| \times |N|$ matrix equals diag($\mathbf{P_L^{min}}$).
$\mathbf{P_{L_{max}}}$	$|N| \times |N|$ matrix equals diag($\mathbf{P_L^{max}}$).
$\mathbf{\lambda_M}$	$|N|$-dimensional vector where λ_{M_i} is the i-th element.
$\mathbf{Q}$	$|N|$-dimensional vector where Q_i is the i-th entry.
$\mathbf{P_R}^k$	$|N|$-dimensional vector where $P_{R_i}^k$ is the i-th entry.

Approximation of Probability Constraint (for r3) We aim to deal with **r3** in the next step, based on the so-called scenario-based convex approximation [79]. In the first place, (2.46h) can be rewritten as:

$$\Pr\left\{ P_{L_i} - P_{G_i} - P_{R_i} + \sum_{j \in \mathcal{N}_i} P_{t_{ij}} \leq 0, \forall i \in N \right\} \geq 1 - \epsilon \tag{2.59}$$

To briefly describe the general scenario-based convex approximation method, consider the prototype probability-constrained problem:

$$\begin{aligned} &\underset{\lambda \in \Lambda}{\text{minimize}} \quad c^T \lambda \\ &\text{subject to} \quad \Pr\left\{ \phi \in \Phi : f(\lambda, \phi) \leq 0 \right\} \geq 1 - \epsilon \end{aligned} \tag{2.60}$$

Here λ is the "design parameter" and ϕ denotes the "uncertainty factor" which is a random variable. To be specific, in our problem (in (2.59)), the set of design parameters $\lambda = \left\{ P_{G_i}, P_{L_i}, P_{t_{ij}} \right\}$, uncertainty factor $\phi = \{P_{R_i}\}$. In this case, notice that $f(\lambda, \phi) : \Lambda \times \Phi \to \mathbb{R}^n$ is convex in λ, for any fixed value of $\phi \in \Phi$. Thus, the Assumption 1 in [79] is met. Then (2.60) can be approximated by

the scenario-based approximation method as follows.

$$\underset{\lambda \in \Lambda}{\text{minimize}} \quad c^T \lambda$$

$$\text{subject to} \quad f(\lambda, \phi^k) \leq 0, \ k = 1, \ldots, M \tag{2.61}$$

where $\phi^1, \phi^2, \ldots, \phi^M$ are M independently generated samples of ϕ. To apply the scenario-based convex approximation method, we first independently generate M samples $P_{R_i}^1, P_{R_i}^2, \ldots, P_{R_i}^M$, and replace the chance-constraint (2.59) with the linear constraints as follows.

$$P_{L_i} - P_{G_i} - P_{R_i}^k + \sum_{j \in \mathcal{N}_i} P_{t_{ij}} \leq 0, \ k = 1, \ldots, M, \ \forall i \in N \tag{2.62}$$

Notice that we select the constraints $f(\lambda, \phi^k) \leq 0$ in a random manner, thus the optimal solution $\hat{\lambda}$ depending on the multi-sample extraction $(\phi^1, \phi^2, \ldots, \phi^M)$ is actually a random variable. Therefore, $\hat{\lambda}$ can be a ϵ-level solution for a given random extraction and not for another. Let parameter β bounds the probability that $\hat{\lambda}$ is not a ϵ-level solution (feasible for problem in (2.60)). Thus, β can be seen as the *risk of failure* associated to the randomized solution algorithm. It is said that if M (specified by the following condition) random scenarios are drawn, the optimal solution in the approximated problem achieves ϵ-level feasibility for the original probability-constrained one with probability no less than $1 - \beta$ [79].

$$M \geq \left\lceil \frac{2}{\epsilon} \ln \frac{1}{\beta} + 2n_\lambda + \frac{2n_\lambda}{\epsilon} \ln \frac{2}{\epsilon} \right\rceil \tag{2.63}$$

where n_λ is the number of design variables, and $\lceil \cdot \rceil$ denotes the ceil function. Tailoring (2.63) to our problem, the minimum sample size $\tilde{M}$ can be provided based on the theorem as follows.

Theorem 5 *Given the power imbalance probability threshold ϵ, and the lower bound*

$$M \geq \tilde{M} := \left\lceil \frac{2}{\epsilon} \ln \frac{1}{\beta} + 2(2|N| + |B|) + \frac{2(2|N| + |B|)}{\epsilon} \ln \frac{2}{\epsilon} \right\rceil$$

then the solution to the reformulated problem with constraint (2.62) *is feasible for the original*

problem with constraint (2.59)*, with probability at least as* $1 - \beta$.

At this point, a convex-relaxed program of (P2) can be obtained. For simplification, we have some new definitions in table 2.5.

Now we can express our computationally tractable reformulation as follows.

$$\text{(P3)} \quad \underset{\mathcal{D}}{\text{minimize}} \quad \left[\, t_1 \,,\, t_2 \,\right]^T \tag{2.64a}$$

$$\text{subject to:} \quad \text{(2.57) - (2.58) and} \tag{2.64b}$$

$$-\mathbf{c} \leq \mathbf{P^t} \leq \mathbf{c} \tag{2.64c}$$

$$-\mathbf{Ct} \leq \mathbf{P^t} \leq \mathbf{Ct} \tag{2.64d}$$

$$\mathbf{P_G^{min}} \leq \mathbf{P^G} \leq \mathbf{P_G^{max}} \tag{2.64e}$$

$$\mathbf{P_{L_{min}}s} \leq \mathbf{P^L} \leq \mathbf{P_{L_{max}}s} \tag{2.64f}$$

$$\lambda_\mathbf{M}^T \mathbf{Q} \geq -t_1 \tag{2.64g}$$

$$\mathbf{0} \leq \mathbf{Q} \leq \mathbf{P^L} \tag{2.64h}$$

$$\mathbf{P^L} + \mathbf{P_{L_{max}}s} - \mathbf{P_L^{max}} \leq \mathbf{Q} \leq \mathbf{P_{L_{max}}s} \tag{2.64i}$$

$$-\mathbf{c} - t_2 \mathbf{1}^T \leq \mathbf{P^t} \leq \mathbf{c} + t_2 \mathbf{1}^T \tag{2.64j}$$

$$\mathbf{P^L} - \mathbf{P_R}^k - \mathbf{P^G} + \mathbf{AP^t} \leq \mathbf{0}, \; k = 1, \ldots, M \tag{2.64k}$$

where $\mathcal{D} := \{\mathbf{P^G}, \mathbf{P^L}, \mathbf{P^t}, \mathbf{Q}, \mathbf{t}, \mathbf{s}, \mathbf{S}^L, \mathbf{S}^t, t_1, t_2\}$ contains all the decision variables (for convenience, "$\leq$" is a component-wise operator in 21b-21k). For a practical use, (2.64k) can be replaced by the following to reduce the number of constraints.

$$\mathbf{P^L} - \mathbf{P^G} + \mathbf{AP^t} \leq \min_{k=1,\ldots,M} \mathbf{P_R}^k \tag{2.65}$$

where $\min_{k=1,\ldots,M} \mathbf{P_R}^k$ is the vector that its *i*-th element corresponds to the minimum value of P_{R_i} among the M samples. Note that (P3) is a SDP relaxed reformulation for our proposed microgrid reconfiguration. To solve (P3), a weighted-sum method based scheme is presented next.

2.3.4 Multi-objective Optimization and Rounding Mechanism

Multi-objective programming is concerned with optimization problems seeking to optimize more than one objective function simultaneously. A general form is casted as follows.

$$
\begin{aligned}
&\text{minimize} \quad \mathbf{C}(\mathbf{u}) = [c_1(\mathbf{u}), c_2(\mathbf{u}), \ldots, c_m(\mathbf{u})]^T \\
&\text{subject to:} \quad \mathbf{u} \in \Omega
\end{aligned}
\tag{2.66}
$$

where $\Omega = \{\mathbf{u} \in \mathbb{R}^n : f(\mathbf{u}) \leq 0, l(\mathbf{u}) = 0\}$ denotes the feasible solution region. $c_i(\mathbf{u})$ represents the i-th objective, $\mathbf{u}$ is the unknown variable, m is the number of the objectives, and $f(\mathbf{u})$ and $l(\mathbf{u})$ account for the involving constraints (inequality and equality) in the problem. Note that, the objectives in the problem have tradeoffs among them (otherwise it becomes a single-objective program), thus no unique best decision exists. Specifically, the so-called *Pareto optimal solution* is used to represent the set of "optimal" solutions of the multi-objective problem. A formal definition is introduced below.

Pareto optimal solution: A feasible solution $\mathbf{u}^*$ of the optimization program (2.66) with no any another feasible solution $\mathbf{u} \in \Omega$ satisfying $c_i(\mathbf{u}) \geq c_i(\mathbf{u}^*)$ for every index i, and for at least an index j that $c_j(\mathbf{u}) > c_j(\mathbf{u}^*)$.

Solving the multi-objective problem (P3)   We used the weighted-sum method in this paper to solve (P3). It naturally transforms the vector of objectives into a scalar form, which can be posed as follows.

$$
\begin{aligned}
&\underset{\mathbf{u} \in \Omega}{\text{minimize}} \quad \sum_{i=1}^{m} w_i c_i'(\mathbf{u}) \\
&\text{subject to} \quad \sum_{i=1}^{m} w_i = 1, \; w_i \geq 0, \; i = 1, 2, \ldots, m
\end{aligned}
\tag{2.67}
$$

where w_i denotes the preference of the i-th objective, and $c_i'(\mathbf{u})$ is the normalized version of objective function $c_i(\mathbf{u})$. It has been proved that if $c_i(\mathbf{u}), i = 1, 2, \ldots, m$ and Ω are all convex, then applying the weighted-sum method in (2.67) can generate any Parteo solution of (2.66) [80].

Based on the facts above, the optimization problem (P3) can be converted in an alternative

way as follows.

$$(P4) \quad \underset{\mathcal{D}}{\text{minimize}} \quad \frac{w_1}{N_1} t_1 + \frac{w_2}{N_2} t_2$$

$$\text{subject to: } (2.64b) - (2.64k) \text{ and}$$

$$w_1 + w_2 = 1, \ w_1, w_2 \geq 0$$

where N_1, N_2 are the known normalization factors of objectives t_1 and t_2 respectively.

We see that (P4) is our final SDP-based microgrid reconfiguration formulation. However, in the results of (P4), the binary variables $\mathbf{s}^*$ and $\mathbf{t}^*$ denoting the status of the circuit breaks might have values between 0 and 1. Thus the next step is to convert these solutions to obtain the valid Boolean status, often referred to as "rounding".

Variable Threshold Rounding   Regard $\mathbf{s}^*$ and $\mathbf{t}^*$ as the values of $\mathbf{s}$ and $\mathbf{t}$ in the optimal solution of (P4) respectively. Let $\mathbf{s}^r$ and $\mathbf{t}^r$ represent the solutions after rounding. Then we adopt a simple rounding mechanism as follows.

We drop the obtained matrices $\left(\mathbf{S}^L\right)^*$ and $(\mathbf{S}^t)^*$, keeping only the vectors $\mathbf{s}^*$ and $\mathbf{t}^*$, and round their elements to 0 or 1. It is stated as:

$$
\begin{aligned}
\mathbf{s}_i^r &= \mathbf{sign}(\mathbf{s}^* - \mu), \quad i = 1, 2, \ldots, |N| \\
\mathbf{t}_i^r &= \mathbf{sign}(\mathbf{t}^* - \mu), \quad i = 1, 2, \ldots, |B|
\end{aligned}
\tag{2.68}
$$

where $\mu \in (0, 1)$ is a predefined threshold and the function $\mathbf{sign}$ is:

$$
\mathbf{sign}(y) = \begin{cases} +1, & y \geq 0; \\ -1, & y < 0. \end{cases}
\tag{2.69}
$$

After the rounding process for the breakers status variables $\mathbf{s}$ and $\mathbf{t}$, we can plug them into (P4) and run the optimization again to obtain the rest of the decision variables in $\mathcal{D}$.
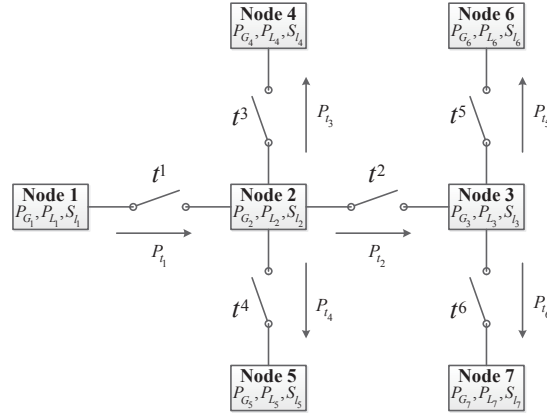
Figure (2.12) Example of 7-node Test Feeder

2.3.5 Performance Evaluation

In this section, a modified 7-node test feeder (shown in Fig. 2.12) [29] was considered to study the performance and properties of the proposed approach. The grid reconfiguration problem formulated in (P4) was solved by the package CVX ([55]) in MATLAB. The power imbalance probability is set to $\epsilon = 0.01(1\%)$, and the parameter β in theorem 2 is 0.05. According to theorem aforementioned, approximately at least 2183 samples are required. In this study, we used 5000 samples of renewable energy generation for each P_{R_i} at i-th node. The renewable energy generation P_{R_i} is modeled as a Gaussian distributed random variable for each node i following distribution $\mathcal{N}(2, 0.2)$. The threshold μ is set to 0.6. The matrix **A** can be formed based on the connectivity topology in Fig. 2.12. The other parameter settings are illustrated in the ensuing context. We first study an initial case with the parameter setting in TABLE 2.6. Fig. 2.13 shows the pareto points generated from the solution of (P4). The average number of iterations for the interior-point solver of CVX were 20 and the computation time was 1.2 seconds on a machine with Intel Duo Core 1.8GHZ. For the system operator, if the objective of total loads served has higher preference than the reconfiguration operation reliability, then the points around the right lower corner of Fig. 2.13 might be good choices. In contrast, if the operation reliability is seen more important than the total loads served, then the points around the left upper corner become right options.

TABLE 2.7-2.8 illustrate the best solutions obtained by SDP approach for minimizing the

Table (2.6) Parameter Setting of 7-node test feeder (initial case)

Parameter	Initial Setting
$\mathbf{c}$	$[2, 2, 2, 2, 2, 2]^T$ (kW)
$\mathbf{P_G^{min}}$	$[0, 0, 0, 0, 0, 0, 0]^T$ (kW)
$\mathbf{P_G^{max}}$	$[1, 0, 2, 6, 6, 6, 6]^T$ (kW)
$\mathbf{P_L^{min}}$	$[2, 2, 2, 2, 2, 2, 2]^T$ (kW)
$\mathbf{P_L^{max}}$	$[5, 5, 5, 5, 5, 5, 5]^T$ (kW)
λ_M	$[1, 0.8, 0.7, 0.6, 0.5, 0.2, 0.1]^T$
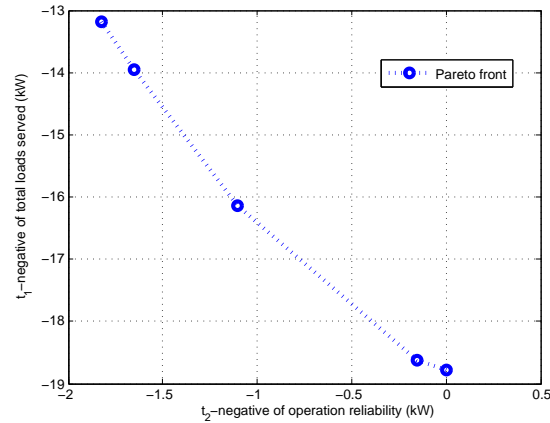
Figure (2.13) Pareto points of reconfiguration problem with initial setting

negative of total loads served (t_1) and the negative of reconfiguration operation reliability (t_2), respectively under the initial setting case. From TABLE 2.7-2.8, it can be observed that the tradeoff between objectives of total loads served and reconfiguration reliability do exist. In particular, the value of t_2 in TABLE 2.7 is extremely close to zero, which means at least one branch almost reaches its power flow limit. This verifies our concern that if only the objective of total loads served is considered, the power flows in branches might reach their limits which will significantly increase the risk of power-line fault/outage.

Table (2.7) Best Solution for the negative of total loads served

Results	Min. t_1
$\mathbf{P^G}$	$[1.00, 0.00, 1.79, 5.79, 5.80, 5.49, 5.47]^T$
$\mathbf{P^L}$	$[4.27, 5.00, 5.00, 5.00, 5.00, 5.00, 5.00]^T$
$\mathbf{P^t}$	$[-2.00, -1.92, -1.90, -1.90, -1.12, -1.13]^T$
$[t_1, t_2]$	$[-18.77, 1.02\text{E-}08]^T$
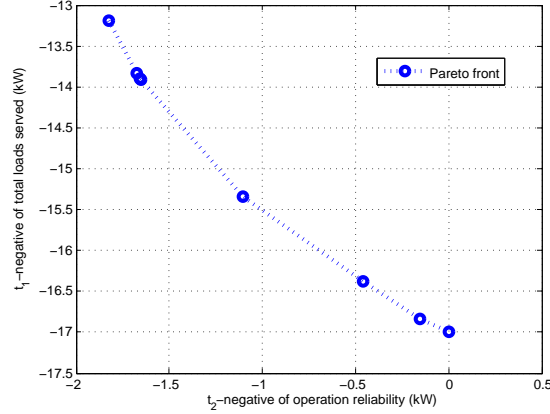
Figure (2.14) Pareto points of reconfiguration problem (case 2)

Table (2.8) Best Solution for the negative of reconfiguration reliability

Results	Min. t_2
$\mathbf{P^G}$	$[1.00, 0.00, 2.00, 5.12, 5.12, 5.15, 5.14]^T$
$\mathbf{P^L}$	$[2.10, 2.00, 3.55, 5.00, 5.00, 5.00, 5.00]^T$
$\mathbf{P^t}$	$[0.17, -0.17, -0.17, -0.17, -0.17, -0.17]^T$
$[t_1, t_2]$	$[-13.19, -1.83]^T$

The effect of changing parameter $\mathbf{P_G^{max}}$ For investigating the impact of variation of parameter $\mathbf{P_G^{max}}$ on the pareto points, we consider the second test case. We use the same parameters as in the initial case except that $\mathbf{P_G^{max}} = [1, 0, 2, 4, 4, 4, 4]^T$. The resulting solution is depicted in Fig. 2.14. We find that the value of t_1 in the right lower extreme point is higher than its counterpart in Fig. 2.13. It implies that the best solution for total loads served in case 2 is inferior to the best solution in the initial case. This phenomenon can be expected since the parameters of maximum power generations of DGs in node $4 - 7$ have been reduced in case 2.

The effect of changing parameter $\mathbf{P_L^{min}}$ Fig. 2.15 illustrates the pareto optimal points obtained through various values of vector $\mathbf{P_L^{min}}$. Lmin=1.5 means the minimum load required for each node is 1.5 kW (every component of $\mathbf{P_L^{min}}$ equals 1.5). We first note that the generated points belong to a same pareto front. This is because in these scenarios, the amount of minimum loads required is relatively small comparing to the power generations. For each node, the "actual" load supplied can always be greater than the minimum limit (Lmin=3.0). Thus, cases with Lmin=2.0,
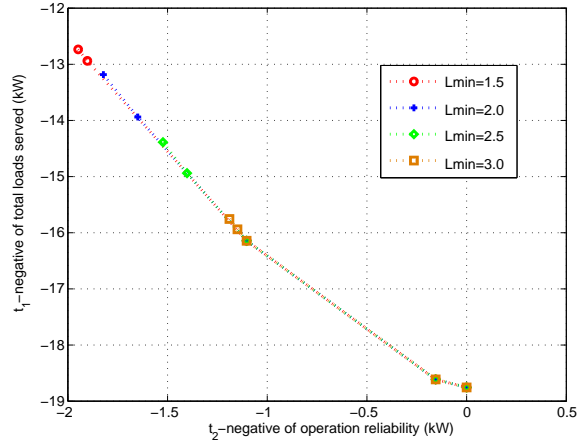
Figure (2.15) Comparison of Pareto points with different values of $\mathbf{P}_L^{min}$

2.5, 3.0 are within the solution set of Lmin=1.5. Fig. 2.15 also implies that when each node wants more energy for its minimum consumption, the reconfiguration operation would be less reliable.

Another thing can be expected is that, for example, if node 1 and node 2 do not have enough generations to satisfy their own minimum loads and the lines connecting node 1 and node 2 with the rest of the network do not have enough capacity to transmit power for satisfying at least their minimum loads, these two nodes might isolate themselves from the network [2]. Due to space limitation, we omit the detailed results on this scenario.
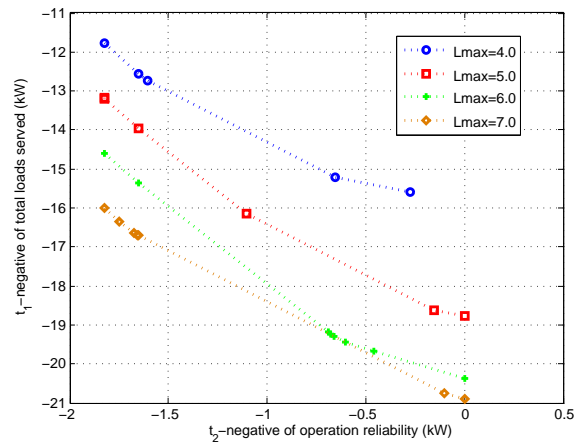
Figure (2.16) Comparison of Pareto fronts with different values of $\mathbf{P}_L^{max}$

---

[2]The breakers status are all obtained to be closed in this paper.

The effect of changing parameter $\mathbf{P_L^{max}}$ In a similar way as section VI-B, we perform tests on different values of vector $\mathbf{P_L^{max}}$. In Fig. 2.16, we notice that with larger value of maximum load needed for each node, the pareto front is lower. We observe that if the system can supply at least some nodes with their maximum loads wanted (most satisfied situation for the nodes), the total objective value with larger Lmax is superior to its counterpart with smaller one.
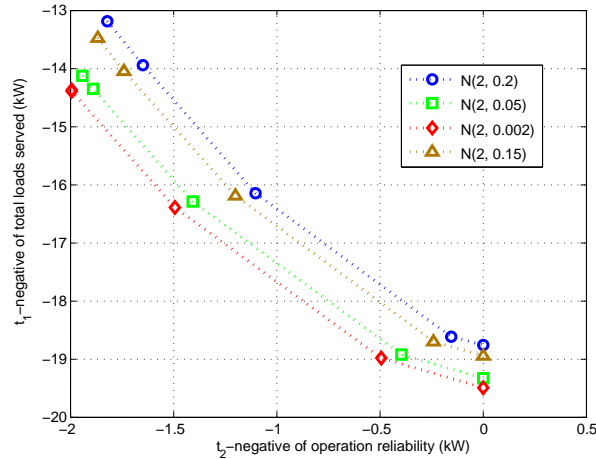
Figure (2.17) Comparison of Pareto fronts with different variances of RES generation

The effect of RES generation uncertainty   We also study the effect of RES generation uncertainty on the Pareto front. The variances of the renewable energy generation P_{R_i} for each node i are set to $0.2, 0.15, 0.05, 0.002$ (the mean value is kept the same). The other settings are the same as the initial case in TABLE 2.6. The simulated results in Fig. 2.17 show that larger variance of RES generation results in less efficient reconfiguration operation. The reason for that might be as follows. If the variance of RES generation is large, the optimal operation would be "conservative" to use RES (since the algorithm is "very unsure" on the RES generation and its amount might be underestimated), which will decrease the total efficiency. Nevertheless, the proposed algorithm would rather sacrifice the efficiency to reduce the probability of power imbalance.

The level of RES penetration in total power generation   In this section, we assess the effect of changing the level of RES penetration in each node. To this end, we fixed the maximum
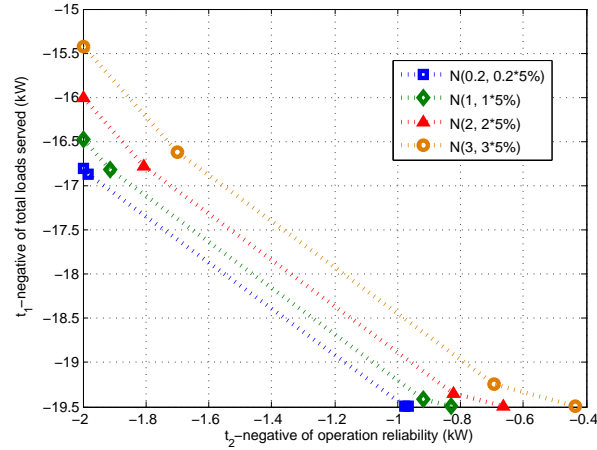
Figure (2.18) Comparison of Pareto fronts with various levels of RES penetration

power generation capacities of the 7 nodes to $[4, 3, 5, 7, 7, 7, 7]$, respectively. The statistical means of RES generation are set to 0.2, 1, 2, 3, and the maximum conventional DG generation vector $\mathbf{P}_G^{\max}$ is changed accordingly. The variances are set to 5% of their means. As shown in Fig. 2.18, an interesting fact is that as the level of RES penetration grows, the efficiency measured by the total objectives decreases. In other words, in these reliability-concerned microgrid restoration problems, the proposed algorithm might prefer more "reliable" power sources (i.e. conventional DG) to RES.
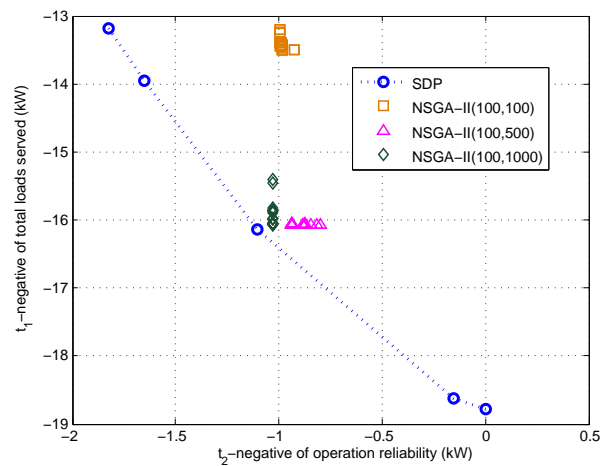
Figure (2.19) Comparison of Pareto points of the SDP-based method and NSGA-II

Comparison of the SDP-based method and NSGA-II   Finally, in Fig. 2.19, we do com-
pare our proposed SDP-based approach with a widely used multi-objective evolutionary algorithm,
called Nondominated Sorting Genetic Algorithm II (NSGA-II) [81] on the initial case (described
in TABLE 2.6). We test NSGA-II with population size 100 and obtain the pareto fronts at gen-
eration 100, 500, and 1000 respectively. It is observed in Fig. 2.19 that the SDP-based method
generate slightly less number of solutions than NSGA-II. Nevertheless, our proposed method has
better solutions in general (also reaches two extreme points). And it avoids the problem of the
clustering of the solution points appearing in NSGA-II.

Summary   A novel multi-objective microgrid reconfiguration scheme was proposed in this
chapter. To overcome the difficulties in solving the resultant optimization problem, the convex
relaxation techniques and the scenario approximation approach were adopted to obtain a compu-
tationally tractable SDP reformulation. The generated Pareto-optimal points from the SDP re-
formulation in various tests illustrated its correctness and meliority in providing critical planning
information to system decision-makers.

PART 3

IN-SITU SEISMIC TOMOGRAPHY COMPUTING

In this part, we propose several frameworks for in-situ data computing in seismic imaging system. In the first, we develop two ADMM-based decentralized algorithms, one is synchronous and the other is asynchronous, which is more fault-tolerant. Next, a gradient-based method has been proposed, which reduces the computational complexity of the ADMM-based algorithm. Considering the characteristics of seismic imaging application, we design a platform with asynchronous updating manner and broadcasting communication pattern, which is most suitable in real-world seismic tomography computing system.

