BIG DATA OPTIMIZATION FOR MODERN COMMUNICATION NETWORKS

A Dissertation

Presented to

the Faculty of the Electrical and Computer Engineering Department

University of Houston

in Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

in Electrical Engineering

by

Lanchao Liu

December 2014

BIG DATA OPTIMIZATION FOR MODERN COMMUNICATION NETWORKS

Lanchao Liu

Approved:

Chair of the Committee
Dr. Zhu Han, Associate Professor
Electrical and Computer Engineering

Committee Members:

Dr. Wei-chuan Shih, Assistant Professor
Electrical and Computer Engineering

Dr. Haluk Ogmen, Professor
Electrical and Computer Engineering

Dr. Saurabh Prasad, Assistant Professor
Electrical and Computer Engineering

Dr. Miao Pan, Assistant Professor
Computer Science
Texas Southern University

Dr. Mingyi Hong, Assistant Professor
Industrial and Manufacturing Systems Engineering
Iowa State University

Dr. Suresh K. Khator, Associate Dean,
Cullen College of Engineering

Dr. Badrinath Roysam, Professor and Chairman,
Electrical and Computer Engineering

Acknowledgements

When I was an undergraduate student in China, I always wished to study in the U.S. and work closely with Dr. Zhu Han, who is one of the world-famous researchers in the area of wireless communication and signal processing. My dream came true when Dr. Haluk Ogmen and Dr. Ji Chen visited my hometown and introduced me to Dr. Zhu Han, and it has been a great honor for me to have Dr. Zhu Han as my Ph.D. advisor.

I would like to express my deepest gratitude to my advisor Dr. Zhu Han for his excellent guidance and continual support during my graduate studies. Working with him felt like the smooth sailing of a cruise ship in the Caribbean summer. I was always fully charged, upbeat, and energetic to attain my research goals. In hindsight, though there was some turbulence along the way, the supporting role that Dr. Zhu Han played was so reaffirming and so warm. I never felt discouraged.

I would also like to thank my dissertation committee, Dr. Wei-Chuan Shih, Dr. Haluk Ogmen, Dr. Saurabh Prasad, Dr. Miao Pan, and Dr. Mingyi Hong for their encouragement, insightful comments, valuable discussions, and accessibility. I would like to give special thanks to Dr. Wotao Yin, who introduced me to the field of mathematical optimization and taught me how to conduct high-impact interdisciplinary research. I also learned the crucial aspects of being a good researcher from him, namely, passion, critical thinking, rigor, and hard work. Dr. Wotao Yin has contributed significantly to both my research and my professional development.

I owe my gratitude to my colleagues and friends at University of Houston, Ali Arab, Erte Pan, Huaqing Zhang, Yanru Zhang, and Yunan Gu. I appreciate all the helpful discussions that I had with them over the years. I am also grateful to Ruoli Jiang, Linsen Wu, Xiyao Xin, Xichen Guo and Yinxi Zhang, who have helped me directly or indirectly in completing my studies and have made my Ph.D. a rewarding experience.

I am deeply indebted to my parents and my family who have been a constant source of support and love throughout this degree and my life. Thank you for everything. Your love will get me home.

BIG DATA OPTIMIZATION FOR MODERN COMMUNICATION NETWORKS

An Abstract

of a

Dissertation

Presented to

the Faculty of the Electrical and Computer Engineering Department

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

in Electrical Engineering

by

Lanchao Liu

December 2014

Abstract

The unprecedented big data in modern communication networks presents us opportunities and challenges. An efficient analytic method for the sheer volume of data is of significant importance for smart grid evolution, intelligent communication network management, efficient medical data management, personalized business model design and smart city development. Meanwhile, the huge volume of data makes it impractical to collect, store and processing in a centralized fashion. Moreover, the massive datasets are noisy, incomplete, heterogeneous, structured, prone to outliers, and vulnerable to cyber-attacks. Overall, we are facing a problem in which the classic resources of computation such as time, space, and energy, are intertwined in complex ways with the massive data sources, and new computational mathematical models as well as methodologies must be explored.

With the rapid development of the modern communication networks comes the need of novel algorithms for large-scale data processing and optimization. In this thesis, we investigate the application of big data optimization methods for smart grid security and mobile data traffic management. Firstly, we review the parallel and distributed optimization algorithms based on an alternating direction method of multipliers for solving big data optimization problems. The mathematical backgrounds of the algorithms are given, and the implementations on large-scale computing facilities are also illustrated. Next, the applications of big data processing techniques for smart grid security are studied from two perspectives: how to exploit the inherent structure of the data, and how to deal with the huge size of the data sets. Explored problems are the sparse optimization approach for false data injection detection, and the distributed parallel approach for the security-constrained optimal power flow problem, respectively. Finally, we consider big data optimization methods for data traffic management in mobile cloud computing by two specific application cases: the mobile data offloading in a software defined network at the network edge, and the management of mobile cloud service request allocation and response routing. It is shown by numerical results that effective management and processing of big data have the potential to significantly improve smart grid security as well as resource utilization and service quality of the mobile cloud computing.

Table of Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction and Background

Nowadays, modern communication networks play an important role in electric power system, mobile cloud computing, smart city evolution and personal health care. The employed novel telecommunication technologies make data collection much easier for power system operation and control, enable more efficient data transmission for mobile applications, and promise a more intelligent sensing and monitoring for metropolitan city-regions. Meanwhile, we are witnessing an unprecedented rise in volume, variety and velocity of information in modern communication networks. A large volume of data are generated by our digital equipments such as mobile devices and computers, smart meters and household appliances, as well as surveillance cameras and sensor-equipped mass rapid transit around the city. The information exposition of big data in modern communication networks makes statistical and computational methods significantly important for data analysis, processing, and optimization. The network operators or service providers who can develop and exploit efficient methods to tackle big data challenges will ensure network security and resiliency, gain market share, increase revenue with distinctive quality of service, as well as achieve intelligent network operation and management.

This chapter gives introduction and background of big data optimization for modern communication networks. In particular, big data processing techniques for smart grid system security and scalable mobile networks traffic management are considered. The rest of this chapter is organized as follows. The motivation and context are provided in Section 1.1. Section 1.2 describes the thesis outline and major contributions. The published results are given in Section 1.3. Finally, Section 1.4 introduces notational conventions used in this thesis.

1.1 Motivation and Context

1.1.1 Big Data Optimization for Smart Grid Security

The smart grid is a modernized power system which enables bidirectional flows of energy as well as using two-way communication and control capabilities to improve efficiency, reliability, economics and sustainablility of the production and distribution of electricity. In the conceptual model of the smart grid, seven components are introduced as described in Table 1.1 [1], and an illustration of their interaction is explained in Fig. 1.1 [1]. The smart grid is an integration of electrical and communication infrastructures. The inevitable coupling between information/communication technologies and physical operations is expected to present unique challenges as well as opportunities for the smart grid.

On one hand, we are observing increasing integration between cyber operations and physical infrastructures for generation, transmission, and distribution control in the electric power grid. Yet security and reliability of the power grid are not always guaranteed and some failures can cause significant problems for the grid. For example, the 2003 Northeast power blackout showed that even a small failure in a part of the grid can have cascading effects causing billions of dollars in economic losses. Nowadays, the consolidation of physical and cyber components gives rise to security threats in power grids, which can result in power outages and even system blackouts [2], or substantial economical loss due to non-optimal operations of the power grid.

On the other hand, the anticipated smart grid data deluge, generated by sensing and measurement devices and reinforced by communication and information technologies, provides us the potential to enhance security and reliability of the power system. For example, the deployment of phasor measurement units (PMUs), which provide real-time assessments of power system health to system operators, for the future North American power grid will generate 4.15 TB phasor data per day. It is estimated that 61.8 million smart meters will be deployed in the U.S. by the end of 2013, and the estimated amount of compressed smart meter data for one million users per year is 27.3TB [3]. Those big data, if effectively managed and translated into actionable insights, has

Table 1.1 Domains and roles/services in the smart grid conceptual model.

	Domain	Roles/Services
1	Customer	The end users of electricity. May also generate, store, and manage the use of energy. Traditionally, three customer types are discussed, each with its own domain: residential, commercial, and industrial.
2	Markets	The operators and participants in electricity markets.
3	Service Provider	The organizations providing services to electrical customers and to utilities
4	Operations	The managers of the movement of electricity.
5	Generation	The generators of electricity. May also store energy for later distribution.
6	Transmission	The carriers of bulk electricity over long distances. May also store and generate electricity.
7	Distribution	The distributors of electricity to and from customers. May also store and generate electricity.

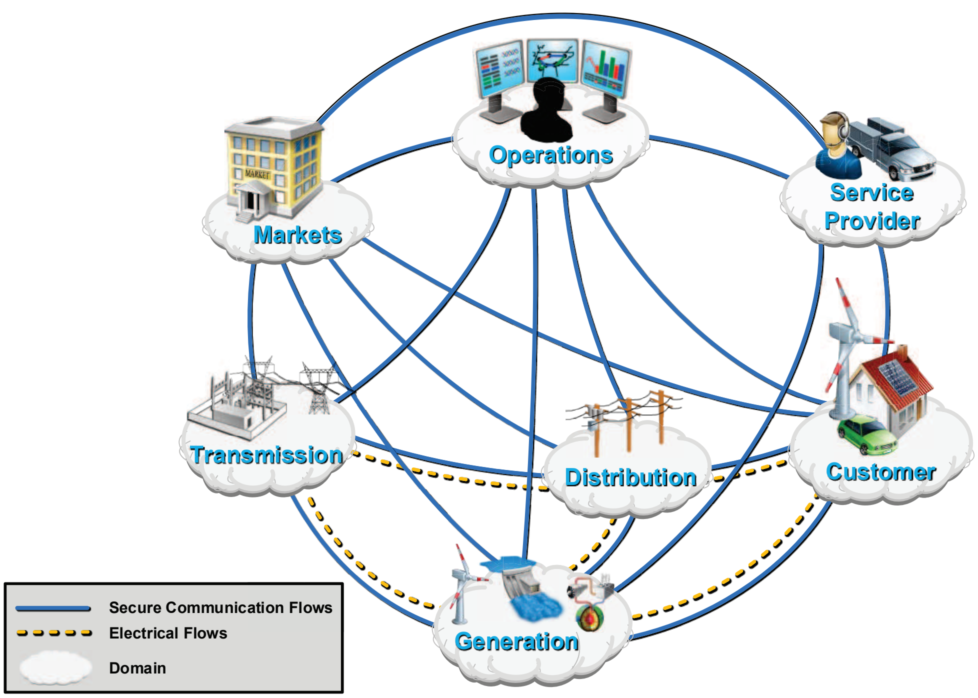


Figure 1.1 An illustration of the updated NIST smart grid framework 3.0.

the potential to increase operational efficiency and ensure grid resiliency of the power system. The adopted methods should be able to utilize the inherent structure of data to extract useful information. Moreover, big data should be processed in a timely fashion. Thus, new computational mathematical models and methodologies must be explored to effectively operate an ever-complicated power grid and achieve the vision of a smart grid.

1.1.2 Scalable Traffic Management for Mobile Networks

Now wireless has become the primary or even the sole access method for more and more people. The global mobile data traffic has reached 1.5 exabytes per month by the end of 2013, and will increase nearly 11-fold between 2013 and 2018, reaching 15.9 exabytes per month by 2018 [4] as shown in Fig. 1.2 [4]. The sheer volume of mobile big data traffic far exceeds the growth in service revenues as well as in budgets required to address these new demands. Mobile service operators need to enhance their infrastructures and services in a timely and cost-effective manner to carry higher volumes of traffic and support more sophisticated services.

The traditional static network architecture is ill-suited to dynamic computing and storage requirements of today's mobile cloud computing environment. Conventional networks are hierarchical, built with tiers of network switches arranged in a tree structure. With the rise of cloud service and the increasingly employing mobile personal device, the traditional network architecture can not address changing traffic patterns and increasing amounts of traffic in the network. The rise of mega data sets is fueling constant demand for additional network capacity. Meanwhile, operators of hyper-scale mobile networks face the daunting task of scaling the network (to a previously unimaginable size), maintaining connectivity, and satisfying the quality of service requirement. Hence, new network paradigm and service traffic management mechanism are necessary to accommodate huge bandwidth needs for big data.

Further, efficient and scalable service management mechanisms are needed to address big data traffic and coordinate different entities (data centers, service hosts, and routers) to provide end users with qualified services at a reasonable cost in the mobile cloud computing. The mobile cloud

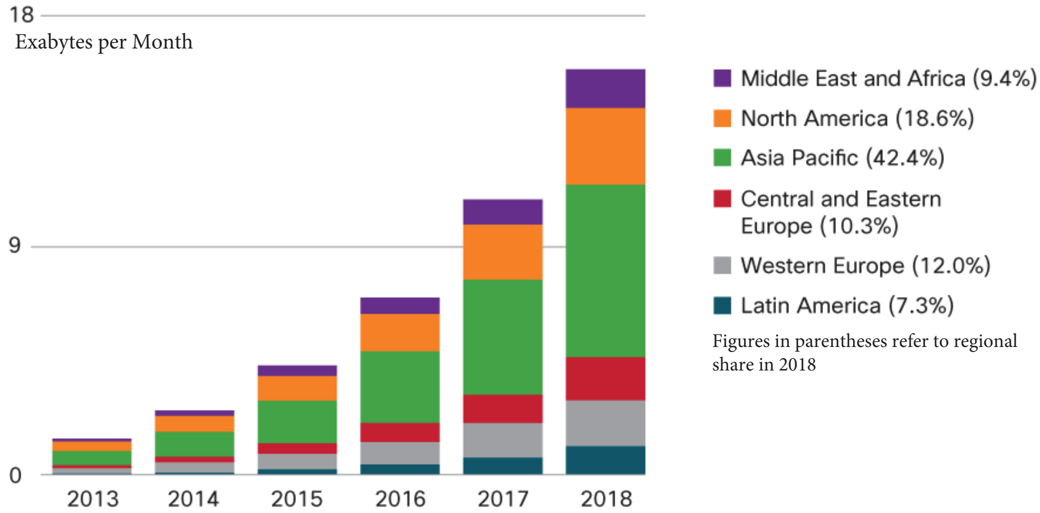


Figure 1.2 The global mobile data traffic forecast by region.

computing can improve the performance of mobile applications by offloading data processing and storage from a mobile device to the cloud. By deploying services on several cloud-enabled data centers, the service provider can optimally locate service instances on the cloud to provide qualified services at a reasonable cost. However, a centralized approach for both request allocation and response routing does not scale due to the large number of mobile clients involved in the service management problem. Moreover, the random and unpredictable wireless network performance (such as delays) complicates the problem. Hence, scalable and distributed mechanisms for service management are needed in the mobile cloud computing.

1.2 Thesis Outline and Contributions

The research dealt with in this thesis contributes to the development of efficient and scalable methods for big data optimization problem in modern communication networks. The proposed methods are based on the alternating direction method of multipliers (ADMM), which are able to leverage the inherent sparse or low rank structure of data as well as enjoy the robustness and scalability. The applications of ADMM for smart grid security and scalable mobile traffic management are investigated. The contributions of this thesis are enumerated as follows

- We reviewed parallel and distributed optimization algorithms based on ADMM for solving big data optimization problems. We introduced the development of ADMM and describe several direct extensions and sophisticated modifications of ADMM from 2-block to N-block settings. The iterative schemes and convergence properties of those extensions/modifications were given, and implementations on large-scale computing facilities were also illustrated.

- We investigated big data processing techniques for smart grid security. In particular, we studied the sparse optimization for false data injection detection, which exploited intrinsic low dimensionality and sparsity of the data set, and the distributed approach for the security constrained optimal power flow, which scalably solved the large-scale optimization problem. Numerical simulations were conducted to validate the performance of the proposed algorithm.

- We considered big data traffic management in mobile networks. We proposed a distributed mechanism for mobile data offloading in software defined network at the network edge, and designed a decentralized approach for service request allocation and response routing in mobile cloud computing. Numerical simulations were performed to test proposed mechanisms.

The elaborate discussion of these contributions outlines the organization of this thesis. In Chapter 2, we review the mathematical background of the ADMM. The dual ascent method and the method of multipliers, two precedents of the ADMM, are introduced first. Then we describe the general form of ADMM and its relationship to the method of multipliers. After that, we review several state-of-the-art N-block ADMM algorithms. For each algorithm, the iterative update scheme is described and its convergence property is discussed. The implementations on large-scale computing facilities such as high performance computers and cloud computing infrastructures are illustrated. Finally, we summarize the relationships among reviewed algorithms.

In Chapter 3, the big data processing techniques for smart grid security are investigated. Two problems, the false data injection attacks detection for state estimation and the security constrained optimal power flow problem, are considered. The state estimation in the electric power grid is vulnerable to false data injection attacks, and diagnosing these kinds of malicious attacks has significant impact on ensuring reliable operations for power systems. By noticing the intrinsic low dimensionality of temporal measurements of power grid states, as well as the sparse nature of false data injection attacks, we propose a novel false data detection mechanism based on the separation of nominal power grid states and anomalies. Two methods, the nuclear norm minimization and the low rank matrix factorization, are presented to solve this problem. It is shown that proposed methods are able to identify proper power system operation states as well as detect malicious attacks, even under situations in which collected measurements are incomplete. Numerical simulation results, both on synthetic and real data, validate the effectiveness of proposed mechanisms. The second problem of security constrained optimal power flow determines the optimal control of power systems under constraints arising from a set of postulated contingencies. This problem is challenging due to the significantly large problem size, the stringent real-time requirement, and the variety of numerous post-contingency states. The ADMM is utilized to solve the resultant large-scale optimization problem with manageable complexity. The problem is decomposed into independent subproblems corresponding to pre-contingency and post-contingency cases. Each computing node addresses its local optimization problem, and computing nodes are coordinated through dual (prices) variables. Numerical tests validate the effectiveness of the proposed algorithm.

In Chapter 4, big data traffic management in mobile networks are considered. Two cases, the mobile data offloading in a software defined network, and the service management in mobile cloud computing, are studied. The mobile data offloading has been introduced to alleviate the congestion of cellular networks and improve the quality of service for mobile end users. We present a distributed mechanism for mobile data offloading in a software defined network at the network edge. The proposed mechanism is based on the proximal Jacobian multi-block ADMM. Base stations and access points perform offloading decision updates concurrently, and are coordinated by the software defined network controller through dual variables to reach a consensus on the offloading decision. Numerical simulations validate the effectiveness of the proposed algorithm. The second problem relates to the service traffic management in mobile cloud computing. The mobile cloud computing has been introduced to improve the performance of mobile applications by offloading data processing and storage from a mobile device to the cloud. By deploying service on several cloud-enabled data centers, the service provider can optimally locate service instances on the cloud to provide qualified services at a reasonable cost. However, a centralized approach for both request allocation and response routing does not scale due to the large number of mobile clients involved in the service management. Moreover, the random and unpredictable wireless network performance complicates the problem. We present a stochastic distributed optimization framework for mobile cloud services management, which takes the impact of random wireless network characteristics into account. Utilizing the ADMM, the optimization problem is decomposed into independent subproblems, which can be solved in a parallel fashion on distributed computing nodes. The convergence issue is addressed, and numerical tests validate the effectiveness of the proposed algorithm.

In chapter 5, we investigate the interdisciplinary research of big data optimization methods. In particular, we study the decentralized approach of the Gauss-Newton method for nonlinear least squares on a wide area network, and the compressive sensing framework for high-throughput hyperspectral imaging. Numerical simulations are performed to validate the effectiveness of proposed methods.

In Chapter 6, we conclude our work and explore possible extensions of our proposed big data optimization frameworks. Three potential applications, the distributed state estimation in electric power system, the efficient air quality monitoring in metropolitan city-regions, and the customer profiles extracting from smart meter reading data are described.

1.3 Published Results

The present Ph.D. work on big data optimization for modern communication networks has resulted in the publication of one book chapter [5] and three journal papers in the Institute of Electrical and Electronic Engineering (IEEE) Transaction on Smart Grid [6], System Journal [7] and European Alliance for Innovation Transactions on Wireless Spectrum [8]. The work has also been disseminated at pertinent conferences, where a total of seven articles have been accepted for presentation [9–15].

1.4 Notational Conventions

In this thesis, matrices are bold capital, vectors are bold lowercase and scalars or entries are not bold. The notation $\mathbf{x} = (\mathbf{x}_1^\top, \ldots, \mathbf{x}_n^\top)^\top$ is used to represent the column vector form by stacking vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$. For a block matrix $\mathbf{M}$, $(\mathbf{M})_{i,j}$ is used to denote the (i, j) block. The notation $\text{diag}(\mathbf{M}_1, \ldots, \mathbf{M}_n)$ is a diagonal matrix whose i^{th} diagonal block is $\mathbf{M}_i$, and the $\otimes$ denotes the Kronecker product. The identity matrix is denoted as $\mathbf{I}_N \in \mathbb{R}^{N \times N}$, where $\mathbb{R}$ denotes the real set. A $N \times 1$ column vector with all ones is denoted as $\mathbf{1}_N$. The $\|\mathbf{x}\|_2$ represents Euclidean norm of vector $\mathbf{x}$ and $\|\mathbf{X}\|_F$ represents the Frobenius norm of matrix $\mathbf{X}$. The norm of $\mathbf{x}$ with respective to a Hermitian positive definite matrix $\mathbf{G}$ is denoted as $\|\mathbf{x}\|_{\mathbf{G}}$. $\mathbf{X}^\top$, $\mathbf{X}^{-1}$, $\sigma_{\max}(X)$ and $\sigma_{\min}(X)$ denote the transpose, the inverse, the largest singular value, and the smallest nonzero singular value of matrix $\mathbf{X}$, respectively.

Chapter 2

Alternating Direction Method of Multipliers

In the era of big data, numerous problems in machine learning, compressed sensing, social network analysis, and computational biology formulate optimization problems with millions or billions of variables. Since classical optimization algorithms are not designed to scale to problems of this size, novel optimization algorithms are emerging to solve problems with big data. An incomprehensive list of such algorithms includes the block coordinate descent method [16–18][1], the stochastic gradient descent method [19–21], the dual coordinate ascent method [22, 23], the alternating direction method of multipliers (ADMM) [24, 25], and the Frank-Wolf method (also known as the conditional gradient method) [26, 27]. Each type of the algorithm enumerated has its own strength and weakness. The list is still growing and due to our limited knowledge and the fast development of this active field of research, many efficient algorithms are not mentioned here.

This chapter gives a brief introduction to the alternating direction method of multipliers (ADMM) for solving big data optimization problems in modern communication networks. The introduction focuses on explaining the algorithm itself along with its motivations and basic properties. We first introduce the background of ADMM in Section 2.1. We briefly review the dual ascent method and the method of multipliers, which provide useful backgrounds and motivations to ADMM. The canonical formulation of ADMM is also given. In Section 2.2, we focus on several direct extensions and sophisticated modifications of ADMM for large-scale optimization problems. The iterative schemes and convergence properties of those extensions/modifications are given, and implementations on large-scale computing facilities are also illustrated. Finally, Section 2.3 concludes the chapter.

[1] [18] proposes a stochastic block coordinate descent method.

2.1 From Dual Ascent to Alternating Direction Method of Multipliers

In this section, we first have a short overview of two important precursors of ADMM, the dual ascent method and the method of multipliers. Then we give the canonical form of ADMM.

2.1.1 Dual Ascent Method

Consider an optimization problem of the form

$$\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{A}\mathbf{x} = \mathbf{c}, \tag{2.1}$$

where $\mathcal{X} \subset \mathbb{R}^n$ is a closed convex set, $\mathbf{A} \in \mathbb{R}^{m\times n}$, and $f : \mathbb{R}^n \to \mathbb{R}$ is a closed convex proper function. The Lagrangian function $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ associated with the problem (2.1) is defined as

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \boldsymbol{\lambda}^\top(\mathbf{A}\mathbf{x} - \mathbf{c}), \tag{2.2}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^m$ is the Lagrangian multiplier associated with the equality constraint $\mathbf{A}\mathbf{x} = \mathbf{c}$. In the dual ascent method, the optimal solution $\mathbf{x}^*$ to the problem (2.1) is obtained by

$$\begin{cases} \mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}^k), \\ \boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \rho^k(\mathbf{A}\mathbf{x}^k - \mathbf{c}), \end{cases} \tag{2.3}$$

where $\rho^k > 0$ is the step size at iteration k. The convergence of the dual ascent method requires an appropriate step size ρ and assumptions of strong convexity as well as finiteness of the objective function f, which limit the spectrum of applications of the dual ascent method.

2.1.2 Method of Multipliers

The method of multipliers finds the optimal $\mathbf{x}^*$ of the constrained optimization problem (2.1) by solving a sequence of unconstrained problems. The augmented Lagrangian function for (2.1) is

$$\mathcal{L}_\rho(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \boldsymbol{\lambda}^\top(\mathbf{A}\mathbf{x} - \mathbf{c}) + \frac{\rho}{2}\|\mathbf{A}\mathbf{x} - \mathbf{c}\|_2^2, \tag{2.4}$$

where the term $\|\mathbf{A}\mathbf{x}-\mathbf{c}\|_2^2$ is called the augmentation, and $\rho > 0$ is the penalty parameter. Therefore, the method of multipliers is also called the augmented Lagrangian methods. In the method of

Algorithm 2.1 Method of multipliers.

Initialize: $\mathbf{x}^0$, $\boldsymbol{\lambda}^0$, $\rho > 0$;

for $k = 0, 1, \dots$ **do**
  $\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} \mathcal{L}_\rho(\mathbf{x}, \boldsymbol{\lambda}^k)$;
  $\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \rho(\mathbf{A}\mathbf{x}^k - \mathbf{c})$;
end for

multipliers, $\mathbf{x}$ and $\boldsymbol{\lambda}$ are updated iteratively as

$$\begin{cases} \mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} \mathcal{L}_\rho(\mathbf{x}, \boldsymbol{\lambda}^k), \\ \boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \rho(\mathbf{A}\mathbf{x}^k - \mathbf{c}), \end{cases} \tag{2.5}$$

where the penalty parameter $\rho > 0$ is fixed during the iteration, which balances the objective descent and constraint satisfaction. A proper update of ρ can noticeably accelerate the convergence.

The method of multipliers finds wide applications in sparse optimization problems. For example, consider the following l_1 norm minimization problem

$$\min_{\mathbf{x} \in \mathbb{R}^n} \| \mathbf{x} \|_1 \quad \text{s.t.} \quad \mathbf{A}\mathbf{x} = \mathbf{c}. \tag{2.6}$$

The iterative scheme of the method of multipliers for (2.6) is

$$\begin{cases} \mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} \| \mathbf{x} \|_1 + \boldsymbol{\lambda}^\top(\mathbf{A}\mathbf{x} - \mathbf{c}) + \frac{\rho}{2}\|\mathbf{A}\mathbf{x} - \mathbf{c}\|_2^2, \\ \boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \rho(\mathbf{A}\mathbf{x}^k - \mathbf{c}), \end{cases} \tag{2.7}$$

where the $\mathbf{x}$-update can be obtained analytically through the soft-thresholding. The method of multipliers returns a pair of primal-dual solutions at the end of iteration. For convex optimization problems, any $\rho > 0$ leads to the convergence. More details about the method of multipliers can be found in [28], and the iterative scheme is illustrated in Algorithm 2.1.

2.1.3 Alternating Direction Method of Multipliers

The ADMM was proposed in [29], [30] and recently revisited by [25]. The general form of ADMM is expressed as

$$\min_{\mathbf{x}_1 \in \mathcal{X}_1, \mathbf{x}_2 \in \mathcal{X}_2} f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2) \quad \text{s.t.} \quad \mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_2\mathbf{x}_2 = \mathbf{c}. \tag{2.8}$$

Algorithm 2.2 Two-block ADMM.

Initialize: $\mathbf{x}^0$, $\boldsymbol{\lambda}^0$, $\rho > 0$;

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

$\mathbf{x}_1^{k+1} = \arg\min_{\mathbf{x}_1} \mathcal{L}_\rho(\mathbf{x}_1, \mathbf{x}_2^k, \boldsymbol{\lambda}^k)$;

$\mathbf{x}_2^{k+1} = \arg\min_{\mathbf{x}_2} \mathcal{L}_\rho(\mathbf{x}_1^{k+1}, \mathbf{x}_2, \boldsymbol{\lambda}^k)$;

$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \rho(\mathbf{A}_1\mathbf{x}_1^{k+1} + \mathbf{A}_2\mathbf{x}_2^{k+1} - \mathbf{c})$;

end for

The augmented Lagrangian function for (2.8) is

$$\mathcal{L}_\rho(\mathbf{x}_1, \mathbf{x}_2, \boldsymbol{\lambda}) = f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2) + \boldsymbol{\lambda}^\top(\mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_2\mathbf{x}_2 - \mathbf{c}) + \frac{\rho}{2}\|\mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_2\mathbf{x}_2 - \mathbf{c}\|_2^2, \quad (2.9)$$

where $\boldsymbol{\lambda} \in \mathbb{R}^m$ is the Lagrangian multiplier, and $\rho > 0$ is the parameter for the quadratic penalty term. The iterative scheme of ADMM is

$$\begin{cases} \mathbf{x}_1^{k+1} = \arg\min_{\mathbf{x}_1} \mathcal{L}_\rho(\mathbf{x}_1, \mathbf{x}_2^k, \boldsymbol{\lambda}^k), \\ \mathbf{x}_2^{k+1} = \arg\min_{\mathbf{x}_2} \mathcal{L}_\rho(\mathbf{x}_1^{k+1}, \mathbf{x}_2, \boldsymbol{\lambda}^k), \\ \boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \rho(\mathbf{A}_1\mathbf{x}_1^{k+1} + \mathbf{A}_2\mathbf{x}_2^{k+1} - \mathbf{c}), \end{cases} \quad (2.10)$$

where at each step, the augmented Lagrangian function is minimized over $\mathbf{x}_1$ and $\mathbf{x}_2$, respectively. In (2.10), functions f_1 and f_2 are treated separately, so easier subproblems can be generated. This feature is quite attractive and advantageous for a broad spectrum of applications. The convergence of ADMM for convex optimization problems with two blocks of variables has been proven in [24], [25], and the iterative scheme is illustrated in Algorithm 2.2.

2.2 Multi-block Alternating Direction Method of Multipliers

In this section, we review several multi-block ADMM algorithms for solving large-scale optimization problems. The direct extensions of ADMM for convex optimization problems with N blocks of variables are first introduced. Then we introduce three sophisticated modifications of ADMM, the variable splitting ADMM [24,25,31], the ADMM with Gaussian back substitution [32] and the Proximal Jacobian ADMM [33, 34]. Specifically, we consider the following convex optimization problem with a canonical form as

$$\min_{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N} \quad f(\mathbf{x}) = f_i(\mathbf{x}_i) + \ldots + f_i(\mathbf{x}_N), \tag{2.11}$$

$$\text{s.t.} \quad \mathbf{A}_i\mathbf{x}_i + \ldots + \mathbf{A}_N\mathbf{x}_N = \mathbf{c}, \text{ and} \tag{2.12}$$

$$\mathbf{x}_i \in \mathcal{X}_i, \quad i = 1, \ldots, N, \tag{2.13}$$

where $\mathbf{x} = (\mathbf{x}_1^\top, \ldots, \mathbf{x}_N^\top)^\top$, $\mathcal{X}_i \subset \mathbb{R}^{n_i}(i = 1, 2, \ldots, N)$ are closed convex sets, $\mathbf{A}_i \in \mathbb{R}^{m\times n_i}(i = 1, 2, \ldots, N)$ are given matrices, $\mathbf{c} \in \mathbb{R}^m$ is a given vector, and $f_i : \mathbb{R}^{n_i} \to \mathbb{R}$ $(i = 1, 2, \ldots, N)$ are closed convex proper but not necessarily smooth functions. The non-smooth functions are usually employed to enforce structured solutions.

2.2.1 Direct Extensions to Multi-block Setting

We can directly extend the ADMM described in algorithm 2.2 to solve the optimization problem (2.11). In the following, we present two kinds of direct extensions, the Gauss-Seidel extension and the Jacobian extension, for problem (2.11). We first give the augmented Lagrangian function of problem (2.11)

$$\mathcal{L}_\rho(\mathbf{x}_1, \ldots, \mathbf{x}_N, \boldsymbol{\lambda}) = \sum_{i=1}^N f_i(\mathbf{x}_i) + \boldsymbol{\lambda}^\top(\sum_{i=1}^N \mathbf{A}_i\mathbf{x}_i - \mathbf{c}) + \frac{\rho}{2}\|\sum_{i=1}^N \mathbf{A}_i\mathbf{x}_i - \mathbf{c}\|_2^2. \tag{2.14}$$

2.2.1.1 The Gauss-Seidel Extension

Intuitively, a natural extension of the classical Gauss-Seidel type update of 2-block variables to N-block variables is straightforward. We can replace the two-block alternating minimization scheme by a sequential update of $\mathbf{x}_i$ for $i = 1, 2, \ldots, N$. In particular, at iteration k, $\mathbf{x}_i$ is updated by

$$\mathbf{x}_i = \arg\min_{\mathbf{x}_i} \mathcal{L}_\rho(\{\mathbf{x}_j^{k+1}\}_{j<i}, \mathbf{x}_i, \{\mathbf{x}_j^k\}_{j>i}, \boldsymbol{\lambda}^k), \tag{2.15}$$

where $\{\mathbf{x}_j\}_{j<i}$ denotes the set of variables prior to i. The augmented Lagrangian function is split and updated alternatingly. The direct Gauss-Seidel type extension is illustrated in Algorithm 2.3.

Algorithm 2.3 Gauss-Seidel multi-block ADMM.

Initialize: $\mathbf{x}^0$, $\boldsymbol{\lambda}^0$, $\rho > 0$;

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

  for $i = 1, \ldots, N$ **do**
    $\{\mathbf{x}_i$ is updated **sequentially**.$\}$
    $\mathbf{x}_i^{k+1} = \arg\min_{\mathbf{x}_i} \mathcal{L}_\rho(\{\mathbf{x}_j^{k+1}\}_{j<i}, \mathbf{x}_i, \{\mathbf{x}_j^k\}_{j>i}, \boldsymbol{\lambda}^k)$;
  end for
  $\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \rho(\sum_{i=1}^N \mathbf{A}_i \mathbf{x}_i^{k+1} - \mathbf{c})$;
end for

Algorithm 2.3 has been utilized in practical problems [35–37] despite a lack of rigourous proof for the convergence. Actually, the convergence of Gauss-Seidel multi-block ADMM is not well understood and is ambiguous for a long time: Neither affirmative convergence proof nor counter examples for convergence failure are shown in the literature. Recently, [38] has shown that the direct extension of Gauss-Seidel mulit-block ADMM is not necessarily convergent. In [39], the convergence of Algorithm 2.3 is proven with a sufficient small step size for Lagrangian multiplier update and additional assumptions on the problem (2.11). It is conjectured in [40] that an independent uniform random permutation of the update order for blocks in each iteration will result in a convergent iteration scheme. [32, 41] propose some slightly modified version of Algorithm 2.3 with provable convergence, competitive iteration simplicity, and computing efficiency. We will illustrate this later in Section 2.2.3.

2.2.1.2 The Jacobian Extension

Another possible iterative scheme for the N-block ADMM is the Jacobian type update, which performs the update of $\mathbf{x}_i$ in a parallel fashion for $i = 1, \ldots, N$. In particular, the update of $\mathbf{x}_i$ is calculated as

$$\mathbf{x}_i = \arg\min_{\mathbf{x}_i} \mathcal{L}_\rho(\mathbf{x}_i, \{\mathbf{x}_j^k\}_{j\neq i}, \boldsymbol{\lambda}^k), \tag{2.16}$$

where $\{\mathbf{x}_j^k\}_{j\neq i}$ denotes the set of variables except for $\mathbf{x}_i$. Different from the sequential update of $\mathbf{x}_i$ in Algorithm 2.3, the update in the Jacobian ADMM can be performed concurrently, i.e., all $\mathbf{x}_i$ can be calculated in a parallel fashion. This advantage makes the Jacobian type ADMM preferred for

Algorithm 2.4 Jacobian multi-block ADMM.

Initialize: $\mathbf{x}^0$, $\boldsymbol{\lambda}^0$, $\rho > 0$;

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

  for $i = 1, \ldots, N$ **do**

    $\{\mathbf{x}_i$ is updated **concurrently**.$\}$

    $\mathbf{x}_i^{k+1} = \arg\min_{\mathbf{x}_i} \mathcal{L}_\rho(\mathbf{x}_i, \{\mathbf{x}_j^k\}_{j \neq i}, \boldsymbol{\lambda}^k)$;

  end for

  $\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \rho(\sum_{i=1}^N \mathbf{A}_i \mathbf{x}_i^{k+1} - \mathbf{c})$;

end for

parallel implementation, and the direct Jacobian type extension is illustrated in Algorithm 2.4.

