New Large-Scale Optimization Methods with Applications in Communication Networks, Medical Imaging, and System Identification

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Navid Reyhanian

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Zhi-Quan Luo
Jarvis Haupt

March, 2023

Acknowledgements

There are many people that I would like to show my appreciation for their precious time spent on me during my Ph.D. journey. First of all, I would like to express my deepest gratitude toward my advisor, Professor Zhi-Quan Luo, for his continuous encouragement, support, and help during my graduate studies. I am very grateful for many hours he spent with me during our brainstorming sessions. It has been a significant privilege to be mentored by such an amazing Professor. He played a significant role in my intellectual and research ability growth.

I thank my co-advisor, Professor Jarvis D. Haupt, who has helped me in developing clear and scientific thought. The completion of this thesis was impossible without him. Professor Haupt is a great mentor who spends a lot of his time with students to make sure that they can find the right way to progress. I also thank Professor Michael Garwood for his hospitality and for accepting me in his group meeting. I obtain a lot knowledge in magnetic resonance imaging from Professor Garwood. His patience, positive attitude, and encouragement easied the process of learning a new area for me.

Professors Luo, Haupt, and Garwood spent a tremendous amount of their time in guiding me in the right direction as well as improving my presentation and writing skills. I also thank Professor Mostafa Kaveh who chaired my Ph.D. final oral exam and provided insightful comments.

Furthermore, I was fortunate to have many wonderful friends during my Ph.D. and I am very appreciative of fun experiences shared with them as well as their incredible support at difficult times.

I give my sincerest gratitude to my father, my mother, and my sister for their unconditional love and support. You are a great source of inspiration, motivation, and encouragement for me. I love you and this thesis is all dedicated to you.

Last but not the least, I acknowledge all the funding supports from the Electrical Engineering Department and the Graduate School at the University of Minnesota as well as National Science Foundation (NSF).

Dedication

This dissertation is dedicated to my parents, Hamid and Fariba, and my sister, Parya, for their unconditional love.

Abstract

Optimizations with large sizes appear frequently in modern problems of quantitative fields to enhance current solutions. This dissertation is concerned with the development and analysis of large-scale optimization methods for communication networks, magnetic resonance imaging, and dynamical systems.

The first chapter concentrates on the algorithm development for the joint problem of mapping virtual network functions to high-volume servers, resource provisioning, and traffic routing where the demand is known. In the next two chapters of this dissertation, we focus on the problem of joint resource reservation in the backhaul and radio access network where user demands and achievable rates of wireless channels are unknown; however, observations from these two randomness sources are available. We propose a novel method to maximize the sum of expected traffic flow rates, subject to link and access point budget constraints, while minimizing the expected outage of wireless channels. We use the proposed resource reservation method to dynamically slice network resources among several slices of different tenants when user demands and achievable rates of wireless connections are uncertain. We propose a two-timescale scheme in which a subset of network slices is activated via a novel sparse optimization framework in the long timescale with the goal of maximizing the expected utilities of tenants while in the short timescale the activated slices are reconfigured according to the time-varying user traffic and channel states.

The next two chapters of this thesis focus on magnetic resonance imaging with single and multiple echoes for fast acquisitions with long readout times, e.g., spiral, when the static magnetic field B_0 is largely nonuniform and inhomogeneous. We study the non-convex problem of joint image and B_0 field map estimation from a set of distorted images due the B_0 inhomogeneity. We propose novel voxel-level decompositions and develop parallel and distributed approaches based on block coordinate descent and golden-section-search methods to iteratively improve the estimates of the field map. Unlike prior works, 2D and 3D spherical harmonics are utilized in proposed algorithms to efficiently regularize field map estimations. The estimated field map is later leveraged to correct different artifacts of distorted images.

The final chapter investigates the problem of estimating the weight matrices of a stable time-invariant linear dynamical system from a single sequence of noisy measurements. Unlike existing methods that identify equivalent systems, we show that if the unknown weight matrices describing the system are in Brunovsky canonical form, we can efficiently estimate the ground truth unknown matrices of the system from a linear system of equations formulated based on the transfer function of the system and iterates of stochastic gradient descent methods.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Optimization theory is concerned with the mathematical study of minimizing or maximizing of a function value on a given domain. Optimization problems of different kinds arise in the large variety of quantitative fields such as engineering, operations research, computer science, and applied mathematics. Optimization problems in different disciplines attain various structures. Many optimizations in the above areas have large dimensions and complicated structures that avoid closed-form solutions. To efficiently solve these problems, different methods and techniques have been developed, and significant effort is being made.

A common characteristic of modern problems in the above disciplines is their large size, which can slow or stop classical methods. Due to this reason, tractable optimization techniques, which exploit the problem structures and also applicable for many other problems, are necessary. The scalability of optimization algorithms should be enhanced, which entails problem decompositions and efficient distributed and parallel implementations. With smart decompositions, algorithms can solve a sequence of smaller subproblems in a parallel fashion using modern multi-core high performance computing technologies. This thesis is concerned with development and analysis of large-scale optimization algorithms for quantitative areas, particularly resource allocations in communication networks, magnetic field estimation and image correction in magnetic resonance imaging, and system identification in linear dynamical systems. In this thesis, we select a number of interesting and practical research topics from the above three areas, study them from the optimization perspective, and demonstrate the superiority of proposed approaches relative to existing methods and techniques. In the rest of this chapter, we separately discuss different studied areas.

Resource allocations in different networks are conventionally carried out under two assumptions. One assumption is that the user demand is known, and network resources should be optimized such that the demand is maximally supported, while costs of network operations and resource utilizations are minimized. The other assumption is that the user demand is unknown and uncertain. Under this assumption, the user demand is predicted and later used to allocate resources such that the robustness of resource allocations for future demands increases. In general, more resources are utilized to support demands when they are uncertain.

For resource allocations with known user demands, the major challenge is the optimization of different types of resources among users a certain time before user demands start to apply. The challenge becomes harder when it is necessary to allocate resource in a short time before the demand starts. Therefore, distributed and fast methods are desirable to carry out resource allocations. The development and analysis of distributed methods that benefit from parallel computing have received significant attention recently [3, 4, 5, 6, 7].

For resource allocations with unknown user demands, the major challenge is to optimize resources among users such that upon the realization of demands, the reserved resources can support user demands similar to when they are known. To carry out this task, the statistics of user demands is usually derived from collected observations and is used to better predict the demand behavior in the future. Many approaches for resource reservations based on different types of user demands are proposed [8, 9, 10, 11, 12]. For example, a number of works use Probability Density Functions (PDFs) of user demands to implement resource reservations [13], while other works, e.g., [7, 14], consider ranges for uncertain demands and assign resources to users accordingly aiming at enhancing the robustness of resource reservations.

In the other part of this dissertation, Magnetic Resonance Imaging (MRI) in the presence of B_0 inhomogeneity is studied. The B_0 magnetic field in MRI is the main static magnetic field that is used to line up the hydrogen atoms in the scanned object. B_0 is a strong magnetic field and it is measured in Tesla (T) units. The B_0 strength depends on the type of scanner, but it is usually between 1.5 T and 3 T for clinical MRI scanners. Typically, MR techniques require a polarizing static magnetic field along the z axis [15]. Therefore, the magnetic field is assumed to be spatially homogeneous along the z axis. However, with more compact and portable high-field MRI systems, the field B_0 is no longer homogeneous. B_0 inhomogeneity degrades the Signal-to-Noise Ratio (SNR) of all MR measurements [15]. Spiral and Echo-Planar-Imaging (EPI) are especially sensitive to B_0 inhomogeneity, and generated images via above two encodings become distorted [16, 17, 18]. Advanced optimization methods are required to estimate the magnetic field from distorted images, and next use it to correct images with artifacts.

In the last part of this dissertation, the identification of linear dynamical systems from a single trajectory is studied. Dynamical systems are studied and investigated by different engineering fields, e.g., electrical and aerospace engineering, mathematics, and data science due to their wide applications. The identification of dynamical systems was traditionally studied by control and systems theory in electrical engineering. More recent system identification approaches concentrate on data-driven methods, which benefit from the vast available data. The identification of linear systems usually requires solving non-convex optimizations, and local solutions are obtained without efficient initializations. More efficient methods are required to enhance the reliability of system identifications. Moreover, existing works usually focus on identifying an equivalent system rather than the ground truth. Therefore, new methods are particularly considered interesting when the ground truth system is needed.

This dissertation is dedicated to the development and analysis of fast, scalable, and robust optimization algorithms to address different problems in the above areas. Different chapters investigate the mathematical aspects of the investigated problems, which are all practical, real-world issues. In the following sections, we separately discuss the studied topics in this dissertation, and explain the motivations.

1.1 Joint Resource Allocation and Routing for Service Function Chaining with In-Subnetwork Processing

Network Function Virtualization (NFV) has emerged as a promising solution, since it enables to replace dedicated hardware implementations with software instances running in a virtualized environment. In NFV, a service is support through a sequence of Virtual

Network Functions (VNF) that can run on common servers by leveraging the virtualization technology. VNFs are placed in a sequence with a specified order through which data flows traverse. This sequence is also known as the Service Function Chain (SFC). Benefiting from virtualization technology, the SFC can be created by embedding the requested VNFs in a SFC in an agile and efficient manner. The advantage of NFV is that one or more VNFs can be dynamically added or deleted at a low cost to cope with the time-varying demands. NFV allows to allocate network resources in a more scalable and flexible manner, which results in a more efficient management and operation mechanism for network functions.

In Chapter 2, we consider a large service-oriented network spanning several subnetworks, which are operated by a different administrator. We decouple the management of different subnetworks by developing decentralized approaches. Each subnetwork owner wishes to route and serve flow required VNF services by servers and links within its own subnetwork when flow rate demands are known. We first propose a centralized optimization approach for joint traffic routing and NFV cost minimization in which flows are encouraged to be processed in local subnetworks. Next, we decompose the centralized problem and propose a distributed approach based on the proximal Alternating Direction Method of Multipliers (ADMM) to jointly optimize the routing and VNF placement cost. The distributed optimization can be locally solved while minimal information is shared with a central controller. The proposed distributed approach addresses scalability for large networks and confidentiality, and also decreases the cost of implementation. We propose efficient an dual algorithm based on successive upper-bound minimization to solve the subproblems of our distributed approach.

In Chapter 2, we further consider the problem of link capacity and sparse server purchase cost minimization when flows are processed in local subnetworks. Given flow demands, we formulate the problem from a sparse optimization perspective and propose an efficient approach based on iteratively solving a sequence of group Least Absolute Shrinkage and Selection Operator (LASSO) problems. Via purchased link capacities and servers, a subnetwork administrator can jointly optimize VNF placement on servers and traffic engineering

1.2 Resource Reservation in Backhaul and Radio Access Network with Uncertain User Demands

Since the demand for wireless services is increasing significantly, efficient end-to-end network planning is important to enhance the network coverage and supported rates. To do network planning, typically, each network layer in wireless data networks is optimized separately, which results in a suboptimal performance and efficiency. Moving from optimizing each networking layer in isolation, we study the end-to-end (from datacenters to users) resource reservation in data networks under the assumption that user demands are unknown and uncertain. Collected user demands are utilized to derive statistics.

In Chapter 3, we study link capacity and transmission resource reservation in wireless data networks prior to the observation of user demands. Using the statistics of user demands and achievable rates of downlinks, we formulated an optimization problem to maximize the sum of user expected supportable traffic while minimizing the expected outage of downlinks. We demonstrated that this problem is non-convex in general. To solve the problem approximately, an efficient Block Coordinate Descent (BCD) approach is proposed which benefits from distributed and parallel computation when each block of variables is chosen to be updated.

In the proposed BCD method, the first block of variables determines the link capacity reservation in the backhaul and the other block of variables specifies the transmission resource reservation in RAN. We alternately optimize the two blocks of variables in the BCD algorithm. We propose a multi-path routing that decomposes the problem down to link-level and parallelizes the computation across backhaul links. Based on the convergence theory for Block Successive Upper-bound Minimization (BSUM) methods in [19], we prove that the proposed multi-path routing is convergent to the global minima of an arbitrary convex cost function with Lipschitz continuous gradient. After updating the link capacity reservations, we update the transmission resource reservation in RAN. Since the resource reservation problem in RAN is possibly non-convex, we propose a distributed algorithm based on the BSUM techniques to iteratively solve a sequence of convex approximations of the original problem. We demonstrated that despite the non-convexity of the problem, our proposed approach converges to a KKT solution to the problem.

1.3 Data-Driven Adaptive Network Resource Slicing for Multi-Tenant Networks

Recently, network slicing has been proposed as a promising solution for creating end-to-end network instance across data networks [20, 21]. Network slicing that enables broadcasting services grows both in the 5G core and RAN. End-to-end network slicing in 5G and beyond enhances the performance of networks to support various demands, which are often difficult to predict and manage. Typically, static network slicing forces certain limitations on the allocated resources [22]. Therefore, certain slices could be heavily loaded while many others would remain significantly underutilized. Such a phenomenon could degrade the Quality-of-Service (QoS) for users. Adaptive network slicing schemes that remain robust even with traffic variations over time are highly demanded.

In Chapter 4, we study dynamically slicing network resources in the backhaul and RAN prior to user demand observations across multiple tenants, where each tenant owns and operates several slices to provide different services to users. In the proposed two time-scale scheme, a subset of network slices is activated via a novel sparse optimization framework in the long time-scale with the goal of maximizing the expected utilities of tenants, while in the short time-scale the activated slices are reconfigured according to the time-varying user traffic and channel states. Specifically, using the statistics from users and channels and also considering the expected utility from serving users of a slice and the reconfiguration cost, we formulate a sparse optimization problem to update the configuration of a slice resources such that the maximum isolation of reserved resources is enforced. The formulated optimization problems for long and short time- scales are non-convex and difficult to solve. We use the ℓ_q-norm, $0 < q < 1$, and group LASSO regularizations to iteratively find convex approximations of the optimization problems. We propose a Frank-Wolfe algorithm to iteratively solve approximated problems in long time-scales. To cope with the dynamical nature of traffic variations, we propose a fast, distributed algorithm to solve the approximated optimization problems in short time-scales.

In the short time-scale, we design utility functions for slices of each tenant based on the acquired revenue from users, the QoSs that slices provide for their users, and the cost of slice reconfiguration. We formulate a sparse optimization problem to adaptively reconfigure network slices only if the expected utility of a tenant significantly changes after the reconfiguration. To tackle the non-convexity and non-continuity of the slice reconfiguration cost, we leverage the group LASSO regularization. Moreover, we propose an ADMM algorithm to solve each (non-convex) group LASSO subproblem in the short time-scale. The proposed ADMM algorithm carries out the slice reconfiguration in the backhaul through link capacity reservation via a fast, distributed algorithm that successively minimizes a sequence of convex approximation of the objective function and parallelizes computations across backhaul links. Additionally, the proposed ADMM algorithm implements the slice reconfiguration in RAN through the transmission resource reservation for slices using two methods: 1) a proximal gradient descent method that decomposes the problem across slices; and 2) a bisection search method. We prove that the proposed ADMM algorithm converges to the global solution of each group LASSO subproblem despite its non-convexity.

1.4 Guided Joint Image and field Map Estimation for MRI with Magnetic Field Inhomogeneity

One considerable assumption in the Fourier model is that the spatial encoding is homogeneous, which is difficult to achieve with modern, compact scanners. There are a number of reasons that make B_0 inhomogeneous, namely eddy currents, coupling between gradient coils, timing inaccuracies, concomitant fields, and gradient amplifier non-linearities. When the magnetic field is not spatial homogeneous, different artifacts hurt images if they are not corrected. Non-Cartesian trajectories, e.g., spiral or EPI, are especially prone to these artifacts, since data are acquired during gradient field variations. Moreover, Cartesian scans that put a high demand on the gradient system can also be vulnerable from field perturbations. The estimation of inhomogeneous magnetic field and leveraging that to correct distorted images comprise the main focus of our effort.

Many methods deployed for MRI suffer from degraded image quality when the static magnetic field (B_0) is non-uniform, as B_0 inhomogeneity causes different types of image artifacts and distortions. This work addresses this problem via a joint image and B_0 map estimation using two datasets, the first one containing distorted images and the second encompassing noisy, undistorted measurements for the same set of objects. A distributed, parallel BCD algorithm is proposed to jointly estimate images and a linear transformation that maps the distorted images to the ground-truth images. The proposed BCD method decomposes the problem down to voxel level. Next, using the approximated linear transformation, a golden-section search algorithm nested inside a grid-search is used to estimate the off-resonance frequency present in each voxel in a parallel fashion. With sufficient accuracy, the global solution for the off-resonance frequency is obtained. The accuracy of the estimated off-resonance frequency in each voxel depends on the acceleration factor used in image acquisition and the noise level in the second dataset. Finally, a linear regression procedure is then used to regularize estimated off-resonance frequencies and represent the surface of off-resonance frequencies as a superposition of spherical harmonic functions.

1.5 Regularized Joint Image and Field Map Estimation in 3D Multi-Echo MRI

As mentioned in the previous section, the estimation of B_0 inhomogeneity that causes image blurs, geometric distortions, and signal dropouts is critical for fast acquisitions with long readout times. Unlike the previous section that considers scans with a single echo, we study multi-echo image correction in this section.

In Chapter 6, we study joint image and 3D field map estimation in the presence of large magnetic field inhomogeneity from multiple echoes, where signals from several 2D slices are collected from multiple echoes received by different coils. The joint image and field map optimization problem is known to be non-convex and difficult to solve in a low SNR regime with phase wrapping ambiguity. The joint optimization is studied under the assumption that the object induced field map from its susceptibility map can be well measured or removed through filtering the k-space raw data.

We propose a BCD method to solve the joint optimization problem to global optimality. The proposed BCD algorithm optimizes image voxels using closed-form solutions, while wrapped phase errors are globally optimized via a golden-section search method nested inside a grid-search. We unwrap the optimized wrapped phase errors and project them to the subspace spanned by spherical harmonics up to the third order, aiming at efficiently regularizing estimations. We extend the proposed BCD approach to estimate time-varying magnetic fields, which can be induced due to several reasons, e.g., eddy currents. Jointly optimizing phase errors in different 2D slices and relating them to one inhomogeneous magnetic field allows the proposed algorithm to improve the accuracy of field map estimation via increasing the number of measurements.

1.6 Stochastic Gradient Descent Learns Linear Dynamical Systems from A Single Trajectory

System identification is a key problem in control theory, reinforcement learning, econometrics, and time-series analysis. With one or multiple trajectories of input and output pairs, the behavior of a dynamical system can be analyzed aiming at estimating the parameters of the underlying system. The identification of dynamical systems gains recent attention due to the inherent connections of dynamical systems with recurrent neural networks. Without non-linear state transitions from a recurrent neural network, state transitions of a linear dynamical system are remained. Due to the composition of multiple linear operators in linear systems, proposed objective functions for the identification of a linear system are typically non-convex. The algorithm design and analysis for the identification methods of dynamical system remain an active research area.

In Chapter 7, we investigate the problem of estimating the weight matrices of a stable time-invariant linear dynamical system from a single sequence of noisy measurements. We show that if the unknown weight matrices describing the system are in Brunovsky canonical form, we can efficiently estimate the ground truth unknown matrices of the system from a linear system of equations formulated based on the transfer function of the system, using both online and offline Stochastic Gradient Descent (SGD) methods. We show that SGD converges linearly in expectation to any arbitrary small Frobenius norm distance from the ground truth weights. In each SGD iteration, Markov parameters are updated and later used to form a linear system of equation, which involves unknown parameters of weight matrices. Extensive numerical tests verify that the performance of the proposed methods is consistent with our theory, and show their superior performance relative to existing state of the art methods.

Finally, Chapter 8 discusses the future directions of this dissertation and possible research opportunities.

Chapter 2

Joint Resource Allocation and Routing for Service Function Chaining with In-Subnetwork Processing

NFV replaces hardware middle-boxes, e.g., firewalls or routers, with more flexible software applications known as virtual network functions [23, 24, 25, 26, 27, 28, 29]. Unlike traditional communication networks where network functions are hosted by specialized hardware, NFV-enabled networks allow VNFs to be hosted by generic high-volume servers which can easily be configured as needed. A virtual network function is an abstract building block which aims to process the network traffic to accomplish a specific task. Each VNF corresponds to a separate network function such as network address translation (NAT), firewalling, intrusion detection, caching and domain name service (DNS). VNFs are instantiated on demand without the installation of new equipment in the network. Each service to a flow consists of a sequence of VNFs in a *specific order*, called a service function chain [30, 31]. The service demand of the flow is satisfied only if all the VNFs in its SFC are received by the flow in the given order [25, 32]. Multiple VNFs can be placed (i.e., instantiated) on a server and each VNF can be shared among several flows. Thus, VNF placement can be formulated as a resource allocation problem [30, 31, 33, 34].

2.1 Prior Work

Network function virtualization is studied widely in Software Defined Networking (SDN) literature, where a centralized controller is responsible for managing VNF instantiation and forwarding the data traffic [23, 24, 32, 35, 36]. For instance, in [23], the authors proposed centralized heuristic algorithms to minimize the total topology construction cost, which includes the costs of link and NFV activation. In the considered model, when a flow enters a server, it receives all its required VNFs before it leaves the server. This assumption may not hold in practice since each server has a finite computational capacity. The approach in [23] consists of generating several feasible solutions using a heuristic algorithm and choosing the one with minimum cost. The drawback of this approach is the lack of scalability for large networks. Similar models and solutions are proposed in [24], where an NP-hard, multi-objective optimization is formulated to achieve a balance between network operation and data center costs, and solved by an exhaustive search-based approach. Instead of heuristic approaches which are slow and not scalable, decentralized and scalable algorithms are preferred to jointly optimize VNF placement and traffic engineering. Virtual Network Embedding (VNE) studied in [24, 25, 26, 27, 37, 38] considers VNF instantiation and traffic steering separately. However, separate VNF instantiation and traffic steering can lead to infeasible solutions for large networks, where links have finite bandwidth and VNF nodes have finite computational capacity.

To minimize the cost of traffic engineering and purchase of servers, a random server selection and many-to-one matching are used in [25] to find a minimum-cardinality subset of servers that are able to serve all flows. Another heuristic algorithm to separately minimize the linear cost of VNF placement and traffic routing is proposed in [39] where a VNF is assumed to be instantiated for each flow at only one node. A brute-force search-based algorithm is developed in [39] to solve the resulting mixed-integer problem.

The problem of multiple service function chain mapping in the presence of highly populated traffic demands is studied in [40]. A shortest path traffic grouping approach is proposed in [40] where numerical tests show that it is able to achieve a near-optimal solution with a relatively small number of SFCs. A genetic algorithm-based VNF placement approach is proposed in [41] for the time-varying demand. Network function instantiation and flow routing are studied in [32] and each required VNF is provided by only one server. The proposed centralized approach in [32] is not scalable and considers the linear routing cost only. Similar to most studies like [23, 24, 35, 36, 39], it is assumed in [32] that links and servers are already installed in the network and available to use.

2.2 Contributions

In this chapter, we consider a large service-oriented network spanning several subnetworks [42, 43]. We assume that each subnetwork belongs to and is operated by a different administrator. Unlike the majority of relevant work [23, 24, 25, 32, 39], we decouple the management of different subnetworks by developing decentralized approaches. Each subnetwork owner wishes to route and serve flow required VNF services by servers and links within its own subnetwork [44, 45]. To the best of our knowledge, this is the first work to decouple the management of neighboring subnetworks. The contributions of this chapter are as follows: 1) We first propose a centralized optimization approach for joint traffic routing and NFV cost minimization when flows are encouraged to be processed in local subnetworks. 2) We decompose the centralized problem and propose a distributed approach based on the proximal ADMM to minimize the routing and VNF placement cost. The distributed optimization can be locally solved while minimal information is shared with a central controller. The proposed distributed approach addresses scalability for large networks and confidentiality, and also decreases the cost of implementation. 3) We propose efficient dual algorithms based on successive upper-bound minimization to solve the subproblems of our distributed approach. 4) We further consider the problem of link capacity and server purchase cost minimization when flows are processed in local subnetworks. Given flow demands, we formulate the problem from a sparse optimization perspective and propose an efficient approach based on iteratively solving a sequence of group LASSO problems. Using purchased link capacities and servers, each subnetwork administrator can jointly optimize VNF placement on servers and traffic engineering. Extensive numerical results are provided to demonstrate the efficiency and performance of the proposed approaches in this chapter and to show that they outperform existing work.

The chapter is organized as follows: The model of the service-oriented network is given in Section 2.3; Section 2.4 presents the optimization formulation for routing and NFV cost minimization with in-subnetwork processing; In Section 2.5, the distributed algorithm is proposed for the problem; Server and link capacity purchase are discussed in Section 2.6; Simulation results are provided in Section 2.7, and concluding remarks are given in Section 2.8.

2.3 System Model

In this section, we formulate the constraints of the VNF placement and flow routing. Suppose that the network is modeled by a directed graph $\mathcal{N} = (\mathcal{V}, \mathcal{L})$ where $\mathcal{V}$ is the set of network nodes and $\mathcal{L}$ is the set of network links. Among all network nodes, a number of them, which are called servers, are capable of providing VNF services to flows. We denote the set of servers by $\mathcal{V}_\psi \subset \mathcal{V}$. Consider K data flows in the network, each of which has a source denoted by $S(k) \in \mathcal{V}, k = 1, 2, \ldots, K$, and a destination shown by $D(k) \in \mathcal{V}$. The rate of flow k is denoted by $\lambda(k)$. Assume that the set of all VNFs is finite. Each server is able to provide a subset of existing VNFs. Each flow requests a sequence of VNFs with a specific order, known as a service function chain [46], which must be instantiated at servers. For an arbitrary flow k, its SFC is denoted by $\mathcal{SC}(k)$. Once a VNF, say s, is instantiated at a server i, the server consumes a number of its pooled computational resources, such as CPU or storage, to process the flow. The set of required resources to provide VNF s is denoted by $\mathcal{Q}_s$. For example, a server which runs a firewall to filter flows uses CPU and hard drive storage. Therefore, these two resources are in $\mathcal{Q}_s$, when s refers to a firewall. The resources of the server i are limited and the capacity of the computational resource f is shown by $\mu_i(f)$. We assume that processing one unit of data flow requires ω_{fs} units of resource f capacity at a server when VNF s is instantiated. Consequently, to process flow k at a server, the server has to have a resource available with at least $\omega_{fs}\,\lambda(k)$ capacity for each required resource $f \in \mathcal{Q}_s$. Flows are allowed to split into multiple sub-flows that traverse different paths. Let $\psi_{i,s}(k)$ be a binary variable to indicate whether or not VNF s is instantiated for flow k at server i. When $\psi_{i,s}(k) = 1$, VNF s is instantiated for flow k at server i. If $\psi_{i,s}(k) = 0$, VNF s is not instantiated for flow k at server i. Therefore, the term $\psi_{i,s}(k)\,\omega_{fs}\,\lambda(k)$ captures the amount of required resource of type f at server i to process flow k when VNF s is provided. If node i is not a server, we set $\psi_{i,s}(k) = 0$. We list the constraints for VNF instantiation and flow routing within the network.

1) *Server Computational Capacity*: As the computational capacity of a server is limited, we have a constraint for each resource f as follows:

$$\sum_{k=1}^{K} \sum_{s \in \mathcal{SC}(k)} \psi_{i,s}(k)\, \omega_{fs}\, \lambda(k) \leq \mu_i(f), \quad \forall f. \tag{2.1}$$

If $f \notin \mathcal{Q}_s$, we set $\omega_{fs} = 0$.

2) *Link Capacity*: The physical capacity of the link between node i and j is denoted by C_{ij}. Then,

$$\sum_{k=1}^{K} r_{ij}(k) \leq C_{ij}, \quad \forall (i,j) \in \mathcal{L}, \tag{2.2}$$

where $r_{ij}(k)$ is the rate of flow k on link (i,j).

3) *VNF Instantiation*: A data flow receives each VNF from exactly one server. Thus, we have

$$\sum_{i \in \mathcal{V}_\psi} \psi_{i,s}(k) = 1, \quad k = 1, \ldots, K, \quad \forall s. \tag{2.3}$$

4) *Flow Conservation*: Assume $\mathcal{SC}(k)$ is equal to $\{s_1, s_2, \ldots, s_{|\mathcal{SC}(k)|}\}$, where s_1 is the first requested VNF by flow k. When the flow leaves its source, it has not received any VNF yet. We label the unprocessed flow originating from $S(k)$ by $r_{ij}(k, s_0)$ while it passes on link (i,j). Each flow leaves its source via several links. The flow conservation for the source of flow k is as follows:

$$\sum_{j:(S(k),j) \in \mathcal{L}} r_{S(k)j}(k, s_0) = \lambda(k), \quad k = 1, \ldots, K. \tag{2.4}$$

Flow labeling is used to certify that flow k receives its requested VNFs according to $\mathcal{SC}(k)$. After receiving VNF $s \in \mathcal{SC}(k)$, the flow passing on link (i,j) is represented by $r_{ij}(k,s)$. Since each flow receives VNFs in a predefined order specified in its SFC, the flow $r_{ij}(k,s)$

has received VNFs in $\mathcal{SC}(k)$ up to s. From definitions of $r_{ij}(k)$ and $r_{ij}(k,s)$, we have

$$r_{ij}(k) = \sum_{s \in \mathcal{SC}(k) \bigcup \{s_0\}} r_{ij}(k,s), \quad \forall k, \, \forall (i,j) \in \mathcal{L}. \tag{2.5}$$

Before arriving at its destination, flow k should have already received the last VNF in $\mathcal{SC}(k)$, which is denoted by $s_{|\mathcal{SC}(k)|}$. Thus, the flow conservation for the destination of flow k is as follows:

$$\sum_{j:(j,D(k)) \in \mathcal{L}} r_{jD(k)}(k, s_{|\mathcal{SC}(k)|}) = \lambda(k), \quad k = 1, \ldots, K. \tag{2.6}$$

Constraints (2.4) and (2.6) ensure that a flow which leaves its source is unprocessed and the arriving flow at its destination has received all its needed VNFs, respectively. Flow conservation for each intermediate node is

$$\psi_{i,s_g}(k) \, \lambda(k) = \sum_{j:(j,i) \in \mathcal{L}} r_{ji}(k, s_{g-1}) - \sum_{j:(i,j) \in \mathcal{L}} r_{ij}(k, s_{g-1}), \tag{2.7}$$

$$\psi_{i,s_g}(k) \, \lambda(k) = \sum_{j:(i,j) \in \mathcal{L}} r_{ij}(k, s_g) - \sum_{j:(j,i) \in \mathcal{L}} r_{ji}(k, s_g). \tag{2.8}$$

When a server does not offer VNF s_g to flow k, i.e., $\psi_{i,s_g}(k) = 0$, the total incoming flow $\sum_{j:(j,i) \in \mathcal{L}} r_{ji}(k, s_{g-1})$ to node i is equal to the total outgoing flow $\sum_{j:(i,j) \in \mathcal{L}} r_{ij}(k, s_{g-1})$. When server i provides VNF s_g to flow k, we have $\psi_{i,s_g} = 1$, and the LHS of (2.7) will be $\lambda(k)$. On the other hand, in this case, all the sub-flows of flow k should enter server i before receiving VNF s_g and exit it after receiving VNF s_g, that is $\sum_{j:(j,i) \in \mathcal{L}} r_{ji}(k, s_{g-1}) = \lambda(k)$ and $\sum_{j:(i,j) \in \mathcal{L}} r_{ij}(k, s_{g-1}) = 0$. When server i provides VNF s_g to flow k, the LHS of (2.8) will also be $\lambda(k)$. All the sub-flows of flow k should exit it after receiving VNF s_g, that is $\sum_{j:(i,j) \in \mathcal{L}} r_{ij}(k, s_g) = \lambda(k)$, and since flow k has not received VNF s_g prior to entering node i, we have $\sum_{j:(j,i) \in \mathcal{L}} r_{ji}(k, s_g) = 0$.

2.4 Problem Formulation

Consider a service-oriented network spanning multiple subnetworks (Fig. 2.6 is an example with four subnetworks). The set of nodes and links within subnetwork $\mathcal{N}^m = (\mathcal{V}^m, \mathcal{L}^m)$ are denoted by $\mathcal{V}^m$ and $\mathcal{L}^m$, respectively. The set of servers within the subnetwork $\mathcal{N}^m$ is represented by $\mathcal{V}_\psi^m$. In this section, we formulate the joint problem of VNF placement and flow routing with local subnetwork processing. Suppose that the source of flow k belongs to the subnetwork $\mathcal{N}^m$, i.e., $S(k) \in \mathcal{V}^m$. Then, the amount of VNF provided to flow k by servers out of the subnetwork $\mathcal{N}^m$ is $\sum_{i \notin \mathcal{V}_\psi^m, i \in \mathcal{V}_\psi} \sum_{s \in \mathcal{SC}(k)} \psi_{i,s}(k)$. Let us define $\mathbf{r} = \{\{r_{ij}(k)\}_{(i,j)\in\mathcal{L}}, \{r_{ij}(k,s)\}_{(i,j)\in\mathcal{L}, s\in\mathcal{SC}(k)\bigcup\{s_0\}}\}_{k=1}^K$ and $\boldsymbol{\psi} = \{\{\psi_{i,s}(k)\}_{i\in\mathcal{V}_\psi, s\in\mathcal{SC}(k)}\}_{k=1}^K$. In light of the above arguments, we introduce two penalties as follows:

1. Outside subnetwork processing penalty: Consider a penalty term defined as

$$P_1 \sum_{m=1}^{M} \sum_{k:S(k)\in\mathcal{V}^m} \sum_{i\notin\mathcal{V}_\psi^m, i\in\mathcal{V}_\psi} \sum_{s\in\mathcal{SC}(k)} \psi_{i,s}(k),$$

   where P_1 is a large number. This penalty *promotes the use of servers within the subnetwork from which the flow has originated.*

2. Outside subnetwork routing penalty: Consider

$$P_2 \sum_{m=1}^{M} \sum_{k:S(k)\notin\mathcal{V}^m} \sum_{(i,j)\in\mathcal{L}^m} r_{ij}(k),$$

   where P_2 is a large number. The role of this penalty term is to *discourage the flows from entering neighboring subnetworks.*

We plan to minimize some convex function of the network flow variables, say $\varphi_{ij}(\cdot)$, plus some convex function of the flow process variables, say $\iota_i(\cdot)$, subject to the set of constraints (2.1), (2.2), (2.3), (2.4), (2.5), (2.6), (2.7), and (2.8). For instance, end-to-end delay is one of the most common cost functions. There are several delay functions proposed for data networks in [47, 48, 49]. We *minimize the routing and VNF instantiation costs while keeping the flow routing and processing locally within subnetworks to the extent possible.* To promote in-subnetwork processing, we use the penalties described above. The objective function that includes penalties is

$$\begin{aligned} T_1\left(\mathbf{r}, \boldsymbol{\psi}\right) = & \sum_{i \in \mathcal{V}_\psi} \iota_i(\boldsymbol{\psi}_i) + \sum_{(i,j)\in\mathcal{L}} \varphi_{ij}\left(\mathbf{r}_{ij}\right) \\ & + P_1 \sum_{m=1}^{M} \sum_{k:S(k)\in\mathcal{V}^m} \sum_{i\notin\mathcal{V}_\psi^m, i\in\mathcal{V}_\psi} \sum_{s\in\mathcal{SC}(k)} \psi_{i,s}(k) + P_2 \sum_{m=1}^{M} \sum_{k:S(k)\notin\mathcal{V}^m} \sum_{(i,j)\in\mathcal{L}^m} r_{ij}(k), \end{aligned}$$

where $\boldsymbol{\psi}_i = \{\psi_{i,s}(k)\}_{s\in\mathcal{SC}(k),k=1:K}$ and $\mathbf{r}_{ij} = \{r_{ij}(k)\}_{k=1}^K$. The optimization formulation is

$$\begin{aligned} \min_{\mathbf{r},\boldsymbol{\psi}} \ & T_1\left(\mathbf{r}, \boldsymbol{\psi}\right) \\ \text{s.t.} \ & (2.1), (2.2), (2.3), (2.4), (2.5), (2.6), (2.7), (2.8), \\ & r_{ij}(k) \geq 0, r_{ij}(k,s) \geq 0, \ \forall\, s \in \mathcal{SC}(k), \forall\, k, \forall\, (i,j) \in \mathcal{L}, \\ & \psi_{i,s}(k) \in \{0,1\}, \ \forall\, i \in \mathcal{V}_\psi, \forall\, s \in \mathcal{SC}(k), \forall\, k. \end{aligned} \tag{2.9}$$

The above problem is a mixed-binary problem which is difficult to solve to global optimality.

2.5 VNF Placement and Routing with In-Subnetwork Processing

Here, we propose efficient algorithms to solve (2.9) for the joint VNF placement and routing.

2.5.1 Centralized VNF Placement and Routing

To approximately solve the problem, we use the Penalized Upper-bound Minimization (PSUM) approach proposed in [50, 51, 32], where the binary variable $\psi_{i,s}(k)$ is relaxed to $0 \leq \psi_{i,s}(k) \leq 1$ and a concave penalty is added to the objective of the optimization problem to promote a binary solution. Suppose that the vector $\boldsymbol{\psi}_s(k)$ is defined as $\boldsymbol{\psi}_s(k) = \{\psi_{i,s}(k)\}_{i\in\mathcal{V}_\psi}$. *Ideally, we want this vector to have only one 1, and be zero at all other positions.* Using this vector, the suggested penalty to promote binary solutions is

$$P_\epsilon(\boldsymbol{\psi}) = \sum_{k=1}^{K} \sum_{s\in\mathcal{SC}(k)} \left\|\boldsymbol{\psi}_s(k) + \epsilon\mathbf{1}\right\|_p^p,$$

where $0 < p < 1$ and ϵ is a positive constant to make the objective function differentiable. When $P_\epsilon(\boldsymbol{\psi})$ is minimized, the optimal solution is binary [50, 51]. Thus, adding $P_\epsilon(\boldsymbol{\psi})$ with appropriate weight to the objective function $T_1(\mathbf{r}, \boldsymbol{\psi})$ promotes a binary solution. Since $P_\epsilon(\boldsymbol{\psi})$ is concave, we linearize this penalty and find an upper-bound as $P_\epsilon(\boldsymbol{\psi}) \leq P_\epsilon(\boldsymbol{\psi}^z) + \nabla P_\epsilon(\boldsymbol{\psi}^z)^T (\boldsymbol{\psi} - \boldsymbol{\psi}^z)$, which will be added to the cost function in (2.9). The added penalty to the objective in $(z+1)^{\text{th}}$ iteration of solving the optimization given in (2.9) is $\sigma^{z+1} \nabla P_\epsilon(\boldsymbol{\psi}^z)^T \boldsymbol{\psi}$, where $0 < \sigma^z \leq \sigma^{z+1}$. Since this linear upper-bound satisfies all four convergence conditions in [19, Assumption 2], the successive upper-bound minimization algorithm converges to a stationery solution of the relaxed problem with $P_\epsilon(\boldsymbol{\psi})$ in the objective function [19, Theorem 2]. We rewrite the linear upper-bound as $\sum_{i\in\mathcal{V}_\psi} \sum_{k=1}^{K} \sum_{s\in\mathcal{SC}(k)} u_{i,s,k}^z \, \psi_{i,s}(k)$, where $u_{i,s,k}^z = \sigma^z \, p \, (\psi_{i,s}^{z-1}(k) + \epsilon)^{p-1}$.

We further tighten the relaxation of (2.1) by using the fact the binary variable $\psi_{i,s}(k)$ is relaxed to $0 \leq \psi_{i,s}(k) \leq 1$. To do so, new variables $\psi_{i,s}$ and ψ_i are added to the optimization problem (2.9) together with the following constraints

$$\psi_{i,s}(k) \leq \psi_{i,s} \leq \psi_i, \qquad \forall k, \forall s \in \mathcal{SC}(k), \forall i \in \mathcal{V}_\psi, \tag{2.10}$$

$$\sum_{k=1}^{K} \psi_{i,s}(k) \, \omega_{fs} \, \lambda(k) \leq \psi_{i,s} \;\; \mu_i(f), \qquad \forall f \in \mathcal{Q}_s, \forall s, \forall i \in \mathcal{V}_\psi, \tag{2.11}$$

$$\sum_{k=1}^{K} \sum_{s\in\mathcal{SC}(k)} \sum_{f\in\mathcal{Q}_s} \psi_{i,s}(k) \, \omega_{fs} \, \lambda(k) \leq \psi_i \sum_{f} \mu_i(f), \qquad \forall i \in \mathcal{V}_\psi, \tag{2.12}$$

$$0 \leq \psi_{i,s} \leq 1, 0 \leq \psi_i \leq 1, \qquad \forall i \in \mathcal{V}_\psi. \tag{2.13}$$

Then, the overall objective function becomes

$$\begin{aligned} T_2(\mathbf{r}, \boldsymbol{\psi}) = &\sum_{i\in\mathcal{V}_\psi} \iota_i(\boldsymbol{\psi}_i) + \sum_{(i,j)\in\mathcal{L}} \varphi_{ij}(\mathbf{r}_{ij}) + \sum_{i\in\mathcal{V}_\psi} \sum_{k=1}^{K} \sum_{s\in\mathcal{SC}(k)} u_{i,s,k}^z \, \psi_{i,s}(k) \\ &+ P_1 \sum_{m=1}^{M} \sum_{k:S(k)\in\mathcal{V}^m} \sum_{i\notin\mathcal{V}_\psi^m, i\in\mathcal{V}_\psi} \sum_{s\in\mathcal{SC}(k)} \psi_{i,s}(k) + P_2 \sum_{m=1}^{M} \sum_{k:S(k)\notin\mathcal{V}^m} \sum_{(i,j)\in\mathcal{L}^m} r_{ij}(k). \end{aligned}$$

```
Algorithm 1: In-subnetwork VNF placement and routing algorithm
  0. Initialization Set $\boldsymbol{\psi} = \mathbf{0}$, $z = 1$;
  repeat
      1. Solve the optimization problem (2.14);
      2. Update $u_{i,s,k}^{z}$ in $T_2\left(\mathbf{r}, \boldsymbol{\psi}\right)$ using $u_{i,s,k}^{z} = \sigma^z\, p\, (\psi_{i,s}^{z-1}(k) + \epsilon)^{p-1}$;
      3. $z = z + 1$;
  until Each $\psi_{i,s}(k)$ is binary;
```

The new optimization problem after relaxation of $\psi_{i,s}(k)$ variables will be

$$\min_{\mathbf{r}, \boldsymbol{\psi}, \{\psi_{i,s}\}_{s, i \in \mathcal{V}_\psi}, \{\psi_i\}_{i \in \mathcal{V}_\psi}} \quad T_2\left(\mathbf{r}, \boldsymbol{\psi}\right) \tag{2.14}$$
$$\text{s.t.} \;\; (2.1), (2.2), (2.3), (2.4), (2.5), (2.6), (2.7), (2.8), (2.10), (2.11), (2.12), (2.13),$$
$$r_{ij}(k) \geq 0, r_{ij}(k,s) \geq 0, \;\forall\, s \in \mathcal{SC}(k), \forall\, k, \forall\, (i,j) \in \mathcal{L},$$
$$\psi_{i,s}(k) \in [0,1], \;\forall\, i \in \mathcal{V}_\psi, \forall\, s \in \mathcal{SC}(k), \forall\, k.$$

We solve (2.14) iteratively and each time we update $u_{i,s,k}^{z}$ in $T_2\left(\mathbf{r}, \boldsymbol{\psi}\right)$ and $z = z + 1$. The termination criteria is that all $\psi_{i,s}(k)$ are either 0 or 1, or the maximum iteration for the PSUM is reached. In the case that the optimized variable $\psi_{i,s}(k)$ after termination is not binary, we map $\psi_{i,s}(k)$ to 0 or 1, whichever value is closer to $\psi_{i,s}(k)$. A central controller collects different information sets of subnetworks and solves one large problem. The summary of the proposed approach is given in Algorithm 1.

Proposition 1. *Suppose that* $(\boldsymbol{r}^z, \boldsymbol{\psi}^z)$ *is the global minimizer of* (2.14)*. Then, every limit point* $(\boldsymbol{r}^z, \boldsymbol{\psi}^z)$ *of the iterates generated by Algorithm 1 is a global minimum of* (2.9) *when* $0 < \sigma^z \leq \sigma^{z+1}$ *and* $\sigma^z \to \infty$*.*

Proof. Assume that $\{\mathbf{r}^z, \mathbf{t}^z\}$ is the global minimizer of $T_1\left(\mathbf{r}^z, \boldsymbol{\psi}^z\right) + \sigma^z\, P_\epsilon(\boldsymbol{\psi}^z)$. Then, we have

$$T_1\left(\mathbf{r}^z, \boldsymbol{\psi}^z\right) + \sigma^z\, P_\epsilon(\boldsymbol{\psi}^z) \leq T_1\left(\mathbf{r}^{z+1}, \boldsymbol{\psi}^{z+1}\right) + \sigma^z\, P_\epsilon(\boldsymbol{\psi}^{z+1}), \tag{2.15a}$$
$$T_1\left(\mathbf{r}^{z+1}, \boldsymbol{\psi}^{z+1}\right) + \sigma^{z+1}\, P_\epsilon(\boldsymbol{\psi}^{z+1}) \leq T_1\left(\mathbf{r}^z, \boldsymbol{\psi}^z\right) + \sigma^{z+1}\, P_\epsilon(\boldsymbol{\psi}^z). \tag{2.15b}$$

We use the above two inequalities to find

$$\sigma^z \left(P_\epsilon(\boldsymbol{\psi}^z) - P_\epsilon(\boldsymbol{\psi}^{z+1})\right) \leq T_1\left(\mathbf{r}^{z+1}, \boldsymbol{\psi}^{z+1}\right) - T_1\left(\mathbf{r}^z, \boldsymbol{\psi}^z\right) \leq \sigma^{z+1}\left(P_\epsilon(\boldsymbol{\psi}^z) - P_\epsilon(\boldsymbol{\psi}^{z+1})\right).$$

Since $\sigma^{z+1} \geq \sigma^z$, we understand that $P_\epsilon(\boldsymbol{\psi}^z) \geq P_\epsilon(\boldsymbol{\psi}^{z+1})$ and $T_1\left(\mathbf{r}^{z+1}, \boldsymbol{\psi}^{z+1}\right) \geq T_1\left(\mathbf{r}^z, \boldsymbol{\psi}^z\right)$. Furthermore, we always have

$$T_1\left(\mathbf{r}^z, \boldsymbol{\psi}^z\right) + \sigma^z\, P_\epsilon(\boldsymbol{\psi}^z) \leq T_1\left(\mathbf{r}^*, \boldsymbol{\psi}^*\right) + \sigma^z\, P_\epsilon(\boldsymbol{\psi}^*).$$

The reason is that $\boldsymbol{\psi}^z$ belongs to a larger feasible set in which $\psi^z_{i,s}(k) \in [0,1]$. However, for variables in $\boldsymbol{\psi}^*$, the feasible set is built from $\psi^*_{i,s}(k) \in \{0,1\}$. We rewrite the above inequality as follows:

$$T_1\left(\mathbf{r}^z, \boldsymbol{\psi}^z\right) + \sigma^z\left(P_\epsilon(\boldsymbol{\psi}^z) - P_\epsilon(\boldsymbol{\psi}^*)\right) \leq T_1\left(\mathbf{r}^*, \boldsymbol{\psi}^*\right).$$

Since σ^z is an increasing sequence and $\sigma^z \to \infty$, we should have $\lim\limits_{z\to\infty} P_\epsilon(\boldsymbol{\psi}^z) = P_\epsilon(\boldsymbol{\psi}^*)$. Furthermore, as $T_1\left(\mathbf{r}^z, \boldsymbol{\psi}^z\right)$ is an increasing sequence, we have $\lim\limits_{z\to\infty} T_1\left(\mathbf{r}^z, \boldsymbol{\psi}^z\right) = T_1\left(\mathbf{r}^*, \boldsymbol{\psi}^*\right)$. □

2.5.2 Distributed VNF Placement and Routing

In some cases, it is easier to solve a sequence of smaller and scalable problems instead of one big problem like (2.14). We can decompose the coupled optimization problem given in (2.14) into $M+1$ smaller subproblems. Solving the problem in (2.14) distributedly is specifically critical when subnetworks belong to different administrators. Due to security and financial concerns, each administrator prefers to route and process flows using its own facilities [44, 45]. Information, such as the computational capacity of each server and the cost of flow processing, may be confidential for subnetwork owners and subnetwork owners do not share sensitive information. Due to temporal congestion, equipment failures, etc., one administrator may ask to use other administrators' facilities. In this case, a coordination between administrators is needed to satisfy flow demands [52, 53].

Each subnetwork $\mathcal{N}^m$ includes cross-links, denoted by $\mathcal{L}^0_m$, which connect the subnetwork $\mathcal{N}^m$ to its neighboring subnetworks. Then, the set of all cross-links can be written

Table 2.1: Notations

$\mathcal{L}$	set of links
$\mathcal{V}_\psi$	set of servers
$\lambda(k)$	rate of flow k
K	number of flows in the network
$S(k)$	source of flow k
$D(k)$	destination of flow k
$r_{ij}(k)$	rate of flow k passing on link (i, j)
$\mu_i(f)$	capacity of server i for resource f
s	a VNF
$\mathcal{V}_\psi^m$	set of servers in subnetwork $\mathcal{N}^m$
$\psi_{i,s}(k)$	indicator whether s is instantiated for flow k at server i
$\mathcal{SC}(k)$	service function chain of flow k
$r_{ij}(k, s)$	flow k after receiving VNFs up to s
$\mathcal{L}^0$	set of cross-links in the network
$\mathcal{L}_m^0$	set of cross-links of subnetwork $\mathcal{N}^m$
$\mathcal{L}^m$	set of links of subnetwork $\mathcal{N}^m$
C_{ij}	capacity of link (i, j)
$\mathcal{N}^m$	the subnetwork m

as $\mathcal{L}^0 = \bigcup_{m=1}^{M} \mathcal{L}_m^0$. For a flow k, if $S(k)$ and $D(k)$ belong to the same subnetwork, it is desired to provide VNFs for flow k locally within that subnetwork. We reformulate the optimization problem (2.14) to enable decomposition and a distributed solution so that passing flows on cross-links between subnetworks are optimized independently within subnetworks. The distributed approach enables the cooperation of subnetworks to serve flows when there is a resource deficiency in one subnetwork. To ensure the final optimized flows on cross-links between the neighboring subnetworks are identical, a central controller in the network coordinates the computation in different subnetworks. In particular, the central controller keeps copies of flow process variables as well as variables corresponding to flows on cross-links.