3.1 ADMM-based Decentralized Approach

We propose two in-situ seismic imaging methods on decentralized large-scale seismic imaging system based on the technique of ADMM. They enjoy the nature of distributed and *fully* decentralized computing and gossip between neighbors only. One is synchronous and the other is asynchronous. They exhibit the following merits: (i) Simple implementation and no data fusion center or coordinator required; (ii) Close to "optimal" (centralized algorithm is considered as the benchmark) imaging quality even in sparse networks with severe packet loss, which are scenarios often occur in volcano monitoring; (iii) Guaranteed linear rate approaching to consensus optimal solution for every node; (iv) Lower communication cost comparing to other potential distributed methods in the literature.

3.1.1 Synchronized Distributed Seismic Tomography Algorithm-sDSTA

Problem Formulation   In this section, we give the formal formulation of decentralized seismic tomography computation problem. We assume the event location and ray tracing steps are completed. Let $\mathbf{t}_i^* = [t_{i1}^*, t_{i2}^*, \ldots, t_{iJ}^*]^T$, where t_{ij}^* is the travel time experienced by node i in the j-th

event. The travel time of a ray is the sum of the slowness in each block times the length of the ray within that block, i.e., $t_{ij}^* = \mathbf{A}_i[j, h] \cdot \mathbf{s}^*[h]$ where $\mathbf{A}_i[j, h]$ is the length of the ray from the j-th event to node i in the h-th block and $\mathbf{s}^*[h]$ is the slowness of the h-th block. Let $\mathbf{t}_i^0 = [t_{i1}^0, t_{i2}^0, \ldots, t_{iJ}^0]^T$ be the unperturbed travel times where $t_{ij}^0 = \mathbf{A}_i[j, h] \cdot \mathbf{s}^0[h]$. We can thus have the following compact form:

$$\mathbf{A}_i \mathbf{s}^* - \mathbf{A}_i \mathbf{s}^0 = \mathbf{A}_i \mathbf{s} \tag{3.1}$$

Let $\mathbf{t}_i = [t_{i1}, t_{i2}, \ldots, t_{iJ}]^T$ denotes the travel time residual such that $\mathbf{t}_i = \mathbf{t}_i^* - \mathbf{t}_i^0$, equation (3.1) is equivalent to:

$$\mathbf{A}_i \mathbf{s} = \mathbf{t}_i \tag{3.2}$$

We now have a linear relationship between the travel time residual observations, $\mathbf{t}_i$, and the slowness perturbations, $\mathbf{s}$. Since each ray path intersects with the model only at a small number of blocks compared with n, the design matrix, $\mathbf{A}_i$, is sparse. The seismic tomography inversion problem is to solve a linear system of equations:

$$\mathbf{A} \mathbf{s} = \mathbf{t} \tag{3.3}$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{t} \in \mathbb{R}^m$ and $\mathbf{s} \in \mathbb{R}^n$ is the vector to be estimated. This system is usually organized to be over-determined (let the number of rays (the number of measurements) m is larger than n) and the inversion aims to find the least-squares solution $\mathbf{s}$ such that,

$$\mathbf{s} = \arg \min_{\mathbf{s}} \| \mathbf{A} \mathbf{s} - \mathbf{t} \|^2 \tag{3.4}$$

Since vector $\mathbf{t}$ is usually noise-corrupted, the system is inconsistent. In consequence, one needs to solve a regularized least-square problem (the regularization term can vary, we use Tikhonov regularization here since it is the most popular one in seismic inversion problem):

$$\min_{\mathbf{s}} \ \frac{1}{2} \|\mathbf{A}\mathbf{s} - \mathbf{t}\|_2^2 + \lambda^2 \|\mathbf{s}\|_2^2 \tag{3.5}$$

From (3.2), we can equivalently express (3.126) as follows.

$$\min_{\mathbf{s}} \sum_{i=1}^{p} c_i(\mathbf{s}) \tag{3.6}$$

where p sensor nodes are considered in the system, $c_i(\mathbf{s}) = \frac{1}{2}\|\mathbf{A}_i\mathbf{s} - \mathbf{t}_i\|_2^2 + \lambda_i^2\|\mathbf{s}\|_2^2$. The i-th sensor node has the knowledge of $\mathbf{A}_i$ and $\mathbf{t}_i$ only. We can see (3.6) is meant a problem that each sensor node jointly optimizes (using their private local functions c_i's) to reach a consensus on the global common interest $\mathbf{s}$ that minimizing the total cost. Notice that minimizing each local function $\mathbf{c}_i$ respectively is clearly a sub-optimal solution since only partial information is utilized and the result would be inevitably biased.

Algorithm Desgin   To fit the generic two-block ADMM framework, a natural reformulation of (3.6) can be posed as:

$$\min_{\{\mathbf{s}_i\}} \sum_{i=1}^{p} c_i(\mathbf{s}_i)$$
$$\text{s.t.} \quad \mathbf{s}_i = \mathbf{s}_j, \ j \in \mathcal{N}_i, \forall i \tag{3.7}$$

where $\mathbf{s}_i$ is the local estimate of the global variable $\mathbf{s}$ at i-th sensor node, and $\mathcal{N}_i$ is the set of neighbors of node i. The constraints in (3.7) are consensus constraints that lead the neighbors to have agreement on the decision variable $\mathbf{s}$. However, in the constraints of (3.7), the estimate $\mathbf{s}_i$ is directly coupled with neighboring nodes, a parallel processing of the optimization problem is not possible.

To overcome this issue, we introduce auxiliary variables $\mathbf{q}_{ij}$, then the constraints in (3.7) are decomposed equivalently into the following form (e.g. see [83]):

$$\min_{\{\mathbf{s}_i, \mathbf{q}_{ij}\}} \sum_{i=1}^{p} c_i(\mathbf{s}_i)$$
$$\text{s.t.} \quad \mathbf{s}_i = \mathbf{q}_{ij}, \mathbf{s}_j = \mathbf{q}_{ij}, \ j \in \mathcal{N}_i, \forall i \tag{3.8}$$

Let k be the iteration index and the generic ADMM solution of (3.8) consists of updates in

(3.10a)-(3.10d). Denote $\mathcal{L}_\rho$ as the augmented Lagrangian of optimization problem (3.8) given by:

$$\mathcal{L}_\rho(\mathbf{s}_i, \mathbf{q}_{ij}, \mathbf{y}_{ij}, \mathbf{z}_{ij}) = \sum_{i=1}^{p} \{c_i(\mathbf{s}_i) + \sum_{j \in \mathcal{N}_i} \{\mathbf{y}_{ij}^T(\mathbf{s}_i - \mathbf{q}_{ij}) +$$
$$\mathbf{z}_{ij}^T(\mathbf{s}_j - \mathbf{q}_{ij}) + \frac{\rho}{2}\|\mathbf{s}_i - \mathbf{q}_{ij}\|_2^2 + \frac{\rho}{2}\|\mathbf{s}_j - \mathbf{q}_{ij}\|_2^2\}\} \tag{3.9}$$

where $\rho > 0$ is a predefined parameter. Note that $\mathbf{y}_{ij}$ and $\mathbf{z}_{ij}$ are the Lagrangian multipliers for the first and second constraint in (3.8), respectively.

$$\mathbf{s}_i^{k+1} = \underset{\{\mathbf{s}_i\}}{\text{argmin}} \ \mathcal{L}_\rho(\mathbf{s}_i, \mathbf{q}_{ij}^k, \mathbf{y}_{ij}^k, \mathbf{z}_{ij}^k) \tag{3.10a}$$

$$\mathbf{q}_{ij}^{k+1} = \underset{\{\mathbf{q}_{ij}\}}{\text{argmin}} \ \mathcal{L}_\rho(\mathbf{s}_i^{k+1}, \mathbf{q}_{ij}, \mathbf{y}_{ij}^k, \mathbf{z}_{ij}^k) \tag{3.10b}$$

$$\mathbf{y}_{ij}^{k+1} = \mathbf{y}_{ij}^k + \rho(\mathbf{s}_i^{k+1} - \mathbf{q}_{ij}^{k+1}) \tag{3.10c}$$

$$\mathbf{z}_{ij}^{k+1} = \mathbf{z}_{ij}^k + \rho(\mathbf{s}_j^{k+1} - \mathbf{q}_{ij}^{k+1}) \tag{3.10d}$$

(3.10a)-(3.10d) are over every node i and all its neighbours $j \in \mathcal{N}_i$. Typically, each node i updates its variables in (3.10a)-(3.10d) and performs only local communications with its immediate neighbors.

Notice that the generic ADMM in (3.10a)-(3.10d) requires sensor node i to update variables $\{\mathbf{s}_i, \mathbf{q}_{ij}, \mathbf{y}_{ij}, \mathbf{z}_{ij}\}$ in each iteration, where each variable is the same size as $\mathbf{s}_i$. It is evidently a huge burden when the size of variable $\mathbf{s}_i$ is large. Fortunately, a simplified efficient version for our problem can be obtained as follows (only needs to transmit $\mathbf{s}_i$ for sensor i).

$$\mathbf{s}_i^{k+1} = \left(\mathbf{A}_i^T\mathbf{A}_i + (2\lambda_i^2 + 2\rho'|\mathcal{N}_i|)\mathbf{I}\right)^{-1}\left(\mathbf{A}_i^T\mathbf{t}_i - \mathbf{u}_i^k + \rho'\left(|\mathcal{N}_i|\mathbf{s}_i^k + \sum_{j \in \mathcal{N}_i}\mathbf{s}_j^k\right)\right) \tag{3.11a}$$

$$\mathbf{u}_i^{k+1} = \mathbf{u}_i^k + \rho'\left(|\mathcal{N}_i|\mathbf{s}_i^{k+1} - \sum_{j \in \mathcal{N}_i}\mathbf{s}_j^{k+1}\right) \tag{3.11b}$$

where $|\mathcal{N}_i|$ represents the cardinality of set $\mathcal{N}_i$. $\rho' = \rho/2$.

Theorem 6 *For the problem in* (3.8)*, the updates of* (3.10a)-(3.10d) *are equivalent to* (3.11a)-

(3.11b).

Proof 3 *The goal here is to reduce the original ADMM steps* (3.10a)-(3.10d) *to* (3.11a)-(3.11b). *To this end, we first focus on* (3.10b). *A closer look reveals that a closed-form solution of* (3.10b) *can be obtained as follows:*

$$\mathbf{q}_{ij}^{k+1} = \frac{1}{2\rho}\left\{\rho(\mathbf{s}_i^{k+1} + \mathbf{s}_j^{k+1}) + (\mathbf{y}_{ij}^k + \mathbf{z}_{ij}^k)\right\} \tag{3.12}$$

Adding both sides of (3.10c) *and* (3.10d) *yields:*

$$\mathbf{y}_{ij}^{k+1} + \mathbf{z}_{ij}^{k+1} = (\mathbf{y}_{ij}^k + \mathbf{z}_{ij}^k) + \rho(\mathbf{s}_i^{k+1} + \mathbf{s}_j^{k+1}) - 2\rho \cdot \mathbf{q}_{ij}^{k+1} \tag{3.13}$$

After plugging (3.12) *into the right side of* (3.13), *we obtain the following result:*

$$\mathbf{y}_{ij}^{k+1} + \mathbf{z}_{ij}^{k+1} = \mathbf{0} \tag{3.14}$$

Consequently, by plugging (3.14) *back into* (3.12), *we have*

$$\mathbf{q}_{ij}^{k+1} = \frac{1}{2}(\mathbf{s}_i^{k+1} + \mathbf{s}_j^{k+1}) \tag{3.15}$$

We now re-express (3.10a) *removing all the independent items with respect to variable* $\mathbf{s}_i$:

$$
\begin{aligned}
\mathbf{s}_i^{k+1} &= \operatorname*{argmin}_{\{\mathbf{s}_i\}} \left\{ c_i(\mathbf{s}_i) + \sum_{j\in\mathcal{N}_i}(\mathbf{y}_{ij}^k)^T\mathbf{s}_i + \frac{\rho}{2}\sum_{j\in\mathcal{N}_i}\|\mathbf{s}_i - \mathbf{q}_{ij}^k\|_2^2 \right\} \\
&= \operatorname*{argmin}_{\{\mathbf{s}_i\}} \left\{ c_i(\mathbf{s}_i) + \sum_{j\in\mathcal{N}_i}(\mathbf{y}_{ij}^k)^T\mathbf{s}_i + \frac{\rho}{2}\sum_{j\in\mathcal{N}_i}\|\mathbf{s}_i - \frac{1}{2}(\mathbf{s}_i^k + \mathbf{s}_j^k)\|_2^2 \right\}
\end{aligned}
\tag{3.16}
$$

Let $\mathbf{u}_i^k = \sum\limits_{j\in\mathcal{N}_i}(\mathbf{y}_{ij}^k)$. *Since function* c_i *is differentiable with respect to* $\mathbf{s}_i$, (3.16) *is equivalent to solving the following equation:*

$$\nabla c_i(\mathbf{s}_i^{k+1}) + \mathbf{u}_i^k + 2\rho|\mathcal{N}_i|\mathbf{s}_i^{k+1} - \rho'\left(|\mathcal{N}_i|\mathbf{s}_i^k + \sum_{j\in\mathcal{N}_i}\mathbf{s}_j^k\right) = 0 \tag{3.17}$$

where ∇ *denotes the gradient of the function. Plugging* $c_i(\mathbf{s}_i^{k+1}) = \frac{1}{2}\|\mathbf{A}_i\mathbf{s}_i^{k+1} - \mathbf{t}_i\|_2^2 + \lambda_i^2\|\mathbf{s}_i^{k+1}\|_2^2$ *into* (3.17) *yields the following equation, which becomes* (3.11a) *immediately.*

$$\left(\mathbf{A}_i^T\mathbf{A}_i + (2\lambda_i^2 + 2\rho'|\mathcal{N}_i|)\mathbf{I}\right)\mathbf{s}_i^{k+1} = \mathbf{A}_i^T\mathbf{t}_i - \mathbf{u}_i^k + \rho'\left(|\mathcal{N}_i|\mathbf{s}_i^k + \sum_{j\in\mathcal{N}_i}\mathbf{s}_j^k\right) \tag{3.18}$$

Now we need to derive the updating equation for $\mathbf{u}_i$*. Substituting* (3.15) *into* (3.10c)*, we can have (consider* $\mathbf{u}_i^{k+1}$*):*

$$\begin{aligned}
\mathbf{u}_i^{k+1} &= \sum_{j\in\mathcal{N}_i}(\mathbf{y}_{ij}^{k+1}) \\
&= \left\{\sum_{j\in\mathcal{N}_i}(\mathbf{y}_{ij}^k)\right\} + \frac{\rho}{2}\left(|\mathcal{N}_i|\mathbf{s}_i^{k+1} - \sum_{j\in\mathcal{N}_i}\mathbf{s}_j^{k+1}\right) \\
&= \mathbf{u}_i^k + \frac{\rho}{2}\left(|\mathcal{N}_i|\mathbf{s}_i^{k+1} - \sum_{j\in\mathcal{N}_i}\mathbf{s}_j^{k+1}\right)
\end{aligned} \tag{3.19}$$

Define $\rho' = \rho/2$*, then* (3.11b) *follows. Interested readers are referred to [84] and [85] for similar derivations of the average consensus problem and the sparse linear regression problem, respectively.*

Note that obtaining $\mathbf{s}_i^{k+1}$ in (3.11a) is solving a linear system of equations. Theoretically, we can use any solver while considering the property of tomography inversion problem, we adopt Bayesian Algebraic Reconstruction Technique (BART) method in this paper [86].

A closer look into (3.11a) and (3.11b) reveals that for node i, the information needed is only the summation of its neighbors's current estimates $\mathbf{s}_j^k$, $j \in \mathcal{N}_i$. Thus a natural implementation is that: in every iteration, all the nodes broadcast their current estimates to all their neighbors. After receiving all the neighbors's estimates, the sensor nodes can perform local updates in parallel. When all the nodes finish their computations, they will broadcast their new estimates again. The idea is summarized in Algorithm 1.

Note that in step 5 of Algorithm 1, the minimization of $\mathbf{s}_i^k$ is not carried out exactly. In this situation, the proposed ADMM-based algorithm will still converge provided certain mild conditions [87](pp.26). More importantly, solving the minimization of $\mathbf{s}_i^k$ to a very high accuracy, especially at the initial iterations, may not be worthwhile. The reason is two-folds: first, at the initial iterations (communications), node i has limited information (even no information) about the whole structure

---

Algorithm 3 Synchronized Distributed Seismic Tomography Algorithm (sDSTA)

1: Initialization: Input $\mathbf{A}_i$, $\mathbf{t}_i$ for sensor i, $\forall i$.
   Initialize $\mathbf{s}_i^0 = \mathbf{0}$, $\mathbf{u}_i^0 = \mathbf{0}$, $\forall i$. Set parameters $\rho' > 0$ and λ_i, $\forall i$.
2: At each iteration k, $(k = 0, 1, \ldots)$, every sensor i broadcasts its $\mathbf{s}_i^k$ to its neighbors $j \in \mathcal{N}_i$.
3: Once every head i receives all its neighbors's estimates,
   do the following local updating.
4: Compute $\mathbf{u}_i^k$ based on (3.11b).
5: Update $\mathbf{s}_i^{k+1}$ by running a finite number of BART iterations on (3.11a), with initial value $\mathbf{s}_i^k$.
6: When all the nodes finish their primal updates $\mathbf{s}_i^{k+1}$, $\forall i$,
   broadcast them to all neighbors, and repeats step 4-6
   until certain stopping criterion has been satisfied.

---

due to its nature of local communication with immediate neighbors. In this situation, the solution $\mathbf{s}_i$ may has a large deviation from the true solution no matter how hard we try for the solution. In addition, solving $\mathbf{s}_i^k$ is an iterative process and might require a large amount of time (iterations) to have a solution with very small relative error.

Regarding the communication cost of this proposed scheme, we see that it depends on the network topology. The quantitive relation can be described in theorem 2.

Theorem 7 *The communication cost of sDSTA for each node is $o(k)$ and the total communication cost is at most $o(kN(N-1))$ (complete network) and can be as low as $o(2kN)$ (ring network) with respective to network size N and communication round k.*

In addition, the convergence speed of the proposed iterative method also depends on the communication topology. In general, network with larger average degree is expected to converge faster. We will discuss it in detail in Section 6.

3.1.2   Asynchronous Distributed Seismic Tomography Algorithm-aDSTA

We can recall that the updating in sDSTA requires each sensor node to wait until receiving the solution of its slowest neighbor. While in some cases, it might be desirable to update with less coordination and in an asynchronous fashion such that every node can *independently* determine its own actions. In this section, we provide a solution methodology to address this problem.

Component 1: Adaptive Communication   One major motivation of this adaptive communication scheme is that: in sensor network applications such as our distributed seismic tomography, communication is the most energy-consuming component. In each iteration, sensor nodes need to broadcast their local estimates $\mathbf{s}_i$'s, which is a long vector (e.g. in real data of Mount St.Helens, the size could be around 768,000). Thus it is highly demanded to design a mechanism that can reduce the communication overhead for the proposed distributed algorithm [88]. Specifically, assume a coordinator exists requiring all the nodes to perform m iterations. However, some node may already owns a good enough solution in less than m iterations, and its value does not change much for several successive iterations. In this situation, it would be a waste if these nodes continue to communicate and update. To address this problem, an adaptive communication scheme is proposed in the following context.

Each sensor node should be able to determine if it needs to communicate and computing. In each iteration, each node will compute an index which depends on the relative-updates. If this index is greater than a threshold, then the node will stop grabbing updates of its neighbors.

Proposed Scheme: For sensor node i, if the following two conditions satisfied:

$$\frac{\|\mathbf{s}_i^k - \mathbf{s}_i^{k-1}\|}{\|\mathbf{s}_i^{k-1}\|} \le \varphi_i \text{ and } \left\|\mathbf{u}_i^k - \mathbf{u}_i^{k-1}\right\| \le \mu_i \tag{3.20}$$

then $ct_i = ct_i + 1$, otherwise $ct_i = 0$, where ct_i is a counter of node i and define E_i as a threshold for node i. When $ct_i > E_i$, sensor node i stops updating, which also means that it stops querying the estimates from its neighbors. When node i stops updating, its neighbors can still ask for the latest value of node i to improve their accuracy. Notice that in this situation, once node j obtains the last update of node i, it will not need more transmissions from i since the estimate of node i will not change. Note that the stopping criterion (3.20) is different from those of conventional ADMM methods where all the sensors shall stop computations at the same time using a common stopping criteria and common primal and dual tolerances. Stopping criteria (3.20) allows a sensor i to stop its computation asynchronously and independently from other sensors. However, these criteria themselves are insufficient for asynchronous implementation. Synchronization is still required for

dual and primal variable updates at iteration $k + 1$ due to their dependencies on values of k-th iteration.

Algorithm 4 Asynchronous Distributed Seismic Tomography Algorithm (aDSTA): For sensor i

---

1: Initialization: Input $\mathbf{A}_i$, $\mathbf{t}_i$.
   Initialize $\mathbf{s}_i^0 = \mathbf{0}$, $\mathbf{u}_i^0 = \mathbf{0}$, $ct_i = 0$. Set parameters $\rho' > 0$, φ_i, μ_i, E_i and λ_i.
2: **while** *criterion $ct_i > E_i$ is not satisfied* **do**
3:     At new iteration $k + 1$, sensor i selects a neighbor j
       according to the probability distribution in (3.21).
4:     Sensor i contacts j to obtain j's current value $\mathbf{s}_j$.
5:     Compute $\mathbf{u}_i^k$ based on (3.11b) with replacing
       $\sum_{j \in \mathcal{N}_i} \mathbf{s}_j$ by $|\mathcal{N}_i| \cdot \mathbf{s}_j$.
6:     Update $\mathbf{s}_i^{k+1}$ by running a finite number of BART iterations on (3.18) with replacing $\sum_{j \in \mathcal{N}_i} \mathbf{s}_j$
       by $|\mathcal{N}_i| \cdot \mathbf{s}_j$.
       Use $\mathbf{s}_i^k$ as the initial guess.
7:     When sensor i finishes its primal updates $\mathbf{s}_i^{k+1}$, go to the following:
8:     **if** *criterion in (3.20) is satisfied* **then**
9:         $ct_i = ct_i + 1$
10:    **else**
11:       $ct_i = 0$

Component 2: Randomized Gossiping To ensure full asynchronous implementation, the update of each node should be performed in a randomized manner (otherwise there must be some predefined order for updating). To this end, we use the doubly stochastic matrix, $\mathbf{T} \in \mathbb{R}^{p \cdot p}$ for deciding the communications among sensors, where $\mathbf{T}_{ij}$ is the probability that a sensor i contacts another sensor j at any iteration. In a mathematical form, we can have

$$\mathbf{T}_{ij} = \begin{cases} h(d_{ij}), & j \in \mathcal{N}_i; \\ 0, & otherwise. \end{cases} \tag{3.21}$$

where $d_{ij}, \{j \in S : j \in \mathcal{N}_i\}$ is the distance between sensor i and j. S is the set of sensors. $h(\cdot)$ is a function of d_{ij}. At iteration $k + 1$, sensor i may need to ask only one of the neighbors j to send its estimate to i, unless i's stopping criteria are already satisfied, whereas sensor j can be contacted by

more than one sensor, even when both of j's stopping criteria are satisfied.

Observation: In seismic tomography, each station has the ray information coming from the events to the station. Thus, the closer the two stations are, the higher the similarity between their obtained ray information is. Based on this observation, we can design the probability $\mathbf{T}_{ij}$ in the following way: At each iteration, for sensor i, the probability of selecting its neighbor sensor j is proportional to their distance d_{ij}. Since we give more weight to the neighbor that contributes more new information to sensor i, our proposed scheme is expected to converge faster than the method selecting its neighbor with uniform distribution (since the unequal weighting might help the sensors reach consensus).

The communication scheme for sensor i can be described as follows. When the conditions in (3.20) are not satisfied, sensor i selects a neighbor j according to the probability distribution of $\mathbf{T}_{ij}$. Upon contacting j, sensor i pulls the current value of $\mathbf{s}_j$ from j. During the computation, first update dual variable $\mathbf{u}_i$ by replacing $\sum_{j \in \mathcal{N}_i} \mathbf{s}_j$ with $|\mathcal{N}_i|\mathbf{s}_j$. Second, for updating primal variable $\mathbf{s}_i$, in the right side of (3.18), the item $\sum_{j \in \mathcal{N}_i} \mathbf{s}_j$ is replaced by $|\mathcal{N}_i|\mathbf{s}_j$. After updating the primal variable $\mathbf{s}_i$, sensor i will again do the similar randomized communication process for receiving new $\mathbf{s}_j$. Sensor i repeats the above steps until conditions in (3.20) are already satisfied. The algorithm with this proposed communication scheme is illustrated in Algorithm 2.

Note that the set of neighbors $\mathcal{N}_i$ of sensor i in fact contains the sensors within the communication range of node i. That is, if all the sensors except sensor i in the network are within the communication range of sensor i, then they are all considered as neighbors of sensor i.

3.1.3 Convergence Analysis of sDSTA and aDSTA

Convergence rate is a critical factor in designing our proposed methods since faster convergence means less communication rounds and more bandwidth saving. In this section, we exploit markov chain and spectral theory to derive the convergence speed of sDSTA and aDSTA. We find that both methods can achieve linear convergence rate in theory.

Theorem 8 *The proposed sDSTA and aDSTA methods converge to an optimal solution at a linear rate $O(\alpha^k)$ for some $\alpha < 1$, in terms of the per-node communications k.*

In the following section, we will discuss in detail the proof of Theorem 1 in the convergence analysis. Our analysis tool is similar to that in [84] [III.A] where it considers the problem of average consensus, which is different from our seismic tomography problem defined in (3.126). The attack plan is construct a linear system capturing the dynamics of the solution and utilize the properties of markov chain (described in Lemma 1 and 2) to obtain the convergence rate [89].

Convergence Rate Analysis In order to obtain the convergence rate of proposed aDSTA framework, The first step is to describe the linear systems described in aDSTA (we will see that sDSTA is a special case under this framework). To this end, we first focus on the state update for sensor i.