Though Algorithm 2.4 is more computational efficient in the sense of parallelization, [42] shows that Algorithm 2.4 is not necessarily convergent in the general case, even in the 2 blocks case. In [33] it is proven that if matrices $\mathbf{A}_i$ are mutually near-orthogonal and have full column-rank, the Algorithm 2.4 converges globally. A proximal Jacobian ADMM is also proposed in [33] with provable convergence, which we will illustrate later in Sec. 2.2.4

2.2.2 Variable Splitting ADMM

We can apply variable splitting [24, 25, 31, 43] for the multi-block variables to solve the optimization problem (2.11). In particular, the optimization problem (2.11) can be reformulated by introducing the auxiliary variable $\mathbf{z}$

$$\min_{\mathbf{x},\mathbf{z}} \quad \sum_{i=1}^N f_i(\mathbf{x}_i) + I_{\mathcal{Z}}(\mathbf{z}) \text{ and} \tag{2.17}$$

$$\text{s.t.} \quad \mathbf{A}_i \mathbf{x}_i + \mathbf{z}_i = \frac{\mathbf{c}}{N}, \quad i = 1, \ldots, N, \tag{2.18}$$

where $\mathbf{z} = (\mathbf{z}_1^\top, \ldots, \mathbf{z}_N^\top)^\top$ is partitioned conformably according to $\mathbf{x}$, and $I_{\mathcal{Z}}(\mathbf{z})$ is the indicator function of the convex set $\mathcal{Z}$, i.e., $I_{\mathcal{Z}}(\mathbf{z}) = 0$ for $\mathbf{z} \in \mathcal{Z} = \{\mathbf{z} | \sum_{i=1}^N \mathbf{z}_i = 0\}$ and $I_{\mathcal{Z}}(\mathbf{z}) = \infty$ otherwise. The augmented Lagrangian function is

$$\mathcal{L}_\rho = \sum_{i=1}^N f_i(\mathbf{x}_i) + I_{\mathcal{Z}}(\mathbf{z}) + \sum_{i=1}^N \boldsymbol{\lambda}_i^\top (\mathbf{A}_i \mathbf{x}_i + \mathbf{z}_i - \frac{\mathbf{c}}{N}) + \frac{\rho}{2} \sum_{i=1}^N \|\mathbf{A}_i \mathbf{x}_i + \mathbf{z}_i - \frac{\mathbf{c}}{N}\|_2^2, \tag{2.19}$$

Algorithm 2.5 Variable splitting multi-block ADMM.

Initialize: $\mathbf{x}^0$, $\mathbf{z}^0$, $\boldsymbol{\lambda}^0$, $\rho > 0$;

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

  for $i = 1, \dots, N$ **do**
    $\{\mathbf{x}_i$, $\mathbf{z}_i$ and $\boldsymbol{\lambda}_i$ are updated **concurrently**.$\}$
    $\mathbf{x}_i^{k+1} = \arg\min_{\mathbf{x}_i} \mathcal{L}_\rho(\mathbf{x}_i, \mathbf{z}_i^k, \boldsymbol{\lambda}_i^k)$;
    $\mathbf{z}_i^{k+1} = \arg\min_{\mathbf{z}_i} \mathcal{L}_\rho(\mathbf{x}_1^{k+1}, \mathbf{z}_i, \boldsymbol{\lambda}_i^k)$;
    $\boldsymbol{\lambda}_i^{k+1} = \boldsymbol{\lambda}_i^k + \rho(\mathbf{A}_i\mathbf{x}_i + \mathbf{z}_i - \frac{\mathbf{c}}{N})$;
  end for

end for

where we have two groups of variables, $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ and $\{\mathbf{z}_1, \dots, \mathbf{z}_N\}$. Hence, we can apply the two-block ADMM to update these two groups of variables iteratively, i.e., we can first update the group $\{\mathbf{x}_i\}$ and then update the group $\{\mathbf{z}_i\}$. In each group, $\mathbf{x}_i$ and $\mathbf{z}_i$ can be updated concurrently in parallel at each iteration. In particular, the update rules for $\mathbf{x}_i$ and $\mathbf{z}_i$ are

$$\begin{cases} \mathbf{x}_i^{k+1} = \arg\min_{\mathbf{x}_i} \mathcal{L}_\rho(\mathbf{x}_i, \mathbf{z}_i^k, \boldsymbol{\lambda}_i^k), \\ \mathbf{z}_i^{k+1} = \arg\min_{\mathbf{z}_i} \mathcal{L}_\rho(\mathbf{x}_1^{k+1}, \mathbf{z}_i, \boldsymbol{\lambda}_i^k), \quad \forall i = 1, \dots, N, \\ \boldsymbol{\lambda}_i^{k+1} = \boldsymbol{\lambda}_i^k + \rho(\mathbf{A}_i\mathbf{x}_i + \mathbf{z}_i - \frac{\mathbf{c}}{N}). \end{cases} \tag{2.20}$$

The variable splitting ADMM is illustrated in Algorithm 2.5. Algorithm 2.5 converges to the optimal solution with the same rate as the 2-block ADMM. However, the number of variables and constraints will increase substantially when N is large, which will impact the efficiency and incur significant burden for computation.

2.2.3 ADMM with Gaussian Back-Substitution

Many efforts have been made to enable the convergence of the Guass-Seidel type multi-block ADMM [32, 41]. In this part, we describe the ADMM with Gaussian back-substitution [32], which asserts that if a new update is generated by correcting the output of Algorithm 2.3 with a Gaussian back-substitution procedure, then the sequence of updates converge to a solution of problem (2.11). We first define vector $\mathbf{v} = (\mathbf{x}_2^\top, \dots, \mathbf{x}_N^\top, \boldsymbol{\lambda}^\top)^\top$, vector $\tilde{\mathbf{v}} = (\tilde{\mathbf{x}}_2^\top, \dots, \tilde{\mathbf{x}}_N^\top, \tilde{\boldsymbol{\lambda}}^\top)^\top$, matrix $\mathbf{H} = \text{diag}(\rho\mathbf{A}_2^\top\mathbf{A}_2, \ldots, \rho\mathbf{A}_N^\top\mathbf{A}_N, \frac{1}{\rho}\mathbf{I}_m)$ and $\mathbf{M}$ as

$$\mathbf{M} = \begin{pmatrix} \rho\mathbf{A}_2^\top\mathbf{A}_2 & 0 & \ldots & \ldots & 0 \\ \rho\mathbf{A}_3^\top\mathbf{A}_2 & \rho\mathbf{A}_3^\top\mathbf{A}_3 & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \rho\mathbf{A}_N^\top\mathbf{A}_2 & \rho\mathbf{A}_N^\top\mathbf{A}_3 & \ldots & \rho\mathbf{A}_N^\top\mathbf{A}_N & 0 \\ 0 & 0 & \ldots & 0 & \frac{1}{\rho}\mathbf{I}_m \end{pmatrix}. \tag{2.21}$$

Each iteration of the ADMM with Gaussian back substitution consists of two procedures, a prediction procedure and a correction procedure. The $\tilde{\mathbf{v}}$ is generated by Algorithm 2.3. In particular, $\tilde{\mathbf{x}}_i$ is updated sequentially as

$$\tilde{\mathbf{x}}_i^k = \arg\min_{\tilde{\mathbf{x}}_i} \mathcal{L}_\rho(\{\tilde{\mathbf{x}}_j^k\}_{j<i}, \mathbf{x}_i, \{\mathbf{x}_j^k\}_{j>i}, \boldsymbol{\lambda}^k), \tag{2.22}$$

where the prediction procedure is performed in a forward manner, i.e., from the first to the last block and to the Lagrangian multiplier. Note that the newly-generated $\tilde{\mathbf{x}}_i$ is used in the update of the next block in accordance with the Gauss-Seidel update fashion. After the update of the Lagrangian multiplier, the correction procedure is performed to update $\mathbf{v}$ using

$$\mathbf{H}^{-1}\mathbf{M}^\top(\mathbf{v}^{k+1} - \mathbf{v}^k) = \alpha(\tilde{\mathbf{v}}^k - \mathbf{v}^k), \tag{2.23}$$

where $\mathbf{H}^{-1}\mathbf{M}^\top$ is a upper-triangular block matrix according to definitions of $\mathbf{H}$ and $\mathbf{M}$. This implies that the update of the correction procedure is in a backward fashion, i.e., first update the Lagrangian multiplier, and then update $\mathbf{x}_i$ from the last block to the first block sequentially. Note that an additional assumption regarding $\mathbf{A}_i^\top\mathbf{A}_i (i = 1, 2, \ldots, N)$ being nonsingular is made here. $\mathbf{x}_1$ serves as an intermediate variable and is unchanged during the correction procedure. The algorithm is illustrated in Algorithm 2.6, and its global convergence is proven in [32].

2.2.4 Proximal Jacobian ADMM

The other type of modification on the ADMM for multiple blocks of variables is based on the Jacobian iteration scheme [33, 34, 42, 44]. Since the Guass-Seidel update is performed sequentially

Algorithm 2.6 The ADMM with Gaussian back-substitution.

Initialize: $\mathbf{x}^0$, $\tilde{\mathbf{x}}^0$, $\boldsymbol{\lambda}^0$, $\tilde{\boldsymbol{\lambda}}^0$, $\rho > 0$, $\alpha \in (0, 1)$;
for $k = 0, 1, \dots$ **do**
  for $i = 1, \dots, N$ **do**
    $\{\mathbf{x}_i$ is updated **sequentially**.$\}$
    $\tilde{\mathbf{x}}_i^k = \arg\min_{\tilde{\mathbf{x}}_i} \mathcal{L}_\rho(\{\tilde{\mathbf{x}}_j^k\}_{j<i}, \mathbf{x}_i, \{\mathbf{x}_j^k\}_{j>i}, \boldsymbol{\lambda}^k)$;
  end for
  $\tilde{\boldsymbol{\lambda}}^{k+1} = \boldsymbol{\lambda}^k + \rho(\sum_{i=1}^N \mathbf{A}_i \tilde{\mathbf{x}}_i^{k+1} - \mathbf{c})$;

  $\{$Gaussian back substitution correction step$\}$
  $\mathbf{H}^{-1}\mathbf{M}^\top(\mathbf{v}^{k+1} - \mathbf{v}^k) = \alpha(\tilde{\mathbf{v}}^k - \mathbf{v}^k)$;
  $\mathbf{x}_1^{k+1} = \tilde{\mathbf{x}}_1^k$;
end for

and is not amenable for parallelization, Jacobian type iteration is preferred by distributed and parallel optimization methods. In this part we describe the proximal Jacobian ADMM [33], in which a proximal term [45] is added to the update to improve convergence. In particular, the update of $\mathbf{x}_i$ is

$$\mathbf{x}_i^{k+1} = \arg\min_{\mathbf{x}_i} \mathcal{L}_\rho(\mathbf{x}_i, \{\mathbf{x}_j^k\}_{j \neq i}, \boldsymbol{\lambda}^k) + \frac{1}{2}\|\mathbf{x}_i - \mathbf{x}_i^k\|_{\mathbf{P}_i}^2, \tag{2.24}$$

where $\|\mathbf{x}_i\|_{\mathbf{P}_i}^2 = \mathbf{x}_i^\top \mathbf{P}_i \mathbf{x}_i$ for symmetric and positive semi-definite matrix $\mathbf{P}_i \succeq 0$. The involvement of the proximal term can make the subproblem of $\mathbf{x}_i$ strictly or strongly convex, and thus make the problem more stable. Moreover, multiple choices of $\mathbf{P}_i$ can make the subproblems easier to solve. The update of the Lagrangian multiplier is

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \gamma\rho(\sum_{i=1}^N \mathbf{A}_i \mathbf{x}_i^{k+1} - \mathbf{c}), \tag{2.25}$$

where $\gamma > 0$ is the damping parameter. The algorithm is illustrated in Algorithm 2.7. The global convergence of the proximal Jacobian ADMM is proven in [33]. Moreover, it enjoys a convergence rate of $o(1/k)$ under conditions on $\mathbf{P}_i$ and γ. More details can be found in [33].

2.2.5 Implementations

The recent developments in high performance computing (HPC) and cloud computing provide flexible and efficient solutions for implementing large-scale optimization algorithms. In this part,

Algorithm 2.7 Proximal Jacobian ADMM.

Initialize: $\mathbf{x}^0$, $\boldsymbol{\lambda}^0$, $\rho > 0$, $\gamma > 0$;

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

  for $i = 1, \dots, N$ **do**
    $\{\mathbf{x}_i$ is updated **concurrently**.$\}$
    $\mathbf{x}_i^{k+1} = \arg\min_{\mathbf{x}_i} \mathcal{L}_\rho(\mathbf{x}_i, \{\mathbf{x}_j^k\}_{j\neq i}, \boldsymbol{\lambda}^k) + \frac{1}{2}\|\mathbf{x}_i - \mathbf{x}_i^k\|_{\mathbf{P}_i}^2$;
  end for
  $\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \gamma\rho(\sum_{i=1}^N \mathbf{A}_i\mathbf{x}_i^{k+1} - \mathbf{c})$;
end for

we describe possible implementation approaches of those distributed and parallel algorithms on current mainstream, large-scale computing facilities.

One possible implementation utilizes available computing-incentive techniques and tools like MPI, OpenMP, and OpenCL. The MPI is a language-independent protocol used for inter-process communications on distributed memory computing platform. It is widely used for high-performance parallel computing today. The (multi-block) ADMM using MPI has been implemented in [25] and [46]. Besides, the OpenMP, which is a shared memory multiprocessing parallel computing paradigm, and the OpenCL, which is a heterogenous distributed-shared memory parallel computing paradigm incorporating CPUs and GPUs, can also implement distributed and parallel optimization algorithms. It is expected that supercomputers will reach one exaFLOPS (10^{18} FLOPS) and even one zettaFLOPS (10^{21} FLOPS) in the near feature, which will largely enhance the computing capacity and significantly expedite the program execution.

Another possible approach exploits the ease-of-use cloud computing engine like Hadoop MapReduce and Apache Spark. The cloud computing infrastructure available for Hadoop MapReduce makes it convenient to use for large-scale problems, though it is awkward to implement ADMM using MapReduce since it is not designed for iterative tasks. The in-memory computing feature of Apache Spark enables it to run iterative computations much faster. It is now prevalent for large-scale machine learning and optimization tasks on computer clusters [47]. This implementation approach is much simpler than previous computing-incentive techniques and tools. The advance in the cloud/cluster computing engine provides a simple method to implement the large-scale data processing. Recently Google, Baidu and Alibaba are developing and deploying massive

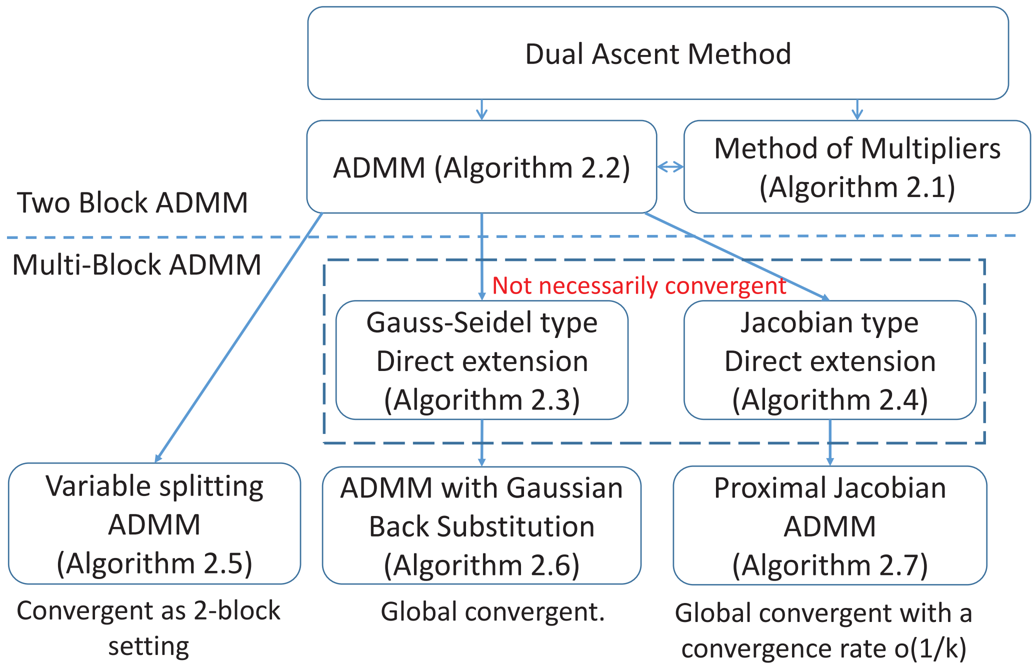


Figure 2.1 An illustration of the relationships between Algorithms $2.1 - 2.7$.

cloud computing engines to perform the large-scale distributed and parallel computation.

2.3 Conclusion

In this chapter, we have given an introduction of ADMM for big data optimization problems. We have described precursors of ADMM and their background. After that, several direct extensions and sophisticated modifications of ADMM have been introduced for large-scale optimization problems. We have explained iterative schemes and convergence properties for those extensions/modifications, and have illustrated implementations on large-scale computing facilities. The relationships among algorithms introduced in this chapter can be summarized in Fig. 2.1.

Chapter 3

Ensuring Power Grids Security Using Big Data

The development of the smart grid, impelled by the increasing demand from industrial and residential customers together with the aging power infrastructure, has become an urgent global priority due to its potential economic, environmental, and societal benefits. The smart grid refers to the next generation electric power system which aims to provide reliable, efficient, secure, and quality energy generation/distribution/consumption using modern information, communications, and electronics technologies. A distributed and user centric system will be introduced in the smart grid, which will incorporate end-consumers into its decision processes to provide a cost-effective and reliable energy supply. In the smart grid, the modern communication infrastructure [48] will play a vital role in managing, controlling, and optimizing different devices and systems. Information and communication technologies will offer the power grid with the capability of supporting two-way energy and information flows, quick isolating and restoring power outages, facilitating the integration of renewable energy sources into the grid and empowering the consumer with tools for optimizing their energy consumption.

In this chapter, the applications of big data processing techniques for the smart grid security are investigated from two perspectives: how to exploit the inherent structure of the data, and how to deal with the huge size of the data sets. Two specific applications are included in this chapter: the sparse optimization for false data injection detection, and the distributed parallel approach for the security constrained optimal power flow (SCOPF) problem. The rest of this chapter is organized as follows. The sparse optimization for false data injection detection is described in Section 3.1. The distributed parallel approach for the security constrained optimal power flow problem is developed in Section 3.2. Finally, some conclusions are drawn in Section 3.3.

3.1 Sparse Optimization for False Data Injection Detection

In this section, we first introduce the state estimation and false data injection attacks in power systems. Then we describe two detection methods, the nuclear norm minimization and the low rank matrix factorization, which exploit inherent structure of state estimation data to detect false data injection attacks. Finally we present numerical simulation results of proposed methods.

3.1.1 State Estimation and False Data Injection Attacks

3.1.1.1 State Estimation in Power systems

State estimation [49], which estimates the power system operating state based on a real-time electric network model, is a key function of the Energy Management System (EMS). A linearized measurement model is often used to estimate states in power systems based on measurements from remote meters on buses or transmission lines. Specifically, every several seconds or minutes, the Energy Control Center (ECC) collects active/reactive power flows and injections from transmission lines and buses across the power grid as measurement data via the Supervisory Control and Data Acquisition (SCADA) system. The state estimation results reflect the real-time power grid operation state and are essential for operators to make decisions in order to maintain security and stability of the system.

In an electric power grid, the control center needs to monitor the voltage phase angles of all buses to make real-time decisions on operations. However, it is impractical to directly measure all bus voltage phase angles. In this regard, the control center collects readings from remote electric meters to estimate the system operation state. Specific measurement data include branch active power flows and bus active power injections, which can be used to estimate bus voltage angles in the system. Let $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_n)^\top$ denotes the power system state variables, where θ_i is the phase angle on bus i. The measurement at the control center is expressed as $\mathbf{z} = (z_1, z_2, \ldots, z_m)^\top$ and is related to $\boldsymbol{\theta}$ by

$$\mathbf{z} = \mathbf{h}(\boldsymbol{\theta}) + \mathbf{e}, \tag{3.1}$$

where $\mathbf{h}(\boldsymbol{\theta}) = (h_1(\boldsymbol{\theta}), h_2(\boldsymbol{\theta}), \ldots, h_m(\boldsymbol{\theta}))^\top$, and $h_i(\boldsymbol{\theta})$ is a nonlinear function relating the i^{th} measurement to the state vector $\boldsymbol{\theta}$. The vector $\mathbf{e}$ denotes independent Gaussian measurement errors with zero mean and known covariance $\mathbf{R}$.

To analyze the efficiency of various state estimation methods considering the measurement configuration in a power system, a simplified DC approximation model is utilized. Assuming that bus voltage magnitudes are already known and normalized, and neglecting all shunt elements and branch resistances, the active power flow from bus i to bus j can be approximated[1] [50] by the first-order Taylor expansion as

$$P_{ij} = \frac{\theta_i - \theta_j}{X_{ij}} + \omega, \tag{3.2}$$

where X_{ij} is the reactance of the transmission line between bus i and bus j, and ω is the measurement error. Similarly, the power injection measurement at bus i can be expressed as

$$P_i = \sum_j P_{ij} + \nu, \tag{3.3}$$

where ν is the measurement error.

The DC model for real power measurements can be written in a linear matrix form as

$$\mathbf{z} = \mathbf{H}\boldsymbol{\theta} + \mathbf{e}, \tag{3.4}$$

where $\mathbf{z}$ is the measurement vector including active power flows and injection measurements, and $\mathbf{H} \in \mathbb{R}^{m \times n}$ is the Jacobian matrix of the power system, which is assumed to be known to the independent system operator (ISO).

Suppose that measurement errors $\mathbf{e}$ in (3.4) are not correlated, and thus the covariance matrix $\mathbf{R}$ is a diagonal matrix. The weighted least squares estimator of the linearized state vector $\boldsymbol{\theta}$ is

$$\hat{\boldsymbol{\theta}} = (\mathbf{H}^\top \mathbf{R}^{-1} \mathbf{H})^{-1} \mathbf{H}^\top \mathbf{R}^{-1} \mathbf{z}. \tag{3.5}$$

Let $\mathbf{K} = (\mathbf{H}^\top \mathbf{R}^{-1} \mathbf{H})^{-1} \mathbf{H}^\top \mathbf{R}^{-1}$, and then measurement residuals can be expressed as

$$\mathbf{r} = \mathbf{z} - \mathbf{H}\hat{\boldsymbol{\theta}} = (\mathbf{I} - \mathbf{K})(\mathbf{H}\boldsymbol{\theta} + \mathbf{e}) = (\mathbf{I} - \mathbf{K})\mathbf{e}, \tag{3.6}$$

[1] In general, one can approximate the impedance of a transmission line with its reactance due to the high reactance over resistance (X/R) ratio.

where $\mathbf{I}$ is the identity matrix and the matrix $(\mathbf{I} - \mathbf{K})$ is called the residual sensitivity matrix.

The detection and identification of bad data in measurements can be accomplished by processing of the measurement residuals. Specifically, the χ^2-Test can be applied on measurement residuals to detect bad data. Regarding the detection of bad data, two kinds of methods, the largest normalized residual test and the hypothesis testing identification method, can be used to identify the specific measurement that actually contains bad data [49].

3.1.1.2 False Data Injection Attacks

The accuracy of state estimation can be affected by bad measurements in the grid. Bad data could be due to topology errors in the grid, measurement abnormalities caused by meter failures, or malicious attacks. To detect and identify bad measurements in the power grid state, techniques based on the statistical testing of measurement residuals [49] have been developed and are widely used. However, [51] reveals the fact that false data injection attacks are able to circumvent traditional detection methods based on residual testing. By exploiting the configuration of a power system, synchronized data injection attacks on meters can be launched to tamper with their measurements. Moreover, attack vectors can be systematically and efficiently constructed even when the attacker is limited in resources required to compromise meters, which will mislead the state estimation process, and thus affect power grid control algorithms. Hence, attention should be given over the vulnerability of state estimation to false data injection attacks, which may cause catastrophic consequences in the power grid.

Malicious attack vectors are able to circumvent existing statistical tests for bad data detection if they leave measurement residuals unchanged. One such example is the false data injection attack, which is defined as follows:

Definition 3.1. *(False data injection attack) [51] The malicious attack vector* $\mathbf{a} = (a_1, a_2, ..., a_m)^\top$ *is called a false data injection attack if* $\mathbf{a}$ *can be expressed as a linear combination of columns of* $\mathbf{H}$*; i.e.,* $\mathbf{a} = \mathbf{Hc}$ *for some vector* $\mathbf{c}$*.*

If a false data injection attack is applied to the power system, the collected measurements at the ISO can be expressed as

$$\mathbf{z_a} = \mathbf{z_0} + \mathbf{a} = \mathbf{H}(\boldsymbol{\theta} + \mathbf{c}) + \mathbf{e}. \tag{3.7}$$

Suppose the state estimate using the malicious measurement $\mathbf{z_a}$ is $\boldsymbol{\theta}_\mathbf{a}$, the norm of measurement residuals $||\mathbf{z_a} - \mathbf{H}\boldsymbol{\theta}_\mathbf{a}||_2$ in this case is

$$||\mathbf{z_a} - \mathbf{H}\boldsymbol{\theta}_\mathbf{a}||_\mathbf{2} = ||\mathbf{z_0} + \mathbf{a} - \mathbf{H}(\boldsymbol{\theta} + \mathbf{c})||_\mathbf{2} = ||\mathbf{z_0} - \mathbf{H}\boldsymbol{\theta}||_\mathbf{2}, \tag{3.8}$$

which means that measurement residuals are unaffected by the injection attack vector $\mathbf{a}$, and the attacker successfully tricks the system into believing that the true state is $\boldsymbol{\theta}_\mathbf{a} = \boldsymbol{\theta} + \mathbf{c}$ instead of $\boldsymbol{\theta}$. Note that $\mathbf{a}$ is the attack vector, which is under the control of attackers, while $\mathbf{c}$ reflects error induced by $\mathbf{a}$.

Unveiling false data injection attacks is crucial to security and reliability of power systems. This task is challenging, since attackers may be able to construct false data attack vectors against the protection scheme, and inject attack vectors into the power grid that can bypass traditional methods for bad measurement detection. Furthermore, the incomplete measurement data due to intended attacks or meter failures complicates the task of malicious attack detection, and thus makes state estimation even more difficult.

The effects of false data injection attacks have been studied in [51–53]. False data injection attacks against state estimation in electric power grid were presented in [51]. By capitalizing on the configuration of the power system, malicious attacks can be launched to bypass the existing bad measurement detection techniques and manipulate results of state estimation. [52, 53] demonstrated that false data injection attacks were able to circumvent bad data identification techniques equipped in the EMS, and could lead to congestion of transmission lines as well as profitable financial misconduct in the power market.

On the other hand, schemes to protect against false data injection attacks are investigated in [13, 54–59]. [54] proposed an efficient method for computing the security index with sparse attack vectors, and described a protection scheme to strengthen system security by placing encrypted devices in the electric power grid appropriately. [55] modeled and analyzed this situation as a zero-sum game between attackers and defenders. [56] characterized two kinds of malicious attacks on electric power grids: the strong attack regime, in which false data injection attacks exist, and the weak attack regime, in which the generalized likelihood ratio test can be used to detect attacks. [13] formulated the bad data detection problem as a low-rank matrix recovery problem, which is solved by a convex optimization method that minimizes a combination of the nuclear norm and the $l1$ norm. In [57], a low-complexity attacking strategy was designed to construct sparse false data injection attack vectors, and strategic protection schemes were also proposed based on greedy approaches. [58] provided a survey of existing detection methods for false data injection attacks, and [59] studied the fundamental limits of cyber-physical security in presence of false data injection attacks in the system.

3.1.1.3 Sparse Optimization Problem Formulation

Denote the measurement of the electric power system observed by the ISO at time k as $\mathbf{z}_k$. In presence of false data injection attacks, the measurement $\mathbf{z_k}$ is contaminated by the attack vector $\mathbf{a_k}$. Denote $\mathbf{Z_0} = [\mathbf{z_1}, \mathbf{z_2}, \ldots, \mathbf{z_t}] \in \mathbb{R}^{m\times t}$ as the measurement of the power state for a time period of t, and $\mathbf{A} = [\mathbf{a_1}, \mathbf{a_2}, \ldots, \mathbf{a_t}] \in \mathbb{R}^{m\times t}$ as the false data attack matrix. The obtained temporal observations $\mathbf{Z_a}$ can be expressed as

$$\mathbf{Z_a} = \mathbf{Z_0} + \mathbf{A}. \tag{3.9}$$

Note that gradually changing power system state variables will typically lead to a low-rank measurement matrix $\mathbf{Z_0}$. In addition, due to the capability limitation of attackers, they are either constrained to some specific measurement meters or unable to compromise measurement meters persistently. Hence, only a small fraction of observations can be anomalous at a given time instant. This implies that the false data injection matrix $\mathbf{A}$ is sparse across both rows and columns. With a slight abuse of notation, we use $\text{Rank}(\mathbf{Z_0})$ to denote the rank of the matrix $\mathbf{Z_0}$, and $\|\mathbf{A}\|_0$ to represent the number of nonzero entries of the matrix $\mathbf{A}$. Noticing intrinsic structures of $\mathbf{Z_0}$ and $\mathbf{A}$, the detection and identification of false data injection attacks can be converted to a matrix separation problem as

$$\min_{\mathbf{Z_0},\mathbf{A}} \text{Rank}(\mathbf{Z_0}) + \|\mathbf{A}\|_0, \quad s.t. \quad \mathbf{Z_a} = \mathbf{Z_0} + \mathbf{A}. \tag{3.10}$$

Solving (3.10) extracts the power state measurement matrix $\mathbf{Z_0}$ and the sparse attack matrix $\mathbf{A}$ from their sum $\mathbf{Z_a}$. Considering missing measurements due to meter failures or communication link outages in practical applications, (3.10) can be formulated as

$$\min_{\mathbf{Z_0},\mathbf{A}} \text{Rank}(\mathbf{Z_0}) + \|\mathbf{A}\|_0, \quad s.t. \quad \mathcal{P}_{\mathbf{\Omega}}(\mathbf{Z_a}) = \mathcal{P}_{\mathbf{\Omega}}(\mathbf{Z_0} + \mathbf{A}), \tag{3.11}$$

where $\mathbf{\Omega}$ is an index subset, and $\mathcal{P}_{\mathbf{\Omega}}(\cdot)$ is the projection operator. Specifically, $\mathcal{P}_{\mathbf{\Omega}}(\mathbf{M})$ is the projection of a matrix $\mathbf{M}$ onto the subspace of matrices whose non-zeros entries are restricted to $\mathbf{\Omega}$

$$[\mathcal{P}_{\mathbf{\Omega}}(\mathbf{M})]_{ij} = 0, \quad \forall (i,j) \notin \mathbf{\Omega}. \tag{3.12}$$

In the following, we propose two methods to solve this problem.

3.1.2 Nuclear Norm Minimization

The optimization problem in (3.10) captures the low rank property of the power state measurement matrix $\mathbf{Z_0}$ as well as the sparseness of the malicious attack matrix $\mathbf{A}$. However, it is known to be impractical to directly solve (3.10). One possible approach is to replace Rank$(\mathbf{Z_0})$ and $\|\mathbf{A}\|_0$ with their convex relaxations, $\|\mathbf{Z_0}\|_*$ and $\|\mathbf{A}\|_1$, respectively. Here, $\|\mathbf{Z_0}\|_*$ is the nuclear norm of $\mathbf{Z_0}$, which is the sum of its singular values, and $\|\mathbf{A}\|_1$ is the l_1 norm of $\mathbf{A}$, which is the sum of absolute values of its entries. Hence, (3.10) can be reformulated as the following convex optimization problem

$$\min_{\mathbf{Z_0},\mathbf{A}} \|\mathbf{Z_0}\|_* + \lambda\|\mathbf{A}\|_1, \quad s.t. \quad \mathbf{Z_a} = \mathbf{Z_0} + \mathbf{A}, \tag{3.13}$$

where λ is a regularization parameter. Correspondingly, (3.11) can be reformulated as

$$\min_{\mathbf{Z_0},\mathbf{A}} \|\mathbf{Z_0}\|_* + \lambda\|\mathbf{A}\|_1, \quad s.t. \quad \mathcal{P}_{\mathbf{\Omega}}(\mathbf{Z_a}) = \mathcal{P}_{\mathbf{\Omega}}(\mathbf{Z_0} + \mathbf{A}). \tag{3.14}$$

The optimization problem in (3.14) has been extensively studied in fields of compressive sensing [60] and matrix completion [61, 62], and can be solved by many off-the-shelf convex optimization algorithms. Motivated by [63], the method of multipliers is utilized here to detect the false data matrix $\mathbf{A}$ as well as to recover the measurement matrix $\mathbf{Z_0}$. According to Algorithm 2.1, the optimization problem in (3.14) can be solved iteratively via the method of multipliers [64], where the augmented Lagrangian for (3.14) is given by

$$L(\mathbf{Z_0}, \mathbf{A}, \mathbf{Y}, \mu) = \|\mathbf{Z_0}\|_* + \lambda\|\mathbf{A}\|_1 + \langle \mathbf{Y}, \mathcal{P}_{\mathbf{\Omega}}(\mathbf{Z_a} - \mathbf{Z_0} - \mathbf{A})\rangle + \frac{\mu}{2}\|\mathcal{P}_{\mathbf{\Omega}}(\mathbf{Z_a} - \mathbf{Z_0} - \mathbf{A})\|_2^2. \quad (3.15)$$

The value of λ is set to $\frac{1}{\sqrt{\max(m,t)}}$, where m and t are dimensions of the measurement matrix $\mathbf{Z_a}$. With $k = 1, 2, \ldots$, indexing iterations, optimal $\mathbf{Z_0}$ and $\mathbf{A}$ are found according to

$$\mathbf{A}^{k+1} = \arg\min_{\mathbf{A}} L(\mathbf{Z_0}^k, \mathbf{A}, u^k, \mathbf{Y}^k) \text{ and} \quad (3.16)$$

$$\mathbf{Z_0}^{k+1} = \arg\min_{\mathbf{Z_0}} L(\mathbf{Z_0}, \mathbf{A}^k, u^k, \mathbf{Y}^k), \quad (3.17)$$

where (3.16) can be explicitly computed from the soft-shrinkage formula, and (3.17) can be solved via the singular value shrinkage operator [65]. Specifically, we define this operator as $\mathcal{S}_\tau\{x\} = \text{sgn}(x)\max(|x| - \tau, 0)$ for a real variable x, where sgn is the sign function. This operator can be extended to vectors and matrices by applying it element-wise. Using this operator, (3.16) can be solved iteratively via

$$\mathbf{A}^{k+1} = \mathcal{S}_{\frac{\lambda}{u^k}}\{\mathbf{Z_a} - \mathbf{Z_0}^k + \frac{\mathbf{Y}^k}{u^k}\}. \quad (3.18)$$

To solve (3.17), a singular value decomposition (SVD) is applied to the matrix $(\mathbf{Z_a} - \mathbf{A}^{k+1} + \frac{\mathbf{Y}^k}{u^k})$:

$$(\mathbf{Z_a} - \mathbf{A}^{k+1} + \frac{\mathbf{Y}^k}{u^k}) = \mathbf{U}\mathbf{S}\mathbf{V}^\top, \quad (3.19)$$

where $\mathbf{U} \in \mathbb{R}^{m\times m}$ and $\mathbf{V} \in \mathbb{R}^{t\times t}$ are unitary matrices, and $\mathbf{S} \in \mathbb{R}^{m\times t}$ is a diagonal matrix containing the singular values of $(\mathbf{Z_a} - \mathbf{A}^{k+1} + \frac{\mathbf{Y}^k}{u^k})$. The singular values are arranged in a decreasing order, and $\mathbf{Z_0}$ is updated via

$$\mathbf{Z_0}^{k+1} = \mathbf{U}\mathcal{S}_{\frac{1}{u^k}}\{\mathbf{S}\}\mathbf{V}^\top. \quad (3.20)$$

During each iteration of the optimization, both Lagrange multipliers $\mathbf{Y}$ and μ are updated,

Algorithm 3.1 Nuclear norm minimization approach.

Input: $\mathbf{Z_a} \in \mathbb{R}^{m\times t}$; $\lambda = \frac{1}{\sqrt{\max(m,t)}}$;
Initialize: $\mathbf{Y}_{[0]} = 0$; $\mathbf{Z_0}_{[0]} = 0$; $\mathbf{A}_{[0]} = 0$; $\mu_{[0]} > 0$; $\alpha > 0$; $k = 0$;
while not converge **do**
$\mathbf{Z_0}^{k+1} = \mathbf{Z_0}^k$; $\mathbf{A}^{k+1} = \mathbf{A}^k$; $j = 0$;
$\mathbf{Y}^{k+1} = \mathbf{Y}^k + u^k(\mathbf{Z_a} - \mathbf{Z_0}^{k+1} - \mathbf{A}^{k+1})$;
$\mu^{k+1} = \alpha\mu^k$;

$k = k + 1$;

end while
return $\mathbf{Z_0}^k$; $\mathbf{A}^k$;
Output $\mathbf{Z_0}^k$; $\mathbf{A}^k$;

which improves the performance of the algorithm

$$\mathbf{Y}^{k+1} = \mathbf{Y}^k + u^k(\mathbf{Z_a} - \mathbf{Z_0}^{k+1} - \mathbf{A}^{k+1}) \text{ and} \tag{3.21}$$

$$\mu^{k+1} = \alpha\mu^k, \tag{3.22}$$

where α is a positive constant. The algorithm is outlined as Algorithm 3.1.