Suppose that optimized variables in the central controller are superscripted by 0. Likewise, the optimization variables kept in the subnetwork $\mathcal{N}^m$ are superscripted by m (see Table 2.1). Thus, we have the following equality constraints

$$r_{ij}^0(k, s) = r_{ij}^m(k, s), \qquad \forall (i, j) \in \mathcal{L}_m^0, \forall m, \tag{2.16a}$$

$$\psi^0_{i,s}(k) = \psi^m_{i,s}(k), \psi^0_{i,s} = \psi^m_{i,s}, \psi^0_i = \psi^m_i, \qquad \forall i \in \mathcal{V}^m_\psi, \forall m, \tag{2.16b}$$

Consider the optimization problem in subnetwork $\mathcal{N}^m$. The constraints (2.1), (2.2), (2.4), (2.5), (2.6), (2.7), (2.8), (2.10), (2.11), (2.12), and (2.13), form a decomposable feasible set across subnetworks, where each decomposed set is denoted by $\mathcal{C}^m$. Link capacity constraints of cross-links are kept in feasible sets of both neighboring subnetworks. Suppose that $\mathcal{C}^L = \bigcup_{m=1}^{M} \mathcal{C}^m$. The feasible set of optimization variables kept in the central controller is

$$\begin{aligned}\mathcal{C}^0 = \Big\{ r^0_{ij}(k,s), \psi^0_i, \psi^0_{i,s}, \psi^0_{i,s}(k) \mid & \sum_{i \in \mathcal{V}_\psi} \psi^0_{i,s}(k) = 1, \\ & \sum_{k=1}^{K} \sum_{s \in \mathcal{SC}(k) \bigcup \{s_0\}} r^0_{ij}(k,s) \le C_{ij}, \forall (i,j) \in \mathcal{L}^0, \\ & r^0_{ij}(k,s) \ge 0, 1 \ge \psi^0_{i,s}(k) \ge 0, 1 \ge \psi^0_{i,s} \ge 0, 1 \ge \psi^0_i \ge 0 \Big\}.\end{aligned} \tag{2.17}$$

Consider all variables corresponding to flows passing on inner links of subnetwork $\mathcal{N}^m$ are represented by the vector $\mathbf{r}^m = \{r^m_{ij}(k,s)\}_{s \in \mathcal{SC}(k) \bigcup \{s_0\}, k=1:K, (i,j) \in \mathcal{L}^m \setminus \mathcal{L}^0_m}$. Furthermore, all cross-link flow variables of subnetwork $\mathcal{N}^m$ are denoted by the vector $\mathbf{r}^m_0$. The variables corresponding to flows on cross-links of subnetwork $\mathcal{N}^m$ kept in the central controller are denoted by $\mathbf{r}^0_m$. All $\psi^m_{i,s}(k), \psi^m_{i,s}$, and ψ^m_i variables which belong to the subnetwork $\mathcal{N}^m$ are represented by the vector $\boldsymbol{\psi}^m$. The set of copied variables $\psi^0_{i,s}(k), \psi^0_{i,s}$, and ψ^0_i in the central controller for subnetwork $\mathcal{N}^m$ is $\boldsymbol{\psi}^0_m$. The central controller coordinates flows passing on cross-links. The central controller is responsible for minimizing the following term:

$$T^0\big(\{\mathbf{r}^0_m\}_{m=1}^M\big) = \frac{1}{2} \sum_{(i,j) \in \mathcal{L}^0} \varphi_{ij} \left(\left\{ \sum_{s \in \mathcal{SC}(k) \bigcup \{s_0\}} r^0_{ij}(k,s) \right\}_{k=1}^{K} \right). \tag{2.18}$$

Moreover, subnetwork $\mathcal{N}^m$ keeps those terms of the objective, $T_2\left(\mathbf{r}, \boldsymbol{\psi}\right)$, which are functions of $\mathbf{r}^m$, $\mathbf{r}_0^m$ or $\boldsymbol{\psi}^m$. The objective function of the problem in $\mathcal{N}^m$ is

$$\begin{aligned}
T^m\left(\mathbf{r}^m, \mathbf{r}_0^m, \boldsymbol{\psi}^m\right) &= \sum_{(i,j)\in\mathcal{L}^m\setminus\mathcal{L}_m^0} \varphi_{ij}\left(\left\{\sum_{s\in\mathcal{SC}(k)\bigcup\{s_0\}} r_{ij}^m(k,s)\right\}_{k=1}^{K}\right)\\
&+ \frac{1}{4}\sum_{(i,j)\in\mathcal{L}_m^0} \varphi_{ij}\left(\left\{\sum_{s\in\mathcal{SC}(k)\bigcup\{s_0\}} r_{ij}^m(k,s)\right\}_{k=1}^{K}\right) + \sum_{i\in\mathcal{V}_\psi^m}\sum_{k=1}^{K}\sum_{s\in\mathcal{SC}(k)} u_{i,s,k}^z\, \psi_{i,s}^m(k)\\
&+ \sum_{i\in\mathcal{V}_\psi^m} \iota_i(\boldsymbol{\psi}_i) + \sum_{k:S(k)\notin\mathcal{V}^m}\Big(P_1 \sum_{i\in\mathcal{V}_\psi^m}\sum_{s\in\mathcal{SC}(k)} \psi_{i,s}^m(k) + P_2 \sum_{(i,j)\in\mathcal{L}^m}\sum_{s\in\mathcal{SC}(k)\bigcup\{s_0\}} r_{ij}^m(k,s)\Big).
\end{aligned}$$

Then, $T_2\left(\mathbf{r}, \boldsymbol{\psi}\right) = T^0\left(\{\mathbf{r}_m^0\}_{m=1}^M\right) + \sum_{m=1}^M T^m\left(\mathbf{r}^m, \mathbf{r}_0^m, \boldsymbol{\psi}^m\right)$. In order to coordinate the required message passing between the central controller and the local subnetworks to jointly carry out VNF placement and flow routing, we use the proximal ADMM algorithm [54, Chapter 7] which is an iterative algorithm. The problem to be solved is

$$\begin{aligned}
&\min_{\mathbf{r}_m^0, \boldsymbol{\psi}_m^0, \mathbf{r}^m, \mathbf{r}_0^m, \boldsymbol{\psi}^m} && T^0\left(\{\mathbf{r}_m^0\}_{m=1}^M\right) + \sum_{m=1}^{M} T^m\left(\mathbf{r}^m, \mathbf{r}_0^m, \boldsymbol{\psi}^m\right) && && (2.19a)\\
&\text{s.t.} && r_{ij}^0(k,s), \psi_{i,s}^0(k), \psi_{i,s}^0, \psi_i^0 \in \mathcal{C}^0, && \forall\,(i,j)\in\mathcal{L}^0, \forall i\in\mathcal{V}_\psi, && (2.19b)\\
& && r_{ij}^m(k,s) \in \mathcal{C}^m, && \forall\,(i,j)\in\mathcal{L}^m, && (2.19c)\\
& && \psi_{i,s}^m(k), \psi_{i,s}^m, \psi_i^m \in \mathcal{C}^m, && \forall i\in\mathcal{V}_\psi^m, && (2.19d)\\
& && r_{ij}^0(k,s) = r_{ij}^m(k,s), && \forall (i,j)\in\mathcal{L}^0, \forall m, && (2.19e)\\
& && \psi_{i,s}^0(k) = \psi_{i,s}^m(k), \psi_{i,s}^0 = \psi_{i,s}^m, \psi_i^0 = \psi_i^m, && \forall i\in\mathcal{V}_\psi^m. && (2.19f)
\end{aligned}$$

Constraints (2.19e)-(2.19f) couple the variables from two different feasible sets, $\mathcal{C}^0$ and $\mathcal{C}^L$. We write the augmented Lagrangian function ([4, eq. (22)]) as follows:

$$\begin{aligned}
L(\cdot) &= T^0\left(\{\mathbf{r}_m^0\}_{m=1}^M\right) + \sum_{m=1}^{M} T^m\left(\mathbf{r}^m, \mathbf{r}_0^m, \boldsymbol{\psi}^m\right) + \sum_{m=1}^{M} \frac{\rho}{2}\left\|\mathbf{r}_m^0 - \mathbf{r}_0^m\right\|_F^2 + \sum_{m=1}^{M} \frac{\rho}{2}\left\|\boldsymbol{\psi}_m^0 - \boldsymbol{\psi}^m\right\|_F^2\\
&+ \sum_{m=1}^{M}\sum_{(i,j)\in\mathcal{L}_m^0}\sum_{k=1}^{K}\sum_{s\in\mathcal{SC}(k)\bigcup\{s_0\}} \pi_{ij}^m(k,s)\left(r_{ij}^0(k,s) - r_{ij}^m(k,s)\right) + \sum_{m=1}^{M}\sum_{i\in\mathcal{V}_\psi^m} \theta_i^m\left(\psi_i^0 - \psi_i^m\right)
\end{aligned}$$

$$+\sum_{m=1}^{M}\sum_{i\in\mathcal{V}_{\psi}^{m}}\sum_{s}\theta_{i,s}^{m}\left(\psi_{i,s}^{0}-\psi_{i,s}^{m}\right)+\sum_{m=1}^{M}\sum_{i\in\mathcal{V}_{\psi}^{m}}\sum_{k=1}^{K}\sum_{s\in\mathcal{SC}(k)}\theta_{i,s}^{m}(k)\left(\psi_{i,s}^{0}(k)-\psi_{i,s}^{m}(k)\right),\quad(2.20)$$

where ρ is the penalty parameter. Moreover, $\boldsymbol{\pi}^m = \{\pi_{ij}^m(k,s)\}_{(i,j)\in\mathcal{L}_m^0, s\in\mathcal{SC}(k)\bigcup s_0, k=1:K}$ and $\boldsymbol{\theta}^m = \{\theta_{i,s}^m(k), \theta_{i,s}^m, \theta_i^m\}_{i\in\mathcal{V}_\psi^m, s\in\mathcal{SC}(k), k=1:K}$ are the Lagrange multipliers assigned to (2.19e) and (2.19f), respectively. After coupling constraints (2.19e)-(2.19f) are dualized and moved to the Lagrangian, the above optimization problem becomes separable by different subnetworks and the central controller (see (2.20)). The optimization problems in subnetworks are solved in parallel to global optimality. In order to guarantee that a unique minima exists, we add regularization terms to the augmented Lagrangian. Suppose that $\mathbf{r}^{m,t}$, $\mathbf{r}_0^{m,t}$ and $\boldsymbol{\psi}^{m,t}$ are the optimized variables in the t^{th} iteration of the proximal ADMM. Suppose that $\mathbf{J}^{m,t}$ is defined as

$$\mathbf{J}^{m,t} = \left[\begin{array}{cc} [\mathbf{r}_0^{m,t}]^T & [\boldsymbol{\psi}^{m,t}]^T \end{array}\right]^T. \quad (2.21)$$

Then, the regularization term is $\left\|\mathbf{J}^m - \mathbf{J}^{m,t}\right\|_F^2$.

Optimization Problem in Subnetwork $\mathcal{N}^m$

We keep those terms in the augmented Lagrangian which include $\mathbf{r}^m$, $\mathbf{r}_0^m$ and $\boldsymbol{\psi}^m$ in $L^m\left(\mathbf{r}^m, \mathbf{r}_0^m, \boldsymbol{\psi}^m, \boldsymbol{\pi}^m, \boldsymbol{\theta}^m\right)$ as follows:

$$\begin{aligned}
L^m\left(\mathbf{r}^m, \mathbf{r}_0^m, \boldsymbol{\psi}^m, \boldsymbol{\pi}^m, \boldsymbol{\theta}^m\right) = \; & T^m\left(\mathbf{r}^m, \mathbf{r}_0^m, \boldsymbol{\psi}^m\right) \\
+ & \sum_{(i,j)\in\mathcal{L}_m^0}\sum_{k=1}^{K}\sum_{s\in\mathcal{SC}(k)\bigcup\{s_0\}}\pi_{ij}^m(k,s)\left(r_{ij}^0(k,s)-r_{ij}^m(k,s)\right) \\
+ & \sum_{i\in\mathcal{V}_\psi^m}\theta_i^m\left(\psi_i^0-\psi_i^m\right)+\sum_{i\in\mathcal{V}_\psi^m}\sum_{s}\theta_{i,s}^m\left(\psi_{i,s}^0-\psi_{i,s}^m\right) \\
+ & \frac{\rho}{2}\left\|\mathbf{r}_m^0-\mathbf{r}_0^m\right\|_F^2+\frac{\rho}{2}\left\|\boldsymbol{\psi}_m^0-\boldsymbol{\psi}^m\right\|_F^2+\sum_{i\in\mathcal{V}_\psi^m}\sum_{k=1}^{K}\sum_{s\in\mathcal{SC}(k)}\theta_{i,s}^m(k)\left(\psi_{i,s}^0(k)-\psi_{i,s}^m(k)\right). \quad (2.22)
\end{aligned}$$

Then, the optimization problem in subnetwork $\mathcal{N}^m$ is

$$\min_{\mathbf{r}^m,\mathbf{r}_0^m,\boldsymbol{\psi}^m} \quad L^m\left(\mathbf{r}^m,\mathbf{r}_0^m,\boldsymbol{\psi}^m,\boldsymbol{\pi}^m,\boldsymbol{\theta}^m\right)+\frac{\phi}{2}\left\|\mathbf{J}^m-\mathbf{J}^{m,t}\right\|_F^2 \tag{2.23a}$$

$$\text{s.t.} \quad r_{ij}^m(k,s)\in\mathcal{C}^m, \psi_{i,s}^m(k),\psi_{i,s}^m,\psi_i^m\in\mathcal{C}^m, \tag{2.23b}$$

where ϕ is a positive quantity and $\frac{\phi}{2}\left\|\mathbf{J}^m-\mathbf{J}^{m,t}\right\|_F^2$ is the proximal term.

Optimization Problem at the Central Controller

We keep those terms in the augmented Lagrangian which include $\mathbf{r}^0=\{\mathbf{r}_m^0\}_{m=1}^M$ and $\boldsymbol{\psi}^0=\{\boldsymbol{\psi}_m^0\}_{m=1}^M$ in $L^0\left(\{\mathbf{r}_m^0\}_{m=1}^M,\{\boldsymbol{\psi}_m^0\}_{m=1}^M,\{\boldsymbol{\pi}^m\}_{m=1}^M,\{\boldsymbol{\theta}^m\}_{m=1}^M\right)$. The optimization at the central controller is

$$\min_{\mathbf{r}_m^0,\boldsymbol{\psi}_m^0} \quad L^0\left(\{\mathbf{r}_m^0\}_{m=1}^M,\{\boldsymbol{\psi}_m^0\}_{m=1}^M,\{\boldsymbol{\pi}^m\}_{m=1}^M,\{\boldsymbol{\theta}^m\}_{m=1}^M\right) \tag{2.24a}$$

$$\text{s.t.} \quad (2.3), r_{ij}^0(k,s),\psi_{i,s}^0(k),\psi_{i,s}^0,\psi_i^0\in\mathcal{C}^0. \tag{2.24b}$$

We note that some terms of $L(\cdot)$ exist both in $L^0(\cdot)$ and $L^m(\cdot)$. We assume that the objective function (2.24a) is separable by $r_{ij}^0(k,s)$. If $\varphi_{ij}(\cdot)$ is not separable, we use the quadratic separable upper-bound given in [55, eq. (12)] and successively minimize the upper-bound [19, 55]. We see that the central controller requires no sensitive information like the sever processing capacities, resource costs, or network topology to solve (2.24a)-(2.24b). In the following two examples, with both delay being the network cost, we show how (2.24a)-(2.24b) can be solved for separable and non-separable objective functions.

In the first example, we assume that delay of each link is a linear function of the flow rate, similar to [7, 56, 57, 58]. Thus, $\varphi_{ij}(\{\sum_{s\in\mathcal{SC}(k)\bigcup\{s_0\}}r_{ij}^0(k,s)\}_{k=1}^K)=\sum_{k=1}^K\sum_{s\in\mathcal{SC}(k)\bigcup\{s_0\}}\eta_{ij}r_{ij}^0(k,s)$, where η_{ij} is communication latency between two nodes i and j. For delay minimization, we have $\iota_i(\boldsymbol{\psi}_i)=0$. $L^0(\cdot)$ is decomposable across links. We assign ν_{ij} to the cross-link capacity constraint (included in $\mathcal{C}^0$ given in (2.24b)) and write the decomposed Lagrangian as follows:

$$L_{ij}^0(\cdot)=\sum_{k=1}^K\sum_{s\in\mathcal{SC}(k)\bigcup\{s_0\}}\left(\frac{1}{2}\eta_{ij}r_{ij}^0(k,s)+\pi_{ij}^m(k,s)(r_{ij}^0(k,s)-r_{ij}^m(k,s))+\frac{\rho}{2}(r_{ij}^0(k,s)-r_{ij}^m(k,s))^2\right)$$

Algorithm 2: Algorithm to solve (2.24a)-(2.24b) with respect to $\{\mathbf{r}_m^0\}_{m=1}^M$

```
0. **Initialization** $e_1 = x_1 = x_2 = x_3 = 0, e_2 =$ a large number;
**repeat**
    1. $e_3 = (e_1 + e_2)/2$;
    2. Solve $\partial L_{ij}^0\left(\cdot\right)/\partial r_{ij}^0(k,s) = 0$ to find $r_{ij}^0(k,s)$ where $\nu_{ij} = e_1$ in $L_{ij}^0\left(\cdot\right)$ and
     compute $x_1 = \sum_{k=1}^K \sum_{s \in \mathcal{SC}(k) \bigcup \{s_0\}} r_{ij}^0(k,s) - C_{ij}$;
    3. Solve $\partial L_{ij}^0\left(\cdot\right)/\partial r_{ij}^0(k,s) = 0$ to find $r_{ij}^0(k,s)$ where $\nu_{ij} = e_2$ in $L_{ij}^0\left(\cdot\right)$ and
     compute $x_2 = \sum_{k=1}^K \sum_{s \in \mathcal{SC}(k) \bigcup \{s_0\}} r_{ij}^0(k,s) - C_{ij}$;
    4. Solve $\partial L_{ij}^0\left(\cdot\right)/\partial r_{ij}^0(k,s) = 0$ to find $r_{ij}^0(k,s)$ where $\nu_{ij} = e_3$ in $L_{ij}^0\left(\cdot\right)$ and
     compute $x_3 = \sum_{k=1}^K \sum_{s \in \mathcal{SC}(k) \bigcup \{s_0\}} r_{ij}^0(k,s) - C_{ij}$;
    5. **if** $x_1.x_3 < 0$ **then**
        $e_2 = e_3$;
    6. **if** $x_2.x_3 < 0$ **then**
        $e_1 = e_3$;
    7. **if** $x_1 < 0,\ x_2 < 0,\ x_3 < 0$ **then**
        13.1. Solve $\partial L_{ij}^0\left(\cdot\right)/\partial r_{ij}^0(k,s) = 0$ where $\nu_{ij} = 0$;
        13.2. Project the found $r_{ij}^0(k,s)$ variable to the positive orthant;
        13.3. $e_2 = e_1$;
**until** $e_2 - e_1$ *is close enough to zero*;
```

$$+\,\nu_{ij}\left(\sum_{k=1}^{K}\sum_{s\in\mathcal{SC}(k)\bigcup\{s_0\}} r_{ij}^0(k,s) - C_{ij}\right).$$

For the linear delay function, using the first order optimality $\partial L_{ij}^0\left(\cdot\right)/\partial r_{ij}^0(k,s) = 0$, we find

$$r_{ij}^0(k,s) = \left[\frac{-1}{\rho}\left(\eta_{ij}/2 + \pi_{ij}^m(k,s) + \nu_{ij}\right) + r_{ij}^m(k,s)\right]^+. \tag{2.25}$$

where $[\cdot]^+$ is a projection operator to the positive orthant. Then, we solve the complementary slackness condition $\nu_{ij}(\sum_{k=1}^K \sum_{s \in \mathcal{SC}(k) \bigcup \{s_0\}} r_{ij}^0(k,s) - C_{ij}) = 0$, where $r_{ij}^0(k,s)$ is found from (2.25). We do bisection on ν_{ij} and use the obtained $r_{ij}^0(k,s)$ to get $\sum_{k=1}^K \sum_{s \in \mathcal{SC}(k) \bigcup \{s_0\}}$ $r_{ij}^0(k,s) - C_{ij} = 0$. If there is no such positive ν_{ij}, we set the first derivative of (2.24a) to zero and project the obtained variables to the positive orthant. The proposed approach to solve (2.24a)-(2.24b) with a general separable $\varphi_{ij}(\cdot)$ is summarized in Algorithm 2.

In the second example, we consider the delay on each link as a non-linear function of flow rate, similar to [59], where

$$\varphi_{ij}(\{\sum_{s\in\mathcal{SC}(k)\bigcup\{s_0\}} r^0_{ij}(k,s)\}_{k=1}^K) = \frac{\sum_{k=1}^K \sum_{s\in\mathcal{SC}(k)\bigcup\{s_0\}} r^0_{ij}(k,s)}{C_{ij} - \sum_{k=1}^K \sum_{s\in\mathcal{SC}(k)\bigcup\{s_0\}} r^0_{ij}(k,s)}.$$

In this case, we substitute the quadratic upper-bound found from [55, eq. (12)] for $\varphi_{ij}(\cdot)$, where the Lipschitz constant is $K + \sum_{k=1}^K |\mathcal{SC}(k)|$. Thus, the quadratic upper-bound in the q^{th} iteration is

$$\begin{aligned}
&\varphi_{ij}(\{\sum_{s\in\mathcal{SC}(k)\bigcup\{s_0\}} r^0_{ij}(k,s)\}_{k=1}^K) \le \varphi_{ij}(\{\sum_{s\in\mathcal{SC}(k)\bigcup\{s_0\}} r^{0,q}_{ij}(k,s)\}_{k=1}^K) \\
&+\sum_{k=1}^K \sum_{s\in\mathcal{SC}(k)\bigcup\{s_0\}} (r^0_{ij}(k,s) - r^{0,q}_{ij}(k,s)) \frac{\partial \varphi_{ij}(\cdot)}{\partial r^0_{ij}(k,s)}|_{r^0_{ij}(k,s)=r^{0,q}_{ij}(k,s)} \\
&+\sum_{k=1}^K \sum_{s\in\mathcal{SC}(k)\bigcup\{s_0\}} \frac{K + \sum_{k=1}^K |\mathcal{SC}(k)|}{2} \left(r^0_{ij}(k,s) - r^{0,q}_{ij}(k,s) \right)^2. \qquad (2.26)
\end{aligned}$$

The above upper-bound is substituted for $\varphi_{ij}(\left\{\sum_{s\in\mathcal{SC}(k)\bigcup\{s_0\}} r^0_{ij}(k,s)\right\}_{k=1}^K)$ in the objective function and the Lagrangian (2.24a) is written accordingly. Similarly, to solve the problem with the upper-bound, we use the first order optimality and bisection on ν_{ij}. Once the problem is solved, we update the upper-bound (2.26) in the Lagrangian (2.24a) and solve again. We repeat this until all $r^0_{ij}(k,s)$ variables converge. It is straightforward to show that this successive upper-bound minimization converges to the global solution to (2.24a)-(2.24b) with respect to $\{\mathbf{r}^0_m\}_{m=1}^M$ [19, Theorem 2]. Minimizing $L^0(\cdot)$ with respect to $\{\mathbf{r}^0_m\}_{m=1}^M$ can be done in parallel for each cross-link.

To minimize $L^0(\cdot)$ with respect to $\psi^0_{i,s}(k)$, we use a similar dual approach. Consider the following problem:

$$\begin{aligned}
\min_{\psi^0_{i,s}(k)} \quad & \theta^m_{i,s}(k)(\psi^0_{i,s}(k) - \psi^m_{i,s}(k)) + \frac{\rho}{2}\left(\psi^0_{i,s}(k) - \psi^m_{i,s}(k)\right)^2 \\
\text{s.t.} \quad & \sum_{i\in\mathcal{V}_\psi} \psi^0_{i,s}(k) = 1, \qquad \forall\, s \in \mathcal{SC}(k), \forall k,
\end{aligned}$$

Algorithm 3: Algorithm to solve (2.24a)-(2.24b) with respect to $\{\boldsymbol{\psi}^0_m\}_{m=1}^M$

```
0. **Initialization** $x_1 = x_2 = x_3 = 0, e_2 =$ a large number, $e_1 = -e2$;
**repeat**
    1. $e_3 = (e_1 + e_2)/2$;
    2. Find $x_1 = \sum_{i\in\mathcal{V}_\psi} \psi^0_{i,s}(k) - 1$ where $\psi^0_{i,s}(k)$ is found from (2.27) and
    $\nu_s(k) = e_1$;
    3. Find $x_2 = \sum_{i\in\mathcal{V}_\psi} \psi^0_{i,s}(k) - 1$ where $\psi^0_{i,s}(k)$ is found from (2.27) and
    $\nu_s(k) = e_2$;
    4. Find $x_3 = \sum_{i\in\mathcal{V}_\psi} \psi^0_{i,s}(k) - 1$ where $\psi^0_{i,s}(k)$ is found from (2.27) and
    $\nu_s(k) = e_3$;
    5. **if** $x_1.x_3 < 0$ **then**
        $e_2 = e_3$;
    6. **if** $x_2.x_3 < 0$ **then**
        $e_1 = e_3$;
**until** $e_2 - e_1$ *is close enough to zero*;
7. $\psi^0_{i,s} = \left[\min\left\{\psi^m_{i,s} - \frac{\theta^m_{i,s}}{\rho}, 1\right\}\right]^+$ and $\psi^0_i = \left[\min\left\{\psi^m_i - \frac{\theta^m_i}{\rho}, 1\right\}\right]^+$;
```

$$1 \geq \psi^0_{i,s}(k) \geq 0, \qquad\qquad \forall\, s \in \mathcal{SC}(k), \forall k, \forall i \in \mathcal{V}_\psi.$$

We write the Lagrangian for the above problem as

$$L'(\cdot) = \theta^m_{i,s}(k)(\psi^0_{i,s}(k) - \psi^m_{i,s}(k)) + \frac{\rho}{2}\left(\psi^0_{i,s}(k) - \psi^m_{i,s}(k)\right)^2 + \nu_s(k)\Big(\sum_{i\in\mathcal{V}_\psi} \psi^0_{i,s}(k) - 1\Big),$$

where $\nu_s(k)$ is the Lagrange multiplier. From the first order optimality condition, we find $\psi^0_{i,s}(k)$ as follows:

$$\psi^0_{i,s}(k) = \left[\min\left\{\psi^m_{i,s}(k) - \frac{\theta^m_{i,s}(k) + \nu_s(k)}{\rho}, 1\right\}\right]^+. \tag{2.27}$$

We do bisection on $\nu_s(k)$ and find $\psi^0_{i,s}(k)$ from (2.27) until we have $\sum_{i\in\mathcal{V}_\psi} \psi^0_{i,s}(k) = 1$. The solution for $\psi^0_{i,s}$ is $\psi^0_{i,s} = \left[\min\left\{\psi^m_{i,s} - \frac{\theta^m_{i,s}}{\rho}, 1\right\}\right]^+$ and for ψ^0_i is $\psi^0_i = \left[\min\left\{\psi^m_i - \frac{\theta^m_i}{\rho}, 1\right\}\right]^+$. The summary of the proposed approach to solve (2.24a)-(2.24b) with respect to $\{\boldsymbol{\psi}^0_m\}_{m=1}^M$ is given in Algorithm 3.

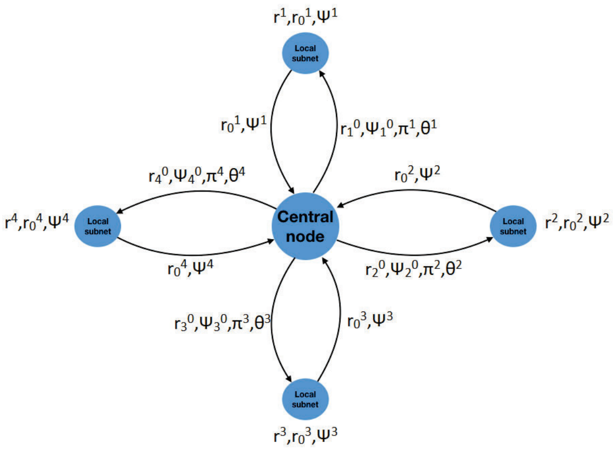


Figure 2.1: Information exchange among four subnetworks.

The problem (2.23a)-(2.23b) is convex, which can be solved efficiently to global optimality using solvers such as Gurobi or CPLEX. In each iteration of the proximal ADMM, optimizations in subnetworks are solved and the optimized variables are sent to the central controller. Then, the optimization in the central controller is solved and the optimized primal and dual variables are sent to subnetworks. This message passing interface for a central controller and four subnetworks is shown in Fig. 2.1. In the t^{th} iteration of the proximal ADMM, the update rules for $\boldsymbol{\pi}^{m,t}$ and $\boldsymbol{\theta}^{m,t}$ are

$$\boldsymbol{\pi}^{m,t} = \boldsymbol{\pi}^{m,t-1} + \rho \left(\mathbf{r}_m^{0,t} - \mathbf{r}_0^{m,t} \right), \boldsymbol{\theta}^{m,t} = \boldsymbol{\theta}^{m,t-1} + \rho \left(\boldsymbol{\psi}_m^{0,t} - \boldsymbol{\psi}^{m,t} \right). \tag{2.28}$$

Once the problem in the central controller is solved, $\{\mathbf{r}_m^{0,t}\}_{m=1}^M$, $\{\boldsymbol{\psi}_m^{0,t}\}_{m=1}^M$, $\{\boldsymbol{\pi}^{m,t}\}_{m=1}^M$ and $\{\boldsymbol{\theta}^{m,t}\}_{m=1}^M$ are sent to subnetworks. Such message passing is repeated until primal and dual residuals are close enough to zero [60, eq. (3.12)]. The brief description of the proximal ADMM approach is given in Algorithm 4. Every limit point of the sequence $\{\mathbf{r}^{m,t}, \mathbf{r}_0^{m,t}, \boldsymbol{\psi}^{m,t}, \boldsymbol{\pi}^{m,t}, \boldsymbol{\theta}^{m,t}\}$ generated by proximal ADMM given in Algorithm 4 converges to the global solution to (2.14) when the optimization (2.14) is convex and feasible [54, pp. 706 – 707]. One advantage of Algorithm 4 is that it does not share any sensitive information of one subnetwork such as the subnetwork topology or a computation capacity with the

Algorithm 4: Proximal ADMM-based joint VNF placement and routing

```
0. Initialization Set $\mathbf{r}_0^{m,0} = \mathbf{0}, \mathbf{r}^{0,0} = \mathbf{0}$, $\boldsymbol{\psi}^{0,0} = \mathbf{0}$, $\boldsymbol{\psi}^{m,0} = \mathbf{0}$, $\boldsymbol{\pi}^{m,0} = \mathbf{0}$, $\boldsymbol{\theta}^{m,0} = \mathbf{0}$,
 $z = 1$;
repeat
    1. $t = 0$;
    repeat
        2. Find (2.22) and $L^0(\cdot)$;
        3. For each subnetwork $\mathcal{N}^m$ solve (2.23a)-(2.23b);
        4. a) In central controller, solve (2.24a)-(2.24b);
           b) Update dual variables using (2.28);
        5. $t = t + 1$;
    until Primal and dual residuals are close to zero;
    6. Update $u_{i,s,k}^z$ in $L^m\left(\mathbf{r}^m, \mathbf{r}_0^m, \boldsymbol{\psi}^m, \boldsymbol{\pi}^m, \boldsymbol{\theta}^m\right)$ using $u_{i,s,k}^z = \sigma^z\, p\,(\psi_{i,s}^{z-1}(k) + \epsilon)^{p-1}$;
    7. $z = z + 1$;
until Each $\psi_{i,s}^m(k)$ is binary;
```

central controller or neighboring subnetworks.

2.6 Link Capacity and Server Purchase with In-Subnetwork Processing

In the existing work [24, 26, 32, 39], the common assumption is that servers and links are already installed in the network. In this section, given flow demands, we propose novel formulations and algorithms to plan the network. In particular, from a sparse optimization perspective, we identify the number and locations of servers that should be purchased and installed among candidate places. Required link capacities to meet flow demands are identified as well. The required network equipment is identified such that each subnetwork is able to carry out the joint routing and VNF placement locally and independently of others. Simultaneously, the overall purchase cost for each subnetwork is minimized.

2.6.1 Mathematical Formulation

If a flow is routed on a link, it is required to purchase a *portion* of that link's capacity. This depends on the extent to which the flow occupies the link's capacity. Suppose the cost of the link capacity is a convex function $\tau_{ij}(\cdot)$. A server can be considered as a candidate place to provide VNFs only if it is purchased and installed in the network [25, 61]. In other words, the cost function for servers is discrete. Sparsity for the purchased servers is preferred. When $\psi_{i,s}(k) > 0$, then $\|\boldsymbol{\psi}_i\|_F \neq 0$. Suppose that the cost of purchasing a server is β. We aim to minimize $\beta \, \left\|\{\|\boldsymbol{\psi}_i\|_F\}_{i\in\mathcal{V}_\psi}\right\|_0$ plus the costs of link capacities. The cost function for link capacity and server purchase is

$$\beta \, \left\|\{\|\boldsymbol{\psi}_i\|_F\}_{i\in\mathcal{V}_\psi}\right\|_0 + \sum_{(i,j)\in\mathcal{L}} \tau_{ij}\left(\left\{\sum_{s\in\mathcal{SC}(k)\bigcup\{s_0\}} r^0_{ij}(k,s)\right\}_{k=1}^{K}\right). \tag{2.29}$$

The ℓ_0-norm in the above objective function is non-convex. In order to minimize the cost function, we use the group LASSO regularization proposed in [62] to approximate ℓ_0-norm by ℓ_1-norm. This method is well-studied in the compressive sensing literature and it is widely used to increase the sparsity of the solution [62, 5]. We substitute $\beta \sum_{i\in\mathcal{V}_\psi} \gamma_i \, \|\boldsymbol{\psi}_i\|_F$ for $\beta \, \left\|\{\|\boldsymbol{\psi}_i\|_F\}_{i\in\mathcal{V}_\psi}\right\|_0$ in (2.29). The coefficient γ_i assigned to $\|\boldsymbol{\psi}_i\|_F$ is updated iteratively. The update rule for the coefficient in the h^{th} iteration is

$$\gamma_{i,h} = \frac{\gamma_{i,0}}{\left\|\boldsymbol{\psi}_{i,h-1}\right\|_F + \delta},$$

where $\gamma_{i,0} > 0$ is an initial positive value. Moreover, δ is a small positive value introduced to make the approximation stable when $\left\|\boldsymbol{\psi}_{i,h}\right\|_F$ is close to zero. It is shown in Section 2.7 that this method is efficient and converges fast. For the simplicity of the model and without loss of generality, an identical price is assumed for all servers.

Similar to the previous section, it is desired to provide VNFs to flows within the subnetworks where they have originated from. The goal is to *minimize the costs of purchased servers and link capacities while keeping the flow routing and processing locally within subnetworks to the extent possible.* To promote local subnetwork processing, we use the penalties introduced in the previous section. It is allowed to purchase servers and link capacities in order to enable flows to receive each VNF from just one server. Thus, we multiply the PSUM penalty by κ in the objective function where $\kappa \geq P_1, P_2, \beta$. The required objective function is

$$
\begin{aligned}
R\left(\mathbf{r}, \boldsymbol{\psi}\right) = & \sum_{(i,j)\in\mathcal{L}} \tau_{ij} \left(\left\{ \sum_{s\in\mathcal{SC}(k)\bigcup\{s_0\}} r^0_{ij}(k,s) \right\}_{k=1}^{K} \right) \\
& + P_1 \sum_{m=1}^{M} \sum_{k:S(k)\in\mathcal{V}^m} \sum_{i\notin\mathcal{V}^m_\psi, i\in\mathcal{V}_\psi} \sum_{s\in\mathcal{SC}(k)} \psi_{i,s}(k) + P_2 \sum_{m=1}^{M} \sum_{k:S(k)\notin\mathcal{V}^m} \sum_{(i,j)\in\mathcal{L}^m} \sum_{s\in\mathcal{SC}(k)\bigcup\{s_0\}} r_{ij}(k,s) \\
& + \beta \sum_{i\in\mathcal{V}_\psi} \gamma_i \left\|\boldsymbol{\psi}_i\right\|_F + \kappa \sum_{i\in\mathcal{V}_\psi} \sum_{k=1}^{K} \sum_{s\in\mathcal{SC}(k)} u^z_{i,s,k}\, \psi_{i,s}(k). \qquad (2.30)
\end{aligned}
$$

2.6.2 A Centralized Optimization Algorithm for Link Capacity and Server Purchase

The optimization problem for minimizing server and link capacity purchase cost is

$$
\begin{aligned}
\min_{\mathbf{r},\boldsymbol{\psi},\{\psi_{i,s}\}_{s,i\in\mathcal{V}_\psi},\{\psi_i\}_{i\in\mathcal{V}_\psi}} \quad & R\left(\mathbf{r}, \boldsymbol{\psi}\right) \qquad (2.31) \\
\text{s.t.} \quad & (2.1), (2.2), (2.3), (2.4), (2.5), (2.6), (2.7), (2.8), (2.10), (2.11), (2.12), (2.13), \\
& r_{ij}(k) \geq 0, r_{ij}(k,s) \geq 0,\ \forall\, s \in \mathcal{SC}(k), \forall\, k, \forall\, (i,j) \in \mathcal{L}, \\
& \psi_{i,s}(k) \in [0,1],\ \forall\, i \in \mathcal{V}_\psi, \forall\, s \in \mathcal{SC}(k), \forall\, k.
\end{aligned}
$$

After replacing ℓ_0-norm by ℓ_1-norm, the new problem is a convex conic optimization. It can be solved efficiently via different classes of algorithms such as interior-point methods or ADMM. The two steps of the centralized algorithm are as follows:

1. Solve (2.31) iteratively and in each iteration update $u^z_{i,s,k}$ in $R\left(\mathbf{r}, \boldsymbol{\psi}\right)$ using $u^z_{i,s,k} = \sigma^z\, p\, (\psi^{z-1}_{i,s}(k) + \epsilon)^{p-1}$ and $z = z + 1$ until each $\psi_{i,s}(k)$ converges to either zero or one.
2. When each $\psi_{i,s}(k)$ is either zero or one, coefficients $\gamma_i, i \in \mathcal{V}_\psi$ are updated.

After updating γ_i, we solve the optimization (2.31) with the updated objective function again. We repeat until γ_i coefficients converge. The summary is given in Algorithm 5.

Algorithm 5: Link and server purchase algorithm

```
0. **Initialization** Assign positive values to $\gamma_{i,0}$ and $\delta$, $h = 0, \boldsymbol{\psi} = \mathbf{0}$;
repeat
    1. $z = 1$;
    repeat
        2. Solve the optimization problem (2.31);
        3. Update $u^z_{i,s,k}$ in $R(\mathbf{r}, \boldsymbol{\psi})$ using $u^z_{i,s,k} = \sigma^z\, p\, (\psi^{z-1}_{i,s}(k) + \epsilon)^{p-1}$;
        4. $z = z + 1$;
    until Each $\psi_{i,s}(k)$ is binary;
    5. $\gamma_{i,h+1} = \frac{\gamma_{i,0}}{\|\boldsymbol{\psi}_{i,h}\|_F + \delta}$;
    6. $h = h + 1$;
until All $\gamma_{i,h}$, $\forall i \in \mathcal{V}_\psi$, converge;
```

2.6.3 A Distributed Optimization Algorithm for Link Capacity and Server Purchase

Here, we propose a proximal ADMM-based algorithm for the optimization problem (2.31). Variables of (2.31) are the same as variables of (2.14). The feasible sets of variables are identical. We decompose $R(\mathbf{r}, \boldsymbol{\psi})$ in the same way that we decomposed $T_2(\mathbf{r}, \boldsymbol{\psi})$ in the previous section. The objective function of the central controller is

$$R^0\left(\{\mathbf{r}^0_m\}_{m=1}^M\right) = \frac{1}{2} \sum_{(i,j)\in\mathcal{L}^0} \tau_{ij} \left(\left\{ \sum_{s\in\mathcal{SC}(k)\bigcup\{s_0\}} r^0_{ij}(k,s) \right\}_{k=1}^K \right). \tag{2.32}$$

The objective function of subnetwork $\mathcal{N}^m$ is

$$\begin{aligned} R^m\left(\mathbf{r}^m, \mathbf{r}^m_0, \boldsymbol{\psi}^m\right) = &\sum_{(i,j)\in\mathcal{L}^m\setminus\mathcal{L}^0_m} \tau_{ij} \left(\left\{ \sum_{s\in\mathcal{SC}(k)\bigcup\{s_0\}} r^m_{ij}(k,s) \right\}_{k=1}^K \right) \\ &+ \frac{1}{4} \sum_{(i,j)\in\mathcal{L}^0_m} \tau_{ij} \left(\left\{ \sum_{s\in\mathcal{SC}(k)\bigcup\{s_0\}} r^m_{ij}(k,s) \right\}_{k=1}^K \right) \\ &+ \beta \sum_{i\in\mathcal{V}^m_\psi} \gamma_i \|\boldsymbol{\psi}_i\|_F + \sum_{k:S(k)\notin\mathcal{V}^m} \Big(P_1 \sum_{i\in\mathcal{V}^m_\psi} \sum_{s\in\mathcal{SC}(k)} \psi^m_{i,s}(k) + P_2 \sum_{(i,j)\in\mathcal{L}^m} \sum_{s\in\mathcal{SC}(k)\bigcup\{s_0\}} r^m_{ij}(k,s) \Big) \end{aligned}$$

$$+ \sum_{i \in \mathcal{V}_{\psi}^{m}} \sum_{k=1}^{K} \sum_{s \in \mathcal{SC}(k)} u_{i,s,k}^{z} \, \psi_{i,s}^{m}(k).$$

We rewrite the optimization given in (2.31) as (2.19a)-(2.19f) with $R^m\left(\mathbf{r}^m, \mathbf{r}_0^m, \boldsymbol{\psi}^m\right)$ instead of $T^m\left(\mathbf{r}^m, \mathbf{r}_0^m, \boldsymbol{\psi}^m\right)$ and (2.32) instead of $T^0\big(\{\mathbf{r}_m^0\}_{m=1}^M\big)$. We copy variables as done in (2.16a)-(2.16b). The optimization problems that subnetworks and the central controller need to solve are given in (2.23a)-(2.23b) and (2.24a)-(2.24b), respectively. After we substitute $R^0\big(\{\mathbf{r}_m^0\}_{m=1}^M\big)$ for $T^0\big(\{\mathbf{r}_m^0\}_{m=1}^M\big)$, we are still able to use the dual approaches of Algorithms 2 and 3 which are developed to solve (2.24a)-(2.24b). Update rules for dual variables are given in (2.28). The distributed approach for link capacity and server purchase is given in Algorithm 6. Every limit points of the sequence $\{\mathbf{r}^{m,t}, \mathbf{r}_0^{m,t}, \boldsymbol{\psi}^{m,t}, \boldsymbol{\pi}^{m,t}, \boldsymbol{\theta}^{m,t}\}$ generated by the proximal ADMM converges to the global solution to (2.31) when the problem is feasible. The reason is that the objective function of the optimization problem (2.31) is convex. Moreover, its constraints are affine [54, pp. 706 – 707].

2.7 Numerical Results

We evaluate the performance of our proposed algorithms through extensive simulations and several settings. First, to compare the performance of our algorithms with existing methods for the one subnetwork case (because the existing algorithms are designed only for this case). Consider the network shown in Fig. 2.2, one subnetwork with 112 nodes and 440 bidirectional links. In Fig. 2.2, ellipses are servers. In addition, the triangular nodes are the network routers, which cannot be chosen as a source of any flow. Suppose there are six servers, which can provide all four VNFs. Assume flows randomly originate from rectangular nodes and the common destination is the black circle node. The rate of each flow is randomly chosen from the interval $(0, 4]$. Moreover, SFCs are chosen randomly for each flow.

We compare the performance of our Algorithms 1 and 4 (which are equivalent for one subnetwork) with algorithms in [25], called SAMA and MDM, and the algorithm in [39], called Kariz. We consider the network depicted in Fig. 2.2 for performance comparisons and assume that we have one subnetwork. Suppose we minimize the linear delay where $\eta_{ij} = 1/C_{ij}$. Suppose CPU is the computational resource at servers and $\omega_{fs} = 1$, $\forall s$. To

Algorithm 6: Proximal ADMM-based link capacity and server purchase

```
0. Initialization Assign positive values to $\gamma_{i,0}$ and $\delta$, $\mathbf{r}_0^{m,0} = \mathbf{0}, \mathbf{r}^{0,0} = \mathbf{0}$, $\boldsymbol{\psi}^{0,0} = \mathbf{0}$,
   $\boldsymbol{\psi}^{m,0} = \mathbf{0}$, $\boldsymbol{\pi}^{m,0} = \mathbf{0}$, $\boldsymbol{\theta}^{m,0} = \mathbf{0}$, $h = 0$, $z = 1$;
repeat
    1. $z = 1$;
    repeat
        2. $t = 0$;
        repeat
            3. Find $L^m(\cdot)$ and $L^0(\cdot)$;
            4. For each subnetwork $\mathcal{N}^m$ solve (2.23a)-(2.23b) with $R^m(\cdot)$ instead of
               $T^m(\cdot)$;
            5. a) In central controller, solve (2.24a)-(2.24b) with $R^0(\cdot)$ instead of
               $T^0(\cdot)$;
               b) Update dual variables using (2.28);
            6. $t = t + 1$;
        until The primal and dual residuals are close to zero;;
        7. Update $u^z_{i,s,k}$ in $L^m\left(\mathbf{r}^m, \mathbf{r}_0^m, \boldsymbol{\psi}^m, \boldsymbol{\psi}^m, \boldsymbol{\theta}^m\right)$ using
           $u^z_{i,s,k} = \sigma^z \; p \; (\psi^{z-1}_{i,s}(k) + \epsilon)^{p-1}$;
        8. $z = z + 1$;
    until Each $\psi^m_{i,s}(k)$ is binary;
    9. $\gamma_{i,h+1} = \frac{\gamma_{i,0}}{\|\boldsymbol{\psi}_{i,h}\|_F + \delta}$;
    10. $h = h + 1$;
until All $\gamma_{i,h}$, $\forall i \in \mathcal{V}_\psi$, converge;
```

compare for a large set of source/destination pairs, we consider 50 randomly generated scenarios each with 40 source/destination pairs. We find VNF placement and routing for delay minimization by approaches in [25, 39]. The ratio of the obtained delay from [25, 39] to the found delay from Algorithm 1 (or equivalently Algorithm 4) is

$$\chi = \frac{\text{Delay obtained from each algorithm given in [25] or [39]}}{\text{Delay found from Algorithm 1 (or equivalently Algorithm 4)}}.$$

We plot the CDF of χ in Fig. 2.3 when we increase the SFC length. When we increase the SFC length, the performance of the heuristic algorithms given in [25, 39] degrades. The reason is that as the SFC length increases, the number of feasible flow-server assignments increases, which negatively affects the heuristic approaches given in [25, 39].

The CPU time is given in Table 2.2 when MATLAB 9.1.0.441655 is used. When the SFC

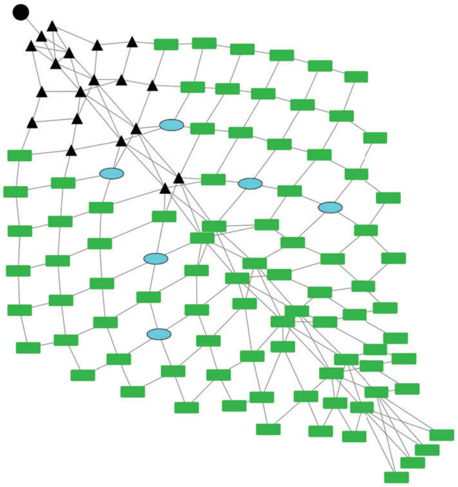


Figure 2.2: A network with 112 nodes and 440 links. Ellipse nodes are servers.

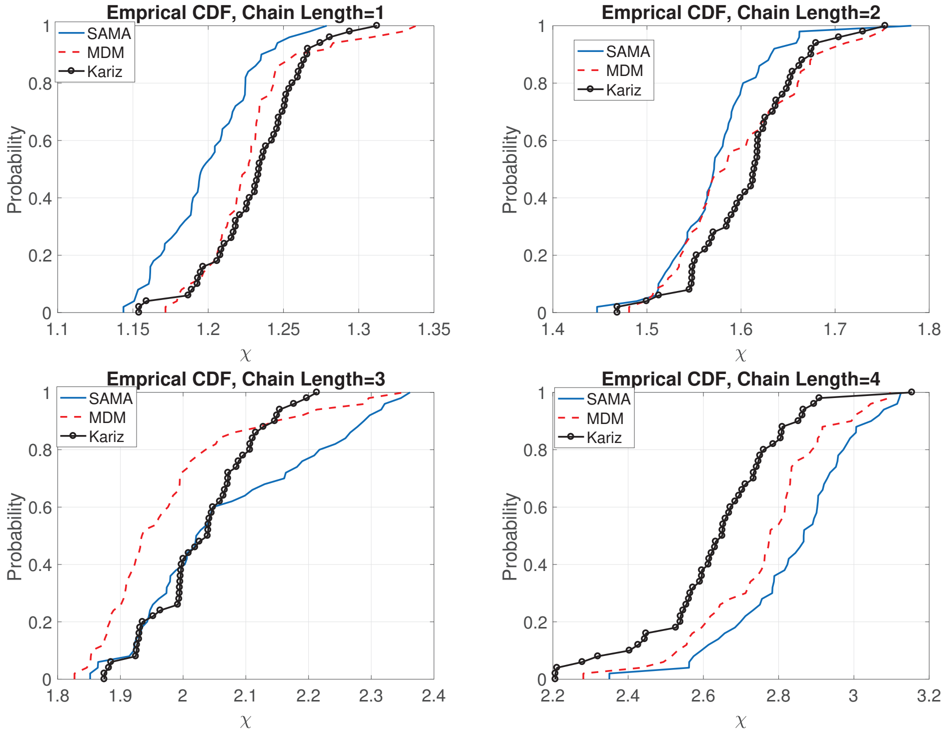


Figure 2.3: The distribution of ratio χ in 50 random scenarios, each with 40 s-d pairs.

CPU time for delay minimization				
No. of flows	SFC length is one			
	SAMA	MDM	Kariz	Algorithm 1 or 4
40	385.54s	377.99s	347.67s	330.22s
CPU time for delay minimization				
	SFC length is two			
	SAMA	MDM	Kariz	Algorithm 1 or 4
40	602.01s	582.60s	580.50s	570.03s
	SFC length is three			
	SAMA	MDM	Kariz	Algorithm 1 or 4
40	791.95s	802.14s	714.94s	608.61s
CPU time for delay minimization				
	SFC length is four			
	SAMA	MDM	Kariz	Algorithm 1 or 4
40	1018.37s	999.53s	908.11s	864.01s

Table 2.2: CPU time for delay minimization of 50 scenarios, each with 40 pairs of source/sink.

length increases, the required CPU times for heuristic approaches increase fast. The reason is that when the SFC length increases, a greater number of flow-server assignments need to be solved by the heuristic approaches. We observe that the proposed Algorithms 1 and 4 not only reduce the total delay considerably, they need a shorter CPU time, specifically, when the SFC length is large.

We compare the performance of the proposed Algorithms 5 and 6 (which are equivalent for one subnetwork) with SAMA and MDM proposed in [25] in terms of minimizing the cost of link capacity and server purchase in Fig. 2.4. We consider 50 scenarios, each of which includes 50 source/destination pairs. The ratio of the obtained link capacity and server cost from [25] to that found from Algorithm 5 (or equivalently Algorithm 6) is denoted by χ', and defined as follows:

$$\chi' = \frac{\text{Purchase cost obtained from each algorithm given in [25]}}{\text{Purchase cost yielded from Algorithm 5 (or equivalently Algorithm 6)}},$$

where the overall cost of link capacity and server is

$$\sum_{(i,j)\in\mathcal{L}} \sum_{k=1}^{50} r_{ij}(k) + 5 \left\| \{\|\boldsymbol{\psi}_i\|_F\}_{i\in\mathcal{V}_\psi} \right\|_0 .$$

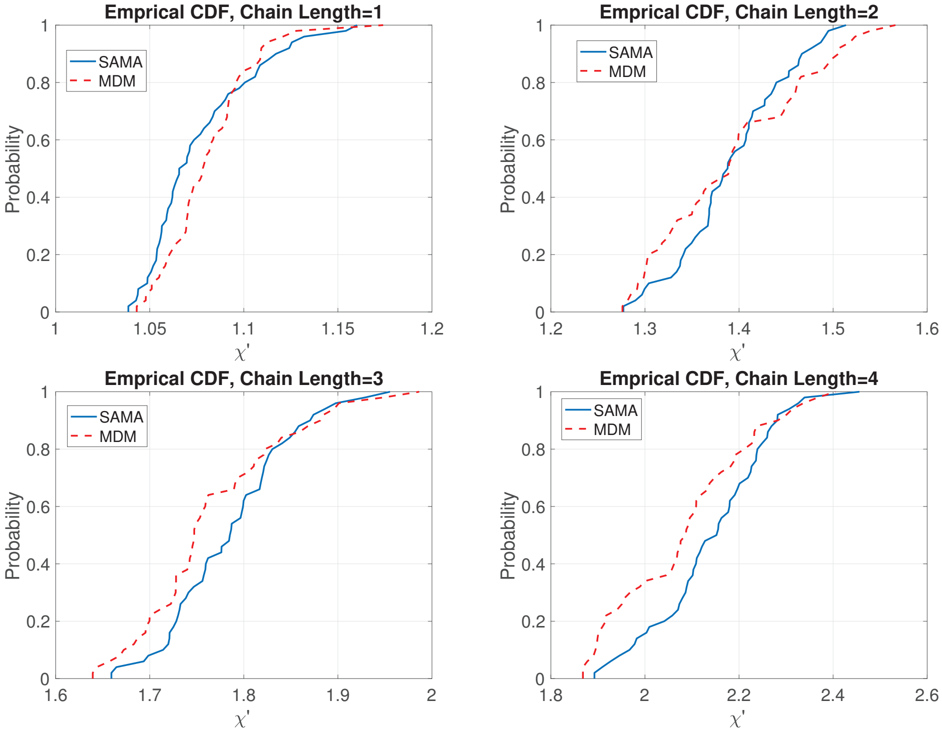


Figure 2.4: The distribution of ratio χ' in 50 random scenarios, each with 50 s-d pairs.

The CDF of the ratio χ' is shown in Fig. 2.4. Analogous to the previous results, Algorithm 5 outperforms the heuristic methods and also requires less CPU time as given in Table 2.3.

Suppose 20 flows are served by six servers in the network shown in Fig. 2.2. The rate of each flow is randomly chosen from the interval $(0, 4]$. Flow sources are chosen randomly from rectangular nodes. The computational capacity of each server is 60. The sum of flow rates is 53. Suppose that the cost of buying one unit of a link capacity is one. In addition, the cost of purchasing a server is 20. The cost of link capacity and server purchase is thus as follows: $\sum_{(i,j)\in\mathcal{L}}\sum_{k=1}^{20} r_{ij}(k) + 20\left\|\{\|\boldsymbol{\psi}_i\|_F\}_{i\in\mathcal{V}_\psi}\right\|_0$. We minimize the purchase cost by buying one server. We see from Fig. 2.5 that we need only two iterations of Algorithm 6 (or equivalently Algorithm 5) to minimize the cost of link capacity and server purchase.

Consider the network in Fig. 2.6, which is made of 4 subnetworks. Each subnetwork is

CPU time for link capacity and server purchase			
No. of flows	SAMA	MDM	Algorithm 5 or 6
	SFC length is one		
50	587.72.44s	588.86s	573.00s
	SFC length is two		
50	804.47s	582.60s	571.19s
	SFC length is three		
50	1047.74s	1068.45s	780.07s
	SFC length is four		
50	1344.32s	1334.39s	1024.50s

Table 2.3: CPU time of link capacity and server purchase for 50 scenarios, each with 50 pairs of source/sink.

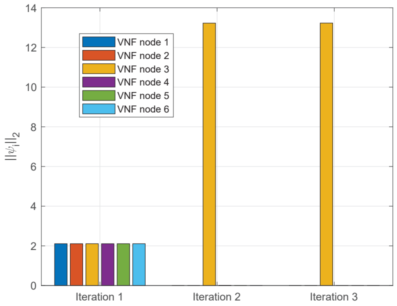


Figure 2.5: Convergence of Algorithm 5 (or equivalently Algorithm 6). One server is purchased.