Recall that in aDSTA, the original updating equation for $\mathbf{s}_i^{k+1}$ in (3.18) is modified by replacing $\sum_{j \in \mathcal{N}_i} \mathbf{s}_j$ with $|\mathcal{N}_i| \cdot \mathbf{s}_j$, for some random neighbor j. According to the distribution in (3.21), the expected value of $|\mathcal{N}_i| \cdot \mathbf{s}_j$ is:

$$|\mathcal{N}_i| \cdot \sum_{j \in \mathcal{N}_i} \mathbf{T}_{ij} \mathbf{s}_j \tag{3.22}$$

At this point, we can rewrite the updating of $\mathbf{s}_i^{k+1}$ in aDSTA as follows.

$$\left(\mathbf{A}_i^T \mathbf{A}_i + (2\lambda_i^2 + 2\rho'|\mathcal{N}_i|)\mathbf{I}\right)\mathbf{s}_i^{k+1} - \rho'|\mathcal{N}_i|\mathbf{s}_i^k$$
$$- \rho'|\mathcal{N}_i| \cdot \left\{ \sum_{j \in \mathcal{N}_i} \mathbf{T}_{ij} \mathbf{s}_j^k \right\} + \mathbf{u}_i^k - \mathbf{A}_i^T \mathbf{t}_i = \mathbf{0} \tag{3.23}$$

The updating rule for $\mathbf{s}_i^k$ can be obtained accordingly:

$$\left(\mathbf{A}_i^T \mathbf{A}_i + (2\lambda_i^2 + 2\rho'|\mathcal{N}_i|)\mathbf{I}\right)\mathbf{s}_i^k - \rho'|\mathcal{N}_i|\mathbf{s}_i^{k-1}$$
$$- \rho'|\mathcal{N}_i| \cdot \left\{ \sum_{j \in \mathcal{N}_i} \mathbf{T}_{ij} \mathbf{s}_j^{k-1} \right\} + \mathbf{u}_i^{k-1} - \mathbf{A}_i^T \mathbf{t}_i = \mathbf{0} \tag{3.24}$$

Also, (3.11b) can be rewritten as:

$$\mathbf{u}_i^{k+1} = \mathbf{u}_i^k + \rho'|\mathcal{N}_i|\mathbf{s}_i^{k+1} - \rho'|\mathcal{N}_i| \cdot \left\{ \sum_{j \in \mathcal{N}_i} \mathbf{T}_{ij} \mathbf{s}_j^{k+1} \right\} \tag{3.25}$$

Combining the previous three equations (3.23)-(3.25) yields:

$$
\begin{aligned}
\left(\mathbf{A}_i^T \mathbf{A}_i + (2\lambda_i^2 + 2\rho'|\mathcal{N}_i|)\mathbf{I}\right)\mathbf{s}_i^{k+1} = \\
\left(\mathbf{A}_i^T \mathbf{A}_i + (2\lambda_i^2 + 2\rho'|\mathcal{N}_i|)\mathbf{I}\right)\mathbf{s}_i^k + 2\rho'|\mathcal{N}_i| \cdot \left\{\sum_{j\in\mathcal{N}_i} \mathbf{T}_{ij}\mathbf{s}_j^k\right\} \\
- \rho'|\mathcal{N}_i|\mathbf{s}_i^{k-1} - \rho'|\mathcal{N}_i| \cdot \left\{\sum_{j\in\mathcal{N}_i} \mathbf{T}_{ij}\mathbf{s}_j^{k-1}\right\}
\end{aligned}
\tag{3.26}
$$

In a clearer form, the state update for sensor i can be expressed as:

$$
\begin{aligned}
\mathbf{s}_i^{k+1} = \mathbf{s}_i^k + 2\rho'|\mathcal{N}_i| \cdot \left(\mathbf{A}_i^T \mathbf{A}_i + (2\lambda_i^2 + 2\rho'|\mathcal{N}_i|)\mathbf{I}\right)^{-1} \cdot \\
\left\{\sum_{j\in\mathcal{N}_i} \mathbf{T}_{ij}\mathbf{s}_j^k\right\} - \rho'|\mathcal{N}_i|\left(\mathbf{A}_i^T \mathbf{A}_i + (2\lambda_i^2 + 2\rho'|\mathcal{N}_i|)\mathbf{I}\right)^{-1}\mathbf{s}_i^{k-1} \\
- \rho'|\mathcal{N}_i| \cdot \left(\mathbf{A}_i^T \mathbf{A}_i + (2\lambda_i^2 + 2\rho'|\mathcal{N}_i|)\mathbf{I}\right)^{-1}\left\{\sum_{j\in\mathcal{N}_i} \mathbf{T}_{ij}\mathbf{s}_j^{k-1}\right\}
\end{aligned}
\tag{3.27}
$$

Now the matrix form for the linear system in aDSTA is:

$$
\mathbb{E}\left[\mathbf{v}^{k+1}\right] = \mathbb{E}[\mathbf{M}] \cdot \mathbf{v}^k
\tag{3.28}
$$

$$
\mathbf{M} = \begin{bmatrix} \mathbf{F} & \mathbf{G} \\ \mathbf{I} & \mathbf{0} \end{bmatrix}
\tag{3.29}
$$

where $\mathbf{F}, \mathbf{G}, \mathbf{I}$ and $\mathbf{0}$ are all $p \times p$ dimensional block matrices respectively. $\mathbf{v}^{k+1} = \left[\bar{\mathbf{s}}^{k+1}, \bar{\mathbf{s}}^k\right]^T$, where $\bar{\mathbf{s}}^k$ stacks all the sensor solutions at iteration k. For notation convenience, we drop $\mathbb{E}[\,\cdot\,]$ in (24) in the later context.

Regarding matrix $\mathbf{F}$, the entry $\mathbf{F}_{ii} = \mathbf{I}$ and entry (i, j) is: $\mathbf{F}_{ij} = 2\mathbf{T}_{ij}\mathbf{P}_i$. With respect to matrix $\mathbf{G}$, we have $\mathbf{G}_{ii} = -\mathbf{P}_i$, $\mathbf{G}_{ij} = -\mathbf{T}_{ij}\mathbf{P}_i$, where $\mathbf{P}_i = \rho'|\mathcal{N}_i|\left(\mathbf{A}_i^T \mathbf{A}_i + (2\lambda_i^2 + 2\rho'|\mathcal{N}_i|)\mathbf{I}\right)^{-1}$.

The state update (24) is valid for $k > 0$. Notice that in aDSTA we initialize $\mathbf{s}_i^0 = \mathbf{0}, \mathbf{u}_i^0 = \mathbf{0}, \forall i$. By plugging in these initial conditions into (3.23), it is shown that $\mathbf{s}_i^1 = \left(\mathbf{A}_i^T \mathbf{A}_i + (2\lambda_i^2 + 2\rho'|\mathcal{N}_i|)\mathbf{I}\right)^{-1}\mathbf{A}_i^T \mathbf{t}_i$.

Thus the initial state vector $\mathbf{v}^1$ can be expressed as:

$$\mathbf{v}^1 = \left[\mathbf{R}_1, \cdots, \mathbf{R}_p, \mathbf{0}, \cdots, \mathbf{0}\right]^T \tag{3.30}$$

where $\mathbf{R}_i = \left(\mathbf{A}_i^T \mathbf{A}_i + (2\lambda_i^2 + 2\rho'|\mathcal{N}_i|)\mathbf{I}\right)^{-1}\mathbf{A}_i^T \mathbf{t}_i$.

Before approaching our final claim, we state two properties of matrix $\mathbf{M}$, which will help derive the convergence rate of proposed algorithms.

Lemma 1 *Matrix* $\mathbf{M}$ *has an eigenvalue equals* 1.

Proof 4 *First, in the second row of matrix* $\mathbf{M}$, *the summation is* $\mathbf{I}$, *thus in the lower part, the summation of every row is* 1. *Now we look at the upper part, considering the facts in about matrices* $\mathbf{F}$ *and* $\mathbf{G}$, *the summation of i-th row can be expressed as:*

$$\begin{aligned}
&\mathbf{F}_{ii} + \mathbf{G}_{ii} + \sum_{j\in\mathcal{N}_i}\mathbf{F}_{ij} + \sum_{j\in\mathcal{N}_i}\mathbf{G}_{ij} \\
=&\mathbf{I} - \mathbf{P}_i + \sum_{j\in\mathcal{N}_i}2\mathbf{T}_{ij}\mathbf{P}_i - \sum_{j\in\mathcal{N}_i}\mathbf{T}_{ij}\cdot\mathbf{P}_i \\
=&\mathbf{I} - \mathbf{P}_i + \sum_{j\in\mathcal{N}_i}\mathbf{T}_{ij}\mathbf{P}_i
\end{aligned} \tag{3.31}$$

Recall that matrix $\mathbf{T}$ *is defined as a doubly stochastic matrix yielding* $\sum_{j\in\mathcal{N}_i}\mathbf{T}_{ij} = 1$. *Hence the above equation equals* $\mathbf{I}$. *Consequently, both the upper and lower part of matrix* $\mathbf{M}$ *satisfies the condition that the summation of each row is* 1. *Based on this fact, it is straightforward to show Lemma 1 (the corresponding eigenvector contains scaling factor for each element in a row).*

Lemma 2 *There exists a selection of value* ρ' *and matrix* $\mathbf{T}$ *such that the second largest eigenvalue of matrix* $\mathbf{M}$ *is smaller than* 1.

Proof 5 *Based on Lemma 3.51, we can construct the corresponding left and right eigenvectors denoted by* l_v^1 *and* r_v^1, *respectively (also scale the eigenvectors such that* $l_v^1 r_v^1 = 1$). *Next, we need to find a limiting point of the state transition matrix. From the definition, we can have* $l_v^1 \mathbf{M} = l_v^1$, $\mathbf{M} r_v^1 = r_v^1$. *Consequently,*

$$\lim_{k\to\infty}\mathbf{M}^k = r_v^1 l_v^1$$

The above fact implies that $r_v^1 l_v^1 = \mathbf{M}^\infty$ is a limiting point [89]. Akin to the techniques used in [90], it can be shown that once the following condition is satisfied, all the other eigenvalues of matrix $\mathbf{M}$ is less than 1. *Thus Lemma 2 is verified.*

$$\|\mathbf{M} - \mathbf{M}^\infty\|_2 < 1 \tag{3.32}$$

Here $\|\cdot\|_2$ is the spectral norm of a matrix.

Since the state transition matrix $\mathbf{M}$ has the two above properties, based on the classical convergence analysis in [89], the convergence rate can be obtained as:

$$\|\mathbf{v}^{k+1} - \mathbf{v}^\infty\|_2 \leqslant C\sigma_2^{k-J+1} \tag{3.33}$$

where C is a constant, J is the size of the largest Jordan block of $\mathbf{M}$. σ_2 denotes the second largest eigenvalue of matrix $\mathbf{M}$. Theorem 1 is thus verified.

Observation: Recall that in (3.22), if we assign the probability $\mathbf{T}_{ij}$ in a uniform distributed manner such that:

$$\mathbf{T}_{ij} = \frac{1}{|\mathcal{N}_i|} \tag{3.34}$$

Then we have,

$$|\mathcal{N}_i| \cdot \sum_{j \in \mathcal{N}_i} \mathbf{T}_{ij}\mathbf{s}_j = \sum_{j \in \mathcal{N}_i} \mathbf{s}_j \tag{3.35}$$

This implies that in this case the convergence rate of aDSTA is the same as sDSTA in expectation.

Considering the convergence rate in (3.33), we see that the second largest eigenvalue of matrix $\mathbf{M}$ is an important factor, which depends on the value of ρ' and the selection of matrix $\mathbf{T}$. In fact, besides our $\mathbf{T}$ matrix selection rule in III-B (given a certain feasible value ρ'), another strategy is to minimize σ_2 by solving the following optimization problem (akin to the fastest mixing Markov chain problem [91]):

$$\begin{aligned} \underset{t,\mathbf{T}}{\text{minimize}} \qquad & t \\ \text{subject to} \quad & -t\mathbf{I} \leq \mathbf{M}(\mathbf{T}) - \mathbf{M}^\infty \leq t\mathbf{I} \end{aligned} \tag{3.36}$$

Here **M(T)** indicates the transition matrix **M** is a function of **T**. $\mathbf{M}^{\infty}$ is a limiting point of **M**. The symbol $\preceq$ in $\mathbf{A} \preceq \mathbf{B}$ means matrix $\mathbf{B} - \mathbf{A}$ is positive semidefinite.

3.1.4   Performance Evaluation

In this section, we evaluate the performance and characteristics of sDSTA and aDSTA respectively. Our experiments are performed through MATLAB and network emulators [92]. In the performance study, both synthetic and real seismic imaging data sets are tested.

(a) Error

(b) Residual
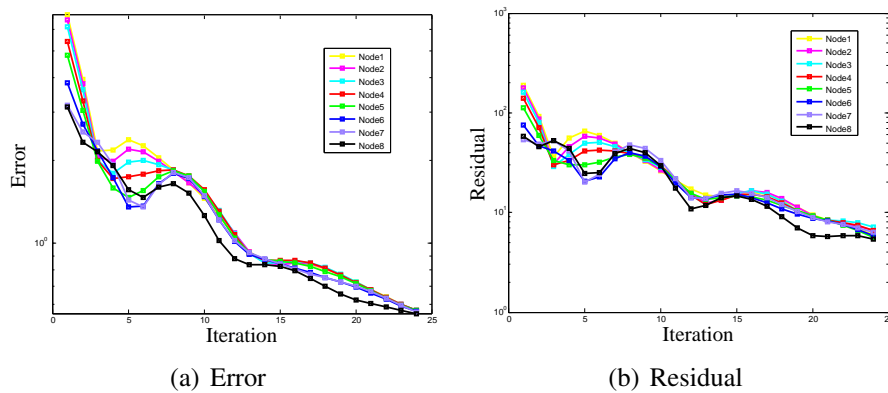
Figure (3.1) Convergence behavior of sDSTA.

Synthetic Data (2-D Model)   The performance analysis here is based on the data set generated using code in [93]. We create a 2-D seismic tomography test problem with an $N \cdot N$ domain, using n sources located on the right boundary and p receivers (seismographs) scattered along the left and top boundary. The rays are transmitted from each source to each receiver. We know that the regularization parameter λ depends on the application and data sets, which means for specific scenario, λ is chosen based on the understanding and knowledge of the application. For simplicity, we fix $\lambda^2 = 1$ and $\lambda_i^2 = 1/p$ in our experiments. The parameter ρ' is set to be 0.5.

Convergence behavior of sDSTA   We first study the performance of sDSTA with a case that $\{N = 16$ (tomography resolution is 16×16), $n = 64$, $p = 32\}$. The associated communicate network is a complete network (every node can communicate with all the other nodes) and there is

(a) Groud truth

(b) Centralized

(c) Complete network

(d) Ring network
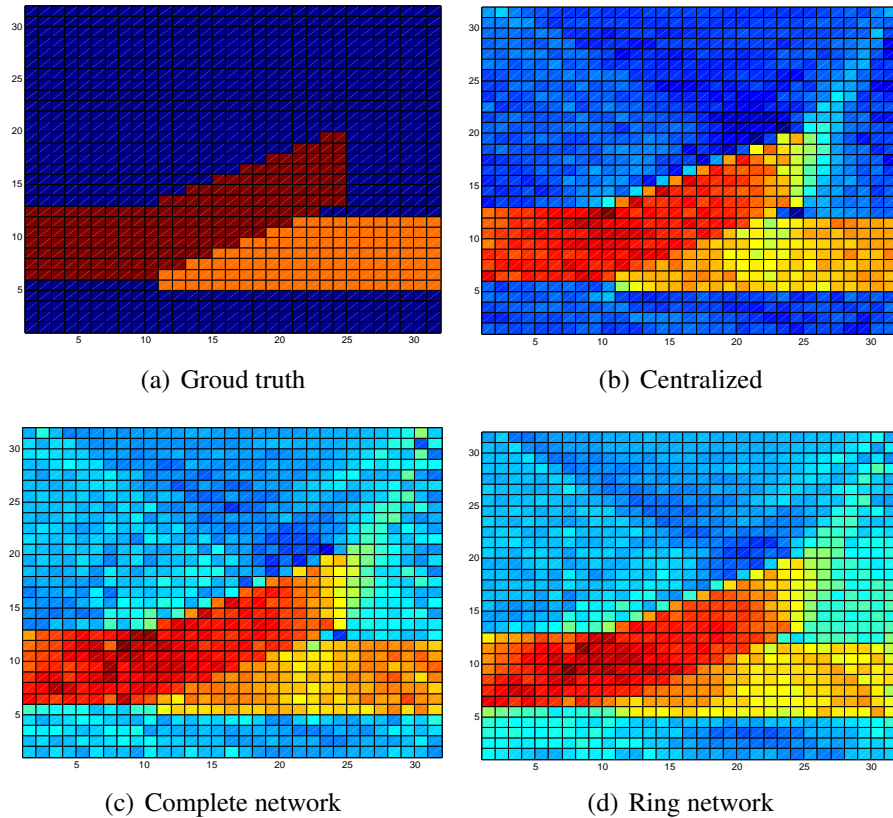
Figure (3.2) Seismic Tomography Comparison (noisy data). Centralized solution in (b) is obtained by running 50 iterations of BART. (c) is the tomography in Node 1 after 50 message communications assumed in a complete network. (d) is Node 1's tomography after 100 message communications assumed in a ring network.

no noise in the data. Note that in this scenario, the size of matrix $\mathbf{A}$ is 2048×256 and the size of each sub-matrix $\mathbf{A}_i$ is 64×256.

In Figure 3.1, we plot the error (measured by $\|\mathbf{s} - \mathbf{s}_t\|_2$, $\mathbf{s}_t$ refers to the ground truth) and residual (measured by $\|\mathbf{As} - \mathbf{t}\|_2$) for sensor 1 - sensor 8 (first 8 nodes) in the system. It shows that these 8 nodes finally reach consensus after around 25 iterations. It is worth noting that one iteration here means one message communication for every node. For the seismic tomography application investigated in this paper, our key concern is to find a communication efficient algorithm since in this problem communication for each node is much more expensive than computation within each node.

(a) Error

(b) Objective value

(c) sDSTA method

(d) aDSTA method
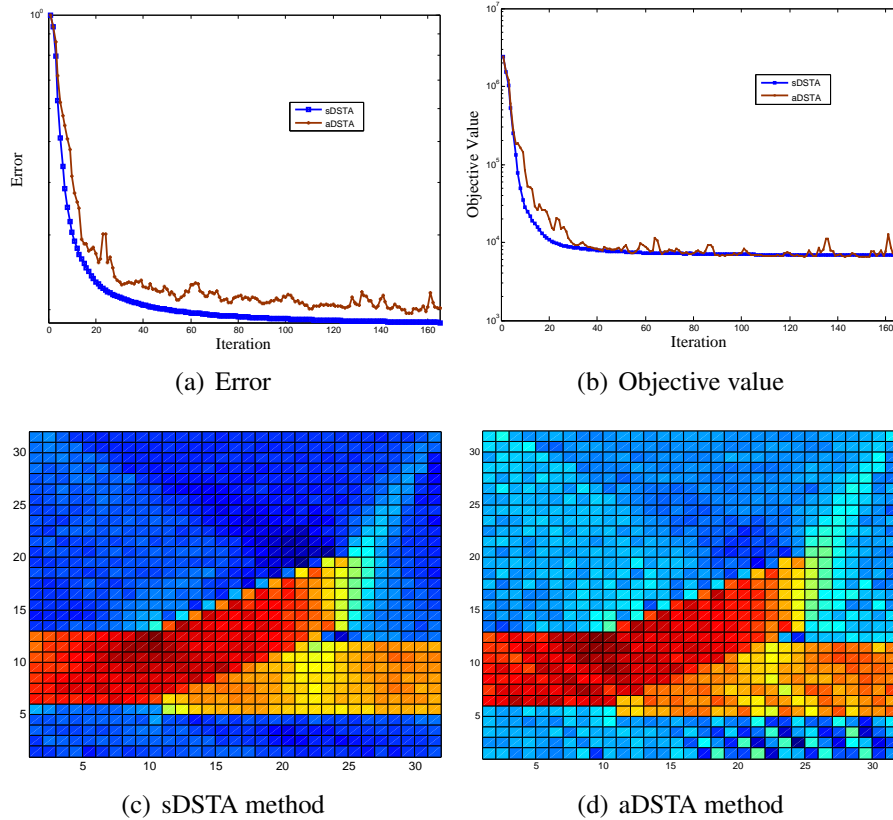
Figure (3.3) sDSTA vs aDSTA. (a) is the relative error plot. (b) shows the objective value curves. (c) & (d) are tomography results.

Tomography results of sDSTA Now we consider a larger data size model with {$N = 32$ (tomography resolution is 32×32), $n = 128$, $p = 64$}. A 5% gaussian noise is added into data vector **t**. The tomography results are depicted in Figure 3.2. Since noise is included, both centralized and sDSTA results are less accurate comparing to the noiseless case. Nevertheless, the outline of the fault zone (the brown part in (a)) is almost recovered. Another interesting point is that when connectivity ratio of the communication network is low (as the ring network in (d)), more communications among the neighbors are required in order to achieve similar results as the high connectivity ratio ones (e.g. (c)). This validates our claim in the last paragraph of Section 3.3.

Performance comparison: sDSTA vs aDSTA. We study a case with {$N = 32$, $n = 256$, $p = 128$}. Noise level is the same as the previous example. A random communication network is created and on average each sensor node has only 3 neighbors. This setting is to emulate the

situation in real that the sensor network is widely spread in the field to cover a large area and for each sensor node, there are very few number of nodes are within its communication range.

In Figure 3.3, we compare the performance of two proposed algorithms sDSTA and aDSTA. In particular, though the tomography is a bit worse, the convergence speed of aDSTA is shown to be close to sDSTA (in (a)-(b)). This observation is conform to the theoretical analysis in Section 5.2. In fact, if the application has a base station or a coordinator, and is not very real-time sensitive, synchronized-based sDSTA will meet the requirement. In other situations, aDSTA might be more suitable.

Remark 3 *We conclude that aDSTA can provide comparable tomography solution with sDSTA and has less communication and coordination.*

In the later context, we will focus on the performance evaluations of aDSTA.

(a) Error (b) Objective value
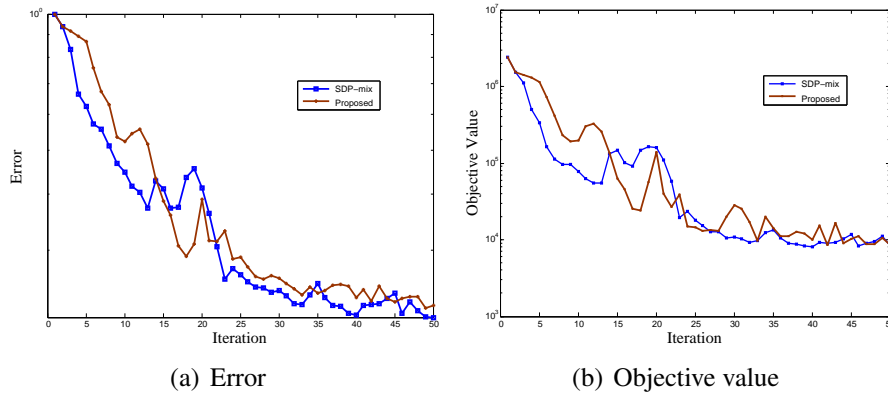
Figure (3.4) Performance of aDSTA with different strategies of choosing probability matrix $\mathbf{T}$.

Impact of probability matrix T on aDSTA It is also interesting to study the effect of using different selection rules for probability matrix $\mathbf{T}$. In Figure 3.4, "Proposed mix" refers to the rule suggested in Section 4.2 (eq. (3.21)) and "SDP-based mix" is the method described in (3.36). For each method, we run 5 realizations of aDSTA and average the solutions. We find that "SDP-based mix" is slightly better than "Proposed mix" in terms of convergence rate. Nevertheless, the distinction between them is not that obvious. In addition, solving (3.36) might need optimization

solver installed in the nodes. In short, if solving (3.36) is not available or wanted in the application, the simple "Proposed mix" can be a good alternative.

(a) Error   (b) Objective value
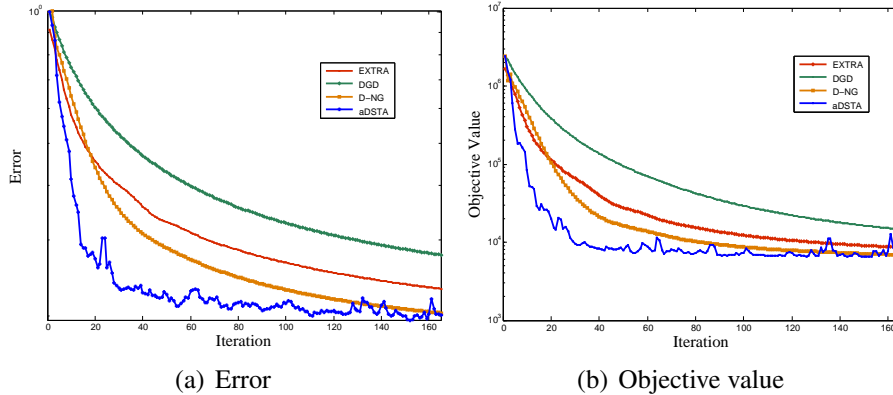
Figure (3.5) Convergence performance comparison: aDSTA, DGD, EXTRA, D-NG.

Compare aDSTA with other methods in the literature   We compare aDSTA with three recently developed general distributed optimization algorithms: Decentralized Gradient Descent (DGD) [94], EXTRA [95] and D-NG method [96]. All of these three methods are tested using hand-tuned optimal parameters, respectively. Figure 3.5 demonstrates that aDSTA (blue curve) significantly outperforms the other methods. Recall that in Figure 3.3(a)-3.3(b), sDSTA is very close to aDSTA, which insinuates that sDSTA is also faster than DGD, EXTRA and D-NG. This comparison conveys a message that the proposed sDSTA and aDSTA algorithms might be more suitable for our communication-sensitive distributed seismic tomography.

Synthetic Data (3-D Model)   We generate a synthetic 3-D model data set and visualize our solutions in Fig 3.6 and 3.7. The synthetic model consists of a magma chamber (low velocity area) in a 10 km^3 cube. 100 stations are randomly distributed on top of the cube and form a network. To construct the matrix **A** and vector **b**, 650 events are generated and we compute the travel times from every event to each node based on the ground truth, and send the event location and travel time to corresponding node with white Gaussian noise.

Layer 56 of 128 layers along Y
Resolution: 128x128

Layer 14 of 32 layers along Y
Resolution: 32x32

Layer 14 of 32 layers along Y
Resolution: 32x32

(a) Ground truth   (b) Centralized   (c) aDSTA
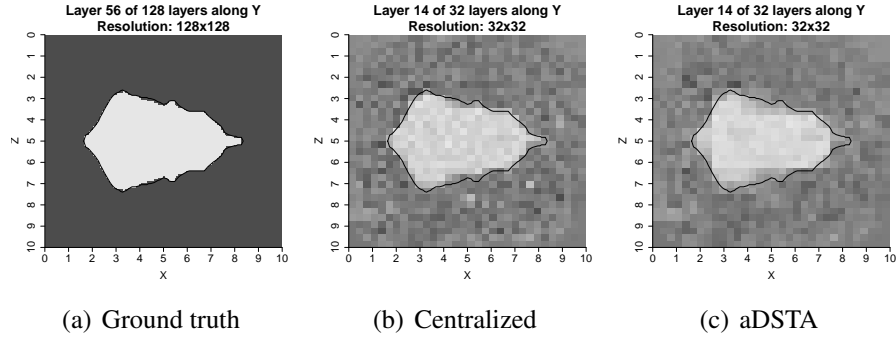
Figure (3.6) Vertical slices of tomography model. The ground truth in (a) is generated by simulating seismic data on resolution 128^3 and 650 events are used. Centralized and aDSTA methods are simulated with resolution 32^3 and 400 events.

Layer 14 of 32 layers along Y
Resolution: 32x32

Layer 14 of 32 layers along Y
Resolution: 32x32

Layer 14 of 32 layers along Y
Resolution: 32x32

(a) No packet loss   (b) 10% packet loss   (c) 30% packet loss
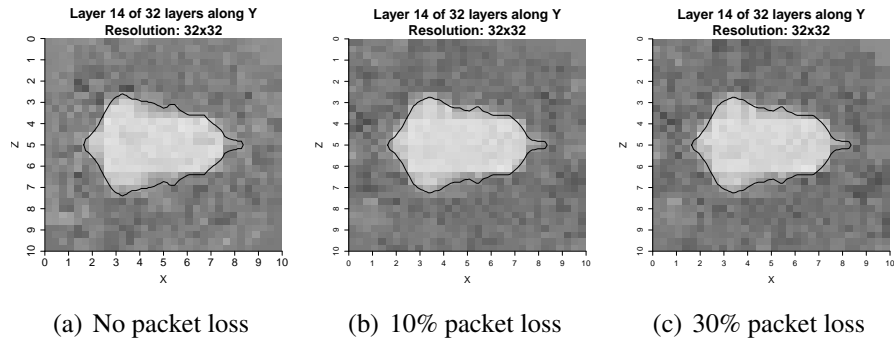
Figure (3.7) Effect of packet loss in aDSTA.

Comparison of tomography results   Figure 3.6 demonstrates that aDSTA is close to centralized method in tomography quality. Another interesting point is that aDSTA almost reconstructs the surface of magma even in a much lower resolution comparing to the ground truth.

Data loss-tolerance of aDSTA   We evaluate the robustness of aDSTA in Figure 3.7. Two packet loss ratios 10% and 30% are tested in the emulator. We can find that even for 30% packet loss ratio, the distinction between the result without packet loss is relatively small. This validates the robustness of aDSTA in dealing with severe packet loss situations.