3.1.3 Low Rank Matrix Factorization

The speed and scalability of the nuclear norm minimization approach are limited by the computational complexity of singular value decomposition. When matrix size and rank increase, computational operations for singular value decomposition will become quite expensive. To improve the scalability of solving large-scale problems of malicious attack detection in power systems, a low rank matrix factorization approach is proposed here.

Given observations $\mathbf{Z_a}$, the measurements $\mathbf{Z_0}$ and the false data injection attack matrix $\mathbf{A}$ can be separated by the minimization problem

$$\min_{\mathbf{U},\mathbf{V},\mathbf{Z_0}} \|\mathbf{Z_a} - \mathbf{Z_0}\|_1, \quad s.t. \quad \mathbf{UV} - \mathbf{Z_0} = \mathbf{0}, \tag{3.23}$$

where the low rank matrix $\mathbf{Z_0}$ is expressed as a product of $\mathbf{U} \in \mathbb{R}^{m\times r}$ and $\mathbf{V} \in \mathbb{R}^{r\times n}$ for some adjustable rank estimate r. Correspondingly, (3.14) can be rewritten as

$$\min_{\mathbf{U},\mathbf{V},\mathbf{Z_0}} \|\mathcal{P}_{\mathbf{\Omega}}(\mathbf{Z_a} - \mathbf{Z_0})\|_1, \quad s.t. \quad \mathbf{UV} - \mathbf{Z_0} = \mathbf{0}. \tag{3.24}$$

Note that a low-rank matrix factorization is explicitly applied to $\mathbf{Z_0}$ instead of minimizing its nuclear norm as in (3.14), which avoids the singular value decomposition completely. To solve the minimization problem in (3.24), the augmented Lagrangian can be expressed as

$$L(\mathbf{U}, \mathbf{V}, \mathbf{Z_0}, \mathbf{Y}, \mu) = \|\mathcal{P}_{\mathbf{\Omega}}(\mathbf{Z_a} - \mathbf{Z_0})\|_1 + \langle \mathbf{Y}, \mathbf{UV} - \mathbf{Z_0} \rangle + \frac{\mu}{2}\|\mathbf{UV} - \mathbf{Z_0}\|_2^2, \tag{3.25}$$

where μ is a penalty parameter and $\mathbf{Y}$ is the vector of Lagrange multipliers corresponding to the constraint $\mathbf{UV} - \mathbf{Z_0} = \mathbf{0}$. Motivated by the idea in the alternating direction method for convex optimization, the augmented Lagrangian can be minimized with respect to block variables $\mathbf{U}, \mathbf{V}$, and $\mathbf{Z_0}$ individually via the following framework at each iteration k [66]

$$\mathbf{U}^{k+1} = \arg\min_{\mathbf{U}} L(\mathbf{U}, \mathbf{V}^k, \mathbf{Z_0}^k, \mathbf{Y}^k, \mu^k), \tag{3.26}$$

$$\mathbf{V}^{k+1} = \arg\min_{\mathbf{V}} L(\mathbf{U}^{k+1}, \mathbf{V}, \mathbf{Z_0}^k, \mathbf{Y}^k, \mu^k), \text{ and} \tag{3.27}$$

$$\mathbf{Z_0}^{k+1} = \arg\min_{\mathbf{Z_0}} L(\mathbf{U}^{k+1}, \mathbf{V}^{k+1}, \mathbf{Z_0}, \mathbf{Y}^k, \mu^k), \tag{3.28}$$

where (3.26) and (3.27) are least squares problems

$$\mathbf{U}^{k+1} = (\mathbf{Z_0} - \frac{\mathbf{Y}^k}{u^k})\mathbf{V}^\top(\mathbf{V}\mathbf{V}^\top)^{-\mathbf{1}} \text{ and} \tag{3.29}$$

$$\mathbf{V}^{k+1} = (\mathbf{U}^\top\mathbf{U})^{-\mathbf{1}}\mathbf{U}^\top(\mathbf{Z_0} - \frac{\mathbf{Y}^k}{u^k}). \tag{3.30}$$

and (3.28) can be solved by the shrinkage formula

$$\mathcal{P}_\Omega(\mathbf{Z_0}^{k+1}) = \mathcal{P}_\Omega(\mathcal{S}_{\frac{1}{u^k}}\{\mathbf{U}^{k+1}\mathbf{V}^{k+1} - \mathbf{Z_a} + \frac{\mathbf{Y}^k}{u^k}\}). \tag{3.31}$$

The Lagrangian multipliers $\mathbf{Y}$ and μ are updated during each iteration as follows

$$\mathbf{Y}^{k+1} = \mathbf{Y}^k + u^k(\mathbf{U}^{k+1}\mathbf{V}^{k+1} - \mathbf{Z_0}^{k+1}) \text{ and} \tag{3.32}$$

$$\mu^{k+1} = \alpha\mu^k, \tag{3.33}$$

where α is a positive constant. At the end of each iteration, a rank estimation strategy [67] is applied to update r to ensure the success of the algorithm. The proposed algorithm is illustrated in Algorithm 3.2.

Algorithm 3.2 Low rank matrix factorization.

Input: $\mathbf{Z_a} \in \mathbb{R}^{m\times t}$; Initial rank estimate r.
Initialize: $\mathbf{U} \in \mathbb{R}^{m\times r}$; $\mathbf{V} \in \mathbb{R}^{r\times t}$; $\mathbf{Z_0}_{[0]} = U * V$; $\mathbf{Y}_{[0]} = 0$; $\mu_{[0]} > 0$; $\alpha > 0$; $k = 0$.
while not converge **do**
  $\mathbf{U}^{k+1} = (\mathbf{Z_0} - \frac{\mathbf{Y}^k}{u^k})\mathbf{V}^\top(\mathbf{V}\mathbf{V}^\top)^{-\mathbf{1}}$;
  $\mathbf{V}^{k+1} = (\mathbf{U}^\top\mathbf{U})^{-\mathbf{1}}\mathbf{U}^\top(\mathbf{Z_0} - \frac{\mathbf{Y}^k}{u^k})$;
  ${\mathbf{Z_0}}^{k+1} = \mathcal{S}_{\frac{1}{u^k}}\{\mathbf{U}^{k+1}\mathbf{V}^{k+1} - \mathbf{Z_a} + \frac{\mathbf{Y}^k}{u^k}\}$;
  $\mathbf{Y}^{k+1} = \mathbf{Y}^k + u^k(\mathbf{U}^{k+1}\mathbf{V}^{k+1} - {\mathbf{Z_0}}^{k+1})$;
  $\mu^{k+1} = \alpha\mu^k$;

  $k = k + 1$;

  Check r, possibly re-estimate r and adjust sizes of the iterates;
end while
return ${\mathbf{Z_0}}^k$;
Output ${\mathbf{Z_0}}^k$; $\mathbf{Z_a} - {\mathbf{Z_0}}^k$;

3.1.4 Numerical Results

Numerical simulations are presented here to evaluate the performance of proposed algorithms. Power flow data for IEEE 57 bus, IEEE 118 bus test cases, and Polish system [68] during winter the peak conditions in 2007-2008 are used to evaluate the proposed algorithms.

3.1.4.1 Receiver Operating Characteristic Analysis

Assume loads on each bus in the power system are uniformly distributed between 50% and 150% of its base load. When state estimation measurements are collected, a small portion ϵ of measurement data are compromised by malicious attackers with an arbitrary amount of injection data, and ϵ is defined as the attack ratio in this context. Methods for false data injection attack construction can be found in [56, 57]. Here, we focus on the protection scheme and suppose that the locations of attacks are chosen randomly and are of duration Δt[1]. Totally a number of T time instance measurements are obtained for analysis. The receiver operating characteristic analysis of proposed algorithms is first given, and then we compare the performance of proposed algorithms with that of the principal component analysis (PCA)[2]. In this analysis, the attack ratio is fixed at

[1]Note that the attack vectors used in this chapter are more general compared to those described in [56, 57] and will not affect the efficiency of proposed algorithms.

[2]For PCA, we retain the largest K singular values of the matrix such that $\frac{\sum_1^K s_i}{\sum_1^N s_i} > 95\%$.

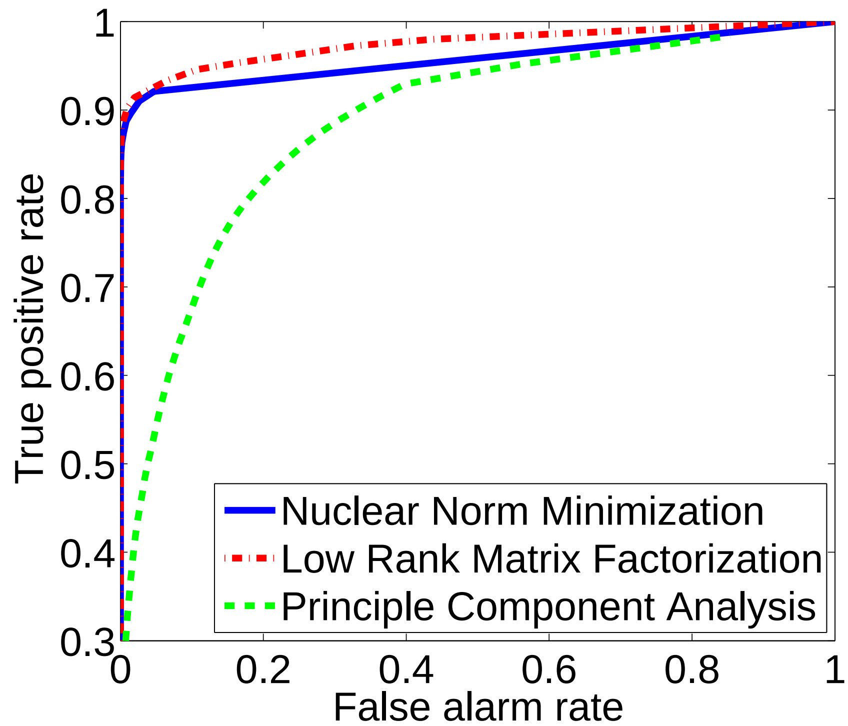


Figure 3.1 The ROC performance for the IEEE 57 bus system. SNR = 10dB.

$\epsilon = 0.1$ and SNR = 10dB.

The ROC curves for IEEE 57 bus and IEEE 118 bus cases are shown in Fig. 3.1 and Fig. 3.2, respectively. From those figures, it is apparent that proposed algorithms can detect the false data accurately at a low false alarm rate. For example, in the IEEE 57 bus system, the true positive rate of nuclear norm minimization is 93%; and it is 95% with low rank matrix factorization when the false alarm rate $p_f = 10\%$. Moreover, the low rank matrix factorization approach performs slightly better than the nuclear norm minimization method. In this case, the sparse attack matrix is not the dominant part in measurements, which makes the low rank matrix factorization approach more suitable. Fig. 3.1 and Fig. 3.2 show that proposed algorithms outperform the PCA-based approach significantly. The PCA method neglects the corruptions of malicious attacks. Even though the matrix $\mathbf{Z_0}$ is of low rank, the sum of $\mathbf{Z_0}$ and $\mathbf{A}$ will not be of low rank any more. Thus, directly applying the PCA method will result in a poor performance. However, proposed algorithms exploit the low rank structure of the anomaly-free measurement matrix, and the fact that malicious attacks

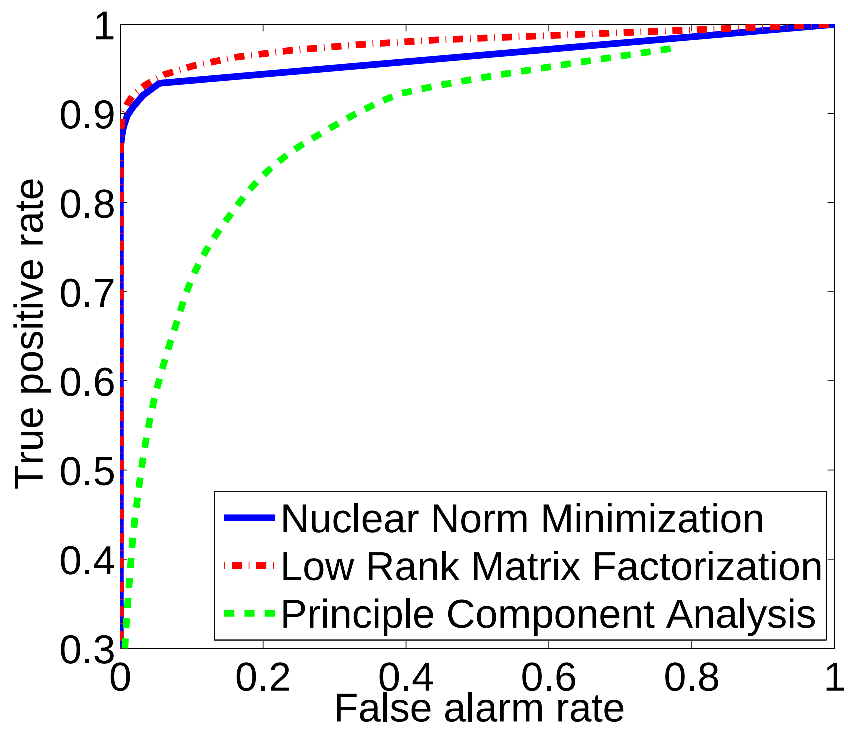


Figure 3.2 The ROC performance for the IEEE 118 bus system. SNR = 10dB.

are quite sparse, which render better performance.

3.1.4.2 Performance vs. Measurement Missing Ratio

Next, we investigate the performance of proposed algorithms under different measurement missing ratios. In particular, we assume that a portion of measurements collected at the control center are missing due to meter failures or communication link outages, and evaluate the performance of proposed algorithms under different measurement missing ratios up to 10% on the IEEE 118 bus system. The attack ratio is fixed at $\epsilon = 0.1$ with SNR = 10dB.

The ROC curves for the IEEE 118 bus case are depicted in Fig. 3.3. From the figure we see that with 10% missing measurements, proposed algorithms are still able to detect the malicious attacks at acceptable true positive rates, and the low rank matrix factorization method performs slightly better. By comparing with Fig. 3.2, we see that the missing measurements deteriorate the performance of proposed algorithms as we would expect. Since the PCA-based method is unable to

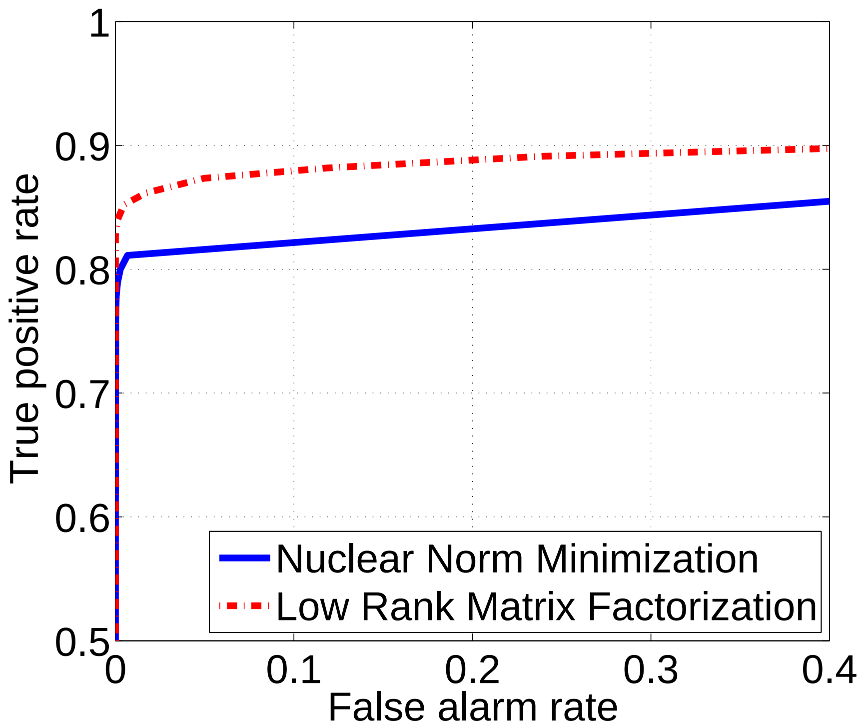


Figure 3.3 The ROC curves of the proposed algorithms for the IEEE 118 bus system. 10% measurements are missing and SNR = 10dB.

detect anomalies in this case, we omitted its simulation results. Note that the existence of missing entries will result in an incorrect estimation of the low-dimensional subspace of matrix $\mathbf{Z_0}$, which leads to the failure of PCA.

To investigate the performance under different measurement missing ratios, the percentage of missing measurements is varied from 0% (no missing) to 10%, and results are shown in Fig. 3.4. The true positive rates are calculated for both algorithms when the false alarm rate equals 10%. It is shown that the performance is improved monotonically as more and more measurements are collected. In the worst case when 10% of measurements are missing, proposed algorithms can still achieve true positive rates of 85% and 90% for the nuclear norm minimization and the low rank matrix factorization methods, respectively.

A more detailed demonstration for recoverability of proposed algorithms for power system states is shown in Fig. 3.5. Here, we assume 10% of the measurements are missing with SNR = 10dB, and cumulative distribution functions of relative reconstruction errors at $t = 50$ and $t = 100$

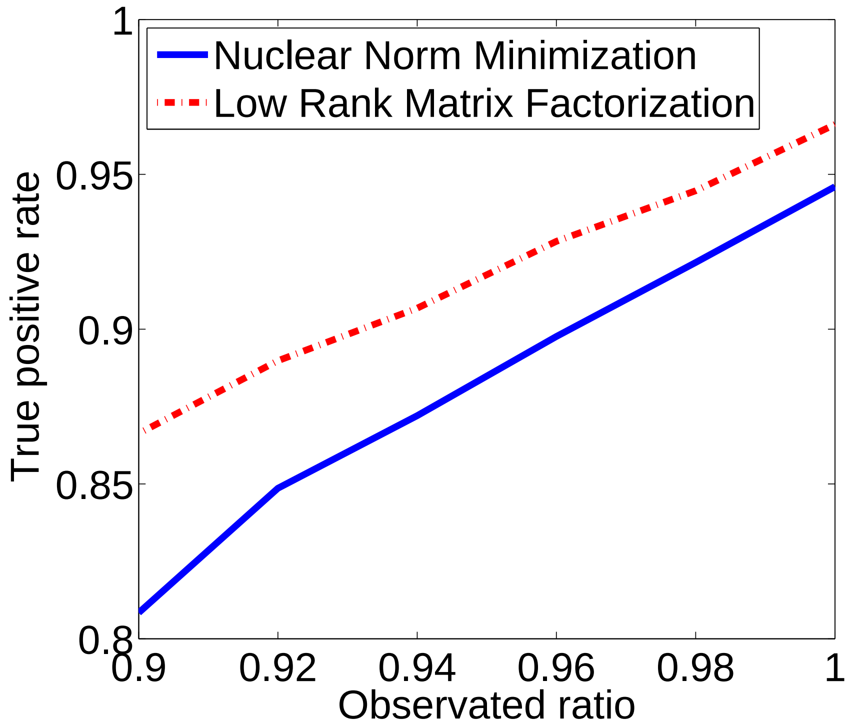


Figure 3.4 Performance of the proposed algorithms under different missing ratios for the IEEE 118 bus system. The false alarm rate is 10% and SNR = 10dB.

are calculated. The relative reconstruction error is defined as

$$\varepsilon = (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})./|\boldsymbol{\theta}|, \tag{3.34}$$

where $./$ denotes componentwise division, and $|\boldsymbol{\theta}|$ denotes the element-wise absolute value of the vector $\boldsymbol{\theta}$. The vector $\boldsymbol{\theta}$ (in radian units) is obtained from the recovered $\mathbf{Z_0}$, and the vector ε represents the relative error of each component in the state vector $\boldsymbol{\theta}$. We calculate the relative error for each bus in the system, and plot the corresponding cumulative distribution functions. From Fig. 3.5 we see that proposed algorithms are able to reconstruct power system states quite accurately. At $t = 50$, the majority of relative errors concentrate between interval $[-0.1, 0.1]$, and similar results are shown at $t = 100$. These imply that proposed algorithms are able to precisely detect the malicious attacks as well as accurately estimate power system states, even under some severe situations of missing partial measurements.

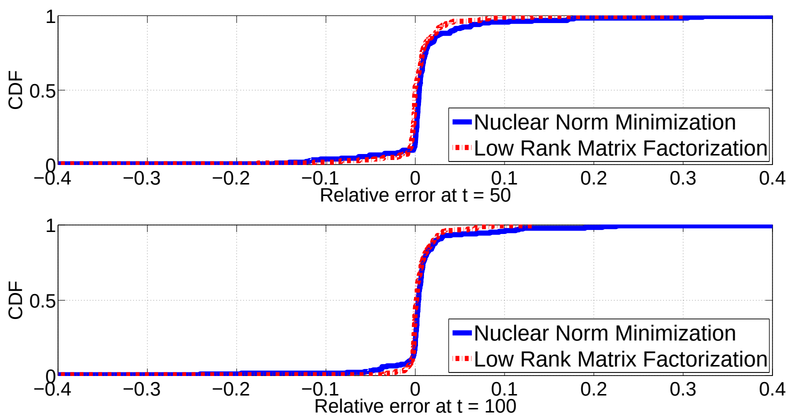


Figure 3.5 Power state reconstruction performance of the proposed algorithms at specific time instance $t = 50$ and $t = 100$. 10% of the measurements are missing and SNR = 10dB.

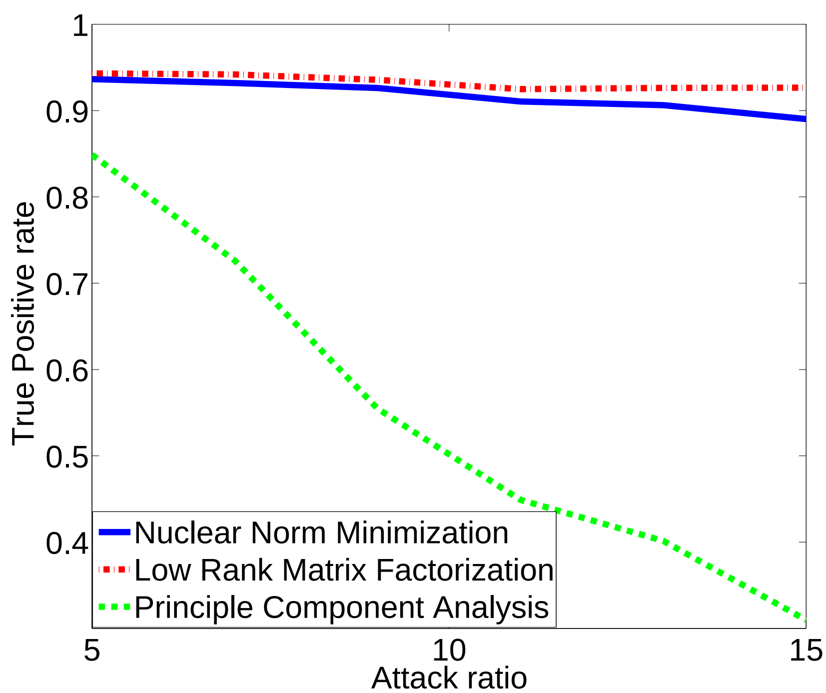


Figure 3.6 Performance of the proposed algorithms under different attack ratios for the IEEE 118 bus system. The false alarm rate is 10% and SNR = 10dB.

3.1.4.3 Performance vs. Attack Ratio

Thirdly, we investigate the performance of proposed algorithms under different attack ratios for the IEEE 118 bus system. In particular, ϵ is varied from 5% to 15%, and SNR = 10dB.

From Fig. 3.6, the true positive rate is quite high at low sparsity ratios for both proposed algorithms. Particularly, when the sparsity ratio is 5%, true positive rates are 93.6% and 94.3% at $f_a = 10\%$ for the nuclear norm minimization method and the low rank matrix factorization method, respectively. Compared with the PCA-based method, the performance of proposed algorithms is quite stable as the attack ratio increases. When the attack ratio reaches 15%, true positive rates for both algorithms are still around 90%. The true positive rates of proposed algorithms will decrease dramatically when attackers attack the power system massively. This is because, when the attack matrix is not sparse enough, the mixed-norm minimization is not able to separate the low rank anomaly-free matrix from the attack matrix.

3.1.4.4 Performance on Large-Scale Systems

Finally, we analyze the scalability and computational efficiency of proposed algorithms on power flow data for the Polish system during winter peak conditions in 2007-2008. The attack ratio is fixed at $\epsilon = 0.1$ with SNR = 10dB.

The ROC curve is shown in Fig. 3.7. It is shown that the performance of proposed algorithms is quite stable on the large scale system compared to the IEEE 57 bus and the IEEE 118 bus. A comparison of the computational efficiency of two proposed algorithms is shown in Fig. 3.8. The data matrix row dimension m is varied from 100 to 3400. The proposed algorithms are applied to a subset of the measurement matrix each time, and the CPU computation time is logged. It is shown in Fig. 3.8 that as the dimension of the measurement matrix increases, the CPU time for computation will increase, and the low rank matrix factorization approach performs better than the nuclear norm minimization method, which demonstrates a better scalability to large problems, as expected.

The numerical results validate the effectiveness of proposed algorithms. According to simulation results, both low rank matrix factorization and nuclear norm minimization techniques can

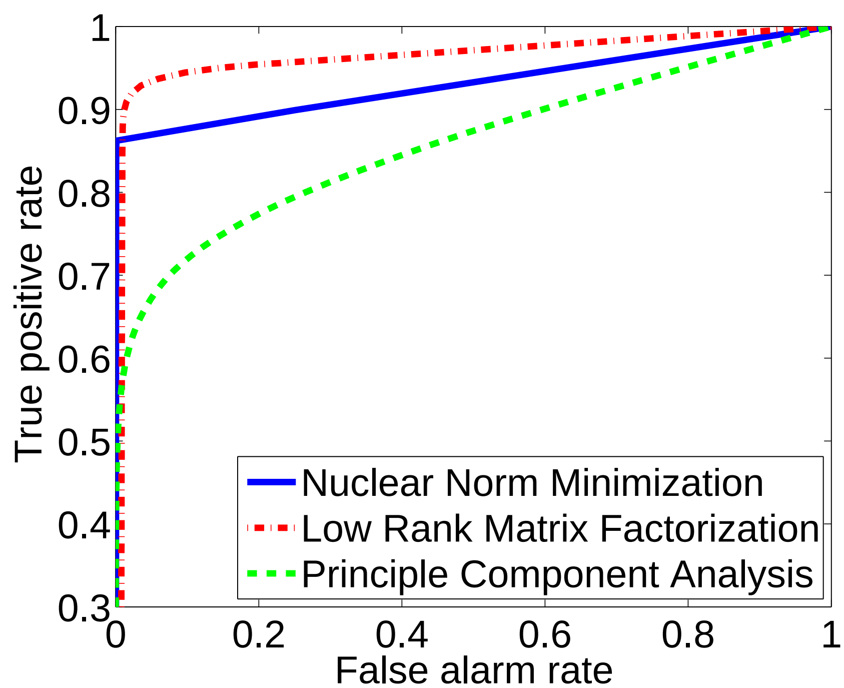


Figure 3.7 Performance on power flow data for the Polish system during winter peak conditions, 2007-2008. SNR = 10dB.

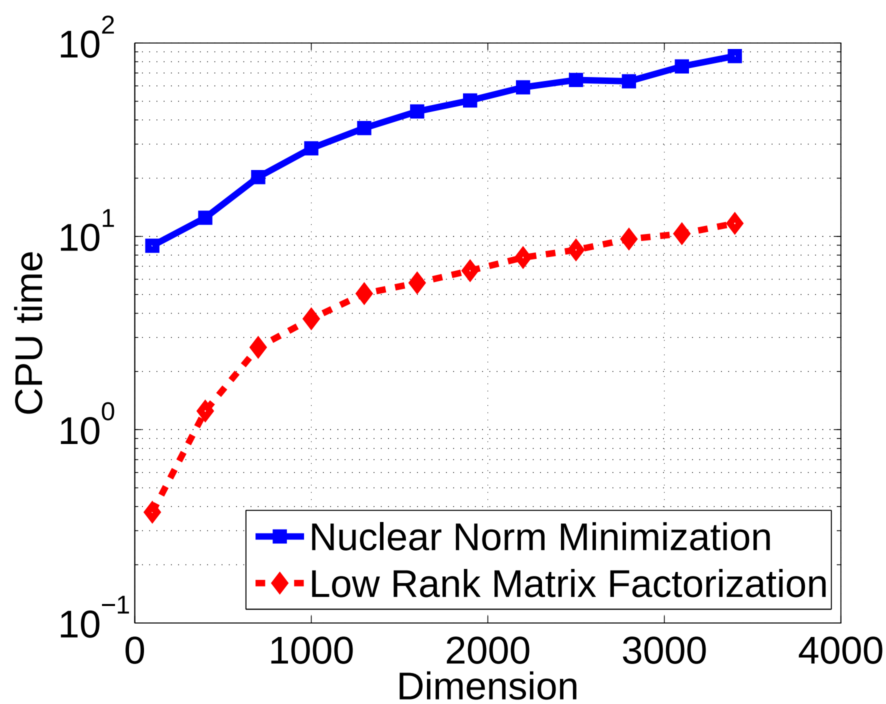


Figure 3.8 The CPU computing time versus matrix dimension for the proposed nuclear norm minimization and low rank matrix factorization algorithms.

solve the matrix separation problem, and the performance of the low rank matrix factorization is slightly better than that of the nuclear norm minimization technique. From the perspective of recoverability, since the false data attack matrix $\mathbf{A}$ is not the dominant part compared with $\mathbf{Z_0}$ in this setting, the performance of the low rank matrix factorization technique is better. From the perspective of computation time, the low rank matrix factorization technique is much faster than the nuclear-norm minimization technique due to its SVD-free feature. A detailed comparison of the complexity of two algorithms is beyond the scope of this chapter, and useful discussions can be found in reference [66].

3.2 Distributed Parallel Approach for Security Constrained Optimal Power Flow

In this section, we first introduce the background of the security constrained optimal power flow (SCOPF) problem. Then we propose a distributed parallel approach to address it. Finally, numerical simulations are given to validate the effectiveness of the proposed algorithm.

3.2.1 Security Constrained Optimal Power Flow

The deregulation of electric power grids offers the opportunity for electricity market participants to exercise least-cost or profit-based operations [69]. Despite the market-driven tendency of the electric power business, security remains a significant concern of sustainable power system operations, which cannot be compromised. Security-constrained optimal power flow [70, 71] aims at minimizing the cost of system operation while satisfying a set of postulated contingency constraints. It is an important management task allowing optimal control of power systems securely.

The SCOPF is an extension of the conventional optimal power flow (OPF) problem [72], whose objective is to determine a generation schedule that minimizes the system operating cost while satisfying the system operation constraints such as hourly load demand, fuel limitations, environmental constraints and network security requirements. It has been recognized [73] that the optimal control of the normal state may violate system operation constraints after the occurrence of some disturbance events, and thus jeopardize the security of power systems. To address this problem, SCOPF is performed by considering both pre-contingency and post-contingency constraints to guarantee sustainable operations of the electric grid. The system security level is improved by taking into account a number of contingencies in a selected contingency list. The solution to SCOPF should satisfy the so called $N-1$ criterion, which requires that operational limits of the power system should not be violated in case of a single contingency (a line and/or generator outage).

The SCOPF can be broadly classified as preventive, where control variables are restricted to their pre-contingency condition settings, and corrective, whose control variables are allowed to be rescheduled [74]. We will focus on the corrective model in this example. The seminal paper [73] proposed the generalized Benders decomposition method to solve the corrective SCOPF problem. Since then, an extensive literature for SCOPF in power systems exists both for traditional operations and under market environments [71,75–78]. The nested Benders decomposition method was utilized in [75] to solve the SCOPF problem for determining the optimal daily generation scheduling in a pool-organized electricity market, and was tested in an actual example of the Spanish power system. [76] embedded SCOPF into the security-constrained unit commitment (SCUC) model, and designed an effective corrective contingency dispatch over a 24-hour period, which balanced the economics and security in the restructured markets. An iterative approach was proposed in [77] to obtain the solution of SCOPF, which aims to efficiently identify a superset of binding contingencies to achieve the SCOPF optimum. [78] applied the Benders decomposition to decompose the traditional SCOPF problem, and the underlying computational complexity was analyzed in this approach. [71] solved the SCOPF problem by a non-decomposed method based on the compression of post-contingency networks, which can reduce the size of security constraints and relieve the computational burden in the problem.

Before presenting the distributed parallel approach for this problem, it is useful to recall a general formulation of the conventional SCOPF problem compactly described as follows:

$$\underset{\mathbf{x}^0,\ldots,\mathbf{x}^C;\mathbf{u}^0,\ldots,\mathbf{u}^C}{\text{minimize}} \quad f^0(\mathbf{x}^0, \mathbf{u}^0) \tag{3.35}$$

$$\text{subject to} \quad \mathbf{g}^0(\mathbf{x}^0, \mathbf{u}^0) = 0, \tag{3.36}$$

$$\mathbf{h}^0(\mathbf{x}^0, \mathbf{u}^0) \leq 0, \tag{3.37}$$

$$\mathbf{g}^c(\mathbf{x}^c, \mathbf{u}^c) = 0, \tag{3.38}$$

$$\mathbf{h}^c(\mathbf{x}^c, \mathbf{u}^c) \leq 0, \text{ and} \tag{3.39}$$

$$|\mathbf{u}^0 - \mathbf{u}^c| \leq \mathbf{\Delta}_c, \quad c = 1, \ldots, C, \tag{3.40}$$

where f^0 is the objective function, which (3.35) aims to maximize the total social welfare or equivalently minimize offer-based energy and production cost, $\mathbf{x}^c$ is the vector of state variables, which includes voltage magnitudes and angles at all buses, and $\mathbf{u}^c$ is the vector of control variables, which can be generator real powers or terminal voltages. The superscript $c = 0$ corresponds to the pre-contingency configuration, and $c = 1, \ldots, C$ correspond to different post-contingency configurations. In addition, $\mathbf{\Delta}_c$ is the maximum allowed adjustment between the normal and contingency states for contingency c.

In the conventional SCOPF problem, the equality constraints 3.38 on $\mathbf{g}^c, c = 0, \ldots, C$, represent the system nodal power flow balance over the entire grid, and the inequality constraints 3.39 on $\mathbf{h}^c, c = 0, \ldots, C$, represent the physical limits on the equipment, such as the operational limits on the branch currents and bounds on the generator power outputs. Constraints (3.36)-(3.37) capture the economic dispatch and enforce the feasibility of the pre-contingency state. Constraints (3.38)-(3.39) incorporate the security-constrained dispatch and enforce the feasibility of the post-contingency state. Constraint (3.40) introduces the security-constrained dispatch with rescheduling, which couples control variables of pre-contingency and post-contingency states and prevents unrealistic post-contingency corrective actions. Note that there are some variations on the objective function and constraints of the SCOPF problem, and we focus on the above conventional formulation in this chapter.

Following the standard approach to formulating the SCOPF problem, the objective here is to minimize the cost of generation while safeguarding the power system sustainability. For the sake of simplicity and computational tractability, constraints (3.36)-(3.39) are modeled with the linear DC load flow, and we assume that the list of contingencies is given. Thus, assuming a DC power network modeling and neglecting all shunt elements, the standard SCOPF problem can be simplified to the following optimization problem

$$\underset{\boldsymbol{\theta}^0,\ldots,\boldsymbol{\theta}^C;\mathbf{P}^{g,0},\ldots,\mathbf{P}^{g,C}}{\text{minimize}} \quad \sum_{i\in\mathcal{G}} f_i^g(\mathbf{P}_i^{g,0}) \tag{3.41}$$

$$\text{subject to} \quad \mathbf{B}_{bus}^0\boldsymbol{\theta}^0 + \mathbf{P}^{d,0} - \mathbf{A}^{g,0}\mathbf{P}^{g,0} = 0, \tag{3.42}$$

$$\mathbf{B}_{bus}^c\boldsymbol{\theta}^c + \mathbf{P}^{d,c} - \mathbf{A}^{g,c}\mathbf{P}^{g,c} = 0, \tag{3.43}$$

$$|\mathbf{B}_f^0\boldsymbol{\theta}^0| - \mathbf{F}_{max} \leq 0, \tag{3.44}$$

$$|\mathbf{B}_f^c\boldsymbol{\theta}^c| - \mathbf{F}_{max} \leq 0, \tag{3.45}$$

$$\underline{\mathbf{P}^{g,0}} \leq \mathbf{P}^{g,0} \leq \overline{\mathbf{P}^{g,0}}, \tag{3.46}$$

$$\underline{\mathbf{P}^{g,c}} \leq \mathbf{P}^{g,c} \leq \overline{\mathbf{P}^{g,c}}, \tag{3.47}$$

$$|\mathbf{P}^{g,0} - \mathbf{P}^{g,c}| \leq \boldsymbol{\Delta}_c, \text{ and} \tag{3.48}$$

$$i \in \mathcal{G}, \quad c = 1, \ldots, C, \tag{3.49}$$

where the notation is given in Table 3.1.