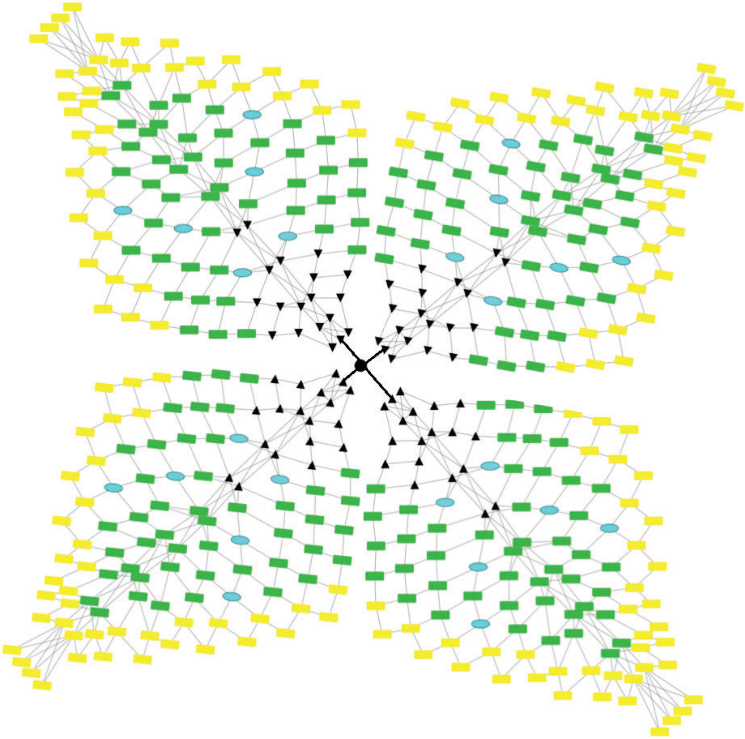
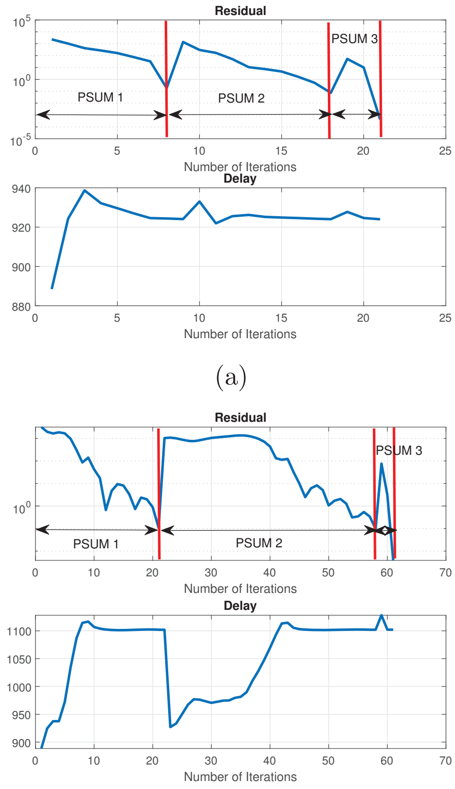


Figure 2.6: One hundred and sixty sources, yellow nodes, in a network with 445 nodes and 1760 links. Rate of each flow is 0.5.



(a)

(b)

Figure 2.7: (a) SFCs of all flows are of length one with (a) enough flow computational capacities in all subnetworks; (b) zero flow computational capacity in two subnetworks.

equal to the network given in Fig. 2.2. Made from four subnetworks, the network has 445 nodes and 1760 links in total. Each subnetwork includes six servers, as in Fig. 2.2, and 40 sources of flows. The common destination for all flows is the black circle node. The rate of each flow is 0.5 and the computational capacity of every server is 40, which is sufficient to process flows locally within subnetworks (setting 1). Suppose $\eta_{ij} = 1$. Methods in [60] are used to adjust ρ in the proximal ADMM. The convergence of the proximal ADMM given in Algorithm 4 is depicted in Fig. 2.7a when SFCs of flows are of length one. Recall from Section 2.5 that PSUM is an iterative algorithm. We need to solve the optimization (2.14) multiple times and each time the linearized PSUM penalty in the objective function is updated. Each time the optimization (2.14) is solved by the proximal ADMM (the

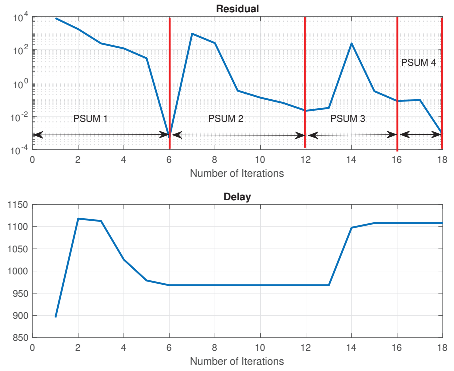


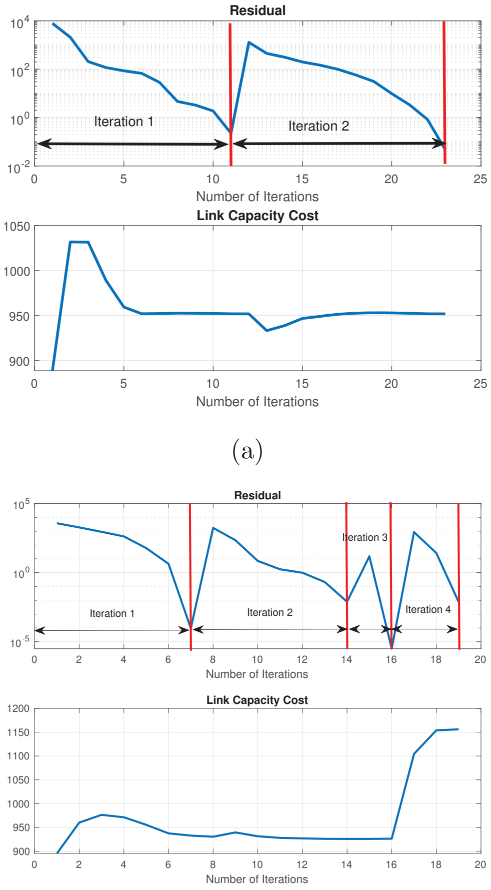
Figure 2.8: SFCs of all flows are of length two and enough flow computational capacities exist in all subnetworks.

inner loop of Algorithm 4). Arrows in Fig. 2.7a show inner loops. It is seen from Fig. 2.7a that three iterations of PSUM are required. The first, second and third iterations of PSUM require 8, 10 and 3 iterations of the proximal ADMM, respectively. To solve the optimization (2.14) distributedly three times, the total number of proximal ADMM iterations is 21.

Next, we change the computational capacity of the servers. Assume that the computational capacity in two subnetworks is zero. Moreover, the computational capacity of each server in the other two subnetworks is 40 (setting 2). Then, 80 of 160 flows need to receive VNFs from neighboring subnetworks. From Fig. 2.7b, we see that three iterations of PSUM are required. The total number of proximal ADMM iterations for solving three iterations of PSUM is 61.

The convergence of Algorithm 4 when the length of each SFC is two is depicted in Fig. 2.8 (setting 3). We consider sufficient computational capacity exists in servers of each subnetwork. Compared to Fig. 2.7a, it is seen that when the SFC length is two, a larger delay is yielded. The reason is that when the number of required VNFs of a flow increases, the average number of servers that a flow needs to go through to receive VNFs increases. Consequently, the path length and delay increase. We need to linearize the PSUM penalty four times and the overall number of proximal ADMM iterations is 18.

Next, we consider Algorithm 6. Suppose SFCs of all flows are of length one. The computational capacity of each server is 20 and the rate of each flow is 0.5. Thus, all flows



(a)

(b)

Figure 2.9: Service function chain of all flows are of length (a) one; (b) two. Enough flow computational capacities exist in purchased servers for all subnetworks.

originating from one subnetwork can be served by one server. The cost of link capacity and server purchase is $\sum_{(i,j)\in\mathcal{L}}\sum_{k=1}^{160} r_{ij}(k) + 40\left\|\{\|\boldsymbol{\psi}_i\|_F\}_{i\in\mathcal{V}_\psi}\right\|_0$. One server is purchased within each subnetwork. This implies that no flow needs to go out of its local subnetwork to receive VNFs. Fig. 2.9a depicts the convergence of the proximal ADMM. It is seen that 23 iterations of the proximal ADMM are required with one update of γ_i (two outer iterations) in Algorithm 6. Consider SFCs of all flows are of length two and the cost of link capacity and server purchase is identical to the previous case. Two servers are purchased in each subnetwork in this case. Three updates of γ_i in Algorithm 6 and total 19 proximal ADMM iterations are required for the convergence.

2.8 Concluding Remarks

In this chapter, we studied the minimization of routing and VNF instantiation cost in service-oriented networks while flows are locally processed within the subnetworks. Each subnetwork is owned and operated by an administrator. We proposed efficient centralized and distributed algorithms to solve the problem. The distributed algorithm is not only scalable for large networks, but also reduces the cost of implementation and keeps sensitive information within subnetworks. We extended the formulation and studied the problem of link capacity and server purchase for joint VNF placement and traffic engineering. Given flow demands, besides the optimization of traffic steering and VNF instantiation, the location/number of servers and link capacities required by administrators are found such that the spent budget for infrastructure purchase by each administrator is minimized. A distributed and scalable approach is also proposed for this problem. Extensive simulation results are given to verify the efficiency of our proposed approaches against existing work. The direction for future research is to consider base stations as destinations for flows and optimize transmission to users jointly with routing and service instantiation.

Chapter 3

Resource Reservation in Backhaul and Radio Access Network with Uncertain User Demands

Resource reservation is an important step in network planning and management due to its significant effects on the user quality of service. For wireless data networks operating in random and dynamic environments, finding resource reservation protocols that remain robust under uncertain user demands is challenging. Resource reservation, which balances network performance and its hardware costs, involves traffic forecasting and resource allocation for the predicted traffic [63, 64, 65]. Resource reservation in the backhaul and RAN should satisfy a wide range of applicable traffic demands. In particular, both the link capacity in the backhaul and transmission resources in RAN should be sliced and reserved for users such that upon the arrival of a new demand, the network is able to support it. To serve each user, a certain amount of resources in the backhaul and RAN is reserved. Along with user demand statistics, the available link capacity in the backhaul and transmission resources in RAN need to be considered to make reservations for each user. Resource reservation is a widely used means for network slicing, e.g., [66, 67, 68, 69, 70, 71].

3.1 Prior Work

For communication networks, where both link budget and node budget are to be reserved, different approaches are proposed for resource reservation. Existing papers either make resource reservations to satisfy minimum rate requirements of users, e.g., [72, 73, 74, 75, 76, 77, 78, 79], or consider variable user demands that are uncertain, e.g., [80, 81, 82, 83, 14, 7, 8, 9, 10, 84, 12, 85]. In [72], the user association and backhaul routing problem with deterministic user demands aiming at the energy efficiency maximization of the network is studied, where an efficient heuristic algorithm is proposed to solve the formulated mixed-integer problem. A novel functional split orchestration scheme that aims at minimizing the RAN deployment cost with deterministic user demands while considering the requirements of its processing network functions and the capabilities of the cloud infrastructure is studied in [73]. The capability of massive multiple-input multiple-output (mMIMO) systems to support the minimum throughput requirement of as many users as possible is investigated in [74], where a new user service scheduling scheme for multi-cell mMIMO is proposed to support a large user population. In [76], the impact of backhaul-RAN coupling on the performance of small cell networks is studied where mmWave is used to convey backhaul by gateways and fixed user requirements are considered. Next, we review papers in which user demands are uncertain. Resource reservation for the uncertain demand was first studied by Gomory and Hu in [80], which reserved link capacities using a single commodity routing problem with a finite number of sources. In traffic oblivious approaches, to make reservations and slice the network resources, user demand and its statistics are not considered in the problem formulation [81, 82, 83]. The drawback of traffic oblivious approaches is that they limit the ability of a network to adapt to any given demand. To reserve link capacities in flow networks, a collection of predicted demand scenarios are considered in [14, 7]. The proposed algorithms in [14, 7] reserve link capacities such that the predicted demand scenarios are supported as much as possible. The accuracy of the reservations in [14, 7] is based on the number of predicted scenarios. However, as the number of scenarios increases, the complexity of solving the optimization problem for the link capacity reservation increases. Short term user demands are predicted by Long Short-Term Memory (LSTM) neural networks in [8, 9, 10]. Recurring resource reservations based on the short-term traffic variations incur reconfiguration costs, service interruptions, and overhead in networks [11]. The mean of user demands is used in [84] to balance the workload among a set of data centers in a network that consists of the backhaul and RAN such that the utilization of resources is maximized. The joint reservation of computational and radio resources is studied in [12], where different ranges are considered for uncertain user demands. A linear program is formulated in [12] to support the uncertain user demands, which vary in given ranges, as much as the network allows. In [85], the transmission resource reservation in RAN is considered where the minimum requirements of users are known and deterministic. The authors of [85] proposed a matching-based algorithm to solve an optimization problem with the goal of minimizing the consumption of network resources while meeting the requirements of users.

Optimal routing is studied widely for many settings, e.g., [86, 87, 88], while optimal resource allocation in RAN has also been studied for different wireless channels, e.g., [9, 89, 90, 91, 92, 93, 94, 85]. The joint routing in the backhaul network and resource allocation RAN is studied in a number of more recent papers [5, 59, 95, 96, 75, 97, 98, 99]. In [5] and [75], the user demand requirements are deterministic and known. On the other hand, in [59, 96, 99, 97, 98], the traffic of users is maximized as much as the network is able to support, regardless of user demand statistics. To find a robust resource reservation, network resources should be reserved based on demand statistics. In [59, 95, 96] and [99, 97, 98], the wireless channel capacity is a deterministic function of input power. Moreover, the convexity of the problem is assumed in [59, 95, 96, 98]. Neither of these assumptions holds in practice, where the wireless channel capacity is random and its distribution is a function of supplied transmission resources [100, 101, 102].

In addition to different proposed formulations for resource allocations and network planning with certain and uncertain user demands in existing literature, several algorithms have been used to solve the resulting optimization problems. Among them, the ADMM has been used widely [7, 3, 5, 103, 6, 104, 105]. ADMM enables flow decoupling in the network optimization process. The efficiency of ADMM depends on the number of auxiliary link variables introduced to make the optimization subproblems separable. For networks with a large number of links, ADMM can be slow, i.e., requiring a large number of iterations. A dual decomposition method for path-based routing is used in [106], where a gradient ascent approach has been proposed to solve the dual problem. Since in most problems the dual function is non-smooth, the gradient ascent approach has to take small steps, resulting in slow convergence. A distributed approach for large-scale revenue management problems in airline networks is proposed by Kemmer *et al.* in [13]. The single-path dynamic programming approach in [13] has shown great success in practice despite the absence of convergence or solution enhancement guarantees.

3.2 Contributions

In this chapter, we propose a resource management scheme for end-to-end resource reservation, i.e., from data centers to users for a data network consisting of the backhaul and RAN, where user demands and achievable rates of downlinks are unseen and uncertain. Unlike [12, 14, 7], we consider that neither the ranges for user demands and achievable rates of downlinks, nor future demand scenarios are available. Instead, we utilize statistics for user demands and downlink achievable rates to obtain resource reservations for users. We consider a multi-path routing in our formulation, where a user can be served by several Access Points (APs) through multiple paths from a data center. We formulate the problem of jointly reserving the transmission resources in RAN and link capacities in the backhaul based on user demand and downlink achievable rate statistics so as to maximize the total expected supportable user traffic, while minimizing the expected outage of downlinks. Since the formulated problem is non-convex and hard to solve, we propose an efficient BCD algorithm, which is convergent to a Karush–Kuhn–Tucker (KKT) solution of the resource reservation problem.

In the proposed BCD approach, one block of variables determines the link capacity reservation in the backhaul and the other block of variables specifies the transmission resource reservation in RAN. We alternately optimize the two blocks of variables in the BCD algorithm. Fixing the transmission resources in RAN, we update the link capacity reservation in the backhaul via a novel multi-path routing algorithm. Inspired by the resource level decomposition ideas in [13], the proposed multi-path routing decomposes the problem down to link-level and parallelizes the computation across backhaul links. Based on the convergence theory for BSUM methods in [19], we prove that the proposed multi-path routing is convergent to the global minima of an arbitrary convex cost function with Lipschitz continuous gradient. The required computation time for each iteration of the proposed

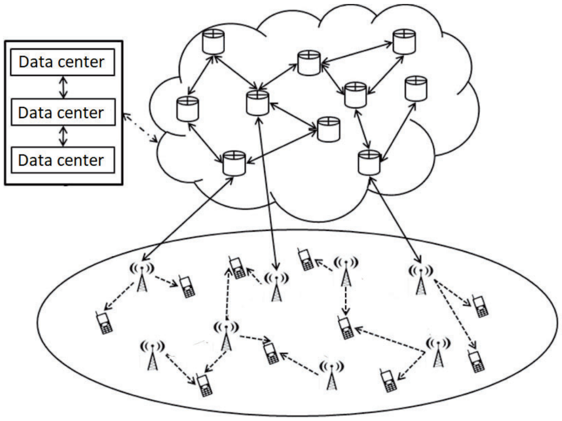


Figure 3.1: A network comprised of APs and backhaul parts.

multi-path routing is equal to that for one link regardless of the network size. After updating the link capacity reservations, we update the transmission resource reservation in RAN. Since the resource reservation problem in RAN is possibly non-convex, we propose a distributed algorithm based on the BSUM techniques to iteratively solve a sequence of convex approximations of the original problem. We prove that the proposed BCD algorithm converges to a KKT solution. To verify the performance of the proposed algorithm, two heuristic algorithms are also developed and used as benchmarks to evaluate the efficiency and the efficacy of the proposed approach via simulations.

The rest of this chapter is organized as follows. The system model and problem formulation are given in Section 3.3. Section 3.4 describes a general scalable and distributed algorithm for the multi-path flow routing. In Section 3.5, we propose a BCD algorithm for the network resource reservation problem. The simulation results are given in Section 3.6, and concluding remarks are given in Section 3.7.

3.3 System Model and Problem Formulation

Consider a typical scenario whereby user data is transmitted via backhaul network links from data centers to APs in RAN, which in turn relay the data to the desired users as depicted in Fig. 3.1. Suppose $\mathcal{B}$ denotes the set of APs and $\mathcal{K}$ denotes the set of mobile users. The set of directed wired links of the backhaul is denoted by $\mathcal{L}$. A path connects a data center and an AP through a sequence of wired links in the backhaul and finally goes through one downlink to reach the end user. The downlinks between APs and users are predetermined according to channel quality, interference levels, and path loss.

We consider each user demands one commodity and there are $K = |\mathcal{K}|$ datastreams in the backhaul network. The proposed scheme can be easily extended to the scenario that each user demands multiple commodities. To serve each user, several candidate paths are selected between the origin and destination, and traffic reservation for the corresponding commodities is implemented over those paths. The candidate paths can go through different APs. The requested traffic of a user splits among multiple paths, and the joint transmission of APs to a user via multi-connectivity is considered in this chapter. Different paths for one user carry different information flows. Only the last hop on each path is wireless.

Each path is denoted by p, and the set of all paths is represented by $\mathcal{P}$. The set of paths that carry user k data is denoted by $\mathcal{P}_k$. The backhaul network links comprising path p for serving user k are represented by the set $\mathcal{L}_k^p$. Similarly, the network nodes on path $p \in \mathcal{P}_k$ are denoted by the set $\mathcal{U}_k^p$. The demand of user k is a random variable represented by d_k. It follows a certain PDF denoted by $f_k(d_k)$. The corresponding Cumulative Density Function (CDF) is represented by $F_k(d_k)$. Let r_k denote the traffic rate reserved for user k. The actual traffic flow of user k supported by the network is a random variable given by

$$\min(d_k, r_k) = \begin{cases} r_k, & \text{if} \;\; r_k \leq d_k, \\ \\ d_k, & \text{otherwise.} \end{cases}$$

We calculate the expected supportable traffic rate for user k as follows:

$$\mathbb{E}\left(\min(d_k, r_k)\right) = \int_0^{r_k} y_k . f_k(y_k) dy_k + r_k \int_{r_k}^{\infty} f_k(y_k) dy_k.$$

Since the network is not able to support the demand when it exceeds the reserved rate, we have the minimum in the above expectation. In the first integral, the random demand of user k falls below the reserved rate. In the second integral, the random demand exceeds r_k.

Since a user receives their data from multiple APs, transmission resources should be reserved in multiple APs for the paths available to the user. The resource reservation in the backhaul and RAN is limited by two physical constraints:

- The aggregate reserved traffic rate for paths that share a link must not exceed the available link capacity. Therefore, we have the following constraint:

$$\sum_{k=1}^{K} \sum_{p:\{p\in\mathcal{P}_k, l\in\mathcal{L}_k^p\}} r_k^p \leq C_l, \qquad \forall l \in \mathcal{L}, \tag{3.1}$$

  where r_k^p is the reserved traffic rate for path p (for serving user k). Moreover, the available capacity of link l is denoted by C_l. Flows on different paths available to one user are treated as separate flows. Thus, we have the inner summation in the above constraint.

- The total reserved transmission resources for different paths must not exceed the available AP capacity. Hence, we have

$$\sum_{k=1}^{K} \sum_{p:\{p\in\mathcal{P}_k, b\in\mathcal{U}_k^p\}} t_k^p \leq C_b, \qquad \forall b \in \mathcal{B}, \tag{3.2}$$

  where t_k^p is the reserved transmission resources in AP b to transmit incoming data from path $p \in \mathcal{P}_k$ to user k. Moreover, the available capacity of AP b is denoted by C_b.

In our formulations, we reserve the available capacity of a backhaul link or the available transmission resource of an AP. If a portion of the backhaul link capacity or AP transmission resource is already reserved or in use, we do not consider that as the available resource to reserve. We consider backhaul links have different capacities available to be reserved. When a link has a large capacity available, it is able to provide high rates to paths that go through that link. This can result in increasing reserved user rates. Similarly, we consider APs have different transmission resource capacities available to be reserved. When an AP has more transmission resource available compared to another AP, it can better serve its users. The reason is that it can allocate more transmission resources to its users and better combat wireless channel fluctuations. In addition to the above physical constraints, our multi-path model enforces another constraint. Since each datastream originating from a data center splits into a number of sub-flows, we have the following constraint:

- The aggregate reserved traffic rate for the different paths, which carry different information flows, to one user is equal to the reserved rate for that user. Hence, we have the following constraint:

$$\sum_{p \in \mathcal{P}_k} r_k^p = r_k, \qquad \forall k. \tag{3.3}$$

In the considered model, we do not make any assumption about the type of the transmission resource. It can be bandwidth, transmission power, or time-slot fraction. We consider time-varying downlinks, where channel state information is not available at transmitting APs and power adaptation is not used by APs. Therefore, the received SNR at each user depends on the channel realization and is a random variable. As the received SNR at each user also depends on the amount of transmission resources, e.g., power or bandwidth, supplied to the downlink (see [107, eq. (4.4)]), the achievable rate of a downlink has a PDF, which is determined by the channel fading type and transmission resources. As only the last hop on each path is wireless, path p uniquely identifies the downlink of the last hop. The achievable rate (i.e., instantaneous capacity) of the downlink of path p is random and follows an arbitrary distribution with a PDF represented by $z_k^p(v_k^p, t_k^p)$ and a CDF denoted by $Z_k^p(v_k^p, t_k^p)$. The PDF is a function of two variables: the achievable rate of the downlink, denoted by v_k^p, and the allocated transmission resource, denoted by t_k^p. When the achievable rate of a downlink falls below the reserved rate r_k^p, some outage is experienced and its amount is $r_k^p - v_k^p$, given that the amount of allocated transmission resources to the downlink is t_k^p. The probability that this amount of outage takes place is $z_k^p(v_k^p, t_k^p)$. In light of the above arguments, the expected outage of the downlink of path p is obtained as follows:

$$\int_0^{r_k^p} z_k^p(v_k^p, t_k^p)\,(r_k^p - v_k^p) dv_k^p. \tag{3.4}$$

Since the achievable rate is a continuous random variable, we have the above integral.

In this chapter, we aim to maximize the expected traffic of users as much as the network is able to support, while minimizing the expected outage of downlinks. We formulate the

Table 3.1: Notations

k	a user
p	a path
l	a backhaul link
b	an AP
$\mathcal{L}$	the set of backhaul links
$\mathcal{B}$	the set of APs
$\mathcal{L}_k^p$	set of links on path p to serve user k
$\mathcal{U}_k^p$	set of nodes on path p to serve user k
r_k	reserved rate for user k
r_k^p	reserved rate for path p to serve user k
w	a downlink
d_k	random demand of user k
t_k^p	reserved transmission resource for path p to serve user k
$f_k(d_k)$	PDF for user k demand
$\mathcal{P}_k$	the set of paths that carry user k data
$z_k^p(v_k^p, t_k^p)$	PDF of the achievable rate of downlink of path p
K	the number of users in the network
$\mathcal{W}_k$	the set of downlinks for serving user k
C_l	the capacity of backhaul link l
C_b	transmission resource budget for AP b
$\psi_k^p(r_k^p)$	a cost function
μ_l	the Lagrange multiplier for link l
$\alpha_{k,l}^{p,j}$	a constant that specifies decomposition in the j^{th} iteration
θ_k	a constant to balance expected outage minimization

following optimization problem to find resource reservations in the backhaul and RAN:

$$\begin{aligned} \max_{\mathbf{r},\mathbf{t}} \quad & \sum_{k=1}^{K}\Big[\mathbb{E}[\min(r_k,d_k)]-\theta_k\sum_{p\in\mathcal{P}_k}\int_0^{r_k^p} z_k^p(v_k^p,t_k^p)\,(r_k^p-v_k^p)dv_k^p\Big] \\ \text{s.t.} \quad & (3.1),(3.2),(3.3), r_k, r_k^p, t_k^p \geq 0, \quad p\in\mathcal{P}_k, \forall k, \end{aligned} \tag{3.5}$$

where $\theta_k : \theta_k \geq 0$ is a coefficient chosen by the system designer that adjusts the priorities of maximizing the expected supportable traffic of user k and the minimization of the aggregate outage of downlinks, which serve user k. As θ_k increases, the optimized expected outage decreases by reducing r_k^p or increasing t_k^p. When θ_k is a large value, r_k goes to zero to ensure that the expected outage tends to zero. The two blocks of variables in the above problem are $\mathbf{r} = \{r_k, r_k^p\}_{p\in\mathcal{P}_k, k=1:K}$ and $\mathbf{t} = \{t_k^p\}_{p\in\mathcal{P}_k, k=1:K}$. The summary of used notations is given in Table 3.1.

Remark 1. *Suppose that multiple paths available to user k share a downlink (the last hop). The aggregate outage of downlinks for serving user k is calculated as follows:*

$$\sum_{w\in\mathcal{W}_k}\int_0^{\sum_{p:\{p\in\mathcal{P}_k, w\in p\}} r_k^p} z_k^w(v_k^w,t_k^w)\times(\sum_{p:\{p\in\mathcal{P}_k,w\in p\}} r_k^p - v_k^w)dv_k^w, \tag{3.6}$$

where $\mathcal{W}_k$ is the set of downlinks, each denoted by w, for serving user k. When multiple paths available to user k share a downlink, the above outage is placed in the objective function of (3.5) *instead of its second term, which includes* (3.4).

The maximization problem (3.5) is not easy to solve to global optimality. The objective function of (3.5) is in general not necessarily jointly concave in $\mathbf{r}$ and $\mathbf{t}$ for an arbitrary PDF $z_k^p(v_k^p, t_k^p)$. The reason is that $\int_0^{r_k^p} \partial^2 z_k^p(v_k^p,t_k^p)/(\partial t_k^p)^2\,(r_k^p - v_k^p)dv_k^p$ is not always non-negative.

Proposition 2. *Given* $\mathbf{t}$*, the optimization in* (3.5) *becomes concave in* $\mathbf{r}$.

Proof. Fixing $\mathbf{t}$, the objective function is separable in k. We find the Hessian with respect to r_k and $\{r_k^p\}_{p\in\mathcal{P}_k}$ for those objective function terms which are associated with user k as follows:

$$\mathbf{H}_k = \begin{pmatrix} -f_k(r_k) & 0 & \dots & 0 \\ 0 & -z_k^1(r_k^1, t_k^1) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & -z_k^{|\mathcal{P}_k|}(r_k^{|\mathcal{P}_k|}, t_k^{|\mathcal{P}_k|}) \end{pmatrix}.$$

The overall Hessian matrix is

$$\mathbf{H} = \begin{pmatrix} \mathbf{H}_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{H}_K \end{pmatrix}.$$

It is observed that the above matrix is negative semidefinite. Since the constraints of problem (3.5) are all affine, it follows that the maximization (3.5) is concave with fixed $\mathbf{t}$. □

Separable constraints on $\mathbf{r}$ and $\mathbf{t}$ in (3.5) motivate the BCD algorithm. It is straightforward to show that with (3.6) instead of (3.4) in the objective function, the optimization in (3.5) remains concave in $\mathbf{r}$.

3.4 Distributed Multi-Path Routing for Flow Networks

This section is concerned with solving (3.5) when $\mathbf{t}$ is kept fixed, and (3.5) is converted to the minimization format after multiplying the objective function by -1. In particular, we study a general multi-path routing to minimize any convex cost function with a Lipschitz continuous gradient. We develop an algorithm that is dual-based and decomposes the problem down to link-level and parallelizes computations across links of the network. The required computation time for each iteration of the proposed multi-path routing algorithm is equal to that for one link regardless of the network size. This interesting property makes the proposed algorithm appropriate for the online optimization of large networks.

For each datastream in the network, several candidate paths are selected. We assume that each flow can be split into multiple sub-flows. To formulate the multi-path routing problem, we first assume that the cost function is separable in variables, i.e., $\psi(\mathbf{r}) = \sum_{k=1}^{K} \sum_{p \in \mathcal{P}_k} \psi_k^p(r_k^p)$, where each $\psi_k^p(r_k^p)$ is strictly convex.

The optimization problem for the multi-path flow routing can be written as follows:

$$\begin{aligned} \min_{\mathbf{r}} \quad & \sum_{k=1}^{K} \sum_{p \in \mathcal{P}_k} \psi_k^p(r_k^p) \\ \text{s.t.} \quad & (3.1), r_k^p \geq 0, \quad p \in \mathcal{P}_k, \forall k. \end{aligned} \tag{3.7}$$

Since typically the number of variables is greater than the number of constraints in the above optimization, solving the problem is easier in the dual domain. The Lagrangian function for the above problem is

$$L_c(\mathbf{r}, \boldsymbol{\mu}, \boldsymbol{\phi}) = \sum_{k=1}^{K} \sum_{p \in \mathcal{P}_k} \psi_k^p(r_k^p) + \sum_{l \in \mathcal{L}} \mu_l (\sum_{k=1}^{K} \sum_{p:\{p \in \mathcal{P}_k, l \in \mathcal{L}_k^p\}} r_k^p - C_l) - \sum_{k=1}^{K} \sum_{p \in \mathcal{P}_k} \phi_k^p r_k^p, \tag{3.8}$$

where $\mu_l : \mu_l \geq 0$ is the Lagrange multiplier for the capacity constraint of link l, and $\phi_k^p : \phi_k^p \geq 0$ is the Lagrange multiplier for constraint $r_k^p \geq 0$. Furthermore, $\boldsymbol{\mu} = \{\mu_l\}_{l \in \mathcal{L}}$ and $\boldsymbol{\phi} = \{\phi_k^p\}_{p \in \mathcal{P}_k, k=1:K}$. We find the dual problem of (3.7) as follows:

$$\begin{aligned} \max_{\boldsymbol{\mu}, \boldsymbol{\phi}} \quad & \min_{\mathbf{r}} L_c(\mathbf{r}, \boldsymbol{\mu}, \boldsymbol{\phi}) \\ \text{s.t.} \quad & \boldsymbol{\mu} \geq \mathbf{0}, \boldsymbol{\phi} \geq \mathbf{0}. \end{aligned} \tag{3.9}$$

For many cost functions, no closed-form solution for $\mathbf{r} = \arg\min_{\mathbf{r}} L_c(\mathbf{r}, \boldsymbol{\mu}, \boldsymbol{\phi})$ exists. Therefore, commonly, the above problem is solved via a primal-dual method such as ADMM [7, 3, 103, 6]. However, the auxiliary link variables introduced to make the per-flow subproblems of optimization in (3.7) separable can slow down ADMM in practice.

Resource level decomposition for large-scale single-path applications was first proposed in [13] to solve the revenue management problems in airline networks. The proposed decomposition in [13] does not involve any auxiliary variables. In spite of the absence of convergence or solution enhancement guarantees, the resource level decomposition has been rather successful in practice. We leverage resource level decomposition ideas to develop a distributed algorithm to solve the general multi-path routing problem (3.7) in a parallel fashion such that the traffic passing on each link can be obtained independently from the other links. Unlike the dynamic programming approach in [13], an optimization-based approach is proposed here to solve subproblems. In each iteration, the proposed dual algorithm decomposes the problem in (3.9) and solves the subproblems globally and in parallel. The optimized μ_l in the j^{th} iteration of the proposed algorithm is denoted by μ_l^j. Here, we explain the decomposition. The dualized link capacity constraints $\sum_{l\in\mathcal{L}}\mu_l(\sum_{k=1}^{K}\sum_{p:\{p\in\mathcal{P}_k,l\in\mathcal{L}_k^p\}} r_k^p - C_l)$ in the Lagrangian (3.8) are separable across links. Each link l receives $\mu_l(\sum_{k=1}^{K}\sum_{p:\{p\in\mathcal{P}_k,l\in\mathcal{L}_k^p\}} r_k^p - C_l)$. In each iteration, based on $\boldsymbol{\mu}^{j-1} = \{\mu_l^{j-1}\}_{l\in\mathcal{L}}$ in the previous iteration, we decompose the non-separable terms in the Lagrangian (3.8), which include r_k^p, across links on path p. Each link l of path p receives a portion of

$$\alpha_{k,l}^{p,j} = \mu_l^{j-1} / \sum_{l'\in\mathcal{L}_k^p} \mu_{l'}^{j-1}, \tag{3.10}$$

In the j^{th} iteration, the decomposed per-link Lagrangian function is as follows:

$$\begin{aligned} L_l(\mathbf{r}_l, \mu_l, \boldsymbol{\phi}_l, \boldsymbol{\mu}^{j-1}) = &\sum_{k=1}^{K} \sum_{p:\{p\in\mathcal{P}_k,l\in\mathcal{L}_k^p\}} \alpha_{k,l}^{p,j}\psi_k^p(r_k^p) \\ &+ \mu_l(\sum_{k=1}^{K} \sum_{p:\{p\in\mathcal{P}_k,l\in\mathcal{L}_k^p\}} r_k^p - C_l) - \sum_{k=1}^{K} \sum_{p:\{p\in\mathcal{P}_k,l\in\mathcal{L}_k^p\}} \alpha_{k,l}^{p,j}\phi_k^p r_k^p, \end{aligned} \tag{3.11}$$

where $\mathbf{r}_l = \{r_k^p\}_{p\in\mathcal{P}_k,l\in\mathcal{L}_k^p,k=1:K}$ and $\boldsymbol{\phi}_l = \{\phi_k^p\}_{p\in\mathcal{P}_k,l\in\mathcal{L}_k^p,k=1:K}$. We notice that based on (3.10), $\{\alpha_{k,l}^{p,j}\}_{l\in\mathcal{L}_k^p}$ in (3.11) is calculated using $\boldsymbol{\mu}^{j-1}$. Based on the above decomposition, we obtain

$$L_c(\mathbf{r}, \boldsymbol{\mu}, \boldsymbol{\phi}) = \sum_{l\in\mathcal{L}} L_l(\mathbf{r}_l, \mu_l, \boldsymbol{\phi}_l, \boldsymbol{\mu}^{j-1}). \tag{3.12}$$

Instead of solving the problem in (3.9), we solve

$$\begin{aligned} \max_{\{\mu_l,\boldsymbol{\phi}_l\}_{l\in\mathcal{L}}} \quad & \sum_{l\in\mathcal{L}} \min_{\mathbf{r}_l} L_l(\mathbf{r}_l, \mu_l, \boldsymbol{\phi}_l, \boldsymbol{\mu}^{j-1}) \\ \text{s.t.} \quad & \mu_l \geq 0, \boldsymbol{\phi}_l \geq \mathbf{0}, \quad l \in \mathcal{L}, \end{aligned} \tag{3.13}$$

iteratively and then update $\alpha_{k,l}^{p,j+1}$ for iteration $j+1$. The above problem is decomposable in $\{\mu_l, \boldsymbol{\phi}_l\}$ and can be solved in parallel for all links. Due to strong duality [108, p. 226–p. 227], each subproblem of (3.13) is equivalent to the following per-link problem in the primal domain:

$$\begin{aligned} \min_{\mathbf{r}_l} \quad & \sum_{k=1}^{K} \sum_{p:\{p\in\mathcal{P}_k, l\in\mathcal{L}_k^p\}} \alpha_{k,l}^{p,j} \psi_k^p(r_k^p) \\ \text{s.t.} \quad & \sum_{k=1}^{K} \sum_{p:\{p\in\mathcal{P}_k, l\in\mathcal{L}_k^p\}} r_k^p \le C_l, \\ & \alpha_{k,l}^{p,j} r_k^p \ge 0, \quad p\in\mathcal{P}_k, \forall k, l\in\mathcal{L}_k^p. \end{aligned} \tag{3.14}$$

The optimal r_k^p and μ_l can be obtained using the first-order optimality condition for the per-link subproblem in (3.14). Here, we list KKT conditions as follows:

$$\frac{\partial L_l(\mathbf{r}_l, \mu_l, \boldsymbol{\phi}_l, \boldsymbol{\mu}^{j-1})}{\partial r_k^p} = \alpha_{k,l}^{p,j} \frac{\partial \psi_k^p(r_k^p)}{\partial r_k^p} + \mu_l - \alpha_{k,l}^{p,j} \phi_k^p = 0, \tag{3.15a}$$

$$\sum_{k=1}^{K} \sum_{p:\{p\in\mathcal{P}_k, l\in\mathcal{L}_k^p\}} r_k^p \le C_l, \tag{3.15b}$$

$$\mu_l \,(\sum_{k=1}^{K} \sum_{p:\{p\in\mathcal{P}_k, l\in\mathcal{L}_k^p\}} r_k^p - C_l) = 0, \;\; \mu_l \ge 0, \tag{3.15c}$$

$$\alpha_{k,l}^{p,j} r_k^p \phi_k^p = 0, \;\; r_k^p \ge 0, \;\; \phi_k^p \ge 0. \tag{3.15d}$$

First, we consider that $r_k^p > 0$ and $\phi_k^p = 0$. Due to the strict convexity of $\psi_k^p(r_k^p)$, $\partial\psi_k^p(r_k^p)/\partial r_k^p$ is strictly increasing. Thus, given μ_l, there is a unique r_k^p to solve $\alpha_{k,l}^{p,j}\partial\psi_k^p(r_k^p)/\partial r_k^p + \mu_l = 0$. Since $\partial\psi_k^p(r_k^p)/\partial r_k^p$ is strictly increasing, we implement a bisection search (see [109, 110]) on r_k^p in the non-negative orthant $r_k^p \ge 0$ to find r_k^p from $\alpha_{k,l}^{p,j}\partial\psi_k^p(r_k^p)/\partial r_k^p + \mu_l = 0$. If the obtained r_k^p is positive, we keep $\phi_k^p = 0$. Otherwise, we set $r_k^p = 0$ and find $\phi_k^p = \partial\psi_k^p(r_k^p)/\partial r_k^p|_{r_k^p=0} + \mu_l/\alpha_{k,l}^{p,j}$. For a given μ_l, we obtain each r_k^p variable associated with link l, i.e., $r_k^p : p\in\mathcal{P}_k, l\in\mathcal{L}_k^p, k = 1:K$. The dual approach for solving the optimization in (3.14) works as follows: implement a bisection search on the Lagrange multiplier μ_l in the positive orthant and numerically find each r_k^p variable from (3.15a) and (3.15d) for each μ_l until we have $\sum_{p:\{p\in\mathcal{P}_k, l\in\mathcal{L}_k^p\}} r_k^p = C_l$. If there is no such positive μ_l, we drop

Algorithm 7: Per-link dual algorithm to solve the optimization in (3.14)

```
0. **Initialization** $s_1 = 0$, $s_2 =$ large number, $q_1 = 0$, $q_2 = 0$, $q_3 = 0$;
**repeat**
    1. $s_3 = (s_1 + s_2)/2$;
    2. Implement a bisection search to solve (3.15a) with $\phi_k^p = 0$ and find
       $r_k^p : r_k^p \geq 0$, where $\mu_l = s_1$;
    3. **if** *there is no positive solution for* $r_k^p$ **then**
        $r_k^p = 0$ and $\phi_k^p = \partial\psi_k^p(r_k^p)/\partial r_k^p|_{r_k^p=0} + \mu_l/\alpha_{k,l}^{p,j}$;
    4. $q_1 = \sum_{k=1}^K \sum_{p:\{p \in \mathcal{P}_k, l \in \mathcal{L}_k^p\}} r_k^p - C_l$;
    5. Implement a bisection search to solve (3.15a) with $\phi_k^p = 0$ and find
       $r_k^p : r_k^p \geq 0$, where $\mu_l = s_2$;
    6. **if** *there is no positive solution for* $r_k^p$ **then**
        $r_k^p = 0$ and $\phi_k^p = \partial\psi_k^p(r_k^p)/\partial r_k^p|_{r_k^p=0} + \mu_l/\alpha_{k,l}^{p,j}$;
    7. $q_2 = \sum_{k=1}^K \sum_{p:\{p \in \mathcal{P}_k, l \in \mathcal{L}_k^p\}} r_k^p - C_l$;
    8. Implement a bisection search to solve (3.15a) with $\phi_k^p = 0$ and find
       $r_k^p : r_k^p \geq 0$, where $\mu_l = s_3$;
    9. **if** *there is no positive solution for* $r_k^p$ **then**
        $r_k^p = 0$ and $\phi_k^p = \partial\psi_k^p(r_k^p)/\partial r_k^p|_{r_k^p=0} + \mu_l/\alpha_{k,l}^{p,j}$;
    10. $q_3 = \sum_{k=1}^K \sum_{p:\{p \in \mathcal{P}_k, l \in \mathcal{L}_k^p\}} r_k^p - C_l$;
    11. **if** $q_1.q_3 < 0$ **then**
        $s_2 = s_3$;
    12. **if** $q_2.q_3 < 0$ **then**
        $s_1 = s_3$;
    13. **if** $q_1 < 0,\ q_2 < 0,\ q_3 < 0$ **then**
        13.1. $\mu_l = 0$;
        13.2. Solve $\partial\psi_k^p(r_k^p)/\partial r_k^p = 0$ to find $r_k^p$;
        13.3. Project the obtained $r_k^p$ variable to the positive orthant;
        13.4. $s_2 = s_1$;
**until** $s_2 - s_1$ *is small enough*;
```

the first constraint from optimization (3.14) and solve (3.14) by setting the gradient of the cost function to zero. Then, we project the solution to the positive orthant. Due to the strict convexity of each subproblem, the optimal primal variables are unique. The optimal μ_l for each per-link subproblem is also unique. We justify this claim. If the link capacity constraint is not tight, then due to (3.15c), μ_l has to be zero. If the link capacity is tight,

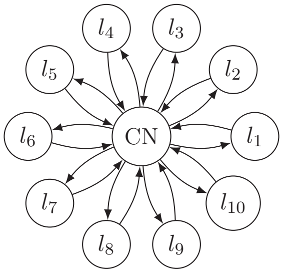


Figure 3.2: Message passing between 10 links and one CN.

then at least one $r_k^p : p \in \mathcal{P}_k, l \in \mathcal{L}_k^p, k = 1 : K$ is non-zero and $\phi_k^p = 0$. Due to a) the strict convexity of $\psi_k^p(r_k^p)$ and the monotone variation of $\psi_k^p(r_k^p)/\partial r_k^p$; and b) the uniqueness of the optimal r_k^p, the obtained μ_l from (3.15a) is unique. We justify the bisection search on μ_l as follows: if the unique optimal μ_l is positive, from (3.15c), we observe that we must have $\sum_{p:\{p\in\mathcal{P}_k, l\in\mathcal{L}_k^p\}} r_k^p = C_l$, where each r_k^p is found from (3.15a) and (3.15d). Such positive μ_l can be uniquely found using a bisection search due to the strictly monotone variation of $\sum_{p:\{p\in\mathcal{P}_k, l\in\mathcal{L}_k^p\}} r_k^p$, with μ_l (strict convexity of $\psi_k^p(r_k^p)$ as explained above). If the optimal μ_l is zero, then (3.15b) and (3.15c) are already satisfied and it is enough to find the unique non-negative minimizer of each $\psi_k^p(r_k^p)$ from (3.15a) and (3.15d). In light of the above arguments, two nested bisection methods are required to solve (3.14): the inner bisection works on r_k^p and the outer one works on μ_l. The summary of the proposed bisection approach to solve the per-link optimization in (3.14) is given in Algorithm 7.

Suppose that the optimization in (3.14) is iteratively solved in parallel for all links of the network. For a link with a large available capacity, the link capacity constraint is not tight and Algorithm 7 finds $\mu_l^j = 0$ and we have $\alpha_{k,l}^{p,j+1} = 0$. For those links, we do not need to continue computation as the KKT conditions listed in (3.15a)–(3.15d) remain satisfied. In the following iterations, we ignore those links and consider links with $\mu_l^j > 0$. We alternate between solving the optimization in (3.14) in parallel for all links and updating $\alpha_{k,l}^{p,j+1}$ until all $\{\mu_l^j\}_{l\in\mathcal{L}}$ variables converge, i.e., $\left\|\boldsymbol{\mu}^j - \boldsymbol{\mu}^{j-1}\right\|_2 < \epsilon$. This is equivalent to have a Central Node (CN) in the network that coordinates the problem decomposition across backhaul links. Each link independently solves its subproblem and sends the optimized dual variable back to the CN. Next, the CN computes $\alpha_{k,l}^{p,j}$ and sends that to link l. This message passing interface is depicted in Fig. 3.2. Once $\boldsymbol{\mu}^j$ converges, for each r_k^p variable, we use the

Algorithm 8: Decentralized multi-path routing algorithm to solve the optimization in (3.7)

```
0. Initialization Assign some small positive number to each $\mu_l^0$, $j = 0$;
repeat
    for all links do
        1. Find $\alpha_{k,l}^{p,j+1} = \mu_l^j / \sum_{l' \in \mathcal{L}_k^p} \mu_{l'}^j$;
        if $\mu_l^j > 0$ then
            2. Apply Algorithm 7 to find $\mu_l^{j+1}$;
        3. $j = j + 1$;
until $\boldsymbol{\mu}^j$ converge;
for all $\{r_k^p\}_{p \in \mathcal{P}_k, k=1:K}$ variables do
    4. Use the latest computed $r_k^p$ by Algorithm 7 from a per-link subproblem,
       where $l \in \mathcal{L}_k^p$ and $\mu_l^j > 0$;
```

computed r_k^p in the last iteration of Algorithm 7 from a subproblem with $\mu_l^j > 0, l \in \mathcal{L}_k^p$. A brief description of the proposed dual algorithm for solving the optimization in (3.7) is given in Algorithm 8.

After Algorithm 8 converges, we use the obtained r_k^p from a per-link problem with tight link capacity constraint, i.e., $l : \mu_l^j > 0$, for the other links on that path for which Algorithm 8 finds $\mu_l^j = 0$. The key property of Algorithm 8 is that after convergence, the obtained r_k^p on different links of one path are identical.

Proposition 3. *Upon convergence of Algorithm 8, the flow rates across links on each path are identical.*

Proof. Algorithm 8 finds r_k^p, from the per-link subproblem for link $l \in \mathcal{L}_k^p, \mu_l^j > 0$, using the following equation:

$$\begin{aligned}\frac{\partial L_l(\mathbf{r}_l, \mu_l^j, \boldsymbol{\phi}_l^j, \boldsymbol{\mu}^{j-1})}{\partial r_k^p} &= \alpha_{k,l}^{p,j} \frac{\partial \psi_k^p(r_k^p)}{\partial r_k^p} + \mu_l^j - \alpha_{k,l}^p \phi_k^{p,j} \\ &= \frac{\mu_l^{j-1}}{\sum_{l' \in \mathcal{L}_k^p} \mu_{l'}^{j-1}} \frac{\partial \psi_k^p(r_k^p)}{\partial r_k^p} + \mu_l^j - \frac{\mu_l^{j-1}}{\sum_{l' \in \mathcal{L}_k^p} \mu_{l'}^{j-1}} \phi_k^{p,j} = 0. \end{aligned} \tag{3.16}$$

Suppose Algorithm 8 has converged in the j^{th} iteration; we have $\left\|\boldsymbol{\mu}^j - \boldsymbol{\mu}^{j-1}\right\|_2 < \epsilon$. Then, we have $\boldsymbol{\mu}^j - \boldsymbol{\mu}^{j-1} = \boldsymbol{\vartheta}$, where $\|\boldsymbol{\vartheta}\|_2 < \epsilon$. We have

$$\frac{1}{\alpha_{k,l}^{p,j}} = \frac{\sum_{l' \in \mathcal{L}_k^p} \mu_{l'}^{j-1}}{\mu_l^{j-1}} = \frac{\sum_{l' \in \mathcal{L}_k^p} (\mu_{l'}^j - \vartheta_{l'})}{\mu_l^j - \vartheta_l}.$$

We multiply (3.16) by $1/\alpha_{k,l}^{p,j}$ and we have the following:

$$\begin{aligned}
&\frac{\partial \psi_k^p(r_k^p)}{\partial r_k^p} + \frac{\sum_{l' \in \mathcal{L}_k^p} (\mu_{l'}^j - \vartheta_{l'})}{\mu_l^j - \vartheta_l} \mu_l^j - \phi_k^{p,j} \\
&= \frac{\partial \psi_k^p(r_k^p)}{\partial r_k^p} + \sum_{l' \in \mathcal{L}_k^p} (\mu_{l'}^j - \vartheta_{l'})(1 + \frac{\vartheta_l}{\mu_l^j - \vartheta_l}) - \phi_k^{p,j} \\
&= \underbrace{\frac{\partial \psi_k^p(r_k^p)}{\partial r_k^p} + \sum_{l \in \mathcal{L}_k^p} \mu_l^j - \phi_k^{p,j}}_{\partial L_c(\mathbf{r}, \boldsymbol{\mu}^j, \boldsymbol{\phi}^j)/\partial r_k^p} - \sum_{l' \in \mathcal{L}_k^p} \vartheta_{l'}(1 + \frac{\vartheta_l}{\mu_l^j - \vartheta_l}) \\
&+ \frac{\vartheta_l(\sum_{l' \in \mathcal{L}_k^p} \mu_{l'}^j)}{\mu_l^j - \vartheta_l} = \frac{\partial L_c(\mathbf{r}, \boldsymbol{\mu}^j, \boldsymbol{\phi}^j)}{\partial r_k^p} - \sum_{l' \in \mathcal{L}_k^p} \vartheta_{l'}(1 + \frac{\vartheta_l}{\mu_l^j - \vartheta_l}) \\
&+ \frac{\vartheta_l(\sum_{l' \in \mathcal{L}_k^p} \mu_{l'}^j)}{\mu_l^j - \vartheta_l} = 0. \qquad (3.17)
\end{aligned}$$

When ϵ tends to zero, then $\boldsymbol{\vartheta} \to \mathbf{0}$ and from (3.17) we find $\partial L_c(\mathbf{r}, \boldsymbol{\mu}^j, \boldsymbol{\phi}^j)/\partial r_k^p = 0$. Moreover, we observe that $\partial L_c(\mathbf{r}, \boldsymbol{\mu}^j, \boldsymbol{\phi}^j)/\partial r_k^p$ is independent of the link index on path p. This means that $\{r_k^p\}_{p \in \mathcal{P}_k}$ variables obtained by solving the link subproblems are identical for all links along each path p for which $\mu_l^j > 0$. They are also equal to the minimizer of Lagrangian function $L_c(\mathbf{r}, \boldsymbol{\mu}^j, \boldsymbol{\phi}^j)$ in (3.8). □

Theorem 1. *If $\psi(\mathbf{r})$ is strictly convex and separable, then the primal and dual iterates of Algorithm 8 will converge to the optimal primal and dual solutions of* (3.7)*.*

Proof. Notice that based on the definition of $\alpha_{k,l}^{p,j}$, given identical feasible variables r_k^p, $\hat{\mu}_l$ and $\hat{\boldsymbol{\phi}}_l$ to both Lagrangian functions in (3.8) and (3.11), from (3.12), we have $\sum_{l \in \mathcal{L}} L_l(\mathbf{r}_l, \hat{\mu}_l, \hat{\boldsymbol{\phi}}_l, \boldsymbol{\mu}^{j-1}) = L_c(\mathbf{r}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\phi}})$. First, we show that $\sum_{l \in \mathcal{L}} \min_{\mathbf{r}_l} L_l(\mathbf{r}_l, \hat{\mu}_l, \hat{\boldsymbol{\phi}}_l, \boldsymbol{\mu}^{j-1})$ is a lower-bound for $\min_{\mathbf{r}} L_c(\mathbf{r}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\phi}})$. Since the minimum of $L_l(\mathbf{r}_l, \hat{\mu}_l, \hat{\boldsymbol{\phi}}_l, \boldsymbol{\mu}^{j-1})$ is less than or equal to the other values of $L_l(\mathbf{r}_l, \hat{\mu}_l, \hat{\boldsymbol{\phi}}_l, \boldsymbol{\mu}^{j-1})$, we have

$$\min_{\mathbf{r}_l} L_l(\mathbf{r}_l, \hat{\mu}_l, \hat{\boldsymbol{\phi}}_l, \boldsymbol{\mu}^{j-1}) \leq L_l(\mathbf{r}_l, \hat{\mu}_l, \hat{\boldsymbol{\phi}}_l, \boldsymbol{\mu}^{j-1}).$$

Thus, we obtain

$$\begin{aligned}
&\sum_{l\in\mathcal{L}} \min_{\mathbf{r}_l} L_l(\mathbf{r}_l, \hat{\mu}_l, \hat{\boldsymbol{\phi}}_l, \boldsymbol{\mu}^{j-1}) \\
&\leq \sum_{l\in\mathcal{L}} L_l(\mathbf{r}_l, \hat{\mu}_l, \hat{\boldsymbol{\phi}}_l, \boldsymbol{\mu}^{j-1}) = L_c(\mathbf{r}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\phi}}),
\end{aligned}$$

where the equality is due to (3.12). In $L_c(\mathbf{r}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\phi}})$, we choose $\mathbf{r}$ to be the minimizer of $L_c(\mathbf{r}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\phi}})$. Thus, we obtain

$$\sum_{l\in\mathcal{L}} \min_{\mathbf{r}_l} L_l(\mathbf{r}_l, \hat{\mu}_l, \hat{\boldsymbol{\phi}}_l, \boldsymbol{\mu}^{j-1}) \leq \min_{\mathbf{r}} L_c(\mathbf{r}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\phi}}). \tag{3.18}$$

From (3.18), we observe that solving the problem in (3.13) iteratively is a successive lower-bound maximization (upper-bound minimization if we rewrite problems (3.9) and (3.13) as minimizations).