Real Data (3-D Model)   To study the performance of the two proposed algorithms in realistic scenarios, we use ten years (2001-2011) real seismic event data of Mount St. Helens in Washington, USA for the experiment. The data were collected from 78 stations and a 3-D velocity

model is used, which assumes the velocity in the volcano changes along *x*-axis, *y*-axis and depth. Notice that unlike synthetic data used in previous section, there is no ground truth in this real data scenario. In other words, the true velocity structure of volcano Mount St. Helens is currently unknown. Hence we focus on the comparison of the proposed methods with centralized processing scheme, which can be seen as a benchmark that *fully* utilize the data available.

(a) Centralized   (b) sDSTA   (c) aDSTA

(d) Centralized   (e) sDSTA   (f) aDSTA

(g) Centralized   (h) sDSTA   (i) aDSTA
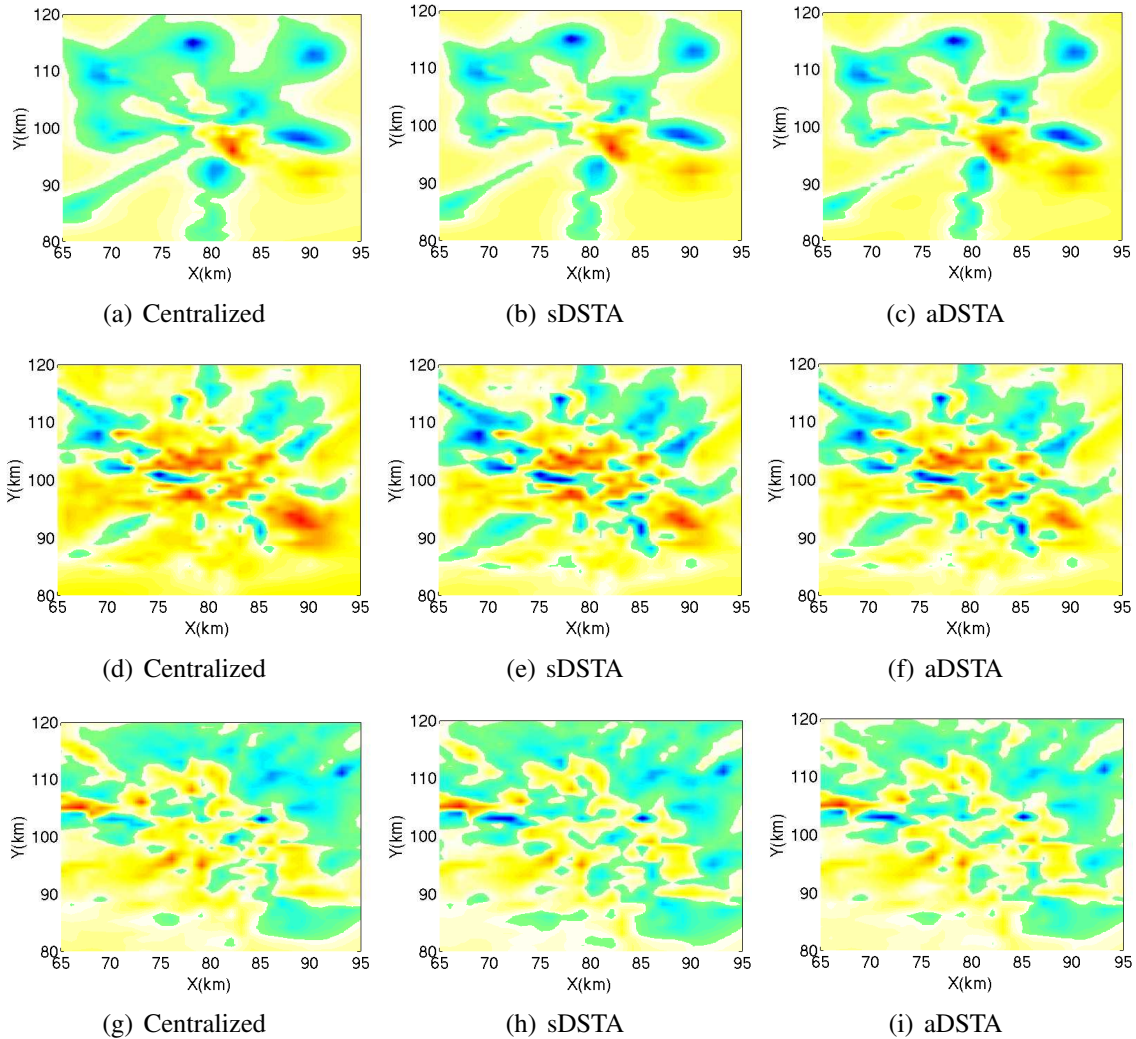
Figure (3.8) Real data tomography inversion results comparison. Figure (a)-(c) are results of layer depth 0.9 km. Figure (d)-(f) are results of layer depth 2.9 km. Figure (g)-(i) are results of layer depth 4.9 km.

Comparison of tomography results in real data Figure 3.8 illustrates vertical slices of tomography model with various depths. The range of *x*-axis is from 65 km to 95 km, and the range of *y*-axis is from 80 km to 120 km. The underlying resolution is $160 \times 200 \times 24$. The color in Figure 3.8 represents the relative velocity perturbation in specific location. More red means larger (negative) value of perturbation. More blue means larger (positive) value of perturbation. It is shown in Figure 3.8 that both sDSTA and aDSTA (at the solution of 300 iterations) can effectively invert the tomography model close to the benchmark (centralized algorithm) using real data.

(a) Total Communication Cost   (b) Communication Distribution
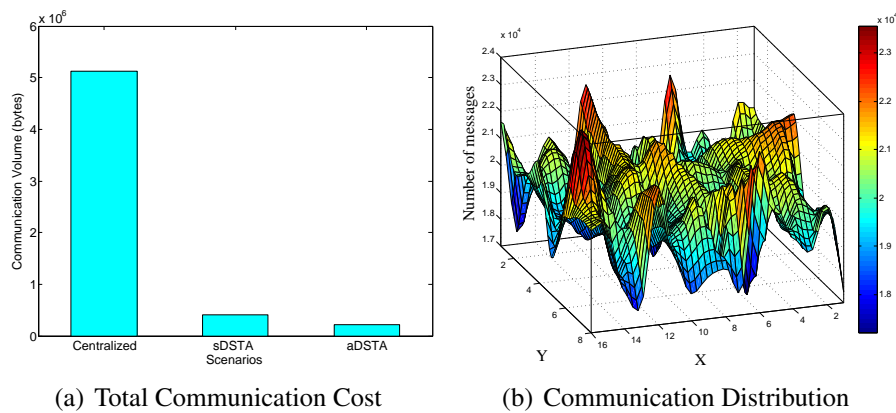
Figure (3.9) Communication cost comparison.

Communication cost evaluation in real data Figure 3.9 visualizes the communication cost characteristics of aDSTA. Figure 3.9(a) shows that sDSTA and aDSTA need much less amount of communication than the centralized method and aDSTA is more efficient than sDSTA. Figure 3.9(b) illustrates the communication distribution of aDSTA on each node with a heat map. We can see that, unlike synchronized sDSTA, in aDSTA some nodes stop communication earlier than others, which saves the bandwidth. Second, the cost disparities between several high cost nodes and the lowest one are around 18%, which implies that the communication load is still balanced over the network.

Summary The work proposed in this chapter opens a relatively under-exploited area where seismic tomography inversion is done over a sensor network in a distributed, decentralized fashion.

This paper presents novel designs for in-network travel-time seismic imaging in cyber-physical sensor systems under synchronous and asynchronous communication scenarios. The performance and features are verified with experiments in both synthetic and real data sets through network emulators. There are two potential benefits of using our proposed algorithms for seismic tomographic inversion problem. First, both sDSTA and aDSTA can be easily modified to accommodate problems with constraints on solution **s**. Second, they can deal with different regularizations (as long as it is convex). For instance, an efficient distributed algorithm can still be derived for the ℓ_1-norm regularized inversion problem in [97]. The merits of the proposed methods elucidate that they are promising solutions for real-time in-situ seismic tomography in the near future.

3.2   Fast Gradient-based Decentralized Optimization

Recent advances in convex optimization provide models and algorithms for decentralized Big Data computing problems, while minimizing the related computation and communication [98]. The problem in this paper has a general form as follows. Consider an undirected connected network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the node set and $\mathcal{E}$ is the edge set. The size of network is $m = |\mathcal{V}|$ and two nodes i, j are called neighbors if $(i, j) \in \mathcal{E}$. Now each node (sensor or agent) i privately holds an objective $F_i : \mathbb{R}^n \to \mathbb{R}$ which describes the data and acquisition process at the node. Here we assume F_i is proper, convex, and continuously differentiable, and its gradient ∇F_i has Lipschitz constant $L_i > 0$. The goal is to find the consensus solution $x \in \mathbb{R}^n$ of the minimization problem

$$\min_{x \in \mathbb{R}^n} \left\{ F(x) := \sum_{i=1}^{m} F_i(x) \right\}, \tag{3.37}$$

while each node can only communicate with its direct neighbors. The problem is called decentralized since the data is acquired and processed in a distributed network, and the nodes are required to collaboratively solve for a consensus solution. Our proposed framework is designed to leverage the computational power of all the sensor nodes. It performs in-network processing such that the "big data" stored in the nodes are processed locally. In addition, each node transmits its local estimate of the whole solution instead of the raw sensor data to its direct neighbors. Different from cen-

tralized approach, this kind of distributed and decentralized mechanism is much more scalable and fault-tolerant, and is a paradigm-shifting approach to solve a class of Big Data problems arising from distributed systems.

The aforementioned decentralized consensus problem attracts much attention recently, especially in distributed machine learning, multi-agent optimization, etc. For solving the problem in (3.74), several (sub)gradient-based methods have been proposed [343535738393435373839343537383934353738393435373839343537383 However, bounded (sub)gradients are usually assumed in analyzing the convergence results in most of the above algorithms. In addition, they cannot converge to an optimal solution of (3.74) if fixed step size is used [40]. In order to guarantee to converge to an optimal solution, Chen [38] and Jakovetic [39] thus adopt diminishing step sizes in their algorithms. More related algorithmic developments can be found in [41424344454647414243444546474142434445464741424344454647414243444546 Jakovetic [39] proposed a D-NC algorithm showing an outer-loop convergence rate of $O(1/k^2)$ in terms of objective error, leveraging Nesterov's acceleration, which is the best theoretical rate known so far. However, significant consensus iterations are required per outer-loop iteration. Without bounded gradient, [48] derives a correction method based on mixing matrix for regular decentralized gradient decent method and obtains $O(1/k)$ convergence rate without diminishing step sizes. In this work, we present a fast decentralized gradient descent method and prove that this new method can reach optimal convergence rate of $O(1/k^2)$ where k is the number of communication/iteration rounds.

3.2.1 Algorithm Design

Notation. Let $x \in \mathbb{R}^n$ be a column vector in problem (3.74), and $x^{(i)} \in \mathbb{R}^n$ be the local copy held privately by node i for every $i \in \mathcal{V}$. Without further remark, vectors are all column vectors. A vector $v = (v_1, \cdots, v_n)^T \in \mathbb{R}^n$ is sometimes written in short as $[v_i]$. Let $\mathbf{W} \in \mathbb{R}^{n \times n}$ be a symmetric positive semidefinite matrix and $\|v\|_{\mathbf{W}}$ is the (semi-)$\mathbf{W}$-norm of v. If $\mathbf{v} = (v^{(1)}, \cdots, v^{(m)})^T \in \mathbb{R}^{m \times n}$ where each $v^{(i)} \in \mathbb{R}^n$, then $\|\mathbf{v}\|_{\mathbf{W}} = \sqrt{\sum_{i=1}^m \|v^{(i)}\|_{\mathbf{W}}^2}$. Subscript k is outer iteration number, which is also the number of communication.

Proposed Algorithms and Interpretation The decentralized gradient descent (DGD) is a standard approach for solving (3.74). Recall that if regular gradient descent is applied at each node i to minimize its own objective F_i independent of other nodes, then a solution $x^{(i)} \in \text{argmin}_x F_i(x)$ can be severely biased due to the insufficient information in data at i. Moreover the solutions $x^{(i)}$ will not be all equal, and their average is not the solution to (3.74) in general. Instead, it is more sophisticated for each node i to request private copies $x^{(j)}$ from its immediate neighbors to gather more information and proceed with its next update of $x^{(i)}$. Motivated by this idea, the DGD iterates

$$x_{k+1}^{(i)} = \sum_{j=1}^{m} w_{ij} x_k^{(j)} - \alpha_k \nabla F_i(x_k^{(i)}) \tag{3.38}$$

at every node i for $k = 0, 1, 2, \cdots$. Here $x_k^{(i)}$ is the local copy held by node i at iteration k, α_k is the step size that satisfies $\alpha_k \leq 1/L$, where

$$L := \max_{1 \leq i \leq m} \{L_i\} \tag{3.39}$$

and L_i is the Lipschitz constant of ∇F_i. The prescribed symmetric mixing matrix $\mathbf{W} = [w_{ij}] \in \mathbb{R}^{m \times m}$ is nonnegative, $w_{ij} = 0$ if $(i, j) \notin \mathcal{E}$ and $i \neq j$, and $\mathbf{W}v = v$ if and only if $v \in \mathbb{R}^m$ is consensual, i.e., all its components are equal, due to the intuition that the mixing should not make changes if all $x^{(i)}$ are already identical. Therefore, each node i collects $x_k^{(j)}$ sent from its immediate neighbors j, mixes them with its own $x_k^{(i)}$ using weights w_{ij}, and performs a gradient descent at $x_k^{(i)}$ in iteration k.

To simplify notation, we define $\mathbf{x} := (x^{(1)}, \cdots, x^{(m)})^T \in \mathbb{R}^{m \times n}$, $\mathbf{F}(\mathbf{x}) = \sum_{i=1}^{m} F_i(x^{(i)}) \in \mathbb{R}$, column vector $\nabla F_i(x) \in \mathbb{R}^n$, and $\nabla \mathbf{F}(\mathbf{x}) = (\nabla F_1(x^{(1)}), \cdots, \nabla F_m(x^{(m)}))^T \in \mathbb{R}^{m \times n}$, then the decentralized minimization (3.74) is equivalent to a consensus optimization problem

$$\min_{\mathbf{x} \in \mathbb{R}^{m \times n}} \{\mathbf{F}(\mathbf{x}) : \mathbf{W}\mathbf{x} = \mathbf{x}\}, \tag{3.40}$$

where the constraint $\mathbf{W}\mathbf{x} = \mathbf{x}$ requires that a solution $\mathbf{x}^* = (x^{(1)}, \cdots, x^{(m)})^T$ needs to be consensual, i.e., $x^{(1)} = x^{(2)} = \cdots = x^{(m)} = x^*$, for some solution $x^* \in \mathbb{R}^n$ to (3.74), namely x^* satisfies

$\sum_{i=1}^{m} \nabla F_i(x^*) = 0$. Furthermore, the DGD algorithm (3.38) can be written as

$$\mathbf{x}_{k+1} = \mathbf{W}\mathbf{x}_k - \alpha_k \nabla \mathbf{F}(\mathbf{x}_k). \tag{3.41}$$

One can immediately observe that $\alpha_k \to 0$ is a necessary condition for the convergence of $\mathbf{x}_k$ to a solution $\mathbf{x}^*$ using (3.41), otherwise there will be $\nabla \mathbf{F}(\mathbf{x}^*) = (\nabla F_1(x^*), \cdots, \nabla F_m(x^*))^T = 0$ upon convergence $\mathbf{x}_k \to \mathbf{x}^*$, implying $\nabla F_i(x^*) = 0$ for all i, which is not true in general.

Algorithm 5 Fast Decentralized Gradient Descent (FDGD) with known Lipschitz constant L

Initialize $\mathbf{y}_0 = 0$ and arbitrary $\mathbf{x}_0$, set $\mathbf{x}_0^{\text{ag}} = \mathbf{x}_0$.
for $k = 0, 1, \cdots,$ **do**

$$\mathbf{y}_{k+1} = \mathbf{y}_k + (\widetilde{\mathbf{W}} - \mathbf{W})\mathbf{x}_k \tag{3.42}$$

$$\mathbf{x}_k^{\text{md}} = (1 - \theta_k)\mathbf{x}_k^{\text{ag}} + \theta_k \widetilde{\mathbf{W}}\mathbf{x}_k \tag{3.43}$$

$$\mathbf{x}_{k+1} = \widetilde{\mathbf{W}}\mathbf{x}_k - \mathbf{y}_{k+1} - \frac{1}{L\theta_k}\nabla \mathbf{F}(\mathbf{x}_k^{\text{md}}) \tag{3.44}$$

$$\mathbf{x}_{k+1}^{\text{ag}} = (1 - \theta_k)\mathbf{x}_k^{\text{ag}} + \theta_k \mathbf{x}_{k+1} \tag{3.45}$$

Output $\mathbf{x}_{k+1}^{\text{ag}}$

In this chapter, we develop a fast decentralized gradient descent method which does not require diminishing step size and the method is accelerated to reach an optimal $O(1/k^2)$ convergence rate for general convex differentiable functions F_i. We adopted the idea of Nesterov's optimal gradient method for centralized smooth optimization [99100101102991001011029910010110299100101102] and mixing matrix method in network gossip and consensus averaging algorithms [1034810348], and developed the following fast decentralized gradient algorithm (Algorithm 5) to solve the seismic tomography problem (3.40). In Algorithm 5, k is the (outer) iteration number which also indicates the number of rounds of communications. Every node privately holds its local copies $y^{(i)}, x^{(i),\text{md}}, x^{(i)}, x^{(i),\text{ag}}$ and F_i. At iteration k, each node i sends its current $x_k^{(i)}$ to all its immediate neighbors $\{j : (i, j) \in \mathcal{E}\}$ and receives $x_k^{(j)}$ from them (one round of communication), then performs weighted sums using w_{ij} and $\tilde{w}_{ij}$ (according to multiplications $\mathbf{W}\mathbf{x}_k$ and $\widetilde{\mathbf{W}}\mathbf{x}_k$), and updates

its $y^{(i)}$, $x^{(i),\text{md}}$, $x^{(i)}$ and $x^{(i),\text{ag}}$. The result $x^{(i),\text{ag}}$ is output as the final reconstruction.

Algorithm 6 Fast Decentralized Gradient Descent with Backtracking (FDGD-BT)

Initialize $\mathbf{y}_0 = 0$ and arbitrary $\mathbf{x}_0$, set $\mathbf{x}_0^{\text{ag}} = \mathbf{x}_0$.

for $k = 0, 1, \cdots,$ **do**

   1.

$$\mathbf{y}_{k+1} = \mathbf{y}_k + (\widetilde{\mathbf{W}} - \mathbf{W})\mathbf{x}_k \tag{3.46}$$

   2.

$$\mathbf{x}_k^{\text{md}} = (1 - \theta_k)\mathbf{x}_k^{\text{ag}} + \theta_k \widetilde{\mathbf{W}}\mathbf{x}_k \tag{3.47}$$

   3.   For each node i, find the smallest integer $j_k^{(i)} = 0, 1, 2, \cdots$ such that

$$F_i(x_{k+1}^{(i),\text{ag}}) \le F_i(x_k^{(i),\text{md}}) + \langle \nabla F_i(x_k^{(i),\text{md}}), x_{k+1}^{(i),\text{ag}} - x_k^{(i),\text{md}} \rangle + \frac{L_k^{(i)}\rho^{j_k^{(i)}}}{2}\|x_{k+1}^{(i),\text{ag}} - x_k^{(i),\text{md}}\|^2 \tag{3.48}$$

   where

$$\mathbf{x}_{k+1} = \widetilde{\mathbf{W}}\mathbf{x}_k - \mathbf{y}_{k+1} - \frac{1}{\theta_k}\mathbf{L}_k^{-1}\nabla\mathbf{F}(\mathbf{x}_k^{\text{md}}) \tag{3.49}$$

$$\mathbf{x}_{k+1}^{\text{ag}} = (1 - \theta_k)\mathbf{x}_k^{\text{ag}} + \theta_k\mathbf{x}_{k+1} \tag{3.50}$$

   Here $(x_{k+1}^{(i),\text{md}})^T$ is the i-th row of $\mathbf{x}_{k+1}^{\text{md}}$, $(x_{k+1}^{(i),\text{ag}})^T$ is the i-th row of $\mathbf{x}_{k+1}^{\text{ag}}$ and $\mathbf{L}_k = \text{diag}(L_k^{(i)}\rho^{j_k^{(i)}})$.

Output $\mathbf{x}_{k+1}^{\text{ag}}$

In Algorithm 5, the superscript "ag" stands for "aggregated", and "md" stands for "middle". Matrix $\widetilde{\mathbf{W}} = (\mathbf{I} + \mathbf{W})/2$ is a half-mixing matrix based on $\mathbf{W}$. A few remarks about this algorithm are in place. Firstly, Algorithm 5 is a first-order method since only $\nabla\mathbf{F}$ is required in each iteration, and hence the subproblem has low computation complexity. Secondly, we do not need to use diminishing step sizes which converge to 0 but still can ensure both of convergence and consensus. Thirdly, if $\theta_k = 1$ for all k, then Algorithm 5 reduces to a version very similar to regular decentralized gradient descent (3.41). However, by the choice of $\theta_k = O(1/k)$ as below, the change from input $\mathbf{x}_k^{\text{md}}$ to output $\mathbf{x}_{k+1}^{\text{ag}}$ is faster than that from $\mathbf{x}_k$ to $\mathbf{x}_{k+1}$. This implies that Algorithm 5 will converge faster than regular DGD. The last remark explains intuitively why the multi-step scheme defined in (3.43), (3.44), and (3.45) could potentially accelerate the convergence of Algorithm 1.

A practical issue with Algorithm 5 is that either the Lipschitz constant $L^{(i)}$ of ∇F_i or the maximum Lipschitz constant L defined in (3.39) may not be available to the nodes. To overcome

this issue, we design a backtracking strategy so that each node can search for its own $L_k^{(i)}$ at iteration k by gradually increasing its previous $L_k^{(i)}$ with multiples of $\rho > 1$ until (3.48) is satisfied. Note that such searching is guaranteed to finish in finitely many times for each iteration k, and the total number of searches is bounded by $\lceil \log_\rho(L^{(i)}/L_0^{(i)}) \rceil$ at each node i for the entire computation. The resulting algorithm with such backtracking strategy is presented in Algorithm 6.

Remark: From a sensor network point of view, the communication operation in Algorithm 1 & 2 is more costly than the computations within each k round (usually communication is more energy-consuming for sensors). Thus it is preferable to evaluate our algorithm performance in terms of the number of communication rounds to reach desirable results. Notice that in both Algorithm 1 & 2, only one communication is needed in one outer k round.

3.2.2 Convergence Analysis

In general we have the following convergence result for Algorithm 6 (Theorem 1). Before the analysis of Theorem 1, we would like to first prove the following Lemma.

Lemma 3 *Suppose $\hat{x}$ is a solution of $\min_{x}(\ell_{\mathbf{F}}(x; y) + \frac{L}{2}\|x - z\|^2)$ with some given y, z and L. Then we have*

$$\ell_{\mathbf{F}}(\hat{x}; y) + \frac{L}{2}\|\hat{x} - z\|^2 \leqslant \ell_{\mathbf{F}}(x; y) + \frac{L}{2}\|x - z\|^2 - \frac{L}{2}\|x - \hat{x}\|^2 \tag{3.51}$$

where $\ell_{\mathbf{F}}(x; y)$ is defined as $\ell_{\mathbf{F}}(x; y) = \mathbf{F}(y) + \nabla \mathbf{F}(y)^T(x - y)$.

Proof 6 *Let $g(x) = \ell_{\mathbf{F}}(x; y) + \frac{L}{2}\|x - z\|^2$, then we see that $g(x)$ is convex and $\nabla g(x) = \nabla \mathbf{F}(y) + L(x - z), \nabla g(y) = \nabla \mathbf{F}(y) + L(y - z)$. Hence $\|\nabla g(x) - \nabla g(y)\| = L\|x - y\|$, which implies that $\nabla g(x)$ has Lipschitz constant L. In consequence, we can have*

$$g(\hat{x}) \leqslant g(x) + \nabla g(x)^T(\hat{x} - x) + \frac{L}{2}\|x - \hat{x}\|$$

Since $\hat{x}$ is a minimizer of function $g(x)$, $\nabla g(x) = 0$. We then have $\nabla g(x)^T(\hat{x} - x) = (\nabla g(x)^T - \nabla g(\hat{x})^T)(\hat{x} - x) = -L\|x - \hat{x}\|^2$. Plugging this fact into the previous inequality yields (3.51).

Theorem 9 *Suppose* $\mathbf{x}^*$ *is a solution of* (3.41)*, and the parameters* $\{\theta_k\}$ *in Algorithm 5 satisfies*

$$\theta_0 = 1, \quad \theta_k \in (0, 1], \quad \frac{1}{\theta_k^2} \geq \frac{1 - \theta_{k+1}}{\theta_{k+1}^2}, \tag{3.52}$$

then the iteration $\{\mathbf{x}_k^{\mathrm{ag}}\}$ *satisfies*

$$\mathbf{F}(\mathbf{x}_k^{\mathrm{ag}}) - \mathbf{F}(\mathbf{x}^*) \leq \frac{L\theta_k^2}{2(1 - \theta_k)} \left(\|\mathbf{x}_0\|_{\widetilde{\mathbf{W}} - \mathbf{W}}^2 + \|\mathbf{x}^* - \mathbf{x}_0\|_{\widetilde{\mathbf{W}}}^2 \right) \tag{3.53}$$

and

$$\sum_{k=1}^{\infty} \|(\mathbf{I} - \frac{1}{m}\mathbf{1}\mathbf{1}^T)\mathbf{x}_k^{\mathrm{ag}}\|^2 < \infty, \tag{3.54}$$

where $\mathbf{I} \in \mathbb{R}^{m \times m}$ *is identity matrix and* $\mathbf{1} = (1, \cdots, 1)^T \in \mathbb{R}^m$.