The solution to (3.41) ensures economical dispatch while guaranteeing power system security, by taking into account a set of postulated contingencies. The major challenge of SCOPF is the problem size, especially for large systems with numerous contingency cases to be considered. Directly solving the SCOPF problem by simultaneously imposing all post-contingency constraints will result in prohibitive memory requirements and a substantial CPU burden. To achieve efficient and secure operations of the entire electrical grid, a distributed approach is proposed in next sections.

3.2.2 Distributed and Parallel Approach for SCOPF

The proposed distributed optimization method is based on the ADMM. The use of ADMM for optimization in power systems has been considered in [79] and [80]. However, the optimiza-

Table 3.1 Notation definitions.

$\mathcal{G}$	Set of generators
$\mathcal{N}$	Set of buses
$\mathcal{B}$	Set of branches
$\boldsymbol{\theta}^c \in \mathbb{R}^{\lvert\mathcal{N}\rvert}$	Vector of voltage angles
$\mathbf{P}^{g,c} \in \mathbb{R}^{\lvert\mathcal{G}\rvert}$	Vector of real power flows
f_i^g	Generation cost function
$\mathbf{P}_i^{g,0}$	Displaceable real power of each individual generation unit i for the pre-contingency configuration
$\mathbf{B}_{bus}^c \in \mathbb{R}^{\lvert\mathcal{N}\rvert \times \lvert\mathcal{N}\rvert}$	Power network system admittance matrix
$\mathbf{B}_f^c \in \mathbb{R}^{\lvert\mathcal{B}\rvert \times \lvert\mathcal{N}\rvert}$	Branch admittance matrix
$\mathbf{P}^{d,c} \in \mathbb{R}^{\lvert\mathcal{N}\rvert}$	Real power demand
$\mathbf{A}^{g,c} \in \mathbb{R}^{\lvert\mathcal{N}\rvert \times \lvert\mathcal{G}\rvert}$	Sparse generator connection matrix, whose (i,j)-th element is 1 if generator j is located at bus i and 0 otherwise
$\mathbf{F}_{max}$	Vector for the maximum power flow
$\overline{\mathbf{P}^{g,c}}$	Upper bound on real power generation
$\underline{\mathbf{P}^{g,c}}$	Lower bound on real power generation
$\boldsymbol{\Delta}_c$	Pre-defined maximum allowed variation of power outputs

tion problem (3.41) cannot be readily solved using ADMM, since the constraint (3.48) couples the pre-contingency and post-contingency variables, and the inequalities make the problem even more complicated. To address these challenges, the optimization problem (3.41) can then be reformulated by introducing a slack variable $\mathbf{p}^c \in \mathbb{R}^{\lvert\mathcal{G}\rvert}$

$$\text{minimize} \quad (3.41) \tag{3.50}$$

$$\text{subject to} \quad \text{Constraints (3.42)-(3.47)}, \tag{3.51}$$

$$\mathbf{P}^{g,0} - \mathbf{P}^{g,c} + \mathbf{p}^c = \boldsymbol{\Delta}_c, \text{ and} \tag{3.52}$$

$$0 \leq \mathbf{p}^c \leq 2\boldsymbol{\Delta}_c, \quad c = 1, \ldots, C. \tag{3.53}$$

The above optimization problem can be solved distributively using ADMM. The scaled augmented Lagrangian can be calculated as

$$\mathcal{L}_\rho(\{\mathbf{P}^{g,c}\}_{c=1}^C; \{\mathbf{p}^c\}_{c=1}^C; \{\boldsymbol{\mu}^c\}_{c=1}^C) = \sum_{i\in\mathcal{G}} f_i^g(\mathbf{P}_i^{g,0}) + \sum_{c=1}^C \frac{\rho^c}{2}\|\mathbf{P}^{g,0} - \mathbf{P}^{g,c} + \mathbf{p}^c - \boldsymbol{\Delta}_c + \boldsymbol{\mu}^c\|_2^2. \quad (3.54)$$

The optimization variables $\mathbf{P}^{g,0}, \mathbf{P}^{g,c}$, and $\mathbf{p}^c$ are arranged into two groups, $\{\mathbf{P}^{g,0}\}$ and $\{\mathbf{P}^{g,c}, \mathbf{p}^c\}$, and updated iteratively. The variables in each group are optimized in parallel on distributed computing nodes, and coordinated by the dual variable vector $\boldsymbol{\mu}^c$ during each iteration.

At the k^{th} iteration, the $\mathbf{P}^{g,0}$-update solves the base scenario with squared regularization terms enforced by the coupling constraints and expressed as

$$\mathbf{P}^{g,0}[k+1] = \arg\min_{\mathbf{P}^{g,0}} \sum_{i\in\mathcal{G}} f_i^g(\mathbf{P}_i^{g,0}) + \sum_{c=1}^C \frac{\rho^c}{2}\|\mathbf{P}^{g,0} - \mathbf{P}^{g,c}[k] + \mathbf{p}^c[k] - \boldsymbol{\Delta}_c + \boldsymbol{\mu}^c[k]\|_2^2,$$

$$\text{subject to} \quad \text{Constraints}(3.42), (3.44), \text{and } (3.46). \quad (3.55)$$

The $\mathbf{P}^{g,c}$-updating solves a number of independent optimization subproblems correspond to post-contingency scenarios and can be calculated distributively at the c^{th} computing nodes via

$$\mathbf{P}^{g,c}[k+1] = \arg\min_{\mathbf{P}^{g,c}, \mathbf{p}^c} \frac{\rho^c}{2}\|\mathbf{P}^{g,0}[k+1] - \mathbf{P}^{g,c} + \mathbf{p}^c - \boldsymbol{\Delta}_c + \boldsymbol{\mu}^c[k]\|_2^2,$$

$$\text{subject to} \quad \text{Constraints}(3.43), (3.45), (3.47), \text{and } (3.53), \quad (3.56)$$

where the scaled dual variable vector is also updated locally at the c^{th} computing utility as

$$\boldsymbol{\mu}^c[k+1] = \boldsymbol{\mu}^c[k] + \mathbf{P}^{g,0}[k+1] - \mathbf{P}^{g,c}[k+1] + \mathbf{p}^c[k+1] - \boldsymbol{\Delta}_c. \quad (3.57)$$

At the k^{th} iteration, the original problem is divided into $C + 1$ subproblems of approximately the same size. The computing node handling $\mathbf{P}^{g,0}$ needs to communicate with all computing nodes solving (3.56) during the iterations. The results of the $\mathbf{P}^{g,0}$-update, $\{\mathbf{P}^{g,0}\}$, will be distributed among the computing nodes for the $\mathbf{P}^{g,c}$-update. After the $\mathbf{P}^{g,c}$-update, the computed $\{\mathbf{P}^{g,c}, \mathbf{p}^c, \boldsymbol{\mu}^c\}$ will be collected to calculate the pre-contingency control variables. The subproblem data are iteratively updated such the block-coupling constraints (3.52) are satisfied at the end. Note that since each of the subproblems is a smaller-scale OPF problem, existing techniques for OPF can be applied with minor modifications.The proposed algorithm is illustrated in Algorithm 3.3.

Algorithm 3.3 Distributed SCOPF.

Input: $\mathbf{B}_{bus}^c$, $\mathbf{B}_f^c$, $\mathbf{A}^{g,c}$, $\mathbf{P}^{d,c}$, $\overline{\mathbf{P}^{g,c}}$, $\underline{\mathbf{P}^{g,c}}$, $\mathbf{\Delta}_c$;
Initialize: $\boldsymbol{\theta}^c$, $\mathbf{P}^{g,c}$, $\mathbf{p}^c$, $\boldsymbol{\mu}^c$, ρ^c, $k = 0$;

while not converge **do**
$\mathbf{P}^{g,0}$-update:
$\mathbf{P}^{g,0}[k+1] = \arg\min_{\mathbf{P}^{g,0}} \sum_{i\in\mathcal{G}} f_i^g(\mathbf{P}_i^{g,0})$
$+\sum_{c=1}^{C} \frac{\rho^c}{2}\|\mathbf{P}^{g,0} - \mathbf{P}^{g,c}[k] + \mathbf{p}^c[k] - \mathbf{\Delta}_c + \boldsymbol{\mu}^c[k]\|_2^2$
subject to Constraints (3.42),(3.44), and (3.46).

$\mathbf{P}^{g,c}$-update, distributively at each computing node:
$\mathbf{P}^{g,c}[k+1] = \arg\min_{\mathbf{P}^{g,c},\mathbf{p}^c} \frac{\rho^c}{2}\|\mathbf{P}^{g,0}[k+1] - \mathbf{P}^{g,c} + \mathbf{p}^c - \mathbf{\Delta}_c + \boldsymbol{\mu}^c[k]\|_2^2$
subject to Constraints (3.43),(3.45),(3.47), and (3.53),
$\boldsymbol{\mu}^c[k+1] = \boldsymbol{\mu}^c[k] + \mathbf{P}^{g,0}[k+1] - \mathbf{P}^{g,c}[k+1] + \mathbf{p}^c[k+1] - \mathbf{\Delta}_c$.
Adjust penalty parameter ρ^c is necessary;
$k = k+1$;

end while

return $\boldsymbol{\theta}^c$, $\mathbf{P}^{g,c}$;

Output $\boldsymbol{\theta}^c$, $\mathbf{P}^{g,c}$;

The ADMM approach is a primal-dual algorithm in which each computing node c solves its own subproblem (3.56), and variations to constraint (3.52) are systematically penalized at certain prices through the scaled dual variable to each individual subproblem. Note that in ADMM frameworks for distributed computing, the dual variables, or prices, are not uniformly set up for all nodes, which will require costly synchronization. For convex optimization problems, the ADMM converges to the optimum geometrically [81], and the convergence rate can be improved by warm start techniques [82].

3.2.3 Numerical Results

In this section, numerical studies are examined to evaluate the performance of the proposed algorithm. Three classical test systems are used: the IEEE 57 bus, the IEEE 118 bus, and the IEEE 300 bus [68], whose structures and characteristics are summarized in Table 3.2.

Two kinds of contingencies are considered in the numerical tests: branch outages and generator failures. The contingencies are artificially generated and the number of contingencies considered

Table 3.2 Characteristics of test cases.

Case	$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{G}\rvert$	$\lvert\mathcal{B}\rvert$	Number of contingency cases
IEEE 57 bus	57	7	80	50
IEEE 118 bus	118	54	186	100
IEEE 300 bus	300	69	411	100

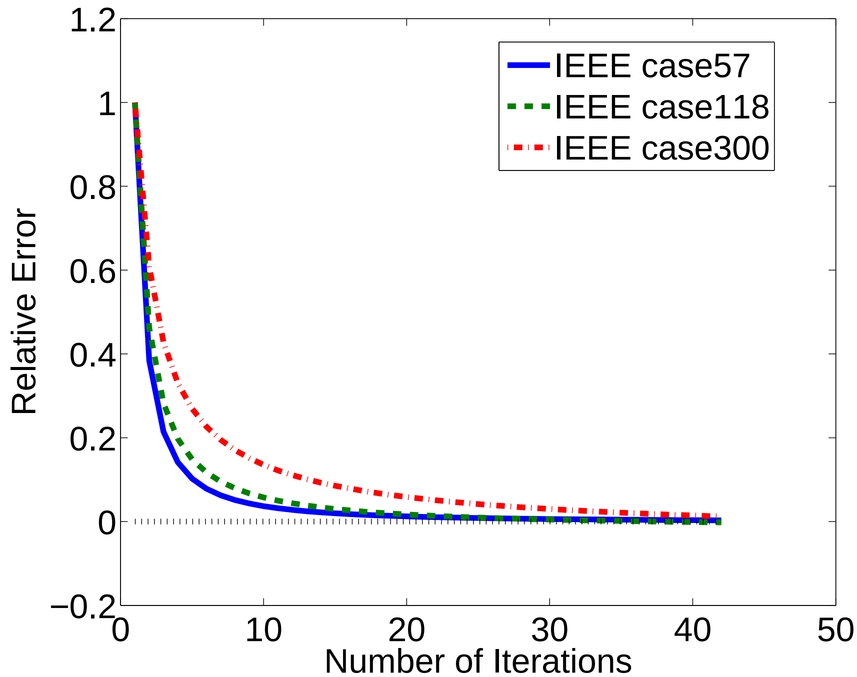


Figure 3.9 Convergence performance of the proposed distributed algorithm on test systems.

are listed in Table 3.2. We follow physical limits on equipments of test systems. The numerical tests are implemented via MATLAB 7.10 on a PC with an Intel Q8200 2.33GHz processor and 8GB memory. The basic OPF problem solver is the same for all test systems. The performance of convergence and computing time of the proposed algorithm are investigated in the following. The results are averaged over a total of 500 Monte Carlo implementations.

Table 3.3 Computing time performance of the proposed algorithm on different test systems.

Cases	Centralized		Distributed ADMM			
	Cost	Time	Cost	Time	Cost (e = 1%)	Time
IEEE 57 bus	487.53	5.22	487.53	3.55	492.40	1.18
IEEE 118 bus	1606.73	36.03	1606.73	18.92	1622.79	7.93
IEEE 300 bus	9567.12	221.67	9567.12	95.74	9662.80	52.87

3.2.3.1 Convergence Performance

We first consider the convergence issue of the proposed algorithm. Since the number of contingencies and the optimal value for each test system differs, the relative error is used here to present results. Suppose $r[k]$ is the resulting value of the objective function at the k^{th} iteration, and r^* is the optimal solution. The relative error e is defined as $e = \left|\frac{r[k]-r^*}{r[0]-r^*}\right|$. The convergence performance is shown in Fig. 3.9. It can be seen that after a moderate number of iterations, the proposed algorithm converges to optimal values in cases considered. From Fig. 3.9, we see that the IEEE 57 system gives the fastest convergence rate. A large system leads to a large scale optimization problem, and a large number of contingencies considered will make the problem scale even larger. Note that, after very few iterations, the algorithm gets very close to the optimal value, which means that the proposed algorithm is able to yield a good approximation to the optimal value in a short time.

3.2.3.2 Computing Time Performance

The computing time for test systems with different numbers of contingency cases is investigated and results are given in Fig. 3.10, Fig. 3.11, and Fig. 3.12. The number of contingencies is increased by 20% each time and the computing time is recorded. It can be seen from these figures that with an increase in the number of contingency cases for the SCOPF problem, the computing time of the centralized algorithm increases much faster than that of the proposed algorithm. Thus, the proposed distributed algorithm is more scalable and stable than the centralized approach.

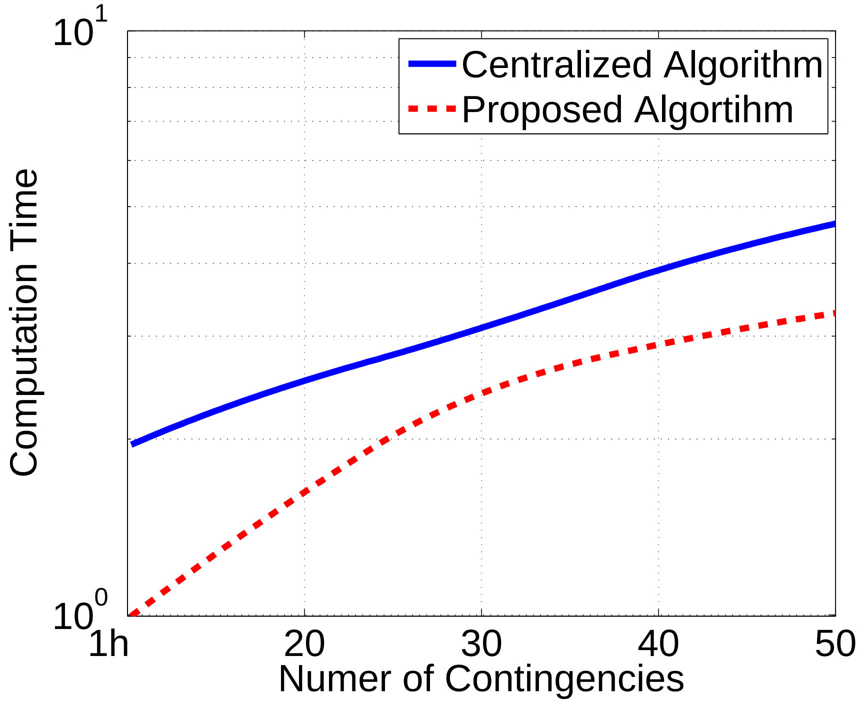


Figure 3.10 Computing time for the IEEE 57 bus system with different numbers of contingencies.

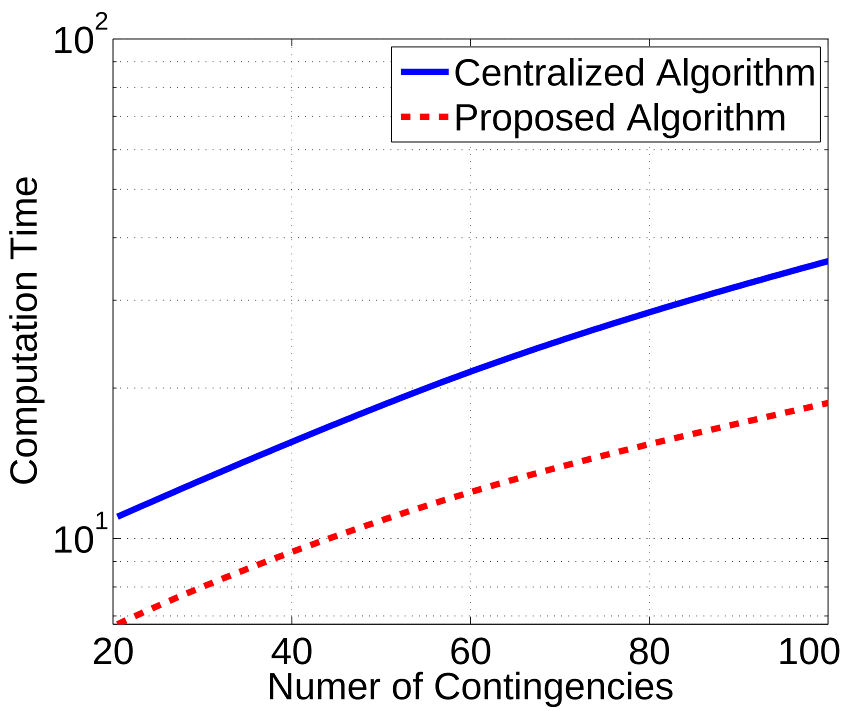


Figure 3.11 Computing time for the IEEE 118 bus system with different numbers of contingencies.

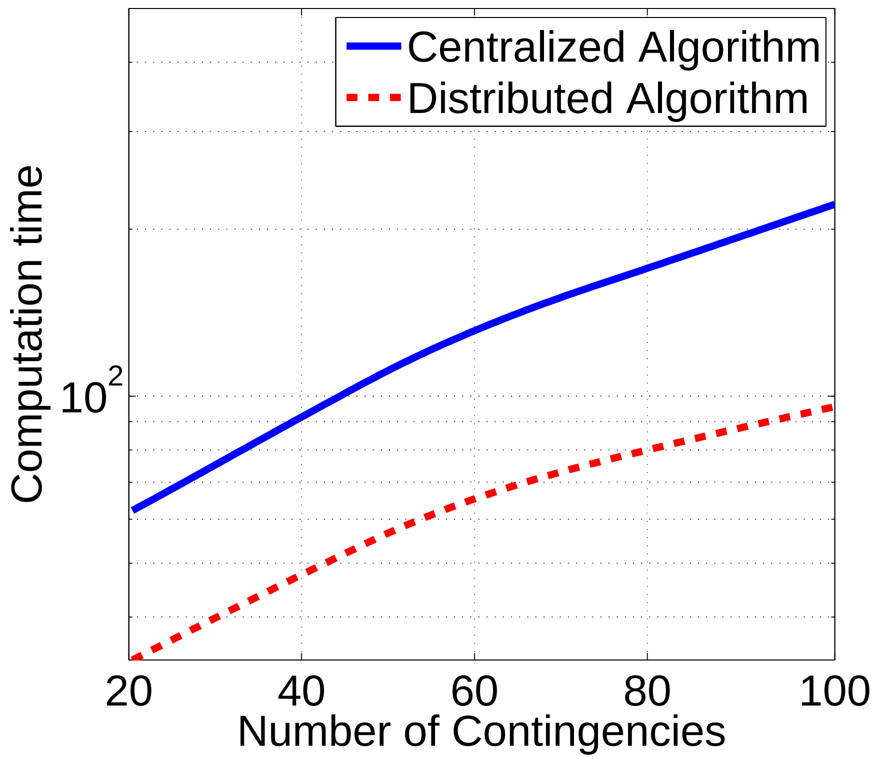


Figure 3.12 Computing time for the IEEE 300 bus system with different numbers of contingencies.

The computing time to achieve an approximate solution with a relative error of $e = 1\%$ is also considered for the distributed case. To better illustrate the numerical results, a speedup factor is defined as $S_p = T_c/T_p$, where T_c is the computing time of the centralized approach, and T_p is the computing time of the distributed approach. The results of the computing time performance are presented in Table 3.3. It is shown in Table 3.3 that the proposed distributed approach obtains the same optimum as the centralized approach, and can achieve a speedup factor S_p of $1.4 \sim 2.4$. Note that if only an approximate result is needed, the speedup factor can even be improved to S_p of $4.4 \sim 4.8$ by using the proposed distributed algorithm. The speedup factor for the smallest test system, IEEE 57 bus, is the smallest, due to the relatively more significant communication overhead between different computing nodes during the simulation. A larger S_p can be achieved on a large-scale test system since the communication overhead is negligible compared with the computing time of the optimization subproblem handled by each computing node.

3.3 Conclusion

In this chapter, we have investigated the applications of big data processing techniques for enhancing security in the smart grid. We have introduced two security concerns, the false data injection attacks against state estimation and the security constrained optimal power flow in power systems. We have explored possibilities of exploiting the inherent structure of data sets and effectively processing large data sets to enhance power system security. We have designed a sparse optimization approach for the false data injection detection problem, and a distributed parallel approach for the security constrained optimal power flow problem. We have performed numerical studies to validate the effectiveness of proposed approaches. We have shown that effective management and processing of big data has the potential to significantly improve smart grid security.

Chapter 4

Scaling into Clouds with Big Data

The mobile cloud computing has become a part of people's daily lives and is expected to play a significant role in the future cloud computing industry. Nowadays, people are used to access various mobile applications such as search engine, email, GPS navigation, streaming video and social networks from their mobile terminals through wireless access networks. Meanwhile, small and medium enterprises seize the opportunity to utilize the cloud computing paradigm as a flexible and economically efficient solution for service provisioning. It has great potential for mobile service providers to generate huge revenues without investing much capital for building and maintaining their own infrastructures. The rapid development of cloud infrastructures, mobile computing and wireless networks poses a complicated mobile cloud computing system, and numerous applications produce a huge amount of data traffic with diverse performance objectives.

The remaining of this chapter is organized as follows. Section 4.1 describes a distributed approach for mobile data offloading in a software defined network. The scalable service management in mobile cloud computing is developed in Section 4.2. Section 4.3 concludes this chapter.

4.1 Distributed Mobile Data Offloading in Software Defined Network

This section presents a distributed mechanism for mobile data offloading in a software defined network (SDN) at the network edge. We first give an introduction to mobile data offloading in the SDN-at-the-edge. Then, the proposed distributed mobile data offloading is described. Finally, we present numerical results of the proposed algorithm.

4.1.1 The Mobile Data Offloading in SDN

The mobile data offloading [83], which refers to offloading traffic from cellular networks to alternate wireless technologies like WiFi or small cell networks, is able to address tremendous growth in mobile data and rapidly evolving mobile services. The mobile data offloading can be

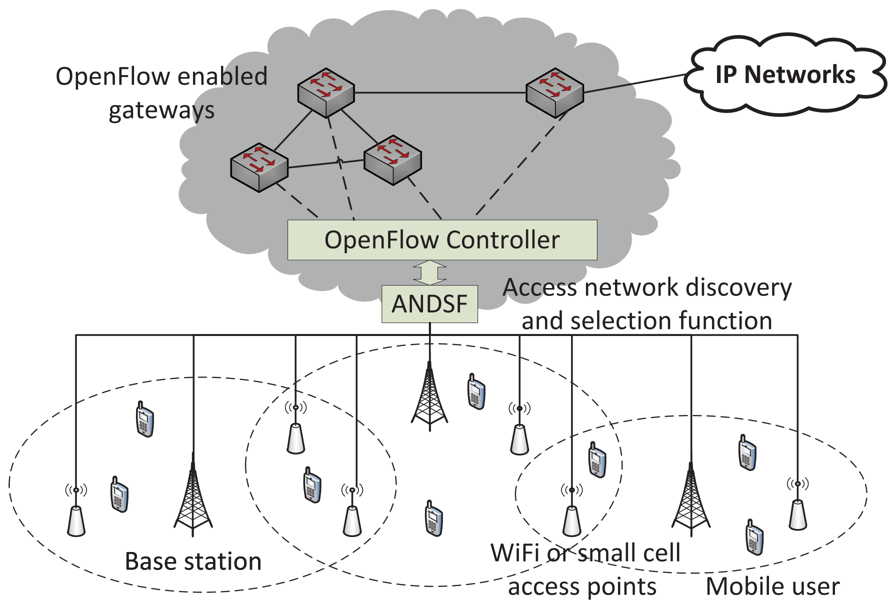


Figure 4.1 An illustration of the network model. The mobile data offloading can be enabled by the SDN at the network edge to dynamically route the data traffic in a mobile network.

enabled by a software defined network (SDN) [84] at the edge, which can dynamically route the traffic in a mobile network. An illustration of mobile data offloading via SDN is shown in Fig. 4.1. In this model, the access network discovery and selection function (ANDSF) can discover wireless networks close to mobile users and perform mobile data offloading. The ANDSF interacts with the virtual SDN centralized controller for offloading management, which can be implemented by standardized interfaces such as OpenFlow [85]. The mobile service operators have already deployed their own WiFi access points or initiated collaboration with existing WiFi networks to enable mobile data offloading, and the SDN-at-the-edge can significantly alleviate both cost and operational difficulties incurred by the simultaneous operation of access networks with multiple wireless technologies.

The benefits of mobile data offloading have been quantitatively studied in [83, 86, 87], which indicate that WiFi or small cell network can largely boost cellular network capacity, offload cellular data traffic, and save a huge amount of battery power for mobile users. [88] proposed an incentive framework to motivate mobile users to leverage their delay tolerance for cellular data traffic offloading. It could opportunistically offload cellular data traffic to WiFi networks or small cells, and relieved the cellular traffic overload. A dynamic resource allocation and parallel execution framework for mobile code offloading was presented in [89], which exploited the concept of smartphone virtualization in cloud computing and provided method-level computation offloading. [90] utilized opportunistic communications to facilitate information dissemination and data offloading in mobile social networks, which can significantly reduce the amount of mobile data traffic. [91] considered a market-based mobile data offloading solution, which utilized the non-cooperative game theory to decide how much traffic should each access point (AP) offload for each base station (BS) and what is the corresponding payment. [92] extended [91] by formulating the offloading problem based on the network utility maximization [93] framework, and proposed an iterative double auction mechanism to solve it.

In this work, we propose a distributed mechanism for mobile data offloading in SDN at the netowrk edge. The SDN controller dynamically routes data traffic in a mobile network to decide how much data should APs offload for BSs. A total revenue maximization problem is formulated by jointly considering the offloading utility of BSs and the cost of APs. The optimization problem is solved in a distributed fashion based on the proximal Jacobian multi-block alternating direction method of multipliers (ADMM). The BSs and APs perform the offloading decision update concurrently, and are coordinated by the SDN controller through dual variables to reach a consensus on offloading demand and supply. The proposed mechanism has following characteristics.

1. Simple computation at the SDN controller: To alleviate the computation burden of mobile data offloading at the SDN controller, the operations at the SDN controller is designed to be simple one-time algebraic calculation instead of solving an optimization in related work [92].

2. Privacy preserving: During the process of the optimization for offloading decision, the utility functions at BSs and cost functions at APs are only known to themselves.

3. Concurrent update at BSs and APs: The updating process at BSs and APs are performed

concurrently.

4.1.2 System Model and Problem Formulation

We consider a mobile network which consists of B cellular base stations (BSs) and A access points (APs). A BS $b \in \{1, \ldots, B\}$ serves a group of mobile users and has the demand to offload its traffic to APs. An AP $a \in \{1, \ldots, A\}$ is a WiFi or femtocell AP which operates in a different frequency band and supplies its bandwidth for data offloading. The maximum available capacity for data offloading of each AP a is denoted by C_a. The SDN controller manages BSs and APs through the ANDSF, and makes mobile data offloading decisions according to various trigger criteria. Such criteria can be the number of mobile users per BS, available bandwidth/IP address of each BS, or aggregate number of flows on a specific port at a BS.

Let $\mathbf{x}_b = [x_{b1}, \ldots, x_{bA}]^\top$ represents offloaded traffic of BS b, where x_{ba} denotes the data traffic of BS b offloaded through AP a. Correspondingly, $\mathbf{y}_a = [y_{a1}, \ldots, y_{aB}]^\top$ represents admitted traffic of AP a, where y_{ab} represents the admitted data traffic from BS b. Generally, a feasible mobile data offloading decision exists when BSs and APs reach an agreement on the amount of offloading data, i.e., $x_{ba} = y_{ab}, \forall a$ and $\forall b$. We assume that mobile data of BSs can be offloaded to all of APs without loss of generality. Moreover, we assume that the time is slotted and during each time slot the offloading demand from BSs is fixed. The SDN controller needs to find a feasible offloading schedule at the beginning of each time slot, and maximize the utility of BSs at a reasonable cost of APs.

We denote BS b's utility by $U_b(\mathbf{x}_b)$, where $U_b(\cdot)$ is designed to be a non-decreasing, non-negative and concave function in $\mathbf{x}_b, \forall b$. For example, the function can be logarithmic, and the concavity is due to the diminishing returns of resources allocated to the offloaded data. Likewise, we use function $L_a(\mathbf{y}_a)$ to describe the AP a's cost of helping BSs offload data, where $L_a(\cdot)$ is a non-decreasing, non-negative and convex function in $\mathbf{y}_a, \forall a$. The cost function can be a linear cost function, which means total cost of APs will increase as the amount of admitted mobile data increases.

For the SDN controller, the total revenue for mobile data offloading is expressed as $\sum_{b=1}^{B} U_b(\mathbf{x}_b) - \sum_{a=1}^{A} L_a(\mathbf{y}_a)$. To maximize the total revenue, the equivalent minimization optimization problem can be formulated as

$$\min_{\{\mathbf{x}_1,\ldots,\mathbf{x}_B\},\{\mathbf{y}_1,\ldots,\mathbf{y}_A\}} \quad \sum_{a=1}^{A} L_a(\mathbf{y}_a) - \sum_{b=1}^{B} U_b(\mathbf{x}_b), \tag{4.1}$$

$$\text{s.t} \quad \sum_{b=1}^{B} y_{ab} \leq C_a, \quad \forall a, \text{ and} \tag{4.2}$$

$$x_{ba} = y_{ab}, \quad \forall a, b, \tag{4.3}$$

where (4.2) stands for the capacity constraint at each AP, and (4.3) represents the consensus of BSs and APs on the amount of mobile data. We propose an algorithm based on ADMM to solve the convex optimization problem (4.1) in a fully distributed fashion.

4.1.3 A Distributed ADMM Approach

The optimization problem (4.1) can be solved in a fully distributed fashion by the multi-block Jacobian ADMM. The computing paradigm of the proposed algorithm is shown in Fig. 4.2. During each iteration, BSs and APs update $\mathbf{x}$ and $\mathbf{y}$ concurrently. The updated $\mathbf{x}$ and $\mathbf{y}$ are gathered by the SDN controller, which performs a simple update on $\boldsymbol{\lambda}$ and scatters dual variables back to BSs and APs. The iteration goes on until a consensus on the offloading demand and supply is reached. According to Algorithm 2.7, we fist calculate the partial Lagrangian of (4.1), which introduces the Lagrange multipliers only for constraint (4.3)

$$\mathcal{L}_\rho(\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda}) = \sum_{a=1}^{A} L_a(\mathbf{y}_a) - \sum_{b=1}^{B} U_b(\mathbf{x}_b) + \sum_{a=1}^{A}\sum_{b=1}^{B} \lambda_{ab}(x_{ba} - y_{ab}) + \frac{\rho}{2}\sum_{a=1}^{A}\sum_{b=1}^{B} \|x_{ba} - y_{ab}\|_2^2, \tag{4.4}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^{AB}$ is the Lagrange multiplier and ρ is the penalty parameter. The updates of BSs and APs can be performed concurrently according to the proximal Jacobian multi-block ADMM. We describe update procedures of BSs, APs, and SDN controller as follows.

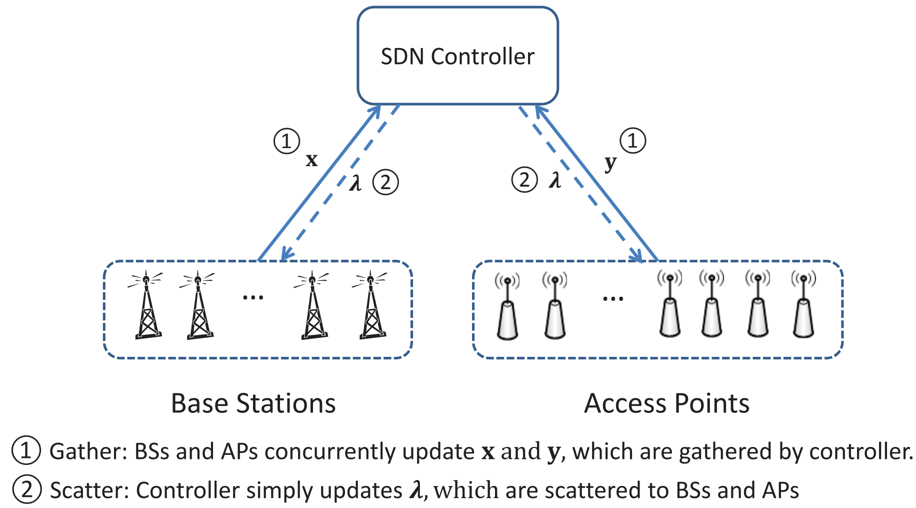


Figure 4.2 Distributed computing paradigm of proposed distributed mobile data offload mechanism.

Base Station Update: At each BS b, the update rule can be expressed as

$$\mathbf{x}_b^{k+1} = \arg\min_{\mathbf{x}_b} (-U_b(\mathbf{x}_b) + \frac{\rho}{2} \sum_{a=1}^{A} \|x_{ba} - p_{ab}^k\|_2^2 + \frac{1}{2}\|\mathbf{x}_b - \mathbf{x}_b^k\|_{\mathbf{P}_i}^2), \tag{4.5}$$

where $\mathbf{P}_i = 0.1\mathbf{I}$, and $\mathbf{I}$ is the identity matrix. The $p_{ab}^k = (y_{ab}^k - \frac{\lambda_{ab}^k}{\rho}), \forall a$ is the 'signal' from the SDN controller to BS b. The update (4.5) is a small scale unconstrained convex optimization problem. At each round of the update, it sends $\mathbf{x}_b$ of size A to the SDN controller. Note that the update of each BS is performed independently and can be calculated locally. Once $\mathbf{x}_b$ is updated, it is sent to the SDN controller while the utility function $U_b(\cdot)$ is kept confidential.

Access Point Update: The update rule at each AP a can be expressed as

$$\mathbf{y}_a^{k+1} = \arg\min_{\mathbf{y}_b} (L_a(\mathbf{y}_a) + \frac{\rho}{2} \sum_{b=1}^{B} \|y_{ab} - q_{ba}^k\|_2^2 + \frac{1}{2}\|\mathbf{y}_a - \mathbf{y}_a^k\|_{\mathbf{P}_i}^2), \quad \text{s.t} \quad \sum_{b=1}^{B} y_{ab} \leq C_a, \tag{4.6}$$

where $\mathbf{P}_i = 0.1\mathbf{I}$, and $q_{ba}^k = (x_{ba}^k + \frac{\lambda_{ab}^k}{\rho}), \forall b$. The q_{ba} is the 'signal' from the SDN controller to AP a. The update (4.6) is a small-scale convex optimization problem with linear inequality constraints. At each round of the update, it sends $\mathbf{y}_a$ of size B to the SDN controller. The update of each AP is also performed independently. During the update, the information of cost function $L_a(\cdot)$ is kept private. The $\mathbf{y}_a$ is sent to the SDN controller once updated.

Algorithm 4.1 Distributed mobile data offloading.