We justify the claim that the primal and dual solutions obtained from solving (3.13) successively converge to the primal and dual solutions of (3.7). We build our proof based on the convergence theory for BSUM given in [19]. We show that the lower-bound satisfies all four convergence conditions given in [19, Assumption 2]:

1. At feasible points $\hat{\boldsymbol{\mu}} \geq \mathbf{0}$ and $\hat{\boldsymbol{\phi}} \geq \mathbf{0}$, we show that $\min_{\mathbf{r}} L_c(\mathbf{r}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\phi}}) = \sum_{l\in\mathcal{L}} \min_{\mathbf{r}_l} L_l(\mathbf{r}_l, \hat{\mu}_l, \hat{\boldsymbol{\phi}}_l, \hat{\boldsymbol{\mu}})$. From KKT conditions for each subproblem, we obtain

$$\begin{aligned}
\frac{L_l(\mathbf{r}_l, \hat{\mu}_l, \hat{\boldsymbol{\phi}}_l, \hat{\boldsymbol{\mu}})}{\partial r_k^p} &= \frac{\hat{\mu}_l}{\sum_{l'\in\mathcal{L}_k^p} \hat{\mu}_{l'}} \frac{\partial \psi_k^p(r_k^p)}{\partial r_k^p} + \hat{\mu}_l \\
&- \frac{\hat{\mu}_l}{\sum_{l'\in\mathcal{L}_k^p} \hat{\mu}_{l'}} \hat{\phi}_k^p = 0.
\end{aligned} \tag{3.19}$$

Assuming $\hat{\mu}_l > 0$, after multiplication by $\frac{\sum_{l' \in \mathcal{L}_k^p} \hat{\mu}_{l'}}{\hat{\mu}_l}$, we obtain

$$\frac{\partial \psi_k^p(r_k^p)}{\partial r_k^p} + \sum_{l \in \mathcal{L}_k^p} \hat{\mu}_l - \hat{\phi}_k^p = \frac{\partial L_c(\mathbf{r}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\phi}})}{\partial r_k^p} = 0. \tag{3.20}$$

When $\psi(\mathbf{r})$ is strictly convex, there is a unique minimizer for each $L_c(\mathbf{r}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\phi}})$ and $L_l(\mathbf{r}_l, \hat{\mu}_l, \hat{\boldsymbol{\phi}}_l, \hat{\boldsymbol{\mu}})$. We observe from (3.19) and (3.20) that, at feasible points $\hat{\boldsymbol{\mu}} \geq \mathbf{0}$ and $\hat{\boldsymbol{\phi}} \geq \mathbf{0}$, the minimizer of $L_c(\mathbf{r}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\phi}})$ is equal to that of $L_l(\mathbf{r}_l, \hat{\mu}_l, \hat{\boldsymbol{\phi}}_l, \hat{\boldsymbol{\mu}})$. From (3.12), applying identical variables r_k^p, $\hat{\mu}_l$ and $\hat{\phi}_l$ to both $L_c(\mathbf{r}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\phi}})$ and $L_l(\mathbf{r}_l, \hat{\mu}_l, \hat{\boldsymbol{\phi}}_l, \hat{\boldsymbol{\mu}})$, we have $L_c(\mathbf{r}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\mu}}) = \sum_{l \in \mathcal{L}} L_l(\mathbf{r}_l, \hat{\mu}_l, \hat{\boldsymbol{\phi}}_l, \hat{\boldsymbol{\mu}})$. We choose each r_k^p to be the minimizer, and we find $\min_{\mathbf{r}} L_c(\mathbf{r}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\mu}}) = \sum_{l \in \mathcal{L}} \min_{\mathbf{r}_l} L_l(\mathbf{r}_l, \hat{\mu}_l, \hat{\boldsymbol{\phi}}_l, \hat{\boldsymbol{\mu}})$.

2. From (3.18), we observe that $\sum_{l \in \mathcal{L}} \min_{\mathbf{r}_l} L_l(\mathbf{r}_l, \hat{\mu}_l, \hat{\boldsymbol{\phi}}_l, \boldsymbol{\mu}^{j-1})$ is a lower-bound.

3. We deploy [111, Proposition 7.1.1] to find the derivative of $\min_{\mathbf{r}_l} L_l(\mathbf{r}_l, \hat{\mu}_l, \hat{\boldsymbol{\phi}}_l, \hat{\boldsymbol{\mu}})$ with respect to μ_l. There are three satisfied conditions that ensure the existence of the derivative: a) the feasible set of (3.14) is compact; b) $L_l(\mathbf{r}_l, \hat{\mu}_l, \hat{\boldsymbol{\phi}}_l, \hat{\boldsymbol{\mu}})$ is continuous in μ_l; and c) for each $\hat{\mu}_l$, the equation $\partial L_l(\mathbf{r}_l, \hat{\mu}_l, \hat{\boldsymbol{\phi}}_l, \hat{\boldsymbol{\mu}})/\partial r_k^p = 0$ has a unique solution for r_k^p due to the strict convexity of $\psi(\mathbf{r})$. Given identical $\hat{\mu}_l$ and $\hat{\phi}_l$ to both Lagrangian functions (3.8) and (3.11), the derivative of $\sum_{l \in \mathcal{L}} \min_{\mathbf{r}_l} L_l(\mathbf{r}_l, \hat{\mu}_l, \hat{\boldsymbol{\phi}}_l, \hat{\boldsymbol{\mu}})$ with respect to μ_l is

$$\sum_{k=1}^{K} \sum_{p:\{p \in \mathcal{P}_k, l \in \mathcal{L}_k^p\}} r_k^p - C_l,$$
$$\text{where } r_k^p = \arg\min_{r_k^p} L_l(\mathbf{r}_l, \hat{\mu}_l, \hat{\boldsymbol{\phi}}_l, \hat{\boldsymbol{\mu}}).$$

The derivative of $\min_{\mathbf{r}} L_c(\mathbf{r}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\phi}})$ with respect to μ_l is

$$\sum_{k=1}^{K} \sum_{p:\{p \in \mathcal{P}_k, l \in \mathcal{L}_k^p\}} r_k^p - C_l, \text{ where } r_k^p = \arg\min_{r_k^p} L_c(\mathbf{r}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\phi}}).$$

As the minimizers of $L_c(\mathbf{r}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\phi}})$ and $L_l(\mathbf{r}_l, \hat{\mu}_l, \hat{\boldsymbol{\phi}}_l, \hat{\boldsymbol{\mu}})$ are equal at point $\hat{\boldsymbol{\mu}}$ due to (3.19) and (3.20), we observe that both above derivatives are equal.

4. $\sum_{l\in\mathcal{L}} \min_{\mathbf{r}_l} L_l(\mathbf{r}_l, \hat{\mu}_l, \hat{\boldsymbol{\phi}}_l, \boldsymbol{\mu}^{j-1})$ is a piecewise linear function of $\hat{\boldsymbol{\mu}}$, and thus, it is a continuous function of $\hat{\boldsymbol{\mu}}$.

Building on the above arguments, Algorithm 8 is a block successive lower-bound maximization method, which satisfies all four convergence conditions given in [19, Assumption 2]. Algorithm 8 converges to the global optimal solution of the concave problem (3.9) [19, Theorem 2], which has an identical objective function to (3.7) at the optimal point as a result of strong duality [108, p. 226–p. 227]. Once Algorithm 8 converges, $\boldsymbol{\mu}^j$ and $r_k^p = \arg\min_{r_k^p} L_l(\mathbf{r}_l, \mu_l^j, \boldsymbol{\phi}_l^j, \boldsymbol{\mu}^{j-1})$ by Algorithm 8 satisfy (3.15a)–(3.15d). The KKT conditions for (3.7) are (3.15b)–(3.15d) in addition to $\frac{\partial L_c(\mathbf{r},\boldsymbol{\mu}^j,\boldsymbol{\phi}^j)}{\partial r_k^p} = 0$. Due to (3.16) and (3.17), when Algorithm 8 converges, minimizers of $L_l(\mathbf{r}_l, \mu_l^j, \boldsymbol{\phi}_l^j, \boldsymbol{\mu}^{j-1})$ and $L_c(\mathbf{r}, \boldsymbol{\mu}^j, \boldsymbol{\phi}^j)$ are identical, and thus, (3.15a) ensures $\frac{\partial L_c(\mathbf{r},\boldsymbol{\mu}^j,\boldsymbol{\phi}^j)}{\partial r_k^p} = 0$. Hence, the primal and dual variables obtained by Algorithm 8 satisfy the KKT conditions for (3.7). □

Remark 2. *If the cost function is convex and has a gradient that is Lipschitz continuous, but the function is not separable in* r_k^p*, i.e.,* $\psi(\mathbf{r})$ *cannot be written as* $\psi(\mathbf{r}) = \sum_{k=1}^{K}\sum_{p\in\mathcal{P}_k} \psi_k^p(r_k^p)$*, we use the quadratic upper-bound given in [55, eq. (12)], which is separable in variables. For an arbitrary convex cost function with a Lipschitz continuous gradient like* $\psi(\mathbf{r})$*, we have the following upper-bound:*

$$\psi(\mathbf{r}) \leq \psi(\mathbf{r}^m) + \nabla\psi(\mathbf{r}^m)^T(\mathbf{r} - \mathbf{r}^m) + \frac{\gamma}{2}\left\|\mathbf{r} - \mathbf{r}^m\right\|_2^2, \tag{3.21}$$

where γ *is the Lipschitz constant, and* $\mathbf{r}^m = \{r_k^{p,m}\}_{p\in\mathcal{P}_k, k=1:K}$ *is the* m^{th} *iterate in the successive upper-bound minimization. We start from an initial point* $\mathbf{r}^0$ *in the feasible set and find the upper-bound* (3.21)*. Then, we apply Algorithm 8 to solve the problem with the upper-bound* (3.21) *to the global optimal solution in a parallel fashion.*

When the upper-bound (3.21) *is substituted for the cost function, the first KKT condition is*

$$\begin{aligned}&\frac{\partial L_l(\mathbf{r}_l, \mu_l, \boldsymbol{\phi}_l, \boldsymbol{\mu}^{j-1})}{\partial r_k^p} = \\ &\alpha_{k,l}^{p,j}\left(\frac{\partial \psi_k^p(r_k^p)}{\partial r_k^p}\,|_{r_k^P = r_k^{p,m}} + \gamma(r_k^p - r_k^{p,m})\right) + \mu_l - \alpha_{k,l}^{p,j}\phi_k^p = 0,\end{aligned} \tag{3.22}$$

Algorithm 9: BSUM multi-path routing algorithm for non-separable cost functions

```
0. Initialization Choose a feasible vector $\mathbf{r}^0$, $m = 0$;
repeat
    1. Find the upper-bound (3.21) using $\mathbf{r}^m$;
    2. Apply Algorithm 8 to find $\mathbf{r}$;
    3. $m = m + 1$ and $\mathbf{r}^m = \mathbf{r}$;
until variables in $\mathbf{r}^m$ converge;
```

instead of (3.15a)*. Once the problem with the upper-bound* (3.21) *is solved, we use the obtained solution to update* $\mathbf{r}^m$ *in the upper-bound* (3.21)*. We repeat this approach until* $\mathbf{r}^m$ *converges. We summarize this approach in Algorithm 9.*

In iteration m*, the value of the upper-bound* (3.21) *and its gradient are* $\psi(\mathbf{r}^m)$ *and* $\nabla\psi(\mathbf{r}^m)$*, respectively, which are equal to the value and the gradient of the non-separable cost function* $\psi(\mathbf{r})$*. Furthermore, the upper-bound in* (3.21) *is continuous, and thus, all four convergence conditions given in [19, Assumption 2] are satisfied. Due to [19, Theorem 2] and the convexity of the non-separable cost function, the obtained solution by Algorithm 9, which implements BSUM, is identical to the solution of the original problem with the non-separable cost function.*

Remark 3. *When the cost function is convex and separable, but not strictly convex, we add a proximal term to the cost function and make it locally strongly convex as follows:*

$$\begin{aligned} \min_{\mathbf{r}} \quad & \sum_{k=1}^{K} \sum_{p \in \mathcal{P}_k} \psi_k^p(r_k^p) + \frac{\kappa}{2} \left\| \mathbf{r} - \mathbf{r}^m \right\|_2^2 \\ s.t. \quad & (3.1), r_k \geq 0, \quad p \in \mathcal{P}_k, \forall k, \end{aligned} \tag{3.23}$$

where κ *is a small positive constant. We use Algorithm 8 to solve the above problem when we use the following equation:*

$$\frac{\partial L_l(\mathbf{r}_l, \mu_l, \boldsymbol{\phi}_l, \boldsymbol{\mu}^{j-1})}{\partial r_k^p} = \alpha_{k,l}^{p,j} \frac{\partial \psi_k^p(r_k^p)}{\partial r_k^p} + \alpha_{k,l}^{p,j} \kappa (r_k^p - r_k^{p,m}) + \mu_l - \alpha_{k,l}^{p,j} \phi_k^p = 0,$$

instead of (3.15a) *to find* r_k^p*, where* $r_k^{p,m}$ *is the value of* r_k^p *in the* m^{th} *iteration of solving* (3.23)*. We successively solve* (3.23) *with Algorithm 8 and update* $\mathbf{r}^m$ *until* $\mathbf{r}^m$ *converges.*

Similar to Remark 2, one can show that the cost function with the proximal term in (3.23) *satisfies the four convergence conditions in [19, Assumption 2] and the global minimum is obtained after successive minimizations, since each local minimum is also global for a convex function.*

3.5 Simultaneous Resource Reservations in The Backhaul and RAN

In this section, we study the joint link capacity and AP transmission resource reservation based on the user demand and downlink statistics. Prior to the observation of user demands, based on the formulated model in (3.5), the network operator finds the optimal amount of reserved resources in the backhaul and APs such that neither the available link capacity nor AP capacity is exceeded.

3.5.1 Resource Reservation in The Backhaul

Let us drop the equality constraint (3.3) from (3.5) and substitute $\sum_{p\in\mathcal{P}_k} r_k^p$ for r_k. Then, we have

$$\begin{aligned} \min_{\mathbf{r},\mathbf{t}} \quad & \sum_{k=1}^{K}\Big[-\mathbb{E}[\min(\sum_{p\in\mathcal{P}_k} r_k^p, d_k)] + \theta_k \sum_{p\in\mathcal{P}_k}\int_0^{r_k^p} z_k^p(v_k^p, t_k^p)\,(r_k^p - v_k^p)dv_k^p\Big] \\ \text{s.t.} \quad & (3.1), (3.2), r_k^p, t_k^p \geq 0, \quad p\in\mathcal{P}_k, \forall k. \end{aligned} \tag{3.24}$$

We solve the above problem using the proposed BCD algorithm. With the fixed $\mathbf{t}$, we minimize (3.24) with respect to $\mathbf{r}$ and update it. With updated $\mathbf{r}$, we minimize (3.24) with respect to $\mathbf{t}$ and update it. We underline the iterates of the BCD algorithm. In the $i+1^{\text{th}}$ iteration of the BCD algorithm, fixing $\underline{\mathbf{t}}^i$, we minimize with respect to $\mathbf{r}$. Then, the minimization problem in (3.24) reduces to the following *convex* one:

$$\begin{aligned} \min_{\mathbf{r}} \quad & \sum_{k=1}^{K}\Big[-\mathbb{E}[\min(\sum_{p\in\mathcal{P}_k} r_k^p, d_k)] + \theta_k \sum_{p\in\mathcal{P}_k}\int_0^{r_k^p} z_k^p(v_k^p, \underline{t}_k^{p,i})\,(r_k^p - v_k^p)dv_k^p\Big] \\ \text{s.t.} \quad & (3.1), r_k^p \geq 0, \quad p\in\mathcal{P}_k, \forall k. \end{aligned} \tag{3.25}$$

It is observed that although the expected outage is separable in r_k^p, variables are coupled in the first term of the objective function. Therefore, we substitute the global quadratic upper-bound given in (3.21) for the expected supportable traffic demand. First, let us calculate the Lipschitz constant for the gradient of $-\mathbb{E}[\min(\sum_{p\in\mathcal{P}_k} r_k^p, d_k)]$. The second derivative of $-\mathbb{E}[\min(\sum_{p\in\mathcal{P}_k} r_k^p, d_k)]$ is

$$-\frac{\partial^2\mathbb{E}[\min(\sum_{p\in\mathcal{P}_k} r_k^p, d_k)]}{\partial r_k^p \partial r_k^{p'}} = f_k(\sum_{p\in\mathcal{P}_k} r_k^p).$$

The Hessian matrix for $-\mathbb{E}[\min(\sum_{p\in\mathcal{P}_k} r_k^p, d_k)]$ is $|\mathcal{P}_k| \times |\mathcal{P}_k|$ dimensional, where all entries are $f_k(\sum_{p\in\mathcal{P}_k} r_k^p)$. The eigenvalues of the Hessian matrix are all zeros except one of them, which is $|\mathcal{P}_k| f_k(\sum_{p\in\mathcal{P}_k} r_k^p)$. Therefore, the Lipschitz constant is $|\mathcal{P}_k|$. We now place the Lipschitz constant in (3.21) and find the upper-bound which is separable in r_k^p as follows:

$$\begin{aligned}
&-\mathbb{E}[\min(\sum_{p\in\mathcal{P}_k} r_k^p, d_k)] \leq -\mathbb{E}[\min(\sum_{p\in\mathcal{P}_k} r_k^{p,m}, d_k)] \\
&+ (F_k(\sum_{p\in\mathcal{P}_k} r_k^{p,m}) - 1)(\sum_{p\in\mathcal{P}_k} r_k^p - \sum_{p\in\mathcal{P}_k} r_k^{p,m}) \\
&+ \frac{|\mathcal{P}_k|}{2}\sum_{p\in\mathcal{P}_k}(r_k^{p,m} - r_k^p)^2, \qquad (3.26)
\end{aligned}$$

where $r_k^{p,m}$ is the m^{th} iterate. We substitute upper-bound (3.26) for the expected supportable demand and the optimization problem in each iteration becomes:

$$\begin{aligned}
\min_{\mathbf{r}} \quad & \sum_{k=1}^{K}\Bigg[(F_k(\sum_{p\in\mathcal{P}_k} r_k^{p,m}) - 1)(\sum_{p\in\mathcal{P}_k} r_k^p - \sum_{p\in\mathcal{P}_k} r_k^{p,m}) \qquad (3.27) \\
& + \frac{|\mathcal{P}_k|}{2}\sum_{p\in\mathcal{P}_k}(r_k^p - r_k^{p,m})^2 + \theta_k \sum_{p\in\mathcal{P}_k}\int_0^{r_k^p} z_k^p(v_k^p, \underline{t}_k^{p,i})\,(r_k^p - v_k^p)dv_k^p\Bigg] \\
\text{s.t.} \quad & (3.1), r_k^p \geq 0, \quad p \in \mathcal{P}_k, \forall k.
\end{aligned}$$

We leverage Algorithm 9 to solve (3.25) in a parallel fashion. In each iteration of Algorithm 9, Algorithm 8 is called to solve the problem in (3.27). Moreover, Algorithm 7 is called within Algorithm 8 and it needs to solve $\frac{\partial L_l(\mathbf{r}_l, \mu_l, \boldsymbol{\phi}_l, \boldsymbol{\mu}^{j-1})}{\partial r_k^p} = 0$. We rewrite (3.15a) for the above optimization problem in the j^{th} iteration of Algorithm 8 as follows:

$$\frac{\partial L_l(\mathbf{r}_l, \mu_l, \boldsymbol{\phi}_l, \boldsymbol{\mu}^{j-1})}{\partial r_k^p} = \alpha_{k,l}^{p,j}(F_k(\sum_{p\in\mathcal{P}_k} r_k^{p,m}) - 1) + \mu_l$$
$$+ \alpha_{k,l}^{p,j}|\mathcal{P}_k|(r_k^p - r_k^{p,m}) + \theta_k\, \alpha_{k,l}^{p,j} Z_k^p(r_k^p, \underline{t}_k^{p,i}) - \alpha_{k,l}^{p,j}\phi_k^p = 0.$$

We observe that for each μ_l, we are able to obtain r_k^p numerically using $r_k^{p,m}$, independent of the other variables. The solution obtained by Algorithm 9 is unique due to the strong convexity of (3.27) and is global minima as explained in Remark 2. After Algorithm 9 converges, we set $\underline{\mathbf{r}}^{i+1} = \mathbf{r}^m$.

Remark 4. *Suppose that the number of paths that are available to user k and share downlink $w \in \mathcal{W}_k$ is φ_k^w. When multiple paths for serving a user share one downlink, we substitute the quadratic upper-bound* (3.21) *for the outage* (3.6) *as follows:*

$$\begin{aligned}(3.6) \leq \sum_{w\in\mathcal{W}_k} \int_0^{\sum_{p:\{p\in\mathcal{P}_k, w\in p\}} r_k^{p,m}} z_k^w(v_k^p, \underline{t}_k^{w,i})(\sum_{p:\{p\in\mathcal{P}_k, w\in p\}} r_k^{p,m} - v_k^p) dv_k^p \\ + \sum_{w\in\mathcal{W}_k} Z_k^w(\sum_{p:\{p\in\mathcal{P}_k, w\in p\}} r_k^{p,m}, \underline{t}_k^{w,i})(\sum_{p:\{p\in\mathcal{P}_k, w\in p\}} (r_k^p - r_k^{p,m})) \\ + \sum_{w\in\mathcal{W}_k} \sum_{p:\{p\in\mathcal{P}_k, w\in p\}} \frac{\varphi_k^w}{2}(r_k^p - r_k^{p,m})^2, \end{aligned} \tag{3.28}$$

where φ_k^w is the Lipschitz constant. In this case, the objective function of (3.27) *is obtained from adding* (3.26) *and the RHS of* (3.28). *We rewrite* (3.15a) *in the j^{th} iteration of Algorithm 8 for this case as follows:*

$$\frac{\partial L_l(\mathbf{r}_l, \mu_l, \boldsymbol{\phi}_l, \boldsymbol{\mu}^{j-1})}{\partial r_k^p} = \alpha_{k,l}^{p,j}(F_k(\sum_{p\in\mathcal{P}_k} r_k^{p,m}) - 1)$$
$$+ \alpha_{k,l}^{p,j}|\mathcal{P}_k|(r_k^p - r_k^{p,m}) + \theta_k\, \alpha_{k,l}^{p,j} Z_k^w(\sum_{p:\{p\in\mathcal{P}_k, w\in p\}} r_k^{p,m}, \underline{t}_k^{w,i})$$
$$+ \theta_k \varphi_k^w \alpha_{k,l}^{p,j}(r_k^p - r_k^{p,m}) + \mu_l - \alpha_{k,l}^{p,j}\phi_k^p = 0.$$

3.5.2 Resource Reservation in RAN

When we minimize (3.24) with respect to $\mathbf{t}$ in the BCD algorithm, we use $\underline{\mathbf{r}}^{i+1}$ obtained by Algorithm 9. We propose a dual approach to minimize with respect to $\mathbf{t}$. The objective function of (3.24) is separable in t_k^p. We are able to parallelize the algorithm across APs since each AP has a separate transmission resource capacity constraint. However, the problem in (3.24) is not necessarily convex in t_k^p for an arbitrary $z_k^p(v_k^p, t_k^p)$. To tackle the potential non-convexity of the problem, we use the BSUM method and convexify the problem locally. We iteratively solve a sequence of convex approximations. Suppose that for each outage term in the objective function of (3.24), we add a proximal term $\frac{\zeta_k^{p,j}}{2} \left\| t_k^p - t_k^{p,j} \right\|_2^2$, $\zeta_k^{p,j} > 0$, to make it locally strongly convex. In the proximal term, $t_k^{p,j}$ is the value of t_k^p in the j^{th} iteration of successively minimizing (3.24) with respect to $\mathbf{t}$. The objective function with the proximal terms is an upper-bound of the original objective function. We find the Lagrangian for (3.24) with respect to $\mathbf{t}$ with proximal terms in the objective function as follows:

$$
\begin{aligned}
L_t(\mathbf{t}, \boldsymbol{\lambda}, \boldsymbol{\beta}) = \sum_{k=1}^{K} \sum_{p \in \mathcal{P}_k} \Bigg(\theta_k \int_0^{\underline{r}_k^{p,i+1}} & z_k^p(v_k^p, t_k^p) \, (\underline{r}_k^{p,i+1} - v_k^p) dv_k^p \\
+ \frac{\theta_k \zeta_k^{p,j}}{2} \left\| t_k^p - t_k^{p,j} \right\|_2^2 \Bigg) + \sum_{b \in \mathcal{B}} \lambda_b (\sum_{k=1}^{K} & \sum_{p:\{p \in \mathcal{P}_k, b \in \mathcal{U}_k^p\}} t_k^p - C_b) - \sum_{k=1}^{K} \sum_{p \in \mathcal{P}_k} \beta_k^p t_k^p,
\end{aligned}
$$

where $\underline{\mathbf{r}}^{i+1}$ block is kept fixed. We can decompose the above Lagrangian across APs as follows:

$$
\begin{aligned}
& L_{t,b}(\mathbf{t}_b, \lambda_b, \boldsymbol{\beta}_b) = \\
& \theta_k \sum_{k=1}^{K} \sum_{p:\{p \in \mathcal{P}_k, b \in \mathcal{U}_k^p\}} \left(\int_0^{\underline{r}_k^{p,i+1}} z_k^p(v_k^p, t_k^p) \, (\underline{r}_k^{p,i+1} - v_k^p) dv_k^p + \frac{\zeta_k^{p,j}}{2} \left\| t_k^p - t_k^{p,j} \right\|_2^2 \right) \\
& + \lambda_b \, (\sum_{k=1}^{K} \sum_{p:\{p \in \mathcal{P}_k, b \in \mathcal{U}_k^p\}} t_k^p - C_b) - \sum_{k=1}^{K} \sum_{p:\{p \in \mathcal{P}_k, b \in \mathcal{U}_k^p\}} \beta_k^p t_k^p,
\end{aligned}
\tag{3.29}
$$

where $\mathbf{t}_b = \{t_k^p\}_{p\in\mathcal{P}_k, b\in\mathcal{U}_k^p, k=1:K}$ and $\boldsymbol{\beta}_b = \{\beta_k^p\}_{p\in\mathcal{P}_k, b\in\mathcal{U}_k^p, k=1:K} \geq \mathbf{0}$. To develop an algorithm to solve each subproblem with respect to $\mathbf{t}_b$, we use KKT conditions. We write the first-order optimality conditions with respect to $\mathbf{t}$ as follows:

$$\begin{aligned}
&\frac{\partial L_{t,b}(\mathbf{t}_b, \lambda_b, \boldsymbol{\beta}_b)}{\partial t_k^p} = \theta_k \int_0^{\underline{r}_k^{p,i+1}} \frac{\partial z_k^p(v_k^p, t_k^p)}{\partial t_k^p} (\underline{r}_k^{p,i+1} - v_k^p) dv_k^p \\
&+ \lambda_b + \theta_k \zeta_k^{p,j} \, (t_k^p - t_k^{p,j}) - \beta_k^p = 0, && \text{(3.30a)} \\
&\sum_{k=1}^{K} \sum_{p:\{p\in\mathcal{P}_k, b\in\mathcal{U}_k^p\}} t_k^p \leq C_b, && \text{(3.30b)} \\
&\lambda_b \Big(\sum_{k=1}^{K} \sum_{p:\{p\in\mathcal{P}_k, b\in\mathcal{U}_k^p\}} t_k^p - C_b\Big) = 0, \;\; \lambda_b \geq 0, && \text{(3.30c)} \\
&\beta_k^p t_k^p = 0, \;\; t_k^p \geq 0, \;\; \beta_k^p \geq 0. && \text{(3.30d)}
\end{aligned}$$

From (3.30a), we observe that a given dual variable λ_b, which corresponds to AP b, identifies the reserved resource t_k^p for all downlinks created by that AP. The proposed dual algorithm works as follows: implement a bisection search on λ_b in the non-negative orthant and find each $t_k^p : t_k^p \geq 0$, which is associated with the AP b, from (3.30a) when $\beta_k^p = 0$. Continue the bisection search until one λ_b is obtained such that for the obtained λ_b, we have $\sum_{k=1}^{K} \sum_{p:\{p\in\mathcal{P}_k, b\in\mathcal{U}_k^p\}} t_k^p = C_b$. If there is no such λ_b, we set $\lambda_b = 0$ and solve (3.30a) and (3.30d) without (3.30b)–(3.30c). Once the optimized variables are obtained, we update $t_k^{p,j}$ and $j = j + 1$. We repeat the same process until $\mathbf{t}^j = \{t_k^{p,j}\}_{p\in\mathcal{P}_k, k=1:K}$ converges. As it is explained in Remark 3, after a sequence of upper-bound minimizations and updating the proximal terms in the objective function, a KKT (local stationary) solution to the original problem is obtained. If expected outage terms for downlinks are non-increasing in $\mathbf{t}$, one can show that the successive upper-bound minimization converges to the global minima with respect to $\mathbf{t}$. After $\mathbf{t}^j$ converges, we set $\underline{\mathbf{t}}^{i+1} = \mathbf{t}^j$.

3.5.3 The Proposed BCD Algorithm

To solve the problem in (3.24) to a KKT point, we optimize with respect to two blocks of variables, $\mathbf{r}$ and $\mathbf{t}$, alternatively with the Gauss-Seidel update style. Therefore, if we choose $\mathbf{r}$ to update first, with $\underline{\mathbf{r}}^{i+1}$, we optimize with respect to $\mathbf{t}$, and then, we update

Algorithm 10: BCD resource reservation algorithm to solve (3.24)

```
0. Initialization Feasible initializations for $\underline{\mathbf{r}}^0$ and $\underline{\mathbf{t}}^0$, $i = 0$;
repeat
    1. Apply Algorithm 9 to solve (3.24) and find $\underline{\mathbf{r}}^{i+1}$;
    2. Solve (3.24) with respect to $\mathbf{t}$ and find $\underline{\mathbf{t}}^{i+1}$;
    3. $i = i + 1$;
until $\left\|\underline{\mathbf{r}}^i - \underline{\mathbf{r}}^{i-1}\right\|_2^2 + \left\|\underline{\mathbf{t}}^i - \underline{\mathbf{t}}^{i-1}\right\|_2^2$ is small enough;
```

$\underline{\mathbf{t}}^{i+1}$. We keep optimizing with respect to $\mathbf{r}$ and $\mathbf{t}$ alternatively until both blocks converge. The summary of the overall BCD approach is given in Algorithm 10.

Proposition 4. *Algorithm 10 converges to a KKT solution to* (3.24).

Proof. First, the objective function of (3.24) is continuously differentiable. Second, feasible sets of two blocks of variables are separate in (3.24). Hence, updating one block of variables does not change the other block. Third, in each iteration of Algorithm 10, a KKT solution is obtained. Therefore, according to [111, Proposition 3.7.1], the proposed Algorithm 10 converges to a KKT solution. □

Here, we briefly discuss the iteration complexity of different proposed algorithms. The convergence of flow variables in Algorithm 7 depends on the convergence of the dual variables $\{\mu_l\}_{l\in\mathcal{L}}$. The optimal μ_l is obtained through a bisection search. Bisection search has a linear convergence rate and the residual error shrinks with a rate of $\mathcal{O}(1/2^m)$, where m represents the bisection iteration number. In addition, numerical simulations (Fig. 3.4) confirm the linear convergence of Algorithm 8, which ensures a fast error reduction. The multipath routing method described in Algorithm 9 successively minimizes an upper-bound. The convergence rate of BSUM methods to globally minimize convex functions is sublinear and equal to $\mathcal{O}(1/m)$ [112]. The minimization with respect to $\mathbf{t}$ is non-convex and is solved by a BSUM method. The convergence of BSUM methods to KKT points of non-convex problems is analyzed in [113], where an error reduction with a rate of $\mathcal{O}(1/m)$ is guaranteed [113]. Finally, each subproblem of Algorithm 10 is solved via a BSUM method, and therefore, a convergence rate of $\mathcal{O}(1/m)$ is obtained. In the next section, we depict the convergence of our proposed algorithms, which are consistent with the theoretical analyses given in above papers.

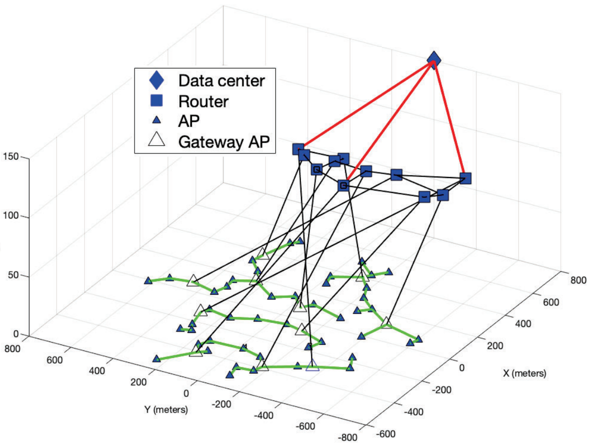


Figure 3.3: A wireless data network consists of 57 APs and 11 routers.

Here, we discuss the energy efficiency of our resource reservation approach. When the transmission resource in the proposed formulations is power, the resource reservation specifies the maximum power that can be used in the network to serve each user. As the average SNR of a user increases with power, the expected outage decreases with the reserved power for a downlink. For certain reserved rates for users, reserving excess power reduces the expected outage of downlinks, and increases the robustness of the resource reservation. However, this robustness entails a lower energy efficiency. The reason is that the feasibility and robustness are enhanced by decreasing the aggregate expected outage at the cost of reserving more power.

3.6 Numerical Tests

In this section, we demonstrate the performance of our proposed approach against two heuristic algorithms.

3.6.1 Simulation Setup

The considered network for evaluations is shown in Fig. 3.3, which includes both the backhaul and radio access parts. A data center is connected to the network through three routers. The network includes 57 APs and 11 network routers. APs are distributed on the X-Y plane and they are connected to each other. One gateway AP is connected to neighboring APs and a router through backhaul links. The backhaul network has 162 links. Wired link capacities are identical in both directions. Backhaul link capacities are determined as

- Links between the data center and routers: 4 Gnats/s;
- Links between routers: 2 Gnats/s;
- Links between routers and APs: 2 Gnats/s;
- 2-hop to the routers: 400 Mnats/s;
- 3-hop to the routers: 320 Mnats/s;
- 4-hop to the routers: 160 Mnats/s.

The considered paths originate from the data center and are extended toward users. We consider 200 users are distributed randomly in the same plane of APs; however, they are not shown in Fig. 3.3. User AP associations are determined by the highest long-term received power. We consider three wireless connections, which have the highest received power, to serve each user. There are three paths for carrying data from a data center to APs. The distribution of the demand is log-normal:

$$d_k \sim \frac{1}{d_k \sigma_k \sqrt{2\pi}} \exp(-\frac{(\ln d_k - \eta_k)^2}{2\sigma_k^2}). \tag{3.31}$$

In addition, it is assumed that η_k is realized randomly from a normal distribution for each user. The power allocations in APs are fixed. The dispensed resource in an AP is bandwidth. The channel between each user and an AP is a Rayleigh fading channel. The CDF of the wireless channel capacity, which is parameterized by the allocated bandwidth t_k^p, is given as follows [100]:

$$Z_k^p(v_k^p, t_k^p) = 1 - \exp(\frac{1 - 2^{v_k^p / t_k^p}}{\overline{\mathrm{SNR}_k^p}}),$$

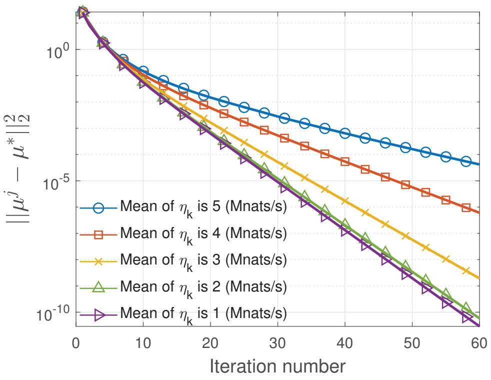


Figure 3.4: The convergence of Algorithm 8.

Table 3.2: CPU time for 60 iterations of Algorithm 9.

Mean of η_k	1 Mnats/s	2 Mnats/s	3 Mnats/s	4 Mnats/s
CPU time	0.051 s	0.082 s	0.104 s	0.121 s
Mean of η_k		5 Mnats/s	6 Mnats/s	
CPU time		0.144 s	0.158 s	

where $\overline{\text{SNR}_k^p}$ is the average SNR. The PDF of the wireless channel capacity is

$$z_k^p(v_k^p, t_k^p) = \frac{\ln(2) 2^{v_k^p/t_k^p} \exp(\frac{1-2^{v_k^p/t_k^p}}{\overline{\text{SNR}_k^p}})}{\overline{\text{SNR}_k^p} t_k^p}. \tag{3.32}$$

Benchmark heuristic algorithms are the single-path and the average-based approaches. In the single-path approach, each user is served through one path from a data center to a user. Moreover, the average-based algorithm only considers the mean of the user demand and the average achievable rate of a downlink. To compare algorithms, with an identical network, we measure the objective function of (3.5), the sum of user expected supportable rates, the aggregate expected outage of downlinks and the amount of traffic that each algorithm can reserve for users. One datastream is associated with each user. In total, we have 600 paths in the backhaul. We use C to implement algorithms.

3.6.2 Learning Probability Density Functions

The optimization problem in (3.5) takes into account PDFs of user demands and achievable rates of downlinks. When PDFs are not given, one can use a data-driven approach to learn PDFs used in (3.5) based on collected observations. Upon the collection of user demands and achievable rates of downlinks, one can estimate the PDFs using a recursive non-parametric estimator. In order to estimate PDFs in an online streaming fashion, one can use efficient recursive kernel estimators, such as the Wolverton and Wagner estimator [114]. Suppose that independent random variables $X_1, X_2, \ldots, X_n$ are observations that are collected from an identical PDF χ with respect to Lebesgue's measure. The estimated PDF is

$$\hat{\chi}_{n,\mathbf{h}_n} = \frac{1}{n}\sum_{k=1}^{n}\frac{1}{h_k}K(\frac{X_k - x}{h_k}),$$

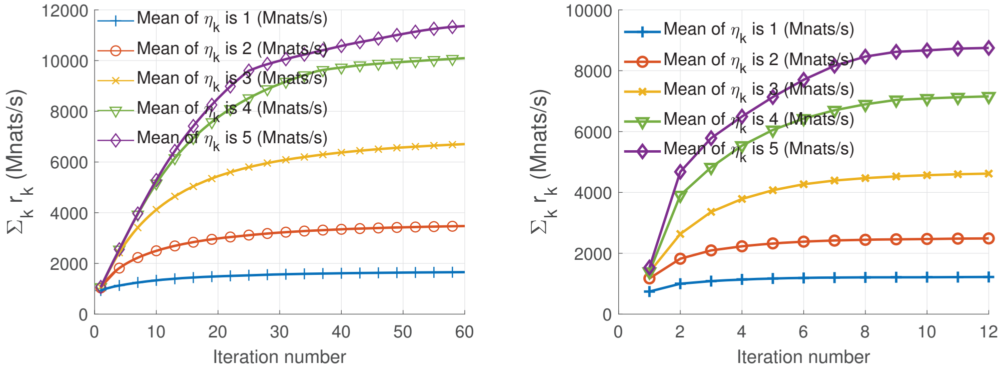
where $\mathbf{h}_n = (h_1, h_2, \ldots, h_n)$, $h_1 > \cdots > h_n$ and $K(\cdot)$ is a kernel function. The advantage of the above estimator is that it can be written in a recursive form as follows:

$$\hat{\chi}_{n+1,\mathbf{h}_{n+1}} = \frac{n}{n+1}\hat{\chi}_{n,\mathbf{h}_n} + \frac{1}{(n+1)h_{n+1}}K(\frac{X_{n+1} - x}{h_{n+1}}),$$

which makes it suitable for real-time applications. The bandwidth selection in [115] can be used for the above estimator. The bandwidth h_k is selected in [115] as $h_k = k^{-\gamma}, k \in \{1, \ldots, n\}$, where $\gamma = \frac{1}{2\beta+1}$ and $\beta > 0$.

3.6.3 Simulation Results

Before demonstrating the performance of Algorithm 10, we depict the convergence of Algorithm 8 in Fig. 3.4. The convergence of Algorithm 8 for different means of the user demand is depicted in Fig. 3.4. It is observed that Algorithm 8 has a fast linear convergence rate for the large network of Fig. 3.3 with 600 paths. Numerical results show that the number of required iterations for Algorithm 9 to converge for the simulation setting described above is at most 60. Fig. 3.4 shows the optimality gap. The CPU time for Algorithm 9 is measured and is given in Table 3.2. We plot the convergence of Algorithm 9 in Fig. 3.5a, where we set $\theta_k = 0.5$ and the AP bandwidth budget is 80 MHz. The convergence of Algorithm 10 is depicted in Fig. 3.5b when the bandwidth budget of each AP is 80 MHz and $\theta_k = 2$. From



(a)

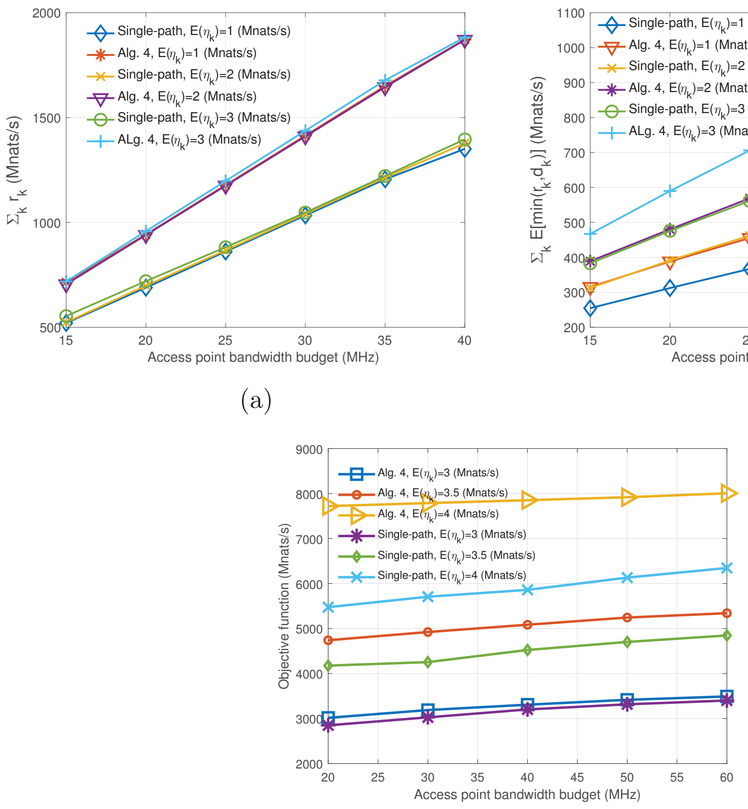
(b)

Figure 3.5: The convergence of Algorithm 9 and Algorithm 10.

Figs. 3.5a-3.5b, we observe that the convergence rates of Algorithm 9 and Algorithm 10 are both sublinear.

First, let us assume that transmission rates on downlinks are deterministic functions of bandwidth in APs. Therefore, no outage (rate loss) is considered. For each downlink, the transmission rate and the allocated bandwidth are connected to each other as $r_k^p = \delta_k^p t_k^p$, where δ_k^p is the spectral efficiency of the downlink of path p to serve user k. Furthermore, suppose that $\sigma_k = 3.6$ and the capacity of each backhaul link listed previously is divided by 4. When the bandwidth budget of each AP increases from 15 MHz to 40 MHz, the aggregate reserved rates for users by Algorithm 10 (multi-path) and the single-path approach are shown in Fig. 3.6a. The single-path approach is used in a number of papers, for instance [72, 116, 117]. The aggregate expected supportable rates of users with both approaches are depicted in Fig. 3.6b. It is observed that Algorithm 10 outperforms the single-path approach. Both approaches utilize all available bandwidth in APs.

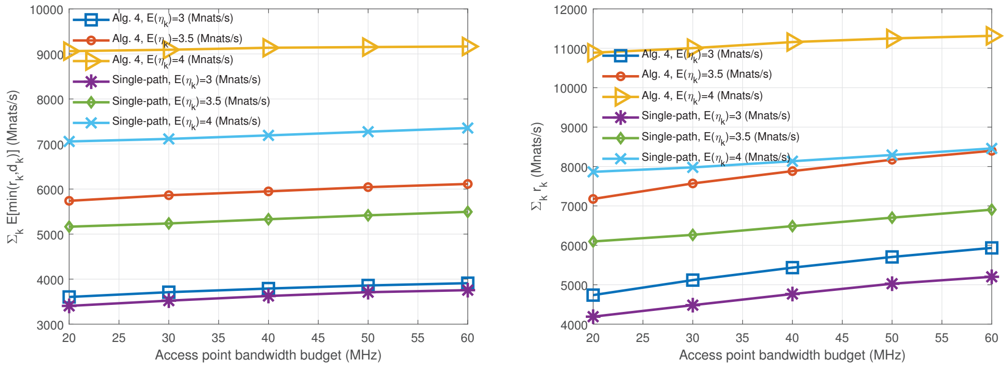
Consider the distribution of each wireless channel (downlink) achievable rate follows (3.32) and backhaul link capacities are as listed previously. Suppose that the available bandwidth in each AP increases by a step size of 10 MHz, where $\theta_k = 1/2$ and $\sigma_k = 0.6$. The objective function of the problem in (3.5) by Algorithm 10 and the single-path approach are compared in Fig. 3.6c. Our proposed Algorithm 10 outperforms the single-path approach. It is observed that with the increase of mean for η_k and the AP bandwidth budget, the objective function increases.



(a) (b)

(c)

Figure 3.6: (a) Reserved rates by Algorithm 10 (multi-path) and the single-path approach when wireless channels are deterministic. (b) The expected supportable rates for users by Algorithm 10 and the single-path approach when wireless channels are deterministic. (c) The objective function of problem (3.5) with the single-path approach and Algorithm 10 when wireless channels are stochastic.

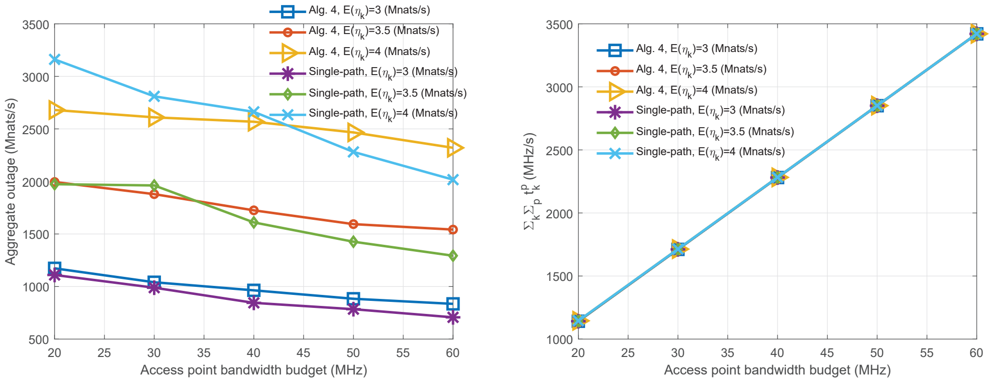


(a) (b)

Figure 3.7: Stochastic wireless channels: performance of the single-path approach and Algorithm 10 in terms of (a) the aggregate expected supportable traffic; and (b) aggregate reserved rates.

The expected supportable demands of users, depicted in Fig. 3.7a, increases when the mean of η_k and the AP bandwidth budget increase. It is observed from Fig. 3.7a that the aggregate expected supportable traffic for users obtained by Algorithm 10 is greater than that by the single-path approach. In Fig. 3.7b, we observe that the aggregate reserved rates for users increases with the increase of mean for η_k. Furthermore, it increases when the bandwidth budgets of APs increase. From Fig. 3.8a, we observe that the aggregate expected outage increases as the mean of η_k increases and decreases when the AP bandwidth budget increases. We observe from Fig. 3.8b that the bandwidth reservation by Algorithm 10 is almost equal to that by the single-path approach. Numerical results show that 15 iterations are sufficient for the convergence of Algorithm 10.

Next, we evaluate the performance of Algorithm 10 against the average-based approach when both the demand and downlink achievable rates are stochastic. The average-based algorithm is oblivious to the user demand and the downlink achievable rate distributions. It only considers the average of each user demand and the average achievable rate of a downlink. This approach is repeatedly used in different papers, e.g., [118, 119]. The average-based approach uses the same set of paths used by Algorithm 10. The bandwidth budget in each AP is 40 MHz. Furthermore, $\sigma_k = 0.6$ and $\theta_k = 1/3$. The demand and downlink achievable rate distributions are as given in (3.31) and (3.32), respectively. Both



(a) (b)

Figure 3.8: Stochastic wireless channels: performance of the single-path approach and Algorithm 10 in terms of (a) expected outage of downlinks; and (b) reserved bandwidth.

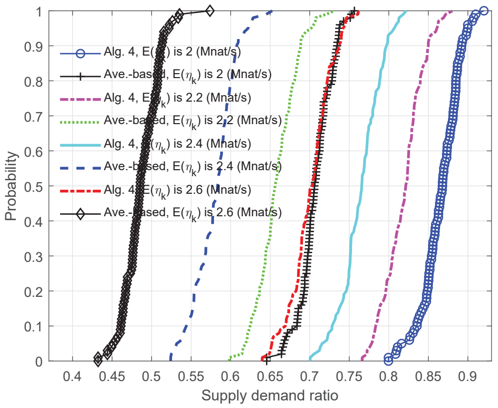


Figure 3.9: The probability of being able to support the user demands up to a certain percentage.

approaches are set to make reservations for users assuming the mean of η_k is 2 Mnast/s. We generate 100 scenarios in which user demands and downlink capacities are random. For each scenario, we measure how much the user demands are satisfied using the reserved resources in the network by both approaches. After collecting results for 100 scenarios, we plot the empirical CDF for the supply demand ratio in Fig. 3.9. It is observed that when the mean of demand exceeds what it was supposed to be, the resource reservation made by Algorithm 10 is more robust and supports random demands better. The total reserved link capacities in the backhaul by Algorithm 10 is 1.0979×10^4 Mnats/s and is 7.74×10^3 Mnats/s by the average-based approach. Furthermore, the total reserved bandwidth in RAN by Algorithm 10 is 2.043×10^3 MHz and is 1.968×10^3 MHz by the average-based approach.

3.7 Concluding Remarks

In this chapter, we studied link capacity and transmission resource reservation in wireless data networks prior to the observation of user demands. Using the statistics of user demands and achievable rates of downlinks, we formulated an optimization problem to maximize the sum of user expected supportable traffic while minimizing the expected outage of downlinks. We demonstrated that this problem is non-convex in general. To solve the problem approximately, an efficient BCD approach is proposed which benefits from distributed and parallel computation when each block of variables is chosen to be updated. We demonstrated that despite the non-convexity of the problem, our proposed approach converges to a KKT solution to the problem. We verified the efficiency and the efficacy of our proposed approach against two heuristic algorithms developed for joint resource reservation in the backhaul and RAN.

In the next chapter, we consider multi-tenant networks and reservation-based network slicing. In addition to users, tenants have different requirements [67, 70, 71], and maximum isolation between sliced resources should be enforced [66]. The demand distribution of users may change over time and the network resources should be sliced for tenants accordingly. However, the slice reconfiguration for each tenant involves cost and overhead. Based on the cost of reconfiguration and newly arrived statistics, we formulate the problem from a sparse optimization perspective and propose an efficient approach based on iteratively solving a sequence of group Least Absolute Shrinkage and Selection Operator (LASSO) problems [67, 71].

Chapter 4

Data-Driven Adaptive Network Resource Slicing for Multi-Tenant Networks

Recently, significant attention has been placed on network slicing as a key element for enabling flexibility and programmability in 5G mobile networks [20, 21]. The network *slice* is a logical, virtualized end-to-end network provided to each tenant to support the demands of users. Each slice consists of link capacities in the backhaul and transmission resources, e.g., bandwidth, in the RAN, while each *tenant* may own multiple network slices and use their reserved resources to serve users. Network slices are dynamically activated, reconfigured and deactivated by the control center of the infrastructure provider.

The life-cycle management of network slices includes the design and creation phase, orchestration and activation phase, and the optimization and reconfiguration phase [120]. In the design and creation phase, network slices are conceptually constructed based on user demands. In the orchestration and activation phase, network slices are installed on the shared physical infrastructure and user traffic starts flowing through the slice. Finally, in the optimization and reconfiguration phase, the performance of network slices is monitored, and based on traffic variations, network slices are reconfigured to maintain the Quality of Service (QoS) requirements of users.

The requirements of network slicing include scalability, flexibility, isolation, and efficient end-to-end orchestration [11]. The scalability enables the network to efficiently adapt to provide a wide variety of applications. The flexibility allows slice reconfiguration to improve the concert of resources across different parts of the network to provide a particular service [121]. In spite of the many benefits, the slice reconfiguration incurs costs and service interruptions. Therefore, slices are reconfigured only if the new configurations significantly improve network performance [122]. The isolation of reserved resources for different slices over shared physical infrastructure enables independent management of network slices and ensures that a rapid increase in the number of users, a slice failure or a security attack on one slice do not affect other slices [123]. The end-to-end orchestration helps to coordinate multiple system facets to maintain the QoS requirements of users. Although network slicing in the backhaul [124, 125, 126] and in RAN [127, 128] are traditionally studied separately, only the end-to-end orchestration can ensure a robust resource allocation and reliable network performance.

To fully leverage the benefits of network slicing in 5G, it is necessary to dynamically reconfigure slices and allocate resources in a flexible data-driven manner. The data-driven methods adapt slices to traffic variations and channel states [129]. Data-driven adaptive network slicing enables efficient and flexible allocation of resources to better support the QoS requirements of users. The QoS attributes include minimum data rate, maximum rate-loss, reliability, and security. Depending on the QoS demands, users can be assigned to different slices.

4.1 Prior Work

The combinatorial problem of network slice activation is studied in [130], where the incoming traffic is supposed to follow a Poisson distribution. In [130], a network slice is kept active based on the gained utility from serving the most recent demand without considering the predictions for the future traffic. The activation cost, which is neither convex nor continuous, is not considered. A heuristic approach for network slice activation is proposed in [131], which only considers the real-time status of network slices. In practice, due to the large cost of a slice activation, network slices are operated based on long-term future traffic rather than the instantaneous demand. No existing paper in the literature considers the mathematical problem of slice activation in an end-to-end network for unseen user demands.