Proof 7 *By the backtracking criterion in Algorithm 6 for each node i and definition of* $\mathbf{F}$*, we have*

$$\begin{aligned} \mathbf{F}(\mathbf{x}_{k+1}^{\mathrm{ag}}) \leq &\mathbf{F}(\mathbf{x}_k^{\mathrm{md}}) + \langle \nabla\mathbf{F}(\mathbf{x}_k^{\mathrm{md}}), \mathbf{x}_{k+1}^{\mathrm{ag}} - \mathbf{x}_k^{\mathrm{md}} \rangle \\ &+ \frac{1}{2}\|\mathbf{x}_{k+1}^{\mathrm{ag}} - \mathbf{x}_k^{\mathrm{md}}\|^2 \end{aligned} \tag{3.55}$$

By the definition of $\ell_{\mathbf{F}}$ *we have*

$$\begin{aligned} \mathbf{F}(\mathbf{x}_{k+1}^{\mathrm{ag}}) &\leq \ell_{\mathbf{F}}(\mathbf{x}_{k+1}^{\mathrm{ag}}; \mathbf{x}_k^{\mathrm{md}}) + \frac{1}{2}\|\mathbf{x}_{k+1}^{\mathrm{ag}} - \mathbf{x}_k^{\mathrm{md}}\|^2 \\ &= (1 - \theta_k)\ell_{\mathbf{F}}(\mathbf{x}_k^{\mathrm{ag}}; \mathbf{x}_k^{\mathrm{md}}) + \theta_k\ell_{\mathbf{F}}(\mathbf{x}_{k+1}; \mathbf{x}_k^{\mathrm{md}}) \\ &+ \frac{\theta_k^2}{2}\|\mathbf{x}_{k+1} - \widetilde{\mathbf{W}}\mathbf{x}_k\|^2 \\ &\leq \theta_k\left(\ell_{\mathbf{F}}(\mathbf{x}_{k+1}; \mathbf{x}_k^{\mathrm{md}}) + \frac{\theta_k}{2}\|\mathbf{x}_{k+1} - \widetilde{\mathbf{W}}\mathbf{x}_k\|^2\right) \\ &+ (1 - \theta_k)\mathbf{F}(\mathbf{x}_k^{\mathrm{ag}}) \end{aligned} \tag{3.56}$$

Recall the update of $\mathbf{x}_{k+1}$ implies the optimality condition

$$\mathbf{x}_{k+1} = \arg\min_{\mathbf{x}} \Big(\ell_{\mathbf{F}}(\mathbf{x}_{k+1}; \mathbf{x}_k^{\mathrm{md}}) + \langle \mathbf{y}_{k+1}, \mathbf{x}_{k+1} \rangle \\ + \frac{\theta_k}{2} \|\mathbf{x}_{k+1} - \widetilde{\mathbf{W}}\mathbf{x}_k\|^2 \Big) \tag{3.57}$$

Hence by Lemma 1 we have

$$\ell_{\mathbf{F}}(\mathbf{x}_{k+1}; \mathbf{x}_k^{\mathrm{md}}) + \langle \mathbf{y}_{k+1}, \mathbf{x}_{k+1} \rangle + \frac{\theta_k}{2} \|\mathbf{x}_{k+1} - \widetilde{\mathbf{W}}\mathbf{x}_k\|^2 \\ \leq \ell_{\mathbf{F}}(\mathbf{x}; \mathbf{x}_k^{\mathrm{md}}) + \langle \mathbf{y}_{k+1}, \mathbf{x} \rangle + \frac{\theta_k}{2} \|\mathbf{x} - \widetilde{\mathbf{W}}\mathbf{x}_k\|^2 \\ - \frac{\theta_k}{2} \|\mathbf{x} - \mathbf{x}_{k+1}\|^2 \tag{3.58}$$

Substitute this back into

$$\mathbf{F}(\mathbf{x}_{k+1}^{\mathrm{ag}}) \leq \theta_k \Big(\ell_{\mathbf{F}}(\mathbf{x}; \mathbf{x}_k^{\mathrm{md}}) + \theta_k \langle \mathbf{y}_{k+1}, \mathbf{x} - \mathbf{x}_{k+1} \rangle \\ + \frac{\theta_k}{2} \|\mathbf{x} - \widetilde{\mathbf{W}}\mathbf{x}_k\|^2 - \frac{\theta_k}{2} \|\mathbf{x} - \mathbf{x}_{k+1}\|^2 \Big) + (1 - \theta_k)\mathbf{F}(\mathbf{x}_k^{\mathrm{ag}}) \tag{3.59}$$

Now we set $\mathbf{x}$ to any solution $\mathbf{x}^*$, subtract $\mathbf{F}(\mathbf{x}^*)$ and divide θ_k^2 on both sides to obtain

$$\frac{1}{\theta_k^2} \Big(\mathbf{F}(\mathbf{x}_{k+1}^{\mathrm{ag}}) - \mathbf{F}(\mathbf{x}) \Big) \leq \frac{1 - \theta_k}{\theta_k^2} \Big(\mathbf{F}(\mathbf{x}_k^{\mathrm{ag}}) - \mathbf{F}(\mathbf{x}) \Big) \\ - \frac{1}{\theta_k} \delta_{\mathbf{F}}(\mathbf{x}; \mathbf{x}_k^{\mathrm{md}}) + \langle \mathbf{y}_{k+1}, \mathbf{x} - \mathbf{x}_{k+1} \rangle + \frac{1}{2} \|\mathbf{x} - \widetilde{\mathbf{W}}\mathbf{x}_k\|^2 \\ - \frac{1}{2} \|\mathbf{x} - \mathbf{x}_{k+1}\|^2 \tag{3.60}$$

where $\delta_{\mathbf{F}}(\mathbf{x}^*; \mathbf{x}_k^{\mathrm{md}})$ is defined as $\delta_{\mathbf{F}}(\mathbf{x}^*; \mathbf{x}_k^{\mathrm{md}}) := \mathbf{F}(\mathbf{x}^*) - \ell_{\mathbf{F}}(\mathbf{x}^*; \mathbf{x}_k^{\mathrm{md}})(\geq 0), \forall \mathbf{x}^*.

From the update of (3.42) we can see that

$$\mathbf{y}_{k+1} = (\widetilde{\mathbf{W}} - \mathbf{W})\mathbf{s}_k \tag{3.61}$$

where

$$\mathbf{s}_k = \sum_{t=0}^{k} \mathbf{x}_t \tag{3.62}$$

for each $k = 1, 2, \cdots$. Furthermore, due to the fact that $\mathbf{x}^*$ is consensual and $(\widetilde{\mathbf{W}} - \mathbf{W})\mathbf{1} = \mathbf{0}$, we have

$$
\begin{aligned}
\langle \mathbf{y}_{k+1}, \mathbf{x}^* - \mathbf{x}_{k+1} \rangle &= \langle (\widetilde{\mathbf{W}} - \mathbf{W})\mathbf{s}_k, \mathbf{x}^* - \mathbf{x}_{k+1} \rangle \\
&= - \langle (\widetilde{\mathbf{W}} - \mathbf{W})\mathbf{s}_k, \mathbf{x}_{k+1} \rangle \\
&= \langle \mathbf{s}_k - \mathbf{s}_{k+1}, (\widetilde{\mathbf{W}} - \mathbf{W})\mathbf{s}_k \rangle \\
&= \frac{1}{2}\Big(\langle (\widetilde{\mathbf{W}} - \mathbf{W})\mathbf{s}_k, \mathbf{s}_k \rangle - \langle \mathbf{s}_{k+1}, (\widetilde{\mathbf{W}} - \mathbf{W})\mathbf{s}_{k+1} \rangle \\
&\quad + \langle \mathbf{s}_k - \mathbf{s}_{k+1}, (\widetilde{\mathbf{W}} - \mathbf{W})(\mathbf{s}_k - \mathbf{s}_{k+1}) \rangle \Big) \\
&= \frac{1}{2}\Big(\langle \mathbf{s}_k, (\widetilde{\mathbf{W}} - \mathbf{W})\mathbf{s}_k \rangle - \langle \mathbf{s}_{k+1}, (\widetilde{\mathbf{W}} - \mathbf{W})\mathbf{s}_{k+1} \rangle \\
&\quad + \langle \mathbf{x}_{k+1}, (\widetilde{\mathbf{W}} - \mathbf{W})\mathbf{x}_{k+1} \rangle \Big)
\end{aligned}
\tag{3.63}
$$

Note that here

$$
\begin{aligned}
&\langle \mathbf{x}_{k+1}, (\widetilde{\mathbf{W}} - \mathbf{W})\mathbf{x}_{k+1} \rangle - \|\mathbf{x}^* - \mathbf{x}_{k+1}\|^2 \\
&= \langle \mathbf{x}_{k+1}, (\widetilde{\mathbf{W}} - \mathbf{W})\mathbf{x}_{k+1} \rangle - \langle \mathbf{x}^* - \mathbf{x}_{k+1}, \mathbf{x}^* - \mathbf{x}_{k+1} \rangle \\
&= \langle \mathbf{x}^* - \mathbf{x}_{k+1}, (\widetilde{\mathbf{W}} - \mathbf{W})(\mathbf{x}^* - \mathbf{x}_{k+1}) \rangle \\
&\quad - \langle \mathbf{x}^* - \mathbf{x}_{k+1}, \mathbf{x}^* - \mathbf{x}_{k+1} \rangle \\
&= - \langle \mathbf{x}^* - \mathbf{x}_{k+1}, (\mathbf{I} + \mathbf{W} - \widetilde{\mathbf{W}})(\mathbf{x}^* - \mathbf{x}_{k+1}) \rangle \\
&= - \langle \mathbf{x}^* - \mathbf{x}_{k+1}, \widetilde{\mathbf{W}}(\mathbf{x}^* - \mathbf{x}_{k+1}) \rangle
\end{aligned}
\tag{3.64}
$$

Moreover, we know the consensual solution $\mathbf{x}^*$ satisfies $\widetilde{\mathbf{W}}\mathbf{x}^* = \mathbf{x}^*$, hence there is

$$
\begin{aligned}
\|\mathbf{x}^* - \widetilde{\mathbf{W}}\mathbf{x}_k\|^2 = \|\widetilde{\mathbf{W}}(\mathbf{x}^* - \mathbf{x}_k)\|^2 &= \langle \mathbf{x}^* - \mathbf{x}_k, \widetilde{\mathbf{W}}^2(\mathbf{x}^* - \mathbf{x}_k) \rangle \\
&= \langle \mathbf{x}^* - \mathbf{x}_k, \widetilde{\mathbf{W}}(\mathbf{x}^* - \mathbf{x}_k) \rangle - \langle \mathbf{x}^* - \mathbf{x}_k, (\widetilde{\mathbf{W}} - \widetilde{\mathbf{W}}^2)(\mathbf{x}^* - \mathbf{x}_k) \rangle
\end{aligned}
\tag{3.65}
$$

Substitute the results we obtained above to, we have

$$\frac{1}{\theta_k^2}\left(\mathbf{F}(\mathbf{x}_{k+1}^{\text{ag}}) - \mathbf{F}(\mathbf{x}^*)\right) - \frac{1-\theta_k}{\theta_k^2}\left(\mathbf{F}(\mathbf{x}_k^{\text{ag}}) - \mathbf{F}(\mathbf{x}^*)\right)$$

$$+ \langle \mathbf{x}^* - \mathbf{x}_k, (\widetilde{\mathbf{W}} - \widetilde{\mathbf{W}}^2)(\mathbf{x}^* - \mathbf{x}_k) \rangle$$

$$\leq \frac{1}{2}\left(\langle \mathbf{s}_k, (\widetilde{\mathbf{W}} - \mathbf{W})\mathbf{s}_k \rangle - \langle \mathbf{s}_{k+1}, (\widetilde{\mathbf{W}} - \mathbf{W})\mathbf{s}_{k+1} \rangle\right)$$

$$+ \frac{1}{2}\Big(\langle \mathbf{x}^* - \mathbf{x}_k, (\widetilde{\mathbf{W}} - \mathbf{W})(\mathbf{x}^* - \mathbf{x}_k) - \langle \mathbf{x}^* - \mathbf{x}_{k+1},$$

$$(\widetilde{\mathbf{W}} - \mathbf{W})(\mathbf{x}^* - \mathbf{x}_{k+1})\rangle\Big) \tag{3.66}$$

Due to the setting of θ_k, we take the sum of $k = 0, 1, \cdots, k$ and obtain

$$\frac{1-\theta_{k+1}}{\theta_{k+1}^2}\left(\mathbf{F}(\mathbf{x}_{k+1}^{\text{ag}}) - \mathbf{F}(\mathbf{x}^*)\right) + \sum_{t=0}^{k}\langle \mathbf{x}^* - \mathbf{x}_t,$$

$$(\widetilde{\mathbf{W}} - \widetilde{\mathbf{W}}^2)(\mathbf{x}^* - \mathbf{x}_t)\rangle$$

$$\leq \frac{1}{2}\left(\langle \mathbf{s}_0, (\widetilde{\mathbf{W}} - \mathbf{W})\mathbf{s}_0 \rangle - \langle \mathbf{s}_{k+1}, (\widetilde{\mathbf{W}} - \mathbf{W})\mathbf{s}_{k+1} \rangle\right)$$

$$+ \frac{1}{2}\Big(\langle \mathbf{x}^* - \mathbf{x}_0, (\widetilde{\mathbf{W}} - \mathbf{W})(\mathbf{x}^* - \mathbf{x}_0) - \langle \mathbf{x}^* - \mathbf{x}_{k+1},$$

$$(\widetilde{\mathbf{W}} - \mathbf{W})(\mathbf{x}^* - \mathbf{x}_{k+1})\rangle\Big) \tag{3.67}$$

Note that $\widetilde{\mathbf{W}} \geq \mathbf{W}$ and that $\mathbf{F} > -\infty$, we obtain

$$-\infty < \frac{1-\theta_{k+1}}{\theta_{k+1}^2}\left(\mathbf{F}(\mathbf{x}_{k+1}^{\text{ag}}) - \mathbf{F}(\mathbf{x}^*)\right)$$

$$+ \sum_{t=0}^{k}\langle \mathbf{x}^* - \mathbf{x}_t, (\widetilde{\mathbf{W}} - \widetilde{\mathbf{W}}^2)(\mathbf{x}^* - \mathbf{x}_t)\rangle \tag{3.68}$$

$$\leq \frac{1}{2}\left(\langle \mathbf{s}_0, (\widetilde{\mathbf{W}} - \mathbf{W})\mathbf{s}_0 \rangle - \langle \mathbf{x}^* - \mathbf{x}_0, (\widetilde{\mathbf{W}} - \mathbf{W})(\mathbf{x}^* - \mathbf{x}_0)\rangle\right)$$

Since $\widetilde{\mathbf{W}} \geq \widetilde{\mathbf{W}}^2$, this inequality implies

$$\sum_{t=0}^{k}\langle \mathbf{x}^* - \mathbf{x}_t, (\widetilde{\mathbf{W}} - \widetilde{\mathbf{W}}^2)(\mathbf{x}^* - \mathbf{x}_t)\rangle < \infty$$

and that

$$\mathbf{F}(\mathbf{x}_{k+1}^{\text{ag}}) - \mathbf{F}(\mathbf{x}^*) \le \frac{\theta_{k+1}^2}{2(1-\theta_k)}\Big(\langle \mathbf{s}_0, (\widetilde{\mathbf{W}} - \mathbf{W})\mathbf{s}_0 \rangle$$
$$- \langle \mathbf{x}^* - \mathbf{x}_0, (\widetilde{\mathbf{W}} - \mathbf{W})(\mathbf{x}^* - \mathbf{x}_0) \rangle \Big)$$

Since there is

$$\widetilde{\mathbf{W}} - \widetilde{\mathbf{W}}^2 = \widetilde{\mathbf{W}}(\mathbf{I} - \widetilde{\mathbf{W}}) = \frac{\mathbf{I} + \mathbf{W}}{2} \cdot \frac{\mathbf{I} - \mathbf{W}}{2} = \frac{1}{4}(\mathbf{I} - \mathbf{W}^2) \tag{3.69}$$

the first inequality implies

$$\frac{1}{4}\sum_{t=0}^{k}\langle (\mathbf{I} - \mathbf{W}^2)\mathbf{x}_t, \mathbf{x}_t \rangle$$
$$= \sum_{t=0}^{k}\langle \mathbf{x}^* - \mathbf{x}_t, (\widetilde{\mathbf{W}} - \widetilde{\mathbf{W}}^2)(\mathbf{x}^* - \mathbf{x}_t) \rangle < \infty \tag{3.70}$$

We decompose each $\mathbf{x}_t$ into two parts, namely $\mathbf{1}\mathbf{1}^T\mathbf{x}_t$ and $(\mathbf{I} - \mathbf{1}\mathbf{1}^T)\mathbf{x}_t$, then the first part is in $\text{Null}(\mathbf{I} - \mathbf{W})$ and hence the inequality above implies

$$\frac{1}{4}(1 - \mu(\mathbf{W})^2)\sum_{t=0}^{k}\|(\mathbf{I} - \mathbf{1}\mathbf{1}^T)\mathbf{x}_t\|^2$$
$$\le \frac{1}{4}\sum_{t=0}^{k}\langle (\mathbf{I} - \mathbf{W}^2)\mathbf{x}_t, \mathbf{x}_t \rangle < \infty \tag{3.71}$$

where $\mu(\mathbf{W})$ denotes the second largest singular value of matrix $\mathbf{W}$ (since $\mathbf{W}$ is symmetric stochastic matrix, $\mu(\mathbf{W}) < 1$). The above fact means that the nonconsensual part of $\mathbf{x}_t$ is suppressed to 0. Since

$$\mathbf{x}_k^{\text{ag}} = \frac{\sum_{t=1}^{k} t\mathbf{x}_t}{\sum_{t=1}^{k} t}, \quad \text{for } k = 1, 2, 3, \ldots \tag{3.72}$$

we know $\mathbf{x}_k^{\text{ag}}$ tends to be consensual as well.

Due to the setting of θ_k, we can readily show that

$$\theta_k \geq \frac{2}{k+2}, \quad and \quad \frac{\theta_k^2}{1-\theta_k} \leq \frac{2}{k(k+2)} \tag{3.73}$$

for all $k = 1, 2, \cdots$, by induction. This implies that

$$\mathbf{F}(\mathbf{x}_{k+1}^{\text{ag}}) - \mathbf{F}(\mathbf{x}^*) \leq \frac{1}{k(k+2)}\Big(\langle \mathbf{s}_0, (\widetilde{\mathbf{W}} - \mathbf{W})\mathbf{s}_0\rangle$$
$$- \langle \mathbf{x}^* - \mathbf{x}_0, (\widetilde{\mathbf{W}} - \mathbf{W})(\mathbf{x}^* - \mathbf{x}_0)\rangle\Big)$$

(a)   (b)
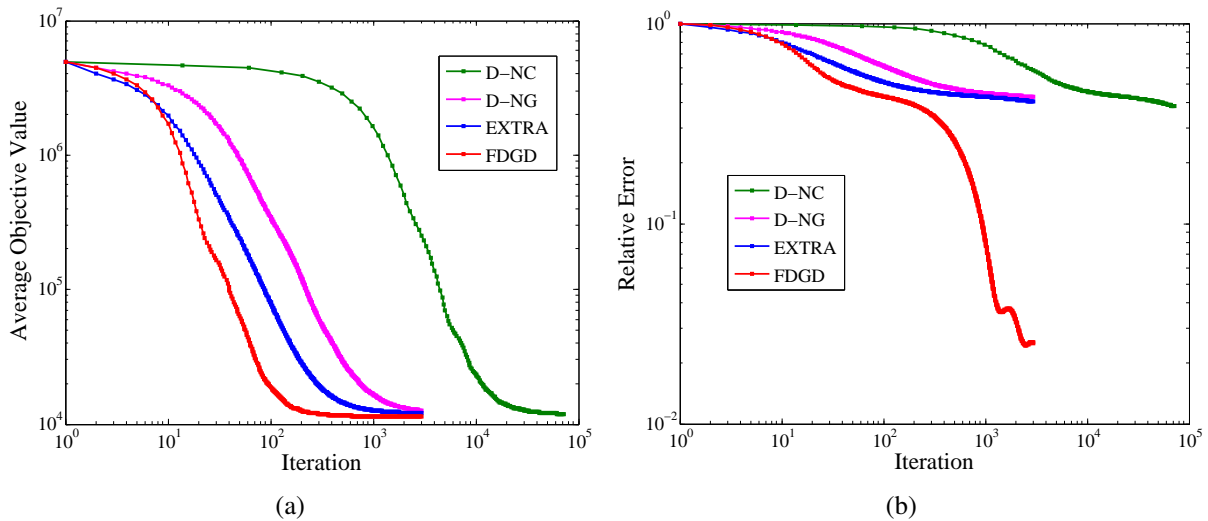
Figure (3.10) FDGD convergence behavior in 2D synthetic seismic data set. (a) is the plot of average objective value comparing FDGD with EXTRA, D-NG and D-NC methods. (b) is the relative error comparison plot.

Table (3.1) Summary of data set parameter settings

DATA SET	SIZE OF A	RESOLUTION	λ
SYNTHETIC 2D	16,384x4,096	64x64	1.0
SYNTHETIC 3D	40,000x32,768	32x32x32	10^{-4}
REAL DATA 3D	18,161x768,000	160x200x24	1.0

Figure (3.11) Convergence behavior comparison of FDGD and FDGD-BT in 2D synthetic data set. (a) and (b) depict the FDGD and FDGD with backtracking line search implementation in terms of average objective value and relative error, respectively.

Table (3.2) Network settings in the data sets

DATA SET	# OF NODES	AVG NODE DEGREE
SYNTHETIC 2D	32	3
SYNTHETIC 3D	100	3
REAL DATA 3D	11	2

3.2.3 Numerical Tests

In this section, we perform experiments on the seismic tomography problem using a regularized least squares model: $\min_x \ \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_2^2$, where λ is the regularization parameter. Simulations are performed on three different datasets: 2D synthetic dataset, 3D synthetic dataset and 3D real seismic tomography datasets. All methods are implemented in MATLAB, and experiments are performed on a PC with an Intel i5-3.0G HZ CPU and 8GB memory. In this experiment, we compare

1. Three recent decentralized methods: EXTRA [48], D-NG & D-NC [39] with our proposed FDGD algorithm.

2. FDGD and FDGD with backtracking line search.

(a) (b) (c) (d)
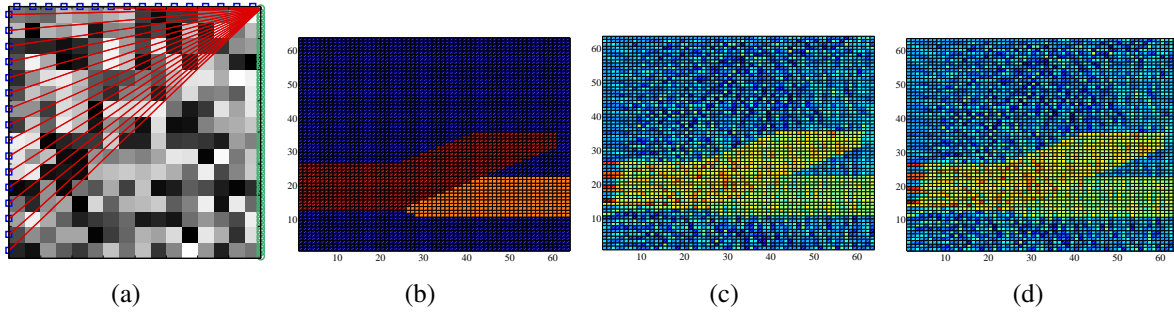
Figure (3.12) FDGD tomography results of 2D synthetic data set. (a) describes the 2D seismic model we used. (b) shows the ground truth of original seismic image. (c)-(d) exhibit the tomography results using centralized solution and FDGD, respectively.

(a) (b)
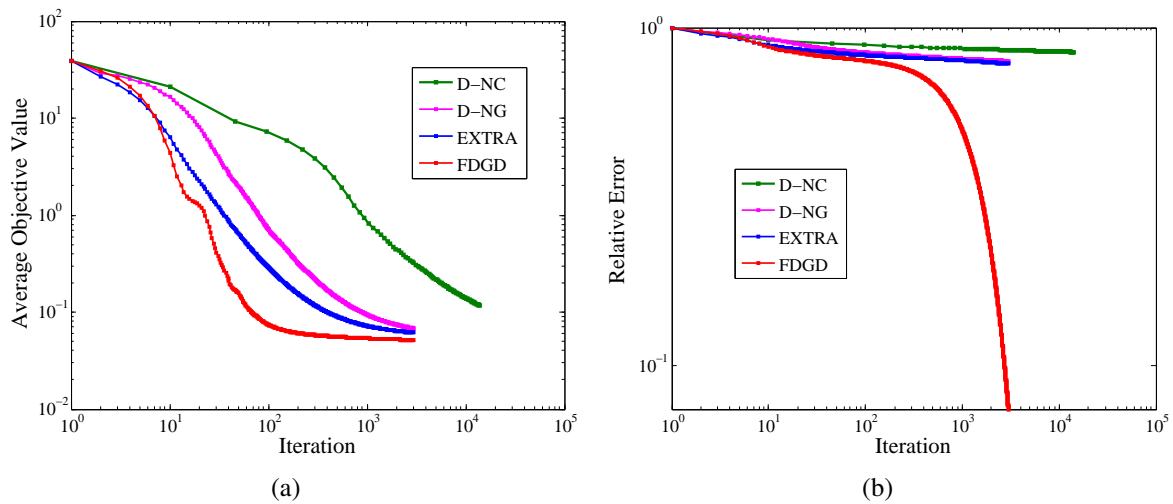
Figure (3.13) Comparison of convergence performance in 3D synthetic data set. (a)-(b) are comparing FDGD with EXTRA, D-NG and D-NC methods.

We plot the results of average objective value ($\frac{1}{m}\sum_{i=1}^{m} f(x_i^k)$), relative error ($\sqrt{\frac{\sum_{i=1}^{m}\|x_i^k - x^*\|_2^2}{\sum_{i=1}^{m}\|x_i^0 - x^*\|_2^2}}$) and tomography images for all the three data sets, where x^* is a pre-computed centralized solution.

In our simulations, the regularization parameter λ is fixed and the corresponding parameter λ_i for each node i is set to $1/p$, where p is the total number of nodes in the system. In each data set, the centralized solution of optimization problem $\min_x \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_2^2$ is obtained by LSQR method. The centralized solution is taken as our benchmark comparing to the decentralized methods tested. The matrix A and vector b is constructed by stacking the sub-matrices of all the nodes. The resolution means the number of blocks along the x, y and z-axis. The communication
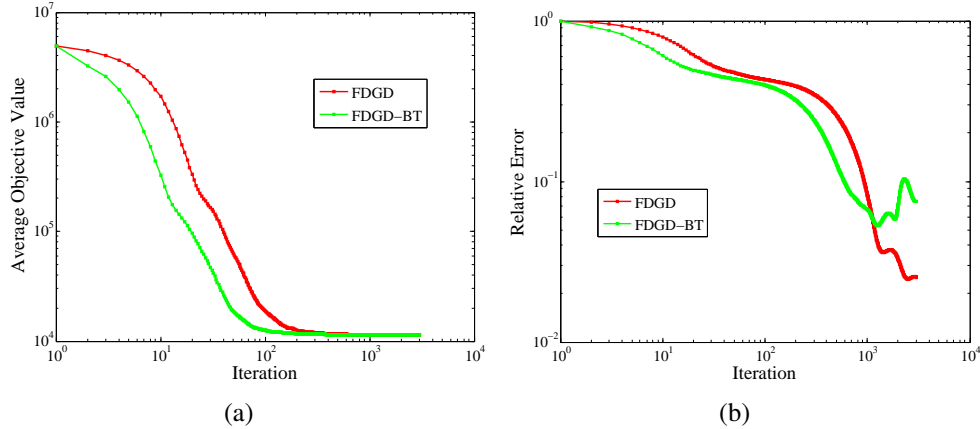
Figure (3.14) Convergence behavior comparison of FDGD and FDGD-BT in 3D synthetic data set. (a) and (b) depict the FDGD and FDGD with backtracking line search implementation in terms of average objective value and relative error, respectively.
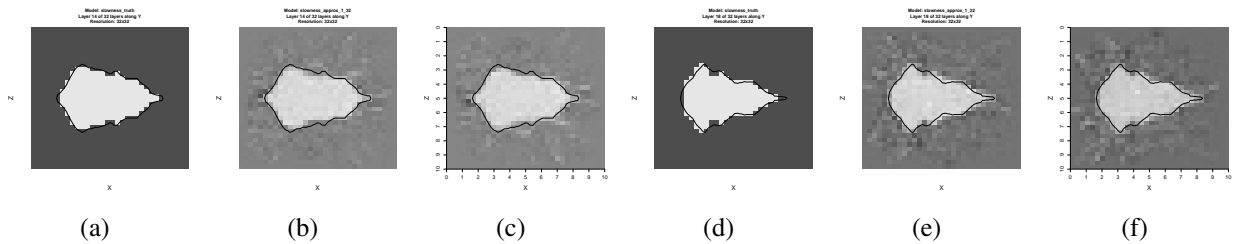
Figure (3.15) Vertical slices of 3D synthetic model tomography. Fig. (a)-(c) are results of layer 14 along *y*-axis and Fig. (d)-(f) are results of layer 18. Left-most column is the ground truth, the middle column shows the centralized solution and the right-most column contains the solution using our proposed FDGD algorithm.

network is generated randomly with certain number of average node degree. The parameters are described in table 3.1 and 3.2.