Initialize: $\mathbf{x}^0$,$\mathbf{y}^0$ $\boldsymbol{\lambda}^0$, $\rho > 0$, $\gamma > 0$;

for $k = 0, 1, \ldots$ **do**
 $\{$Update $\mathbf{x}_b$ and $\mathbf{y}_a$ for $b = 1, \ldots, B$ and $a = 1, \ldots, A$, **concurrently**.$\}$
 $\{$**Base station update,** $\forall b\}$
 $\mathbf{x}_b^{k+1} = \arg\min_{\mathbf{x}_b} -U_b(\mathbf{x}_b) + \frac{\rho}{2}\sum_{a=1}^{A} \|x_{ba} - y_{ab}^k + \frac{\lambda_{ab}^k}{\rho}\|_2^2 + \frac{1}{2}\|\mathbf{x}_b - \mathbf{x}_b^k\|_{\mathbf{P}_i}^2$;
 $\{$**Access point update,** $\forall a\}$
 $\mathbf{y}_a^{k+1} = \arg\min_{\mathbf{y}_b} L_a(\mathbf{y}_a) + \frac{\rho}{2}\sum_{b=1}^{B} \|x_{ba}^k - y_{ab} + \frac{\lambda_{ab}^k}{\rho}\|_2^2 + \frac{1}{2}\|\mathbf{y}_a - \mathbf{y}_a^k\|_{\mathbf{P}_i}^2$;
 $\{$**SDN controller update**$\}$
 $\lambda_{ab}^{k+1} = \lambda_{ab}^k + \gamma\rho\sum_{b=1}^{B}\sum_{a=1}^{A}(x_{ba}^{k+1} - y_{ab}^{k+1})$;
end for

Output $\mathbf{x}$, $\mathbf{y}$;

SDN Controller Update: At the SDN controller, the update rule can be expressed as

$$\lambda_{ab}^{k+1} = \lambda_{ab}^k + \gamma\rho\sum_{b=1}^{B}\sum_{a=1}^{A}(x_{ba}^{k+1} - y_{ab}^{k+1}). \tag{4.7}$$

After gathering $\mathbf{x}$ and $\mathbf{y}$ from BSs and APs, the SDN controller performs a simple update on the dual variable $\boldsymbol{\lambda}$ by a simple algebra operation. After that, the 'signal' variables p_{ba} and q_{ba} are scattered back to corresponding BSs and APs, respectively.

Remark that in the Jacobian type update, the iterations of BSs and APs are performed concurrently. There is no direct communication between BSs and APs, and the intermediated update results of $\mathbf{x}$ and $\mathbf{y}$ are kept private. The updates at iteration $k+1$ only depend on previous values at iteration k, which enables a fully distributed implementation.

At each iteration, the update operations at BSs and APs are quite simple. The updates at each BS and AP are simple small-scale convex optimization problems, which can be quickly solved by many off-the-shelf tools like CVX [94]. As for the communication overhead, for each iteration the message between each BS and the SDN controller is of size $2A$ (size of $\mathbf{x}_b$ and $p_{ba}, \forall a$). Likewise, the message between each AP and the SDN controller is of size $2B$ (size of $\mathbf{y}_a$ and $q_{ba}, \forall b$). The sizes of those messages are quite small compare with the size of offloading data. The proposed distributed algorithm is described in Algorithm 4.1.

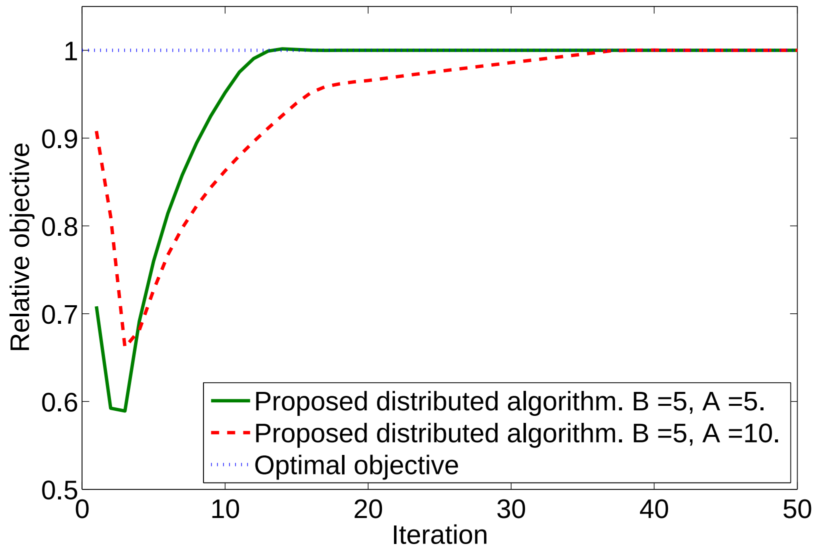


Figure 4.3 Convergence performance of the proposed distributed mobile data offloading algorithm by objective value when $(B = 5, A = 5)$ and $(B = 5, A = 10)$.

4.1.4 Numerical Results

4.1.4.1 Evaluation Settings

We consider a wireless access network consists of $B = 5$ base stations and $A = \{5, 10\}$ access points coordinated by the SDN controller. The SDN controller will offload mobile data traffic of BSs to APs, and the available capacity of each AP for offloading is $C_a = 10Mbps$. The utility function of BS b is $U_b(\mathbf{x}_b) = \log(\mathbf{x}_b^\top \mathbf{1} + 1)$, where $\mathbf{1}$ is the all one vector. The cost function of AP a is a linear cost expressed as $L_a(\mathbf{y}_a) = \theta_a * \mathbf{y}_a^\top \mathbf{1}$, where $\theta_a > 0$ is the cost coefficient. The value of θ_a is application specific. During numerical tests, we assume θ_a is a Gaussian random variable which has a distribution $\mathcal{N}(0, 1)$ for simplicity. We perform numerical tests on the offloading decision for one time slot, and simulation results are presented as follows.

4.1.4.2 Convergence Performance

We investigate the convergence performance of the proposed algorithm in the sense of optimization objective and residual. Two scenarios, $(B = 5, A = 5)$ and $(B = 5, A = 10)$, are considered here. Since different scenarios have different optimal objectives, we use the relative

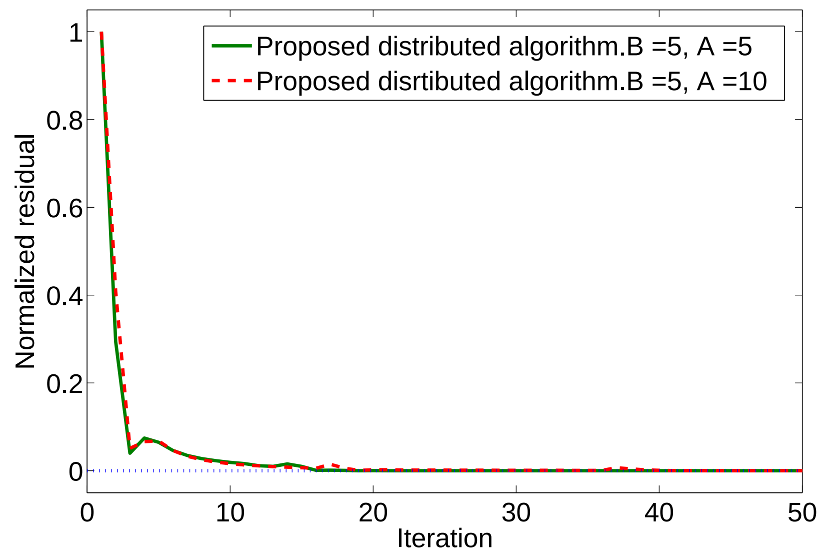


Figure 4.4 Convergence performance of the proposed distributed mobile data offloading algorithm by residual when $(B = 5, A = 5)$ and $(B = 5, A = 10)$.

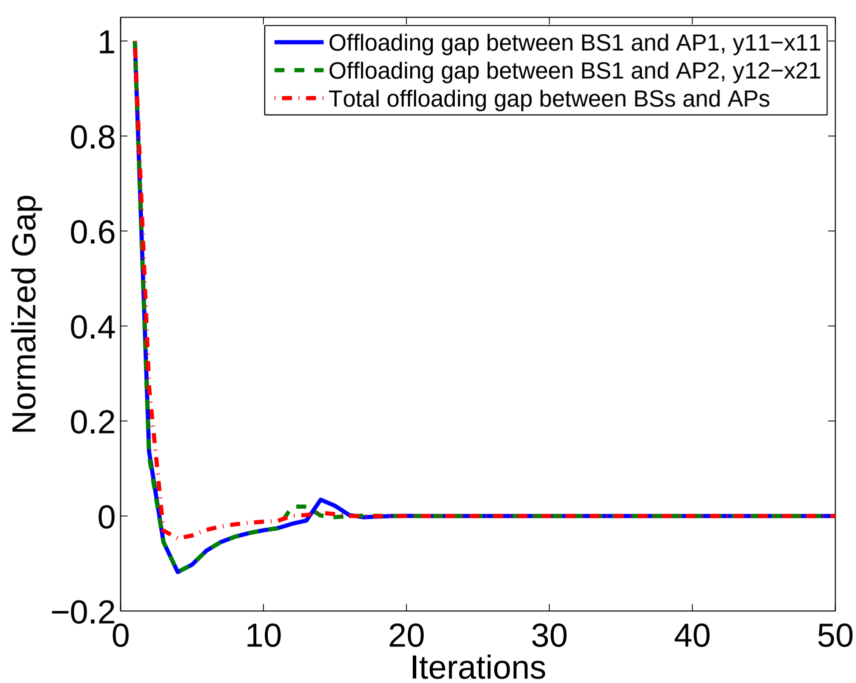


Figure 4.5 The offloading gap of the proposed distributed mobile data offloading algorithm.

objective o^k/o^* in our simulations. The o^k is the objective value calculated by the proposed distributed algorithm at iteration k, and o^* is the optimal objective obtained by the centralized method. The residual is defined as $\sum_{a=1}^{A}\sum_{b=1}^{B}\|x_{ba}-y_{ab}\|_2^2$. We also normalize the maximal residual to 1 for better readability.

The convergence of the proposed distributed algorithm is shown in Fig. 4.3 and Fig. 4.4. Fig. 4.3 shows that the proposed algorithm converges to the optimal objective in a moderate number of iterations when $B = 5$ and $A = 5$. It takes a longer time for the proposed algorithm to converge when $A = 10$. It indicates that when these are more APs in the access network, it will take a longer time for the SDN controller to coordinate BSs and APs for a consensus on the offloading demand and supply. The normalized residual is shown in Fig. 4.4. It is shown that after several times of iterations the residual of optimization problem (4.1) reduces to zero for both scenarios.

4.1.4.3 Offloading Performance

We study the performance of mobile data offloading by considering the offloading gap between BSs demand and APs supply. Here we only consider the scenario $(B = 5, A = 5)$. Note that a feasible offloading exists when $x_{ba} = y_{ab}, \forall a$ and $\forall b$. Thus the total market gap, $\sum_{a=1}^{A}\sum_{b=1}^{B}(x_{ba}-y_{ab})$, is calculated here. To understand the efficiency of the proposed algorithm for each base station, we zoom in gaps between BS 1 $\&$ AP 1 ($y_{11}-x_{11}$), and BS 1 $\&$ AP 2 ($y_{12}-x_{21}$), respectively. Due to different scales of those gaps, we normalized their maximum to 1.

The numerical results are shown in Fig. 4.5. It is shown that after several times of iterations, the total market gap reduces to zero, which means that BSs and APs have reached on a consensus on mobile data offloading demand and supply. The offloading gaps between BS 1 $\&$ AP 1 ($y_{11}-x_{11}$), and BS 1 $\&$ AP 2 ($y_{12}-x_{21}$) also gradually converge to zero. Note that the convergence of those gaps are not necessary synchronized. When those gaps all converge to zero, the decision of mobile data offloading is made and the maximal total revenue is achieved.

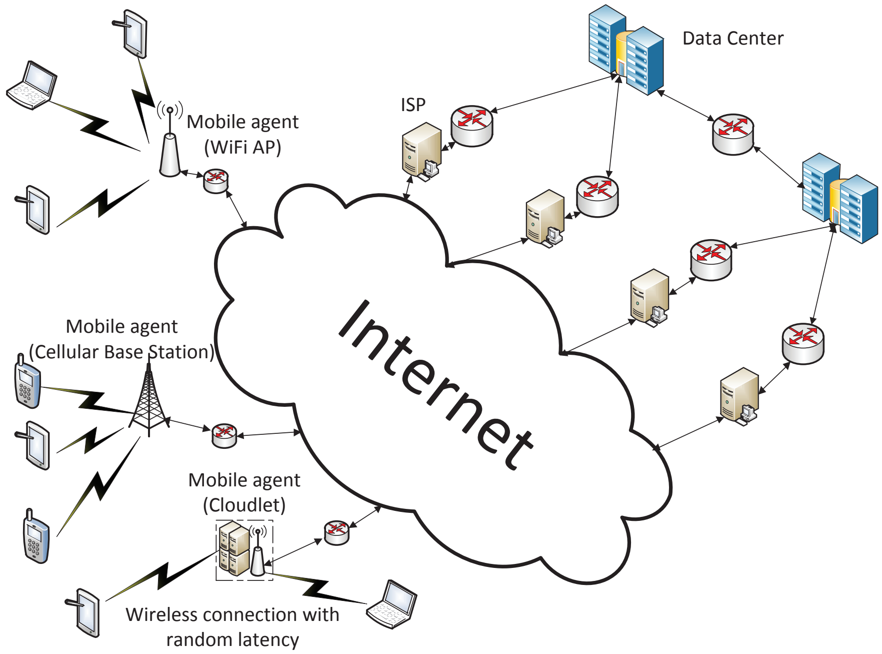


Figure 4.6 An illustration of mobile cloud computing infrastructure.

4.2 Scalable Service Management in Mobile Cloud Computing

In this section, we first give a brief introduction to mobile cloud computing (MCC). Then, the proposed algorithm for mobile cloud service management is described. Finally, we presented numerical results of the proposed algorithm.

4.2.1 An Introduction to Mobile Cloud Computing

4.2.1.1 Background

In mobile cloud computing, mobile end users can offload local workload [95] and back up personal data to clouds without explicitly noticed where the service is actually hosted. The service provider needs to dynamically acquire computing resources for service provisioning, and delicately manage online services to optimize the end-to-end performance experienced by their customers. It is known that even a small increase in latency will result in a significant revenue loss for service providers. Thus, mobile service providers usually deploy their services on several cloud-enabled data centers, and perform service management tasks to optimally locate mobile service instances. An illustration of the mobile cloud computing infrastructure is shown in Fig. 4.6.

To efficiently manage mobile cloud services, a mobile service provider should appropriately locate client requests to a data center (request allocation), and select an upstream Internet service provider (ISP) link of data center to carry on the traffic back to the client (response routing). Those two tasks are crucial to the success of mobile cloud service, and should be managed adaptively to variations in MCC, such as end user demands, link latency, computation costs, as well as electricity and bandwidth price. Nowadays, the decision of request allocation and response mapping is handled separately, which results in poor service performance and high cost. For example, too many client requests may be allocated to the same data center with limited upstream link bandwidth, or a data center may response to client requests through an expensive ISP link. The management tasks are also computationally intensive due to the large number of mobile devices and the stringent response-time requirement of mobile services. Furthermore, the uncertainty in the wireless link latency of mobile network complicates the problem.

4.2.1.2 Related Work

The service management faced by the mobile service provider can be seen as a network utility maximization (NUM) problem [93], which described a unifying framework for understanding and designing distributed control and resource allocation in communication networks.

Our work is closely related to the mobile service allocation and the traffic engineering in MCC. The framework for offloading mobile computation workload to clouds was proposed in [89], which managed to enhance the energy resource utilization and reduced the computation time of mobile devices. In [96], a mobile service management technology was presented to support novel MCC applications. The network services were reactively relocated to guarantee adequate performance for the client-sever communication. A decentralized design for service request allocation was described in [97], which directed client requests to appropriate server replicas to offer better performance. The problem of optimizing the performance of carrying traffic for an online service provider was studied in [98]. The multi-homed traffic engineering for autonomous systems to optimize cost and performance was investigated in [99], and the effect of temperature on cloud service workload management for geo-distributed data centers was analyzed in [100].

The cooperative server selection and traffic engineering between network and content providers who have conflict objectives was proposed in [101], the concept of Nash bargaining solution was utilized to enhance the cooperation between ISPs and content providers. [102] extended the optimality result by incorporating practical considerations such as DC-level load balancing and capacity constraints. Recent work [103] considered a coordination of request mapping and response routing for geo-distributed cloud services, and developed a distributed algorithm to solve the large-scale optimization problem. Our work explores and analyzes the effect of random wireless nature on the service management problem, and proposes a distributed stochastic optimization framework with proved convergence property for service management.

In this work, we present a scalable distributed management framework for mobile cloud services, which takes the the impact of wireless network characteristics into account. The tasks of clients request allocation and data center response routing are jointly considered, and the management tasks are formulated as a service revenue maximization problem. In particular, the mobile service provider optimally locates client requests to provide qualified service at a reasonable cost under the stochastic wireless link latency. Our major contributions are as follows.

1. A stochastic optimization framework for mobile cloud service management is formulated. The clients request allocation and data center response routing are jointly optimized, and the impact of wireless network characteristics on service performance is considered.

2. A distributed approach to solve the large-scale stochastic optimization problem based on ADMM is proposed. The update steps are modified according to the stochastic setting, which can be solved in a parallel fashion on distributed agents and coordinated through dual variables.

3. We prove the convergence of the proposed stochastic distributed optimization algorithm. We

Table 4.1 Summary of key notations.

$\mathcal{I}$	Set of agents, indexed by $i \in \{1, \ldots, N\}$.
$\mathcal{J}$	Set of data centers, indexed by $j \in \{1, \ldots, J\}$.
$\mathcal{K}_j$	Set of ISP links of data center j.
R_i	Bandwidth capacity of mobile service agent i.
$a_{i,j}$	Application request of client i severed by data center j.
$\mathbf{b}_{j,i}$	Traffic routed from data center j to agent i.
$\mathbf{L}_{j,i}$	Average delay from data center j to agent i.
ξ_i	Latency variation of wireless link for agent i.
$F_i(\cdot)$	Utility function of agent i.
$G_j(\cdot)$	Performance metric function of data center j.
$\mathbf{p}_{j,i}$	Price of routing traffic from data center j to agent i.
$\mathbf{Q}_j$	Capacity of ISP links at data center j.
C_j	Capacity of data center j.

evaluate the effectiveness of the proposed algorithm through numerical simulations from both computation perspective and service management perspective.

4.2.2 System Model and Problem Formulation

We first present the mobile cloud infrastructure, and then describe the mobile cloud service management problem. The summary of key notations are listed in Table 4.1.

4.2.2.1 Mobile Cloud Infrastructure

We consider a set $\mathcal{I}$ of agents in mobile cloud service. An agent $i \in \{1, \ldots, N\}$ is defined as an access point (AP) of wireless access networks, and the bandwidth capacity of mobile service agent i is R_i. A set $\mathcal{J}$ of data centers are indexed by $j \in \{1, \ldots, J\}$. Data centers are interconnected over a backbone network and each data center is multi-homed to K ISP links. The set of ISP links of data center j is denoted by $\mathcal{K}_j$. We assume that all data centers have the same number of ISP links for simplicity. The capacity of ISP links at data center j is denoted by $\mathbf{Q}_j = [Q_{j,1}, \ldots, Q_{j,K}]^\top$. The service provider observes a propagation delay $\mathbf{L}_{j,i} = [L_{j,i,1}, \ldots, L_{j,i,K}]^\top$ over wired connection, where $L_{j,i,k}$ is the average delay between agent i and data center j on the k^{th} ISP link. The wireless link latency between agent i and mobile devices is ξ_i.

The requests from mobile devices are first handled by the mobile service agent. After that, one data center at a specific location is assigned to process the request. We use $a_{i,j}$ as agent i's application requests processed by data center j. The request allocation decision variables of mobile service agent i are denoted as $\mathbf{a}_i = [a_{i,1}, \ldots, a_{i,J}]^\top$. In practices, a mobile service agent can be a cloudlet [104] or be implemented on servers that provide mobile network services. Additionally, in this work we assume that an agent has the fine-grained control of the network traffic, which is a reasonable assumption in nowadays commercial products [97] and techniques like OpenFlow [85]. Once data center j has finished the job, the response traffic will be routed back through ISP links. We use vector $\mathbf{b}_{j,i} = [b_{j,i,1}, \ldots, b_{j,i,K})]^\top$ to denote the traffic routed from the data center j to agent i through K different ISP links, and the matrix $\mathbf{B}_j = [\mathbf{b}_{j,1}, \ldots, \mathbf{b}_{j,N}]^\top$ denote response routing decisions of data center j.

4.2.2.2 Mobile Cloud Service Management

In mobile cloud service management, the application requests are allocated to appropriate data centers in order to achieve maximal utility and minimize the cost. The utility and cost functions in the service management can be elaborated as follows.

1) **Utility of mobile service agents**: The performance objective of agent i is characterized by a utility function $F_i(\cdot)$, which depends on total transmission rate and wireless access network latency. The utility functions can be different among mobile service agents. In this work, $F_i(\cdot)$ is designed to be a non-decreasing, non-negative and concave function in $\sum_{j \in \mathcal{J}} a_{i,j}$. For example, $F_i(\mathbf{a}_i, \xi_i) = \frac{1}{\xi_i} \log_2(\sum_{j \in \mathcal{J}} a_{i,j} + 1)$, or can be a more general class of functions that represent the elasticity of service request and/or determine the fairness of resource allocation. Such functions are typically used for the TCP congestion control [105, 106].

2) **Cost of data centers**: The cost of data center j is characterized by function G_j as

$$G_j(\cdot) = \beta_{j,1}\gamma_{j,1}(\cdot) + \beta_{j,2}\gamma_{j,2}(\cdot) - \beta_{j,3}\gamma_{j,3}(\cdot), \tag{4.8}$$

which has three parts and parameterized by positive coefficients $\beta_{j,1}$, $\beta_{j,2}$ and $\beta_{j,3}$, respectively, to incorporate different degrees of sensitivity to operation cost, link price and user perceived latency. The determination of the values of coefficients is service specific, and the mobile service provider is responsible for choosing the values of parameters based on its service types, data centers it used, and its profit model.

In the first part of $G_j(\cdot)$, $\gamma_{j,1}(\cdot)$, accounts for the operation cost of data center j. Here, $\gamma_{j,1}(\cdot)$ is designed to be a non-decreasing, non-negative and convex function in $\sum_{i\in\mathcal{I}} \mathbf{b}_{j,i}^\top \mathbf{1}$, where $\mathbf{1} \in \mathbb{R}^K$ is an all-one vector. The design of operation cost can incorporate the price of computing resource rental, maintenance cost, and electricity bills [100, 107, 108]. For example, to represent electricity bills [109] at data center j, $\gamma_{j,1}(\cdot)$ can be

$$\gamma_{j,1}(\mathbf{B}_j) = Pr_j \times P_e \times [P_{idle} + (P_{peak} - P_{idle}) \sum_{i\in\mathcal{I}} \mathbf{b}_{j,i}^\top \mathbf{1}], \tag{4.9}$$

where Pr_j is the spot electricity price at data center j, and P_e is the power usage efficiency. P_{peak} and P_{idle} are server peak power and server idle power, respectively.

In the second part of $G_j(\cdot)$, $\gamma_{j,2}(\cdot)$, stands for the cost of routing traffic from data center j to mobile service agents through ISP links. A linear cost model for ISP links can be adopted

$$\gamma_{j,2}(\mathbf{B}_j) = \sum_{i\in\mathcal{I}} \mathbf{b}_{j,i}^\top \mathbf{p}_{j,i}, \tag{4.10}$$

where $\mathbf{p}_{j,i} = (p_{j,i,1}, \ldots, p_{j,i,K})^\top$ is the price vector for ISP links at data center j. We assume that the cost of routing traffic on the k^{th} ISP's link from data center j to agent i, which is denoted by $p_{j,i,k}$, is known and fixed. Note that nowadays ISPs are adopting sophisticated charging policy, e.g., the 95-percentile charging scheme. It is shown that a linear cost optimization in charging intervals can reduce the 95-percentile cost [98].

In the last part of $G_j(\cdot)$, $\gamma_{j,3}(\cdot)$, captures the user-perceived latency in the response routing from data center j to mobile service agents. $\gamma_{j,3}(\cdot)$ can be a non-decreasing, non-negative and concave function in $\mathbf{B}_j$

$$\gamma_{j,3}(\mathbf{B}_j) = \sum_{i \in \mathcal{I}} \mathbf{b}_{j,i}^{\top}(\mathbf{L}_{max} - \mathbf{L}_{j,i}), \tag{4.11}$$

where $\mathbf{L}_{max}$ is the maximum tolerable latency. The ISP link delay $\mathbf{L}_{j,i}$ is known and can be obtained through active measurements [110]. We consider the latency between data center j and mobile service agents as a performance metric since user-perceived latency is one of the most important metrics for mobile cloud computing service. Even a small increment can result in a significant revenue loss.

3) **Total revenue**: The goals of maximizing mobile service utility and minimizing data centers' cost usually contradict each other. Allocating users' requests to data centers that offer lower latencies usually incurs higher costs, and over-utilizing the low-cost link for response routing will degrade system performance due to the increased congestion. By jointly considering utilities of mobile service agents and cost of data centers, the total revenue for mobile cloud service management can be formulated as

$$\text{Revenue} = \alpha \sum_{i \in \mathcal{I}} \mathbb{E}_{\xi_i}\{F_i(\mathbf{a}_i, \xi_i)\} - \sum_{j \in \mathcal{J}} G_j(\mathbf{B}_j), \tag{4.12}$$

where the parameter α is introduced to find a balance between service utility and cost. The mobile service provider needs to customize the cost-performance tradeoff to obtain the best revenue.

4.2.2.3 Maximizing Total Revenue

The mobile service provider performs a revenue maximization to improve resource utilization in MCC. The objective of the optimization problem consists of two terms: (i) the service utility from all agents by fulfilling mobile client requests, and (ii) the data center cost for serving service requests. The resulting stochastic optimization problem is presented in its equivalent minimization form as

$$\underset{\{\mathbf{a}_i\}_{i=1}^N,\{\mathbf{B}_j\}_{j=1}^J}{\text{minimize}} \quad \sum_{j\in\mathcal{J}} G_j(\mathbf{B}_j) - \alpha \sum_{i\in\mathcal{I}} \mathbb{E}_{\xi_i}\{F_i(\mathbf{a}_i,\xi_i)\} \tag{4.13}$$

$$\text{subject to} \quad \sum_{j\in\mathcal{J}} a_{i,j} \le R_i, \quad \forall i, \tag{4.14}$$

$$\sum_{i\in\mathcal{I}} a_{i,j} \le C_j, \quad \forall j, \tag{4.15}$$

$$\sum_{i\in\mathcal{I}} \mathbf{b}_{j,i} \preceq \mathbf{Q}_j, \quad \forall j, \tag{4.16}$$

$$a_{i,j} = \mathbf{b}_{j,i}^\top \mathbf{1}, \quad \forall i,j, \text{ and} \tag{4.17}$$

$$a_{i,j} \ge 0, \quad \mathbf{b}_{j,i} \succeq 0, \quad \forall i,j, \tag{4.18}$$

where (4.14) is the bandwidth capacity constraint for each mobile service agent. (4.15) and (4.16) are data center capacity constraint and link capacity constraint, respectively. (4.17) is the workload conservation constraint between each pair of agent and data center.

The solution to the above stochastic optimization problem ensures the optimal allocation of mobile application requests, while achieving the maximum revenue. Traditionally, this problem is solved in a centralized manner to find the optimal solution. However, the major challenge of the centralized service revenue maximization is the problem size, especially for large systems with an enormous number of agents, communication links and data centers. Additionally, the randomness of the wireless link latency makes the problem more complicated. To achieve efficient and scalable management of the mobile cloud service, a distributed optimization framework is proposed.

4.2.3 Distributed Stochastic ADMM for Service Management

In this section, we first introduce the background of stochastic ADMM and analyze its convergence property. Then we present the proposed method for mobile cloud service management.

4.2.3.1 Stochastic ADMM Background

The general form of stochastic ADMM can be expressed as

$$\begin{aligned} &\underset{\mathbf{x}\in\mathcal{X},\mathbf{z}\in\mathcal{Z}}{\text{minimize}} \quad \mathbb{E}_{\xi}\{f(\mathbf{x},\boldsymbol{\xi})\} + g(\mathbf{z}) \\ &\text{subject to} \quad \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{c}, \end{aligned} \tag{4.19}$$

where $\boldsymbol{\xi}$ is a random variable with unknown distribution. Note that the first function in (4.19) is an expectation function over $\boldsymbol{\xi}$ instead of a deterministic function in (2.8). The Lagrangian function associated with (4.19) is

$$\mathcal{L}_{\rho}(\mathbf{x},\mathbf{z},\boldsymbol{\lambda}) = \mathbb{E}_{\xi}\{f(\mathbf{x},\boldsymbol{\xi})\} + g(\mathbf{z}) + \langle \boldsymbol{\lambda}, \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}\rangle + \frac{\rho}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}\|_2^2, \tag{4.20}$$

and update rules for $\mathbf{x}$, $\mathbf{z}$ and $\boldsymbol{\lambda}$ are

$$\mathbf{x}^{t+1} = \arg\min_{\mathbf{x}} f(\mathbf{x}^t) + \langle \nabla f(\mathbf{x}^t, \boldsymbol{\xi}^t), \mathbf{x}\rangle + \frac{\|\mathbf{x} - \mathbf{x^t}\|_2^2}{2\eta^t} + \langle \boldsymbol{\lambda}^t, \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z}^t - \mathbf{c}\rangle + \frac{\rho}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z}^t - \mathbf{c}\|_2^2, \tag{4.21}$$

$$\mathbf{z}^{t+1} = \arg\min_{\mathbf{z}} g(\mathbf{z}) + \langle \boldsymbol{\lambda}^t, \mathbf{A}\mathbf{x}^{t+1} + \mathbf{B}\mathbf{z} - \mathbf{c}\rangle + \frac{\rho}{2}\|\mathbf{A}\mathbf{x}^{t+1} + \mathbf{B}\mathbf{z} - \mathbf{c}\|_2^2, \text{ and} \tag{4.22}$$

$$\boldsymbol{\lambda}^{t+1} = \boldsymbol{\lambda}^t + \rho(\mathbf{A}\mathbf{x}^{t+1} + \mathbf{B}\mathbf{z}^{t+1} - \mathbf{c}), \tag{4.23}$$

where η^t is the penalty parameter and is essential for the algorithm convergence. The update rule of $\mathbf{x}$ has the same flavor of the stochastic mirror descent method [111,112]. The convergence property of stochastic ADMM can be analyzed using the variational inequality (VI) base on the Lagrangian (4.20) similar to the deterministic case [113,114]. Before introducing the proof of convergence, we describe following lemmas which will be useful for the proof.

Lemma 4.1. *[115] If $\mathbf{y}^* = \arg\min_{\mathbf{y}\in\mathcal{Y}}\{J_1(\mathbf{y}) + J_2(\mathbf{y})\}$, where $J_1 : \mathbb{R}^n \mapsto \mathbb{R}$ and $J_2 : \mathbb{R}^n \mapsto \mathbb{R}$ are convex functions, $\mathcal{Y}$ is a polyhedral subset of $\mathbb{R}^n$, and J_2 is continuously differentiable, then*

$$\mathbf{y}^* = \arg\min_{\mathbf{y}\in\mathcal{Y}}\{J_1(\mathbf{y}) + \nabla J_2(\mathbf{y}^*)^\top \mathbf{y}\}. \tag{4.24}$$

Proof. The $\mathbf{y}^* = \arg\min_{\mathbf{y}\in\mathcal{Y}}\{J_1(\mathbf{y}) + J_2(\mathbf{y})\}$ is equivalent to

$$(\mathbf{y}^*, \mathbf{y}^*) = \arg\min_{\mathbf{y}\in\mathcal{Y},\mathbf{z}\in\mathbb{R}^n,\mathbf{y}=\mathbf{z}}\{J_1(\mathbf{y}) + J_2(\mathbf{z})\}. \tag{4.25}$$

By the Lagrange multiplier theorem [115], there exists $\boldsymbol{\lambda} \in \mathbb{R}^n$ such that

$$\mathbf{y}^* = \arg\min_{\mathbf{y}\in\mathcal{Y}} \{J_1(\mathbf{y}) + \boldsymbol{\lambda}^\top \mathbf{y}\} \text{ and} \tag{4.26}$$

$$\mathbf{y}^* = \arg\min_{\mathbf{z}\in\mathbb{R}^n} \{J_2(\mathbf{z}) - \boldsymbol{\lambda}^\top \mathbf{z}\}. \tag{4.27}$$

From (4.27) we obtain $\boldsymbol{\lambda} = \nabla J_2(\mathbf{y}^*)$. Substitute $\boldsymbol{\lambda}$ back to (4.26) proves the result. $\square$

Lemma 4.2. *For $t \geq 0$, we have that the following inequality holds:*

$$g(\mathbf{z}^{t+1}) - g(\mathbf{z}) + \langle \mathbf{z}^{t+1} - \mathbf{z}, \mathbf{B}^\top \boldsymbol{\lambda}^{t+1}\rangle \leq 0. \tag{4.28}$$

Proof. By applying Lemma 4.1 with identifications $J_1(\mathbf{z}) = g(\mathbf{z})$ and $J_2(\mathbf{z}) = \langle \boldsymbol{\lambda}^t, \mathbf{A}\mathbf{x}^{t+1} + \mathbf{B}\mathbf{z} - \mathbf{c}\rangle + \frac{\rho}{2}\|\mathbf{A}\mathbf{x}^{t+1} + \mathbf{B}\mathbf{z} - \mathbf{c}\|_2^2$ in (4.22) we have

$$g(\mathbf{z}) - g(\mathbf{z}^{t+1}) + \langle \mathbf{z} - \mathbf{z}^{t+1}, \mathbf{B}^\top[\boldsymbol{\lambda}^t + \rho(\mathbf{A}\mathbf{x}^{t+1} + \mathbf{B}\mathbf{z}^{t+1} - \mathbf{c})]\rangle \geq 0. \tag{4.29}$$

Substituting $\boldsymbol{\lambda}^{t+1} = \boldsymbol{\lambda}^t + \rho(\mathbf{A}\mathbf{x}^{t+1} + \mathbf{B}\mathbf{z}^{t+1} - \mathbf{c})$ in to (4.29), we get

$$g(\mathbf{z}^{t+1}) - g(\mathbf{z}) + \langle \mathbf{z}^{t+1} - \mathbf{z}, \mathbf{B}^\top \boldsymbol{\lambda}^{t+1}\rangle \leq 0, \tag{4.30}$$

which proves the result. $\square$

Lemma 4.3. *For $t \geq 0$, we have that the following inequality holds*

$$\langle \mathbf{x}^{t+1} - \mathbf{x}, \mathbf{A}^\top \lambda^{t+1}\rangle \leq \langle \nabla f(\mathbf{x}^t, \boldsymbol{\xi}^t), \mathbf{x} - \mathbf{x}^{t+1}\rangle + \langle \mathbf{x} - \mathbf{x}^{t+1}, \rho\mathbf{A}^\top(\mathbf{B}\mathbf{z}^t - \mathbf{B}\mathbf{z}^{t+1})\rangle$$

$$+ \frac{1}{2\eta^t}(\|\mathbf{x} - \mathbf{x}^t\|_2^2 - \|\mathbf{x} - \mathbf{x}^{t+1}\|_2^2 - \|\mathbf{x}^t - \mathbf{x}^{t+1}\|_2^2).$$

Proof. By applying Lemma 4.1 with identifications $J_1(\mathbf{x}) = f(\mathbf{x}^t) + \langle \nabla f(\mathbf{x}^t, \boldsymbol{\xi}^t), \mathbf{x}\rangle$ and $J_2(\mathbf{x}) = \langle \boldsymbol{\lambda}^t \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z}^t - \mathbf{c}\rangle + \frac{\rho}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z}^t - \mathbf{c}\|_2^2 + \frac{\|\mathbf{x} - \mathbf{x^t}\|_2^2}{2\eta^t}$ in (4.21), we obtain

$$J_1(\mathbf{x}) - J_1(\mathbf{x}^{t+1}) + \langle \mathbf{x} - \mathbf{x}^{t+1}, \mathbf{A}^\top[\boldsymbol{\lambda}^t + \rho(\mathbf{A}\mathbf{x}^{t+1} + \mathbf{B}\mathbf{z}^t - \mathbf{c})]\rangle + \frac{1}{\eta^t}\langle \mathbf{x} - \mathbf{x}^{t+1}, \mathbf{x}^{t+1} - \mathbf{x}^t\rangle \geq 0. \tag{4.31}$$

We analyze each of three terms on the left hand side (LHS) of (4.31). The first term

$$J_1(\mathbf{x}) - J_1(\mathbf{x}^{t+1}) = \langle \nabla f(\mathbf{x}^t, \boldsymbol{\xi}^t), \mathbf{x} - \mathbf{x}^{t+1}\rangle, \tag{4.32}$$

and the second term

$$\langle \mathbf{x}-\mathbf{x}^{t+1}, \mathbf{A}^\top[\boldsymbol{\lambda}^t+\rho(\mathbf{A}\mathbf{x}^{t+1}+\mathbf{B}\mathbf{z}^t-\mathbf{c})]\rangle = \langle \mathbf{x}-\mathbf{x}^{t+1}, \mathbf{A}^\top\boldsymbol{\lambda}^{t+1}\rangle+\langle \mathbf{x}-\mathbf{x}^{t+1}, \rho\mathbf{A}^\top(\mathbf{B}\mathbf{z}^t-\mathbf{B}\mathbf{z}^{t+1})\rangle, \tag{4.33}$$

which holds since $\boldsymbol{\lambda}^{t+1} = \boldsymbol{\lambda}^t + \rho(\mathbf{A}\mathbf{x}^{t+1} + \mathbf{B}\mathbf{z}^{t+1} - \mathbf{c})$. The last term is

$$\frac{1}{\eta^t}\langle \mathbf{x} - \mathbf{x}^{t+1}, \mathbf{x}^{t+1} - \mathbf{x}^t\rangle = \frac{1}{2\eta^t}(\|\mathbf{x} - \mathbf{x}^t\|_2^2 - \|\mathbf{x} - \mathbf{x}^{t+1}\|_2^2 - \|\mathbf{x}^t - \mathbf{x}^{t+1}\|_2^2). \tag{4.34}$$

Substitute (4.32), (4.33) and (4.34) back to (4.31) proves the result. □

Theorem 4.4. *Assume* $\mathbb{E}\|\nabla f(\mathbf{x}, \boldsymbol{\xi})\|_2^2 \leq G^2, \forall \xi$. $\max[\frac{\rho}{2}\|\mathbf{A}\mathbf{x}+\mathbf{B}\mathbf{z}^0-\mathbf{c}\|_2^2+\frac{1}{2\rho}\|\boldsymbol{\lambda}-\boldsymbol{\lambda}^0\|_2^2] = L$ *and* $\max\|\mathbf{x}^t - \mathbf{x}\|_2^2 \leq D_x, \forall \mathbf{x} \in \mathcal{X}, \boldsymbol{\lambda} \in \mathbb{R}^p$*. Define* $\boldsymbol{\omega}^t = (\mathbf{x}^t; \mathbf{z}^t; \boldsymbol{\lambda}^t)$, $h(\boldsymbol{\omega}) = \mathbb{E}\{f(\mathbf{x}, \boldsymbol{\xi})\} + g(\mathbf{z})$ *and* $F(\boldsymbol{\omega}) = (\mathbf{A}^\top\boldsymbol{\lambda}; \mathbf{B}^\top\boldsymbol{\lambda}; -(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}))$*. Let* $\{\boldsymbol{\omega}^t\}$ *be the sequence generated by (4.21)-(4.23) and* $\eta^t = \frac{\epsilon}{\sqrt{t}}$*, where* $\epsilon \geq 0$*, and* $\bar{\boldsymbol{\omega}} = \frac{1}{T+1}\sum_{t=0}^T \boldsymbol{\omega}^t$*. Then for any integer number* $t > 0$ *and* $\boldsymbol{\omega} \in \mathcal{X} \times \mathcal{Z} \times \mathbb{R}^p$*,*

$$\mathbb{E}\{h(\bar{\boldsymbol{\omega}}^t) - h(\boldsymbol{\omega}) + \langle \bar{\boldsymbol{\omega}}^t - \boldsymbol{\omega}, F(\bar{\boldsymbol{\omega}}^t)\rangle\} \leq \mathcal{O}(\frac{1}{\sqrt{t}}). \tag{4.35}$$

Proof. By Lemma 4.3 and the convexity of function f, we obtain (4.36).