Caballero *et al.* studied the allocation of network transmission resources among several tenants in [127], where transmission rates to users are maximized based on a weighted proportional fairness. The number of users served by each tenant is assumed to be a random number in [127]. An optimal dynamic resource allocation problem is formulated and shown to be NP-hard. A greedy approach is proposed in [127] to solve the problem without a proof of convergence. To find the resource allocations for different slices of a network, a game-theoretic approach which maximizes an $\alpha-$fairness is proposed in [128]. Static and dynamic pricing frameworks are proposed in [128] to allocate base station transmission resources to slices, where each slice has a non-cooperative, strategic behavior. The drawbacks of [128] include neglecting traffic statistics and failing to address the joint resource allocation in the backhaul and RAN. An adaptive forecasting and bandwidth allocation for cyclic demands in service-oriented networks is proposed in [132] without accounting for the reconfiguration cost. The user-slice association problem to find the best slice to support requirements of each user is studied in our previous work [66].

An optimization framework for flexible inter-tenant resource sharing with transmission power control is proposed in [133] to improve network capacity and the utilization of base station resources, where interference levels are controllable. However, achievable rates of channels are assumed to be deterministic, which is an impractical assumption as the achievable rates are random in wireless channels [100, 101]. Network slicing algorithms for slice recovery and reconfiguration under stochastic demands in service-oriented networks are studied in [124]. Wang *et al.* use ℓ_1-norm to promote sparsity in slice reconfigurations in [122]. To tackle the non-differentiability of ℓ_1-norm, affine constraints that allow limited slice variations are considered instead of ℓ_1-norm in [122]. The statistics of the demand are not used in [122] for network slicing. In [129], Pozza *et al.* propose a heuristic divide-and-conquer approach for finding a sequence of feasible solutions to reconfigure a slice made of a chain of network functions under bandwidth and latency constraints, while RAN is not considered in the formulations. To tackle the high computational complexity caused by the large number of variables in the slice reconfiguration problem, a heuristic depth-first-search algorithm is proposed in [134] to find a set of possible reconfigurations, and a reinforcement

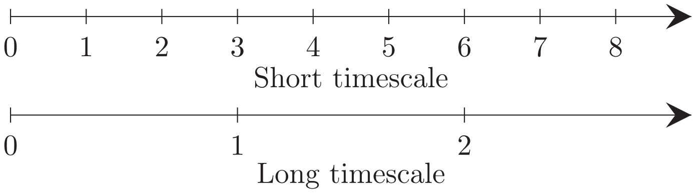


Figure 4.1: The time horizon slotted in two-timescales.

learning approach is used to explore the multi-dimensional discrete action space. An end-to-end network slicing method for 5G networks without imposing constraints on available resources is proposed in [135], where the slice status is monitored and necessarily reconfigured in an online fashion via a heuristic approach. However, reconfiguring end-to-end slices based on instantaneous demands significantly increases the reconfiguration costs.

Network slicing is carried out for a certain time period, which is called *timescale*. Unlike most papers, e.g., [128, 127, 133], that slice the network based on traffic variations in single timescale, a few recent papers, e.g., [136, 89, 137], propose two-timescale frameworks to improve network management and efficiency of resource allocations. In two-timescales frameworks, network resources are grouped and managed according to two different criteria in two different time periods as depicted in Fig. 4.1. In practice, in long timescales, costly network configurations are determined, and less costly reconfigurations are implemented in short timescales to better adapt the network to a changing environment. In [89], a two-timescale resource management scheme for network slicing in cloud RAN is proposed. Zhang *et al.* propose a long timescale inter-slice resource reservation for slices and a short timescale intra-slice resource allocation in [89]. However, the formulation of [89] does not consider the slice reconfiguration cost, which is non-continuous and non-convex [122], nor the resource reservation in the backhaul. Thus, [89] fails to provide end-to-end QoS guarantees to users. A multi-timescale decentralized online orchestration of software-defined networks is studied in [138], where a set of network controllers are activated based on the temporal and spatial variations in traffic requests. The slice activation and reconfiguration in a two-timescale framework are not studied in the existing literature; these topics comprise the main focus of this work.

4.2 Contributions

In this chapter, we propose a two-timescale resource management scheme for end-to-end reservation-based network slicing in the backhaul and RAN. In both timescales, the expected utilities of network tenants from serving users are maximized through two different mechanisms. In the long timescale, each tenant decides whether or not to activate a slice to serve users, while in the short timescale, the tenant reconfigures active slices to make them adaptive to demands of users and channel states. Resource management in both timescales is implemented under two major assumptions: the user traffic and channel states vary over time and they are uncertain.

In the long timescale, we design a slice utility function for each tenant based on the expected acquired revenue from users, the expected outage of downlinks, and the cost of slice activation. We formulate a sparse mixed-binary optimization problem to activate network slices if the expected utility of a tenant significantly improves after the activation. We use the ℓ_q, $0 < q < 1$, regularization to tackle the non-convexity and non-continuity of the slice activation cost and also to promote binary solutions in the relaxed problem. We propose a Frank-Wolfe algorithm to successively minimize convex approximations of the original problem and jointly implement the slice configuration in the backhaul and RAN. Via numerical tests, we demonstrate that the proposed method obtains solutions that are near to the optimal ones. To the best of our knowledge, this is the first endeavor to mathematically study the sparse slice activation problem for unseen user traffic based on derived statistics.

For network slicing in the short timescale, we design a slice utility function for each tenant based on the acquired revenue from users, the QoS that the slice guarantees for its users, and the cost of slice reconfiguration. We formulate a sparse optimization problem to adaptively reconfigure network slices if the expected utility of a tenant significantly changes after the reconfiguration. We use the group LASSO regularization to tackle the non-convexity and non-continuity of the slice reconfiguration cost. We propose an ADMM algorithm to solve each (non-convex) group LASSO subproblem in the short timescale. The proposed ADMM algorithm implements the slice reconfiguration in the backhaul through link capacity reservation via a fast, distributed algorithm that successively minimizes a convex approximation of the objective function and parallelizes computations across backhaul

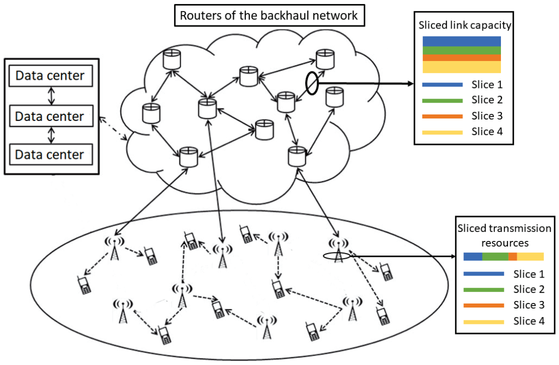


Figure 4.2: A network comprised of RUs and backhaul parts.

links. Furthermore, the proposed ADMM algorithm implements the slice reconfiguration in RAN through the transmission resource reservation for slices using 1) a proximal gradient descent method that decomposes the problem across slices; and 2) a bisection search method. We prove that the proposed ADMM algorithm converges to the global solution of each group LASSO subproblem despite its non-convexity. Extensive numerical simulations verify that the proposed approach outperforms the existing state-of-the-art method.

The rest of this chapter is organized as follows. The system model is given in Section 4.3. In Section 4.4, we formulate the optimization for slice activation in long timescales and the optimization for slice reconfiguration in short timescales. In Sections 4.5 and 4.6, we propose approaches to solve problems in long and short timescales, respectively. The simulation results are given in Section 4.7, and concluding remarks are given in Section 4.8.

4.3 System Model and Problem Formulation

Consider a typical scenario whereby user data is transmitted via backhaul network links from data centers to multiple geographically separated Radio Units (RUs) in RAN as depicted in Fig. 4.2. To serve each user, multi-connectivity is considered and multiple RUs jointly transmit data to each user. We denote the set of mobile users by $\mathcal{K} = \{1, \ldots, K\}$ and represent the set of RUs in RAN by $\mathcal{B}$. Furthermore, let $\mathcal{L}$ represent the set of backhaul links. We make no assumption on the type of a backhaul link. Depending on the requirements of a network and the implementation costs, either fiber line, coaxial, etc., is chosen. The user-RU association and downlinks between RUs and users are predetermined according to average interference, path loss, shadowing and fading. A *path* connects a data center and an RU through a sequence of wired links in the backhaul and then goes through one downlink to reach the end user. The first and last nodes of a path are a data center and a user, respectively. The node before the last node is an RU in RAN. The other nodes are network routers that route the traffic on the backhaul. Since each user is served by multiple RUs, several candidate paths are considered to connect RUs to a data center. These paths share an identical origin (the data center) and destination (the end user). As RUs are geographically separated, multiple paths carry data to transmitting RUs, and this multi-path routing in the backhaul avoids congestions and increases user throughput. We denote a path by p and represent the set of paths selected to carry user k data by $\mathcal{P}_k$. We assume that a single commodity is requested by a user, and therefore, there are K datastreams in the backhaul network. The proposed framework can be easily extended to a scenario in which each user demands multiple commodities. We represent the total reserved rate to serve user k via multiple paths by r_k. We consider that a data center can provide any data rate that a user demands; however, the network resources in terms of backhaul link capacity or transmission resources in RAN limit the network ability to thoroughly support a user demand.

Suppose that a network tenant is denoted by j where $\mathcal{J} = \{1, \dots, J\}$ is the set of all tenants. Each tenant owns several slices, which differ in supported features and network function optimization. Multiple slices, which deliver similar features, can also be deployed by each tenant. However, they are responsible for serving different groups of users. A network slice is represented by s and the set of slices possessed by tenant j is denoted by $\mathcal{S}_j$. The set of users served by slice s is denoted by $\mathcal{K}_s$ and the set of users served by tenant j is represented by $\mathcal{K}_j = \{\mathcal{K}_s\}_{s\in\mathcal{S}_j}$. The user-slice association is known and fixed [139, 128, 133]. For user-slice associations, see [66, 140].

In the considered model, each path belongs to one slice. The set of backhaul links on path p (to serve user k) is denoted by $\mathcal{L}_k^p$. The set of network nodes on path p is denoted by $\mathcal{U}_k^p$. The reserved rate for path p to serve user k is denoted by r_k^p. To wirelessly transmit the incoming data from each path, transmission resources should be sliced and reserved in RUs. The two physical constraints that limit network resource slicing are as follows:

- The aggregate amount of reserved traffic for those paths that go through a link cannot exceed the link capacity:

$$\sum_{k=1}^{K} \sum_{p:\{p\in\mathcal{P}_k, l\in\mathcal{L}_k^p\}} r_k^p \leq C_l, \qquad \forall l \in \mathcal{L}, \tag{4.1}$$

  where C_l is the capacity of link l.

- The available resources in an RU are limited and are allocated to different downlinks created by that RU. The overall reserved transmission resources for those paths that share an RU must not exceed its capacity:

$$\sum_{k=1}^{K} \sum_{p:\{p\in\mathcal{P}_k, b\in\mathcal{U}_k^p\}} t_k^p \leq C_b, \qquad \forall b \in \mathcal{B}, \tag{4.2}$$

  where t_k^p is the reserved transmission resource of RU b to transmit incoming data from path p. Moreover, C_b is the capacity of RU b.

- The minimum aggregate reserved rate for users served by slice s and tenant j are denoted by R_s^{slc} and R_j^{ten}, respectively, and we have

$$\sum_{k\in\mathcal{K}_s} \sum_{p\in\mathcal{P}_k} r_k^p \geq R_s^{\text{slc}}, \forall s, \text{ and } \sum_{k\in\mathcal{K}_j} \sum_{p\in\mathcal{P}_k} r_k^p \geq R_j^{\text{ten}}, \forall j. \tag{4.3}$$

  The constraints for minimum transmission resources for each slice and each tenant are as follows:

$$\sum_{k\in\mathcal{K}_s} \sum_{p\in\mathcal{P}_k} t_k^p \geq B_s^{\text{slc}}, \forall s, \text{ and } \sum_{k\in\mathcal{K}_j} \sum_{p\in\mathcal{P}_k} t_k^p \geq B_j^{\text{ten}}, \forall j. \tag{4.4}$$

In addition to the above constraints, our multi-path model imposes another constraint.

- The total reserved traffic for different paths that carry data to one user is equal to the reserved rate for that user. Hence, we have the following constraint:

$$\sum_{p\in\mathcal{P}_k} r_k^p = r_k, \qquad \forall k. \tag{4.5}$$

We denote the reserved link capacity for slice s on link l by r_s^l, which is calculated as follows:

$$r_s^l = \sum_{k \in \mathcal{K}_s} \sum_{p:\{p \in \mathcal{P}_k, l \in \mathcal{L}_k^p\}} r_k^p, \qquad \forall s, \forall l \in \mathcal{L}. \tag{4.6}$$

Similarly, we denote the reserved transmission resources at RU b by t_s^b, which is calculated as follows:

$$t_s^b = \sum_{k \in \mathcal{K}_s} \sum_{p:\{p \in \mathcal{P}_k, b \in \mathcal{U}_k^p\}} t_k^p, \qquad \forall s, \forall b \in \mathcal{B}. \tag{4.7}$$

Furthermore, we define vectors of reserved resources for slices in the backhaul and RAN as $\mathbf{r}_s = \{r_s^l\}_{l \in \mathcal{L}}$ and $\mathbf{t}_s = \{t_s^b\}_{b \in \mathcal{B}}$. End-to-end network slicing ensures that sufficient transmission resources are reserved to transmit a data flow to a user, and also sufficient backhaul link capacity is reserved is carry data to RUs in order to utilize the reserved transmission resources.

4.4 Adaptive Multi-Tenant Network Slicing

In this section, we propose a reservation-based network slicing approach. With the user demand and channel statistics, we adaptively optimize the activation and reconfiguration of network slices. Each tenant predicts future user demands and based on the expected revenue from serving users, it activates a number of its slices. After tenants activate a number of their slices to serve users, resource reservation for different slices is adaptively reconfigured across the network such that:

1. The expected revenue of tenants is maximized;
2. The slice reconfiguration cost is minimized;
3. The maximum isolation among reserved resources for slices is enforced; and
4. The QoS requirements of users are met.

With a slotted time horizon, we consider a *two-timescale* scheme to reserve resources for slices in the network. Resource reservations are carried out such that the revenue of tenants is maximized in long and short timescales. Each tenant activates a subset of its slices to provide services to users in a long timescale, while it reconfigures activated slices based on the statistics of user demands and channel capacities in short timescales to improve the robustness of resource reservations and enhance QoS for users. In each timescale, the statistics of user demands and channel capacities remain identical within that time period or change slowly. Examples for the duration of long and short timescales are several days and a couple of hours, respectively [129, 141]. We consider that the long and short timescale durations are predetermined and kept fixed over time.

4.4.1 User Demand and Downlink Statistics

We add a subscript S to the derived statistics for short timescales and L to those derived for long timescales. The demand of user k, represented by d_k, follows a certain PDF, denoted by $f_{k,L_n}(d_k)$, in the n^{th} long timescale with a Cumulative Density Function (CDF) $F_{k,L_n}(d_k)$. Using the reserved rate for user k, r_k, the supportable demand of user k is $\min(d_k, r_k)$ since the network can only provide r_k to user k if the demand of user k exceeds r_k. Within each timescale, a tenant collects user demands served by a slice and stores them. Collected samples are later used to estimate each PDF using Wolverton and Wagner estimator in [115]. The bandwidth in the recursive PDF estimator is chosen such that the estimator gradually forgets old samples, and gives a higher priority to new samples.

In a practical fading environment, the transmission rate to each user in the coverage area depends on the random channel capacity (i.e., instantaneous achievable rate), which is a function of the amount of resources, e.g., bandwidth, supplied to the downlink [100, 101]. In the proposed model, we do not make any assumption about the type of allocated transmission resources in RUs. This can be power, bandwidth, or time-slot. Since a path connects a data center to a user, a given path uniquely identifies the downlink by which a user is served. The achievable rate of a downlink is denoted by v_k^p. Typically, achievable rates are discrete based on available modulation and coding schemes. We consider that v_k^p can be approximated by a continuous variable and follows any arbitrary PDF. Let $z_{k,L_n}^p(v_k^p, t_k^p)$ denote the PDF of the achievable rate of the downlink of path p in the n^{th} long timescale, where the amount of the transmission resource supplied to the downlink is t_k^p. The CDF of the downlink achievable rate is denoted by $Z_{k,L_n}^p(v_k^p, t_k^p)$ in the n^{th}

Table 4.1: Notations

k	a user
p	a path
l	a backhaul link
b	an RU
$\mathcal{L}$	the set of backhaul links
$\mathcal{B}$	the set of RUs
$\mathcal{L}_k^p$	set of links on path p to serve user k
$\mathcal{U}_k^p$	set of nodes on path p to serve user k
r_k	reserved rate for user k
r_k^p	reserved rate for path p to serve user k
w	a wireless downlink
d_k	random demand of user k
t_k^p	reserved transmission resource for path p to serve user k
$f_{k,S_n}(d_k)$	PDF for user k demand
$\mathcal{P}_k$	the set of paths that carry user k data
$z_{k,S_n}^p(v_k^p, t_k^p)$	PDF of the achievable rate of downlink of path p
K	the number of users in the network
$\mathcal{W}_k$	the set of downlinks for serving user k
C_l	the capacity of backhaul link l
C_b	transmission resource budget for RU b
$\phi_{k,L_n}(r_k^p)$	revenue function by serving user k in a long timescale
μ_l	the Lagrange multiplier for link l
$\phi_{k,S_n}(r_k^p)$	revenue function by serving user k in a short timescale
θ_s	a constant to balance expected outage
$\mathcal{K}_s$	set of users served by slice s
$x_{s,1}$	an indicator that slice s is activated or not
c_a	the cost of activating a slice
c_r	the cost of reconfiguring a slice
R_s^{slc}	minimum reserved rate for slice s
R_j^{ten}	minimum reserved rate for tenant j
B_s^{slc}	minimum reserved transmission resource for slice s
B_j^{ten}	minimum reserved transmission resource for tenant j

long timescale and determines the probability that the achievable rate of the downlink of path p to serve user k is at most v_k^p. After the user data is transmitted, the control plane starts to collect the channel state information feedback sent by the user to each RU. Based on observed achievable rate information fed back to an RU, a recursive PDF estimator is deployed to derive the statistics for a downlink. When the random achievable rate of a downlink is less than the reserved rate r_k^p, the experienced outage is $r_k^p - v_k^p$. The probability that this amount of outage occurs in the n^{th} short timescale is $z_{k,S_n}^p(v_k^p, t_k^p)$. In light of the above arguments, the expected value of the outage of the downlink of path p is obtained as

$$\int_0^{r_k^p} z_{k,S_n}^p(v_k^p, t_k^p)\,(r_k^p - v_k^p)dv_k^p, \qquad \forall p \in \mathcal{P}_k, \forall k. \tag{4.8}$$

Since the achievable rate is a continuous random variable, we have the above integral. As $z_{k,S_n}^p(v_k^p, t_k^p)$ is always non-negative, then the expected outage is non-decreasing in r_k^p. In addition, we assume that the expected outage is non-increasing in t_k^p.

Consider that $\phi_{k,S_n}(\cdot)$ is the revenue function of a tenant from serving user k in the n^{th} short timescale. We consider $\phi_{k,S_n}(\cdot)$ to be a concave and non-decreasing function, e.g., $\phi_{k,S_n}(x) = 1 - \exp(-x), x \geq 0$. The expected revenue gained from serving user k is calculated as follows:

$$\mathbb{E}_{d_k}\left[\phi_{k,S_n}\Big(\min(r_k, d_k)\Big)\right] = \int_0^{r_k}\phi_{k,S_n}(y)f_{k,S_n}(y)dy$$
$$+\int_{r_k}^{\infty}\phi_{k,S_n}(r_k)f_{k,S_n}(y)dy. \tag{4.9}$$

In the first integral, the random demand lies in $[0,r_k]$, and in the second $d_k \in [r_k,\infty)$. Using this revenue function, we can maximize the expected supportable rates of users through the maximization of the revenue functions.

Consider that $\phi_{k,L_n}(\cdot)$ is the revenue function of a tenant acquired by serving user k in long timescales. We consider $\phi_{k,L_n}(\cdot)$ to be a concave and non-decreasing function. In long timescales, the set of users served by each slice, denoted by $\mathcal{K}_s$, is random and follows a certain probability mass function. To find the expected revenue of each slice from serving users in a long timescale, we need to also take an expectation with respect to $\mathcal{K}_s$. The expected gained revenue by slice s is the summation of expected revenue obtained from users served by slice s [139] and is calculated as follows:

$$\mathbb{E}_{\mathcal{K}_s}\left[\sum_{k\in\mathcal{K}_s}\mathbb{E}_{d_k}\Big[\phi_{k,L_n}\Big(\min(r_k,d_k)\Big)\Big]\right]. \tag{4.10}$$

Since the set of users served by each slice is uncertain in long timescales, we need to take an expectation with respect to $\mathcal{K}_s$ from (4.1)–(4.4) and (4.8). The argument of revenue functions $\phi_{k,S_n}(\cdot)$ and $\phi_{k,L_n}(\cdot)$ in (4.9) and (4.10), which is $\min(r_k, d_k)$, is a concave function of r_k. Since $\phi_{k,S_n}(\cdot)$ and $\phi_{k,L_n}(\cdot)$ are both non-decreasing concave functions, based on the rule for function compositions in [108, eq. (3.10)], revenue functions are also concave. After taking expectations, (4.9) and (4.10) remain concave functions.

4.4.2 Slice Activation

A tenant operates a slice only if the acquired revenue from activating that slice is considerable. The slice activation involves binary variables, which determine whether or not a slice is activated. Suppose that $x_{s,1}$ is a binary variable and if $x_{s,1} = 1$, then slice s is activated and $x_{s,1} = 0$ if slice s is not activated. When one slice is not activated due to operational costs, its potential users are served by other active slices of the same tenant that provide similar services. In addition, we consider the binary $x_{s,2}$ as a complement variable for $x_{s,1}$ such that

$$x_{s,1} + x_{s,2} = 1, \qquad \forall s \in \mathcal{S}_j, \forall j. \tag{4.11}$$

The cost of activating a slice is denoted by c_a and the cost of activated slices of tenant j is

$$c_a \left\|\{x_{s,1}\}_{s\in\mathcal{S}_j}\right\|_0 . \tag{4.12}$$

When $x_{s,1} = 1$, this non-zero element is counted by the ℓ_0-norm. While each tenant desires to maximize the expected gained revenue from users by increasing reserved rates, to improve the user QoS, it minimizes the expected outage of downlinks. Therefore, the overall utility function of the j^{th} tenant from the activation of a subset of slices is

$$\sum_{s\in\mathcal{S}_j}\mathbb{E}_{\mathcal{K}_s}\Bigg[\sum_{k\in\mathcal{K}_s}\mathbb{E}_{d_k}\Big[\phi_{k,L_n}\Big(\min(r_k,d_k)\Big)\Big]\Bigg] - c_a\left\|\{x_{s,1}\}_{s\in\mathcal{S}_j}\right\|_0$$
$$-\sum_{s\in\mathcal{S}_j}\theta_s\mathbb{E}_{\mathcal{K}_s}\sum_{k\in\mathcal{K}_s}\left[\sum_{p\in\mathcal{P}_k}\int_0^{r_k^p} z_{k,L_n}^p(v_k^p,t_k^p)\,(r_k^p-v_k^p)dv_k^p\right], \tag{4.13}$$

where θ_s is a constant adjusted by the system engineer.

When a slice is activated, resources are reserved for that slice. To relate $x_{s,1}$ to the reserved resources for a slice, we consider the following constraints:

$$\mathbb{E}_{\mathcal{K}_s}\left[\sum_{k\in\mathcal{K}_s}\sum_{p\in\mathcal{P}_k} r_k^p\right] \geq x_{s,1}R_s^{\text{slc}}, \forall s\in\mathcal{S}_j, \forall j, \tag{4.14}$$

$$\mathbb{E}_{\mathcal{K}_s}\left[\sum_{k\in\mathcal{K}_s}\sum_{p\in\mathcal{P}_k} t_k^p\right] \geq x_{s,1}B_s^{\text{slc}}, \forall s\in\mathcal{S}_j, \forall j. \tag{4.15}$$

We note that (4.14) and (4.15) imply that the slice requirement constraints are only considered for each activated slice s. Additionally, we consider

$$\mathbb{E}_{\mathcal{K}_s}\left[\sum_{k\in\mathcal{K}_s}\sum_{p\in\mathcal{P}_k} r_k^p\right] \leq x_{s,1}\Psi, \forall s\in\mathcal{S}_j, \forall j, \tag{4.16}$$

$$\mathbb{E}_{\mathcal{K}_s}\left[\sum_{k\in\mathcal{K}_s}\sum_{p\in\mathcal{P}_k} t_k^p\right] \leq x_{s,1}\Psi, \forall s\in\mathcal{S}_j, \forall j, \tag{4.17}$$

to enforce zero resource reservation for a slice when $x_{s,1}=0$. In (4.16) and (4.17) , Ψ is a large number and when $x_{s,1}=1$, (4.16) and (4.17) are relaxed. The sparse slice activation optimization is formulated as follows:

$$\min_{\mathbf{r},\mathbf{t},\mathbf{x}} \qquad -\sum_{j=1}^{J}\psi_j\sum_{s\in\mathcal{S}_j}\mathbb{E}_{\mathcal{K}_s}\left[\sum_{k\in\mathcal{K}_s}\mathbb{E}_{d_k}\Big[\phi_{k,L_n}\Big(\min(r_k,d_k)\Big)\Big]\right]$$

$$+\sum_{j=1}^{J}\psi_j\sum_{s\in\mathcal{S}_j}\theta_s\mathbb{E}_{\mathcal{K}_s}\sum_{k\in\mathcal{K}_s}\left[\sum_{p\in\mathcal{P}_k}\int_0^{r_k^p} z_{k,L_n}^p(v_k^p,t_k^p)\,(r_k^p-v_k^p)dv_k^p\right]$$

$$+\sum_{j=1}^{J}c_a\left\|\{x_{s,1}\}_{s\in\mathcal{S}_j}\right\|_0 \tag{4.18a}$$

$$\text{s.t.} \quad \sum_{j=1}^{J}\sum_{s\in\mathcal{S}_j}\mathbb{E}_{\mathcal{K}_s}\left[\sum_{k\in\mathcal{K}_s}\sum_{p:\{p\in\mathcal{P}_k,l\in\mathcal{L}_k^p\}}r_k^p\right]\le C_l,\forall l \tag{4.18b}$$

$$\sum_{j=1}^{J}\sum_{s\in\mathcal{S}_j}\mathbb{E}_{\mathcal{K}_s}\left[\sum_{k\in\mathcal{K}_s}\sum_{p:\{p\in\mathcal{P}_k,b\in\mathcal{U}_k^p\}}t_k^p\right]\le C_b,\forall b \tag{4.18c}$$

$$\sum_{s\in\mathcal{S}_j}\mathbb{E}_{\mathcal{K}_s}\left[\sum_{k\in\mathcal{K}_s}\sum_{p\in\mathcal{P}_k}r_k^p\right]\ge R_j^{\text{ten}},\forall j, \tag{4.18d}$$

$$\sum_{s\in\mathcal{S}_j}\mathbb{E}_{\mathcal{K}_s}\left[\sum_{k\in\mathcal{K}_s}\sum_{p\in\mathcal{P}_k}t_k^p\right]\ge B_j^{\text{ten}},\forall j, \tag{4.18e}$$

$$(4.5),(4.11),(4.14),(4.15),(4.16),(4.17),$$

$$x_{s,1},x_{s,2}\in\{0,1\},r_k^p,t_k^p\ge 0,\forall p\in\mathcal{P}_k,\forall k,\forall s,$$

where ψ_j is a positive weight given to tenant j in order to adjust priorities. Moreover, $\mathbf{r}=\{r_k^p\}_{p\in\mathcal{P}_k,k\in\mathcal{K}}$, $\mathbf{t}=\{t_k^p\}_{p\in\mathcal{P}_k,k\in\mathcal{K}}$, and $\mathbf{x}=\{x_{s,1},x_{s,2}\}_{s\in\mathcal{S}_j,j\in\mathcal{J}}$.

Remark 5. *Suppose that multiple paths available to user k share a downlink (the last hop). The aggregate outage of downlinks for serving user k is calculated as follows:*

$$\sum_{w\in\mathcal{W}_k}\int_0^{\sum_{p:\{p\in\mathcal{P}_k,w\in p\}}r_k^p} z_{k,L_n}^w(v_k^w,t_k^w)(\sum_{p:\{p\in\mathcal{P}_k,w\in p\}}r_k^p-v_k^w)dv_k^w, \tag{4.19}$$

where $\mathcal{W}_k$ is the set of downlinks, each denoted by w, for serving user k. When multiple paths available to user k share a downlink, the above outage is placed in (4.18a) *instead of its second term, which includes* (4.8).

4.4.3 Slice Reconfiguration

Here, we consider slicing network resources, both in the backhaul and RAN, among several network tenants in short timescales. The number of users that are served by each slice is known in short timescales. As PDFs of user demands and achievable rates of downlinks change over time, allocated resources to each slice should be adapted. However, this change involves a reconfiguration cost and service interruptions. Therefore, an adaptive approach should reconfigure each slice only if the utility function for that tenant significantly changes with new PDFs. The state of a slice is reflected by the reserved resources in the backhaul and RAN. The reconfiguration cost of a slice is a discrete function of the state difference [122, 67]. We underline the vector of reserved resources in each short timescale. If we have $\left\|\underline{\mathbf{r}}_s^n - \underline{\mathbf{r}}_s^{n-1}\right\|_2 > 0$ or $\left\|\underline{\mathbf{t}}_s^n - \underline{\mathbf{t}}_s^{n-1}\right\|_2 > 0$, then slice s varies in two consecutive timescales. The reconfiguration cost is proportional to the number of layout changes for slices. We use ℓ_0-norm to detect reconfigurations for slices. Consider that the slice reconfiguration cost is denoted by c_r. To minimize the slice reconfiguration cost in the network from time $n-1$ to n, we minimize

$$c_r\big(\big\|\{\|\underline{\mathbf{r}}_s^n - \underline{\mathbf{r}}_s^{n-1}\|_2\}_{s\in\mathcal{S}_j,j\in\mathcal{J}}\big\|_0 + \big\|\{\|\underline{\mathbf{t}}_s^n - \underline{\mathbf{t}}_s^{n-1}\|_2\}_{s\in\mathcal{S}_j,j\in\mathcal{J}}\big\|_0\big). \tag{4.20}$$

Here, we discuss the isolation of reserved resources for slices. When the random traffic demand from slice s exceeds the reserved traffic rate for it, the tenant needs to allocate more resources to support the demand. This can violate the resource reservation for other slices and hurt the QoS for users served by other slices. Each slice desires to isolate its reserved resources as much as possible [142, 9]. Although static network slicing provides complete resource isolation among slices, it performs poorly due to its inflexibility in supporting time-varying user demands [143]. The dynamic network slicing better supports user demands, although it increases the risk for violation of resource isolation for slices. To increase the slice isolation, we minimize the expected excessive demand from each slice, which is calculated as follows:

$$\int_{\sum_{k\in\mathcal{K}_s}\sum_{p\in\mathcal{P}_k} r_k^p}^{\infty} (y - \sum_{k\in\mathcal{K}_s}\sum_{p\in\mathcal{P}_k} r_k^p) f_{s,S_n}(y)dy. \tag{4.21}$$

In the above expression, y is the integration variable and corresponds to the aggregate demanded rate from slice s, $\sum_{k\in\mathcal{K}_s} d_k$, and when $y \in [\sum_{k\in\mathcal{K}_s}\sum_{p\in\mathcal{P}_k} r_k^p,\infty)$, the slice is unable to support the demand. In (4.21), $f_{s,S_n}(\cdot)$ is the PDF for the aggregate demand from slice s. We utilize the data-driven density estimator given in [115] to numerically estimate $f_{s,S_n}(\cdot)$ for each short timescale.

Each slice provides its users with particular QoS guarantees. We consider that each slice guarantees that the expected outage (4.8) for a served user is less than a certain fraction, denoted by β_s, of the reserved rate for that user in a short timescale. Therefore, we have

$$\int_0^{r_k^p} z_{k,S_n}^p(v_k^p, t_k^p)\,(r_k^p - v_k^p)dv_k^p \leq \beta_s r_k^p, \qquad \forall k \in \mathcal{K}_s. \tag{4.22}$$

The objective function that we minimize is

$$\begin{aligned}\Gamma(\mathbf{r}) = &\sum_{j=1}^{J}\sum_{s\in\mathcal{S}_j}\int_{\sum_{k\in\mathcal{K}_s}\sum_{p\in\mathcal{P}_k} r_k^p}^{\infty} (y - \sum_{k\in\mathcal{K}_s}\sum_{p\in\mathcal{P}_k} r_k^p) f_{s,S_n}(y)dy \\ &- \sum_{j=1}^{J}\psi_j\Bigg(\sum_{s\in\mathcal{S}_j}\sum_{k\in\mathcal{K}_s}\mathbb{E}_{d_k}\Bigg[\phi_{k,S_n}\Big(\min(\sum_{p\in\mathcal{P}_k} r_k^p, d_k)\Big)\Bigg]\Bigg).\end{aligned} \tag{4.23}$$

Using (4.23), the formulated joint optimization to minimize the reconfiguration costs, maximize the isolation of reserved resources, guarantee QoS for slice users, and maximize the utilities of tenants from serving their users in short timescales is as follows:

$$\begin{aligned}\min_{\mathbf{r},\mathbf{t}\geq\mathbf{0}} \quad & \Gamma(\mathbf{r}) + c_r \left\|\{\left\|\mathbf{r}_s - \underline{\mathbf{r}}_s^{n-1}\right\|_2\}_{s\in\mathcal{S}_j, j\in\mathcal{J}}\right\|_0 \\ & + c_r \left\|\{\left\|\mathbf{t}_s - \underline{\mathbf{t}}_s^{n-1}\right\|_2\}_{s\in\mathcal{S}_j, j\in\mathcal{J}}\right\|_0 \\ \text{s.t.} \quad & (4.1), (4.2), (4.3), (4.4), (4.6), (4.7), (4.22).\end{aligned} \tag{4.24}$$

Remark 6. *If multiple paths available to user k share a downlink, (4.22) can be rewritten as follows:*

$$\int_0^{\sum_{p:\{p\in\mathcal{P}_k, w\in p\}} r_k^p} z_{k,S_n}^w(v_k^w, t_k^w)(\sum_{p:\{p\in\mathcal{P}_k, w\in p\}} r_k^p - v_k^w)dv_k^w$$

$$\leq \beta_s(\sum_{p:\{p\in\mathcal{P}_k, w\in p\}} r_k^p), \qquad \forall k \in \mathcal{K}_s. \tag{4.25}$$

.

Our formulations allow multiple slices to be owned and operated by network tenants. Each tenant can operate one or multiple slices to support users within a local and private area of service. We consider different QoS supports for users served by active slices. Therefore, one can consider enhanced QoS for an exclusive group of users served by each private slice, compared to users of public slices. To enhance the performance of slices, our proposed approaches take account of local statistics including user demands and channel state information. We jointly optimize the allocation of available network resources to different slices. One realistic application of our approaches is spectrum slicing, where a dedicated chunk of the 5G spectrum is allocated to vertical industries for running 5G networks on their premises. Multiple spectrum slices are deployed to locally support user demands, e.g., the usage is limited to a factory site or a campus, or regionally. An example of the dedicated spectrum is the 100 MHz spectrum allocation for local operation in Germany in the frequency band of 3.7–3.8 GHz [144, 145].

4.5 The Proposed Approach for Slice Activation

In optimization (4.18), the constraints (4.14)–(4.17) and (4.18b)–(4.18e) can be rewritten in the affine form after we substitute a weighted average for the expectation with respect to the set of users associated with a slice, i.e., $\mathbb{E}_{\mathcal{K}_s}[\cdot]$. For example, we can rewrite (4.18b) in the following form:

$$\sum_{j=1}^{J}\sum_{s\in\mathcal{S}_j}\sum_{\mathcal{K}_s^u} P_u(\sum_{k\in\mathcal{K}_s^u}\sum_{p:\{p\in\mathcal{P}_k, l\in\mathcal{L}_k^p\}} r_k^p) \leq C_l, \quad l \in \mathcal{L},$$

where $\mathcal{K}_s^u$ is the u^{th} possible set of potential users served by slice s with a P_u probability of occurring. One can similarly rewrite (4.14)–(4.17) and (4.18c)–(4.18e) in the above affine form.

We let $\mathbf{x}_{j,1} = \{x_{s,1}\}_{s\in\mathcal{S}_j}$ and $\mathbf{x}_{j,2} = \{x_{s,2}\}_{s\in\mathcal{S}_j}$. Before developing an approach to solve the slice activation optimization, we consider the following ℓ_q-norm:

$$\left\|\mathbf{x}_{j,1} + \epsilon\mathbf{1}\right\|_q^q, \tag{4.26}$$

where $0 < q < 1$ and ϵ is a small positive number to make (4.26) differentiable. When q and ϵ are close to zero, one can use $c_a \left\|\mathbf{x}_{j,1} + \epsilon\mathbf{1}\right\|_q^q$ to efficiently approximate (4.12) [146, 147, 148, 70, 104, 105].

Although (4.26) approximates the ℓ_0-norm, we note that $\left\|\mathbf{x}_{j,1} + \epsilon\mathbf{1}\right\|_q^q$ is concave and (4.26) is not easy to minimize. Therefore, if we add $c_a \left\|\mathbf{x}_{j,1} + \epsilon\mathbf{1}\right\|_q^q$ to the objective function of (4.18), it is still hard to solve. To tackle this problem, we consider a quadratic upper-bound [149, eq. (12)] for (4.26) and successively minimize the upper-bound [19]. We denote the i^{th} iterate of $\mathbf{x}_{j,1}$ by $\underline{\mathbf{x}}_{j,1}^i$ and obtain the following upper-bound:

$$\begin{aligned}&\left\|\mathbf{x}_{j,1} + \epsilon\mathbf{1}\right\|_q^q \leq \left\|\underline{\mathbf{x}}_{j,1}^i + \epsilon\mathbf{1}\right\|_q^q \\ &+ q\big\langle\{(\underline{x}_{s,1}^i + \epsilon)^{q-1}\}_{s\in\mathcal{S}_j}, (\mathbf{x}_{j,1} - \underline{\mathbf{x}}_{j,1}^i)\big\rangle + \frac{e}{2}\left\|\mathbf{x}_{j,1} - \underline{\mathbf{x}}_{j,1}^i\right\|_2^2,\end{aligned} \tag{4.27}$$

where $e > 0$ is a small number and $\langle\cdot,\cdot\rangle$ represents the inner product. We also relax the constraint $x_{s,1}, x_{s,2} \in \{0,1\}$ and instead include $0 \leq x_{s,1} \leq 1$ and $0 \leq x_{s,2} \leq 1$. We substitute the RHS of (4.27) for $\left\|\{x_{s,1}\}_{s\in\mathcal{S}_j}\right\|_0$ and iteratively solve (4.18). The above upper-bound approximation is locally tight up to the first order. In other words, in any arbitrary point $\mathbf{x}_{j,1} = \underline{\mathbf{x}}_{j,1}^i$, $\left\|\mathbf{x}_{j,1} + \epsilon\mathbf{1}\right\|_q^q$ and the upper-bound have the same value and the same gradient. The above upper-bound is also continuous. Thus, the upper-bound satisfies all four convergence conditions given in [19, Assumption 2]. Based on [19, Theorem 2], the obtained solution by the successive upper-bound minimization is a stationary (KKT) solution to problem (4.18) with relaxed $\mathbf{x}$ and (4.26) in the objective function.

We note that (4.18) is convex with respect to $\mathbf{r}$. However, (4.18a) is not always convex in $\mathbf{t}$. To address this issue, we add the following proximal terms

$$\sum_{j=1}^{J}\sum_{k\in\mathcal{K}_j}\sum_{p\in\mathcal{P}_k}\frac{\zeta_k^p}{2}\left\|t_k^p - \hat{t}_k^p\right\|_2^2, \tag{4.28}$$

with sufficient weights $\zeta_k^p > 0$ to the objective function of (4.18) with the purpose of convexifying the objective function with respect to $\mathbf{t}$ locally. In the proximal term, $\hat{t}_k^p$ is the most recent iterate. The objective function with proximal terms is an upper-bound for the original objective function, and we successively minimize the upper-bound [19].

To efficiently solve (4.18) in each iteration of minimizing (4.18a) with (4.27) and (4.28), we propose a Frank-Wolfe algorithm. The Frank-Wolfe algorithm linearizes the objective function and finds a feasible descent direction within the set of constraints [111, Sec. 3.2.2]. We superscript the m^{th} iterate of the Frank-Wolfe algorithm by m. Using the linearized objective function, the direction-finding subproblem in the $m+1^{\text{th}}$ iteration of the Frank-Wolfe algorithm becomes

$$\begin{aligned}\min_{\bar{\mathbf{r}},\bar{\mathbf{t}},\bar{\mathbf{x}}\geq\mathbf{0}} \quad & \sum_{j=1}^{J}\psi_j\sum_{s\in\mathcal{S}_j}\sum_{\mathcal{K}_s^u}P_u\Big(\sum_{k\in\mathcal{K}_s^u}\sum_{p\in\mathcal{P}_k}\omega_k^p(r_k^{p,m},t_k^{p,m})\bar{t}_k^p\Big)\\ & +\sum_{j=1}^{J}\psi_j\sum_{s\in\mathcal{S}_j}\sum_{\mathcal{K}_s^u}P_u\Big(\sum_{k\in\mathcal{K}_s^u}\sum_{p\in\mathcal{P}_k}\chi_k^p(\{r_k^{p,m}\}_{p\in\mathcal{P}_k},t_k^{p,m})\bar{r}_k^p\Big)+\sum_{j=1}^{J}\sum_{s\in\mathcal{S}_j}\varrho_{s,1}^1(x_{s,1}^m)\bar{x}_{s,1} \quad (4.29)\\ \text{s.t.} \quad & (4.11),(4.14)-(4.17),(4.18\text{b})-(4.18\text{e}),0\leq\bar{x}_{s,1},\bar{x}_{s,2}\leq 1,\end{aligned}$$

where $\omega_k^p(r_k^{p,m},t_k^{p,m})$ and $\chi_k^p(\{r_k^{p,m}\}_{p\in\mathcal{P}_k},t_k^{p,m})$ are

$$\begin{aligned}&\omega_k^p(r_k^{p,m},t_k^{p,m})=\zeta_k^p(t_k^{p,m}-\hat{t}_k^p)\\ &+\theta_s\int_0^{r_k^{p,m}}\Big(r_k^{p,m}-v_k^p\Big)\frac{\partial z_{k,L_n}^p(v_k^p,t_k^p)}{\partial t_k^p}\,|_{t_k^p=t_k^{p,m}}\,dv_k^p,\end{aligned} \quad (4.30)$$

and

$$\begin{aligned}&\chi_k^p(\{r_k^{p,m}\}_{p\in\mathcal{P}_k},t_k^{p,m})=\theta_s Z_{k,L_n}^p(r_k^{p,m},t_k^{p,m})\\ &-\left(\phi'_{k,L_n}\Big(\sum_{p\in\mathcal{P}_k}r_k^{p,m}\Big)\Big(1-F_{k,L_n}(\sum_{p\in\mathcal{P}_k}r_k^{p,m})\Big)\right).\end{aligned} \quad (4.31)$$

Furthermore, $\varrho_{s,1}^1(x_{s,1}^m)$ is calculated as follows:

$$\varrho_{s,1}^1(x_{s,1}^m)=c_a\left(\frac{q}{(\underline{x}_{s,1}^i+\epsilon)^{1-q}}+e(x_{s,1}^m-\underline{x}_{s,1}^i)\right). \quad (4.32)$$

Algorithm 11: The proposed Frank-Wolfe Algorithm

```
0. Initialization Assign values to $\mathbf{r}^0$, $\mathbf{t}^0$, $\mathbf{x}^0_{j,1}$, $\mathbf{x}^0_{j,2}$, $m = 0$;
repeat
    1. $\hat{\mathbf{t}} = \mathbf{t}^m$;
    repeat
        2. Find $\omega^p_k(r^{p,m}_k, t^{p,m}_k)$, $\chi^p_k(\{r^{p,m}_k\}_{p\in\mathcal{P}_k}, t^{p,m}_k)$, and $\varrho^1_{s,1}(x^m_{s,1})$;
        3. Solve (4.29) and update variables using (4.33a)–(4.33d);
        4. $m = m + 1$;
    until $\mathbf{r}^m, \mathbf{t}^m, \mathbf{x}^m$ converges;
until $\hat{\mathbf{t}}$ converges;
5. $\underline{\mathbf{x}}^{i+1} = \mathbf{x}^m$;
```

The solution to (4.29) can be easily obtained using CPLEX or Gurobi. We update variables in the $m + 1^{\text{th}}$ iteration of the Frank-Wolfe algorithm as follows:

$$t^{p,m+1}_k = t^{p,m}_k + \pi^m(\bar{t}^p_k - t^{p,m}_k), \tag{4.33a}$$

$$r^{p,m+1}_k = r^{p,m}_k + \pi^m(\bar{r}^p_k - r^{p,m}_k), \tag{4.33b}$$

$$x^{m+1}_{s,1} = x^m_{s,1} + \pi^m(\overline{x}_{s,1} - x^m_{s,1}), \tag{4.33c}$$

$$x^{m+1}_{s,2} = x^m_{s,2} + \pi^m(\overline{x}_{s,2} - x^m_{s,2}), \tag{4.33d}$$

where $\pi^m = \frac{2}{2+m}$. We update $\omega^p_k(r^{p,m+1}_k, t^{p,m+1}_k)$, $\chi^p_k(\{r^{p,m+1}_k\}_{p\in\mathcal{P}_k}, t^{p,m+1}_k)$, and $\varrho^1_{s,1}(x^{m+1}_{s,1})$ as given in (4.30), (4.31), and (4.32), respectively. We continue until each $r^{p,m}_k$, $t^{p,m}_k$, $x^m_{s,1}$, and $x^m_{s,2}$ converges. After the convergence of the Frank-Wolfe algorithm, we update $\hat{\mathbf{t}} = \mathbf{t}^m$. We continue solving with the Frank-Wolfe algorithm until $\hat{\mathbf{t}}$ converges. The summary of the proposed Frank-Wolfe approach is given in Algorithm 11.

Remark 7. *If* (4.19) *is considered in* (4.18a)*, then* (4.30) *and* (4.31) *are changed to*

$$\begin{aligned}
&\omega^w_k(\{r^{p,m}_k\}_{p:\{p\in\mathcal{P}_k, w\in p\}}, t^{p,w}_k)\\
&= \theta_s \int_0^{\sum_{p:\{p\in\mathcal{P}_k, w\in p\}} r^{p,m}_k} \Big(\sum_{p:\{p\in\mathcal{P}_k, w\in p\}} r^{p,m}_k - v^w_k \Big)\\
&\times \frac{\partial z^p_{k,L_n}(v^w_k, t^w_k)}{\partial t^w_k} |_{t^w_k = t^{w,m}_k} \, dv^w_k + \zeta^w_k(t^{w,m}_k - \hat{t}^w_k), w \in p, \forall k,
\end{aligned}$$

and

$$\chi_k^p(\{r_k^{p,m}\}_{p\in\mathcal{P}_k}, t_k^{w,m}) = \theta_s Z_{k,L_n}^p(\sum_{p:\{p\in\mathcal{P}_k, w\in p\}} r_k^{p,m}, t_k^{w,m})$$
$$-\left(\phi'_{k,L_n}\Big(\sum_{p\in\mathcal{P}_k} r_k^{p,m}\Big)\Big(1 - F_{k,L_n}(\sum_{p\in\mathcal{P}_k} r_k^{p,m})\Big)\right), w \in p, \forall k.$$

After Algorithm 11 converges, we update $\underline{\mathbf{x}}^{i+1}$ as $\underline{\mathbf{x}}^{i+1} = \mathbf{x}^m$ and solve again with Algorithm 11. When we solve a few iterations with Algorithm 11, it is possible that $\underline{\mathbf{x}}^i$ does not become binary. We design a penalty term using ℓ_q-regularization to promote binary solutions. Consider the following optimization problem:

$$\min_{\mathbf{x}_s} \quad \|\mathbf{x}_s + \epsilon\mathbf{1}\|_q^q \tag{4.34a}$$
$$\text{s.t.} \quad (4.11), 0 \le x_{s,1} \le 1, 0 \le x_{s,2} \le 1, \forall s, \tag{4.34b}$$

where $\mathbf{x}_s = \{x_{s,1}, x_{s,2}\}$. The optimal solution of (4.34) is always binary, i.e., $x_{s,1}^*, x_{s,2}^* \in \{0, 1\}$ [50, 51]. Similar to (4.26), (4.34a) is concave. Therefore, we consider a quadratic upper-bound [149, eq. (12)] for it as follows:

$$\|\mathbf{x}_s + \epsilon\mathbf{1}\|_q^q \le \left\|\underline{\mathbf{x}}_s^i + \epsilon\mathbf{1}\right\|_q^q + q\Big\langle \left\{(\underline{x}_{s,1}^i + \epsilon)^{q-1}, (\underline{x}_{s,2}^i + \epsilon)^{q-1}\right\}$$
$$, (\mathbf{x}_s - \underline{\mathbf{x}}_s^i)\Big\rangle + \frac{e}{2}\left\|\mathbf{x}_s - \underline{\mathbf{x}}_s^i\right\|_2^2, \quad \forall s. \tag{4.35}$$

We can give a weight γ^i to the RHS of (4.35) and include it for all slices in (4.18a) to promote binary solutions for $x_{s,1}$ and $x_{s,2}$. We iteratively solve (4.18) with Algorithm 11. In the $m+1^{\text{th}}$ iteration of the Frank-Wolfe algorithm, the linearized quadratic upper-bound (4.35) becomes

$$\varrho_{s,1}^2(x_{s,1}^m)x_{s,1} + \varrho_{s,2}^2(x_{s,2}^m)x_{s,2}, \tag{4.36}$$

where

$$\varrho_{s,1}^2(x_{s,1}^m) = \left(\frac{q}{(\underline{x}_{s,1}^i + \epsilon)^{1-q}} + e(x_{s,1}^m - \underline{x}_{s,1}^i)\right),$$

Algorithm 12: Proposed algorithm to solve (4.18)

```
0. Initialization Assign values to $\underline{\mathbf{x}}^0$, $i = 0$;
repeat
    1. Apply Algorithm 11 to find $\underline{\mathbf{x}}^{i+1}$;
    2. Update $\gamma^{i+1}$;
    3. $i = i + 1$;
until $\underline{\mathbf{x}}^i$ becomes binary;
```

$$\varrho^2_{s,2}(x^m_{s,2}) = \left(\frac{q}{(\underline{x}^i_{s,2} + \epsilon)^{1-q}} + e(x^m_{s,2} - \underline{x}^i_{s,2}) \right).$$

We exclude the linearized RHS of (4.35), i.e., (4.36), in the first I steps of applying Algorithm 11 to solve (4.18). Therefore, we set $\gamma^i = 0$ when $i \le I$. Next, we increase γ^i iteratively and continue with $\gamma^{i+1} \ge \gamma^i > 0$ when $i > I$. As the upper-bound in the RHS of (4.35) iteratively receives a higher weight, the solutions become closer to 0 or 1. The summary of the proposed approach to solve (4.18) is given in Algorithm 12. We iteratively continue solving with Algorithm 12 until each $x^m_{s,1}$ and $x^m_{s,2}$ becomes binary.

The requirement for the practical implementation of the proposed slice activation approach is a control center that collects statistical information from different slices and solves a central optimization problem. This involves a message passing overhead. Solving a central optimization problem for a large network can be costly. However, the linearization of the problem helps to lower the complexity of solving the problem.

4.6 The Distributed and Scalable Algorithm for Slice Reconfiguration

Problem (4.24) is difficult to solve for two reasons: 1) ℓ_0-norm is neither convex nor continuous; and 2) (4.22) is not necessarily a convex constraint in r^p_k and t^p_k for an arbitrary $z^p_{k,\mathcal{S}_n}(v^p_k, t^p_k)$. To tackle the difficulty of solving (4.24), we copy variables as $\mathbf{r} = \mathbf{g}$ and decouple $\left\| \{ \left\| \mathbf{r}_s - \underline{\mathbf{r}}^{n-1}_s \right\|_2 \}_{s \in \mathcal{S}_j, j \in \mathcal{J}} \right\|_0$ and (4.22) from (4.24). To alleviate the non-convexity of (4.24) due to ℓ_0-norm, we use the group LASSO regularization [62] with copied $\mathbf{g}$. We substitute

$$\sum_{j=1}^{J}\sum_{s\in\mathcal{S}_j} c_r(a_s^{1,i}\left\|\mathbf{g}_s-\underline{\mathbf{r}}_s^{n-1}\right\|_2+a_s^{2,i}\left\|\mathbf{t}_s-\underline{\mathbf{t}}_s^{n-1}\right\|_2), \tag{4.37}$$

for ℓ_0-norms and iteratively minimize (4.37), where i is the iteration counter. The update rules for coefficients in the i^{th} iteration are

$$a_s^{1,i}=\frac{a_s^{1,0}}{\left\|\mathbf{g}_s^{i-1}-\underline{\mathbf{r}}_s^{n-1}\right\|_2+\epsilon_1} \text{ and } a_s^{2,i}=\frac{a_s^{2,0}}{\left\|\mathbf{t}_s^{i-1}-\underline{\mathbf{t}}_s^{n-1}\right\|_2+\epsilon_2}, \tag{4.38}$$

where $a_s^{1,0}, a_s^{2,0} > 0$. Furthermore, ϵ_1 and ϵ_2 are small positive numbers to ensure that a zero-valued norm in the denominator does not strictly prohibit $\left\|\mathbf{g}_s^{i}-\underline{\mathbf{r}}_s^{n-1}\right\|_2 = 0$ in the next step.