Synthetic Data (2D Model) The performance analysis here is based on the data set generated using code in [93]. We create a 2D seismic tomography test problem with a square domain, using sources located on the right boundary (green dots) and receivers (seismographs) scattered along the left and top boundary (blue squares). The rays are transmitted from each source to each receiver (red lines) (see Figure 3.12(a)). The experiment results are demonstrated in Figure 3.10-3.12.
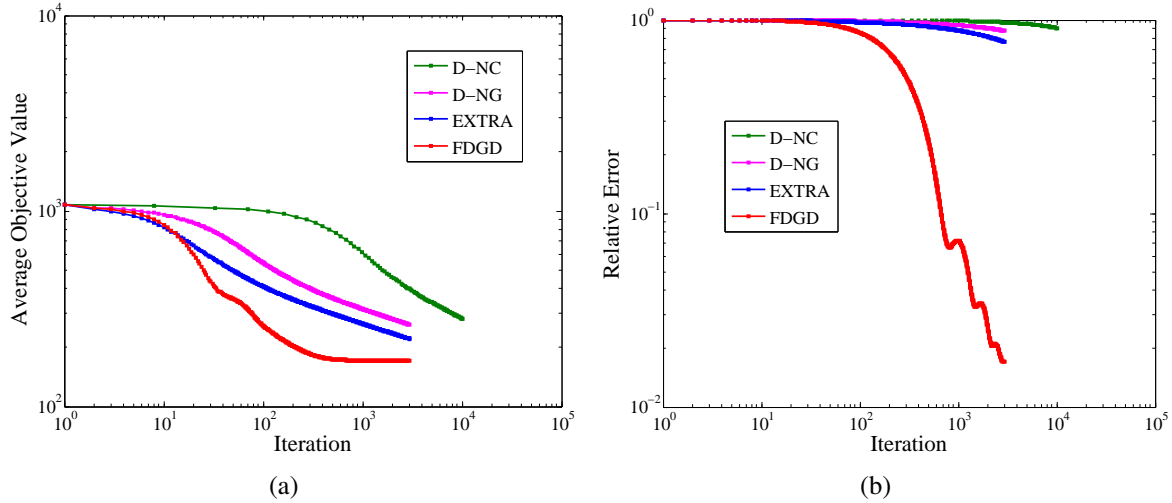
Figure (3.16) Real data tomography inversion results comparison. (a)-(b) are comparing FDGD with EXTRA, D-NG and D-NC methods. (c) and (d) show peformances of FDGD and FDGD with backtracking line search implementation. (e)-(f) describe the solutions of vertical slices of at depth 0.9 *km* (left:centralized, right:FDGD). (g)-(h) exhibit the tomography results at depth 4.9 *km*.

Synthetic Data (3D Model)   In this section, the evaluation of algorithm is illustrated by simulating seismic data on a synthetic model of resolution 32^3 consisting of a magma chamber (low velocity area) in a 10 km^3 cube. 100 stations are randomly distributed on top of the cube and form a network. To construct the matrix A and vector b, 400 events are generated and we compute the travel times from every event to each node based on the ground truth, and send the event location and travel time to corresponding node with white Gaussian noise. Figure 3.13-3.15 illustrate the experiment results in this data set.

Real-world Data (3D Model)   To study the performance of the two proposed algorithms in realistic scenarios, we use ten years (2001-2011) real seismic event data of Mount St. Helens in Washington, USA for the experiment. The data were collected from 78 stations and we construct them into 11 clusters and form a network based on the clusters. Notice that unlike synthetic data used in previous section, there is no ground truth in this real data scenario. Hence we focus on the comparison of the proposed methods with centralized processing scheme, which can be seen as a benchmark that *fully* utilize the data available. Results are shown in Figure 3.16-3.18.

Remark: Previous simulation results have demonstrated the superiority of proposed FDGD
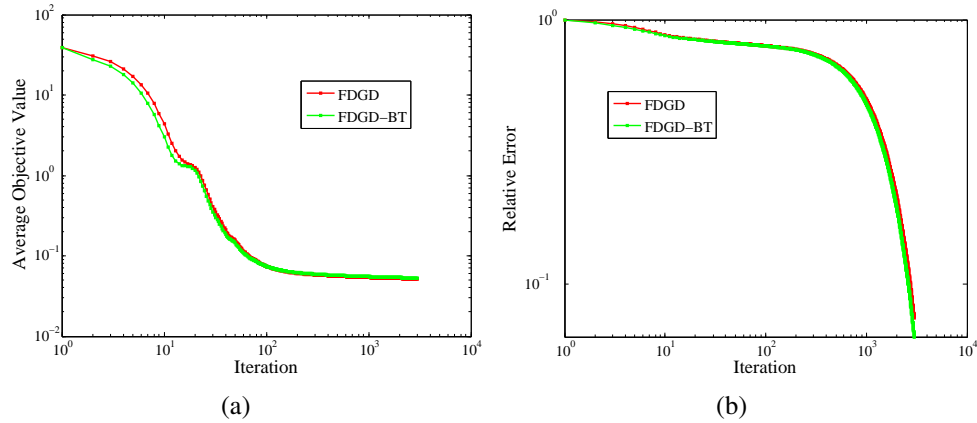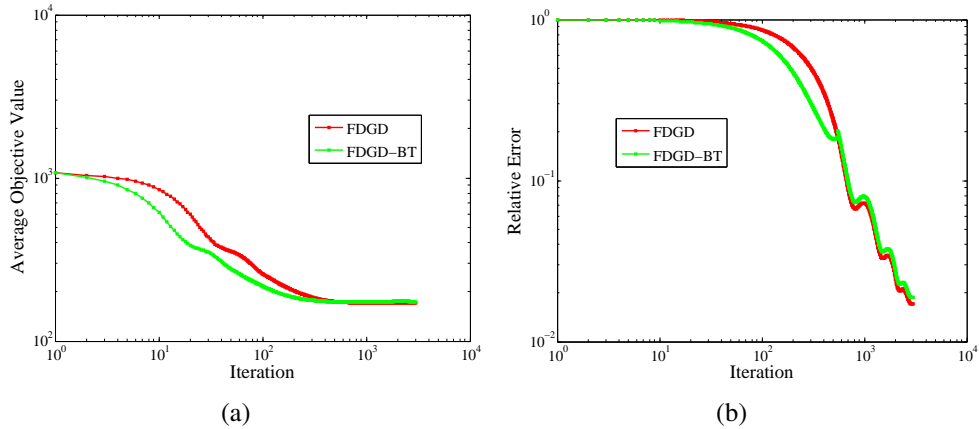
Figure (3.17) Convergence behavior comparison of FDGD and FDGD-BT in 3D real data set. (a) and (b) show performances of FDGD and FDGD with backtracking line search implementation.
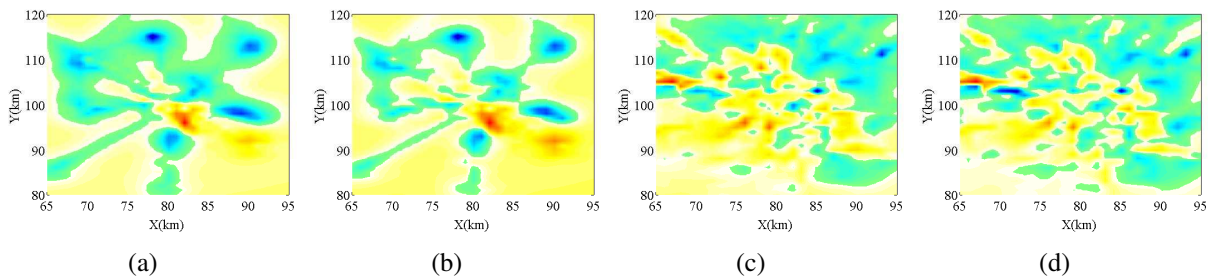
Figure (3.18) Seismic tomography comparison of the 3D real data set. (a)-(b) describe the solutions of vertical slices of at depth 0.9 *km* (left:centralized, right:FDGD). (c)-(d) exhibit the tomography results at depth 4.9 *km*. The range of *x*-axis is from 65 to 95 *km* and the *y*-axis is from 80 to 120 *km*. The color in the figure represents the relative velocity perturbation in specific location. More red means larger (negative) value of perturbation. More blue means larger (positive) value of perturbation

over other existing methods. In all the data sets, FDGD can obtain near "optimal" (the centralized approach) solution with reasonable number of communication rounds even in extremely low-connectivity networks. The performance of FDGD-BT is almost the same as FDGD implying we can still achieve similar results without knowing Lipschitz constant *L*. Please note that the value of regularization parameter λ also determines the convexity property of the objective function. We do observe linear convergence rate of EXTRA for strongly convex functions as claimed in [48]. However, we found that in the simulated synthetic data sets, smaller λ is better and more suitable for image recovery. In the real data case, since no "ground truth" is available, for simplicity, we

also choose $\lambda = 1$ in our experiments. In fact, $\lambda = 1$ is relative small comparing to the data fitting term such that the objective function is not quite "strongly convex". That explains why EXTRA does not show linear convergence in our results. To show an example of this scenario, we also perform the experiment on 2D synthetic data set with $\lambda = 20$ (see Fig. 3.19).
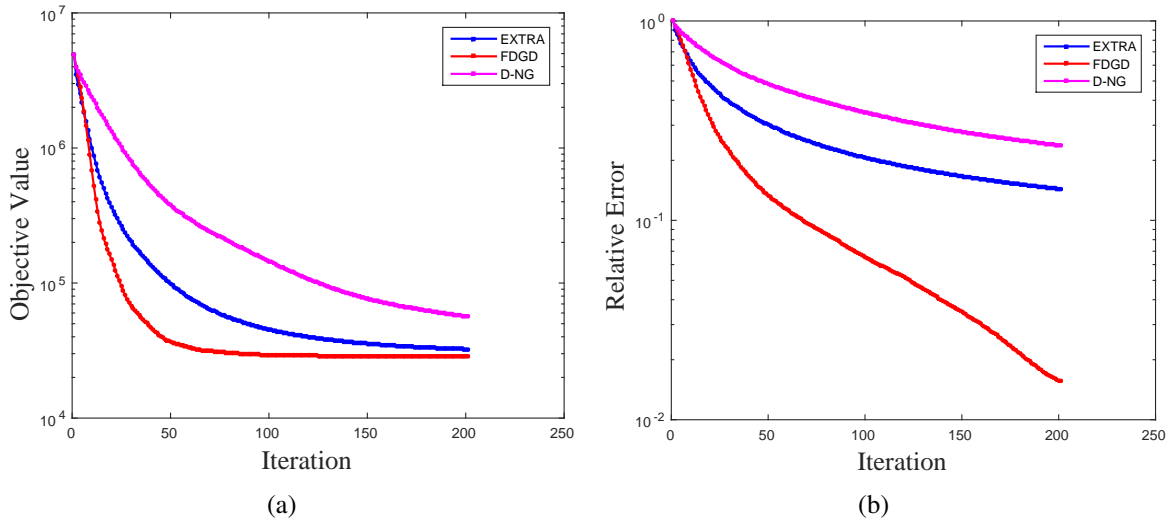
Figure (3.19) Convergence behavior comparison of 2D synthetic data set with regularization parameter $\lambda = 20$.

Summary Distributed and decentralized optimization is well suited to Big Data applications, and in particular to analytics in distributed architectures. In this chapter we developed a novel fast decentralized gradient descent method whose convergence does not require diminishing step sizes as in regular decentralized gradient descent methods, and prove that this new method can reach optimal convergence rate of $O(1/k^2)$ where k is the communication/iteration number. In the seismic tomography application, we conducted experiments on synthetic and real-world sensor network seismic data. The results exhibit that the proposed algorithms significantly outperform the current state-of-the-arts.

3.3 Asynchronous Broadcast-based Decentralized Algorithm

3.3.1 Motivation

First, in tough environment like the volcano, node failures can happen all the time. It is highly demanded to design an asynchronous algorithm such that each sensor node can determine its operation independently and it does not have to wait the slowest neighbor for updating. Second, when deploying wireless sensor systems into field, it is highly beneficial to save communication cost and energy by choosing broadcast communication in order to leverage the wireless medium. Asynchronous broadcast-based algorithm is most practical for seismic tomography problem.

A number of synchronous methods have been proposed to fully distributed (decentralized) consensus optimization problem [34], [35], [36], [37], [38], [96], [41], [42], [45], [46], [47], [95] and etc. To apply the aforementioned algorithms into sensor network, synchronization among neighboring nodes is required. Furthermore, for each node, it needs to wait for its slowest neighbor in order to perform update.

Distributed optimization methods for asynchronous models have been designed in [49], [50] and [43]. Wei [49] and Iutzeler [50] leverage ADMM for the computation part, and in each iteration, one node needs to randomly wake up one of its neighbors to exchange information. However, the communication schemes in these two works are based on unicast, which is much less preferable than broadcast communication, especially in real-world wireless sensor network scenario. Tsitsiklis [43] proposed an asynchronous model for distributed optimization, while in its model each node maintains a partial vector of the global variable. It is different from our goal of decentralized consensus such that each node contains an estimate of the global common interest.

The first broadcast-based distributed/decentralized consensus method was proposed in [104] for consensus average problem. However, the consensus solution generated can not guarantee to be the true average. Inspired by the push-sum algorithm in [105], several works have been done recently to overcome the issue [106], [107] and [108]. Iutzeler [106] applies the push-sum algorithm in [105] into the broadcast gossip model in [104]. The algorithm developed in [106] requires the node to broadcast a pair of variables (instead of only the solution variable) in the

communication stage. The method enjoys not only the similar convergence speed as the broadcast gossip in [104], but also the guarantee of converging to the true average. The broadcast-based works above are developed for consensus average problem while it is also essential to investigate problems with "real" objective functions.

Nedic [109] first filled this gap by considering general decentralized convex optimization under the asynchronous broadcast setting. It adopted the asynchronous broadcast model in [104] and developed a gradient-based update rule for its computation. In this work, we modify Nedic's algorithm [109] in order to fit the purpose of seismic tomography in sensor networks. From sensor network point of view, communication is more time and energy consuming than computation, thus it is more important to reduce the communication cost for this application. Note that there is a trade-off between the computational complexity and communication cost for decentralized optimization methods. Hence in order to obtain a more suitable platform for seismic imaging, we propose a scheme, which needs less communication rounds than Nedic's method by sacrificing its computational complexity (higher complexity than Nedic's algorithm but it is not the main concern here since the computation time in general is much less comparing to the communication time in seismic tomography).

Algorithm Settings and Assumptions The problem in this paper has a general form as follows. Consider an undirected connected network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the node set and $\mathcal{E}$ is the edge set. The size of network is $m = |\mathcal{V}|$ and two nodes i, j are called neighbors if $(i, j) \in \mathcal{E}$. Now each node (sensor or agent) i privately holds an objective $F_i : \mathbb{R}^n \to \mathbb{R}$ which describes the data and acquisition process at the node. The goal is to find the consensus solution $x \in \mathbb{R}^n$ of the minimization problem

$$\min_{x \in \mathbb{R}^n} \left\{ F(x) := \sum_{i=1}^{m} F_i(x) \right\}, \tag{3.74}$$

Setting 1: Each sensor node has its local clock that ticks at a user-customized Poisson rate for unit time, which is independently of the clocks of the other nodes.

Setting 2: Each node broadcasts its current estimate to its neighbors at each tick of its local clock.

Setting 3: During broadcasting, each sensor receives neighbor's information subject to link failure. For example, when node i broadcasts, its neighbor j will receive j's iterate with probability p_{ij}.

The asynchronous model adopted in this paper is similar to the one in [103] and [104]. The difference is that we consider link failure instead of assuming reliable link (setting 3) in this work.

Remark 4 *It is equivalent to consider a virtual global clock existing in the network for the algorithm analysis. Since the Poisson clock of each node (suppose rate = 1) is independent of each other, it is same as a global clock with Poisson rate m. We can then analyze the problem given that in each global iteration only one node broadcasts its value.*

Assumption 1: The network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is uniformly strongly connected.

Assumption 2: The solution set of (3.74) is nonempty. The private local function F_i, $i \in \mathcal{V}$ is convex.

Assumption 3: The (sub)gradient of function F_i is bounded such that $\|\nabla F_i\| \leq G$, where $G > 0$ is some positive number.

3.3.2 Algorithm Design

Notation. Let $x \in \mathbb{R}^n$ be a column vector in problem (3.74), and $x^{(i)} \in \mathbb{R}^n$ be the local copy held privately by node i for every $i \in \mathcal{V}$. Without further remark, vectors are all column vectors. Subscript k is outer iteration number, which is also the number of communication.

Proposed Algorithm   In this section, we first describe the asynchronous broadcast-based algorithm. Assume in iteration k (according to the virtual global Poisson clock), one node i_k is randomly chosen from $\mathcal{V}$ and broadcasts its value $x_k^{i_k}$ to a subset J_k of its neighbors $N(i_k)$. Then the nodes $i \in J_k$ performs the following computation using its own F_i:

$$x_{k+1}^i = \operatorname*{argmin}_{x \in \mathbb{R}^n} \left\{ \frac{1}{2\alpha_{i,k}} \|x - (\theta x_k^{i_k} + (1-\theta)x_k^i)\|^2 + F_i(x) \right\} \tag{3.75}$$

where $\theta \in (0, 1)$ (e.g. $\theta = \frac{1}{2}$) is the weight factor and F_i is the objective function at node i. For nodes $i \notin J_k$ and node i_k, there is no update performed.

The step size $\alpha_{i,k}$ is set to $1/\Gamma_k(i)$, where $\Gamma_k(i)$ is the number of updates performed at node i up to iteration k (same as the one in [109]).

To begin with, we set $J_k = N(i_k)$,, which indicates reliable broadcast communication. Later on we construct J_k such that it consists only nodes j where the broadcast from i_k successfully arrives at j with probability p_{ij}.

An optimal solution of (3.74) should be consensual and optimized at the same time. Hence two important measures must be considered for analyzing our algorithm. The measures should be versus iteration number k, which corresponds to the number of broadcast rounds in the entire network.

1. Optimal objective: it determines whether the objective function of the averaged solution reaches the optimal objective value.

$$\sum_{i=1}^{m} F_i(\bar{x}_k), \ \ where \ \ \bar{x}_k = (1/m) \sum_{i=1}^{m} x_k^i \tag{3.76}$$

2. Nodes consensus: it measures how close every node in the network reaches the consensus solution.

$$\sum_{i=1}^{m} \|x_k^i - \bar{x}_k\|^2 \tag{3.77}$$

Remark 5 *From a sensor network point of view, the communication operation is more costly than the computation within each round (usually communication is more energy-consuming for sensors). Thus it is preferable to evaluate our algorithm performance in terms of the number of communication rounds to reach desirable results.*

To summarize, our proposed algorithm can be expressed in a formal way:

1. **Initialization**: Each sensor node is equipped with a local Poisson clock (rate 1), which is independent of other clocks. Set the weight factor θ.

2. **Communication**: Assume the local clock of node i ticks (only one tick at the same time in the whole network). Node i broadcasts its estimate x^i to its neighbors. The neighbors receive the information with probability $p_{ij}, j \in N(i)$.

3. **Update**: The neighbors who receive node i's broadcast will perform updating as follows. Node i and the other nodes who do not receive node i's iterate keep unchanged.

$$x^j \leftarrow \operatorname*{argmin}_{x \in \mathbb{R}^n} \left\{ \frac{1}{2\alpha_j} \|x - (\theta x^i + (1 - \theta)x^j)\|^2 + F_j(x) \right\} \tag{3.78}$$

where α_j is the step size defined in previous.

4. **Repeat**: The network stays silent until a local clock ticks and repeats steps 2-3.

3.3.3 Algorithm Interpretation

Remember that the major goal and challenge of the seismic imaging problem is to reduce the total communication cost since we are highly constrained by the physical bandwidth and energy consumption. We realize that the in the decentralized consensus problem, a trade-off exists between the computational effort and the communication cost. That is, for any decentralized algorithm, if we prefer to have less communication rounds to achieve the desirable result, the computational complexity in each update round should be higher. It is reasonable to see that if the update stage can be done more completely and then it should provide a better estimate, which can speed up the convergence of the network after broadcasting. Another consideration is that the total time consists of the communication and the computation while the latter is much less than the former in general. Hence designing an algorithm, which is better in terms of communication cost, would be more meaningful in the application of seismic imaging.

Our proposed algorithm is similar to Nedic's in [109]. In below, we describe the differences and improvements of the proposed method over Nedic's. Nedic's update rule can be expressed as follows assuming nodes $i \in J_k$ receive the estimate $x_{k-1}^{i_k}$ (i_k is the index of the node selected at iteration k).

$$y_k^i = \theta x_{k-1}^{i_k} + (1 - \theta)x_{k-1}^i$$
$$x_k^i = y_k^i - \alpha_{i,k} \nabla F_i(y_k^i) \tag{3.79}$$

The first step in the last equation is the **consensus-like** step. It is a mechanism to align nodes estimates toward a common point. The second step is the **local gradient-based** equation. It is used to minimize its own objective function. To better analyze the algorithm, we equivalently convert (3.79) into its compact form:

$$
\begin{aligned}
x_{k+1}^i &= \underset{x \in \mathbb{R}^n}{\text{argmin}} \left\{ \frac{1}{2\alpha_{i,k}} \|x - y_k^i\|^2 + \langle \nabla F_i(y_k^i), x \rangle \right\} \\
&= \underset{x \in \mathbb{R}^n}{\text{argmin}} \left\{ \frac{1}{2\alpha_{i,k}} \|x - (\theta x_{k-1}^{I_k} + (1 - \theta)x_{k-1}^i)\|^2 + \langle \nabla F_i(y_k^i), x \rangle \right\}
\end{aligned}
\tag{3.80}
$$

Comparing (3.80) with our developed algorithm in (3.75), we can see that the difference is on the second term within the argmin function. In fact, the inner product item in (3.80) is a linearization of $F_i(x)$ at point y_k^i. The effect is that in each round of update, Nedic's algorithm performs an approximation to the solution of the local minimization problem. Instead, our designed method needs to solve the local optimization problem "completely" in each update round. According to the trade-off (between computation complexity and communication cost) we observed, our proposed algorithm is thus expected to outperform Nedic's in terms of communication cost and total execution time, which are considered to be our goals for implementing seismic imaging system.

3.3.4   Convergence Analysis

In this section, we analyze the convergence behavior of the proposed algorithm in terms of consensus and optimal solution measure. We follow the idea in [109] to derive the analysis of our developed scheme.

Before conducting the analysis, we can rewrite the proposed algorithm (3.75) in a form that similar to Nedic's method as follows.

$$
\begin{aligned}
y_k^i &= \theta x_{k-1}^{i_k} + (1 - \theta)x_{k-1}^i \\
x_k^i &= y_k^i - \alpha_{i,k} \nabla F_i(x_k^i)
\end{aligned}
\tag{3.81}
$$

Note that the second equation of (3.81) comes from the optimality condition of (3.75):

$$\frac{x_k^i - y_k^i}{\alpha_{i,k}} + \nabla F_i(x_k^i) = 0 \tag{3.82}$$

Since the above inequality holds for any $x \in \mathbb{R}^n$, the left part in the inner product has to be equal to 0. This turns out to be the second equation in (3.81).

A compact form for all k and $i \in \mathcal{V}$ can be expressed as:

$$y_k^i = \sum_{j=1}^{m} [W_k]_{i,j}\, x_{k-1}^j$$

$$x_k^i = y_k^i - \left[y_k^i - \alpha_{i,k} \nabla F_i(x_k^i) \right] I(i \in J_k) - y_k^i I(i \in J_k) \tag{3.83}$$

where $I(\cdot)$ is the indicator function and matrix W_k is defined as:

$$[W_k]_{i,i} = 1 - \theta \ for \ i \in J_k, \ \ [W_k]_{i,i} = 1 \ \ otherwise$$

$$[W_k]_{i,i_k} = \theta \ for \ i \in J_k, \ \ [W_k]_{i,j} = 0 \ \ otherwise. \tag{3.84}$$

In addition, three lemmas which would be used later are described here.

Lemma 4 ([110]: lemma 11) *Assume σ_k, φ_k, ω_k, and ε_k are nonnegative random variables and assume the following hold*

$$E\left(\sigma_{k+1}/\Omega_k\right) \le (1 + \omega_k)\sigma_k - \varphi_k + \varepsilon_k \ almost \ surely,$$

$$\sum_{k=0}^{\infty} \omega_k < \infty \ almost \ surely, \ \ \sum_{k=0}^{\infty} \varepsilon_k < \infty \ almost \ surely.$$

where $E\left(\sigma_{k+1}|\Omega_k\right)$ represents the conditional expectation given all the past history of σ_k, φ_k, ω_k, and ε_k up to iteration k. Then it concludes that

$$\sigma_k \to \sigma \ almost \ surely, \ \ \sum_{k=0}^{\infty} \varphi_k < \infty \ almost \ surely.$$

where $\sigma \ge 0$ is some random variable.

Lemma 5 ([109]:lemma 2) *The random matrix* $W_k - \frac{1}{m}\mathbf{1}\mathbf{1}^T W_k$ *is independent and has identical distribution with each other.*

$$\mu := \mu\left(E\left[\left(W_k - \frac{1}{m}\mathbf{1}\mathbf{1}^T W_k\right)^T \left(W_k - \frac{1}{m}\mathbf{1}\mathbf{1}^T W_k\right)\right]\right) < 1 \tag{3.85}$$

where μ *denotes the largest eigenvalue of the following matrix.*

Lemma 6 ([109]) *The upperbounds of step size* $\alpha_{i,k}$ *are obtained as follows when k is large enough* $(k > \tilde{k}(m,q))$

$$\alpha_{i,k} \le \frac{2}{k\delta_i}, \; \alpha_{i,k}^2 \le \frac{4m^2}{k^2 p_*^2}, \; \left|\alpha_{i,k} - 1/k\delta_i\right| \le \frac{2}{k^{\frac{3}{2}-q} p_*^2}. \tag{3.86}$$

where δ_i *is the total probability that node i updates.* p_* *denotes the minimum among all* p_{ij}*'s.* $q \in \left(0, \frac{1}{2}\right)$ *is some constant.* $\tilde{k}(m,q)$ *is certain integer determined by the number of nodes m and q.*

Consensus We first show that the disagreement of each node's solution to the average of all the nodes's converges to zero almost surely.

Theorem 10 *Let* $\left\{x_k^i\right\}, \forall i \in \mathcal{V}, k \ge 0$ *be the sequence generated by the algorithm in* (3.75) *and given that all the assumptions are satisfied. Then we can have:*

$$\sum_{k=1}^{\infty} \frac{1}{k}\|x_{k-1}^i - \bar{x}_{k-1}\| < \infty, \; and \; \lim_{k\to\infty} \|x_k^i - \bar{x}_k\| = 0 \; almost \; surely. \tag{3.87}$$

Proof: We define s_k^t to be the vector with components $\left[x_k^i\right]_t, \forall i \in \mathcal{V}$, where $\left[x_k^i\right]_t$ is the t-th element of node i's estimate at iteration k. Using the second equation in (3.83) we can have:

$$s_k^t = W_k s_{k-1}^t + d_k^t \tag{3.88}$$

Here d_k^t is defined as a vector with

$$[d_k^t]_i = \left[-\alpha_{i,k}\nabla F_i(x_k^i)\right]_t \; for \; i \in J_k, \; [d_k^t]_i = 0 \; otherwise. \tag{3.89}$$

From the definition of s_k^t, it can be shown that:

$$[\bar{x}_k]_t = \frac{1}{m}\mathbf{1}^T s_k^t \tag{3.90}$$

where $\mathbf{1}^T$ denotes a row vector containing all 1's. In addition, plugging (3.90) into (3.88) yields:

$$[\bar{x}_k]_t = \frac{1}{m}\left(\mathbf{1}^T W_k s_{k-1}^t + \mathbf{1}^T d_k^t\right) \tag{3.91}$$

Combing (3.88) and (3.91) derives the following.

$$s_k^t - [\bar{x}_k]_t\,\mathbf{1} = \left(W_k - \frac{1}{m}\mathbf{11}^T W_k\right)s_{k-1}^t + \left(\mathbf{I} - \frac{1}{m}\mathbf{11}^T\right)d_k^t \tag{3.92}$$

where $\mathbf{I}$ represents the identity matrix. Based on the definition and leveraging the stochasticity of matrix W_k, it can be shown that $W_k\mathbf{1} = \mathbf{1}$. Hence we can have:

$$\left(W_k - \frac{1}{m}\mathbf{11}^T W_k\right)[\bar{x}_{k-1}]_t\,\mathbf{1} = 0 \tag{3.93}$$

Adding the right part of (3.93) into both sides of (3.91) yields:

$$s_k^t - [\bar{x}_k]_t\,\mathbf{1} = \left(W_k - \frac{1}{m}\mathbf{11}^T W_k\right)(s_{k-1}^t - [\bar{x}_{k-1}]_t\,\mathbf{1}) + \left(\mathbf{I} - \frac{1}{m}\mathbf{11}^T\right)d_k^t \tag{3.94}$$

To simply the notation, we define $Q_k = W_k - \frac{1}{m}\mathbf{11}^T W_k$ and $U = \mathbf{I} - \frac{1}{m}\mathbf{11}^T$.