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{\xi}}\{f(\mathbf{x}^{t+1}, \boldsymbol{\xi})\} - \mathbb{E}_{\boldsymbol{\xi}}\{f(\mathbf{x}, \boldsymbol{\xi})\} + \langle \mathbf{x}^{t+1} - \mathbf{x}, \mathbf{A}^\top\lambda^{t+1}\rangle \\
&\leq \langle \nabla f(\mathbf{x}^t, \boldsymbol{\xi}^t), \mathbf{x} - \mathbf{x}^{t+1}\rangle - \langle \mathbb{E}_{\boldsymbol{\xi}}\{\nabla f(\mathbf{x}^{t+1}, \boldsymbol{\xi})\}, \mathbf{x} - \mathbf{x}^{t+1}\rangle + \langle \mathbf{x} - \mathbf{x}^{t+1}, \rho\mathbf{A}^\top(\mathbf{B}\mathbf{z}^t - \mathbf{B}\mathbf{z}^{t+1})\rangle \\
&+ \frac{1}{2\eta^t}(\|\mathbf{x} - \mathbf{x}^t\|_2^2 - \|\mathbf{x} - \mathbf{x}^{t+1}\|_2^2 - \|\mathbf{x}^t - \mathbf{x}^{t+1}\|_2^2) \\
&= \langle \nabla f(\mathbf{x}^t, \boldsymbol{\xi}^t) - \nabla f(\mathbf{x}^{t+1}), \mathbf{x} - \mathbf{x}^t\rangle + \langle \nabla f(\mathbf{x}^t, \boldsymbol{\xi}^t) - \nabla f(\mathbf{x}^{t+1}), \mathbf{x}^t - \mathbf{x}^{t+1}\rangle \\
&+ \langle \mathbf{x} - \mathbf{x}^{t+1}, \rho\mathbf{A}^\top(\mathbf{B}\mathbf{z}^t - \mathbf{B}\mathbf{z}^{t+1})\rangle + \frac{1}{2\eta^t}(\|\mathbf{x} - \mathbf{x}^t\|_2^2 - \|\mathbf{x} - \mathbf{x}^{t+1}\|_2^2 - \|\mathbf{x}^t - \mathbf{x}^{t+1}\|_2^2) \\
&\leq \langle \nabla f(\mathbf{x}^t, \boldsymbol{\xi}^t) - \nabla f(\mathbf{x}^{t+1}), \mathbf{x} - \mathbf{x}^t\rangle + \langle \mathbf{x} - \mathbf{x}^{t+1}, \rho\mathbf{A}^\top(\mathbf{B}\mathbf{z}^t - \mathbf{B}\mathbf{z}^{t+1})\rangle \\
&+ \frac{\eta^t}{2}\|\nabla f(\mathbf{x}^t, \boldsymbol{\xi}^t) - \nabla f(\mathbf{x}^{t+1})\|_2^2 + \frac{1}{2\eta^t}(\|\mathbf{x} - \mathbf{x}^t\|_2^2 - \|\mathbf{x} - \mathbf{x}^{t+1}\|_2^2).
\end{aligned} \tag{4.36}$$

Using definitions of $\boldsymbol{\omega}^t$, $h(\boldsymbol{\omega})$ and $F(\boldsymbol{\omega})$, adding (4.36) and (4.28) results in (4.37).

$$h(\mathbf{w}^{t+1}) - h(\mathbf{w}) + \langle \mathbf{w}^{t+1} - \mathbf{w}, F(\mathbf{w}^{t+1}) \rangle$$

$$\leq \langle \mathbf{x} - \mathbf{x}^{t+1}, \rho \mathbf{A}^\top (\mathbf{B}\mathbf{z}^t - \mathbf{B}\mathbf{z}^{t+1}) \rangle + \frac{1}{\rho} \langle \boldsymbol{\lambda} - \boldsymbol{\lambda}^{t+1}, \boldsymbol{\lambda}^{t+1} - \boldsymbol{\lambda}^t \rangle$$

$$+ \langle \nabla f(\mathbf{x}^t, \boldsymbol{\xi}^t) - \nabla f(\mathbf{x}^{t+1}), \mathbf{x} - \mathbf{x}^t \rangle + \frac{\eta^t}{2} \|\nabla f(\mathbf{x}^t, \boldsymbol{\xi}^t) - \nabla f(\mathbf{x}^{t+1})\|_2^2$$

$$+ \frac{1}{2\eta^t} (\|\mathbf{x} - \mathbf{x}^t\|_2^2 - \|\mathbf{x} - \mathbf{x}^{t+1}\|_2^2). \tag{4.37}$$

For the first term on the right hand side (RHS) of (4.37), we have

$$\langle \mathbf{x} - \mathbf{x}^{t+1}, \rho \mathbf{A}^\top (\mathbf{B}\mathbf{z}^t - \mathbf{B}\mathbf{z}^{t+1}) \rangle$$

$$= \frac{\rho}{2} [\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z}^t - \mathbf{c}\|_2^2 - \|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z}^{t+1} - \mathbf{c}\|_2^2$$

$$+ \|\mathbf{A}\mathbf{x}^{t+1} + \mathbf{B}\mathbf{z}^{t+1} - \mathbf{c}\|_2^2 - \|\mathbf{A}\mathbf{x}^{t+1} + \mathbf{B}\mathbf{z}^t - \mathbf{c}\|_2^2]$$

$$\leq \frac{\rho}{2} [\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z}^t - \mathbf{c}\|_2^2 - \|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z}^{t+1} - \mathbf{c}\|_2^2 + \frac{1}{\rho^2} \|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\|_2^2], \tag{4.38}$$

and the second term is

$$\frac{1}{\rho} \langle \boldsymbol{\lambda} - \boldsymbol{\lambda}^{t+1}, \boldsymbol{\lambda}^{t+1} - \boldsymbol{\lambda}^t \rangle = \frac{1}{2\rho} (\|\boldsymbol{\lambda} - \boldsymbol{\lambda}^t\|_2^2 - \|\boldsymbol{\lambda} - \boldsymbol{\lambda}^{t+1}\|_2^2 - \|\boldsymbol{\lambda}^t - \boldsymbol{\lambda}^{t+1}\|_2^2). \tag{4.39}$$

Let $\bar{\boldsymbol{\omega}} = \frac{1}{T+1} \sum_{t=0}^T \boldsymbol{\omega}^t$. Since $h(\bar{\mathbf{w}})$ is a convex function of $\bar{\boldsymbol{\omega}}$ and F is a monotonic operator, using (4.38) and (4.39) to rewrite (4.37) we have the relationship in (4.40).

$$h(\bar{\mathbf{w}}^{t+1}) - h(\bar{\mathbf{w}}) + \langle \bar{\mathbf{w}}^{t+1} - \mathbf{w}, F(\bar{\mathbf{w}}^{t+1}) \rangle$$

$$\leq \frac{1}{T+1} \sum_{t=0}^T [h(\mathbf{w}^{t+1}) - h(\mathbf{w}) + \langle \mathbf{w}^{t+1} - \mathbf{w}, F(\mathbf{w}^{t+1}) \rangle]$$

$$\leq \frac{1}{T+1} \sum_{t=0}^T \langle \nabla f(\mathbf{x}^t, \boldsymbol{\xi}^t) - \nabla f(\mathbf{x}^{t+1}), \mathbf{x} - \mathbf{x}^t \rangle + \frac{1}{T+1} \sum_{t=0}^T \frac{1}{2\eta^t} (\|\mathbf{x} - \mathbf{x}^t\|_2^2 - \|\mathbf{x} - \mathbf{x}^{t+1}\|_2^2)$$

$$+ \frac{1}{T+1} \sum_{t=0}^T \frac{\eta^t}{2} \|\nabla f(\mathbf{x}^t, \boldsymbol{\xi}^t) - \nabla f(\mathbf{x}^{t+1})\|_2^2 + \frac{1}{T+1} (\frac{\rho}{2} \|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y}^0 - \mathbf{c}\|_2^2 + \frac{1}{2\rho} \|\boldsymbol{\lambda} - \boldsymbol{\lambda}^0\|_2^2), \tag{4.40}$$

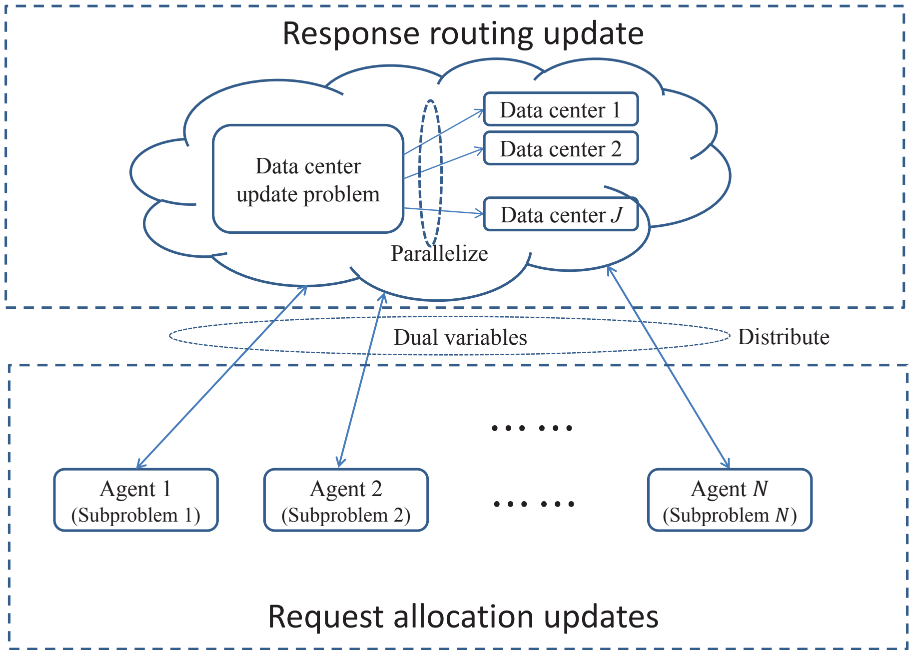


Figure 4.7 The original service revenue maximization problem is decoupled into two parts, response routing update and request allocation update. Two parts are coordinated through dual variables.

By assumptions that $\mathbb{E}\|\nabla f(\mathbf{x},\boldsymbol{\xi})\|_2^2 \leq G^2$, $\max[\frac{\rho}{2}\|\mathbf{A}\mathbf{x}+\mathbf{B}\mathbf{z}^0-\mathbf{c}\|_2^2+\frac{1}{2\rho}\|\boldsymbol{\lambda}-\boldsymbol{\lambda}^0\|_2^2] = L$, and $\max\|\mathbf{x}^t-\mathbf{x}\|_2^2 \leq D_x$ for all $\mathbf{x}\in\mathcal{X}, \boldsymbol{\lambda}\in\mathbb{R}^p$. Summing over t, we have

$$\mathbb{E}\{h(\bar{\mathbf{w}}^{t+1})-h(\bar{\mathbf{w}})+\langle\bar{\mathbf{w}}^{t+1}-\mathbf{w},F(\bar{\mathbf{w}}^{t+1})\rangle\}$$

$$\leq \frac{L}{T+1}+\frac{1}{T+1}\sum_{0}^{T}\frac{\eta^t}{2}G^2+\frac{1}{T+1}\frac{\sqrt{T+1}}{2\epsilon}D_x \leq \mathcal{O}(\frac{1}{\sqrt{t}}), \tag{4.41}$$

where the last step holds since $\mathbf{x}$ and $\boldsymbol{\xi}^t$ are independent and due to the fact that $\eta^t = \frac{\epsilon}{\sqrt{t}}$. $\square$

4.2.3.2 Distributed Scalable Design

We propose a distributed design to solve the optimization problem (4.13). Specifically, the decision variables $\mathbf{a}_i$ and $\mathbf{B}_j$ are arranged into two groups, which correspond to the mobile service agents request allocation and the data center response routing, respectively. During the optimization, the variables of each group are optimized in a distributed and parallel fashion. In particular, each mobile service agent i solves $\mathbf{a}_i$ and each data center j obtains $\mathbf{B}_j$, and those two groups of decision variables are coordinated through dual variables. The architecture of the proposed mechanism is illustrated in Fig. 4.7.

The problem (4.13) is not readily solved in a distributed fashion due to the coupling of decision variables $a_{i,j}$ across all mobile service agents in constraint (4.15) and the coupling of $a_{i,j}$ and $\mathbf{b}_{i,j}$ in constraint (4.17). To design a distributed approach for (4.13), we first rewrite the constraint (4.15) as

$$\sum_{i\in\mathcal{I}} \mathbf{b}_{j,i}^{\top}\mathbf{1} \leq C_j, \quad \forall j, \tag{4.42}$$

which is separable among data centers. We define sets $\mathcal{A}_i = \{\mathbf{a}_i | \sum_{j\in\mathcal{J}} a_{i,j} \leq R_i, a_{i,j} \geq 0, \forall j \in \mathcal{J}\}$ and $\mathcal{B}_j = \{\mathbf{B}_j | \sum_{i\in\mathcal{I}} \mathbf{b}_{j,i}^{\top}\mathbf{1} \leq C_j, \sum_{i\in\mathcal{I}} \mathbf{b}_{j,i} \preceq \mathbf{Q}_j, \mathbf{b}_{j,i} \succeq 0, \forall i \in \mathcal{I}\}$ for compactness. Accordingly, $\mathcal{A} = \{\bigcup \mathcal{A}_i\}_{i=1}^{I}$ and $\mathcal{B} = \{\bigcup \mathcal{B}_j\}_{j=1}^{J}$. Then the stochastic optimization problem (4.13) can be solved distributively in parallel using ADMM.

By applying the stochastic ADMM to solve the optimization problem (4.13), we first calculate the partial Lagrangian function, which introduces the Lagrange multipliers only for constraint (4.17):

$$\mathcal{L}_{\rho}(\{\mathbf{a}_i\}_{i=1}^{N}, \{\mathbf{B}_j\}_{j=1}^{J}, \{\mu_{i,j}\}_{i=1,j=1}^{N,J}) = \sum_{j\in\mathcal{J}} G_j(\mathbf{B}_j) - \alpha \sum_{i\in\mathcal{I}} \mathbb{E}_{\xi_i}\{F_i(\mathbf{a}_i, \xi_i)\}$$

$$+ \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} \langle \mu_{i,j}, a_{i,j} - \mathbf{b}_{j,i}^{\top}\mathbf{1}\rangle + \frac{\rho}{2}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} \|a_{i,j} - \mathbf{b}_{j,i}^{\top}\mathbf{1}\|_2^2, \tag{4.43}$$

where $\mu_{i,j}$ is the Lagrange multiplier. The decision variables $\mathbf{a}_i$ and $\mathbf{B}_j$ are arranged into two groups and updated iteratively. The update procedure has two major parts: the request allocation update and the response routing update, which are illustrated below.

Request allocation updates at mobile service agents: The request allocation updates at mobile service agents are performed by minimizing (4.43) with respect to $\{\mathbf{a}_i\}_{i=1}^{N} \in \mathcal{A}$. Specifically, at the t^{th} iteration, $\mathbf{a}_i$ is updated by

$$\{\mathbf{a}_i^{t+1}\}_{i=1}^{N} = \underset{\{\mathbf{a}_i\}_{i=1}^{N}\in\mathcal{A}}{\arg\min} -\alpha \sum_{i\in\mathcal{I}} \mathbb{E}_{\xi_i}\{F_i(\mathbf{a}_i, \xi_i)\} + \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} \langle \mu_{i,j}^{t}, a_{i,j} - \mathbf{1}^{\top}\mathbf{b}_{j,i}^{t}\rangle$$

$$+ \frac{\rho}{2}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} \|a_{i,j} - \mathbf{1}^{\top}\mathbf{b}_{j,i}^{t}\|_2^2. \tag{4.44}$$

In (4.44) the optimization is performed to maximize the utility at mobile service agents with regularization terms. The optimization problem (4.44) can be decomposed into N subproblems, and each subproblem is handled at a local computing unit. A computing unit can be a computing node in computer clusters, a CPU in a computing node, or even a core of a CPU. Each computing unit i solves a stochastic optimization problem as

$$\mathbf{a}_i^{t+1} = \arg\min_{\mathbf{a}_i \in \mathcal{A}_i} -\alpha \mathbb{E}_{\xi_i}\{F_i(\mathbf{a}_i, \xi_i)\} + \sum_{j\in\mathcal{J}} \langle \mu_{i,j}^t, a_{i,j} - \mathbf{1}^\top \mathbf{b}_{j,i}^t \rangle + \frac{\rho}{2}\sum_{j\in\mathcal{J}} \|a_{i,j} - \mathbf{1}^\top \mathbf{b}_{j,i}^t\|_2^2. \quad (4.45)$$

The optimization problem (4.45) can be solved by the following proposition.

Proposition 4.5. *The $\mathbf{a}_i$-update can be solved by the stochastic approximation (SA) approach as:*

$$\begin{aligned} \mathbf{a}_i^{t+1} = \arg\min_{\mathbf{a}_i \in \mathcal{A}_i} \quad & \sum_{j\in\mathcal{J}} \langle \mu_{i,j}^t, a_{i,j} - \mathbf{1}^\top \mathbf{b}_{j,i}^t \rangle + \frac{\rho}{2}\sum_{j\in\mathcal{J}} \|a_{i,j} - \mathbf{1}^\top \mathbf{b}_{j,i}^t\|_2^2 \\ & - \alpha \left(F_i(\mathbf{a}_i^t) + \langle \nabla F_i(\mathbf{a}_i^t, \xi_i^t), \mathbf{a}_i \rangle + \frac{\|\mathbf{a}_i - \mathbf{a}_i^t, \|_2^2}{2\eta^t} \right) \end{aligned} \quad (4.46)$$

where $\eta^t = \frac{\epsilon}{\sqrt{t+1}}$, and $\epsilon \geq 0$ is the penalty parameter.

Remark: In (4.46), the stochastic optimization (4.45) is solved by an SA approach, where a quadratic approximation of function $F(\mathbf{a}_i, \xi_i)$ at $\mathbf{a}_i^t$ is utilized. The computing unit corresponding to each mobile service agent will calculate its own request allocation decision $\mathbf{a}_i$ independently, by taking the stochastic wireless link latency into account.

Response routing update at data center: At the t^{th} iteration, the request allocation updates are performed by minimizing (4.43) with respect to $\{\mathbf{B}_j\}_{j=1}^J \in \mathcal{B}$ as

$$\{\mathbf{B}_j^{t+1}\}_{j=1}^J = \arg\min_{\{\mathbf{B}_j\}_{j=1}^J \in \mathcal{B}} \sum_{j\in\mathcal{J}} G_j(\mathbf{B}_j) + \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} \langle \mu_{i,j}^t, a_{i,j}^{t+1} - \mathbf{b}_{j,i}^\top \mathbf{1} \rangle + \frac{\rho}{2}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} \|a_{i,j}^{t+1} - \mathbf{b}_{j,i}^\top \mathbf{1}\|_2^2. \quad (4.47)$$

Problem (4.47) aims at minimizing the total cost of all data centers, which can be readily solved at each data center in parallel. Each data center j will determine response routing variables $\mathbf{b}_{j,i}$ independently by solving the following optimization problem

$$\mathbf{B}_j^{t+1} = \arg\min_{\mathbf{B_j} \in \mathcal{B}_j} G_j(\mathbf{B}_j) + \sum_{i\in\mathcal{I}} \langle \mu_{i,j}^t, a_{i,j}^{t+1} - \mathbf{b}_{j,i}^\top \mathbf{1} \rangle + \frac{\rho}{2}\sum_{i\in\mathcal{I}} \|a_{i,j}^{t+1} - \mathbf{b}_{j,i}^\top \mathbf{1}\|_2^2. \quad (4.48)$$

Finally, the dual variables are updated as

$$\mu_{i,j}^{t+1} = \mu_{i,j}^{t} + \rho(a_{i,j}^{t+1} - \mathbf{1}^\top \mathbf{b}_{j,i}^{t+1}). \tag{4.49}$$

Remark: In the response routing update, each data center j performs the optimization of (4.48) to find the $\mathbf{B}_j$, which minimizes the cost at each data center. After that, the dual variables $\{\mu_{i,j}\}_{i=1}^{N}$ for data center j are updated together at the computing unit corresponding to data center j. Each element $\mu_{i,j}$ can be interpreted as the 'price' of placing mobile agent i's service at data center j.

The large-scale stochastic optimization problem (4.13) for the service management can be done periodically on a designed cluster. At the beginning of each scheduling period, computing units corresponding to mobile agents perform request allocation updates to obtain $\mathbf{a}_i$. Then $\mathbf{a}_i$ is sent to computing units corresponding to data centers through internal network in the cluster. After the response routing update performed by data centers, $\mathbf{B}_j$ and $\mu_{i,j}^{t+1}$ are transmitted back to mobile agent computing units. The iterations are terminated once the revenue maximization problem is solved. The outputs are sent back to mobile agents and data centers for service allocation and response routing.

4.2.4 Numerical Results

The proposed algorithms are evaluated by numerical simulations from perspectives of computation performance and service management performance. The evaluation setup is introduced briefly, and then numerical results are presented.

4.2.4.1 Evaluation Settings

We consider a mobile cloud service which provides applications for N mobile service agents, $N \in \{100, 200, \ldots, 1000\}$. The service is deployed on 10 cloud enabled geographically distributed data centers, and each data center is multi-homed to 3 ISP links to deliver services to mobile clients. The capacity of each mobile agent is generated from a uniform distribution $\mathcal{U}(8000, 10000)$, with a mean of 9000 data units. The capacity of each data center is generated in a similar fashion such that

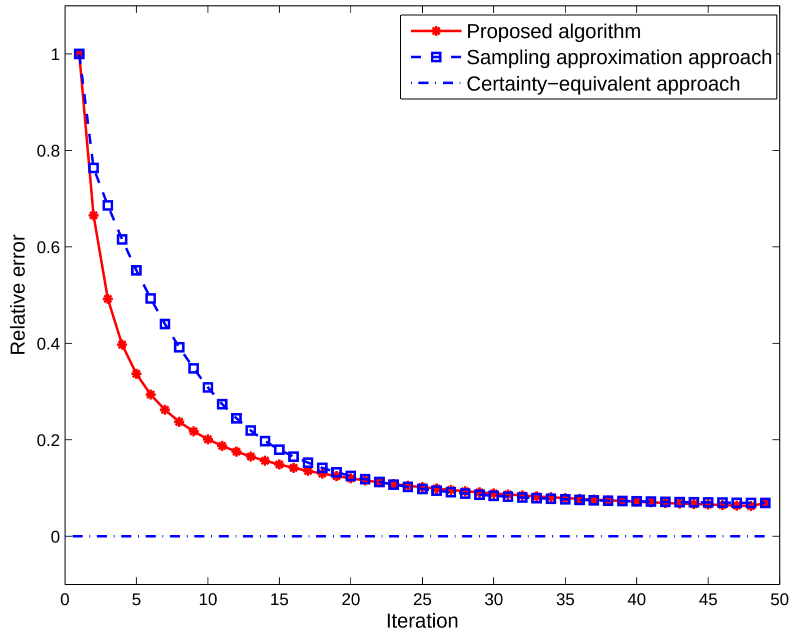


Figure 4.8 Convergence performance for 100 agents.

the total capacity of data centers is 1.5 times of mobile service agents total capacity. For simplicity, the latency of ISP links is randomly generated from $\mathcal{U}(25, 300)$ with a unit of milliseconds, and the stochastic wireless latency ξ_i is generated from an exponential distribution with a mean of 5 ms.

To describe the service cost at data centers, the 2011 annual average day-ahead on peak prices at 10 different local markets are used for data centers [103]. The server peak power and server idle power are set to $200W$ and $100W$, respectively. The power usage efficiently is 1.5. The prices of ISP links are chosen randomly from a finite set of $\{0.005, 0.01, 0.015\}$ monetary units per data unit.

4.2.4.2 Convergence Performance

The convergence performance of the proposed algorithm is shown in Fig. 4.8-Fig. 4.10. We compare the proposed algorithm with sampling approximation approach and certainty-equivalent approach. They are two state-of-the-art approaches for stochastic optimization. For the sampling approximation approach, the $\mathbb{E}_{\xi_i}\{F_i(\mathbf{a}_i, \xi_i)\}$ is approximated by $\frac{1}{Ns}\sum_{n=1}^{Ns}\{F_i(\mathbf{a}_i, \xi_i^n)\}$, where Ns is the number of samples of ξ_i from its distribution. The expectation of ξ_i, $\mathbb{E}\{\xi_i\}$, is used for

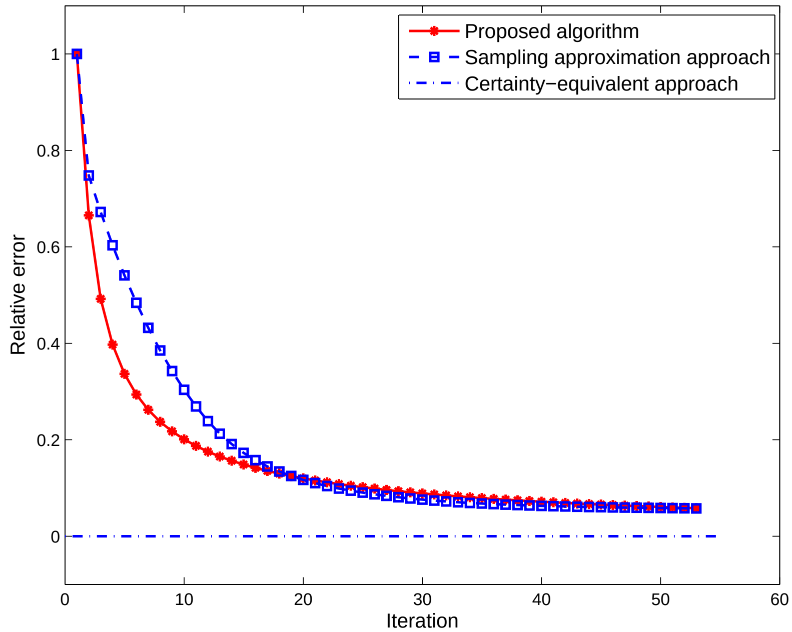


Figure 4.9 Convergence performance for 500 agents.

certainty-equivalent approach. Here, the scaled relative error is used to demonstrate results. Suppose that r^t is the value of the objective function at the t^{th} iteration, and r^* is the optimal solution of the certainty-equivalent approach. The scaled relative error e is defined as $e = |\frac{r^t - r^*}{r^0 - r^*}|$.

It is shown in Fig. 4.8-Fig 4.10 that the proposed algorithm converges for different number of mobile service agents. In Fig 4.8, when the number of mobile service agents is 100, the proposed algorithm takes a moderate number of iterations to converge. Furthermore, after very few iterations, the proposed algorithm yields close objective value to the sampling approximation approach, which demonstrates the effectiveness of proposed algorithm for solving the stochastic optimization problem. Similar performance can be found in Fig. 4.9 and Fig. 4.10, when the numbers of mobile service agents are 500 and 1000, respectively. As the number of mobile service agents increases, the proposed algorithm converges with only a small increment of iterations, which demonstrates the scalability of the proposed algorithm. Remark that the proposed method only utilize one realization of ξ at each iteration to solve the stochastic optimization problem. In the SAA method, Ns samples are used to approximate $\mathbb{E}_{\xi_i}\{F_i(\mathbf{a}_i, \xi_i)\}$ at each iteration. Hence the proposed method significantly

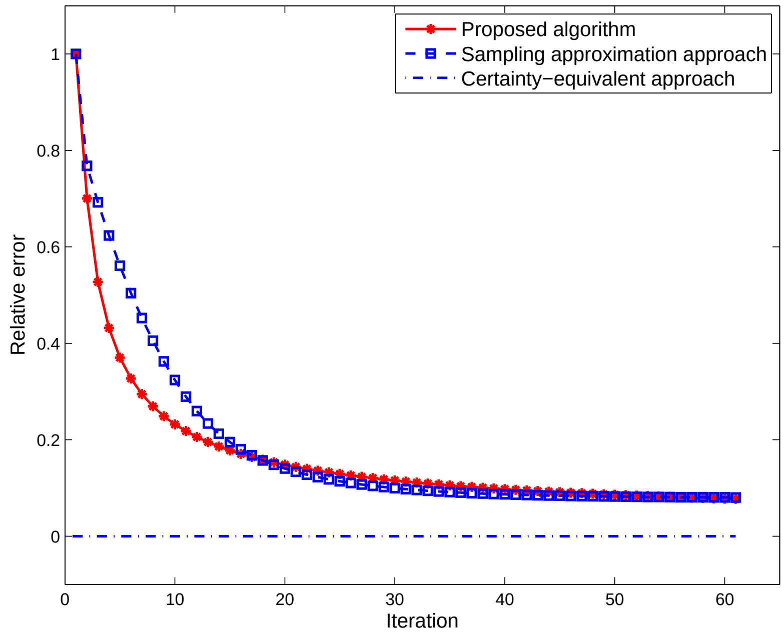


Figure 4.10 Convergence performance for 1000 agents.

reduce the computational time.

In Fig. 4.11, we show the relative error of the proposed algorithm for different values of penalty parameter ρ. It is observed that the proposed algorithm converges in a moderate number of iterations for all values of ρ between 0.01 and 1. We choose $\rho = 0.1$ in numerical simulations. An inappropriate choice of ρ will result in oscillating objective value and slow convergence rate.

4.2.4.3 Service Management Performance

In the following we show the effectiveness of the proposed algorithm on service management. We compare the proposed algorithm with two service management approaches. One the ‘cheapest selection’ which aims at minimizing the data center cost solely, and the other is the ‘minimum latency selection’ which aims at minimizing the ISP link latency solely. We specify the number of mobile service agents to 100, and the performance comparisons are shown in Fig. 4.12 and Fig. 4.13.

The cumulative density function (CDF) of the request latency for three mechanisms is shown in Fig. 4.12. It is observed that 90% of requests are served with latency less than 100ms for the pro-

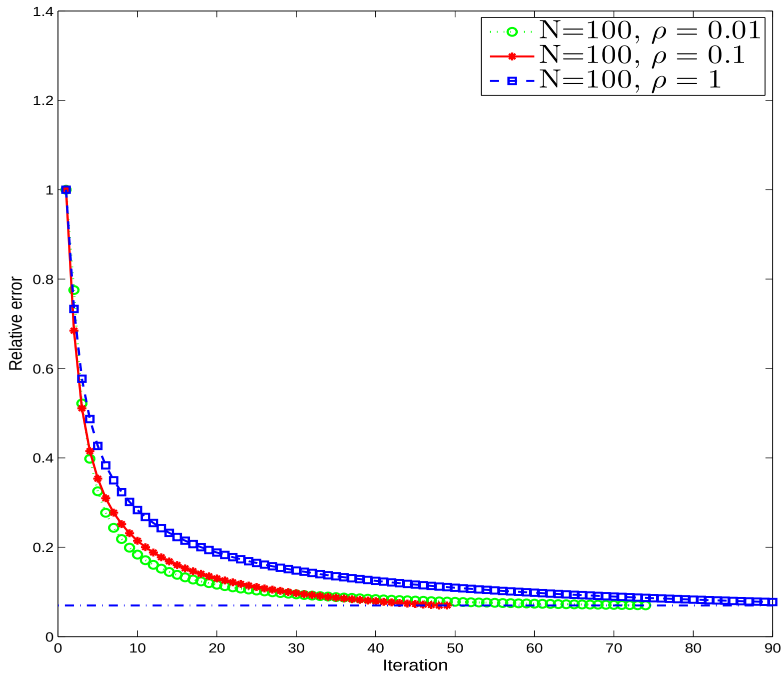


Figure 4.11 Rate of convergence of proposed algorithm for different values of penalty parameter ρ.

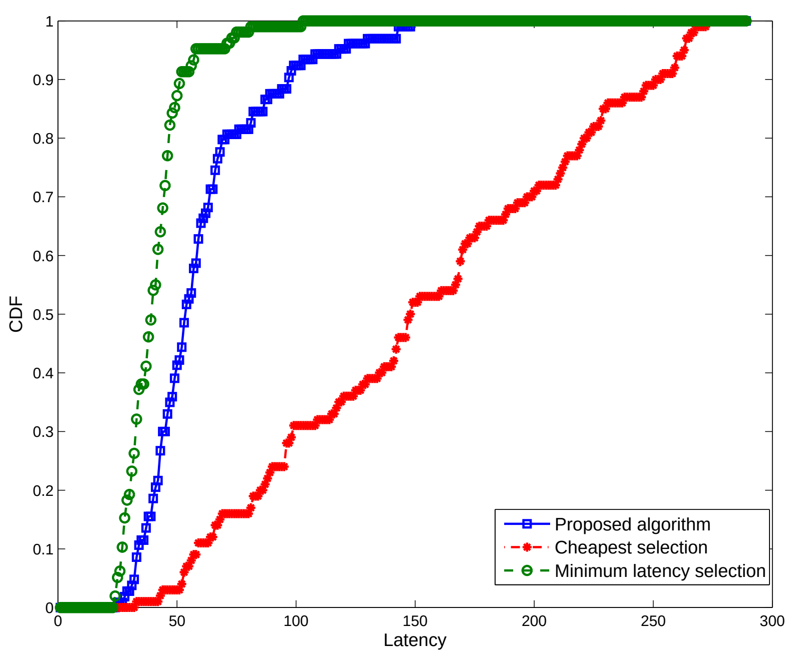
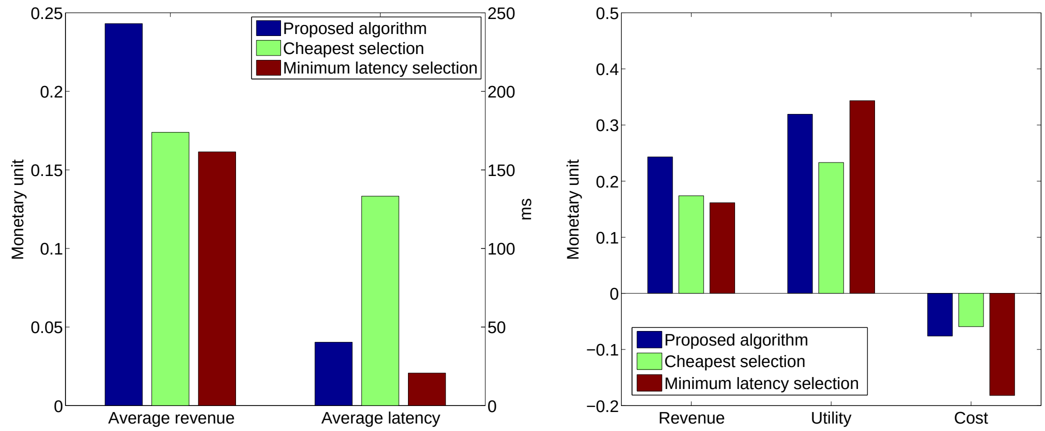


Figure 4.12 The CDF of the latency for three service management approaches.