Using the group LASSO regularization, (4.24) can be approximated by the following problem:

$$\begin{aligned}\min_{\mathbf{r},\mathbf{t},\mathbf{g}\geq\mathbf{0}} \quad & \Gamma(\mathbf{r})+\sum_{j=1}^{J}\sum_{s\in\mathcal{S}_j} c_r a_s^{1,i}\left\|\mathbf{g}_s-\underline{\mathbf{r}}_s^{n-1}\right\|_2 \\ & +\sum_{j=1}^{J}\sum_{s\in\mathcal{S}_j} c_r a_s^{2,i}\left\|\mathbf{t}_s-\underline{\mathbf{t}}_s^{n-1}\right\|_2 \\ \text{s.t.} \quad & (4.1),(4.2),(4.3),(4.4),(4.6),(4.7),(4.22),\mathbf{r}=\mathbf{g}.\end{aligned} \tag{4.39}$$

To solve each subproblem of the group LASSO (4.39), we propose an ADMM algorithm. We dualize the constraint $\mathbf{r}=\mathbf{g}$ and find the augmented Lagrangian as follows:

$$\begin{aligned}L(\mathbf{r},\mathbf{t},\mathbf{g},\boldsymbol{\tau}) =& \sum_{j=1}^{J}\sum_{k\in\mathcal{K}_j}\sum_{p\in\mathcal{P}_k}\Big(\tau_k^p(r_k^p-g_k^p)+\frac{\rho}{2}(r_k^p-g_k^p)^2\Big) \\ &+\Gamma(\mathbf{r})+\sum_{j=1}^{J}\sum_{s\in\mathcal{S}_j} c_r(a_s^{1,i}\left\|\mathbf{g}_s-\underline{\mathbf{r}}_s^{n-1}\right\|_2+a_s^{2,i}\left\|\mathbf{t}_s-\underline{\mathbf{t}}_s^{n-1}\right\|_2),\end{aligned} \tag{4.40}$$

where τ_k^p is a Lagrange multiplier and ρ is a penalty parameter. We substitute (4.40) in the objective function of (4.39) and alternatively minimize with respect to $\mathbf{r}$ as one block and

$$
\begin{aligned}
&\underbrace{\sum_{j=1}^{J}\sum_{s\in\mathcal{S}_j}\int_{\sum_{k\in\mathcal{K}_s}\sum_{p\in\mathcal{P}_k}\hat{r}_k^p}^{\infty}(y-\sum_{k\in\mathcal{K}_s}\sum_{p\in\mathcal{P}_k}\hat{r}_k^p)f_{s,S_n}(y)dy}_{\Upsilon_1}\\
&-\sum_{j=1}^{J}\sum_{s\in\mathcal{S}_j}(1-F_{s,S_n}(\sum_{k\in\mathcal{K}_s}\sum_{p\in\mathcal{P}_k}\hat{r}_{k,s}^p))(\sum_{k\in\mathcal{K}_s}\sum_{p\in\mathcal{P}_k}r_k^p-\sum_{k\in\mathcal{K}_s}\sum_{p\in\mathcal{P}_k}\hat{r}_k^p)\\
&+\sum_{j=1}^{J}\sum_{s\in\mathcal{S}_j}\frac{\sum_{k\in\mathcal{K}_s}|\mathcal{P}_k|}{2}\sum_{k\in\mathcal{K}_s}\sum_{p\in\mathcal{P}_k}(r_k^p-\hat{r}_k^p)^2\underbrace{-\sum_{j=1}^{J}\psi_j\Bigg(\sum_{s\in\mathcal{S}_j}\sum_{k\in\mathcal{K}_s}\mathbb{E}_{d_k}\Bigg[\phi_{k,S_n}\Big(\min(\sum_{p\in\mathcal{P}_k}\hat{r}_k^p,d_k)\Big)\Bigg]\Bigg)}_{\Upsilon_2}\\
&+\sum_{j=1}^{J}\psi_j\Bigg(\sum_{s\in\mathcal{S}_j}\sum_{k\in\mathcal{K}_s}\Bigg(-\phi'_{k,S_n}\Big(\sum_{p\in\mathcal{P}_k}\hat{r}_k^p\Big)\Big(1-F_{k,S_n}(\sum_{p\in\mathcal{P}_k}\hat{r}_k^p)\Bigg)\Bigg(\sum_{p\in\mathcal{P}_k}(r_k^p-\hat{r}_k^p)\Bigg)\\
&+\frac{Q}{2}\sum_{s\in\mathcal{S}_j}\sum_{k\in\mathcal{K}_s}\sum_{p\in\mathcal{P}_k}(r_k^p-\hat{r}_k^p)^2\Bigg). \qquad (4.42)
\end{aligned}
$$

$\{\mathbf{g},\mathbf{t}\}$ as the second block. We optimize with respect to both blocks in the $\pi+1^{\text{th}}$ iteration and do $\boldsymbol{\tau}^{\pi+1}=\boldsymbol{\tau}^{\pi}+\rho(\mathbf{r}^{\pi+1}-\mathbf{g}^{\pi+1})$.

4.6.1 Subproblem with respect to r

The subproblem with respect to $\mathbf{r}$ is as follows:

$$
\begin{aligned}
\min_{\mathbf{r}\geq\mathbf{0}} \quad & \Gamma(\mathbf{r})+\sum_{j=1}^{J}\sum_{k\in\mathcal{K}_j}\sum_{p\in\mathcal{P}_k}\Big(\tau_k^{p,\pi}(r_k^p-g_k^{p,\pi})+\frac{\rho}{2}(r_k^p-g_k^{p,\pi})^2\Big)\\
\text{s.t.} \quad & (4.1). \qquad (4.41)
\end{aligned}
$$

To solve (4.41), we substitute the quadratic upper-bound [149, eq. (12)] for the expected excessive demand from slice s given in (4.21), which is convex with a Lipschitz continuous gradient. The Lipschitz constant for the gradient of (4.21) is $\sum_{k\in\mathcal{K}_s}|\mathcal{P}_k|$. Similarly, we substitute a quadratic upper-bound for (4.9). If Q is the Lipschitz constant for the gradient of (4.9) (Q can be calculated through differentiation of (4.9) and using extremum values of the first and second derivatives of $\phi_{k,S_n}(\cdot)$), then (4.42) is an upper-bound for $\Gamma(\mathbf{r})$ around the current iterate $\hat{\mathbf{r}}$. In (4.42), Υ_1 and Υ_2 are constants and we consider $\Upsilon = \Upsilon_1 + \Upsilon_2$. We notice that (4.42) can be written in the following shorter form:

$$\sum_{j=1}^{J} \sum_{s \in \mathcal{S}_j} (\frac{\psi_j Q}{2} + \frac{\sum_{k \in \mathcal{K}_s} |\mathcal{P}_k|}{2}) \sum_{k \in \mathcal{K}_s} \sum_{p \in \mathcal{P}_k} (r_k^p - \hat{r}_k^p)^2 \\ + \sum_{j=1}^{J} \sum_{s \in \mathcal{S}_j} \sum_{k \in \mathcal{K}_s} \sum_{p \in \mathcal{P}_k} h_k^p (r_k^p - \hat{r}_k^p) + \Upsilon, \tag{4.43}$$

where h_k^p is a constant coefficient. When (4.43) is substituted for $\Gamma(\mathbf{r})$ in (4.41), (4.41) becomes a QP with a separable objective function in $\{r_k^p\}_{p \in \mathcal{P}_k, k=1:K}$. With the quadratic upper-bound (4.43), we can solve (4.41) using CPLEX or Gurobi or in a distributed fashion using the approach given in [150, 151], which decomposes the problem across backhaul links. We associate one Lagrange multiplier μ_l to each constraint in (4.1) and find the Lagrangian as follows:

$$L_c(\mathbf{r}, \boldsymbol{\mu}) = \sum_{j=1}^{J} \sum_{s \in \mathcal{S}_j} (\frac{\psi_j Q}{2} + \frac{\sum_{k \in \mathcal{K}_s} |\mathcal{P}_k|}{2}) \sum_{k \in \mathcal{K}_s} \sum_{p \in \mathcal{P}_k} (r_k^p - \hat{r}_k^p)^2 \\ + \sum_{j=1}^{J} \sum_{s \in \mathcal{S}_j} \sum_{k \in \mathcal{K}_s} \sum_{p \in \mathcal{P}_k} h_k^p (r_k^p - \hat{r}_k^p) + \Upsilon + \sum_{l \in \mathcal{L}} \mu_l \left(\sum_{j=1}^{J} \sum_{k \in \mathcal{K}_j} \sum_{p: \{p \in \mathcal{P}_k, l \in \mathcal{L}_k^p\}} r_k^p - C_l \right).$$

We can decompose the Lagrangian across backhaul links and parallelelize the computation across links as $L_c(\mathbf{r}, \boldsymbol{\mu})$ is strongly convex in $\mathbf{r}$ and has a Lipschitz continuous gradient [151, Theorem 1]. Each term in the Lagrangian that includes r_k^p is decomposed across links that construct path p. We denote the Lagrange multiplier in the m^{th} iteration of the distributed approach by $\widetilde{\mu}_l^m$. In the m^{th} iteration, each link of path p receives a portion of

$$\alpha_{k,l}^{p,m} = \frac{\widetilde{\mu}_l^{m-1}}{\sum_{l' \in \mathcal{L}_k^p} \widetilde{\mu}_{l'}^{m-1}}, \tag{4.44}$$

from those Lagrangian terms that include r_k^p. The decomposed Lagrangian on a backhaul link is as follows:

$$
\begin{aligned}
&L_l(\mathbf{r}_l, \mu_l, \widetilde{\boldsymbol{\mu}}^{m-1}) = \sum_{j=1}^{J} \sum_{k\in\mathcal{K}_j} \sum_{p:\{p\in\mathcal{P}_k, l\in\mathcal{L}_k^p\}} \alpha_{k,l}^{p,m} \tau_k^{p,\pi}(r_k^p - g_k^{p,\pi}) \\
&+ \sum_{j=1}^{J} \sum_{k\in\mathcal{K}_j} \sum_{p:\{p\in\mathcal{P}_k, l\in\mathcal{L}_k^p\}} \frac{\rho}{2}\, \alpha_{k,l}^{p,m} (r_k^p - g_k^{p,\pi})^2 \\
&+ \sum_{j=1}^{J} \sum_{s\in\mathcal{S}_j} (\frac{\psi_j Q}{2} + \frac{\sum_{k\in\mathcal{K}_s} |\mathcal{P}_k|}{2}) \sum_{k\in\mathcal{K}_s} \sum_{p:\{p\in\mathcal{P}_k, l\in\mathcal{L}_k^p\}} \alpha_{k,l}^{p,m} (r_k^p - \hat{r}_k^p)^2 \\
&+ \sum_{j=1}^{J} \sum_{k\in\mathcal{K}_j} \sum_{p:\{p\in\mathcal{P}_k, l\in\mathcal{L}_k^p\}} \alpha_{k,l}^{p,m} h_k^p (r_k^p - \hat{r}_k^p) + \mu_l \left(\sum_{j=1}^{J} \sum_{k\in\mathcal{K}_j} \sum_{p:\{p\in\mathcal{P}_k, l\in\mathcal{L}_k^p\}} r_k^p - C_l \right),
\end{aligned}
$$

where $\alpha_{k,l}^{p,m}$ is a constant and $\mathbf{r}_l = \{r_k^p\}_{p\in\mathcal{P}_k, l\in\mathcal{L}_k^p, k=1:K}$. The optimal r_k^p and μ_l are obtained from the KKT conditions for the per-link subproblem as follows:

$$
\frac{\partial L_l(\mathbf{r}_l, \mu_l, \widetilde{\boldsymbol{\mu}}^{m-1})}{\partial r_k^p} = 0, r_k^p \geq 0, \mu_l \geq 0, \qquad p \in \mathcal{P}_k, l \in \mathcal{L}_k^p, \forall k, \tag{4.45a}
$$

$$
\mu_l \left(\sum_{k=1}^{K} \sum_{p:\{p\in\mathcal{P}_k, l\in\mathcal{L}_k^p\}} r_k^p - C_l \right) = 0, \tag{4.45b}
$$

$$
\sum_{k=1}^{K} \sum_{p:\{p\in\mathcal{P}_k, l\in\mathcal{L}_k^p\}} r_k^p \leq C_l, \tag{4.45c}
$$

The solution to (4.45a) can be obtained in closed-form for the l^{th} backhaul link as follows:

$$
r_k^p = \max\Big(0, \frac{\alpha_{k,l}^{p,m}\big((\psi_j Q + \sum_{k\in\mathcal{K}_s}|\mathcal{P}_k|)\hat{r}_k^p + \rho g_k^{p,\pi} - h_k^p - \tau_k^{p,\pi}\big) - \mu_l}{\alpha_{k,l}^{p,m}(\rho + \psi_j Q + \sum_{k\in\mathcal{K}_s}|\mathcal{P}_k|)}\Big). \tag{4.46}
$$

To find the optimal r_k^p and μ_l for each per-link subproblem, we implement a bisection search on μ_l in the positive orthant and obtain the corresponding r_k^p from (4.46) until one μ_l that satisfies the complementary slackness condition (4.45b) is obtained. If there is no such μ_l, then we set $\widetilde{\mu}_l^m = 0$ and $\alpha_{k,l}^{p,m+1} = 0$. For these links, we do not need to continue computation as the KKT conditions remain satisfied. In the following iterations, we ignore

Algorithm 13: The proposed approach to solve (4.41)

```
0. Initialization Assign some small positive number to each $\widetilde{\mu}_l^0$;
for a few iterations do
    1. Assign a positive number to each $\widetilde{\mu}_l^0$, $m = 0$;
    repeat
        for all links do
            2. Find $\alpha_{k,l}^{p,m+1} = \widetilde{\mu}_l^m / \sum_{l' \in \mathcal{L}_k^p} \widetilde{\mu}_{l'}^m$;
            if $\widetilde{\mu}_l^m > 0$ then
                3. Find $\widetilde{\mu}_l^{m+1}$ from (4.45a) and (4.45b);
            4. $m = m + 1$;
    until all $\widetilde{\mu}_l^m$ variables converge;
    for all $r_k^p$ variables do
        5. Find $\widetilde{r}_k^{p,m}$ from (4.46) for a per-link subproblem, where $l \in \mathcal{L}_k^p$ and
           $\widetilde{\mu}_l^m > 0$;
    6. $\hat{\mathbf{r}} = \widetilde{\mathbf{r}}^m$;
```

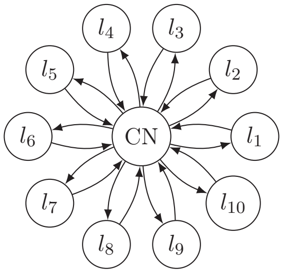


Figure 4.3: Message passing between 10 links and one CN.

these links and consider links with $\widetilde{\mu}_l^m > 0$. We find μ_l in parallel for all links. Once each μ_l is obtained, we set $\widetilde{\mu}_l^m = \mu_l, \widetilde{r}_k^{p,m} = r_k^p$ and update $\alpha_{k,l}^{p,m+1} = \widetilde{\mu}_l^m / \sum_{l' \in \mathcal{L}_k^p} \widetilde{\mu}_{l'}^m$. We alternate between finding $\widetilde{\mu}_l^m$ and $\alpha_{k,l}^{p,m+1}$ until each $\widetilde{\mu}_l^m$ converges. Upon convergence, we update $\hat{\mathbf{r}} = \widetilde{\mathbf{r}}^m$. We repeat this until $\hat{\mathbf{r}}$ converges. The summary of the proposed approach to solve (4.41) is given in Algorithm 13 and the message passing is depicted in Fig. 4.3. The convergence of Algorithm 13 is depicted in Fig. 4.4, where the user demand is log-normal as given in (4.51).

We do successive upper-bound minimization [19] and solve (4.41) for a few iterations with Algorithm 13 and update $\hat{\mathbf{r}}$ in each iteration. Then, we update $\mathbf{r}^{\pi+1} = \hat{\mathbf{r}}$. As the

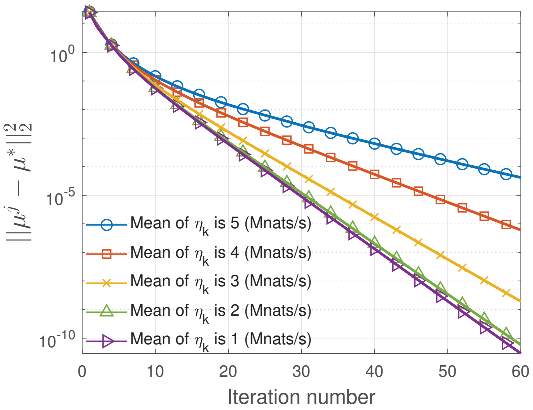


Figure 4.4: The convergence of Algorithm 13.

quadratic upper-bound (4.42) satisfies the four convergence conditions for block successive upper-bound minimization methods [19, Assumption 2], the successive upper-bound minimization converges to the KKT solution of (4.41).

Here, we discuss the practical implementation of the proposed slice reconfiguration method. In Algorithm 3, each per-link subproblem is solved in parallel and then the next decomposition is found accordingly using (44). This is equivalent to having a Central Node (CN) in the network that coordinates the problem decomposition across backhaul links. Each link independently solves its subproblem and sends the optimized dual variable back to the CN. Next, the CN computes $\alpha_{k,l}^{p,m}$ and sends that to each link. This message passing interface is depicted in Fig. 3.

4.6.2 Subproblem with respect to g and t

The ℓ_2-norms $\left\|\mathbf{g}_s - \underline{\mathbf{r}}_s^{n-1}\right\|_2$ and $\left\|\mathbf{t}_s - \underline{\mathbf{t}}_s^{n-1}\right\|_2$ are not smooth. Instead of these two, we consider the following smooth approximation with a Lipschitz continuous gradient:

$$U(\mathbf{g}_s, \mathbf{t}_s) = -2\delta c_r(a_s^{1,i} + a_s^{2,i}) + c_r a_s^{1,i}\sqrt{\sum_{l\in\mathcal{L}}(\sum_{k\in\mathcal{K}_s}\sum_{p:\{p\in\mathcal{P}_k, l\in\mathcal{L}_k^p\}} g_k^p - \underline{r}_s^{l,n-1})^2 + \delta^2}$$
$$+ c_r a_s^{2,i}\sqrt{\sum_{b\in\mathcal{B}}(\sum_{k\in\mathcal{K}_s}\sum_{p:\{p\in\mathcal{P}_k, b\in\mathcal{U}_k^p\}} t_k^p - \underline{t}_s^{b,n-1})^2 + \delta^2},$$

where δ is a small positive number. We note that the minima of $c_r(a_s^{1,i}\left\|\mathbf{g}_s-\underline{\mathbf{r}}_s^{n-1}\right\|_2+a_s^{2,i}\left\|\mathbf{t}_s-\underline{\mathbf{t}}_s^{n-1}\right\|_2)$ is identical to that of $U(\mathbf{g}_s,\mathbf{t}_s)$, which is $\{\mathbf{g}_s,\mathbf{t}_s\}=\{\underline{\mathbf{r}}_s^{n-1},\underline{\mathbf{t}}_s^{n-1}\}$. The subproblem with respect to $\{\mathbf{g}_s\}_{\mathcal{S}_j,j=1:J}$ and $\{\mathbf{t}_s\}_{\mathcal{S}_j,j=1:J}$ becomes:

$$\min_{\{\mathbf{g}_s,\mathbf{t}_s\}_{\mathcal{S}_j,j=1:J}\geq\mathbf{0}} \quad \sum_{j=1}^{J}\sum_{s\in\mathcal{S}_j}\left(U(\mathbf{g}_s,\mathbf{t}_s)+(\mathbf{r}_s^{\pi+1}-\mathbf{g}_s)'\boldsymbol{\tau}_s^{\pi}+\frac{\rho}{2}\left\|\mathbf{r}_s^{\pi+1}-\mathbf{g}_s\right\|_2^2\right) \tag{4.47a}$$

$$\text{s.t.} \quad \int_0^{g_k^p} z_{k,S_n}^p(v_k^p,t_k^p)\,(g_k^p-v_k^p)dv_k^p\leq\beta_s g_k^p, p\in\mathcal{P}_k, k\in\mathcal{K}_s, \tag{4.47b}$$

$$(4.2),(4.3),(4.4). \tag{4.47c}$$

In the above problem, $\mathbf{g}_s=\{g_s^l\}_{l\in\mathcal{L}}$ and $\mathbf{t}_s=\{t_s^b\}_{b\in\mathcal{B}}$. We can rewrite (4.47a) as a function of $\{g_k^p,t_k^p\}$ if we substitute $\{g_k^p\}_{p\in\mathcal{P}_k,k\in\mathcal{K}_s}$ for $\mathbf{g}_s$ and substitute $\{t_k^p\}_{p\in\mathcal{P}_k,k\in\mathcal{K}_s}$ for $\mathbf{t}_s$. The above problem is not separable in $\mathbf{g}_s$ and $\mathbf{t}_s$ due to (4.47b). To alleviate the complexity of solving the above problem, we decouple (4.47b) from (4.47) by copying optimization variables and using ADMM. We add the constraints $\mathbf{o}=\mathbf{g}$ and $\mathbf{f}=\mathbf{t}$. We dualize these two constraints and find the augmented Lagrangian. In the $m+1^{\text{th}}$ iteration of the inner ADMM, the subproblem with respect to $\{\mathbf{g}_s\}_{\mathcal{S}_j,j=1:J}$ and $\{\mathbf{t}_s\}_{\mathcal{S}_j,j=1:J}$ becomes:

$$\begin{aligned}\min_{\{\mathbf{g}_s,\mathbf{t}_s\}_{\mathcal{S}_j,j=1:J}} \quad & \sum_{j=1}^{J}\sum_{s\in\mathcal{S}_j}\Bigg((\mathbf{r}_s^{\pi+1}-\mathbf{g}_s)'\boldsymbol{\tau}_s^{\pi}+\frac{\rho}{2}\left\|\mathbf{r}_s^{\pi+1}-\mathbf{g}_s\right\|_2^2\\ & +(\mathbf{o}_s^m-\mathbf{g}_s)'\boldsymbol{\iota}_s^m+\frac{\rho_{11}}{2}\left\|\mathbf{o}_s^m-\mathbf{g}_s\right\|_2^2\\ & +(\mathbf{f}_s^m-\mathbf{t}_s)'\boldsymbol{\lambda}_s^m+\frac{\rho_{12}}{2}\left\|\mathbf{f}_s^m-\mathbf{t}_s\right\|_2^2+U(\mathbf{g}_s,\mathbf{t}_s)\Bigg)\\ \text{s.t.} \quad & (4.2),(4.3),(4.4),\mathbf{g}_s,\mathbf{t}_s\geq\mathbf{0}.\end{aligned} \tag{4.48}$$

Problem (4.48) is separable and strongly convex in $\{\mathbf{g}_s\}_{\mathcal{S}_j,j=1:J}$ and $\{\mathbf{t}_s\}_{\mathcal{S}_j,j=1:J}$. To separately optimize with respect to $\{\mathbf{g}_s\}_{\mathcal{S}_j,j=1:J}$ and $\{\mathbf{t}_s\}_{\mathcal{S}_j,j=1:J}$, we can use the proximal gradient descent algorithm or existing solvers. Then, we decompose (4.48) across slices and update $\{\mathbf{g}_s^{m+1},\mathbf{t}_s^{m+1}\}$. Therefore, the subproblem with respect to $\mathbf{o}_s$ and $\mathbf{f}_s$ is

$$\min_{\mathbf{o}_s,\mathbf{f}_s\geq\mathbf{0}} \quad (\mathbf{f}_s-\mathbf{t}_s^{m+1})'\boldsymbol{\lambda}_s^m+\frac{\rho_{12}}{2}\left\|\mathbf{f}_s-\mathbf{t}_s^{m+1}\right\|_2^2 \tag{4.49}$$

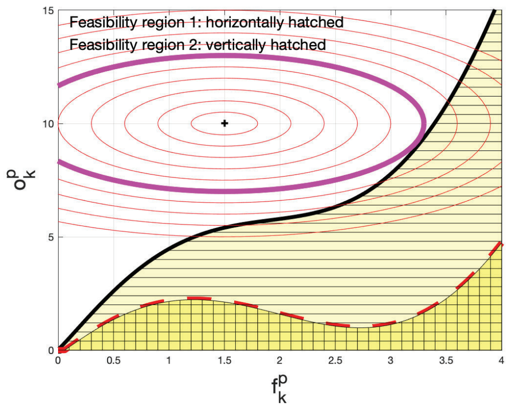


Figure 4.5: The feasibility region and level sets.

$$+ (\mathbf{o}_s - \mathbf{g}_s^{m+1})'\boldsymbol{\iota}_s^m + \frac{\rho_{11}}{2}\left\|\mathbf{o}_s - \mathbf{g}_s^{m+1}\right\|_2^2 \qquad \text{s.t.} \qquad (4.47\text{b}).$$

We observe that (4.49) is decomposable across paths, and we find a separate subproblem for each (o_k^p, f_k^p) pair. For each o_k^p, we can find the lowest f_k^p that ensures (4.47b). Then, a feasible set for optimization variables is characterized (e.g., feasibility region 1 in Fig. 4.5). Since the objective function of (4.49) is strongly convex, one can find the global solutions with respect to $\mathbf{o}_s$ and $\mathbf{f}_s$ by setting the gradient of the objective function to zero and projecting the obtained solutions to the positive orthant. If the obtained solutions satisfy (4.47b), then problem (4.49) is solved. If the minimizers of the objective function of (4.49) do not satisfy (4.47b), then we solve the problem using a bisection search. The bisection search works on level sets of the objective function of (4.49) (e.g., Fig. 4.5). For each level set considered by the bisection, we determine whether or not the considered level set intersects with the characterized feasibility set. This can be done by a simple search as follows. We consider the rightmost point and the bottommost point of the level set. With a sufficiently small constant step-size, we move to the right on the level set from the bottommost point. In each point, we measure the vertical distance between the level set and the border of the feasibility region. If the distance becomes zero in one point, then the level set intersects with the feasibility region. If the vertical distance variation is not monotone, there is a chance of intersection. Using the bisection search on the level sets, we find the level set that is tangent to the feasibility region. The obtained point of the

Algorithm 14: The proposed approach to solve (4.49)

```
0. Initialization $A_1 = 0$, $A_2 =$ a large number
while $|A_1 - A_2|$ is not small enough do
    1. $A_3 = (A_1 + A_2)/2$;
    2. Find the level sets of the objective function of (4.49) that correspond to
       $A_1, A_2$ and $A_3$;
    if the level set corresponding to $A_3$ intersects with the feasibility region then
        3. $A_2 \leftarrow A_3$;
    else
        4. $A_1 \leftarrow A_3$;
5. Return the feasible point touched by the tangent level set;
```

Algorithm 15: The ADMM approach to solve (4.47)

```
0. Initialization $\boldsymbol{\iota}^0 = \mathbf{0}$, $\boldsymbol{\lambda}^0 = \mathbf{0}$, $m = 0$;
repeat
    1. Use proximal gradient descent to solve (4.48);
    2. Use Algorithm 14 to solve (4.49);
    3. $\boldsymbol{\iota}^{m+1} = \boldsymbol{\iota}^m + \rho_{11}(\mathbf{o}^{m+1} - \mathbf{g}^{m+1})$ and $\boldsymbol{\lambda}^{m+1} = \boldsymbol{\lambda}^m + \rho_{12}(\mathbf{f}^{m+1} - \mathbf{t}^{m+1})$;
    4. $m = m + 1$;
until the primal and dual residuals are small enough;
```

feasibility region, which is touched by the tangent level set, is the minimizer of (4.49). The summary of the proposed approach to solve (4.49) is given in Algorithm 14.

Once we obtain the optimal $\{\mathbf{o}_s, \mathbf{f}_s\}$, we update $\{\mathbf{o}_s^{m+1}, \mathbf{f}_s^{m+1}\}$. We alternatively minimize with respect to $\{\mathbf{g}_s, \mathbf{t}_s\}$ and $\{\mathbf{o}_s, \mathbf{f}_s\}$. Then, we update Lagrange multipliers as $\boldsymbol{\iota}^{m+1} = \boldsymbol{\iota}^m + \rho_{11}(\mathbf{o}^{m+1} - \mathbf{g}^{m+1})$ and $\boldsymbol{\lambda}^{m+1} = \boldsymbol{\lambda}^m + \rho_{12}(\mathbf{f}^{m+1} - \mathbf{t}^{m+1})$. The summary of the proposed approach to solve (4.47) is given in Algorithm 15. After Algorithm 15 converges (the primal and dual residuals are small enough, see [152, p. 15–18]), we use obtained $\mathbf{g}$ to update $\mathbf{g}^{\pi+1}$. If the ADMM subproblems are solved distributedly, a message passing between slices and a CN is required.

Proposition 5. *Algorithm 14 converges to the global stationary solution of the problem in* (4.49) *when the expected outage is non-increasing in the transmission resource.*

Proof. Consider that we denote the expected outage in (8) by $\varphi_1(r, t)$. Based on the assumption in Proposition 5, $\varphi_1(r, t)$ is non-increasing in t. We claim that the feasibility region characterized by (8) cannot narrow when t increases (e.g., feasibility region 2 in Fig. 4.5). We prove this claim by contradiction. Consider $t_1 < t_2$ and the feasibility region narrows as t increases, where (r, t_1) belongs to the feasibility region while (r, t_2) does not belong to the feasibility region. Then, we have $\varphi_1(r, t_1) \leq \beta_s r$ and $\varphi_1(r, t_2) > \beta_s r$. Based on this, we should have $\varphi_1(r, t_1) < \varphi_1(r, t_2)$. This is a contradiction as we assumed $\varphi_1(r, t)$ is non-increasing in the transmission resource. Therefore, the feasibility region cannot narrow as t increases similar to feasibility region 1 in Fig. 4.5. Consider that the border line that specifies the feasibility region is characterized by the following equation $\varphi_2(r) = t$, where $\varphi_2(r)$ is non-decreasing. Then, the feasibility region of (8) can be represented as $\varphi_2(r) \leq t$. We consider the following optimization problem:

$$\begin{aligned} \min_{r,t \geq 0} \quad & \varphi_3(r) + \varphi_4(t) \\ \text{s.t.} \quad & \varphi_2(r) \leq t, \end{aligned}$$

where $\varphi_3(r)$ an $\varphi_4(t)$ are strongly convex functions with global minimas $r^\star$ and $t^\star$, respectively. The above problem is not always convex as $\varphi_2(r)$ is not always convex.

We can find the level sets of the objective function from $\varphi_3(r) + \varphi_4(t) = \kappa$, where we change κ. The solution of the above problem is a point in the feasible set that gives the least objective function value. If the minimizer of the objective function lies in the feasible set, then that is the problem solution. Otherwise, the problem solution lies on the level set that is tangent to the feasible set and is exactly the point on the feasible set border line that is touched by the level set. As the gradient of each point on a level set is different from the gradient of the other points, only one point of the feasible set is touched by the tangent level set. Algorithm 4 can find the tangent level set using a bisection search since there is only one tangent level set, and the rest of level sets are either intersecting with the feasibility region or do not intersect. In each iteration, we quickly evaluate whether or not three considered level sets by the bisection method are intersecting the feasible set or not. Since the objective function of (4.49) is strongly convex, the obtained solution is unique and Algorithm 4 solves the problem globally. □

Algorithm 16: The ADMM algorithm to solve (4.24)

```
0. **Initialization** $\boldsymbol{\tau}^0 = \mathbf{0}$, $\pi = 0$, assign values to $a_s^{1,0}$ and $a_s^{2,0}$, $i = 1$;
repeat
    1. Find $a_s^{1,i}$ and $a_s^{2,i}$ from (4.38);
    repeat
        2. Apply Algorithm 13 to obtain $\mathbf{r}^{\pi+1}$;
        3. Apply Algorithm 15 to obtain $\mathbf{g}^{\pi+1}$;
        4. $\boldsymbol{\tau}^{\pi+1} = \boldsymbol{\tau}^{\pi} + \rho(\mathbf{r}^{\pi+1} - \mathbf{g}^{\pi+1})$;
        5. $\pi = \pi + 1$;
    until the primal and dual residuals are small enough and $\hat{\mathbf{r}}$ converges;
    6. $i = i + 1$;
until both $a_s^{1,i}$ and $a_s^{2,i}$ converge;
7. $\underline{\mathbf{r}}_s^n = \mathbf{r}_s^{\pi}$ and $\underline{\mathbf{t}}_s^n = \mathbf{t}_s^{\pi}$
```

Remark 8. *If* (4.25) *is considered instead of* (4.8)*, then we rewrite* (4.47b) *as*

$$\int_0^{g_k^w} z_{k,S_n}^p(v_k^w, t_k^w)\,(g_k^w - v_k^w)dv_k^w \leq \beta_s g_k^w, k \in \mathcal{K}_s, \tag{4.50}$$

where $g_k^w = \sum_{p:\{p\in\mathcal{P}_k, w\in p\}} g_k^p$*. To solve* (4.49) *with* (4.50)*, we dualize* $g_k^w = \sum_{p:\{p\in\mathcal{P}_k, w\in p\}} g_k^p$ *and find the augmented Lagrangian. We can deploy an ADMM algorithm to solve the problem. We use Algorithm 14 to solve the subproblem with respect to* $\{\{g_k^w\}_{w\in\mathcal{W}_k, k=1:K}, \{t_k^w\}_{w\in\mathcal{W}_k, k=1:K}\}$*. Moreover, the augmented Lagrangian minimizers with respect to* $\{g_k^p\}_{p\in\mathcal{P}_k, k=1:K}$ *can be obtained in closed-form.*

Proposition 6. *Algorithm 15 converges to the global stationary solution to the problem in* (4.47)*.*

Proof. The first subproblem of Algorithm 5, given in (48), is strongly convex and has a unique global optimal solution. Furthermore, we proved in Proposition 5 that the unique global solution is obtained by Algorithm 4 for (49). Therefore, the obtained solution by Algorithm 5 is global stationary [111, p. 698]. □

4.6.3 The proposed ADMM Algorithm

In the proposed ADMM algorithm, we alternatively optimize with respect to **r** as one block and jointly with respect to $\{\mathbf{g}, \mathbf{t}\}$ as the other block. In the proposed ADMM algorithm,

$\mathbf{r}$ is updated through a block successive upper-bound minimization in Section 4.6.1, while $\{\mathbf{g}, \mathbf{t}\}$ is updated through an inner ADMM in Section 4.6.2. In the $\pi + 1^{\text{th}}$ iteration of the ADMM algorithm, we update the Lagrange multipliers as $\boldsymbol{\tau}^{\pi+1} = \boldsymbol{\tau}^{\pi} + \rho(\mathbf{r}^{\pi+1} - \mathbf{g}^{\pi+1})$. We continue alternating between optimization with respect to $\mathbf{r}$ and $\{\mathbf{g}, \mathbf{t}\}$ until both variable blocks and $\hat{\mathbf{r}}$ converge. We solve each subproblem of the group LASSO regularization using the proposed ADMM algorithm. We then update $a_s^{1,i+1}$ and $a_s^{2,i+1}$ using (4.38) and solve again. We continue until both $a_s^{1,i}$ and $a_s^{2,i}$ converge. The summary of the proposed approach to solve (4.24) is given in Algorithm 16.

Proposition 7. *The ADMM approach given in Algorithm 16 converges to the global solution of* (4.39).

Proof. As the quadratic continuous upper-bound (42) has identical value and gradient to the value and gradient of $\Gamma(\mathbf{r})$ in point $\mathbf{r} = \hat{\mathbf{r}}$, the convergence conditions [19, Assumption 2] are satisfied, and the successive upper-bound (42) minimizations converge to a stationary solution with respect to $\mathbf{r}$, which is also global due to the strong convexity of (41). Moreover, due to Proposition 6, the global solution is obtained by Algorithm 5 for the subproblem with respect to $\{\mathbf{g}, \mathbf{t}\}$. Therefore, the obtained solution by the ADMM approach in Algorithm 6 is global stationary [111, p. 698]. □

We discuss the iteration complexity of different proposed algorithms. The Frank-Wolfe algorithm proposed to solve each subproblem of the upper-bound minimization for slice activation has a sublinear convergence rate, and the residual error shrinks with a rate of $\mathcal{O}(1/m)$ [153], where m denotes the iteration number. The minimization with respect to $\mathbf{t}$ and $\mathbf{x}$ is non-convex and is solved by a successive upper-bound minimization method. The convergence of successive upper-bound minimization methods to KKT points of non-convex problems is analyzed in [113], where an error reduction with a rate of $\mathcal{O}(1/m)$ is guaranteed [113].

Now, we discuss the convergence of the proposed method to solve the slice reconfiguration problem (24). The convergence of flow variables in each per-link subproblem depends on the convergence of the dual variable μ_l. The optimal μ_l is obtained through a bisection search. The bisection search has a linear convergence rate, and the residual error shrinks with a rate of $\mathcal{O}(1/2^m)$, where m represents the bisection iteration number. In addition, numerical simulations (given in Fig. 4) confirm the linear convergence of Algorithm 3 that ensures a fast error reduction. The multi-path routing method to solve (41) successively minimizes an upper-bound. The convergence rate of successive upper-bound minimization methods to globally minimize convex functions is sublinear and equal to $\mathcal{O}(1/m)$ [112]. The minimization (48) with respect to $\mathbf{g}$ and $\mathbf{t}$ is separable and strongly convex. Therefore, the proximal gradient descent algorithm attains a linear convergence rate [153]. The minimization (48) with respect to $\mathbf{o}$ and $\mathbf{f}$ is solved to global optimality via a bisection search method and the solution is obtained at a linear convergence rate of $\mathcal{O}(1/2^m)$. Based on the above discussion, one of the subproblems of (ADMM) Algorithm 6 is solved at a sublinear rate and the other is solved at a linear rate. Therefore, the overall convergence rate of Algorithm 6 is sublinear.

4.7 Simulation Results

In this section, we evaluate the performance of the proposed approaches. The considered network for evaluations is the one given in the previous chapter, which includes both the backhaul and radio access components. A data center is connected to routers of the network through three gateway routers, GW 1, GW 2, and GW 3. The network includes 57 RUs and 11 network routers. RUs are distributed on the X-Y plane, and they are connected to each other and to routers via wired links. Data rates are measured in Mnats/s or Gnats/s. The backhaul network has 162 links. Wired link capacities are identical in both directions. Backhaul link capacities are determined as

- Links between the data center and routers: 4 Gnats/s;
- Links between routers: 2 Gnats/s;
- Links between routers and RUs: 2 Gnats/s;
- 2-hop to the routers: 400 Mnats/s;
- 3-hop to the routers: 320 Mnats/s;
- 4-hop to the routers: 160 Mnats/s.

In one time instance, we consider users are uniformly distributed in the network area. The considered paths originate from the data center and are extended toward users. User-RU associations are determined by the highest long-term received power. We consider three wireless connections that have the highest received power to serve each user. There are three paths for carrying each user's data from a data center to RUs. The distribution of the demand is log-normal [154, 155]:

$$d_k \sim \frac{1}{d_k \sigma_k \sqrt{2\pi}} \exp(-\frac{(\ln d_k - \eta_k)^2}{2\sigma_k^2}). \tag{4.51}$$

In addition, it is assumed that η_k is realized randomly from a normal distribution for each user. The power allocations in RUs are fixed. The dispensed resource in an RU is bandwidth. The channel between each user and an RU is a Rayleigh fading channel. The CDF of the wireless channel capacity, which is parameterized by the allocated bandwidth t_k^p, is given as [100]

$$1 - \exp(\frac{1 - 2^{r_k^p / t_k^p}}{\overline{\mathrm{SNR}_k^p}}),$$

where $\overline{\mathrm{SNR}_k^p}$ is the average SNR in one timescale. The PDF of the wireless channel capacity is

$$\frac{\ln(2) 2^{r_k^p / t_k^p} \exp(\frac{1 - 2^{r_k^p / t_k^p}}{\overline{\mathrm{SNR}_k^p}})}{\overline{\mathrm{SNR}_k^p} t_k^p}.$$

4.7.1 Slice Activation

Suppose that 200 users are served by five different slices of a single tenant. We consider $\psi_j = 1$ and $\theta_s = 3$, and we increase the RU bandwidth budget from 20 MHz to 80 MHz with a step size of 15 MHz. The minimum required reserved rate for the slices is 600 Mnats/s. In the considered setup, $\phi_{k,L_n}(r_k) = 90 - \exp(-0.045 r_k + 4.5)$, where r_k is in Mnats/s. The cost of activating a slice is 1000 units of utility, and $q = 0.1$ and $\epsilon = 0.05$. We consider four possible $\mathcal{K}_s$ for each slice. Furthermore, we consider two scenarios. We list the considered

RU Budget	Scenario 1				
	Optimal solution		Algorithm 12		
	Activated slices	Obj. fun.	Activated slices	Obj. fun.	App. ratio
20 MHz	1, 4	1,446	1, 4	1,446	1
35 MHz	1, 4	2,055	1, 4	2,055	1
50 MHz	1, 4, 5	2,358	1, 4	2,331	0.988
65 MHz	1, 4, 5	2,625	1, 4, 5	2,625	1
80 MHz	1, 2, 4, 5	2,863	1, 4, 5	2,813	0.982
RU Budget	Scenario 2				
20 MHz	1, 4	1,655	1, 4	1,655	1
35 MHz	1, 4	2,401	1, 4	2,401	1
50 MHz	1, 4, 5	3,066	1, 3, 4	3,000	0.978
65 MHz	1, 2, 4	3,436	1, 3, 4, 5	3,386	0.985
80 MHz	1, 2, 3, 4	3,852	1, 2, 4, 5	3,852	0.999

Table 4.2: The performance of Algorithm 12 against the optimal solution.

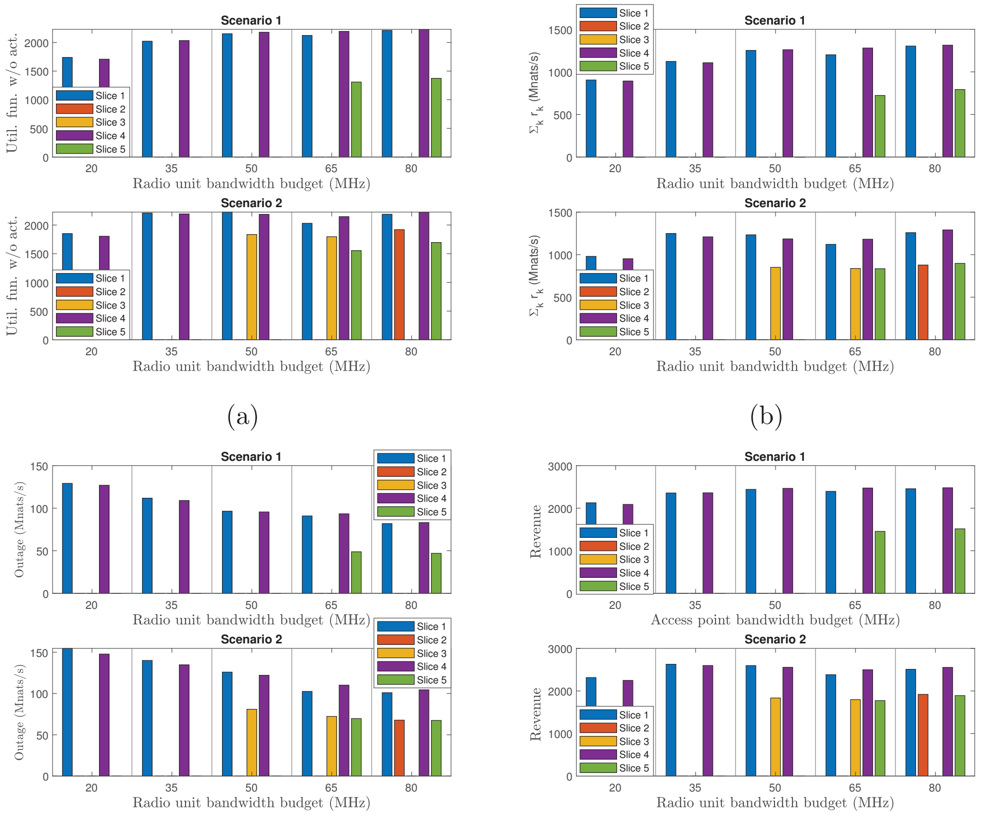
scenarios as follows:

1. Scenario 1: $\mathbb{E}[\eta_k]$ for users of slices 1 and 4 is 3.5 Mnats/s, and $\mathbb{E}[\eta_k]$ for users of slices 2, 3, and 5 is 2.5 Mnats/s;

2. Scenario 2: $\mathbb{E}[\eta_k]$ for users of slices 1 and 4 is 4 Mnats/s, and $\mathbb{E}[\eta_k]$ for users of slices 2, 3, and 5 is 3 Mnats/s.

In Fig. 4.6a, we plot the gained utility by each slice given in (4.13) without the cost of activation. When the depicted gained utility from a slice is zero in Fig. 4.6a, that slice is not activated. It is observed that in both scenarios, the number of activated slices increases as the bandwidth budget of RUs increases. The reason is that when the available bandwidth in RUs increases, the expected outage of downlinks (Fig. 4.6c) decreases, and therefore, the overall expected gained utilities of slices increase. Furthermore, we observe from Fig. 4.6a that as users served by slices 1 and 4 request greater data rates, these slices, which provide higher revenue, are activated first.

From Fig. 4.6b, we observe that the aggregate reserved rates for users of each slice



(a)

(b)

(c)

(d)

Figure 4.6: (a) The utility function without activation costs; (b) aggregate reserved rate for users of slices; (c) the expected outage of downlinks; and (d) the expected acquired revenue for slices by Algorithm 2.

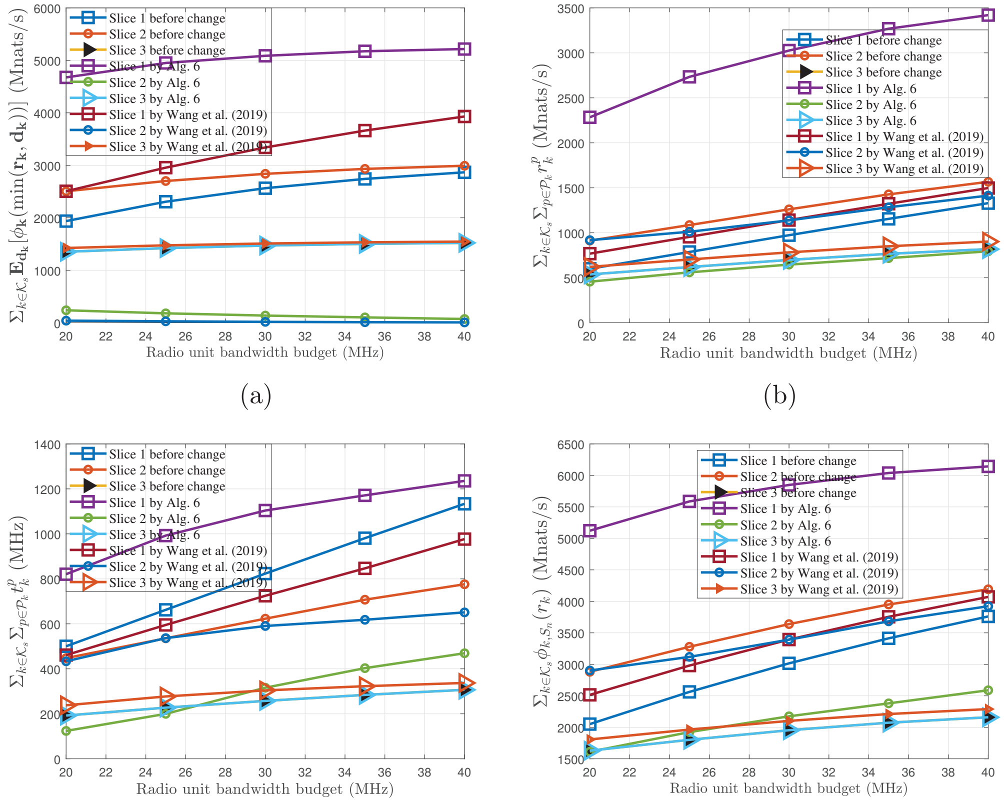
increases as the bandwidth budget of RUs increases if the number of active slices does not increase. When the number of active slices increases, the amount of allocated resources to users of a slice becomes smaller and reserved rates for users of a slice can decrease. We observe from Fig. 4.6c that the expected outage of users decreases as the bandwidth budget of RUs increases. This helps the slices to improve their utility functions. The collected revenue of each slice from serving users is depicted in Fig. 4.6d. We compare the solutions returned by Algorithm 12 against the optimal ones, obtained by an exhaustive search, in Table 4.2. According to Table 4.2, the obtained objective functions (including the activation costs) from Algorithm 12 are identical to their optimal values in six out of ten different experiments. The lowest approximation ratio of a solution obtained by Algorithm 12 is 0.978, which clearly demonstrates the efficiency of Algorithm 12.

In our experiments, the number of required iterations for the convergence of Frank-Wolfe algorithm is at most 60. Furthermore, Algorithm 12 converges after at most 30 iterations. The CPU time for each test (which corresponds to one scenario and an RU budget) in our C implementation is less than 14 seconds.

4.7.2 Slice Reconfiguration

We consider that 200 users are served by three different tenants, where each owns a single slice. The first and second tenants serve 80 users each. The third tenant serves 40 users. We consider $\psi_j = 1$ and increase the RU bandwidth budget from 20 MHz to 40 MHz with a step size of 5 MHz.The minimum required reserved rates for the slices are 500 Mnats/s, 500 Mnats/s and 400 Mnats/s, respectively. In the considered setup, $\boldsymbol{\beta} = [0.1, 0.2, 0.3]$ and $\phi_{k,S_n}(r_k) = 90 - \exp(-0.09r_k + 4.5)$, where r_k is in Mnats/s. We consider $\mathbb{E}[\eta_k] = 2.5$ Mnats/s for all users. Before the arrival of new statistics, we observe from Fig. 4.7a that the obtained revenue by slice 2 is greater than that by slice 1. The reason is that $\beta_2 > \beta_1$. Thus, providing a certain rate to a user in slice 2 requires less bandwidth. As a result, more rates are reserved for slice 2, as it is depicted in Fig. 4.7b, and less bandwidth reserved for slice 2, as depicted in Fig. 4.7c. We observe that the expected revenue for each slice enhances with the increase of bandwidth budget. Consider that the traffic statistics of each user served by the first and the second slice change dramatically as follows:

- For the users served by the first slice, the mean of η_k becomes 3.5 Mnats/s.



(a)

(b)

(c)

(d)

Figure 4.7: (a) Expected gained revenue by slices; (b) reserved rates for slices; (c) reserved bandwidth; and (d) the maximum revenue that can be obtained for slices by Algorithm 16.

- The distribution of the demand for each user served by slice 2 becomes exponential with a mean of 2 Mnats/s.

The PDF of the demand for each user supported by the third slice remains identical. Furthermore, $\boldsymbol{\beta}$ changes to $\boldsymbol{\beta} = [0.1, 0.6, 0.3]$. As η_k increases, the expected aggregate demanded traffic of users served by the first slice increases compared to its value in the first place. Furthermore, the expected aggregate demanded traffic of users served by the second slice diminishes compared to its previous value. Consider that the minimum reserved rate constraints of the first and second slices are relaxed after the change. However, the minimum reserved rate constraint of the third slice remains 400 Mnats/s. It is observed from Fig. 4.7a that the expected obtained revenue by the third slice does not change due to the enforced sparsity by the ℓ_0-norm. Numerical simulation confirms that after three updates, coefficients $a_s^{1,i}$ and $a_s^{2,i}$ converge. To solve each group LASSO subproblem, at most 15 iterations of the outer ADMM in Algorithm 16 with 7 iterations of the inner ADMM in Algorithm 15 are required. The overall CPU time is less than 27 seconds in our C implementation.

We compare Algorithm 16 against the one proposed in [122]. In [122], ℓ_1-norm is used to promote sparsity in slice reconfigurations. To tackle ℓ_1-norm non-differentiability, Wang *et al.* introduced affine constraints that allow limited slice variations instead of keeping ℓ_1-norm in the objective function. We allow 1 Mnats/s variation for the reserved rate for each path and 1 MHz variation for the reserved bandwidth for each path. Under the same bandwidth budgets in RUs, we observe from Figs. 4.7a, 4.7b and 4.7c that our approach is able to better reconfigure slices to achieve a higher expected revenue. Moreover, it reserves higher rates and more resources for users. In Fig. 4.7d, we depict the maximum revenue that can be obtained by slices. This takes place when the demanded rate by each user is at least equal to the reserved rate for that user. We observe that the maximum obtained revenue by slices using the reserved resources in the backhaul and RAN via Algorithm 16 is greater than that by [122]. Since Algorithm 16 does not introduce new constraints to limit slice variations, our algorithm is able to go beyond [122] in reconfiguring slices and allocating more resources if needed.

4.8 Concluding Remarks

In this chapter, we studied jointly slicing link capacity in the backhaul and transmission resources in RAN for multiple network tenants prior to the observation of user demands. We proposed a novel two-timescale framework for the activation of network slices and also reconfiguring active slices based on the time varying statistics from user demands and channel states. We proposed ℓ_q-norm, $0 < q < 1$, regularization to promote sparsity in the activation of network slices. Due to the non-convexity of the formulated activation problem, we successively solved a sequence of convex approximations of the problem via a novel Frank-Wolfe algorithm. Furthermore, we formulated the slice reconfiguration problem and since the reconfiguration of network slices can be costly, we used group LASSO regularization to enhance the sparsity of reconfigurations for active slices. An efficient, distributed and parallel algorithm is proposed to solve each group LASSO subproblem. Through extensive numerical tests, we verified the efficacy and efficiency of our approaches against the optimal solutions and existing state-of-the-art method.

Chapter 5

Guided Joint Image and field Map Estimation for MRI with Magnetic Field Inhomogeneity

The magnets used in MRI are traditionally designed to produce a highly uniform static magnetic field (B_0), which nominally varies in magnitude by less than ~ 5 ppm over the object to be imaged. With a goal of implementing MRI with more compact and portable high-field MRI systems in the future, recent research has focused on ways to image with B_0 inhomogeneity > 100 ppm. An overview of different techniques to perform MR imaging in the presence of large B_0 inhomogeneity is given in [15], e.g., using Single Point Imaging (SPI), echo shifting, or other techniques. Although several different multi-shot (slow) pulse sequences are available that tolerate large B_0 inhomogeneity, rapid single-shot methods like spiral and Echo-Planar Imaging (EPI) acquisitions remain highly intolerant to large B_0 inhomogeneity. The long readout time of these methods leads to image artifacts when the magnitude of B_0 varies spatially [156, 157, 158]. B_0 inhomogeneity causes spatially distributed off-resonance frequencies, which result in images with blur, geometric distortion, and signal dropout. Yet, these methods are critically important for performing a myriad of MRI measurements, such as diffusion-based tractography and functional neuroimaging, due to their abilities to acquire data with full Nyquist sampling in a single or few shots. Thus, effective approaches to overcome their intolerance to large B_0 inhomogeneity are needed.