The next step is to take the norm and conditional expectation on both sides of (3.94):

$$E\left[\left\|s_k^t - [\bar{x}_k]_t\,\mathbf{1}\right\| |\Omega_{k-1}\right] \le E\left[\left\|Q_k\,(s_{k-1}^t - [\bar{x}_{k-1}]_t\,\mathbf{1})\right\| |\Omega_{k-1}\right] + E\left[\left\|U d_k^t\right\| |\Omega_{k-1}\right] \tag{3.95}$$

where E is the expectation operator and Ω_k is the σ-algebra containing the past history up to iteration k, i.e.

$$\Omega_k = \left\{x_0^i, i_t, j_t, \forall i \in \mathcal{V}, t = 0, 1, \cdots k\right\} \tag{3.96}$$

By using Lemma 5 (μ is the largest eigenvalue mentioned there), we can upper bounding the first

term in the right hand side of (3.95) as follows.

$$E\left[\left\|Q_k\left(s_{k-1}^t - [\bar{x}_{k-1}]_t\,\mathbf{1}\right)\right\|^2 |\Omega_{k-1}\right] \le \mu\left\|s_{k-1}^t - [\bar{x}_{k-1}]_t\,\mathbf{1}\right\|^2 \tag{3.97}$$

Based on the property that $E\left[\|z\|\right] \le \sqrt{E\left[\|z\|^2\right]}$, (3.97) can be transformed to:

$$E\left[\left\|Q_k\left(s_{k-1}^t - [\bar{x}_{k-1}]_t\,\mathbf{1}\right)\right\| |\Omega_{k-1}\right] \le \sqrt{\mu}\left\|s_{k-1}^t - [\bar{x}_{k-1}]_t\,\mathbf{1}\right\| \tag{3.98}$$

The remaining part is that we need to upper bound the second item in the right hand side of (3.95). We find that U is a projection matrix since:

$$U\mathbf{1} = \left(\mathbf{I} - \frac{1}{m}\mathbf{1}\mathbf{1}^T\right)\mathbf{1} = \mathbf{0} \tag{3.99}$$

Then the norm of matrix U is 1. In consequence, we can obtain the following (also from the definition of d_k^t in (3.89)):

$$\left\|Ud_k^t\right\|^2 \le \left\|d_k^t\right\|^2 \le \sum_{i \in J_k}\left\|\alpha_{i,k}\nabla F_i(x_k^i)\right\|^2 \tag{3.100}$$

The bound of $\alpha_{i,k}$ is shown in Lemma 6, thus the right hand side of (3.100) can be further bounded as follows.

$$\left\|Ud_k^t\right\|^2 \le \sum_{i \in J_k}\alpha_{i,k}^2\left\|\nabla F_i(x_k^i)\right\|^2 \le \frac{4m^2}{k^2 p_*^2}\sum_{i \in J_k}\left\|\nabla F_i(x_k^i)\right\|^2 \tag{3.101}$$

Now take conditional expectation on both sides of (3.101), together with the assumed bounded (sub)gradient condition, also the cardinality of J_k is no more than m, we obtain

$$E\left[\left\|Ud_k^t\right\|^2 |\Omega_{k-1}\right] \le \frac{4m^3}{k^2 p_*^2}G^2 \tag{3.102}$$

Using the inequality (converting (3.97) to (3.98)) again yields

$$E\left[\left\|Ud_k^t\right\| |\Omega_{k-1}\right] \le \frac{2m\sqrt{m}}{kp_*}G \tag{3.103}$$

At this point, we have all the bounds of the terms in the right hand side of (3.95). Plugging (3.98) and (3.103) into (3.95) yields

$$E\left[\left\|s_k^t - [\bar{x}_k]_t \mathbf{1}\right\| |\Omega_{k-1}\right] \le \sqrt{\mu}\left\|s_{k-1}^t - [\bar{x}_{k-1}]_t \mathbf{1}\right\| + \frac{2m\sqrt{m}}{kp_*}G \tag{3.104}$$

Since $\frac{1}{k-1} > \frac{1}{k}$, it can be shown that

$$\frac{1}{k}E\left[\left\|s_k^t - [\bar{x}_k]_t \mathbf{1}\right\| |\Omega_{k-1}\right] \le \frac{1}{k-1}\left\|s_{k-1}^t - [\bar{x}_{k-1}]_t \mathbf{1}\right\| - \frac{1-\sqrt{\mu}}{k}\left\|s_{k-1}^t - [\bar{x}_{k-1}]_t \mathbf{1}\right\| + \frac{2m\sqrt{m}}{k^2 p_*}G \tag{3.105}$$

Using the second claim of Lemma 4, we can see the following almost surely for any t.

$$\sum_{k=1}^{\infty}\frac{1}{k}\left\|s_{k-1}^t - [\bar{x}_{k-1}]_t \mathbf{1}\right\| < \infty \tag{3.106}$$

In addition, based on the definition of s_k^t, for any node i, it follows that

$$\sum_{k=1}^{\infty}\frac{1}{k}\left\|x_{k-1}^i - \bar{x}_{k-1}\mathbf{1}\right\| < \infty \tag{3.107}$$

Thus the first part of theorem 10 is proved. For the second claim, we need to first show the following almost surely.

$$\lim_{k\to\infty}\left\|s_k^t - [\bar{x}_k]_t \mathbf{1}\right\| = 0 \tag{3.108}$$

(3.106) implies that

$$\liminf_{k\to\infty}\left\|s_k^t - [\bar{x}_k]_t \mathbf{1}\right\| = 0 \tag{3.109}$$

In order to show (3.108), then we have to prove that the convergence of $\left\|s_k^t - [\bar{x}_k]_t \mathbf{1}\right\|$ when k goes to infinity.

We will use the first claim of Lemma 4 to prove this. First take the square norm and condi-

tional expectation on both sides of (3.94). Further derivations are shown as follows.

$$E\left[\left\|s_k^t - [\bar{x}_k]_t \mathbf{1}\right\|^2 |\Omega_{k-1}\right] \leq E\left[\left\|Q_k\left(s_{k-1}^t - [\bar{x}_{k-1}]_t \mathbf{1}\right)\right\|^2 |\Omega_{k-1}\right] + E\left[\left\|Ud_k^t\right\|^2 |\Omega_{k-1}\right]$$
$$+ 2\sqrt{E\left[\left\|Q_k\left(s_{k-1}^t - [\bar{x}_{k-1}]_t \mathbf{1}\right)\right\|^2 |\Omega_{k-1}\right]}\sqrt{E\left[\left\|Ud_k^t\right\|^2 |\Omega_{k-1}\right]} \quad (3.110)$$

Plugging (3.97) - (3.98) and (3.102)- (3.103) into (3.110) yields

$$E\left[\left\|s_k^t - [\bar{x}_k]_t \mathbf{1}\right\|^2 |\Omega_{k-1}\right] \leq \mu\left\|s_{k-1}^t - [\bar{x}_{k-1}]_t \mathbf{1}\right\|^2 + \frac{4m^3}{k^2 p_*^2}G^2$$
$$+ \sqrt{\mu}\left\|s_{k-1}^t - [\bar{x}_{k-1}]_t \mathbf{1}\right\|\frac{2m\sqrt{m}}{kp_*}G \quad (3.111)$$

Using (3.106) and the first claim of Lemma 4, it is clear to see that $\left\|s_k^t - [\bar{x}_k]_t \mathbf{1}\right\|$ converges almost surely for any t. With this (3.108) is proved. Leveraging the definition of s_k^t one more time, the almost sure convergence of the disagreement is verified (second conclusion). This completes the entire proof of theorem 10.

Optimal Solution   In this section, we derive the convergence analysis in term of solution of each node x_k^i.

Theorem 11 *Let* $\left\{x_k^i\right\}, \forall i \in \mathcal{V}, k \geq 0$ *be the sequence generated by the algorithm in* (3.75) *and given that all the assumptions are satisfied. Then the sequences converge to a same optimal point almost surely for any node i.*

Proof: We start by looking at (3.83) with $i \in J_k$. Subtracting x (some point in the feasible set) on both sides of (3.83) and taking square norm yields the following.

$$\left\|x_k^i - x\right\|^2 \leq \left\|y_k^i - x\right\|^2 + \alpha_{i,k}^2\left\|\nabla F_i(x_k^i)\right\|^2 - 2\alpha_{i,k}\left(\nabla F_i(x_k^i)\right)^T\left(y_k^i - x\right) \quad (3.112)$$

By using the equality $\alpha_{i,k} = \left(\alpha_{i,k} - \frac{1}{k\delta_i}\right) + \frac{1}{k\delta_i}$ and the inequality in Lemma 6, we can bound the

right-most term in (3.112) as follows.

$$\left\| x_k^i - x \right\|^2 \leq \left\| y_k^i - x \right\|^2 + \alpha_{i,k}^2 \left\| \nabla F_i(x_k^i) \right\|^2$$
$$- \frac{2}{k\delta_i} \left(\nabla F_i(x_k^i) \right)^T \left(y_k^i - x \right)$$
$$+ \frac{4}{k^{3/2-q} p_*^2} \left\| \left(\nabla F_i(x_k^i) \right)^T \left(y_k^i - x \right) \right\| \tag{3.113}$$

Based on the fact that $2a^T b \leq \|a\|^2 + \|b\|^2$, the inner product in the right hand side of the (3.113) can be bounded as follows.

$$\left\| x_k^i - x \right\|^2 \leq (1 + \beta_k) \left\| y_k^i - x \right\|^2 - \frac{2}{k\delta_i} \left(\nabla F_i(x_k^i) \right)^T \left(y_k^i - x \right)$$
$$+ \left(\alpha_{i,k}^2 + \beta_k \right) \left\| \nabla F_i(x_k^i) \right\|^2 \tag{3.114}$$

where $\beta_k = \frac{2}{k^{3/2-q} p_*^2}$. By the convexity of function F_i and the bounded (sub)gradient condition, the following inequality holds for arbitrary a, b, and c [109].

$$\nabla F_i(a)^T (a - b) \geq F_i(c) - F_i(b) - G \|a - c\| \tag{3.115}$$

Plugging (3.115) into (3.114) with $a = x_k^i$, $b = x$, $c = \bar{x}_{k-1}$ we can have

$$\left\| x_k^i - x \right\|^2 \leq (1 + \beta_k) \left\| y_k^i - x \right\|^2 - \frac{2}{k\delta_i} \left(F_i(\bar{x}_{k-1}) - F_i(x) \right)$$
$$+ \frac{2G}{k\delta_i} \left\| y_k^i - \bar{x}_{k-1} \right\| + \frac{4G}{k\delta_i} \left\| x_{k-1}^i - \bar{x}_{k-1} \right\| + \tau_k \left\| \nabla F_i(x_k^i) \right\|^2 \tag{3.116}$$

where $\tau_k = \frac{4m^2}{k^2 p_*^2} + \beta_k$. To bound the term $\left\| y_k^i - \bar{x}_{k-1} \right\|$, we will use the property in [109] described in (3.121). Now taking conditional expectation and the bounded (sub)gradient condition, it follows

that

$$E\left[\left\|x_k^i - x\right\|^2 |\Omega_{k-1}, i_k, J_k\right] \leq (1 + \beta_k)\left\|y_k^i - x\right\|^2 - \frac{2}{k\delta_i}\left(F_i(\bar{x}_{k-1}) - F_i(x)\right)$$
$$+ \frac{2G}{k\delta_i}\left\|y_k^i - \bar{x}_{k-1}\right\| + \frac{4G}{k\delta_i}\left\|x_{k-1}^i - \bar{x}_{k-1}\right\| + \tau_k G^2 \qquad (3.117)$$

Recall the definition of δ_i, it is the probability of node i updates (the event that it receives broadcast from any of its neighbors). Hence the fact holds $\delta_i \geq \frac{p_*}{m}$. Then (3.117) can be modified to

$$E\left[\left\|x_k^i - x\right\|^2 |\Omega_{k-1}, i_k, J_k\right] \leq (1 + \beta_k)\left\|y_k^i - x\right\|^2 - \frac{2}{k\delta_i}\left(F_i(\bar{x}_{k-1}) - F_i(x)\right)$$
$$+ \frac{2mG}{kp_*}\left\|y_k^i - \bar{x}_{k-1}\right\| + \frac{4mG}{kp_*}\left\|x_{k-1}^i - \bar{x}_{k-1}\right\| + \tau_k G^2 \qquad (3.118)$$

Now let $x = x^*$ where x^* is an optimal point of the objective function. Substituting this into (3.118) yields

$$E\left[\left\|x_k^i - x^*\right\|^2 |\Omega_{k-1}, i_k, J_k\right] \leq (1 + \beta_k)\left\|y_k^i - x^*\right\|^2 - \frac{2}{k\delta_i}\left(F_i(\bar{x}_{k-1}) - F_i(x^*)\right)$$
$$+ \frac{2mG}{kp_*}\left\|y_k^i - \bar{x}_{k-1}\right\| + \frac{4mG}{kp_*}\left\|x_{k-1}^i - \bar{x}_{k-1}\right\| + \tau_k G^2 \qquad (3.119)$$

Incorporating the case when $i \notin J_k$ ($x_k^i = y_k^i$) with the current formula (which assumes $i \in J_k$), and also with the definition that δ_i denotes the total probability that node i updates, we obtain

$$E\left[\left\|x_k^i - x^*\right\|^2 |\Omega_{k-1}\right] \leq (1 + \beta_k)E\left[\left\|y_k^i - x^*\right\|^2 |\Omega_{k-1}\right] - \frac{2}{k}\left(F_i(\bar{x}_{k-1}) - F_i(x^*)\right)$$
$$+ \frac{2mG}{kp_*}E\left[\left\|y_k^i - \bar{x}_{k-1}\right\| |\Omega_{k-1}\right] + \frac{4mG}{kp_*}E\left[\left\|x_{k-1}^i - \bar{x}_{k-1}\right\| |\Omega_{k-1}\right] + \delta_i \tau_k G^2$$
$$(3.120)$$

It can be shown that for any x the following property holds [109].

$$\sum_{i=1}^{m} E\left[\left\|y_k^i - x\right\| |\Omega_{k-1}\right] \le \sum_{i=1}^{m} \left\|x_{k-1}^i - x\right\| \tag{3.121}$$

At this point, summing up both sides of (3.122) over all the nodes $i \in \mathcal{V}$, applying (3.121) and the definition in (3.74) yields

$$\sum_{i=1}^{m} E\left[\left\|x_k^i - x^*\right\|^2 |\Omega_{k-1}\right] \le (1 + \beta_k) \sum_{i=1}^{m} \left\|x_{k-1}^i - x^*\right\|^2 - \frac{2}{k}\left(F(\bar{x}_{k-1}) - F(x^*)\right)$$
$$+ \frac{6mG}{kp_*} \sum_{i=1}^{m} \left\|x_{k-1}^i - \bar{x}_{k-1}\right\| + \sum_{i=1}^{m} \delta_i \tau_k G^2 \tag{3.122}$$

It can be seen that the summation of β_k over k (from 1 to ∞) is bounded. Furthermore, the last term in (3.122) meets the condition in Lemma 4 due to the definition of τ_k. From the first claim of theorem 10, we can see that the following holds almost surely.

$$\sum_{k=1}^{\infty} \frac{6mG}{kp_*} \sum_{i=1}^{m} \left\|x_{k-1}^i - \bar{x}_{k-1}\right\| < \infty \tag{3.123}$$

Considering the last two terms in (3.122) as one item along with the fact that $F(\bar{x}_{k-1}) - F(x^*) \ge 0$, we can see that all the conditions of Lemma 4 have been satisfied. Hence it concludes that the sequence $\left\{\sum_{i=1}^{m} \left\|x_k^i - x^*\right\|^2\right\}$ converges and

$$\sum_{k=\tilde{k}}^{\infty} \frac{1}{k}\left(F(\bar{x}_{k-1}) - F(x^*)\right) < \infty \tag{3.124}$$

Similar as the proof in theorem 10, it can be deducted from (3.124) that

$$\liminf_{k \to \infty} F(\bar{x}_{k-1}) = F(x^*) \tag{3.125}$$

Since the sequence $\left\{\sum_{i=1}^{m} \left\|x_k^i - x^*\right\|^2\right\}$ converges and (3.125) holds for any point x^* in the set of optimal solutions X^*, we know that there exists a subsequence $\left\{\bar{x}_{k_j}\right\}$ (of sequence $\{\bar{x}_k\}$) such that $\bar{x}_{k_j} \to \hat{x}$ for some $\hat{x}$ in the feasible set X and $\lim_{j \to \infty} F(\bar{x}_{k_j}) = F(x^*)$. By using continuity of

function F and the fact that $\bar{x}_{k_j}$ converges to $\hat{x}$, it follows that $\lim_{j \to \infty} F(\bar{x}_{k_j}) = F(\hat{x})$. Hence we can have $F(\hat{x}) = F(x^*)$, which means $\hat{x}$ belongs to the optimal solution set X^*. Now we can see the sequence $\left\{ \sum_{i=1}^{m} \left\| x_k^i - \hat{x} \right\|^2 \right\}$ converges, together with the fact that $\left\{ \sum_{i=1}^{m} \left\| x_k^i - \bar{x}_k \right\|^2 \right\} \to 0$ as $k \to \infty$ (based on the second claim of theorem 10), we know $\| \bar{x}_k - \hat{x} \|^2$ converges. Since the subsequence $\left\| \bar{x}_{k_j} - \hat{x} \right\|^2 \to 0$, there is $\| \bar{x}_k - \hat{x} \| \to 0$, which shows that $\{\bar{x}_k\}$ converges to an optimal point $(\hat{x})$ of the problem. Finally, using the second claim of theorem 10 again, it can be obtained that the sequence $\left\{ x_k^i \right\}$ generated by any node $i \in \mathcal{V}$ converges to the same optimal solution point almost surely. The proof of the theorem is thus complete.

Mathematical Formulation of Seismic Tomography Problem   In previous, we have discussed the mathematical formulation of seismic tomography problem. In the next, we will demonstrate how the developed method fits into it.

Computing seismic tomography can be modeled as solving an inversion problem as follows.

$$\min_{\mathbf{x}} \quad \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda^2 \|\mathbf{x}\|_2^2 \tag{3.126}$$

where λ is the regularization parameter.

Notice that problem (3.126) can be decomposed since the ray information and travel-time information in $\mathbf{A}$ and $\mathbf{b}$ are originally generated in a distributed manner. Hence the local objective function F_i of in this scenario can be regarded as:

$$F_i = \frac{1}{2} \|\mathbf{A}_i \mathbf{x} - \mathbf{b}_i\|_2^2 + \lambda_i^2 \|\mathbf{x}\|_2^2 \tag{3.127}$$

where i-th sensor node has the knowledge of $\mathbf{A}_i$ and $\mathbf{b}_i$ only.

It can be seen that assumption 2 holds given the problem formulation here. With the domain knowledge in seismic tomography, the bounded gradient assumption 3 is also valid.
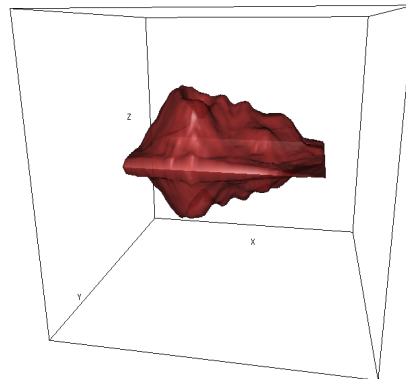
Figure (3.20) Ground truth of the magma model

3.3.5 Experiment Results

Experiment Settings   In this section, we investigate the performance of the proposed algorithm, particularly on the seismic tomography problem.

We study the performance in a 3-D synthetic data set, which is constructed simulating the problem of seismic tomography in sensor network. The proposed scheme is implemented in MATLAB. Experiments are performed on a PC with an Intel i5-3.0G HZ CPU and 8 GB memory.

Regarding the problem model, we use the regularized least squares model in (3.126) mentioned in the last section,, where the regularization parameter λ is set to 1 in all scenarios. The matrix $\mathbf{A}$ and vector $\mathbf{t}$ is constructed by stacking the sub-matrices (sub-vectors) of all the nodes. The resolution means the number of blocks along the x, y and z-axis over the interested region. In the data set, the solution of this "centralized" problem is pre-computed using LSQR method. This centralized solution is used in the convergence measures and seen as the best benchmark for other decentralized algorithms in terms of the tomography result.

In the presenting work, Nedic's algorithm (as a benchmark) is compared with our designed method. For the decentralized formulation, the corresponding parameter λ_i for each node i is set to $1/p$, where p is the total number of nodes in the network. The communication networks are

generated randomly with certain average node degrees. We exam the quantitative performance of the decentralized algorithms by using the metrics mentioned in (3.76) and (3.77).

The 3-D synthetic model has resolution $32 \times 32 \times 32$. The model contains a magma chamber (low velocity area) in a 10 km^3 cube. The number of sensor nodes is set to 100 and they are randomly distributed on top of the region. 400 events are generated and we compute the travel times from every event to each node based on the ground truth, and send the event location and travel time to corresponding node. To simulate the event location estimation and ray tracing errors, a white Gaussian noise is added to the travel time to construct the sensor node observations (arrival times). **A** is formed as a $40,000 \times 32,768$ sparse matrix. The ground truth of the magma model with resolution $128 \times 128 \times 128$ is depicted in Figure 3.20.
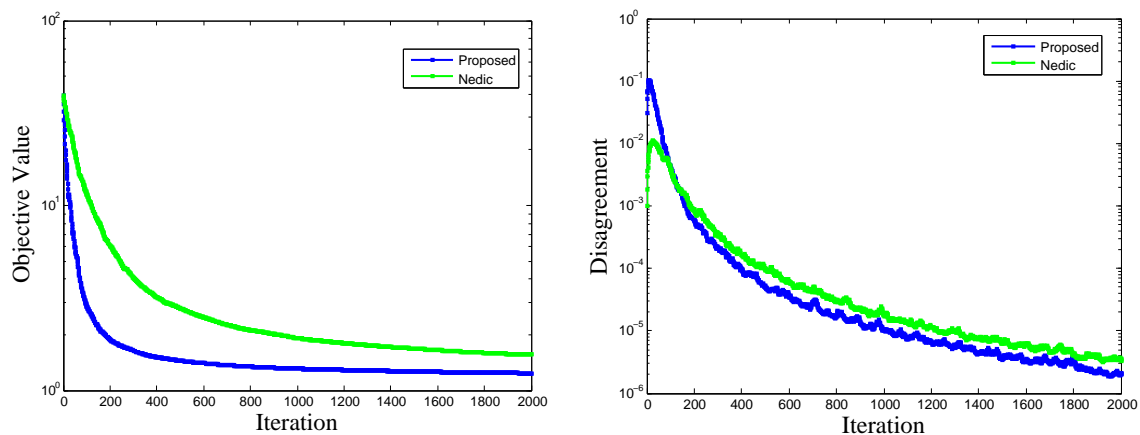

Figure (3.21) Convergence behavior comparison in 3-D synthetic data set.

Comparison of convergence speed We conduct the experiment using a randomly generated communication network, such that each senor is assumed to have 10 neighbors on average within its communication range. From Figure 3.21 we can see that the proposed algorithm (blue) outperforms Nedic's (green) in terms of objective function value and consensus measure. This is consistent with our anlaysis before. We observe that the improvement of proposed algorithm over the benchmark is more significant in the plot of the average objective value. In fact, this effect is favored in system-level design since in realistic scenario, we might take the average of all the avail-

able solutions (from working nodes) rather than randomly select one node. Thus the performance in term of average objective value is more important than the consensus measure for real system. Notice that the *x*-axis and *y*-axis in the figures are all in log-scale, hence the difference between the two methods is in fact large if seen in normal linear scale visually.
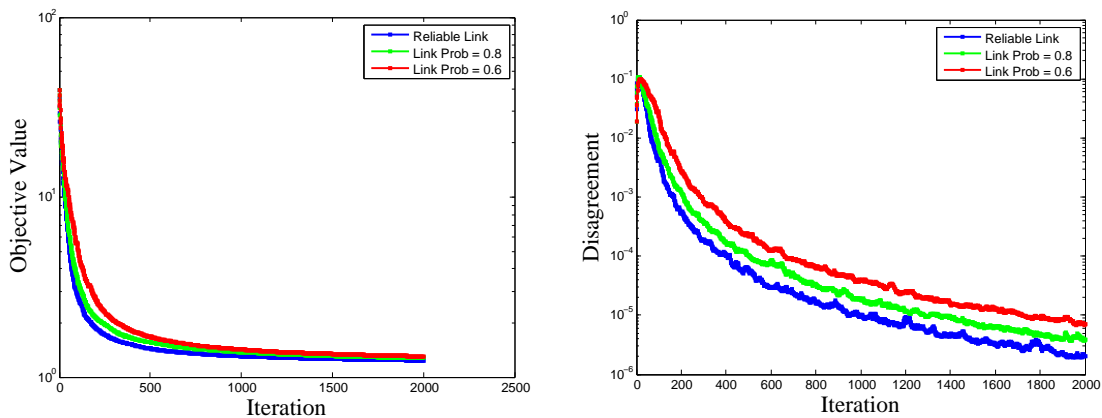

Figure (3.22) Effect of link failure in convergence behavior.

Performance under link failures   We study the performance of the proposed algorithm under link failures. Reliable link ($p_{ij} = 1$), link connection probability $p_{ij} = 0.8, 0.6$ are considered, respectively. It is clear to find the convergence is faster if the link is more reliable (since in each communication rounds, more nodes receive the broadcast and perform updates. This speeds up the convergence of the network as a whole). More interestingly, the curves with link failures are kind of "close" to the one with perfect link especially in the plot of the average objective value. This demonstrates the robustness of the developed method under link failures. The tomography results with link failures are illustrated in Figure 3.23.