(a) Average latency and revenue

(b) Average revenue, utility and cost

Figure 4.13 Comparisons of average latency, revenue, utility and cost for three service management approaches.

posed algorithm, and the latency performance of the proposed algorithm is close to the ‘minimum latency selection’ approach. The ‘cheapest selection’ dose not take the latency performance into consideration explicitly, and thus has the worst performance. For completeness, the comparisons of average latency, revenue, utility, and cost for three service management approaches are shown in Fig. 4.13. It is shown in Fig. 4.13(a) that the proposed algorithm outperforms other two from perspectives of both average revenue and latency. A detail analysis of the average revenue is shown in Fig. 4.13(b). It is shown that the proposed algorithm chooses the data center neither conservatively to reduce the cost, like the ‘cheapest selection’ approach, nor aggressively to grasp the utility, like the ‘minimum latency selection’. It manages the mobile cloud service strategically to balance utility and cost.

Next we compare the revenue of the proposed service management algorithm with the mechanism without the consideration of wireless latency, i.e., without the consideration of ξ_i in (4.13). The number of agents is varied from 100 to 1000, and the comparison of revenue is shown in Fig. 4.14. The evaluations are performed 50 times at each number of agents. The box plot of the revenue gain at different number of agents and the plot of the mean of revenue gain are shown in Fig. 4.14. The box plot depicts groups of numerical data through their quartiles. It is shown in Fig. 4.14 that when

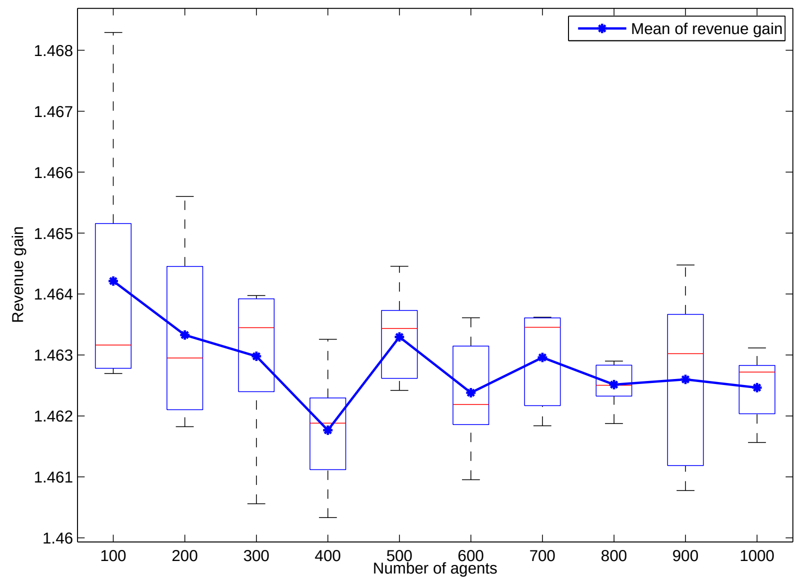


Figure 4.14 Comparisons of revenue for service management with/without consideration of wireless latency at different number of agents.

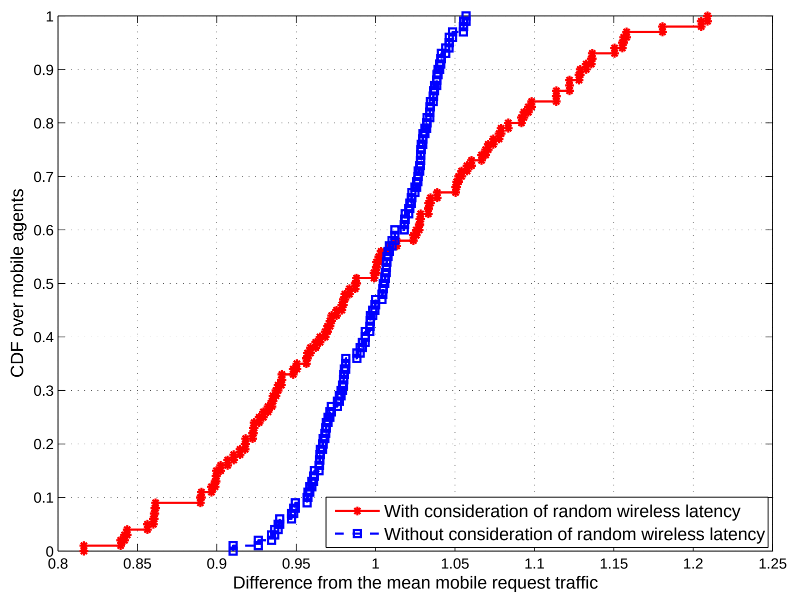


Figure 4.15 The CDF of traffic variations among all mobile service agents with/without consideration of wireless latency.

taking the random wireless latency into consideration, the proposed service management algorithm achieves a revenue gain of 46% at different numbers of mobile service agents, compared with that without the consideration of wireless latency. Moreover, such kind of improvement is relatively stable across different numbers of mobile agents.

To further understand the effect of wireless latency on mobile service management, we investigate traffic variations among all mobile service agents. It is shown in Fig. 4.15 that the CDF with consideration of random wireless nature is more skewed, implying that the request traffic varies significantly across mobile clients. By taking wireless latency into consideration, mobile service agents can adaptively admit request traffic according to the wireless link condition.

4.3 Conclusion

In this chapter, we have investigated distributed approaches for the mobile data offloading in a SDN and the service management for mobile cloud computing. We have shown that efficient and scalable management of big data services and data traffic can improve resource utilization and service quality of the could computing.

Chapter 5

Interdisciplinary Studies

This chapter presents interdisciplinary studies of big data optimization methods. We first described a decentralized approach of the Gauss-Newton (GN) method for nonlinear least squares (NLLS) on a wide area network (WAN). In a multi-agent system, a centralized GN for NLLS requires the global GN Hessian matrix available at a central computing unit, which may incur large communication overhead. In the proposed decentralized alternative, each agent only needs local GN Hessian matrix to update iterates with the cooperation of neighbors. For the hyperspectral imaging, we proposed a novel imaging method to identify substances in the scene of interest. In particular, instead of point-by-point scanning of the whole scene, a part of the scene is acquired through coded measurements of spatial and spectral information. Given spectral signatures of substances, the original data cube containing spatial and spectral information can be correctly reconstructed by the l_1 optimization method. The total variation is then performed to recovered the whole scene. Numerical results are provided to validate the performance of proposed methods.

The remaining of this chapter is organized as follows. Section 5.1 describes the decentralized nonlinear least squares on wide area networks. The compressive hyperspectral imaging is illustrated in Section 5.2.2. Finally, Section 5.3 gives a short conclusion.

5.1 Decentralized Nonlinear Least Squares

The significant importance of nonlinear least squares (NLLS) in applications of state estimation in power system [116], signal detection in wireless networks [117], and target tracking in mobile networks [118] have been appreciated for decades. The Gauss-Newton (GN) method, which can be seen as a modification of the Newton's method, is widely used to solve the NLLS [28]. The GN method finds the minimizer of the NLLS in an iterative fashion, and obtains the solution with provable local optimality and convergence rate. In this work, a decentralized GN method for NLLS on a WAN is presented. In particular, only local GN Hessian matrix is used and limited communication is performed between neighboring agents. The decentralized optimization enjoys the advantage of scalability to network size, robustness to dynamic topologies, and privacy preservation in data-sensitive applications [115, 119–121]. A detailed formulation of the decentralized optimization problem for NLLS on a WAN is provided, and the updating rule at each agent is explicitly given. We also investigate the convergence property of the proposed algorithm, which turns out the convergence rate is related to the number of agents as well as the minimum node degree in the network. Numerical tests validate the performance of the proposed algorithm.

The contributions of this work are threefold. Firstly, we do not assume any specific structure for the global Hessian matrix, and proposed a decentralized GN method for NLLS use only local Hessian matrix. Whereas the localization application in [118] has a block-wise Jacobian matrix which is convenient to decompose, and needs the global Hessian matrix for network-wide consensus. [116] proposes a generalized gossip-based GN method, which still requires the global Hessian matrix through Gossip exchange. Secondly, we proved the local superlinearly convergence property of the proposed algorithm. Finally, we validated the proposed method through numerical simulations.

5.1.1 The Nonlinear Least Squares Problem

Consider an unknown variable $\tilde{\mathbf{x}} \in \mathbb{R}^n$ in a network, and m observations are obtained through a vector-valued function $\mathbf{h}(\tilde{\mathbf{x}}) = (h_1(\tilde{\mathbf{x}}), \ldots, h_m(\tilde{\mathbf{x}})) : \mathbb{R}^n \to \mathbb{R}^m$. Each entry in function $\mathbf{h}(\tilde{\mathbf{x}})$ is a real value function and not necessarily convex. Let $\mathbf{z} \in \mathbb{R}^m$ denotes the observations as $\mathbf{z} = \mathbf{h}(\tilde{\mathbf{x}}) + \mathbf{e}$, where $\mathbf{e}$ stands for measurement errors. The covariance of $\mathbf{e}$ is $\mathbf{R} \in \mathbb{R}^{m \times m}$. The unknown variable $\tilde{\mathbf{x}} \in \mathbb{R}^n$ can be estimated by the NLLS as

$$\underset{\tilde{\mathbf{x}}}{\text{minimize}} \quad (\mathbf{z} - \mathbf{h}(\tilde{\mathbf{x}}))^\top \mathbf{R}^{-1} (\mathbf{z} - \mathbf{h}(\tilde{\mathbf{x}})). \tag{5.1}$$

The GN method can be adopted to solve (5.1) given that all observations and functions are available at a central computing node. Specifically, define $\mathbf{r}(\tilde{\mathbf{x}}) = \mathbf{R}^{-1/2}(\mathbf{z} - \mathbf{h}(\tilde{\mathbf{x}}))$ and its Jacobian $\mathbf{J}(\tilde{\mathbf{x}}) = \partial \mathbf{r}(\tilde{\mathbf{x}}) / \partial \tilde{\mathbf{x}}$. Let $F(\tilde{\mathbf{x}}) = \|r(\tilde{\mathbf{x}})\|^2$. Problem (5.1) can be solved iteratively as $\tilde{\mathbf{x}}^{k+1} = \tilde{\mathbf{x}}^k + \alpha^k \mathbf{d}^k$, where the descent direction $\mathbf{d}^k$ at each iteration can be obtained by solving

$$\mathbf{J}^\top(\tilde{\mathbf{x}}^k)\mathbf{J}(\tilde{\mathbf{x}}^k)\mathbf{d} = \mathbf{J}^\top(\tilde{\mathbf{x}}^k)\mathbf{r}(\tilde{\mathbf{x}}^k). \tag{5.2}$$

The GN method can solve the problem (5.1) at a superlinear convergence rate with order at least two under Assumption 1. Note that majority of NLLS problems are non-convex, and in this section we only consider the local convergence property of the algorithm. We assume Assumption 1 holds throughout this section.

Assumption 1. *Consider a function $F(\tilde{\mathbf{x}})$, suppose following assumptions hold.*

1. *The function $F(\tilde{\mathbf{x}})$ is continuous, differentiable and bounded below.*
2. *There exists a vector $\tilde{\mathbf{x}}^*$ such that the greatest lower bound can be achieved.*
3. *For $\delta > 0$, let S_δ denote the sphere $\{\tilde{\mathbf{x}} | \|\tilde{\mathbf{x}} - \tilde{\mathbf{x}}^*\|^2 \leq \delta\}$. The Hessian matrix $\mathbf{J}^\top(\tilde{\mathbf{x}})\mathbf{J}(\tilde{\mathbf{x}})$ is invertible in the sphere S_δ. For some $L > 0$, $M > 0$, $\delta > 0$, and for all $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$ in S_δ, we have $\|\mathbf{J}^\top(\tilde{\mathbf{x}})\mathbf{J}(\tilde{\mathbf{x}}) - \mathbf{J}^\top(\tilde{\mathbf{y}})\mathbf{J}(\tilde{\mathbf{y}})\| \leq L\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|$ and $\sigma_{\min}(\mathbf{J}^\top(\tilde{\mathbf{x}})\mathbf{J}(\tilde{\mathbf{x}})) \geq \frac{1}{M} \geq 0$.*

In a multi-agent system consisted of N networked agents, each agent is engaged in its own monitoring and controlling task in the network. At the same time, each agent is cooperating with other agents in the context of estimating global system states $\tilde{\mathbf{x}}$. Suppose these N agents are loosely coupled; there is very little, if any, central coordination and control among those agents, and each agent is able to exchange information with its neighbors. The system states $\tilde{\mathbf{x}}$ can be obtained by solving the following optimization problem

$$\begin{aligned} \underset{\mathbf{x}_i \ldots \mathbf{x}_N}{\text{minimize}} \quad & f(\mathbf{x}) = \sum_{i=1}^{N} (\mathbf{z}_i - \mathbf{h}_i(\mathbf{x}_i))^\top \mathbf{R}_i^{-1} (\mathbf{z}_i - \mathbf{h}_i(\mathbf{x}_i)), \\ \text{subject to} \quad & \mathbf{x}_i = \ldots = \mathbf{x}_N, \end{aligned} \tag{5.3}$$

where $\mathbf{z}_i$ is the local observation which is a subset of $\mathbf{z}$, i.e., $\mathbf{z} = (\mathbf{z}_1; \ldots; \mathbf{z}_N)$, and $\mathbf{h}_i$ is the local observation function which is a subset of $\mathbf{h}$, i.e., $\mathbf{h} = (\mathbf{h}_1; \ldots; \mathbf{h}_N)$. $\mathbf{R}_i^{-1}$ is the covariance matrix of local noise vector $\mathbf{e}_i$. $\mathbf{x}_i \in \mathbb{R}^n$ is the local duplicate of $\tilde{\mathbf{x}}$, and $\mathbf{x} = (\mathbf{x}_1; \ldots; \mathbf{x}_N) \in \mathbb{R}^{Nn}$.

5.1.2 Decentralized Nonlinear Least Squares on Wide Area Network

For concreteness, the network model of agents is first described. Specifically, consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. $\mathcal{V}$ represents the set of agents, and $\mathcal{E}$ represents the set of communication links between each pair of agents. An arc e is associated with an order pair (i, j) as $e \sim (i, j)$, which means the information is transmitted from agent i to agent j. Assume the graph formed by the agents is connected. By introducing auxiliary variables $\mathbf{w}_{ij} \in \mathbb{R}^n$ associated with each arc $e \sim (i, j) \in \mathcal{E}$, problem (5.3) can be reformulated as

$$\underset{\{\mathbf{x}_1,\ldots,\mathbf{x}_N\}}{\text{minimize}} \quad \sum_{i=1}^{N} (\mathbf{z}_i - \mathbf{h}_i(\mathbf{x}_i))^\top \mathbf{R}_i^{-1} (\mathbf{z}_i - \mathbf{h}_i(\mathbf{x}_i)) \text{ and} \tag{5.4}$$

$$\text{subject to} \quad \mathbf{x}_i = \mathbf{w}_{ij}, \quad \mathbf{x}_j = \mathbf{w}_{ij}, \quad \forall (i, j) \in \mathcal{E}, \tag{5.5}$$

where $\mathbf{w}_{ij}$ is used to enforce the equality of variables $\mathbf{x}_i$ and $\mathbf{x}_j$ for agents i and j connected by arc (i, j). We use compact notations in the following for the sake of discussion simplicity. Concatenating $\mathbf{w}_{ij}$ in vector $\mathbf{w}$, problem (5.4) can be reformulated as

$$\underset{\mathbf{x},\mathbf{w}}{\text{minimize}} \quad f(\mathbf{x}), \quad \text{subject to} \quad \mathbf{A}_s\mathbf{x} - \mathbf{w} = 0, \quad \mathbf{A}_d\mathbf{x} - \mathbf{w} = 0, \tag{5.6}$$

where $\mathbf{A}_s$ and $\mathbf{A}_d$ are extended arc source matrix and extended arc destination matrix for the network graph $\mathcal{G}$, respectively. Stacking $\mathbf{A}_s$ and $\mathbf{A}_d$ to form $\mathbf{A} = [\mathbf{A}_s; \mathbf{A}_d] \in \mathbb{R}^{2Mn \times Nn}$. The optimization problem (5.6) reduces to

$$\underset{\mathbf{x},\mathbf{w}}{\text{minimize}} \quad f(\mathbf{x}), \quad \text{subject to} \quad \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{w} = 0, \tag{5.7}$$

where $\mathbf{B} = [-\mathbf{I}_{Mn}; -\mathbf{I}_{Mn}] \in \mathbb{R}^{2Mn \times Mn}$. The GN method is utilized to solve the optimization problem (5.7), where updates of $\mathbf{x}$ are implemented in a decentralized fashion. The local update rule at each agent is given in the following proposition.

Proposition 5.1. *Consider iterates $\mathbf{x}^k$ and $\mathbf{z}^k$ with the initialization $\mathbf{E}_u\mathbf{x}^0 = 2\mathbf{w}^0$, the iterates $\mathbf{x}_i^k$ at each agent i can be iteratively generated by following recursions for $k > 0$:*

$$\mathbf{x}_i^{k+1} = \mathbf{x}_i^k + \alpha_i^k \mathbf{d}_i^k, \tag{5.8}$$

where α_i^k is a positive constant, and $\mathbf{d}_i^k$ is the descent direction which can be determined as

$$(\mathbf{J}_i^\top(\mathbf{x}_i^k)\mathbf{J}_i(\mathbf{x}_i^k) + \rho\nu_i\mathbf{I}_n)\mathbf{d}_i^k = \mathbf{J}_i^\top(\mathbf{x}_i^k)\mathbf{r}_i(\mathbf{x}_i^k) + \rho[\nu_i\mathbf{x}_i - \frac{1}{2}\sum_{j\in\mathcal{N}_i}(\mathbf{x}_i^{k-1} + \mathbf{x}_j^{k-1})],$$

where $\mathbf{r}_i(\mathbf{x}_i) = \mathbf{R}_i^{-1/2}(\mathbf{z}_i - \mathbf{h}_i(\mathbf{x}_i))$ and $\mathbf{J}_i(\mathbf{x}_i)$ is its Jacobian $\mathbf{J}_i(\mathbf{x}_i) = \partial\mathbf{r}_i(\mathbf{x}_i)/\partial\mathbf{x}_i$. ν_i is the degree of agent i in the network, and $\mathcal{N}_i$ denotes the neighbors of agent i.

The local convergence property and convergence rate of the proposed decentralized approach are given by the following theorem.

Theorem 5.2. *Suppose the Assumption 1 holds, and the start point of each agent $\mathbf{x}_i^0$ is in S_δ. The sequence $\{\mathbf{x}^k\}$ generated by the update rule given in Proposition 5.1 is defined, and converges to $\mathbf{x}^* = \{\tilde{\mathbf{x}}^*; \ldots; \tilde{\mathbf{x}}^*\}$. Furthermore, we have*

$$\|\mathbf{x}^{k+1} - \mathbf{x}^*\| \leq \frac{ML\sqrt{N}}{2(1 + M\rho\, max(\nu_i))}\|\mathbf{x}^k - \mathbf{x}^*\|^2, \tag{5.9}$$

where N is the number of agents and $\max(\nu_i)$ is the maximum node degree in the network.

Remark that Theorem. 5.2 illustrates the local convergence property of the proposed decentralized approach, which converges to the optimal solution superlinearly. The convergence rate is related to the number of agents as well as the minimum node degree in the network.

5.1.3 Numerical Results

A bidirectionally connected ring network composed of $N = 100$ agents is considered here, in which each agent connects to exactly two agents. The unknown system states in the network is $\tilde{\mathbf{x}} \in \mathbb{R}^3$. The observation function $\mathbf{h}_i(\mathbf{x}_i)$ at each agent i is defined as

$$\mathbf{h}_i(\mathbf{x}_i) = a_i(\mathbf{x}_i(1)^2 + \mathbf{x}_i(2)^2) + b_i\mathbf{x}_i(2)\sin(\mathbf{x}_i(2) - \mathbf{x}_i(3)) + c_i\mathbf{x}_i(1)\mathbf{x}_i(2), \tag{5.10}$$

where a_i, b_i, and c_i are i.i.d. random variables follow the standard normal distribution. It is seen that the observation function $\mathbf{h}_i(\mathbf{x}_i)$ is a nonlinear function with a quadratic term, a trigonometric term and a cross product term. The agents in the network work cooperatively to estimate unknown system states $\tilde{\mathbf{x}}$ in a decentralized fashion. The convergence result is depicted in Fig. 5.1. It is

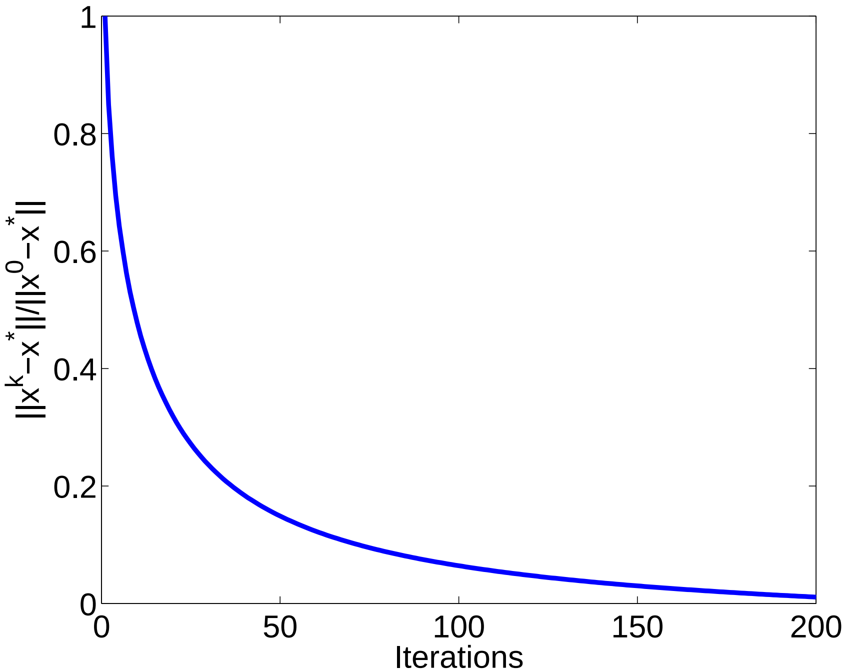


Figure 5.1 Convergence performance of the proposed algorithm.

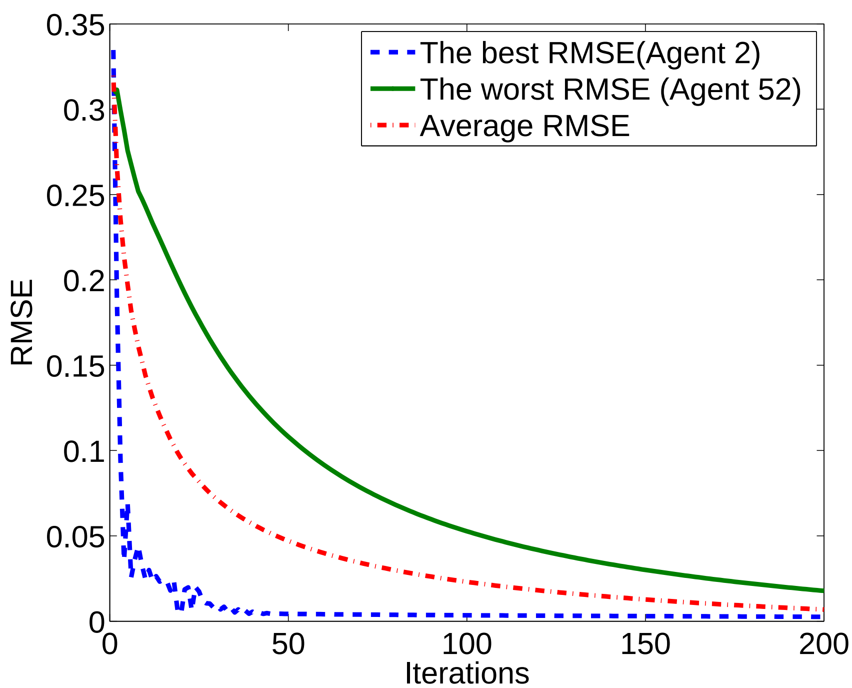


Figure 5.2 RMSE performance of the proposed algorithm.

shown that the proposed algorithm is effective in the sense that after a moderate number of iterations, the iterates converge to the optimal value. To investigate the performance of the proposed decentralized approach at each agent, the root-mean-square error (RMSE) of the estimate at each agent is calculated. The best RMSE (agent 2), the worst RMSE(agent 52) and the average RMSE are described in Fig. 5.2. It can be seen that at each agent, the RMSE decreases as the iteration increases. Furthermore, the convergence rates at each agent are different.

5.2 Compressive Hyperspectral Imaging

5.2.1 Optical Imaging Model

The proposed imaging system is comprised of a telescopic system for scene, and a dispersive system commonly used as a traditional dispersive spectroscope. A spatial light modulator occupies the plane between these two systems, which modulates the spatial information over all wavelengthes with the programmed pattern. The spectral intensity is captured by the focal plane array after dispersion. A schematic of the proposed imaging system is shown in Fig. 5.3.

The thermal emission from the region of interest, $f_s(x, y; \lambda)$, is first demagnified and imaged to the object plane of the telescopic system, $f_0(x, y; \lambda)$. (x, y) is the spatial coordinate and λ represents the wavelength. After passing through the spatial light modulator, the resulting field is expressed as $T(x, y)f_0(x, y; \lambda)$. $f_0(x, y; \lambda)$ is the spectral density of the scene, and $T(x, y)$ is the binary or gray-scale reflection function

$$T(x, y) = \sum_{m',n'} \alpha_{m',n'} \text{rect}(\frac{x}{\beta\Delta} - m', \frac{y}{\beta\Delta} - n'), \tag{5.11}$$

where $\alpha_{m',n'}$ is $0/1$ according to the configuration of the modulator at (m', n') and Δ is the size of detector pixel. The feature size of the spatial light modulator can be an integer multiple β of Δ. The choice of spatial light modulator here is a digital micro-mirrors device(DMD). Each mirror rotates about a hinge and can be positioned in one of two orientations, $+12$ degrees and -12 degrees from horizontal. The light falling on the DMD can be reflected in two directions depending on

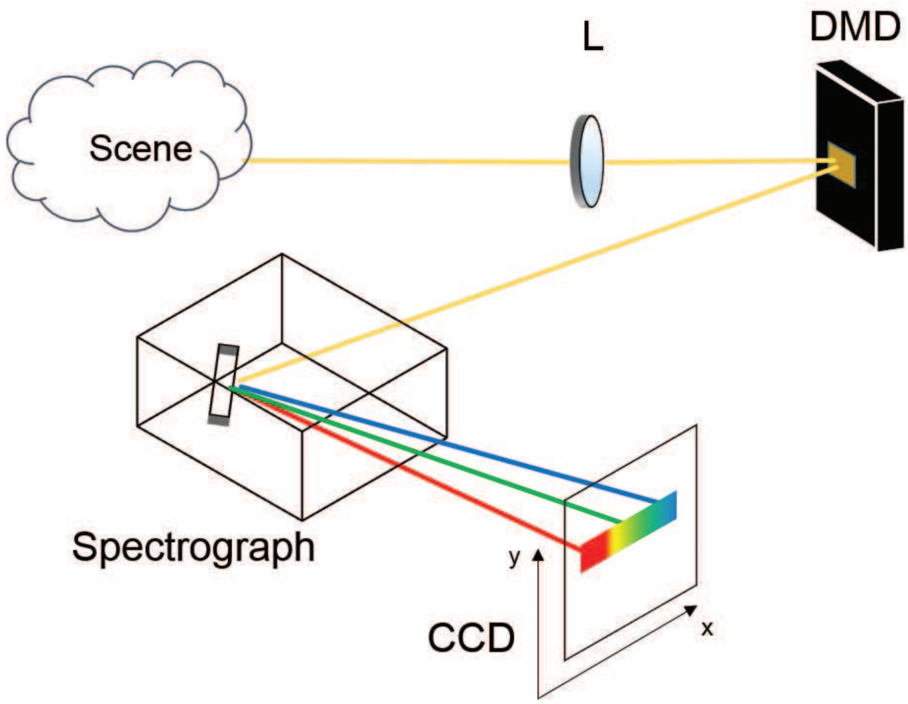


Figure 5.3 The schematic of imaging system.

orientations of mirrors, which correspond to 0 and 1, respectively.

The resulting field, $T(x,y)f_0(x,y;\lambda)$, is then imaged by the dispersive system with a grating placed at its Fourier plane. The field at the detector plane can be written as the convolution of the point spread function of the spectrograph and $T(x,y)f_0(x,y;\lambda)$

$$\begin{aligned} f(x,y;\lambda) = &\iint \delta(x' - (x + \gamma(\lambda - \lambda_c)))\delta(y' - y) \\ &\times T(x',y')f_0(x',y';\lambda)\,dx'dy' \\ &= f_0(x + \gamma(\lambda - \lambda_c), y; \lambda)T(x + \gamma(\lambda - \lambda_c), y), \end{aligned} \tag{5.12}$$

where γ is the linear dispersion of the dispersive element, and λ_c is the center wavelength of interest. The field received at the detector array contains a modulated mixture of spatial and spectral information about the scene, and the spatial shift of every spectral band happens only in the dispersion direction. Since the detector array's sensitivity covers the wavelength range of $7 - 14\mu m$, the field received can be expressed as a integration over the entire wavelength

$$g(x,y) = \int f(x,y;\lambda)\, d\lambda. \tag{5.13}$$

Equation (5.13) serves as the general imaging model in the continuous spatial domain. Recognizing that both mask and detector arrays are in fact pixilated g(x,y)

$$\begin{aligned}
g_{mn} &= \iint g(x,y)\text{rect}(\frac{x}{\Delta} - m, \frac{y}{\Delta} - n)\, dxdy + w_{mn} \\
&= \iiint f(x,y;\lambda)\text{rect}(\frac{x}{\Delta} - m, \frac{y}{\Delta} - n)\, dxdyd\lambda + w_{mn} \\
&= \sum_{m',n'} \alpha_{m',n'} \iiint f_0(x+\gamma(\lambda-\lambda_c), y; \lambda) \times \text{rect}(\frac{x+\gamma(\lambda-\lambda_c)}{\beta\Delta} - m', \frac{y}{\beta\Delta} - n') \\
&\quad \times \text{rect}(\frac{x}{\Delta} - m, \frac{y}{\Delta} - n)\, dxdyd\lambda + w_{mn}.
\end{aligned} \tag{5.14}$$

Remark that both the spatial light modulator and the detector array are in fact pixilated, we can discretize the modulated data as a three dimension data cube $(N \times M \times K)$. M and N are numbers of spatial channels, and K is the number of spectral channels of the thermal emission. The sampling measurements g on the detector can be represented as a $N \times (M + K - 1)$ matrix. Spectral channels of original image's adjacent columns has a column of pixels displacement due to the dispersive system.

5.2.2 Compressive Hyperspectral Imaging Methods

The three dimension data cube representing the scene, $f_0(x,y;\lambda)$, can be expressed as

$$f_0(x,y;\lambda) = \sum_{x,y} \mathbf{\Phi}\mathbf{v}(x,y), \tag{5.15}$$

where $\mathbf{\Phi} = (\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, ..., \boldsymbol{\phi}_J)$ is the emissivity spectrum of J different substances. $\mathbf{v}(x,y)$ is $J \times 1$ vector from the set $\mathcal{V} = \{(1,0,...,0),(0,1,...,0),...,(0,0,...,1)\}$. Note that $\mathbf{v}(x,y)$ is highly sparse in the sense that at most one component is non-zero. It means that for each pixel location, its emissivity spectrum can be uniquely determined by selecting the appropriate spectrum in $\mathbf{\Phi}$.

We can write the resulting field passed the spatial light modulator in a similar way

$$f(x,y;\lambda) = \sum_{(x,y)\in\mathcal{S}} \mathbf{\Phi}\mathbf{v}(x,y), \tag{5.16}$$

where $(x,y) \in \mathcal{S} = \{(x,y) : T(x,y) \neq 0\}$. We reformulate equation (5.16) into a matrix form by stacking $\mathbf{v}(x,y)$ into a vector $\mathbf{u}$ as

$$\mathbf{g} = \mathbf{H}\mathbf{\Phi}\mathbf{u} + \mathbf{w}, \tag{5.17}$$

where $\mathbf{u}$ is a $(M \times N \times J) \times 1$ vector and g is a $(K + M - 1) \times N \times 1$ vector. $\mathbf{H\Phi}$ is a matrix of size $((K + M - 1) \times N) \times (M \times N \times J)$, where H is a linear operator that represents the effect of the imaging system.

By exploiting the sparsity of $\mathbf{u}$, the recovery of $\mathbf{g}$ in problem (5.17) can be formulated as

$$\underset{\mathbf{u}}{\text{minimize}}\, ||\mathbf{g} - \mathbf{H\Phi u}||_2^2 + \lambda||\mathbf{u}||_1, \tag{5.18}$$

where λ is a parameter balancing the data fidelity term and the regularization term. (5.18) is a convex problem and can be solved by existing polynomial time algorithms. What's more, we can deal with a nonnegative l_1 minimization problem that will enhance sparsity. Note that the recovered $\mathbf{u}$ is the partial image here. The whole image can be recovered by the total variation imprinting optimization as

$$\underset{\hat{\mathbf{v}}}{\text{minimize}}\, \|\hat{\mathbf{v}}\|_{TV}, \quad \text{subject to} \quad \hat{\mathbf{v}}|\mathcal{S} = \mathbf{v}|\mathcal{S}, \tag{5.19}$$

where $\mathbf{v}|\mathcal{S}$ denotes values of $\mathbf{v}$ on set $\mathcal{S}$.

5.2.3 Numerical Results

The experiment results are given in this section. The scene of interested is demonstrated in Fig. 5.4, and emissivity signatures are shown in Fig. 5.5. Here we consider two kinds of substances and the scene background. The performance of the proposed imaging method is compared with the point-by-point sampling method, and experiment results are shown below.

The signal-to-noise ratio (SNR) of the imaging system is varied from 10dB to 30dB, and recovered scenes by the point-to-point sampling method and the proposed method are shown in

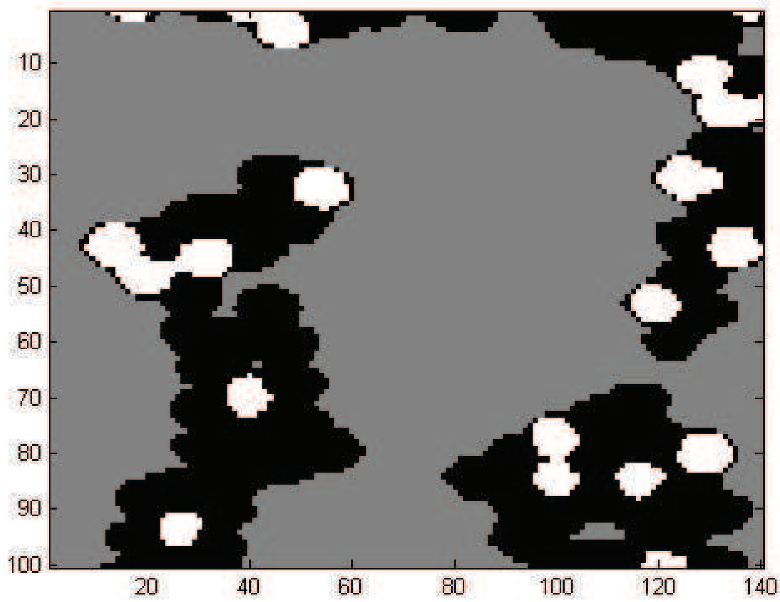


Figure 5.4 Original scene.

Table 5.1 The accuracy of two methods at different SNRs.