When spatial information is frequency encoded, as done in spiral and EPI, frequency offsets arising from B_0 inhomogeneity cause phase errors in the readout direction in the k-space domain. When the off-resonance frequency increases, the blurring disperses information more broadly with respect to the number of voxels. Classical approaches to correct artifacts require prior information about the distribution of the off-resonance frequencies [159, 17, 18], called the field map. The field map information requirement makes the above methods sensitive to field map estimation errors.

5.1 Prior Work

Recent methods concentrate on the joint estimation of the image and the field map; joint image and field map estimation is studied in [1, 160, 161], where two or more MR images are acquired at different echo times T_E. The authors of [1] and [160] propose regularized gradient methods to optimize trigonometric cost functions, aiming to estimate the phase error present in each voxel. The method in [161] processes images with different resolutions from one object and utilizes a golden-section search to directly locate possible field map values. The formulation for the multi-echo acquisition in [1, 160, 161] is separable across voxels, which significantly simplifies the estimation of the field map. Joint estimation of image and field map in single-shot MRI from raw k-space data is studied in [162] and [163]. BCD methods are proposed in [162] and [163] to approximately solve the formulated non-linear regression problems.

Unlike the classical iterative methods that depend on the accuracy of the field map information, such as [17] and [159], more recent methods train Convolutional Neural Networks (CNNs) to deblur MR images. CNNs utilize prior information about characteristics of phase errors in the training data, and therefore they do not require accurate information about the field map. A residual CNN is proposed in [164] that corrects off-resonance artifacts. Short-readout scans are used in [164] as ground-truth images and augmented with additional off-resonance information for supervised learning. The authors of [164] propose a CNN with multiple convolutional layers that is based on the prominent residual neural network in [165]. A 3-layer residual CNN is proposed in [16] that corrects image domain off-resonance artifacts at the articulator vocal tract in spiral real time MRI of human speech production using a single coil without field map information. The authors of [16] approximate a linear transformation that maps distorted images to reference images. This linear transformation is decomposed across three convolutional layers. Trained CNNs in [164] and [16] can correct distorted images quickly, although their correction abilities are limited to relatively B_0 small inhomogeneities.

5.2 Contributions

Large magnetic field inhomogeneities are difficult for the aforementioned methods to handle. In this chapter, we address this scenario, and study the joint estimation of images and the field map using two datasets: 1) a set of images distorted due to large variations in B_0; and 2) a set of noisy, undistorted, fully sampled measurements from one or multiple Radio Frequency (RF) coil(s) for the same set of objects. For each 2D image in the first dataset, there are signal measurements for the same slice in the second dataset. All 2D images in the first dataset are taken to be at the same physical location within the magnet, and are affected by the same distribution of off-resonance frequencies. Distortions of images in the first dataset are generated after naive reconstructions from signal measurements without any treatment for phase errors. We formulate a joint optimization to estimate images and a linear transformation that maps distorted images to the ground-truth images. As the number of variables in our proposed optimization problem is large, we decompose the problem down to the voxel level and propose a distributed, parallel BCD algorithm to solve the optimization.

The approximated linear transformation is used to estimate off-resonance frequencies present in each voxel. To that end, we propose a golden-section search nested inside a grid-search to perform the estimation. The accuracy of the estimated off-resonance frequency in each voxel depends on the acceleration of k-space data acquisition for the first dataset images and the noise level in the second dataset. When the noise level is high or the acceleration factor is large, off-resonance frequencies in a number of voxels, especially voxels with small intensities, cannot be estimated. However, using the approximated off-resonance frequencies in a subset of voxels, we formulate a linear regression to estimate the coefficients of harmonics that provide a spherical harmonic representation for the overall B_0 inhomogeneity. Using the estimated coefficients for spherical harmonics, we then reconstruct the surface of off-resonance frequencies and estimate the missing off-resonance frequencies.

For single-shot acquisitions, we propose a novel approach that alternates between estimating off-resonance frequencies and the coefficients for harmonics. In each iteration of the proposed alternating approach, we increase the scale of optimization variables and use the previous iterates to predict the region in which the cost function is unimodal around the global minimizer. Increasing the scale of the optimization variables helps to improve the accuracy in estimating accumulated phase errors due to long T_E and readout times. Using approximated off-resonance frequencies, we can estimate the field map matrix. We observe that when the SNR of measurements in the second dataset is sufficiently large, the field map matrix becomes accurate enough to enable correction of the distorted images (generated via different acquisition methods). Here, the NYU DICOM (Digital Imaging and Communications in Medicine) dataset [166], which includes undistorted images from different slices and different contrasts obtained from homogeneous magnetic fields, is utilized to synthetically generate distorted images via several encoding approaches in the presence of large magnetic field inhomogeneity. Noisy undistorted measurements in the second dataset are acquired via the standard multi-shot 2D Cartesian encoding without any significant phase error. We validate the performance of our proposed methods for different encoding methods in the presence of large magnetic field inhomogeneity and demonstrate accurate field map estimations and image corrections when the SNR level in the second dataset is low.

The rest of this chapter is organized as follows. The joint image and field map estimation problem is formulated in Section 5.3. In Section 5.4, a problem decomposition and optimization algorithms to solve the formulated problem are proposed. Simulation results are given in Section 5.5, and concluding remarks are given in Section 5.6.

5.3 Problem Formulation and Notations

5.3.1 Notation

The transpose of a matrix $\mathbf{A}$ is denoted by $\mathbf{A}^T$. Further, the complex conjugate transpose of $\mathbf{A}$ is denoted by $\mathbf{A}^H$, and the conjugate of $\mathbf{A}$ is represented by $\mathbf{A}^*$. The Hadamard (element-wise) product of matrices $\mathbf{A}$ and $\mathbf{B}$ is denoted by $\mathbf{A} \circ \mathbf{B}$. The pseudo-inverse of

$\mathbf{A}$ is represented by $\mathbf{A}^{\dagger}$. The j^{th} row of $\mathbf{A}$ is represented by $\mathbf{A}(j,:)$. If $\mathbf{A}$ and $\mathbf{B}$ have the same dimension, $\mathbf{A} \oslash \mathbf{B}$ denotes the element-wise division. The j^{th} element of the i^{th} row is represented by $[\mathbf{A}]_{ij}$.

5.3.2 Two Datasets of Common Objects

We consider two datasets that include information from a common set of objects. The first dataset comprises naively reconstructed images (without any treatment for phase errors) that are distorted due to the inhomogeneous B_0. We consider that only distorted images are available in the first dataset, while raw signal measurements used to reconstruct distorted images are unavailable (e.g., when measurements are discarded due to storage limitations). Distorted images are reconstructed using one (or multiple) spiral trajectory to encode the k-space information. The second dataset comprises fully sampled signal measurements for the same set of images (2D slices). Fully sampled measurements are assumed to be acquired by an encoding method that is resistant to the inhomogeneity of B_0 (e.g., 3D phase encoding or single point imaging [167, 168, 169]), but the acquisition method is slow and provides low SNR efficiency.

5.3.3 The First Dataset: Distorted Images

In this section, we discuss the way that distorted 2D images are reconstructed from measurements, where the k-space information is encoded using one (or multiple) spiral trajectory. Spatial information in the third dimension is encoded by multishot phase encoding, which is not affected by field inhomogeneity [15]. Each coil is represented by c, where $c \in \{1, \dots, C\}$. The $j^{\text{th}} \in \{1, \dots, n\}$ voxel of the k^{th} unknown image is denoted by x_j^k. Further, the spatially-varying magnetic field inhomogeneity present at position $\mathbf{r} = (\mathtt{x}, \mathtt{y}, \mathtt{z})$ is represented by $\Delta B_0(\mathbf{r})$, which generates a local phase error at position j. The model assumes that the inhomogeneity is piece-wise constant over a voxel. Considering the magnet isocenter is at position $(0, 0, 0)$, the signal equation for the c^{th} coil in the q^{th} shot is [15]

$$y_c^{k,q}(t) = \epsilon_c^{k,q}(t) + \int M^k(\mathbf{r}) s_c(\mathbf{r}) \exp\left(-\iota\gamma\left(\mathbf{r}\cdot\int_0^{T_E+t}\mathbf{G}^q(t')dt' + \int_0^{T_E+t}\Delta B_0(\mathbf{r})dt'\right)\right) d\mathbf{r}, \tag{5.1}$$

where $s_c(\mathbf{r})$ is the c^{th} coil sensitivity and γ is the gyromagnetic ratio. Noise in the measurement is $\epsilon_c^{k,q}(t_i)$, which is taken here to be an independent complex normally distributed noise at each time t. In the above equation, $M^k(\mathbf{r})$ is the magnetization present at position $\mathbf{r}$, $\iota = \sqrt{-1}$, and $\mathbf{G}^q(t)$ is the value of the applied gradient function at time t. Equation (5.1) can be written as

$$\begin{aligned} y_c^{k,q}(t_i) =& \sum_{j=1}^{n} M^k(\mathbf{r}_j) s_c(\mathbf{r}_j) \exp\Bigg(-\iota\gamma\Big(\mathbf{r}_j \cdot \int_0^{T_E+t_i} \mathbf{G}^q(t')dt' \\ &+ \int_0^{T_E+t_i} \Delta B_0(\mathbf{r}_j)dt'\Big)\Bigg) + \epsilon_c^{k,q}(t_i), \end{aligned} \tag{5.2}$$

since only a finite number of k-space samples can be acquired. The acquisition time is between T_E and $T_E + T_{\text{acq}}$, and $t_i \in [0, T_{\text{acq}}]$, where T_E is the echo time and T_{acq} is the acquisition (or readout) duration. At time t_i, the discretized signal by the c^{th} coil in the q^{th} shot can be written as [17]

$$y_c^{k,q}(t_i) = \sum_{j=1}^{n} x_j^k s_c(\mathbf{r}_j) e^{-\iota 2\pi f_j (T_E + t_i)} e^{-\iota 2\pi \mathbf{k}_i^q \cdot \mathbf{r}_j} + \epsilon_c^{k,q}(t_i), \tag{5.3}$$

where $i \in \{1, \ldots, m_1\}$ and f_j is the off-resonance frequency present at position $\mathbf{r}_j$. Based on (5.1) and (5.3), one can obtain the following relations:

$$f_j = \frac{\gamma \Delta B_0(\mathbf{r}_j)}{2\pi}, \tag{5.4}$$

$$\mathbf{k}_i^q = \frac{\gamma}{2\pi} \int_0^{T_E+t_i} \mathbf{G}^q(t')dt'. \tag{5.5}$$

We desire to represent (5.3) in a matrix multiplication form. For spiral encoding, as a prototypical example, the encoding matrix is formed based on $\mathtt{q}$ trajectories, and is denoted by $\mathbf{A}_f^s \in \mathbb{C}^{m_1 \mathtt{q} \times n}$. One can expand $\mathbf{A}_f^s$ as follows:

$$\mathbf{A}_f^s = \left\{\mathbf{A}_f^s(q)\right\}_{q=1}^{\mathtt{q}},$$

where one element of $\mathbf{A}_f^s(q)$ is

$$[\mathbf{A}_f^s(q)]_{ij} = \exp\left(-\iota\gamma\left(\mathbf{r}_j \cdot \int_0^{T_E+t_i} \mathbf{G}^q(t')dt' + \Delta B_0(\mathbf{r}_j)(T_E + t_i)\right)\right),$$

and

$$\mathbf{A}_f^s(q) = \mathbf{A}_0^s(q) \circ \mathbf{\Xi}. \tag{5.6}$$

For $\mathbf{A}_0^s(q)$ and $\mathbf{\Xi}$, we have

$$[\mathbf{A}_0^s(q)]_{ij} = e^{-\iota 2\pi \mathbf{k}_i^q \cdot \mathbf{r}_j}, \quad \text{and} \quad [\mathbf{\Xi}]_{ij} = e^{-\iota 2\pi f_j(T_E+t_i)}. \tag{5.7}$$

In the matrix of phase errors, denoted by $\mathbf{\Xi}$, it is assumed that $t_{i+1} - t_i = \Delta t$, $i \geq 1$. Using the spiral encoding matrix, one can obtain the signal equation (5.3) for the c^{th} coil in the matrix form as follows:

$$\mathbf{y}_c^k = \mathbf{A}_f^s \mathbf{S}_c \mathbf{x}^k + \boldsymbol{\epsilon}_c^k, \tag{5.8}$$

where $\mathbf{S}_c$ is a diagonal matrix and the vector of its diagonal elements is $\mathbf{s}_c = \{s_c(\mathbf{r}_j)\}_{j=1}^n$. Moreover, $\mathbf{y}_c^k = \{\{y_c^{k,q}(t_i)\}_{i=1}^{m_1}\}_{q=1}^{\mathfrak{q}}$, $\boldsymbol{\epsilon}_c^k = \{\{\epsilon_c^{k,q}(t_i)\}_{i=1}^{m_1}\}_{q=1}^{\mathfrak{q}}$, and $\mathbf{x}^k = \{x_j^k\}_{j=1}^n \in \mathbb{C}^{n\times 1}$. The matrix that includes all images in its columns is represented by $\mathbf{X} = \{\mathbf{x}^k\}_{k=1}^K \in \mathbb{C}^{n\times K}$, where $K \geq n$. We consider that the variance of $\boldsymbol{\epsilon}_c^k$ is sufficiently small.

Consider that the distorted images are generated from a naive reconstruction under the assumption that there is no inhomogeneity in the magnetic field. Without any treatment for phase errors, the k^{th} distorted reconstructed image, represented by $\tilde{\mathbf{x}}^k$, is generated from the following optimization:

$$\begin{aligned}\tilde{\mathbf{x}}^k &= \arg\min_{\tilde{\mathbf{x}}^k} \quad \sum_{c=1}^{C} \frac{1}{2}\left\|\mathbf{A}_0^s \mathbf{S}_c \tilde{\mathbf{x}}^k - \mathbf{y}_c^k\right\|_2^2 + \frac{1}{2}\left\|\mathbf{R}\tilde{\mathbf{x}}^k\right\|_2^2 \\ &= \arg\min_{\tilde{\mathbf{x}}^k} \sum_{c=1}^{C} \frac{1}{2}\left\|\mathbf{A}_0^s \mathbf{S}_c \tilde{\mathbf{x}}^k - \mathbf{A}_f^s \mathbf{S}_c \mathbf{x}^k - \boldsymbol{\epsilon}_c^k\right\|_2^2 + \frac{1}{2}\left\|\mathbf{R}\tilde{\mathbf{x}}^k\right\|_2^2,\end{aligned} \tag{5.9}$$

where $\mathbf{A}_0^s = \mathbf{A}_f^s \mid_{\Delta B_0(\mathbf{r})=0}$. We consider that while $\{\tilde{\mathbf{x}}^k\}_{k=1}^K$ is available, there may be no access to $\{\mathbf{y}_c^k\}_{k=1}^K$. Therefore, $\mathbf{A}_f^s$ cannot be directly estimated from measurements, and thus it is unknown. In the above optimization, $\mathbf{R}$ is a regularization matrix to improve the condition number of (5.9).

5.3.4 The Second Dataset: Noisy Measurements

In addition to the first dataset that includes distorted reconstructed images $\{\tilde{\mathbf{x}}^k\}_{k=1}^K$, in a second dataset we consider a set of noisy fully sampled measurements (acquired from the same 2D slices). Fully sampled measurements are assumed to be acquired using an encoding method resistant to magnetic field inhomogeneity (e.g., 3D phase encoding or single point imaging [167, 168, 169]). Consider that the vector of measurements from the k^{th} image using the c^{th} coil is denoted by $\mathbf{w}_c^k$. The encoding matrix is denoted by $\mathbf{A}_0^p$. One can obtain the vector of measurements as

$$\mathbf{w}_c^k = \mathbf{A}_0^p \mathbf{S}_c \mathbf{x}^k + \boldsymbol{\zeta}_c^k.$$

Moreover, $\mathbf{w}_c^k = \{w_c^k(t_i)\}_{i=1}^{m_2}$. It is assumed that $m_2\,C \geq n$ and sufficient measurements exist to reconstruct the k^{th} image via SENSE methods [170]. We consider that the variance of $\boldsymbol{\zeta}_c^k$ is significantly larger than that of $\boldsymbol{\epsilon}_c^k$.

5.3.5 Joint Optimization Problem

We formulate the following optimization problem to jointly estimate images in $\mathbf{X}$ and the field map matrix $\mathbf{A}_f^s$:

$$\min_{\mathbf{X},\mathbf{A}_f^s} \quad \sum_{k=1}^{K}\sum_{c=1}^{C}\frac{1}{2}\left\|\mathbf{A}_0^p\mathbf{S}_c\mathbf{x}^k - \mathbf{w}_c^k\right\|_2^2 \tag{5.10a}$$

$$\text{s.t.} \quad \tilde{\mathbf{x}}^k = \arg\min_{\tilde{\mathbf{x}}^k}\sum_{c=1}^{C}\frac{1}{2}\left\|\mathbf{A}_0^s\mathbf{S}_c\tilde{\mathbf{x}}^k - \mathbf{y}_c^k\right\|_2^2 + \frac{1}{2}\left\|\mathbf{R}\tilde{\mathbf{x}}^k\right\|_2^2. \tag{5.10b}$$

In the above problem, $\mathbf{A}_f^s$ is included in $\mathbf{y}_c^k$ as given in (5.9). In addition to the magnetic field inhomogeneity and noise, a long T_E makes solving the above problem difficult, since a long T_E makes it harder to distinguish between the global and local minima of (5.10).

5.4 Methods

Minimizing (5.10a) simultaneously with respect to $\mathbf{X}$ and $\mathbf{A}_f^s$ is difficult because of the product $\mathbf{A}_f^s\mathbf{S}_c\mathbf{x}^k$ in $\mathbf{y}_c^k$ of (5.10b), which involves both $\mathbf{A}_f^s$ and $\mathbf{X}$ and makes (5.10b) a non-convex constraint. To simplify solving the problem in (5.10), we decouple $\mathbf{A}_f^s$ and $\mathbf{X}$ and propose a BCD algorithm, which alternatively optimizes with respect to $\mathbf{X}$ as one block, and optimizes a transformation of $\mathbf{A}_f^s$ as the second block. Once the optimization (5.10) is solved, one can apply the inverse transformation and extract $\mathbf{A}_f^s$.

5.4.1 Problem Decomposition and Algorithm Design

In order to solve (5.10) with respect to $\mathbf{X}$, we first simplify (5.10b) and obtain an equivalent penalty term to include in the objective function (5.10a). The first order optimality condition for (5.10b) can be written as follows:

$$\nabla_{\tilde{\mathbf{x}}^k}\left(\sum_{c=1}^{C}\frac{1}{2}\left\|\mathbf{A}_0^s\mathbf{S}_c\tilde{\mathbf{x}}^k-\mathbf{y}_c^k\right\|_2^2+\frac{1}{2}\left\|\mathbf{R}\tilde{\mathbf{x}}^k\right\|_2^2\right)=\sum_{c=1}^{C}(\mathbf{A}_0^s\mathbf{S}_c)^H(\mathbf{A}_0^s\mathbf{S}_c\tilde{\mathbf{x}}^k-\mathbf{y}_c^k)+\mathbf{R}^H\mathbf{R}\tilde{\mathbf{x}}^k$$

$$=\sum_{c=1}^{C}(\mathbf{A}_0^s\mathbf{S}_c)^H(\mathbf{A}_0^s\mathbf{S}_c\tilde{\mathbf{x}}^k-\mathbf{A}_f^s\mathbf{S}_c\mathbf{x}^k-\boldsymbol{\epsilon}_c^k)+\mathbf{R}^H\mathbf{R}\tilde{\mathbf{x}}^k$$

$$=\sum_{c=1}^{C}\left[(\mathbf{A}_0^s\mathbf{S}_c)^H\mathbf{A}_0^s\mathbf{S}_c\tilde{\mathbf{x}}^k-(\mathbf{A}_0^s\mathbf{S}_c)^H\mathbf{A}_f^s\mathbf{S}_c\mathbf{x}^k-(\mathbf{A}_0^s\mathbf{S}_c)^H\boldsymbol{\epsilon}_c^k\right]+\mathbf{R}^H\mathbf{R}\tilde{\mathbf{x}}^k=\mathbf{0}.$$

One can solve the above equation with respect to $\tilde{\mathbf{x}}^k$ by finding $\tilde{\mathbf{x}}^k$ that minimizes the right-hand side in the formulation below:

$$\left\|\left[\sum_{c=1}^{C}(\mathbf{A}_0^s\mathbf{S}_c)^H\mathbf{A}_0^s\mathbf{S}_c+\mathbf{R}^H\mathbf{R}\right]\tilde{\mathbf{x}}^k-\left[\sum_{c=1}^{C}(\mathbf{A}_0^s\mathbf{S}_c)^H\mathbf{A}_f^s\mathbf{S}_c\right]\mathbf{x}^k\right\|_2^2\le\beta^k. \tag{5.11}$$

Note that $\mathbf{S}_c$ is a diagonal matrix, and the vector of elements on the diagonal is $\mathbf{s}_c$. Therefore, $\mathbf{S}_c=\text{diag}(\mathbf{s}_c)$, and one can have the following simplification:

$$\sum_{c=1}^{C}(\mathbf{S}_c)^H(\mathbf{A}_0^s)^H\mathbf{A}_0^s\mathbf{S}_c=\sum_{c=1}^{C}(\text{diag}(\mathbf{s}_c))^H(\mathbf{A}_0^s)^H\mathbf{A}_0^s\text{diag}(\mathbf{s}_c)$$

$$= \underbrace{\Big(\sum_{c=1}^{C}(\mathbf{s}_c)^*(\mathbf{s}_c)^T\Big) \circ \Big((\mathbf{A}_0^s)^H\mathbf{A}_0^s\Big)}_{\mathbf{\Gamma}}.$$

Using the above equation, the following simplification can be obtained:

$$\sum_{c=1}^{C}(\mathbf{A}_0^s\mathbf{S}_c)^H\mathbf{A}_f^s\mathbf{S}_c = \underbrace{(\sum_{c=1}^{C}(\mathbf{s}_c)^*(\mathbf{s}_c)^T)}_{\mathbf{V}} \circ \underbrace{\Big((\mathbf{A}_0^s)^H\mathbf{A}_f^s\Big)}_{\mathbf{\Psi}} = \mathbf{\Theta}. \tag{5.12}$$

Based on the definitions of $\mathbf{A}_0^s$ and $\mathbf{A}_f^s$,

$$\mathbf{\Psi} = (\mathbf{A}_0^s)^H\mathbf{A}_f^s = \sum_{q=1}^{\mathtt{q}}(\mathbf{A}_0^s(q))^H\mathbf{A}_f^s(q), \tag{5.13}$$

can be obtained, where

$$[\mathbf{\Psi}]_{oj} = \sum_{q=1}^{\mathtt{q}}\sum_{i=1}^{m_1} e^{-\iota 2\pi f_j(T_E+t_i)} e^{\iota 2\pi \mathbf{k}_i^q.(\mathbf{r}_o-\mathbf{r}_j)}. \tag{5.14}$$

The above equation is used in the next section to estimate f_j. We simplify (5.11) as follows:

$$\left\|(\mathbf{\Gamma}+\mathbf{R}^H\mathbf{R})\tilde{\mathbf{x}}^k - (\mathbf{V}\circ\mathbf{\Psi})\mathbf{x}^k\right\|_2^2 \leq \beta^k, \qquad \forall k. \tag{5.15}$$

Note that (5.15) makes solving (5.10) difficult, since the optimization variable $\mathbf{A}_f^s$, embedded in $\mathbf{\Psi}$, is multiplied by $\mathbf{x}^k$. This makes (5.15) a non-convex constraint. With a sufficient number of coils, $\mathbf{\Theta}$ becomes invertible. To tackle the non-convexity issue when $\mathbf{\Theta}$ is invertible, we define a new variable $\mathbf{\Phi} = \mathbf{\Theta}^{-1}$. One can rewrite (5.15) as follows:

$$\left\|\mathbf{\Phi}(\mathbf{\Gamma}+\mathbf{R}^H\mathbf{R})\tilde{\mathbf{x}}^k - \mathbf{x}^k\right\|_2^2 \leq \beta^k, \qquad \forall k. \tag{5.16}$$

Using the above inequality, (5.10) can be reformulated as follows:

$$\min_{\mathbf{X},\mathbf{\Phi},\boldsymbol{\beta}} \quad \sum_{k=1}^{K}\sum_{c=1}^{C}\frac{1}{2}\left\|\mathbf{A}_0^p\mathbf{S}_c\mathbf{x}^k - \mathbf{w}_c^k\right\|_2^2 + \Upsilon\sum_{k=1}^{K}\beta^k$$

$$\text{s.t.} \quad (5.16), \tag{5.17}$$

where $\Upsilon \geq 0$. One can rewrite the above optimization problem in the following unconstrained form:

$$\min_{\mathbf{X},\boldsymbol{\Phi}} \quad \sum_{k=1}^{K} \Big(\sum_{c=1}^{C} \frac{1}{2} \left\| \mathbf{A}_0^p \mathbf{S}_c \mathbf{x}^k - \mathbf{w}_c^k \right\|_2^2 + \Upsilon \left\| \boldsymbol{\Phi}(\boldsymbol{\Gamma} + \mathbf{R}^H \mathbf{R}) \tilde{\mathbf{x}}^k - \mathbf{x}^k \right\|_2^2 \Big). \tag{5.18}$$

The number of optimization variables is $nK + n^2$. When this number is large, solving (5.18) is computationally demanding. One can alternatively optimize with respect to $\mathbf{X}$ as one block and $\boldsymbol{\Phi}$ as the second block. To ensure the convergence of the proposed BCD, the subproblem with respect to each block should have a unique solution [111, Proposition 3.7.1]. In practice, it is difficult to ensure that $(\boldsymbol{\Gamma} + \mathbf{R}^H \mathbf{R})\tilde{\mathbf{X}}$ has a full row rank. Thus, we utilize the change of variables. Before that, let us define

$$\begin{aligned} \mathbf{Q}_0^p &= \{\mathbf{A}_0^p \mathbf{S}_c\}_{c=1}^{C} \in \mathbb{C}^{m_2 C \times n}, \\ \mathbf{W} &= \{\{\mathbf{w}_c^k\}_{c=1}^{C}\}_{k=1}^{K} \in \mathbb{C}^{m_2 C \times K}. \end{aligned}$$

One can simplify (5.18) using the above notations as follows:

$$\min_{\mathbf{X},\boldsymbol{\Phi}} \quad \frac{1}{2} \left\| \mathbf{Q}_0^p \mathbf{X} - \mathbf{W} \right\|_F^2 + \Upsilon \left\| \boldsymbol{\Phi}(\boldsymbol{\Gamma} + \mathbf{R}^H \mathbf{R}) \tilde{\mathbf{X}} - \mathbf{X} \right\|_F^2 .$$

Let us define a new variable block as $\boldsymbol{\Lambda} = \boldsymbol{\Phi}(\boldsymbol{\Gamma} + \mathbf{R}^H \mathbf{R})$. Using this new variable block, one obtains

$$\min_{\mathbf{X},\boldsymbol{\Lambda}} \quad \frac{1}{2} \left\| \mathbf{X} - (\mathbf{Q}_0^p)^{\dagger} \mathbf{W} \right\|_F^2 + \Upsilon \left\| \boldsymbol{\Lambda} \tilde{\mathbf{X}} - \mathbf{X} \right\|_F^2 . \tag{5.19}$$

To update each block of variables, we enable a voxel-level decomposition, where one voxel from each image is optimized along with one row of $\boldsymbol{\Lambda}$. In the above optimization problem, each row of $\boldsymbol{\Lambda}$ corresponds to one position (or voxel). Moreover, it can be observed that the problem is separable across voxels, as the coefficient matrix for $\mathbf{X}$ in both terms is an

```
Algorithm 17: Voxel-level BCD Algorithm to solve (5.20)
  0. Initialization $\mathbf{X}^0(j,:) = \mathbf{0}$, $\mathbf{\Lambda}^0(j,:) = \mathbf{0}$, and $t = 0$;
  repeat
      1. Update $\mathbf{X}^{t+1}(j,:)$ using (5.21);
      2. Update $\mathbf{\Lambda}^{t+1}(j,:)$ using (5.22);
      3. $t = t + 1$;
  until $\left\|\mathbf{\Lambda}^t(j,:) - \mathbf{\Lambda}^{t-1}(j,:)\right\|_2^2 + \left\|\mathbf{X}^t(j,:) - \mathbf{X}^{t-1}(j,:)\right\|_2^2$ is small enough;
  4. $\hat{\mathbf{\Lambda}}(j,:) = \mathbf{\Lambda}^t(j,:)$ and $\hat{\mathbf{X}}(j,:) = \mathbf{X}^t(j,:)$;
```

identity matrix. The per-voxel subproblem corresponding to the j^{th} voxel is

$$\min_{\mathbf{X}(j,:),\mathbf{\Lambda}(j,:)} \quad \frac{1}{2}\left\|\mathbf{X}(j,:) - (\mathbf{Q}_0^p)^\dagger(j,:)\mathbf{W}\right\|_2^2 + \Upsilon\left\|\mathbf{\Lambda}(j,:)\tilde{\mathbf{X}} - \mathbf{X}(j,:)\right\|_2^2. \tag{5.20}$$

In the subproblem for the j^{th} voxel, one block is $\mathbf{X}(j,:)$ and the other block is $\mathbf{\Lambda}(j,:)$. To solve the problem in (5.20), one can optimize with respect to two blocks of variables, $\mathbf{X}(j,:)$ and $\mathbf{\Lambda}(j,:)$, alternatively. Therefore, if we choose $\mathbf{X}(j,:)$ to update first, with $\mathbf{\Lambda}^t(j,:)$, we optimize with respect to $\mathbf{X}(j,:)$. Then, we update $\mathbf{\Lambda}^{t+1}(j,:)$. Optimizing with respect to $\mathbf{X}(j,:)$ and $\mathbf{\Lambda}(j,:)$ alternatively is continued until both blocks converge. In the t^{th} iteration of Algorithm 17, the two blocks are updated as follows:

$$\mathbf{X}^{t+1}(j,:) = \frac{1}{1+2\Upsilon}\left((\mathbf{Q}_0^p)^\dagger(j,:)\mathbf{W} + \mathbf{\Lambda}^t(j,:)\tilde{\mathbf{X}}\right), \tag{5.21}$$

$$\mathbf{\Lambda}^{t+1}(j,:) = \mathbf{X}^{t+1}(j,:)\tilde{\mathbf{X}}^\dagger. \tag{5.22}$$

Since the coefficient of $\mathbf{X}(j,:)$ is identity in (5.20) , the subproblem with respect to $\mathbf{X}(j,:)$ has a unique solution. The coefficient of $\mathbf{\Lambda}(j,:)$ in (5.20) is $\tilde{\mathbf{X}}$. The full row rankness of $\tilde{\mathbf{X}}$ ensures that the subproblem with respect to $\mathbf{\Lambda}(j,:)$ has a unique solution. To increase the chance of $\tilde{\mathbf{X}}$ having a full row rank, the following factors are helpful:

1. Dataset images should be different, linearly independent, and $K \geq n$;

2. The number of receiving coils should be increased to twice the acceleration factor;

3. The number of acquired measurements to reconstruct images, i.e., m_1, in the first dataset, should be increased to $2n/C$.

The summary of the BCD approach is given in Algorithm 17. After the convergence criteria is met, we set $\hat{\mathbf{\Lambda}} = \mathbf{\Lambda}^t$ and $\hat{\mathbf{X}} = \mathbf{X}^t$.

If $\mathbf{\Theta}$ is not invertible or $\tilde{\mathbf{X}}$ does not have a full row rank, one can estimate images from noisy measurements and use that to approximate $\mathbf{\Theta}$ as follows:

$$\hat{\mathbf{X}} = (\mathbf{Q}_0^p)^\dagger \mathbf{W}, \qquad \hat{\mathbf{\Theta}} = (\mathbf{\Gamma} + \mathbf{R}^H \mathbf{R}) \tilde{\mathbf{X}} \hat{\mathbf{X}}^\dagger.$$

5.4.2 Estimation of Off-Resonance Frequencies

Suppose that (5.20) is solved for all voxels in parallel. Then, $\mathbf{\Lambda}$ is estimated. Since $\mathbf{\Gamma}+\mathbf{R}^H\mathbf{R}$ is invertible, one can estimate $\mathbf{\Phi}$ via $\hat{\mathbf{\Phi}} = \hat{\mathbf{\Lambda}}(\mathbf{\Gamma} + \mathbf{R}^H\mathbf{R})^{-1}$. Then, $\mathbf{\Theta}$ can be estimated via $\hat{\mathbf{\Theta}} = \hat{\mathbf{\Phi}}^{-1}$. The other way of estimating $\mathbf{\Theta}$ is solving the following optimization problem:

$$\hat{\mathbf{\Theta}} = \min_{\mathbf{\Theta}} \quad \left\| (\mathbf{\Gamma} + \mathbf{R}^H \mathbf{R}) \tilde{\mathbf{X}} - \mathbf{\Theta} \hat{\mathbf{X}} \right\|_F^2, \tag{5.23}$$

which is a linear regression problem that can be solved via gradient methods. After $\mathbf{\Theta}$ is approximated, one can estimate $\mathbf{\Psi}$ using $\hat{\mathbf{\Psi}} = \hat{\mathbf{\Theta}} \oslash \mathbf{V}$.

From (5.12), we observe that $\mathbf{A}_f^s$ is involved in $\mathbf{\Psi}$. Therefore, $\mathbf{\Psi}$ can be used to estimate $\mathbf{A}_f^s$. From (5.6), we see that $\mathbf{A}_f^s(q)$ is formed from $\mathbf{A}_0^s(q)$ and matrix $\mathbf{\Xi}$ that includes phase errors in different voxels. Since $\mathbf{A}_0^s(q)$ is known, to estimate $\mathbf{A}_f^s(q)$, it suffices to estimate the matrix of phase errors $\mathbf{\Xi}$. This matrix is identical for all shots. Note that to reconstruct the matrix of phase errors, one needs to estimate $\{e^{-\iota 2\pi f_j \Delta t}\}_{j=1}^n$. In order to characterize $\{e^{-\iota 2\pi f_j \Delta t}\}_{j=1}^n$, it is enough to estimate $\{f_j\}_{j=1}^n$. From (5.14), one can estimate $\{f_j\}_{j=1}^n$ from $\mathbf{\Psi}$ directly using the following regression:

$$\min_{\{f_j\}_{j=1}^n} \sum_{j=1}^{n} \sum_{o=1}^{n} \left| [\hat{\mathbf{\Psi}}]_{oj} - \sum_{q=1}^{\mathtt{q}} \sum_{i=1}^{m_1} e^{-\iota 2\pi f_j (T_E + t_i)} e^{\iota 2\pi \mathbf{k}_i^q . (\mathbf{r}_o - \mathbf{r}_j)} \right|^2.$$

The above problem is highly non-convex with respect to $\{f_j\}_{j=1}^n$, which makes solving it challenging. Due to this fact, we concentrate on estimating $\{2\pi f_j \Delta t\}_{j=1}^n$ instead of $\{f_j\}_{j=1}^n$. We thus propose a novel method to estimate $\{2\pi f_j \Delta t\}_{j=1}^n$ from $(\mathbf{A}_0^s)^H \mathbf{A}_f^s = \mathbf{\Psi}$.

Let us denote $2\pi f_j \Delta t$ by g_j. From (5.6), it is observed that $\mathbf{A}_f^s(:, j)$ is parameterized by

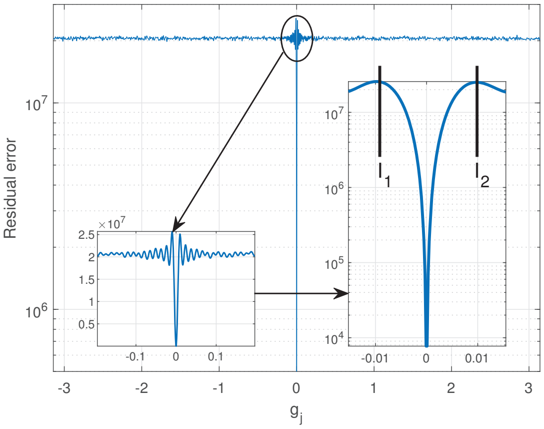


Figure 5.1: The function in (5.24) in $[-\pi, \pi]$, which includes one global minima and several local minima. The global minimizer can be obtained via a golden-section search nested inside a grid-search, regardless of the cost function non-convexity.

$e^{-\iota g_j}$. Based on equation (5.14), the following non-linear cost function, $\Omega(g_j)$, is formulated:

$$\sum_{o=1}^{n} \left| [\hat{\mathbf{\Psi}}]_{oj} - \sum_{q=1}^{\mathsf{q}} \sum_{i=1}^{m_1} e^{-\iota \frac{g_j t_1}{\Delta t}} e^{-\iota g_j (i-1+\frac{T_E}{\Delta t})} e^{\iota 2\pi \mathbf{k}_i^q . (\mathbf{r}_o - \mathbf{r}_j)} \right|^2, \tag{5.24}$$

and used to estimate g_j as follows:

$$\min_{g_j} \quad \Omega(g_j).$$

The cost function for the above regression is depicted in Fig. 5.1. In practice, f_j is limited, and Δt is in the order of microseconds. Hence, $|2\pi f_j \Delta t| < \pi$. In order to estimate $e^{-\iota 2\pi f_j \Delta t}$, it is enough to implement a grid-search in the interval $(-\pi, \pi)$ for $2\pi f_j \Delta t$ to obtain an interval $[I_1, I_2]$, in which the cost function is unimodal around the global minimizer. When the grid-search has a sufficient accuracy, the sharp minimum in the cost function is detected and interval $[I_1, I_2]$ is identified. When m_1 in (5.24) increases, the difference between the function value of the sharp minima in (5.24) and the other values becomes larger. Therefore, it becomes easier to detect the global minima.

Once we obtain the neighborhood in which the cost function is unimodal, we implement a golden-section search method, which is a gradient-free approach, to extract the global

Algorithm 18: Grid-search approach to solve (5.24)

```
0. Initialization $g_j = -\pi$, $\Delta g_j$ is a small enough positive number, $h_1 = \text{inf}$,
 $h_2 = 0$, $h_3 = 0$, $h_4 = 0$, $h_5 = 1$, $h_6 = 1$;
repeat
    if $h_1 > \Omega(g_j)$ then
        1. $h_1 = \Omega(g_j)$;
        2. $h_2 = g_j$;
    3. $g_j = g_j + \Delta g_j$;
until $g_j \leq \pi$;
4. $h_3 = h_2$;
repeat
    5. $h_3 = h_3 - \Delta g_j$, $h_3^o = h_2$;
    if $\Omega(g_j) \mid_{g_j = h_3^o} < \Omega(g_j) \mid_{g_j = h_3}$ then
        6. $h_3^o = h_3$;
        7. $h_3 = h_3 - \Delta g_j$;
        8. $h_5 = 1$;
    else
        9. $h_5 = 0$;
until $h_5 = 0$;
10. $h_4 = h_2$;
repeat
    11. $h_4 = h_4 + \Delta g_j$, $h_4^o = h_2$;
    if $\Omega(g_j) \mid_{g_j = h_4^o} < \Omega(g_j) \mid_{g_j = h_4}$ then
        12. $h_4^o = h_4$;
        13. $h_4 = h_4 + \Delta g_j$;
        14. $h_6 = 1$;
    else
        15. $h_6 = 0$;
until $h_6 = 0$;
16. $I_1 = h_3^o$;
17. $I_2 = h_4^o$;
18. Run the golden-section search algorithm;
```

minimizer. The proposed approach to solve (5.24) is described in Algorithm 18. The golden-section search approach is briefly described in Algorithm 19. In Algorithm 18, the golden-section search is called, and it extracts the global minimizer from $[I_1 \;\, I_2]$. Once the global minimizer, $\hat{g}_j$, is derived, one can obtain the off-resonance frequency using $\hat{f}_j = \hat{g}_j / 2\pi\Delta t$.

Algorithm 19: The golden-section search approach

```
0. Initialization $A_1 = I_1$, $A_2 = I_2$;
1. $G_r = \frac{\sqrt{5}-1}{2}$;
2. $A_3 = A_1 + (1 - G_r)(A_2 - A_1)$;
3. $A_4 = A_1 + G_r(A_2 - A_1)$;
4. $V_1 = \Omega(g_j) \mid_{g_j = A_3}$;
5. $V_2 = \Omega(g_j) \mid_{g_j = A_4}$;
repeat
    if $V_1 < V_2$ then
        6. $A_2 = A_4$;
        7. $A_4 = A_3$;
        8. $A_3 = A_1 + (1 - G_r)(A_2 - A_1)$;
        9. $V_1 = \Omega(g_j) \mid_{g_j = A_3}$;
        10. $V_2 = \Omega(g_j) \mid_{g_j = A_4}$;
    else
        11. $A_1 = A_3$;
        12. $A_3 = A_4$;
        13. $A_4 = A_1 + G_r(A_2 - A_1)$;
        14. $V_1 = \Omega(g_j) \mid_{g_j = A_3}$;
        15. $V_2 = \Omega(g_j) \mid_{g_j = A_4}$;
until $|A_1 - A_2|$ is small enough;
16. $\hat{g}_j = A_3$;
17. $\hat{f}_j = \frac{\hat{g}_j}{2\pi\Delta t}$;
```

5.4.3 Estimation of The Inhomogeneous Magnetic Field

In this section, estimating the inhomogeneity of the magnetic field from off-resonance frequencies returned by Algorithm 18 is discussed. When the SNR level of measurements in the second dataset is low, Algorithm 18 cannot estimate the off-resonance frequency in a voxel if the voxel value is small. This is due to the fact that the drop in (5.24) disappears. The advantage of estimating the field inhomogeneity is that if f_j cannot be estimated for position j due to the additive noise, f_j can be approximated from (5.4) using the estimated field inhomogeneity. The set of voxels for which the off-resonance frequencies can be estimated is denoted by $\mathcal{V}$. We propose a linear regression that uses off-resonance frequencies from a subset of voxels, $\mathcal{V}$, to reconstruct $\Delta B_0(\mathbf{r})$. Based on [171, eq. (1)], any

$$\mathbf{e}_j = \begin{pmatrix}1 & \mathtt{x}_j - \mathtt{o}_1 & \mathtt{y}_j - \mathtt{o}_2 & (\mathtt{x}_j - \mathtt{o}_1)^2 + (\mathtt{y}_j - \mathtt{o}_2)^2 & (\mathtt{x}_j - \mathtt{o}_1)^2 - (\mathtt{y}_j - \mathtt{o}_2)^2 & (\mathtt{x}_j - \mathtt{o}_1)(\mathtt{y}_j - \mathtt{o}_2)\end{pmatrix}. \tag{5.25}$$

Table 5.1: Spherical harmonic functions in Cartesian coordinates

n	m	Cartesian harmonic
1	0	$\mathtt{z} - \mathtt{o}_3$
1	1	$\mathtt{x} - \mathtt{o}_1$
1	1	$\mathtt{y} - \mathtt{o}_2$
2	0	$(\mathtt{z} - \mathtt{o}_3)^2 - \left((\mathtt{x} - \mathtt{o}_1)^2 + (\mathtt{y} - \mathtt{o}_2)^2\right)/2$
2	1	$(\mathtt{x} - \mathtt{o}_1)(\mathtt{z} - \mathtt{o}_3)$
2	1	$(\mathtt{y} - \mathtt{o}_2)(\mathtt{z} - \mathtt{o}_3)$
2	2	$(\mathtt{x} - \mathtt{o}_1)^2 - (\mathtt{y} - \mathtt{o}_2)^2$
2	2	$2(\mathtt{x} - \mathtt{o}_1)(\mathtt{y} - \mathtt{o}_2)$
3	0	$(\mathtt{z} - \mathtt{o}_3)\left((\mathtt{z} - \mathtt{o}_3)^2 - 3((\mathtt{x} - \mathtt{o}_1)^2 + (\mathtt{y} - \mathtt{o}_2)^2)/2\right)$
3	1	$(\mathtt{x} - \mathtt{o}_1)\left(4(\mathtt{z} - \mathtt{o}_3)^2 - (\mathtt{x} - \mathtt{o}_1)^2 - (\mathtt{y} - \mathtt{o}_2)^2\right)$
3	1	$(\mathtt{y} - \mathtt{o}_2)\left(4(\mathtt{z} - \mathtt{o}_3)^2 - (\mathtt{x} - \mathtt{o}_1)^2 - (\mathtt{y} - \mathtt{o}_2)^2\right)$
3	2	$(\mathtt{z} - \mathtt{o}_3)\left((\mathtt{x} - \mathtt{o}_1)^2 - (\mathtt{y} - \mathtt{o}_2)^2\right)$
3	2	$2(\mathtt{x} - \mathtt{o}_1)(\mathtt{y} - \mathtt{o}_2)(\mathtt{z} - \mathtt{o}_3)$
3	3	$(\mathtt{x} - \mathtt{o}_1)^3 - 3(\mathtt{x} - \mathtt{o}_1)(\mathtt{y} - \mathtt{o}_2)^2$
3	3	$3(\mathtt{x} - \mathtt{o}_1)^2(\mathtt{y} - \mathtt{o}_2) - (\mathtt{y} - \mathtt{o}_2)^3$

static magnetic field can be separated into a linear combination of several spherical harmonic functions. A list of low order 2D spherical harmonics in spherical and Cartesian coordinates is given in [171, Table 1]. In practice, 2D spherical harmonic functions up to the second order suffice to well represent the field inhomogeneity. 2D spherical harmonic functions up to the third order in Cartesian coordinates are listed in Table 5.1. In Table 5.1, $\mathbf{o} = \{\mathtt{o}_1, \mathtt{o}_2, \mathtt{o}_3\}$ represents the isocenter of the magnet. In the 2D plane of an image, $\mathtt{z}$ is a constant and does not change with a voxel position. As $\mathtt{z}$ is a constant, the harmonic $(\mathtt{x} - \mathtt{o}_1)(\mathtt{z} - \mathtt{o}_3)$ is equal to $\mathtt{x} - \mathtt{o}_1$ after multiplication with a constant coefficient depending on $(\mathtt{z} - \mathtt{o}_3)$. Based on this fact, one can represent an arbitrary, inhomogeneous magnetic field as follows:

$$\Delta B_0(\mathbf{r}) = c_1 + c_2(\mathtt{x} - \mathtt{o}_1) + c_3(\mathtt{y} - \mathtt{o}_2) + c_4((\mathtt{x} - \mathtt{o}_1)^2 + (\mathtt{y} - \mathtt{o}_2)^2) \\ + c_5((\mathtt{x} - \mathtt{o}_1)^2 - (\mathtt{y} - \mathtt{o}_2)^2) + c_6(\mathtt{x} - \mathtt{o}_1)(\mathtt{y} - \mathtt{o}_2), \tag{5.26}$$

where $c_1, c_2, c_3, c_4, c_5, c_6 \in \mathbb{R}$. The spatial distribution of f_j builds a surface in the 2D plane of an image, as depicted in Fig. 5.3. From (5.4), it follows $f_j = \frac{\gamma \Delta B_0(\mathbf{r}_j)}{2\pi}$. This equation is used to estimate $\{f_j\}_{j=1}^n$, which can be leveraged to approximate the coefficients for the harmonics as follows:

$$\min_{\mathbf{c}} \quad \sum_{j \in \mathcal{V}} \left| \frac{\gamma \mathbf{e}_j \mathbf{c}}{2\pi} - \hat{f}_j \right|^2, \tag{5.27}$$

where $\mathbf{e}_j$ is defined in (5.25) and $\hat{f}_j = \frac{\log(e^{-\iota \hat{g}_j})}{2\pi \Delta t}$. The above problem can be solved in closed-form as follows:

$$\hat{\mathbf{c}} = \frac{2\pi}{\gamma} \mathbf{E}^\dagger \hat{\mathbf{f}}, \tag{5.28}$$

where $\mathbf{E} = \{\mathbf{e}_j\}_{j \in \mathcal{V}}$. Using the estimated $\hat{\mathbf{c}}$, one can predict f_j for a position $j \notin \mathcal{V}$ from (5.4). The isocenter of the magnet $\mathbf{o}$ is used in $\mathbf{e}_j$. In practice, it is difficult to ensure where the isocenter of the magnet is. In the following proposition, we prove that without information about the magnet isocenter, we are still able to find a representation for the surface of off-resonance frequencies and carry out the regularization. In this case, an arbitrary $\mathbf{o}$ is chosen and (5.27) is solved, although the estimated $\mathbf{c}$ in this case is not necessarily equal to the ground-truth $\mathbf{c}$.

Proposition 8. *If the isocenter of the magnet* $\mathbf{o}$ *is known, the ground-truth* $\mathbf{c}$ *can be obtained from* (5.27)*. If the isocenter of the magnet* $\mathbf{o}$ *is unknown and arbitrarily chosen, one representation for the surface of off-resonance frequencies is obtained.*

Proof. Equation (5.26) can be simplified as follows:

$$\Delta B_0(\mathbf{r}) = \underbrace{(c_4 + c_5)}_{c_1'} \mathtt{x}^2 + \underbrace{(c_4 - c_5)}_{c_2'} \mathtt{y}^2 + \underbrace{(c_2 - 2\mathtt{o}_1 c_4 - 2\mathtt{o}_1 c_5 - \mathtt{o}_2 c_6)}_{c_3'} \mathtt{x}$$

$$+\underbrace{(c_3 - 2\mathtt{o}_2 c_4 + 2\mathtt{o}_2 c_5 - \mathtt{o}_1 c_6)}_{c'_4}\mathtt{y} + \underbrace{c_6}_{c'_5}\mathtt{xy}$$

$$+\underbrace{c_1 - c_2\mathtt{o}_1 - c_3\mathtt{o}_2 + c_4(\mathtt{o}_1^2 + \mathtt{o}_2^2) + c_5(\mathtt{o}_1^2 - \mathtt{o}_2^2) + c_6\mathtt{o}_1\mathtt{o}_2}_{c'_6}.$$

In order to reconstruct the surface, it is enough to estimate $\mathbf{c}'$. One can estimate $\mathbf{c}'$ using the following linear regression problem:

$$\min_{\mathbf{c}'} \sum_{j\in\mathcal{V}} \left| \begin{pmatrix} \mathtt{x}_j^2 & \mathtt{y}_j^2 & \mathtt{x}_j & \mathtt{y}_j & \mathtt{x}_j\mathtt{y}_j & 1 \end{pmatrix} \mathbf{c}' - 2\pi\hat{f}_j/\gamma \right|^2. \tag{5.29}$$

Let us define

$$\kappa = \left\{ \begin{pmatrix} \mathtt{x}_j^2 & \mathtt{y}_j^2 & \mathtt{x}_j & \mathtt{y}_j & \mathtt{x}_j\mathtt{y}_j & 1 \end{pmatrix} \right\}_{j\in\mathcal{V}}.$$

It is straightforward to verify that κ has a full column rank. Therefore, a unique solution for $\mathbf{c}'$ is guaranteed and one representation is obtained. Using c'_1 and c'_2, one can estimate c_4 and c_5 via a linear system. Moreover, we have $c_6 = c'_5$. When $\mathbf{o}$ is unknown, four unknowns c_2, c_3, $\mathtt{o}_1$, and $\mathtt{o}_2$ are involved in specifying c'_3 and c'_4. Therefore, unique solutions for c_2 and c_3 cannot be obtained. When $\mathbf{o}$ is known, c_2 and c_3 are uniquely identified from c'_3 and c'_4 using known c_4, c_5 and c_6. Finally, with known $\mathbf{o}$, based on estimated $\{c_2, c_3, c_4, c_5, c_6\}$, one can extract c_1 from c'_6. It can be shown that the same result is true when the magnetic field requires expansion up to the third order. We omit it due to page limitations. □

After $\mathbf{c}$ is approximated from (5.28), the off-resonance frequency in each voxel can be estimated using $f_j = \frac{\gamma\Delta B_0(\mathbf{r}_j)}{2\pi}$. After obtaining off-resonance frequencies in all voxels, one can estimate $\mathbf{A}_f^s(q)$, and consequently $\mathbf{\Theta}$. For signal-shot readout gradients, such as one-interleaf spiral, the accuracy of the estimated phase error $e^{-\iota i g_j}$ can degrade as i increases due to the long readout. The same accuracy issue can happen for the estimation of $e^{-\iota 2\pi f_j T_E}$. In order to enhance the estimation accuracy of $e^{-\iota 2\pi f_j T_E}$ and $e^{-\iota m_1 g_j}$, we propose an algorithm that combines Algorithm 18 and the surface estimation.

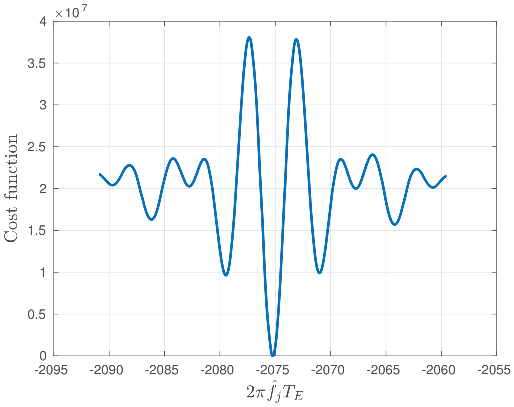


Figure 5.2: The cost function of (5.31) attains multiple local minima. The interval in which the cost function is unimodal around the global minimizer can be predicted based on the previous iterate.