Influence of network connectivity ratio   The effect of changing network connectivity ratio is also investigated. In Figure 3.24 we find that the higher the connectivity ratio is, the faster the convergence in terms of average objective value and the consensus. Note that Ratio2 (node degree = 10) is our default setting and it shows close performance to Ratio3 (node degree = 15). Furthermore, we have demonstrated in previous that the proposed algorithm performs well in terms

(a) Ground truth   (b) Centralized   (c) Reliable link   (d) Reliable probability = 0.6

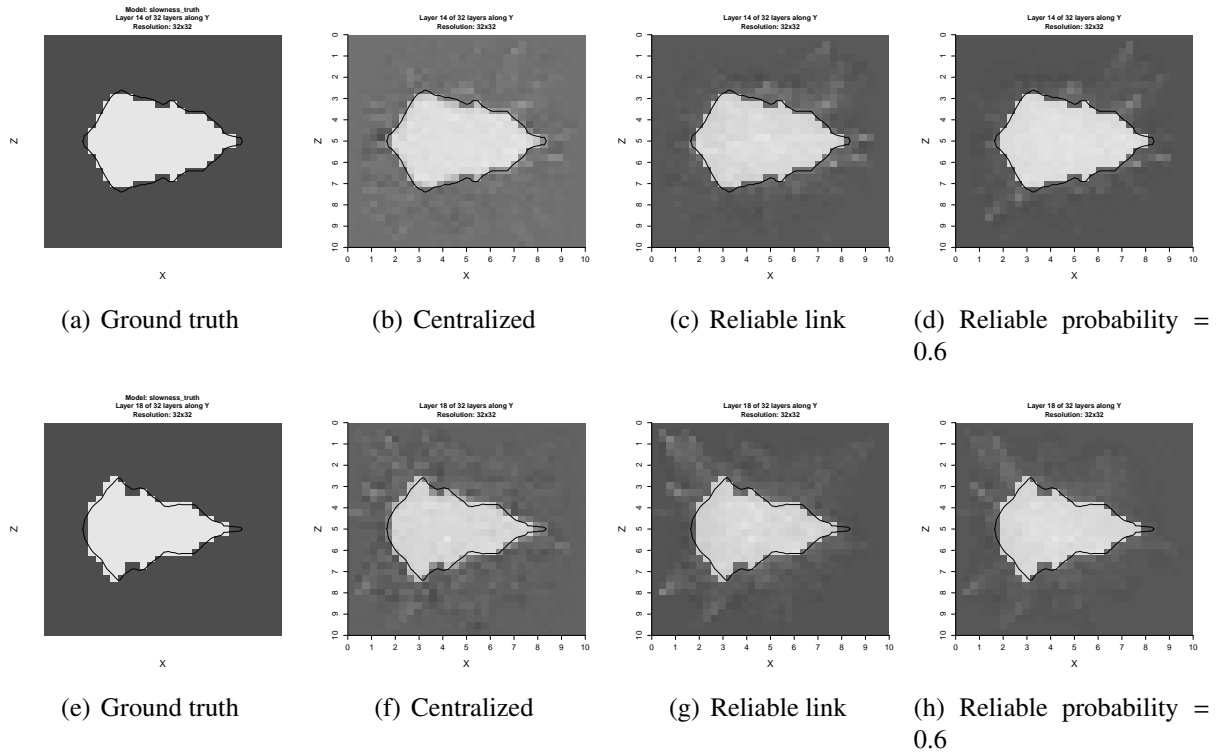(e) Ground truth   (f) Centralized   (g) Reliable link   (h) Reliable probability = 0.6

Figure (3.23) Seismic Tomography Comparison at Layer 14 (upper row) and 18 (bottom row). (a) and (e) are the ground truth. Centralized solution in (b) and (f) are obtained using LSQR method to the centralized formulation. (c) and (g), (d) and (h) are the tomography results of decentralized solutions (at 10th iteration) with reliable link probability =1 and 0.6, respectively.

of tomography results with this small network connectivity ratio (100 nodes in the network). We conclude that the developed scheme still works in very sparse networks, which can be leveraged to cover large-scale region in seismic imaging. Note that the experiments on the influence of network connectivity ratio are conducted as empirical study of the proposed algorithm. We will analyze the theoretical justification as our future work.

Effect of packet loss   We evaluate the robustness of our proposed algorithm from another perspective - packet loss. We simulate packet loss by setting partial vector of the broadcast to zero. We test with packet loss ratios 10% and 30%, respectively. The results at 50 iterations are shown in Figure 3.25. It is clear to see that the distinction between the result without packet loss is relatively small even at the case of 30% packet loss ratio. This validates the fault-tolerances and robustness of our method, especially in applications like seismic tomography, which always suffers
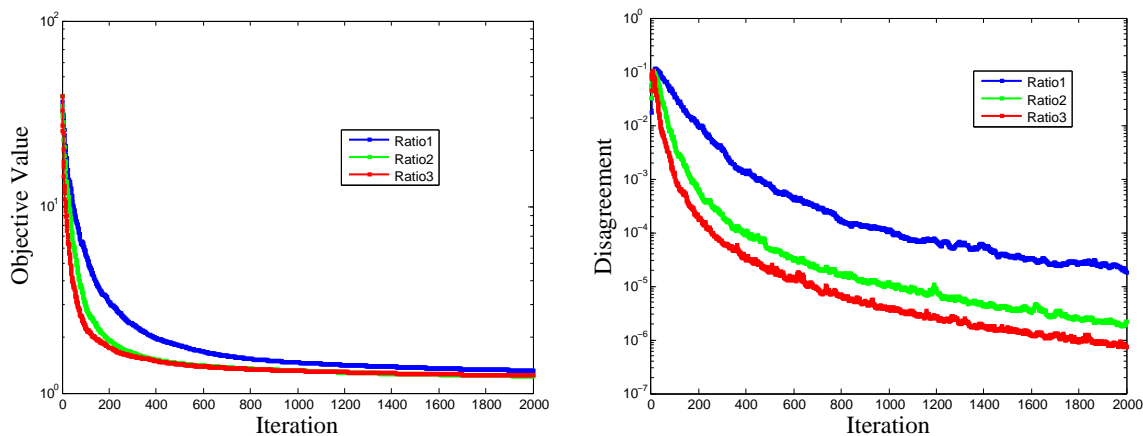
Figure (3.24) Influence of network connectivity ratio. Ratio1 (blue): average node degree = 5, Ratio2 (green): average node degree = 10, Ratio3 (red): average node degree = 15.

from severe packet loss. Similar as the previous subsection, the theoretical analysis of the effect of packet loss to the proposed algorithm is also an interesting direction to explore in the future.

(a) No packet loss (b) 10% packet loss (c) 30% packet loss
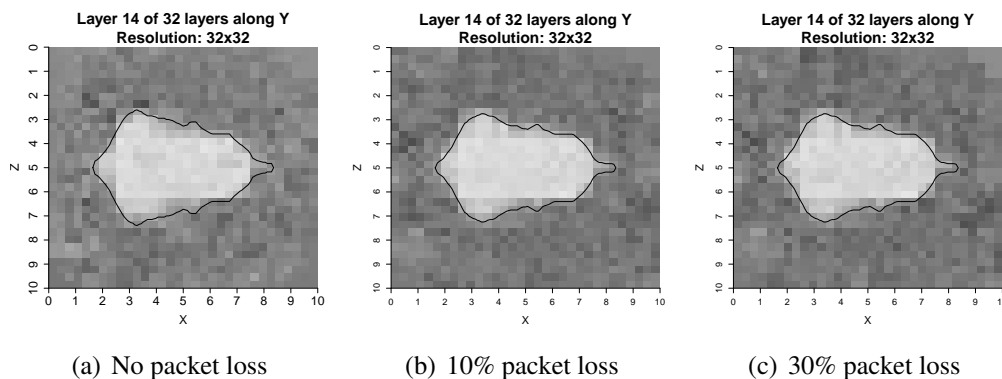
Figure (3.25) Effect of packet loss (vertical slice at layer 14).

Summary   We proposed an asynchronous and broadcast-based algorithm, which does not require any synchronization among the sensor nodes. We conducted extensive tests of proposed algorithm on various aspects. The experiment results show that our designed method outperforms the benchmark. The merits of the proposed methods elucidate that it is a promising solution for real-time in-situ seismic tomography in the near future.

PART 4

CONCLUSIONS

This doctoral thesis proposes solutions for challenges in CPS systems based on the idea of in-situ data analytics. We consider two important and typical study cases: Smart Grid and Seismic Imaging System. For smart grid case, in the first stage, we developed data computing schemes for topology identification and power-line outage detection. Our proposed methods can identify the grid topology with high accuracy given very limited measurements. In addition, a light-weight distributed data analytics framework was designed to overcome the high computational complexity in existing literature, which limits the practical usage of outage detection only for the scenario of single-line or double-line outages. Also the distributed platform can improve the security and privacy for the power grids. For the seismic imaging system, we present several decentralized optimization algorithms for the purpose of real-time in-situ seismic imaging. In the first work, we reformulated the conventional seismic tomography problem and exploit the ADMM method to design two distributed algorithms. One is a synchronous algorithm and the other is asynchronous and more fault-tolerant and scalable. Consider the computation complexity of the proposed ADMM-based algorithms, we then devised a gradient descent based decentralized method, which is even cheaper in in terms of computation. Finally, we developed an asynchronous broadcast-based decentralized algorithm, which is most suitable for the application of seismic imaging in practice. We theoretically proved the convergence and rates of our proposed algorithms in terms of the number of communication rounds. Extensive evaluations on both synthetic and real data sets demonstrated that they not only achieve near-optimal (compare to centralized solution) high quality tomography but also retain low communication cost even in very sparse distributed networks.

Bibliography

[1] Nature, "Overview of the smart grid architecture," http://www.nature.com/nature/journal/v499/n7457/images/499145a-i2.0.jpg.

[2] A. Gomez-Exposito and A. de la Villa Jaen, "Reduced substation models for generalized state estimation," *IEEE Transactions on Power Systems*, vol. 16, no. 4, pp. 839–846, 2001.

[3] Wikipedia, "Smart grid," http://en.wikipedia.org/wiki/Smart_grid.

[4] M. Knezev, Z. Djekic, and M. Kezunovic, "Automated circuit breaker monitoring," in *IEEE Power Engineering Society General Meeting*, 2007, pp. 1–6.

[5] A. Gomez-Exposito, A. Abur, A. de la Villa Jaen, and C. Gomez-Quiles, "A multilevel state estimation paradigm for smart grids," *Proceedings of the IEEE*, vol. 99, no. 6, pp. 952–976, 2011.

[6] R. Brown, "Impact of smart grid on distribution system design," in *Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE*, July 2008, pp. 1–4.

[7] A. Tsikalakis, Y. Katsigiannis, P. Georgilakis, and N. Hatziargyriou, "Determining and exploiting the distribution function of wind power forecasting error for the economic operation of autonomous power systems," in *IEEE Power Engineering Society General Meeting*, 2006, pp. 1–8.

[8] P. Bacher, H. Madsen, and H. A. Nielsen, "Online short-term solar power forecasting," *Solar Energy*, vol. 83, no. 10, pp. 1772–1783, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0038092X09001364

[9] J. M. Lees, "Seismic tomography of magmatic systems," *Journal of Volcanology and Geothermal Research*, vol. 167, no. 1-4, pp. 37–56, 2007.

[10] Y. Sharon, A. Annaswamy, A. Motto, and A. Chakraborty, "Topology identification in distribution network with limited measurements," in *Innovative Smart Grid Technologies (ISGT), 2012 IEEE PES*, Jan 2012, pp. 1–6.

[11] R. Emami and A. Abur, "Tracking changes in the external network model," in *North American Power Symposium (NAPS)*, 2010, pp. 1–6.

[12] ——, "External system line outage identification using phasor measurement units," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 1035–1040, May 2013.

[13] J. Tate and T. Overbye, "Line outage detection using phasor angle measurements," *IEEE Transactions on Power Systems*, vol. 23, no. 4, pp. 1644–1652, 2008.

[14] ——, "Double line outage detection using phasor angle measurements," *IEEE Power Energy Society General Meeting*, pp. 1–5, 2009.

[15] H. Zhu and G. Giannakis, "Sparse overcomplete representations for efficient identification of power line outages," *IEEE Transactions on Power Systems*, vol. 27, no. 4, pp. 2215–2224, 2012.

[16] M. He and J. Zhang, "Fault detection and localization in smart grid: A probabilistic dependence graph approach," in *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, 2010, pp. 43–48.

[17] J.-C. Chen, W.-T. Li, C.-K. Wen, J.-H. Teng, and P. Ting, "Efficient identification method for power line outages in the smart power grid," *IEEE Transactions on Power Systems*, vol. 29, no. 4, pp. 1788–1800, July 2014.

[18] J. Wu, J. Xiong, and Y. Shi, "Ambiguity group based location recognition for multiple power line outages in smart grids," in *Innovative Smart Grid Technologies Conference (ISGT), 2014 IEEE PES*, Feb 2014, pp. 1–5.

[19] T. Banerjee, Y. Chen, A. Garcia, and V. Veeravalli, "Power system line outage detection and identification - a quickest change detection approach," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2014.

[20] X. Wei, S. Lin, and H. Liu, "Distributed fault detection observer for rail vehicle suspension systems," in *Control and Decision Conference (CCDC), 2012 24th Chinese*, 2012, pp. 3396–3401.

[21] J. Gao, J. Wang, and X. Zhang, "Hmrf-based distributed fault detection for wireless sensor networks," in *Global Communications Conference (GLOBECOM), 2012 IEEE*, 2012, pp. 640–644.

[22] R. S. Rao, S. V. L. Narasimham, and M. Ramalingaraju, "Optimization of distribution network configuration for loss reduction using artificial bee colony algorithm," *International Journal of Electrical Power and Energy Systems Engineering*, pp. 116–122, 2008.

[23] A. Abdelaziz, F. Mohamed, S. Mekhamer, and M. Badr, "Distribution system reconfiguration using a modified tabu search algorithm," *Electric Power Systems Research*, vol. 80, no. 8, pp. 943–953, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S037877961000012X

[24] K. S. Kumar and T. Jayabarathi, "Power system reconfiguration and loss minimization for an distribution systems using bacterial foraging optimization algorithm," *International Journal of Electrical Power and Energy Systems*, vol. 36, no. 1, pp. 13–17, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0142061511002651

[25] H. Khodr, J. Martinez-Crespo, M. Matos, and J. Pereira, "Distribution systems reconfiguration based on opf using benders decomposition," *IEEE Transactions on Power Delivery*, vol. 24, no. 4, pp. 2166–2176, Oct 2009.

[26] D. Das, "A fuzzy multiobjective approach for network reconfiguration of distribution systems," *IEEE Transactions on Power Delivery*, vol. 21, no. 1, pp. 202–209, Jan 2006.

[27] S. R. S. N.Shanmuga Vadivoo, "Distribution system restoration using genetic algorithm with distributed generation," *Modern Applied Science*, 2009.

[28] S. Bose, S. Pal, B. Natarajan, C. Scoglio, S. Das, and N. Schulz, "Analysis of optimal reconfiguration of shipboard power systems," *IEEE Transactions on Power Systems*, vol. 27, no. 1, pp. 189–197, Feb 2012.

[29] S. Gupta, W. Yeoh, E. Pontelli, P. Jain, and S. Ranade, "Modeling microgrid islanding problems as dcops," in *North American Power Symposium (NAPS)*, Sept 2013, pp. 1–6.

[30] R. M. Vitorino, H. M. Jorge, and L. P. Neves, "Multi-objective optimization using nsga-ii for power distribution system reconfiguration," *International Transactions on Electrical Energy Systems*. [Online]. Available: http://dx.doi.org/10.1002/etep.1819

[31] G. Mateos, I. D. Schizas, and G. B. Giannakis, "Performance Analysis of the Consensus-Based Distributed LMS Algorithm," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, 2010.

[32] I. D. Schizas, G. Mateos, and G. B. Giannakis, "Distributed LMS for consensus-based in-network adaptive processing," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2365–2382, 2009.

[33] A. H. Sayed and C. G. Lopes, "Distributed Recursive Least-Squares Strategies Over Adaptive Networks," in *Signals, Systems and Computers, 2006. ACSSC '06. Fortieth Asilomar Conference on*, Nov. 2006, pp. 233–237.

[34] I. Matei and J. Baras, "Performance evaluation of the consensus-based distributed subgradient method under random communication topologies," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 5, no. 4, pp. 754–771, Aug 2011.

[35] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *Automatic Control, IEEE Transactions on*, vol. 54, no. 1, pp. 48–61, Jan 2009.

[36] A. Nedic and A. Olshevsky, "Distributed optimization over time-varying directed graphs," in *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, Dec 2013, pp. 6855–6860.

[37] ——, "Stochastic gradient-push for strongly convex functions on time-varying directed graphs," *arXiv:1406.2075*, 2014.

[38] I.-A.Chen, "Fast distributed first-order methods," *PhD thesis, Massachusetts Institute of Technology*, 2012.

[39] J. M. F. M. Dusan Jakovetic, Joao Xavier, "Fast distributed gradient methods," *arXiv:1112.2972v4*, 2014.

[40] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," *arXiv:1310.7063*, 2013.

[41] M. Zargham, A. Ribeiro, and A. Jadbabaie, "A distributed line search for network optimization," in *American Control Conference (ACC), 2012*, June 2012, pp. 472–477.

[42] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*, ser. IPSN '05. Piscataway, NJ, USA: IEEE Press, 2005. [Online]. Available: http://dl.acm.org/citation.cfm?id=1147685.1147698

[43] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *Automatic Control, IEEE Transactions on*, vol. 31, no. 9, pp. 803–812, Sep 1986.

[44] J.N.Tsitsiklis, "Problems in decentralized decision making and computation," *Technical report, DTIC Document*, 1984.

[45] U. T. H. Terelius and R. Murray, "Decentralized multi-agent optimization via dual decomposition," *IFAC*, 2011.

[46] G. Shi and K. H. Johansson, "Finite-time and asymptotic convergence of distributed aver-aging and maximizing algorithms," *arXiv:1205.1733*, 2012.

[47] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *Information Processing in Sensor Networks, 2004. IPSN 2004. Third International Symposium on*, April 2004, pp. 20–27.

[48] W. Shi, Q. Ling, G. Wu, and W. Yin, "Extra: An exact first-order algorithm for decentralized consensus optimization," *arXiv:1404.6264*, 2014.

[49] E. Wei and A. Ozdaglar, "On the o(1/k) convergence of asynchronous distributed alternating direction method of multipliers," *arXiv:1307.8254*, 2013.

[50] P. C. F. Iutzeler, P. Bianchi and W. Hachem, "Asynchronous distributed optimization using a randomized alternating direction method of multipliers," *arXiv:1303.2837*, 2013.

[51] D. MacKay, *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003. [Online]. Available: http://books.google.com/books?id=AKuMj4PN_EMC

[52] E. Caro, A. Conejo, and A. Abur, "Breaker status identification," *IEEE Transactions on Power Systems*, vol. 25, no. 2, pp. 694–702, 2010.

[53] Z.-Q. Luo, W.-K. Ma, A.-C. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 20–34, 2010.

[54] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

[55] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.0 beta," http://cvxr.com/cvx, Sep. 2013.

[56] L. Zhao, W.-Z. Song, L. Tong, Y. Wu, and J. Yang, "Topology identification in smart grid with limited measurements via convex optimization," in *Innovative Smart Grid Technologies - Asia (ISGT Asia), 2014 IEEE*, May 2014, pp. 803–808.

[57] U. Faigle, W. Kern, and G. Still, *Algorithmic Principles of Mathematical Programming*, ser. Texts in the Mathematical Sciences. Springer, 2002. [Online]. Available: http://books.google.com/books?id=_z8OP0GmM58C

[58] D. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[59] R. Zimmerman, C. Murillo-Sanchez, and R. Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, 2011.

[60] NERC, "Technical analysis of the august 14, 2003, blackout: What happened, why, and what did we learn?" http://www.nerc.com/docs/docs/blackout/NERC_Final_Blackout_Report_07_13_04.pdf.

[61] V. Terzija, G. Valverde, D. Cai, P. Regulski, V. Madani, J. Fitch, S. Skok, M. Begovic, and A. Phadke, "Wide-area monitoring, protection, and control of future electric power networks," *Proceedings of the IEEE*, vol. 99, no. 1, pp. 80–93, 2011.

[62] Wikipedia, "Kirchhoff's circuit laws," http://en.wikipedia.org/wiki/Kirchhoff's_circuit_laws.

[63] A. Willsky and H. Jones, "A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems," *IEEE Transactions on Automatic Control*, vol. 21, no. 1, pp. 108–112, 1976.

[64] O. Kosut, L. Jia, R. Thomas, and L. Tong, "Malicious data attacks on smart grid state estimation: Attack strategies and countermeasures," in *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, Oct 2010, pp. 220–225.

[65] ——, "Malicious data attacks on the smart grid," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 645–658, Dec 2011.

[66] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011. [Online]. Available: http://dx.doi.org/10.1561/2200000016

[67] R. Rockafellar, *Convex Analysis*, ser. Princeton landmarks in mathematics and physics. Princeton University Press, 1997. [Online]. Available: http://books.google.com/books?id=1TiOka9bx3sC

[68] G. Strang, *Introduction to Linear Algebra*, 4th ed. Wellesley Cambridge Press, 2009.

[69] C. C. Paige and M. A. Saunders, "Lsqr: An algorithm for sparse linear equations and sparse least squares," *ACM Trans. Math. Softw.*, vol. 8, no. 1, pp. 43–71, Mar. 1982. [Online]. Available: http://doi.acm.org/10.1145/355984.355989

[70] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, vol. 33, no. 1, pp. 1–22, 2010. [Online]. Available: http://www.jstatsoft.org/v33/i01/

[71] S. Azizi, A. Dobakhshari, S. Nezam Sarmadi, and A. Ranjbar, "Optimal pmu placement by an equivalent linear formulation for exhaustive search," *IEEE Transactions on Smart Grid*, vol. 3, no. 1, pp. 174–182, 2012.

[72] N. Hatziargyriou, H. Asano, R. Iravani, and C. Marnay, "Microgrids," *IEEE Power and Energy Magazine*, vol. 5, no. 4, pp. 78–94, July 2007.

[73] S. Karnouskos, "Cyber-physical systems in the smartgrid," in *Industrial Informatics (IN-DIN), 2011 9th IEEE International Conference on*, July 2011, pp. 20–23.

[74] M. Fukuda and M. Kojima, "Branch-and-cut algorithms for the bilinear matrix inequality eigenvalue problem," *Comput. Optim. Appl.*, vol. 19, no. 1, pp. 79–105, Apr. 2001. [Online]. Available: http://dx.doi.org/10.1023/A:1011224403708

[75] O. Toker and H. Ozbay, "On the np-hardness of solving bilinear matrix inequalities and simultaneous stabilization with static output feedback," in *American Control Conference, Proceedings of the 1995*, vol. 4, Jun 1995, pp. 2525–2526.

[76] Y. Nesterov and A. Nemirovskii, *Interior Point Polynomial Algorithms in Convex Programming*, ser. Studies in Applied Mathematics. Society for Industrial and Applied Mathematics, 1994. [Online]. Available: http://books.google.com/books?id=C-MjQ98V9eoC

[77] F. Barahona, M. Junger, and G. Reinelt, "Experiments in quadratic 0-1 programming," *Math. Program.*, vol. 44, no. 2, pp. 127–137, Jul. 1989. [Online]. Available: http://dx.doi.org/10.1007/BF01587084

[78] A. S. Nemirovski and M. J. Todd, "Interior-point methods for optimization," *Acta Numerica*, pp. 191–234, 2008.

[79] G. Calafiore and M. Campi, "The scenario approach to robust control design," *IEEE Transactions on Automatic Control*, vol. 51, no. 5, pp. 742–753, May 2006.

[80] K. Miettinen, *Nonlinear multiobjective optimization*. Kluwer Academic Publishers, Boston, 1999.

[81] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, Apr 2002.

[82] R. P. Bording, A. Gersztenkorn, L. R. Lines, J. A. Scales, and S. Treitel, "Applications of seismic travel-time tomography," *Geophysical Journal of the Royal Astronomical Society*, vol. 90, no. 2, pp. 285–303, 1987. [Online]. Available: http://dx.doi.org/10.1111/j.1365-246X.1987.tb00728.x

[83] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.

[84] T. Erseghe, D. Zennaro, E. Dall'Anese, and L. Vangelista, "Fast consensus by the alternating direction multipliers method," *Signal Processing, IEEE Transactions on*, vol. 59, no. 11, pp. 5523–5537, Nov 2011.

[85] G. Mateos, J. Bazerque, and G. Giannakis, "Distributed sparse linear regression," *Signal Processing, IEEE Transactions on*, vol. 58, no. 10, pp. 5262–5276, Oct 2010.

[86] J. M. Lees and R. S. Crosson, *Bayesian ART versus conjugate gradient methods in tomographic seismic imaging: an application at Mount St. Helens, Washington*, ser. Lecture Notes–Monograph Series.   Hayward, CA: Institute of Mathematical Statistics, 1991, vol. Volume 20, pp. 186–208. [Online]. Available: http://dx.doi.org/10.1214/lnms/1215460502

[87] J. Eckstein and D. Bertsekas, "On the douglas-rachford splitting method and the proximal point algorithm for maximal monotone operators," 10 1989.

[88] K. Tsianos, S. Lawlor, J. Y. Yu, and M. Rabbat, "Networked optimization with adaptive communication," in *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*, Dec 2013, pp. 579–582.

[89] J. S. Rosenthal, "Convergence rates of markov chains," *SIAM Rev.*, vol. 37, pp. 387–405, 1995.

[90] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems and Control Letters*, vol. 53, pp. 65–78, 2003.

[91] S. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing markov chain on a graph," *SIAM Rev.*, vol. 46, no. 4, pp. 667–689, Apr. 2004. [Online]. Available: http://dx.doi.org/10.1137/S0036144503423264

[92] J. Ahrenholz, "Comparison of core network emulation platforms," in *MILITARY COMMUNICATIONS CONFERENCE, 2010 - MILCOM 2010*, Oct 2010, pp. 166–171.

[93] P. C. Hansen and M. Saxild-Hansen, "Air tools - a matlab package of algebraic iterative reconstruction methods," *Journal of Computational and Applied Mathematics*, vol. 236,

no. 8, pp. 2167 – 2178, 2012, inverse Problems: Computation and Applications. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377042711005188

[94] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," *arXiv:1310.7063*, 2013.

[95] W. Shi, Q. Ling, G. Wu, and W. Yin, "Extra: An exact first-order algorithm for decentralized consensus optimization," *arXiv:1404.6264*, 2014.

[96] J. M. F. M. Dusan Jakovetic, Joao Xavier, "Fast distributed gradient methods," *arXiv:1112.2972v4*, 2014.

[97] I. Loris, G. Nolet, I. Daubechies, and F. A. Dahlen, "Tomographic inversion using l1-norm regularization of wavelet coefficients," *Geophysical Journal International*, pp. 359–370.

[98] V. Cevher, S. Becker, and M. Schmidt, "Convex optimization for big data: Scalable, randomized, and parallel algorithms for big data analytics," *Signal Processing Magazine, IEEE*, vol. 31, no. 5, pp. 32–43, Sept 2014.

[99] Y. Nesterov, "Gradient methods for minimizing composite objective function," Universit catholique de Louvain, Center for Operations Research and Econometrics (CORE), CORE Discussion Papers 2007076, 2007. [Online]. Available: http://EconPapers.repec.org/RePEc:cor:louvco:2007076

[100] ——, *Introductory Lectures on Convex Optimization: A Basic Course (Applied Optimization)*, 1st ed. Springer Netherlands. [Online]. Available: http://www.worldcat.org/isbn/1402075537

[101] ——, "A method for unconstrained convex minimization problem with the rate of convergence o(1/k2)." *In Doklady AN SSSR, volume 269, pages 543547*, 1983.

[102] P. Tseng, "On accelerated proximal gradient methods for convex-concave optimization," *submitted to SIAM Journal on Optimization*, 2008.

[103] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE/ACM Trans. Netw.*, vol. 14, no. SI, pp. 2508–2530, Jun. 2006. [Online]. Available: http://dx.doi.org/10.1109/TIT.2006.874516

[104] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione, "Broadcast Gossip Algorithms for Consensus," *Signal Processing, IEEE Transactions on*, vol. 57, no. 7, pp. 2748–2761, Jul. 2009. [Online]. Available: http://dx.doi.org/10.1109/tsp.2009.2016247

[105] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-Based Computation of Aggregate Information," in *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, ser. FOCS '03, 2003, pp. 482–491.

[106] F. Iutzeler, P. Ciblat, W. Hachem, and J. Jakubowicz, "New broadcast based distributed averaging algorithm over wireless sensor networks," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2012, pp. 3117–3120.

[107] D. Ustebay, M. Coates, and M. Rabbat, "Greedy gossip with eavesdropping," in *Wireless Pervasive Computing, 2008. ISWPC 2008. 3rd International Symposium on*, May 2008, pp. 759–763.

[108] S. Wu, B. Liu, X. Bai, and Y. Hou, "Eavesdropping-based gossip algorithms for distributed consensus in wireless sensor networks," *IEEE Signal Processing Letters*, vol. 22, no. 9, pp. 1388–1391, Sept 2015.

[109] A. Nedic, "Asynchronous broadcast-based convex optimization over a network," *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1337–1351, June 2011.

[110] B.T.Polyak, *Introduction to Optimization*.