SNR(dB)	Point to point method	Proposed method
10	0.9305	0.9209
20	0.9310	0.9224
30	0.9308	0.9241

Fig. 5.6 and Fig. 5.7, respectively. The accuracy of two methods at different SNRs are listed in Table 5.1. The reconstruction results by the point-to-point recovery method is shown in Fig. 5.6. Here, 14000 times of sampling are taken, and 93% pixels are recovered correctly in this case. For the proposed imaging method, 4% of pixels are sampled each time, and 80% of pixels are collected for imaging recovery, i.e., totally 20 times of sampling are taken in our numerical simulations. By comparison, the proposed imaging method outperforms the point-to-point sampling method in sampling efficiency of 700 times speed up with comparable recovery correctness.

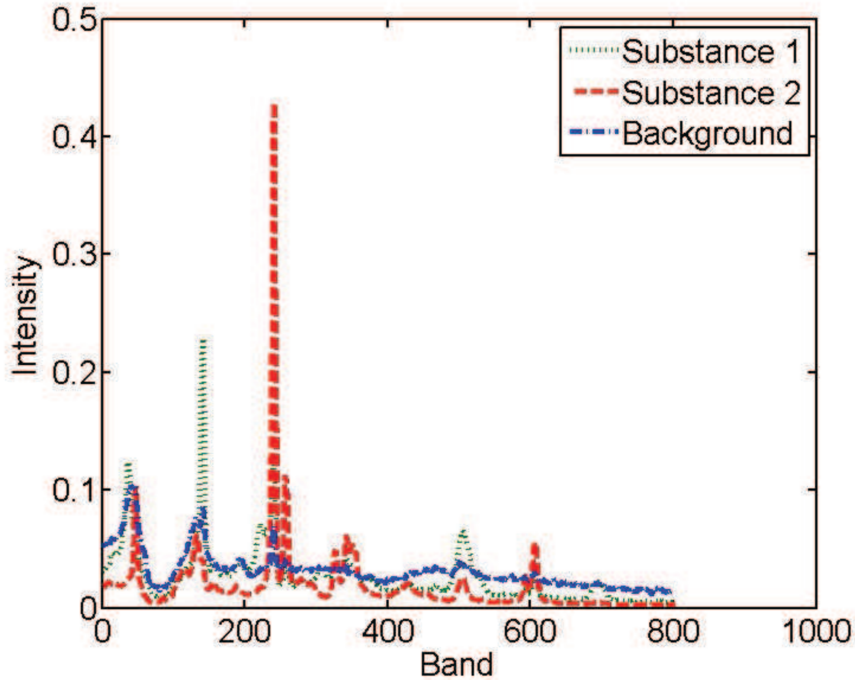
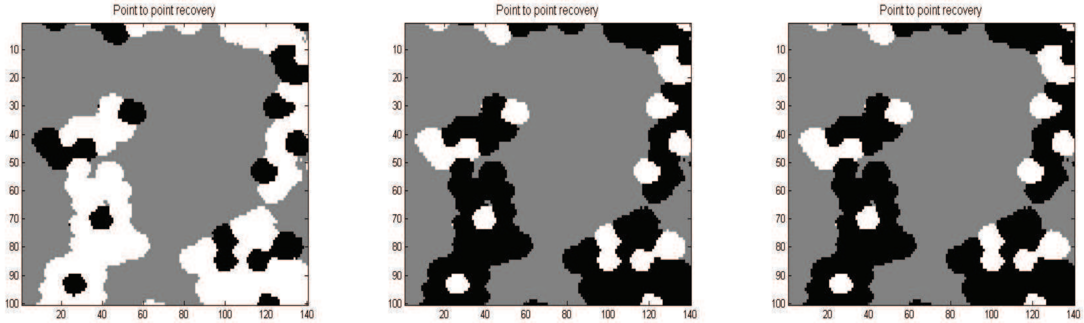


Figure 5.5 The schematic of imaging system.

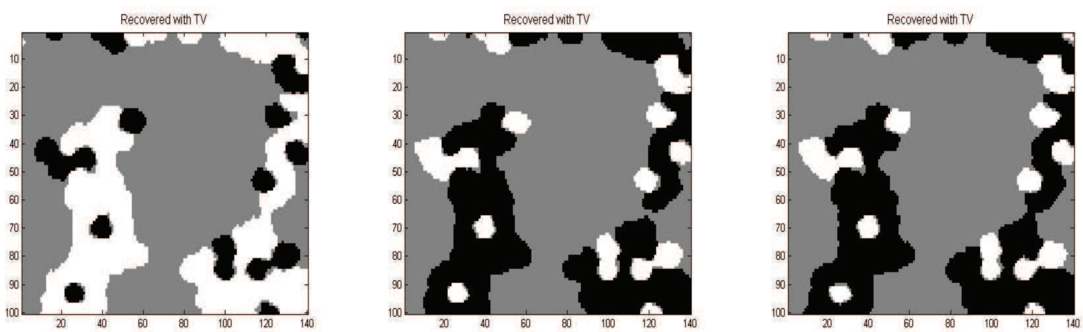
5.3 Conclusion

In this chapter, we have presented applications of big data optimization methods for the non-linear least squares on a wide area network and the compressive hyperspectral imaging. The effectiveness of proposed methods are validated by numerical simulations.



(a) SNR = 10dB　　(b) SNR = 20dB　　(c) SNR = 30dB

Figure 5.6 Recovered scene by point-to-point method.



(a) SNR = 10dB　　(b) SNR = 20dB　　(c) SNR = 30dB

Figure 5.7 Recovered scene by proposed method.

Chapter 6

Conclusions and Future Work

This dissertation dealt with big data optimization for modern communication networks. In this final chapter, we conclude our work and suggest directions for future research.

6.1 Conclusion Remarks

This thesis explored applications of big data optimization in modern communication networks. The techniques and methods which have been developed in this thesis are listed as follows:

- We have reviewed several distributed and parallel optimization methods based on the ADMM for big data optimization problems. We have introduced the background of ADMM from its two precedents: the dual ascent method and the method of multipliers. We have also described several direct extensions and sophisticated modifications of ADMM from 2-block to N-block settings. We have explained iterative schemes and convergence properties for those extensions/modifications. The implementations of reviewed algorithms on large-scale computing facilities are also illustrated.

- We have investigated big data processing techniques for smart grid security. For the security of system state estimation, we have exploited the temporal correlation of time-series state measurements and the sparse nature of malicious attacks to detect the false data injection in the power grid. We have formulated the false data detection problem as a matrix separation problem. Two methods, the nuclear norm minimization method and the low rank matrix factorization method, are proposed to recover electric power states and to detect malicious attacks on the power grid. The proposed methods can also deal with missing measurements. Numerical simulations have been performed to evaluate proposed algorithms. For the security of economical dispatch, we have proposed a distributed parallel approach based on the ADMM to deal with the resulting large-scale optimization problem with manageable complexity. Specifically, we have decoupled and divided the SCOPF problem into independent subproblems of approximately the same size corresponding to pre-contingency and post-contingency cases. Subproblems have been optimized in a parallel fashion on distributed nodes, and dual (price) variables have been designed delicately for coordination. Numerical tests on IEEE buses have validated the effectiveness of the proposed algorithm.

- We have proposed scalable mechanisms for big data traffic management in mobile networks. For mobile data offloading, We have formulated a total revenue maximization problem by jointly considering offloading utilities of BSs and cost of APs. We have applied the proximal Jacobian multi-block ADMM to solve the optimization problem in a fully distributed fashion. We have evaluated the proposed algorithm by numerical simulations. For the service management mechanism in mobile cloud computing, we have jointly considered tasks of clients request allocation and data center response routing, and taken the effect of wireless link latency into account. We have formulated a service revenue maximization problem, in which the mobile service provider optimally locates clients requests to provide qualified service at a reasonable cost. We have used the ADMM to solve the large-scale stochastic optimization problem with manageable complexity, and analyzed the convergence property under the stochastic setting. Our algorithm can decompose the optimization problem into a set of independent subproblems. These subproblems can be solved in a parallel fashion on distributed nodes and coordinated through dual variables. Our numerical tests have validated the effectiveness of the proposed algorithm.

6.2 Future Work

6.2.1 Decentralized State Estimation in Smart Grid

Previous work on smart grid security presented in Chapter 3 used direct current (DC) power flow approximation for system state estimation and optimal power flow dispatch. The DC approximation model can provide quick operation instructions for the system. For precise system status monitoring and operation, alternating current (AC) power flow equations are needed

$$P_i = \sum_{k=1}^{N} |V_i||V_k|(G_{ik}\cos\theta_{ik} + B_{ik}\sin\theta_{ik}) \text{ and} \tag{6.1}$$

$$Q_i = \sum_{k=1}^{N} |V_i||V_k|(G_{ik}\sin\theta_{ik} - B_{ik}\cos\theta_{ik}), \tag{6.2}$$

where P_i and Q_i are real power flow and inactive power flow at bus i, respectively. V_i is the voltage magnitude at bus i. G_{ik} and B_{ik} are the real and imaginary part of the $(i,k)^{th}$ element of the bus admittance matrix. θ_{ik} is the voltage phase angle difference between bus i and bus j. The problem of state estimation is how to find voltage magnitudes and phase angles given nonlinear equations of real and inactive power flows in the system. To estimate the system state in a decentralized fashion, we propose to investigate the following:

- In order to deal with non-convex and non-linearity in AC power flow equations, we can study and design a second-order method such as the Gauss-Newton method to find the solution. We can also consider proper relaxations to make it a convex optimization problem.
- The topology of the electric grid is quite sparse. An optimal partition of the electric grid can simplify the optimization problem by reducing coupling components, facilitating the decentralized computing, and mitigating communication overhead of the algorithm.
- The proposed algorithm can be mapped to high performance computing facilities like high performance computer clusters, which enable real-time monitoring of system states.

6.2.2 Smart Meter Reading Data Clustering

The advanced metering infrastructure (AMI) enables two-way communications with the meter. The smart meters are able to record the consumption of electric energy of each household and send readings to data centers of utility companies for billing and customer service. This provides real time information about electric energy consumption and behaviors of consumers, which can be used for data mining. The smart meters record electric energy consumption of consumers every fifteen minutes, which means that a substantial amount of data are generated daily in the U.S.. By investigating those data, we can better understand profiles of consumers to ensure the quality of service, develop targeted electric energy plans, and accurately predict energy consumption of the power system. We propose to investigate the following for smart meter data clustering:

- In order to reduce the dimension of collected smart meter data for clustering, we can study efficient distributed and parallel methods to perform the principle component analysis. A non-parametric clustering method can be developed to classify consumers into different types even though the number of clusters is unknown before clustering.

- The large amount of smart meter data will incur a huge computational burden for clustering. Even though the computation can be performed in a parallel fashion, the strict time requirement may still be difficult to meet. To further accelerate the computation, we can develop a sub-linear algorithm for clustering.

- We can formulate a dynamic optimization problem to decide the economical dispatch of the power system given the current supply of the electric grid, and design a dynamic pricing mechanism based on clustering results and consumers' profiles.

6.2.3 Efficient Air Quality Monitoring

The air pollution has been an utmost concern for public health nowadays. In 2012, around seven million people dead worldwide due to the air pollution. However, the existing air-quality monitoring network has very low spatial and temporal coverage, which severely limits its ability to predict air quality and to analyze its impact on environment, climate, and public health. Fortunately, there exists a large amount of diverse data, such as satellite remote sensing data, meteorological data (temperature, wind, pressure, humidity, etc.), and traffic data (volume, speed, congestion) which can be utilized. Instead of solely relying on the traditional monitoring network to provide us the air quality data, many heterogeneous big data sources can be used to develop innovative big data processing methods in air quality research. We propose to conduct efficient air quality monitoring research by investigating the following:

- The various data from traditional, emerging, and new sources can be collected and integrated to predict highly temporal and spatial resolved air quality data. The heterogeneous static and dynamic data of different spatiotemporal scales will be collected, integrated, and fed into spatiotemporal models for pollution mapping and source apportionment.

- Computationally tractable models can be trained from various heterogeneous spatiotemporal big data to predict air pollutant concentrations at times and places, where direct readings are not available. A multi-view learning framework which incorporates multiple different temporal and spatial models can be exploited. This model can deal with the huge data size, different types of data taken at different times and locations with different sampling frequencies, and the lack of labeled data.

- To identify and pinpoint major emission sources of air pollutants, we can apply compressed sensing techniques to solve this inverse problem. The sequential compressed sensing and online numerical methods can be developed to deal with the nonlinear process of pollutant formation and its online nature, respectively.

Bibliography

[1] C. Greer, D. A. Wollman, D. E. Prochaska, P. A. Boynton, J. A. Mazer, C. T. Nguyen, G. J. FitzPatrick, T. L. Nelson, G. H. Koepke, A. R. H. Jr, V. Y. Pillitteri, T. L. Brewer, N. T. Golmie, D. H. Su, A. C. Eustis, D. G. Holmberg, and S. T. Bushby, "NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 3.0," The National Institute of Standards and Technology, Tech. Rep. NIST SP - 1108r3, Oct. 2014.

[2] S. Gorman, "Effect of Stealthy Bad Data Injection on Network Congestion in Market Based Power System," *The Wall Street Journal*, Apr 2009.

[3] S. Borlase, *Smart Grids: Infrasture, Technology and Solutions*. Boca Raton, FL: CRC Press, 2012.

[4] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013−2018," Feb. 2014.

[5] L. Liu, Z. Han, S. Cui, and V. H. Poor, "Big data processing for smart grid security," in *Big Data over Networks*, A. Hero, J. Moura, T. Luo, and S. Cui, Eds. Cambridge, U.K.: Cambridge University Press, 2014, pp. 122–156.

[6] L. Liu, M. Esmalifalak, Q. Ding, V. A. Emesih, and Z. Han, "Detecting False Data Injection Attacks on Power Grid by Sparse Optimization," *IEEE Transactions on Smart Grid*, vol. 5, no. 2, pp. 612–621, Mar. 2014.

[7] M. Esmalifalak, L. Liu, N. Nguyen, R. Zheng, and Z. Han, "Detecting Stealthy False Data Injection Using Machine Learning in Smart Grid," IEEE Systems Journal, to appear, 2014.

[8] L. Liu, Z. Han, Z. Wu, and L. Qian, "Spectrum Sensing and Primary User Localization in Cognitive Radio Networks via Sparsity," *EAI Transactions on Wireless Spectrum*, vol. 4, no. 1, pp. 1–14, Jan. 2014.

[9] L. Liu and Z. Han, “Multi-block ADMM for Big Data Optimization in Smart Grid,” in *International Conference on Computing, Networking and Communications*, Anaheim, CA, Feb. 2015.

[10] L. Liu, S. Ren, and Z. Han, “Scalable Workload Management for Water Efficiency in Data Centers,” in *IEEE Globe Communication Conference*, Austin, TX, Dec. 2014.

[11] L. Liu, Q. Ling, and Z. Han, “Decentralized Gauss-Newton Method for Nonlinear Least Squares on Wide Area Network,” in *The 2nd Radio and Antenna Days of the Indian Ocean (RADIO)*, Mauritius, Apr. 2014.

[12] L. Liu, A. Khodaei, W. Yin, and Z. Han, “A Distribute Parallel Approach for Big Data Scale Optimal Power Flow with Security Constraints,” in *IEEE International Conference on Smart Grid Communications*, Vancouver, Canada, Oct. 2013.

[13] L. Liu, M. Esmalifalak, and Z. Han, “Detection of False Data Injection in Power Grid Exploiting Low Rank and Sparsity,” in *IEEE International Conference on Smart Grid Communications*, Budapest, Hungary, Jun. 2013.

[14] L. Liu, H. Li, and Z. Han, “Sampling Spectrum Occupancy Data over Random Fields: A Matrix Completion Approach,” in *IEEE International Conference on Communications*, Ottawa, Canada, Jun. 2012.

[15] L. Liu, Z. Han, Z. Wu, and L. Qian, “Collaborative Compressive Sensing based Dynamic Spectrum Sensing and Mobile Primary User Localization in Cognitive Radio Networks,” in *IEEE Globe Communication Conference*, Houston, TX, Dec. 2011.

[16] P. Tseng and S. Yun, “A Coordinate Gradient Descent Method for Non-smooth Separable Minimization,” *Mathematical Programming*, vol. 117, no. 1, pp. 387–423, 2009.

[17] Y. Li and S. Osher, “Coordinate Descent Optimization for L1 Minimization with Applications to Compressed Sensing: A Greedy Algorithm,” UCLA CAM, Tech. Rep. Report 09-17, 2009.

[18] Y. Nesterov, “Efficiency of Coordiate Descent Methods on Huge-scale Optimization Problems,” *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 341–362, 2012.

[19] L. Bottou and O. Bousquet, “The Tradeoffs of Large Scale Learning,” in *Advances in Neural Information Processing Systems*, Vancouver, Canada, Dec. 2008.

[20] M. Zinkevich, M. Weimer, A. Smola, and L. Li, “Parallelized Stochastic Gradient Descent,” in *Advances in Neural Information Processing Systems*, Vancouver, Canada, Dec. 2010.

[21] F. Niu, B. Recht, C. Re, and S. J. Wright, “Hogwild: A Lock-free Approach to Parallelizing Stochastic Gradient Dscent,” in *Advances in Neural Information Processing Systems*, Granada, Spain, Dec. 2011.

[22] C. J. Hsieh, K. W. Chang, C. J. Lin, S. S. Keerthi, and S. Sundararajan, “A Dual Coordinate Descent Method for Large-scale Linear SVM,” in *International Conference on Machine Learning*, Helsinki, Finland, Jul. 2008.

[23] S. Shalev-Shwartz and T. Zhang, “Stochastic Dual Coordinate Ascent Methods for Regularized Loss Minimization,” *Journal of Machine Learning Research*, vol. 14, pp. 567–599, 2013.

[24] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods (2nd ed.)*. Belmont, MA: Athena Scientific, 1997.

[25] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers,” *Foundation and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, Nov. 2010.

[26] R. M. Freund and P. Grigas, “New Analysis and Results for the Frank-wolfe Method,” online at http://arxiv.org/abs/1307.0873, 2014.

[27] S. Lacoste-Julien, M. Jaggi, M. Schmidt, and P. Pletscher, “Block-coordinate Frank-wolfe Optimization for Structural SVMs,” in *International Conference on Machine Learning*, Atlanta, U.S.A, Jun. 2013.

[28] D. P. Bertsekas, *Nonlinear Programming*. Nashua, USA: Athena Scientific, 1999.

[29] R. Glowinski and A. Marrocco, “Sur L’approximation par Éléments Finis et la Résolution par Pénalisation-dualité D’une Classe de Problèmes de Dirichlet Non Linéaires,” *Revue Française d’Automatique, Informatique, Recherche Operationnelle, Série Rouge*, vol. R-2, pp. 41–76, 1975.

[30] D. Gabay and B. Mercier, “A Dual Algorithm for the Solution of Nonlinear Variational Problems via Finite Element Approximation,” *Computers & Mathematics with Applicaions*, vol. 2, no. 1, pp. 17–40, 1976.

[31] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo, “Fast Image Recovery using Variable Splitting and Constrained Optimization,” *IEEE Transactions on Image Processing*, vol. 19, no. 9, pp. 2345–2356, Sep. 2010.

[32] B. He, M. Tao, and X. Yuan, “Alternating Direction Method with Gaussian Back Substitution for Separable Convex Programming,” *SIAM Journal of Optimization*, vol. 22, no. 2, pp. 313–340, 2012.

[33] W. Deng, M. Lai, Z. Peng, and W. Yin, “Parallel Multi-block ADMM with o(1/k) Convergence,” online at http://arxiv.org/abs/1312.3040, 2014.

[34] B. He, M. Tao, and X. Yuan, “On the Proximal Jacobian Decomposition of ALM for Multiple-block Separable Convex Minimization Problems and its Relationship to ADMM,” http://www.optimization-online.org/DB_FILE/2013/11/4142.pdf, 2013.

[35] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, “RASL: Robust Alignment by Sparse and Low Rank Decomposition for Linearly Correlated Images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2233–2246, Nov. 2012.

[36] M. Tao and X. Yuan, “Recovering Low-rank and Sparse Components of Matrices from Incomplete and Noisy Observations,” *SIAM Journal on Optimization*, vol. 21, no. 1, pp. 51–87, 2011.

[37] H. Xu, C. Feng, and B. Li, “Temperature Aware Workload Management in Geo-distributed Datacenters,” IEEE Transactions on Parallel and Distributed Systems, to appear, 2014.

[38] C. Chen, B. He, Y. Ye, and X. Yuan, “The Direct Extension of ADMM for Multi-block Convex Minimization Problems is not Necessarily Convergent,” preprint, 2013.

[39] M. Hong and Z. Luo, “On the Linear Convergence of the Alternating Direction Method of Multipliers,” online at http://arxiv.org/abs/1208.3922, 2012.

[40] C. Chen, B. He, Y. Ye, and X. Yuan, “The Direct Extension of ADMM for Multi-block Convex Minimization Problems is not Necessarily Convergent,” Online at http://web.stanford.edu/∼yyye/ADMM_5, 2014.

[41] M. Hong, T. Chang, X. Wang, M. Razaviyayn, S. Ma, and Z. Luo, “A Block Successive Upper Bound Minimization Method of Multipliers for Linearly Constrained Convex Optimization,” online at http://arxiv.org/abs/1401.7079, 2014.

[42] B. He, L. Hou, and X. Yuan, “On Full Jacobian Decomposition of the Augmented Lagrangian Method for Separable Convex Programming,” online at http://www.optimization-online.org/DB_HTML/2013/05/3894.html, 2013.

[43] X. Wang, M. Hong, S. Ma, and Z. Luo, “Solving Multiple-block Separable Convex Minimization Problems using Two-block Alternating Direction Method of Multipliers,” online at http://arxiv.org/abs/1308.5294, 2013.

[44] H. Wang, A. Banerjee, and Z. Luo, “Parallel Direction Method of Multipliers,” online at http://arxiv.org/abs/1406.4064, 2014.

[45] N. Parikh and S. Boyd, “Proximal Algorithms,” *Foundation and Trends in Optimization*, vol. 1, no. 3, pp. 123–231, 2013.

[46] Z. Peng, M. Yan, and W. Yin, “Parallel and Distributed Sparse Optimization,” in *IEEE Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, U.S.A, Nov. 2013.

[47] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster Computing with Working Sets," in *2nd USENIX Conference on Hot Topics in Cloud Computing*, Boston, U.S.A, Jun. 2010.

[48] E. Hossain, Z. Han, and H. V. Poor, *Smart Grid Communications and Networking*. Cambridge, UK: Cambridge University Press, 2012.

[49] A. Abur and A. G. Exposito, *Power System State Estimation: Theory and Implementation*. New York: Marcel Dekker, Inc., 2004.

[50] J. J. Grainger and W. D. S. Jr, *Power System Analysis*. New York: McGraw-Hill, 1994.

[51] Y. Liu, M. K. Reiter, and P. Ning, "False Data Injection Attacks Against State Estimation in Electric Power Grids," in *Proc. 16th ACM Conference on Computer and Communications Security*, Chicago, IL, Nov. 2009.

[52] L. Xie, Y. Mo, and B. Sinopoli, "False Data Injection Attacks in Electricity Markets," in *Proc. IEEE International Conference on Smart Grid Communications*, Gaithersburg, MD, Oct. 2010.

[53] M. Esmalifalak, Z. Han, and L. Song, "Effect of Stealthy Bad Data Injection on Network Congestion in Market Based Power System," in *Proc. IEEE Wireless Communications and Networking Conference*, Paris, France, Apr. 2012.

[54] G. Dán and H. Sandberg, "Stealth Attacks and Protection Schemes for State Estimators in Power Systems," in *Proc. IEEE International Conference on Smart Grid Communications*, Gaithersburg, MD, Oct. 2010.

[55] M. Esmalifalak, G. Shi, Z. Han, and L. Song, "Bad Data Injection Attack and Defense in Electricity Market using Game Theory Study," *IEEE Transactions on Smart Grid*, vol. 4, no. 1, pp. 160–169, Mar. 2013.

[56] O. Kousut, L. Jia, R. J. Thomas, and L. Tong, "Malicious Data Attacks on the Smart Grid," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 645–658, Dec. 2011.

[57] T. T. Kim and H. V. Poor, “Strategic Protection Against Data Injection Attacks on Power Grids,” *IEEE Transactions on Smart Grid*, vol. 2, no. 2, pp. 326–333, Jun. 2011.

[58] S. Cui, Z. Han, S. Kar, T. T. Kim, H. V. Poor, and A. Tajer, “Coordinated Data-injection Attack and Detection in the Smart Grid: A Detailed Look at Enriching Detection Solutions,” *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 106–115, Sep. 2012.

[59] Y. Zhao, A. Goldsmith, and H. V. Poor, “Fundamental Limits of Cyber-physical Security in Smart Power Grids,” in *Proc. IEEE 52nd Annual Conference on Decision and Control*, Florence, Italy, Dec. 2013.

[60] Z. Han, H. Li, and W. Yin, *Compressive Sensing for Wireless Communication*. Cambridge, UK: Cambridge University Press, 2012.

[61] E. J. Cands and B. Recht, “Exact Matrix Completion via Convex Optimization,” *Communications of the ACM*, vol. 55, no. 6, pp. 111–119, Jun. 2009.

[62] E. J. Candès, X. Li, Y. Ma, and J. Wright, “Robust Principal Component Analysis?” *Journal of the ACM*, vol. 58, no. 3, pp. 1–37, May. 2011.

[63] Z. Lin, M. Chen, L. Wu, and Y. Ma, “The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-rank Matrices,” UIUC, Tech. Rep. UILU-ENG-09-2215, Urbana, FL, 2009.

[64] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA: Athena Scientific, 1999.

[65] J. Cai, E. J. Candès, and Z. Shen, “A Singular Value Thresholding Algorithm for Matrix Completion,” *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, Jan. 2010.

[66] Y. Shen, Z. Wen, and Y. Zhang, “Augmented Lagrangian Alternating Direction Method for Matrix Separation Based on Low-rank Factorization,” Rice CAAM, Tech. Rep. TR11-02, Houston, TX, 2011.

[67] Z. Wen, W. Yin, and Y. Zhang, "Solving a Low-rank Factorization Model for Matrix Completion by A Nonlinear Successive Over-relaxation Algorithm," Rice CAAM, Tech. Rep. TR10-07, Houston, TX, 2010.

[68] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "MAT-POWER 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, Feb. 2011.

[69] M. Shahidehpour, W. F. Tinney, and Y. Fu, "Impact of Security on Power System Operation," *Proceedings of the IEEE*, vol. 93, no. 11, pp. 2013–2025, Nov. 2001.

[70] O. Alsac and B. Scott, "Optimal Load Flow with Steady-state Security," *IEEE Transaction on Power Apparatus and System*, vol. 93, no. 3, pp. 745–751, May. 1974.

[71] M. V. F. Pereira, A. Monticelli, and L. M. V. G. Pinto, "Security-constrained Dispatch with Corrective Rescheduling," in *Proc. IFAC Symposium on Planning and Operation of Electric Energy System*, Rio de Janeiro, Brazil, Jul. 1985.

[72] A. J. Wood and B. F. Wollenberg, *Power Generation Operation and Control*. New York: Wiley, 1996.

[73] A. Monticelli, M. V. F. Pereira, and S. Granville, "Security-constrained Optimal Power Flow with Post-contingency Corrective Rescheduling," *IEEE Transactions on Power Systems*, vol. 2, no. 1, pp. 175–180, Feb. 1987.

[74] F. Capitanescu, J. L. M. Ramos, P. Panciatici, D. Kirschen, A. M. Marcolini, L. Platbrood, and L. Wehenkel, "State-of-the-art, Challenges, and Future Trends in Security Constrained Optimal Power Flow," *Electric Power System Research*, vol. 81, no. 8, pp. 1731–1741, Aug. 2011.

[75] J. Martínez-Crespo, J. Usaola, and J. L. Fernández, "Security-constrained Optimal Generation Scheduling in Large-scale Power Systems," *IEEE Transactions on Power Systems*, vol. 21, no. 1, pp. 321–332, Feb. 2006.

[76] Y. Fu, M. Shahidehpour, and Z. Li, “AC Contingency Dispatch Based on Security-constrained Unit Commitment,” *IEEE Transactions on Power Systems*, vol. 21, no. 2, pp. 897–908, May. 2006.

[77] F. Capitanescu and L. Wehenkel, “A New Iterative Approach to the Corrective Security-constrained Optimal Power Flow Problem,” *IEEE Transactions on Power Systems*, vol. 23, no. 4, pp. 1533–1541, Nov. 2008.

[78] Y. Li and J. D. McCalley, “Decomposed SCOPF for Improving Efficiency,” *IEEE Transactions on Power Systems*, vol. 24, no. 1, pp. 494–495, Feb. 2009.

[79] R. Baldick, B. H. Kim, C. Chase, and Y. Luo, “A Fast Distributed Implementation of Optimal Power Flow,” *IEEE Transactions on Power Systems*, vol. 14, no. 3, pp. 858–864, Aug. 1989.

[80] M. Kraning, E. Chu, J. Lavaei, and S. Boyd, “Dynamic Network Energy Management via Proximal Message Passing,” *Foundation and Trends in Optimization*, vol. 1, no. 2, pp. 70–122, 2013.

[81] W. Deng and W. Yin, “On the Global and Linear Convergence of the Generalized Alternating Direction Method of Multipliers,” Rice CAAM, Tech. Rep. TR12-14, Houston, TX, 2012.

[82] J. Nocedal and S. J. Wright, *Numerical Optimization (2nd ed.)*. New York: Springer, 2006.

[83] K. Lee, J. Lee, Y. Yi, I. Rhee, and S. Chong, “Mobile Data Offloading: How Much Can WiFi Deliver?” *IEEE/ACM Transactions on Networking*, vol. 21, no. 2, pp. 536–550, Apr. 2013.

[84] Cisco, “Software-defined Networking: The New Norm for Networks,” Apr. 2012.

[85] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “OpenFlow: Enabling Innovation in Campus Networks,” in *ACM SIGCOMM*, Seattle, WA, Aug. 2008.

[86] A. Balasubramanian, R. Mahajan, and A. Venkataramani, “Augmenting Mobile 3G Using WiFi,” in *8th International Conference on Mobile Systems, Applications, and Services*, San Francisco, CA, Jun. 2010.

[87] S. Dimatteo, P. Hui, B. Han, and V. O. K. Li, “Cellular Traffic Offloading through WiFi Networks,” in *IEEE 8th International Conference on Mobile Adhoc and Sensor Systems (MASS)*, Valencia, Spain, Oct. 2011.

[88] X. Zhuo, W. Gao, G. Cao, and Y. Dai, “Win-coupon: An Incentive Framework for 3G Traffic Offloading,” in *19th IEEE International Conference on Network Protocols (ICNP)*, Vancouver, Canada, Oct. 2011.

[89] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, “ThinkAir: Dynamic Resource Allocation and Parallel Execution in the Cloud for Mobile Code Offloading,” in *IEEE INFOCOM*, Orlando, FL, Mar. 2012.

[90] B. Han, P. Hui, V. S. A. Kumar, M. V. Marathe, J. Shao, and A. Srinivasan, “Mobile Data Offloading through Opportunistic Communications and Social Participation,” *IEEE Transactions on Mobile Computing*, vol. 11, no. 5, pp. 821–834, May 2012.

[91] L. Gao, G. Iosifidis, J. Huang, and L. Tassiulas, “Economics of Mobile Data Offloading,” in *IEEE SDP Workshop*, Turin, Italy, Apr. 2013.

[92] G. Iosifidis, L. Gao, J. Huang, and L. Tassiulas, “An Iterative Double Auction for Mobile Data Offloading,” in *11th International Symposium on Modeling Optimization in Mobile, Ad Hoc Wireless Networks (WiOpt)*, IIT Bombay, Mumbai, May 2013.

[93] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, “Layering as Optimization Decomposition: A Mathematical Theory of Network Architectures,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 255–312, Jan. 2007.

[94] M. Grant and S. Boyd, “CVX: Matlab Software for Disciplined Convex Programming, Version 2.1,” http://cvxr.com/cvx, Mar. 2014.

[95] B. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, “CloneCloud: Elastic Execution between Mobile Device and Cloud,” in *sixth conference on computer systems*, Salzburg, Austria, Apr. 2011.

[96] R. Bifulco, M. Brunner, R. Canonico, R. Hasselmeyer, and F. Mir, “Scalability of a Mobile Cloud Management System,” in *ACM SIGCOMM workshop on mobile cloud computing*, Helsinki, Finland, Aug. 2012.

[97] P. Wendell, J. W. Jiang, M. J. Freedman, and J. Rexford, “DONAR: Decentralized Sever Selection for Cloud Services,” in *ACM SIGCOMM*, New Delhi, India, Aug. 2010.

[98] Z. Zhang, M. Zhang, A. Greenberg, Y. C. Hu, R. Mahajan, and B. Christian, “Optimizing Cost and Performance in Online Service Provider Networks,” in *NSDI*, San Jose, CA, Apr. 2010.

[99] D. K. Goldenberg, L. Qiu, H. Xie, Y. R. Yang, and Y. Zhang, “Optimizing Cost and Performance for Multihoming,” in *ACM SIGCOMM*, Protland, OR, Aug. 2002.

[100] H. Xu, C. Feng, and B. Li, “Temperature Aware Workload Management in Geo-distributed Data Centers,” in *USENIX ICAC*, San Jose, CA, Jun. 2013.

[101] J. W. Jiang, R. Zhang-Shen, J. Rexford, and M. Chiang, “Cooperative Content Distribution and Traffic Engineering in an ISP Network,” in *ACM SIGMETRICS*, Seattle, WA, Jun. 2009.

[102] S. Narayana, J. W. Jiang, J. Rexford, and M. Chiang, “To Coordinate or not to Coordinate? Wide-area Traffic Management for Data Centers,” in *ACM CoNEXT*, Nice, France, Dec. 2012.

[103] H. Xu and B. Li, “Joint Request Mapping and Response Routing for Geo-distributed Cloud Services,” in *IEEE INFOCOM*, Turin, Italy, Apr. 2013.

[104] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, “The Case for VM-based Cloudlets in Mobile Computing,” *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, Oct. 2009.

[105] M. Jeonghoon and J. Walrand, "Fair End-to-end Window-based Congestion Control," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 556–5671, Oct. 2000.

[106] F. P. Kelly, A. Maulloo, and D. Tan, "Rate Control for Communication Networks: Shadow Price, Proportional Fairness and Stability," *J. Operational Research Society*, vol. 49, pp. 237–252, Mar. 1998.

[107] L. Rao, X. Liu, L. Xie, and W. Liu, "Minimizing Electricity Cost: Optimization of Distributed Internet Data Centers in a Multi-electricity-market Environment," in *IEEE INFOCOM*, San Diego, CA, Mar. 2010.

[108] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The Cost of a Cloud: Research Problems in Data Center Networks," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 1, pp. 68–73, Jan. 2009.

[109] X. Fan, W. Weber, and L. A. Barroso, "Power Provisioning for a Warehouse-size Computer," in *ACM International Symposium on Computer Architecture*, San Diego, CA, Jun. 2007.

[110] H. V. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "iplane: An Information Plane for Distributed Services," in *NSDI*, Seattle, WA, Nov. 2006.

[111] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, "Robust Stochastic Approximation Approach to Stochastic Programming," *SIAM J. on Optimization*, vol. 19, no. 4, pp. 1574–1609, Jan. 2009.

[112] H. Ouyang, N. He, L. Q. Tran, and A. Gray, "Stochastic Alternating Direction Method of Multipliers," in *ICML*, Atlanta, GA, Jun. 2013.

[113] B. He and X. Yuan, "On the $\mathcal{O}(\frac{1}{t})$ Convergence Rate of the Douglas-Rachford Alternating Direction Method," *SIAM J. Numer. Anal*, vol. 50, no. 2, pp. 700–709, Apr. 2012.

[114] H. Wang and A. Banerjee, "Online Alternating Direction Method," in *ICML*, Edinburgh, Scotland, Jun. 2012.

[115] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computing: Numerical Methods*. NJ, USA: Prentice-Hall, 1989.

[116] X. Li and A. Scaglione, “Convergence and Applications of Gossip-Based Gauss-Newton Algorithm,” *IEEE Transactions on Signal Processing*, vol. 61, no. 21, pp. 5231–5246, Nov. 2013.

[117] P. Stoica, R. Mose, B. Friedlander, and T. Soderstrom, “Maximum Likelihood Estimation of the Parameters of Multiple Sinusoids from Nosiy Measurements,” *IEEE Transaction on Acoustics, Speech and Signal Processing*, vol. 37, no. 3, pp. 378–392, Mar. 1989.

[118] K. Zhou and S. Roumeliotis, “Multirobot Active Target Tracking with Combinations of Relative Observations,” *IEEE Transaction on Robotics*, vol. 27, no. 4, pp. 678–695, Aug. 2011.

[119] A. Nedic and A. Ozdaglar, “Distributed Subgradient Methods for Multi-agent Optimization,” *IEEE Transaction on Automatic Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[120] S. Ram, A. Nedic, and V. Veeravalli, “Distributed Stochastic Subgradient Projection Algortims for Convex Optimization,” *Journal of Optimization Theory and Applications*, vol. 147, no. 3, pp. 516–545, Dec. 2010.

[121] K. Srivastava and A. Nedic, “Distributed Asynchronous Constrained Stochastic Optimization,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 772–790, Aug. 2011.