5.4.4 Increasing the Scale of Optimization Variables

Since $2\pi f_j T_E > g_j$ and $2\pi f_j m_1 \Delta t > g_j$, the accuracy of estimation for g_j improves if we estimate $2\pi f_j T_E$ or $2\pi f_j m_1 \Delta t$ instead of g_j. The difficulty of optimizing $e^{-\iota 2\pi f_j T_E}$ (or $e^{-\iota 2\pi f_j m_1 \Delta t}$) is that when $|f_j| > \frac{1}{2T_E}$, the wrapped phase $\log(e^{-\iota 2\pi f_j T_E})/\iota$ is unequal to $-2\pi f_j T_E$. Therefore, one cannot directly relate $\log(e^{-\iota 2\pi f_j T_E})$ to g_j using $g_j = \frac{\log(e^{-\iota 2\pi f_j T_E})\Delta t}{-\iota T_E}$. To resolve the phase ambiguity problem, we first predict the unwrapped phase using the estimated coefficients for spherical harmonics given in the previous section. When $\mathbf{c}$ is estimated from (5.28), one can use $\hat{\mathbf{c}}$ and predict the off-resonance frequency in voxel j, denoted by $\hat{u}_j^1$. Then, the predicted unwrapped phase of $e^{-\iota 2\pi f_j 2\Delta t}$ is $-2\pi \hat{u}_j^1 2\Delta t$. In (5.24), the optimization variable g_j is changed to δ_j^2, where $\delta_j^2 = 2\pi f_j 2\Delta t$. Since δ_j^2 can be predicted (as $-2\pi \hat{u}_j^1 2\Delta t$), one can choose an interval around the predicted δ_j^2 in which the cost function is unimodal. Problem (5.24) can be solved with new variables using Algorithm 18. In the h^{th} iteration, $\delta_j^h = 2\pi h f_j \Delta t$ is optimized. We formulate the following problem to optimize δ_j^h using $\hat{u}_j^{h-1} = \frac{\gamma \mathbf{e}_j \hat{\mathbf{c}}}{2\pi}$:

$$\begin{aligned} \min_{\delta_j^h, j \in \mathcal{V}} \; & \sum_{o=1}^{n} \left| [\hat{\mathbf{\Psi}}]_{oj} - \sum_{q=1}^{\mathtt{q}} \sum_{i=1}^{m_1} e^{-\iota \frac{\delta_j^h t_1}{h\Delta t}} e^{-\iota \delta_j^h (\frac{i-1}{h} + \frac{T_E}{h\Delta t})} e^{\iota 2\pi \mathbf{k}_i^q . (\mathbf{r}_o - \mathbf{r}_j)} \right|^2 \\ \text{s.t.} \quad & 2\pi h \hat{u}_j^{h-1} \Delta t - b_1 \leq \delta_j^h \leq 2\pi h \hat{u}_j^{h-1} \Delta t + b_2, \end{aligned} \tag{5.30}$$

where b_1 and b_2 are two positive numbers that are adjusted such that the cost function is unimodal around the estimated global minimizer. We continue optimizing δ_j^h until $h = m_1$. If $T_E > m_1 \Delta t$, we optimize $\delta_j^{m_1+1} = 2\pi f_j T_E$ in the final step using the following problem:

$$\begin{aligned} \min_{\delta_j^{m_1+1}, j \in \mathcal{V}} \quad & \sum_{o=1}^{n} \Bigg| [\hat{\mathbf{\Psi}}]_{oj} \\ & - \sum_{q=1}^{\mathsf{q}} \sum_{i=1}^{m_1} e^{-\iota \frac{\delta_j^{m_1+1} t_1}{T_E}} e^{-\iota \delta_j^{m_1+1} (1 + \frac{(i-1)\Delta t}{T_E})} e^{\iota 2\pi \mathbf{k}_i^q . (\mathbf{r}_o - \mathbf{r}_j)} \Bigg|^2 \\ & \text{s.t.} \quad 2\pi \hat{u}_j^{m_1} T_E - b_1 \leq \delta_j^{m_1+1} \leq 2\pi \hat{u}_j^{m_1} T_E + b_2. \end{aligned} \tag{5.31}$$

The cost function of this optimization is depicted in Fig. 5.2. When T_E increases, the function value of a local minima around the global minimizer (see Fig. 5.2) decreases. Hence, it becomes more difficult to distinguish between the global and local minima. The optimizations in (5.30)–(5.31) can be solved via Algorithm 18, similar to (5.24). Finally, $\{\hat{f}_j\}_j$ is estimated via $\hat{f}_j = \frac{\hat{\delta}_j^{m_1+1}}{2\pi T_E}$, and the estimated $\mathbf{f}$ can be used to approximate $\mathbf{c}$ from (5.27).

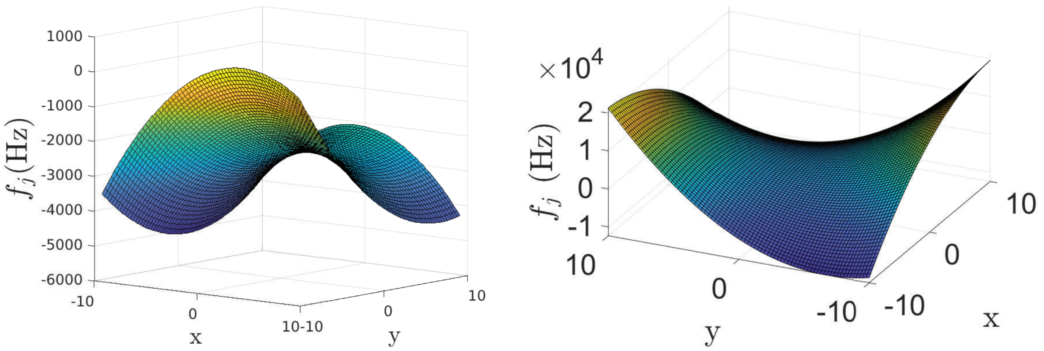
As the noise level in the first dataset is assumed to be low, one can use approximated $\mathbf{A}_f^s(q)$ to estimate the ground-truth images from the first dataset. Mathematically, one can solve the following optimization to estimate $\mathbf{x}^k$:

$$\min_{\mathbf{x}^k} \quad \left\| (\mathbf{\Gamma} + \mathbf{R}^H \mathbf{R}) \tilde{\mathbf{x}}^k - \hat{\mathbf{\Theta}} \mathbf{x}^k \right\|_2^2. \tag{5.32}$$

When $\hat{\mathbf{\Theta}}$ is rank deficient, different regularization methods, e.g., including ℓ_2, ℓ_1 or ℓ_0-norm in the cost function, can be used in the above optimization to efficiently estimate $\mathbf{x}^k$ [170].

5.5 Numerical Simulations

We validate the performance of our algorithm and its efficiency using real and synthetic experimental data. The NYU DICOM (Digital Imaging and Communications in Medicine) dataset of brain images is used, which includes multiple slices with different contrasts. We resize $7,500$ images of the above dataset into 64×64. To experimentally obtain a B_0 map, large B_0 inhomogeneity was experimentally generated by deliberately mis-setting the



(a)

(b)

Figure 5.3: The ground-truth surfaces of off-resonance frequencies. (a) The surface for distorted images of size 64×64, for which a multi-shot Cartesian encoding is used. (b) The surface of off-resonance frequencies estimated from images of size 80×80, where a single interleaf spiral encoding is used to generate images in the first dataset.

room temperature shims. A 3D Cartesian k-space was sampled using 2D phase-encoding and 1D frequency encoding using a MP-SSFP sequence [172] to refocus inhomogeneity with low peak RF amplitude. The resulting B_0 inhomogeneity was mapped by taking two measurements: once with the center of the frequency encoded readout aligned to the center of the spin echo, and once with the center of the readout gradient shifted 100 μs from the center of the spin echo. For analysis, axial slices of the 3D images were used, with in-plane field of view of 19.2 cm × 19.2 cm, 64 × 64 k-space points, and readout dwell time = 33.6 μs.

Using the measured B_0 inhomogeneity, we form the encoding matrix with phase errors, and subsequently generate distorted images for the first dataset. The surface of off-resonance frequencies is depicted in Fig. 5.3a. The inhomogeneity is made of harmonics up to the second order. Based on Table 5.1, there are eight harmonics up to the second order. The magnet isocenter is unknown. We combine harmonics that are identical except one constant coefficient. Therefore, there are 6 different aggregated harmonics, for which coefficients are to be estimated.

One coil is unitized to collect signal measurements in both datasets, and $\mathbf{s} = \mathbf{1}$ is considered for its sensitivity profile. The SNR of measurements in the second dataset is 15 dB, and the noise level in the first dataset is negligible. We depict one ground-truth image,

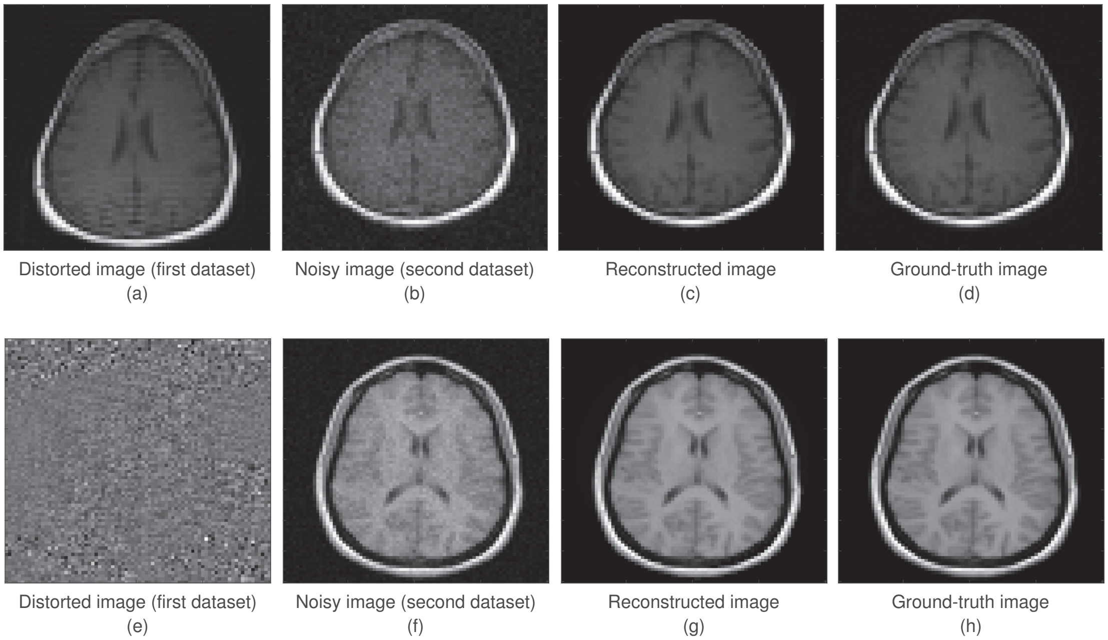
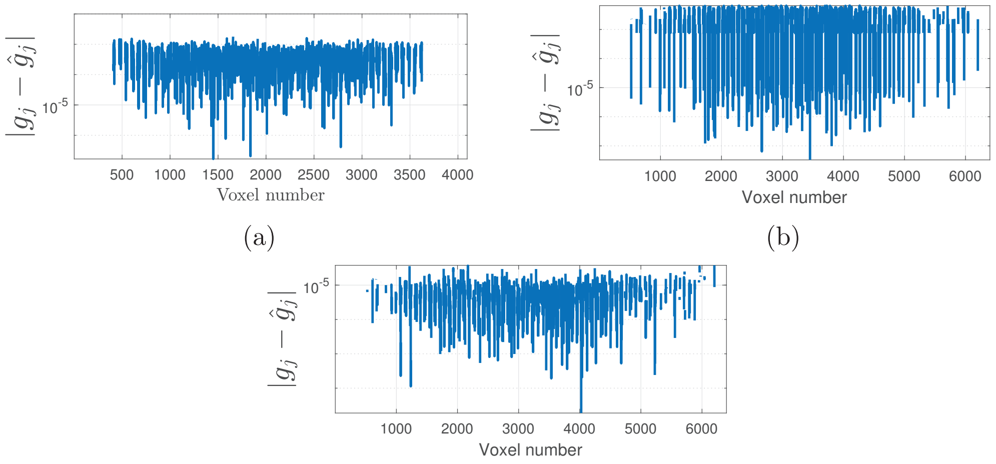

Distorted image (first dataset)
(a)
Noisy image (second dataset)
(b)
Reconstructed image
(c)
Ground-truth image
(d)
Distorted image (first dataset)
(e)
Noisy image (second dataset)
(f)
Reconstructed image
(g)
Ground-truth image
(h)

Figure 5.4: Two examples of leveraging the estimated magnetic field inhomogeneity to correct distorted images. Images in the top and bottom rows are of size 64×64 and 80×80, respectively. (a), (e) Two distorted images are depicted, where the surfaces of off-resonance frequencies are depicted in 5.3a and 5.3b, respectively. The utilized encoding methods for reconstructing (a), (b), and (f) are fully sampled multi-shot Cartesian trajectories. The single-interleaf spiral k-space acquisition in (e) is accelerated four times via 4x fewer turns (and also for all images in the first dataset). (b), (f) Two noisy images from the second datasets of two experiments. (c), (g) Reconstructed images using the estimated magnetic field. (d), (h) Ground-truth images for two experiments.

the corresponding distorted image, and the noisy image in Fig. 5.4d, Fig. 5.4a, and Fig. 5.4b, respectively. It is observed from Fig. 5.4a that the image is distorted along the $\mathtt{y}$ axis, which is the readout direction. In Fig. 5.5a, the accuracy of the estimated g_j in 4096 voxels is depicted. Due to the noisy measurements in the second dataset, g_j in a number of voxels is missing and cannot be estimated. The reason is that noise has eliminated the drop in cost functions for those voxels. Those voxels are mainly outside of the region in which an object lies. Using the approximated g_j, we estimate the coefficients of spherical harmonics via (5.28). Then, we use the estimated coefficients of spherical harmonics to approximate the surface of off-resonance frequencies. The value of the estimation error in each voxel is



(a)

(b)

(c)

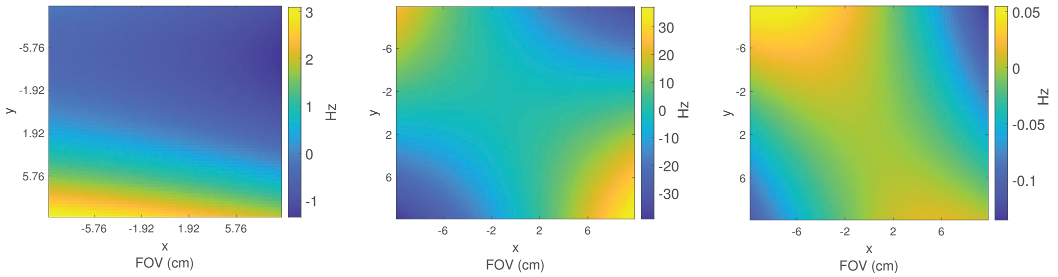
Figure 5.5: The accuracy of Algorithm 18 in optimizing g_j is depicted. (a) Considered images are of size 64×64, for which Cartesian encoding is used to collect k-space samples, and the surface of off-resonance frequencies is given in Fig. 5.3a. (b) Considered images are of size 80×80 with a single-interleaf spiral encoding, for which the surface of off-resonance frequencies is given in Fig. 5.3b. (c) The improved accuracy of optimized g_j variables after increasing the scale of optimization variables.

Table 5.2: Proportional error of estimated coefficients for spherical harmonics

Encoding	Proportional error $\|\frac{c_i - \hat{c}_i}{c_i}\|$					
	c_1	c_2	c_3	c_4	c_5	c_6
Cartesian	$1{\times}10^{-4}$	$3{\times}10^{-4}$	$16{\times}10^{-4}$	$28{\times}10^{-4}$	$4{\times}10^{-4}$	$3{\times}10^{-4}$
Spiral (a)		$7{\times}10^{-4}$	$76{\times}10^{-4}$	$71{\times}10^{-4}$	$5{\times}10^{-4}$	$22{\times}10^{-4}$
Spiral (b)		$0.0933{\times}10^{-4}$	$0.3374{\times}10^{-4}$	$0.1647{\times}10^{-4}$	$0.0263{\times}10^{-4}$	$0.0483{\times}10^{-4}$

depicted in Fig. 5.6a. The maximum error of estimation is less than 4 Hz.

Finally, we present the proportional errors of the estimated coefficients for spherical harmonics in the first row of Table 5.2. Despite the intensity of noise in the second dataset, the accuracy of the estimated $\mathbf{A}_f^s$ is enough to ensure that one can extract the undistorted image from the first dataset via solving (5.32). In this reconstructed, corrected image depicted in Fig. 5.4c, the estimated voxel values in the black region are mapped to zero. Estimating a ground-truth image from a distorted image is especially important when the noisy measurements in the second dataset are not available. We depict images reconstructed



(a) (b) (c)

Figure 5.6: Error of the estimated f_j, obtained from approximated spherical harmonic coefficients, is depicted. The accuracy for each estimated coefficient of one spherical harmonic is listed in Table 5.2. (a) Considered images are of size 64×64, for which a Cartesian encoding is used to collect k-space samples, and the surface of off-resonance frequencies is given in Fig. 5.3a. (b) Considered images are of size 80×80 with a single-interleaf spiral encoding, for which the surface of off-resonance frequencies is given in Fig. 5.3b. (c) Errors of estimated off-resonance frequencies after increasing the variable scales.

only using the estimated field map and distorted images from multiple slices with different contrasts when the second dataset is not available. We note that the accuracy of estimated $\mathbf{A}_f^s$ depends on the estimated off-resonance frequency accuracy. In a low SNR regime, a good accuracy for the coefficients of spherical harmonics can be obtained, although it may not be accurate enough to make $\mathbf{A}_f^s$ sufficiently accurate to be used effectively in (5.32).

We evaluate the performance of the proposed methods when the k-space information is encoded via a single interleaf spiral. The spiral trajectory is generated using the Michigan Image Reconstruction Toolbox (MIRT); here, k-space samples to reconstruct images in the first dataset are accelerated four times via 4x fewer turns. Eight coils are utilized to collect measurements. Coil sensitivity maps are calculated using the ESPIRiT approach [173]. For images in the second dataset, the k-space information is fully sampled using a multi-shot Cartesian acquisition. The sampling interval for the spiral trajectory is 4 microseconds. The field of view is $20\,\text{cm} \times 20\,\text{cm}$. We use $10,000$ images from NYU DICOM resized to 80×80 to simulate proposed methods. The echo time is 30 ms. The SNR level of measurements is 20 dB. The number of optimization variables in $\mathbf{\Lambda}$ and $\mathbf{X}$ involved in (5.19) are 40.96×10^6 and 64×10^6, respectively.

To generate the inhomogeneity, we assign random numbers to harmonics up to the second order. The distribution of off-resonance frequencies is depicted in Fig. 5.3b. The isocenter of the magnet is set to $(0,0,0)$, where the off-resonance frequency is zero. Therefore, the constant term in (5.26) is zero, and there are five coefficients for spherical harmonics to estimate. The maximum and minimum off-resonance frequencies are 21.352 kHz and -12.393 kHz, respectively. The noise level in the first dataset is negligible. It is observed from Fig. 5.4e that the image quality of the first dataset is seriously degraded by phase errors.[1] As the k-space sampling is accelerated 4 times, the number of rows of $\mathbf{A}_f^s$ is 1600. The accuracy of the estimated g_j is depicted in Fig. 5.5b.

To estimate elements of $\mathbf{A}_f^s$ for a single-shot trajectory, from one row to another, the estimated g_j is multiplied by a larger number each time in the exponent of the complex exponential term in $\boldsymbol{\Xi}$. Therefore, the residual error of estimated g_j is multiplied by a larger number from one row to another, and the accuracy of the estimated row degrades. Due to this reason, the estimated $\mathbf{A}_f^s$ in this experiment is not accurate enough to map distorted images to the ground-truth images using (5.32). As explained in Section 5.4.4, we increase the scale of optimization variables. For each voxel that belongs to $\mathcal{V}$, we predict the value of $\delta_j^{m_1+1} = 2\pi f_j T_E$, where $T_E = 30$ ms. The interval that we consider for the feasible set in (5.31) is $[2\pi\hat{u}_j^{m_1}T_E - \pi, 2\pi\hat{u}_j^{m_1}T_E + \pi]$. We depict the accuracy of each estimated g_j (calculated from $\hat{\delta}_j^{m_1+1}$) after increasing the scale of optimization variables in Fig. 5.5c. Using estimated $\{\delta_j^{m_1+1}\}_{j\in\mathcal{V}}$, we predict the coefficients of spherical harmonics. Based on those, the surface of off-resonance frequencies is reconstructed. The residual error of estimations is depicted in Fig. 5.6c. Compared to 5.6b, which depicts the accuracy of the reconstructed surface before increasing the scale of optimization variables, the accuracy of estimated surface of off-resonance frequencies improves after increasing the scale of variables. The proportional errors of the estimated coefficients for spherical harmonics before and after increasing the scale of optimization variables are listed in Table 5.2.

We observe from Fig. 5.4g that the accuracy of $\mathbf{A}_f^s$ is enough to correct distorted images in the first dataset using (5.32). We show that the estimated field map can be used to correct distorted images, which correspond to different slices and contrasts, without any

[1] It is worth noting, for contrast, that when the image quality is significantly degraded as shown in Fig. 5.4e, the artifact reduction performance of CNN-based methods diminishes, since the similarities between the ground-truth and distorted images have vanished.

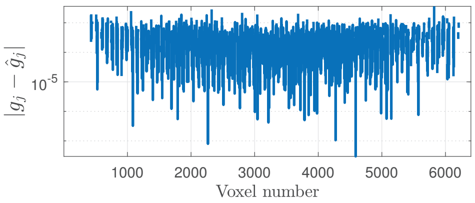


Figure 5.7: The accuracy of estimated g_j returned by Algorithm 2 is depicted. Considered images are of size 80×80 with a 10-interleaf spiral encoding. The SNR of measurements in the second dataset is 23 dB.

Table 5.3: Proportional error of estimated coefficients for spherical harmonics

Encoding	Proportional error $\lvert\frac{c_i-\hat{c}_i}{c_i}\rvert$					
	c_1	c_2	c_3	c_4	c_5	c_6
Spiral		$8.506{\times}10^{-5}$	$2.482{\times}10^{-4}$	$1.382{\times}10^{-4}$	$2.168{\times}10^{-6}$	$2.069{\times}10^{-5}$

guidance from the second dataset. The flexibility of the proposed algorithm in term of processing images from different slices and contrasts makes it easier to construct a dataset, which is important to ensure practicality of the method.

We do one additional experiment when a spiral encoding with 10 interleaves, generated by MIRT, is used in the first dataset to encode the k-space information. In this experiment, the echo time for each shot is 1.975 ms and the ground truth surface of off-resonance frequencies is shown in Fig. 5.3b. We set the isocenter of the magnet to $(0, 0, 0)$, where the off-resonance frequency is zero. For images in the second dataset, the k-space information is fully sampled using a multi-shot Cartesian acquisition. The field of view is $20\,\text{cm} \times 20\,\text{cm}$. We use $10,000$ images from NYU DICOM resized to 80×80 to simulate proposed methods. The SNR of measurements in the second dataset is 23 dB. We accelerate k-space sample collection for images in the first dataset four times. Eight coils are unitized to collect signal measurements. We depict the accuracy of estimated g_j in Fig. 5.7. The residual error of off-resonance frequency estimation is depicted in Fig. 5.9. From Fig. 5.8, we observe that the accuracy of the estimated $\mathbf{A}_f^s$ is enough to correct the distorted image in the first dataset via solving (5.32). Proportional errors of estimated coefficients for spherical harmonics are listed in Table 5.3.

For the three discussed encoding approaches used to reconstructed images in the first

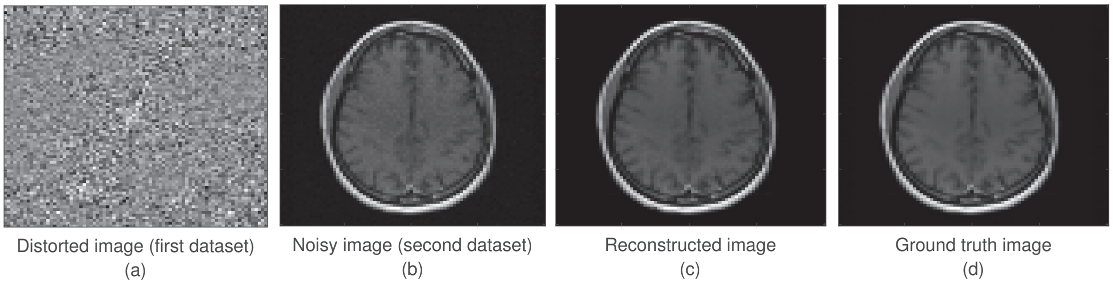


Figure 5.8: An example of leveraging the estimated magnetic field to correct distorted images. Images are all of size 80 × 80. (a) One distorted image is depicted, where the encoding used to generate the image is a spiral with 10 interleaves. The k-space acquisition is accelerated four times for images in the first datasets. The surface of off-resonance frequencies is depicted in Fig. 5.3. (b) One noisy image in the second dataset, for which a multi-shot Cartesian encoding is used. The conducted measurements for the first dataset are noise-free and the SNR level for measurements in the second dataset is 23 dB. The k-space is fully sampled for images in the second dataset. (c) The reconstructed image using the estimated magnetic field. (d) One ground-truth image.

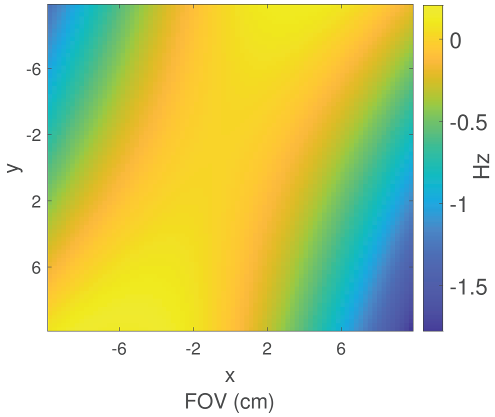


Figure 5.9: The residual error of f_j estimation, obtained from approximated spherical harmonic coefficients, is depicted. Considered images in the first dataset are of size 80×80 with a ten-interleaf spiral encoding, for which the surface of off-resonance frequencies is the surface depicted in Fig. 5.3b.

dataset, we present more images to better show distorted and reconstructed images, from multiple slices with different contrasts, without any guidance from the second dataset. For each encoding scheme, we solve the field map estimation problem and use the approximated field map to correct a number of distorted images without having any guidance from the second dataset. The first row of Figs. 5.10, 5.11, and 5.12 depict ground-truth images. The second row of figures represent distorted images.

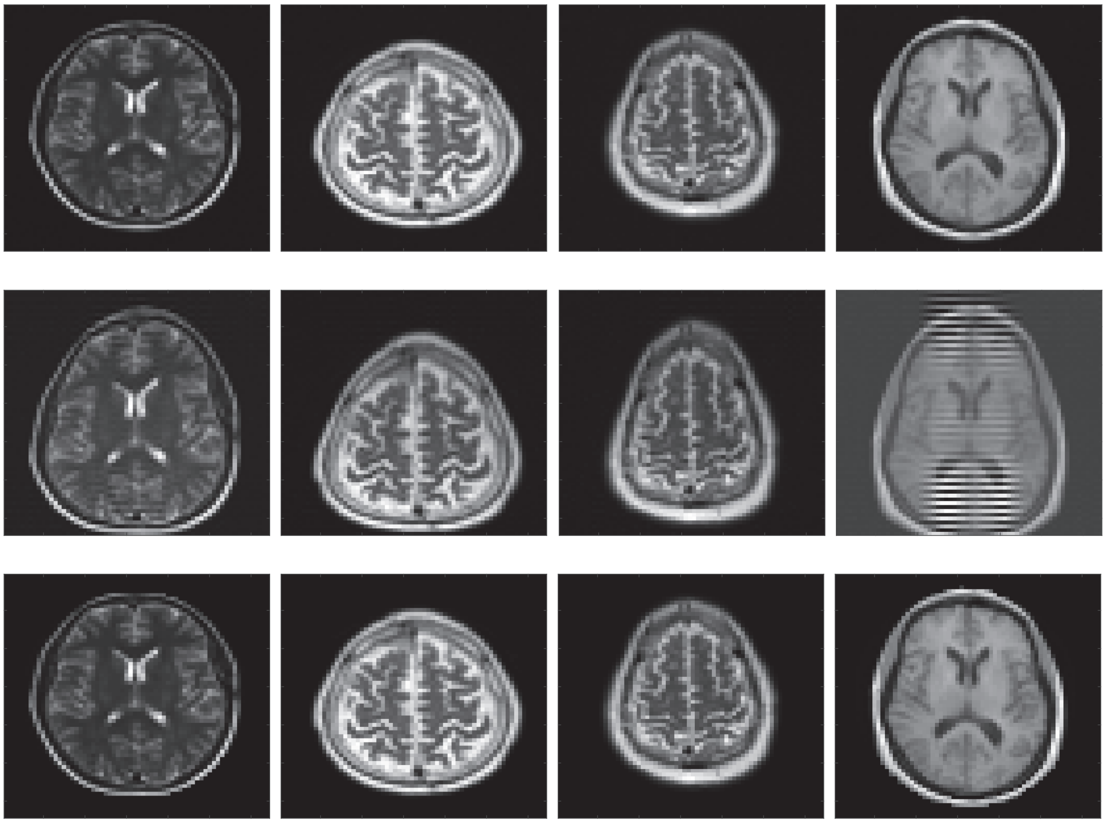


Figure 5.10: Examples of using the field map to correct distorted images. The used readout trajectory for images in the first dataset is multi-shot Cartesian, respectively. Top row represents ground-truth images. The middle row depicts distorted images. The last row shows corrected images using the middle row images and the estimated field maps.

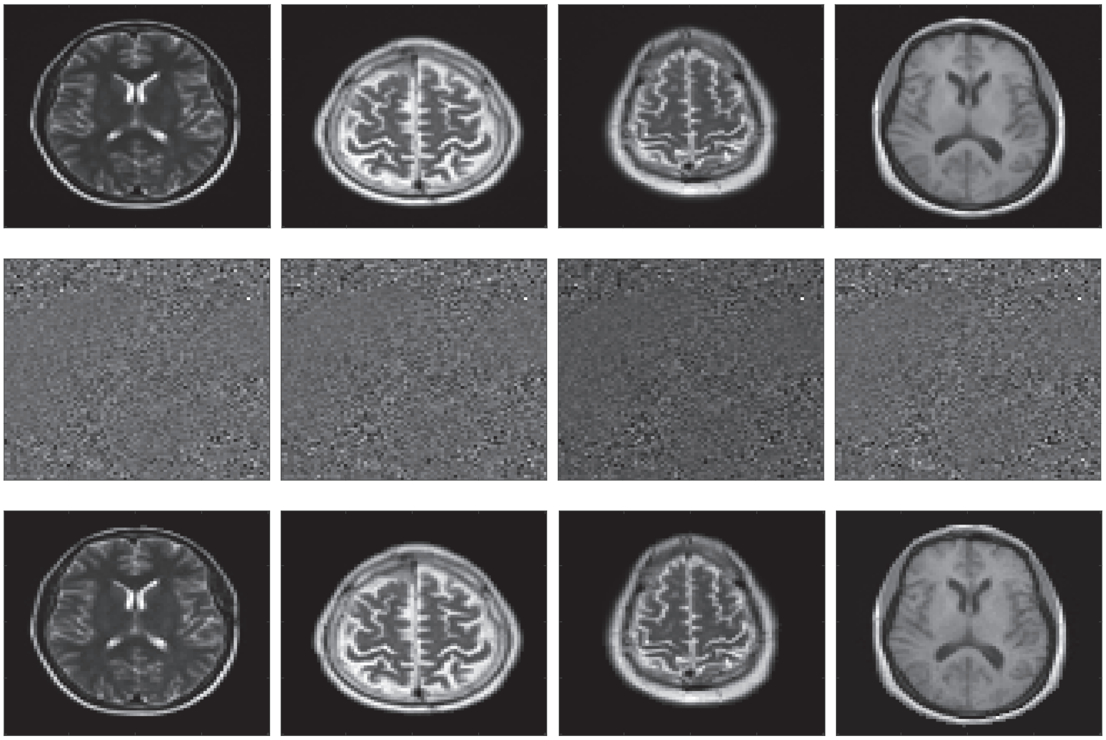


Figure 5.11: Examples of using the field map to correct distorted images. The used readout trajectory for images in the first dataset is single-interleaf spiral, respectively. Top row represents ground-truth images. The middle row depicts distorted images. The last row shows corrected images using the middle row images and the estimated field maps.

We evaluate the performance of Algorithm 17 to solve the joint optimization (5.19). After optimizing $\mathbf{X}$, we estimate $\boldsymbol{\Theta}$ from (5.23). The disjoint optimization is that the images are reconstructed from noisy measurements available in the second dataset, then $\boldsymbol{\Theta}$ is approximated from (5.23). The performance of Algorithm 17 and the disjoint optimization in terms of estimating $\mathbf{X}$ and $\boldsymbol{\Theta}$ are given in Table 5.4 and Table 5.5, respectively. In each experiment, we consider that the k-space subsampling to reconstruct images in the first dataset is either 4x or 8x, when one (or ten)-interleave spiral is used to encode the k-space information, and the additive noise is negligible. The SNR level of fully sampled measurements in the second dataset is listed for each experiment. We observe that the

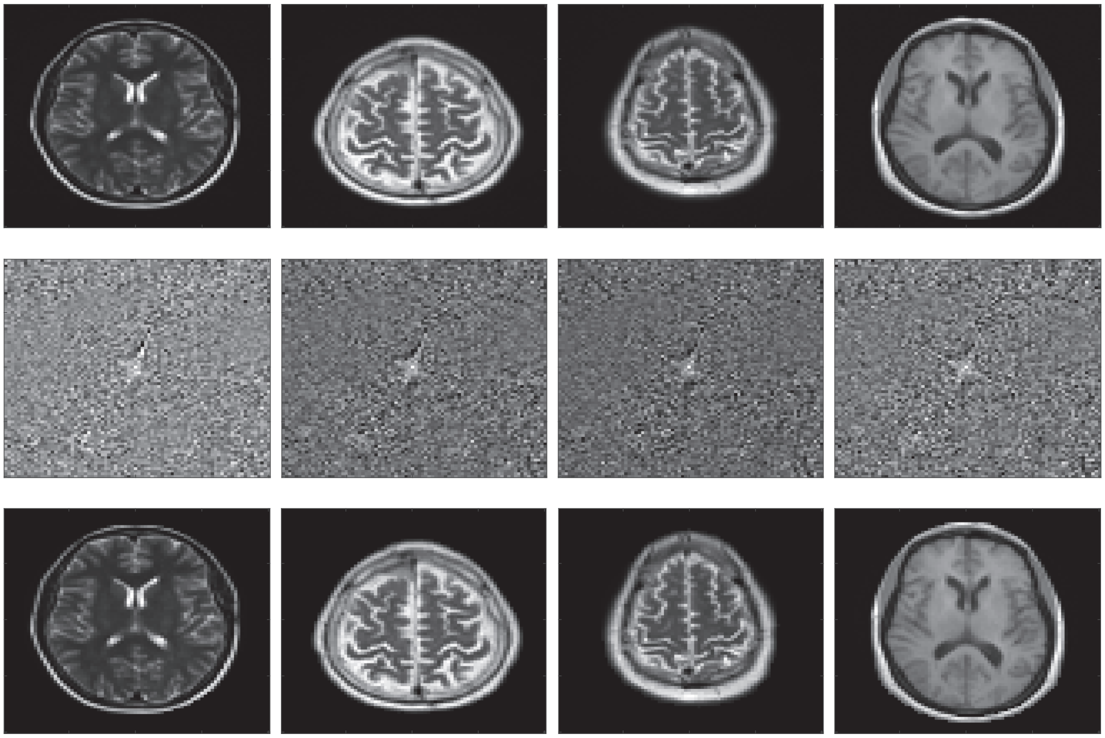


Figure 5.12: Examples of using the field map to correct distorted images. The used readout trajectory for images in the first dataset is ten-interleaf spiral, respectively. Top row represents ground-truth images. The middle row depicts distorted images. The last row shows corrected images using the middle row images and the estimated field maps.

proportional errors in estimation of both $\mathbf{X}$ and $\boldsymbol{\Theta}$ are smaller when Algorithm 17 is used.

5.6 Concluding Remarks

This chapter studied joint image and field map estimation using two datasets of a common set of objects. The first dataset includes 2D images, all of which are distorted by the same inhomogeneity of the magnetic field. The second dataset consists of noisy, undistorted measurements for the same 2D slices considered in the first dataset. We proposed a parallel, distributed methods to solve a non-convex optimization to estimate the off-resonance frequency in each voxel. Estimated 2D off-resonance frequencies are regularized through spherical harmonics that represent the surface of off-resonance frequencies.

One of the major contributions of our proposed framework is that it achieves a reasonable accuracy in field map and image correction in the presence of large magnetic field inhomogeneity using noisy reference measurements. As alluded above, this becomes increasingly important in a number of desirable scenarios, including rapid imaging (where readout times are long), with a compact magnet (e.g., a head-only magnet), where field inhomogeneities are large. That our approach makes progress in accurately imaging in these challenging scenarios, using datasets that are qualitatively similar to those used in some of the aforementioned works, is to us a promising potential approach.

Table 5.4: The performance of Algorithm 17 to solve the joint optimization.

SNR	Single interleaf spiral			
	4x acceleration		8x acceleration	
	$\frac{\|\boldsymbol{\Theta}-\hat{\boldsymbol{\Theta}}\|_F^2}{\|\boldsymbol{\Theta}\|_F^2}$	$\frac{\|\mathbf{X}-\hat{\mathbf{X}}\|_F^2}{\|\mathbf{X}\|_F^2}$	$\frac{\|\boldsymbol{\Theta}-\hat{\boldsymbol{\Theta}}\|_F^2}{\|\boldsymbol{\Theta}\|_F^2}$	$\frac{\|\mathbf{X}-\hat{\mathbf{X}}\|_F^2}{\|\mathbf{X}\|_F^2}$
15 dB	9.080×10^{7}	1.250×10^{-2}	3.054×10^{7}	1.250×10^{-2}
20 dB	2.581×10^{7}	4×10^{-3}	9.250×10^{6}	4×10^{-3}
25 dB	1.304×10^{7}	1.3×10^{-3}	5.570×10^{6}	1.2×10^{-3}
30 dB	7.024×10^{6}	3.9536×10^{-4}	2.167×10^{6}	3.9517×10^{-4}
35 dB	2.922×10^{6}	1.2504×10^{-4}	7.633×10^{5}	1.2502×10^{-4}
40 dB	9.357×10^{5}	3.953×10^{-5}	4.301×10^{5}	3.955×10^{-5}
	10-interleaf spiral			
SNR	4x acceleration		8x acceleration	
	$\frac{\|\boldsymbol{\Theta}-\hat{\boldsymbol{\Theta}}\|_F^2}{\|\boldsymbol{\Theta}\|_F^2}$	$\frac{\|\mathbf{X}-\hat{\mathbf{X}}\|_F^2}{\|\mathbf{X}\|_F^2}$	$\frac{\|\boldsymbol{\Theta}-\hat{\boldsymbol{\Theta}}\|_F^2}{\|\boldsymbol{\Theta}\|_F^2}$	$\frac{\|\mathbf{X}-\hat{\mathbf{X}}\|_F^2}{\|\mathbf{X}\|_F^2}$
15 dB	5.545×10^{7}	1.247×10^{-4}	1.590×10^{8}	1.253×10^{-2}
20 dB	2.144×10^{7}	4×10^{-3}	4.695×10^{7}	4.01×10^{-3}
25 dB	4.062×10^{6}	1.303×10^{-3}	1.650×10^{7}	1.20×10^{-3}
30 dB	2.192×10^{6}	3.9515×10^{-4}	4.926×10^{6}	3.9514×10^{-4}
35 dB	1.065×10^{6}	1.2502×10^{-4}	3.482×10^{6}	1.2505×10^{-4}
40 dB	4.718×10^{5}	3.955×10^{-5}	7.656×10^{5}	3.953×10^{-5}

Table 5.5: The performance of the disjoint optimization.

SNR	Single interleaf spiral			
	4x acceleration		8x acceleration	
	$\frac{\|\boldsymbol{\Theta}-\hat{\boldsymbol{\Theta}}\|_F^2}{\|\boldsymbol{\Theta}\|_F^2}$	$\frac{\|\mathbf{X}-\hat{\mathbf{X}}\|_F^2}{\|\mathbf{X}\|_F^2}$	$\frac{\|\boldsymbol{\Theta}-\hat{\boldsymbol{\Theta}}\|_F^2}{\|\boldsymbol{\Theta}\|_F^2}$	$\frac{\|\mathbf{X}-\hat{\mathbf{X}}\|_F^2}{\|\mathbf{X}\|_F^2}$
15 dB	1.003×10^{13}	1.58×10^{-2}	1.112×10^{13}	1.58×10^{-2}
20 dB	9.110×10^{12}	5×10^{-3}	9.739×10^{12}	5×10^{-3}
25 dB	5.687×10^{12}	1.60×10^{-3}	5.937×10^{12}	1.60×10^{-3}
30 dB	2.267×10^{12}	4.9969×10^{-4}	2.412×10^{12}	4.9988×10^{-4}
35 dB	8.662×10^{11}	1.5811×10^{-4}	9.121×10^{11}	1.5809×10^{-4}
40 dB	3.000×10^{11}	4.9988×10^{-5}	3.513×10^{11}	5.0007×10^{-5}
	10-interleaf spiral			
SNR	4x acceleration		8x acceleration	
	$\frac{\|\boldsymbol{\Theta}-\hat{\boldsymbol{\Theta}}\|_F^2}{\|\boldsymbol{\Theta}\|_F^2}$	$\frac{\|\mathbf{X}-\hat{\mathbf{X}}\|_F^2}{\|\mathbf{X}\|_F^2}$	$\frac{\|\boldsymbol{\Theta}-\hat{\boldsymbol{\Theta}}\|_F^2}{\|\boldsymbol{\Theta}\|_F^2}$	$\frac{\|\mathbf{X}-\hat{\mathbf{X}}\|_F^2}{\|\mathbf{X}\|_F^2}$
15 dB	9.855×10^{12}	1.58×10^{-2}	1.208×10^{13}	1.580×10^{-2}
20 dB	9.564×10^{12}	5×10^{-3}	1.094×10^{13}	5×10^{-3}
25 dB	6.836×10^{12}	1.6×10^{-3}	5.149×10^{12}	1.60×10^{-3}
30 dB	2.366×10^{12}	4.9988×10^{-4}	2.309×10^{12}	4.9969×10^{-4}
35 dB	8.982×10^{11}	1.5809×10^{-4}	9.981×10^{11}	1.5807×10^{-4}
40 dB	3.348×10^{11}	4.998×10^{-5}	3.144×10^{11}	4.998×10^{-5}

Chapter 6

Regularized Joint Image and Field Map Estimation in 3D Multi-Echo MRI

EPI and spiral samplings are widely used in MRI due to their favorable properties. However, they are sensitive to magnetic field inhomogeneity. Inhomogeneity of the magnetic field generates spatially distributed off-resonance frequencies, which result in images with blur and geometric distortion. EPI and spiral sampling methods are both sensitive to magnetic field inhomogeneity and can generate images with artifacts as a consequence of induced phase errors. To correct these images, field map information is necessary.

6.1 Prior Work

Different approaches have been proposed to estimate the field map depending on the scanner properties. One common technique is acquiring multiple scans. To enhance the SNR level of measurements, the time interval among measurements should be increased. However, increasing the time interval results in phase wrapping, which complicates the estimation problem. The authors of [160] study joint image and field map estimation using multiple echo times and a single coil. A gradient method with diagonal scaling is proposed in [160], where first-order and second-order difference regularizers are used to promote smooth solutions.

Water-fat imaging using multiple echoes in the presence of magnetic field inhomogeneity is studied in [1, 174, 161, 175, 176]. Unlike [160], the exponential signal decay is not considered in [1, 174, 161, 175, 176]. In [1], a gradient method for joint water-fat and field map estimation is proposed. The conjugate gradient approach given in [1] deploys an iteration-dependent preconditioner based on an incomplete Cholesky factorization in order to improve computational and storage efficiency. A similar problem is studied in [174], where a region growing approach is used to initialize phase error estimations. In the proposed region growing approach, the approximated phase errors in a small set of voxels are used to correct the initial phase errors in neighboring voxels, then gradient methods are used to iteratively optimize phase errors. One potential drawback of the region growing method is that errors may be propagated if phases are wrongly estimated for the initial set of voxels. The proposed method in [161] starts with low resolution images and utilizes a golden-section search to directly locate possible field map values. It gradually increases the resolution of images and refines and propagates field map estimates to increasingly finer resolutions until the full-resolution field map is obtained. A grid-search approach is proposed in [175], where optimization variables are off-resonance frequencies rather than phase errors. Using prior knowledge about the field map, a grid-search is implemented to find the global minimizer. Although directly optimizing off-resonance frequencies resolves the phase wrapping ambiguity problem, it makes solving the problem harder by increasing the search interval, which slows down the proposed approach in practice. The proposed method in [176] quantizes the formulated optimization in [175], aiming at limiting the feasible set for an exhaustive search. However, the accuracy of returned solutions directly depend on the number of quantized points.

In [177], image values, phase errors, and R_2^* are jointly estimated from measurements collected from multi-echo EPI using an ADMM method. The ADMM subproblems are non-convex and solved approximately. Therefore, [177] is not guaranteed to converge to a stationary solution. In [178], a deep learning network for fast water–fat separation is proposed with different convolutional layers and skip connections. Convolutional layers are utilized in [178] to extract differences and similarities between two consequential echo times. In addition, multiple convolutional layers and skip connections are utilized to separate water and fat. The drawback of [178] is that the proposed method can handle relatively small inhomogeneity.

6.2 Contributions

We propose a BCD algorithm for the joint optimization problem of 3D field map and voxel intensities from multiple 2D slices. In the proposed BCD approach, one block of variables determines voxel intensities and the other block of variables specifies phase errors in the same voxels. The problem with respect to the voxel intensity is strongly convex, and we update voxel intensities using a closed-form solution. The problem with respect to phase errors is neither convex nor concave, and we update wrapped phase errors in different voxels via a golden-section search nested inside a grid-search, which can obtain the global minimizer for a phase error in each iteration. Since the SNR level degrades over time, we start with a few initial echo times that have high enough SNRs. We start from voxels near the magnet center where the wrapped phase equals the unwrapped phase. Phase error estimates of multiple 2D slices derived from a few initial echo times are projected to the subspace spanned by 3D spherical harmonics up to the third order that model inhomogeneity, aiming at efficiently regularizing phase errors. We alternately optimize the two blocks of variables in the BCD algorithm until both blocks converge. After convergence, we use the estimated coefficients for spherical harmonics to predict the 3D magnetic field inhomogeneity and the surface of off-resonance frequencies for each 2D slice, which can be used to approximate unwrapped phase errors in all voxels.

We gradually expand the size of the considered voxel subset and incorporate neighboring positions. We alternate between solving with the BCD approach and expanding the considered subset of voxels until we process all image voxels. After we process all image voxels, we increase the number of considered echo times. In order to assure that the estimates from latter echo times do not hurt the accuracy of estimates from the initial echoes, we separately solve the joint optimization for each latter echo. Then, we compare the obtained phase error for one voxel from a per-echo problem with the estimated value derived from the few initial echo times. If the difference is below a certain threshold, we keep the measurement for that voxel. Otherwise, we consider that measurement as a toxic sample and discard it. Once we filter latter echo times, we combine measurements from all echoes and jointly estimate voxel intensities and phase errors.

We leverage the proposed scheme to jointly estimate voxel intensities and phase errors when the magnetic field is time variant. In this case, we sequentially and separately solve the joint optimizations for different echo times and benefit from collected measurements from different coils. When we process one echo time, we use the estimated phase errors from the previous echoes and keep them fixed. We only optimize the phase errors added during one echo time rather than the aggregated phase error. This allows the proposed approach to handle a large magnetic field inhomogeneity by assuring that there exists a subset of voxels near the magnet center, for which the wrapped and unwrapped phases are equal. When the time-varying magnetic field is generated by eddy currents, one can assume that the induced angular frequencies in odd echoes are identical, and the same is true for even echoes. When eddy currents make the magnetic field time variant, we sequentially solve joint optimizations for all echoes. Next, we group measurements for even echo times and solve the joint optimization for them. We solve a similar problem for measurements of odd echo times as well. We demonstrate the superiority of the proposed approach against a state-of-the-art papers [1] through numerical simulations.

The rest of this chapter is organized as follows. The problem formulation is given in Section 6.3. Section 6.4 describes a distributed BCD optimization algorithm for the joint image and voxel optimization in the presence of a large static inhomogeneous magnetic field. In Section 6.5, we extend the proposed BCD algorithm for time-varying inhomogeneous magnetic fields. The simulation results are given in Section 6.6, and concluding remarks are given in Section 6.7.

6.3 Problem Formulation

Consider that the signal from the j^{th} voxel of the k^{th} object received by the c^{th} coil during the i^{th} scan is denoted by $y_{cj}^{k}(t_i)$ and is calculated as follows:

$$y_{cj}^{k}(t_i) = e^{\iota\omega_j^k t_i - R_j^k t_i} s_c^k(\mathbf{r}_j) x_j^k + \epsilon_{cj}^{k}(t_i), \tag{6.1}$$

where $\epsilon_{cj}^{k}(t_i)$ is the additive noise and $\omega_j^k t_i$ is the phase error in position j due to the magnetic field inhomogeneity. Additionally, x_j^k is the j^{th} voxel of the $k^{\text{th}} \in \{1, \ldots, K\}$

ground-truth image, and $s_c^k(\mathbf{r}_j)$ is the sensitivity of the c^{th} coil in the j^{th} position of the k^{th} image. In (6.1), R_j^k denotes the R_2^* value of for the j^{th} voxel.

Since the additive noise is zero-mean complex Gaussian, one can use the maximum likelihood estimator to jointly optimize images and the field map as follows:

$$\min_{\boldsymbol{\omega}^k, \mathbf{x}_k} \quad \sum_{i=1}^{I} \sum_{c=1}^{C} \sum_{j \in \mathcal{V}_k} \left| y_{cj}^k(t_i) - e^{\iota \omega_j^k t_i - R_j^k t_i} s_c^k(\mathbf{r}_j) x_j^k \right|^2, \tag{6.2}$$

where $\mathcal{V}_k$ denotes the set of image k voxels. Since the phase error is generated due to the inhomogeneous magnetic field, we have the following equation for each position:

$$\omega_j^k = 2\pi f_j^k, \tag{6.3}$$

where f_j^k is the spatially dependent off-resonance frequency present in the j^{th} position. The off-resonance frequency in one position can be written as a superposition of three components as follows:

$$f_j^k = f_j^m + f_j^{k,s} + f_j^{k,r},$$

where f_j^m, $f_j^{k,s}$, and $f_j^{k,r}$ are off-resonance frequencies generated from the magnet, the object susceptibility map, and the object out of the field on view.

Based on the magnet design of the scanner, the inhomogeneity of the main magnetic field can be described by a spherical harmonic expansion:

$$\Delta B_0(\mathbf{r}_j) = \sum_{\mathtt{m}=1}^{u} \sum_{\mathtt{n}=1}^{u} c_{\mathtt{m}}^{\mathtt{n}} \Psi_{\mathtt{m}}^{\mathtt{n}}(\mathtt{x}, \mathtt{y}, \mathtt{z}, \mathbf{o}), \tag{6.4}$$

where the spherical harmonic orthogonal function $\Psi_{\mathtt{m}}^{\mathtt{n}}(\mathtt{x}, \mathtt{y}, \mathtt{z}, \mathbf{o})$ is of order $\mathtt{n}$ and degree $\mathtt{m}$. Moreover, $c_{\mathtt{m}}^{\mathtt{n}}$ is the corresponding coefficient. The maximum order is u. In the above equation, $\mathbf{o}$ represents the magnet center and $\mathbf{o} = (\mathtt{o}_1, \mathtt{o}_2, \mathtt{o}_3)$. The list of sixteen 3D spherical harmonics up to the third order are listed in Table 6.1. For a stable time-invariant magnetic field, coefficients do not change from one echo to another. Typically, coefficients are determined by the scanner and its calibration. The object-induced magnetic field can be calculated using the susceptibility map from [179, eq. 4], where the susceptibility map

Table 6.1: Spherical harmonic functions in Cartesian coordinates

n	m	Cartesian harmonic
0	0	1
1	0	$z - o_3$
1	1	$x - o_1$
1	1	$y - o_2$
2	0	$(z - o_3)^2 - \left((x - o_1)^2 + (y - o_2)^2\right)/2$
2	1	$(x - o_1)(z - o_3)$
2	1	$(y - o_2)(z - o_3)$
2	2	$(x - o_1)^2 - (y - o_2)^2$
2	2	$2(x - o_1)(y - o_2)$
3	0	$(z - o_3)\left((z - o_3)^2 - 3((x - o_1)^2 + (y - o_2)^2)/2\right)$
3	1	$(x - o_1)\left(4(z - o_3)^2 - (x - o_1)^2 - (y - o_2)^2\right)$
3	1	$(y - o_2)\left(4(z - o_3)^2 - (x - o_1)^2 - (y - o_2)^2\right)$
3	2	$(z - o_3)\left((x - o_1)^2 - (y - o_2)^2\right)$
3	2	$2(x - o_1)(y - o_2)(z - o_3)$
3	3	$(x - o_1)^3 - 3(x - o_1)(y - o_2)^2$
3	3	$3(x - o_1)^2(y - o_2) - (y - o_2)^3$

can be accurately measured as explained in [180]. Off-resonance frequencies induced by the object can be calculated from the induced field via [179, eq. 2], which will be discussed later. The magnetic field from the object out of the FOV can be considered linearly variant [179], and therefore can be represented via spherical harmonics.

A more complicated formulation addresses the scenario where the magnetic field is time-varying. These types of magnetic fields are referred to as higher order field perturbations. One of the factors that can generate higher order field perturbations is the eddy current. The other cause is dynamically operated shim systems with imperfect driver hardware. When the magnetic field inhomogeneity is time-variant, the total additive phase error in each scan is different. Without precise information array about the magnetic field, one cannot relate the phase errors in different echo times. Equation (6.1) changes to the following equation when the magnetic field is time-variant:

$$y_{cj}^{k}(t_i) = e^{\iota\omega_j^k(t_i)t_i - R_j^k t_i} s_c^k(\mathbf{r}_j) x_j^k + \epsilon_{cj}^k(t_i). \tag{6.5}$$

In order to collect sufficient measurements to formulate the maximum likelihood estimation, we consider that multiple coils are available. We formulate the following optimization to jointly estimate images and phase errors for different echo times:

$$\min_{\mathbf{x}_k, \{\boldsymbol{\omega}^k(t_i)\}_{i=1}^{I}} \quad \sum_{i=1}^{I} \sum_{c=1}^{C} \sum_{j \in \mathcal{V}_k} \left| y_{cj}^k(t_i) - e^{\iota \omega_j^k(t_i) t_i - R_j^k t_i} s_c^k(\mathbf{r}_j) x_j^k \right|^2. \tag{6.6}$$

To solve (6.2) and (6.6), one can use the fact that the induced field from the magnet is smooth. To promote smooth solutions for the magnet induced field, one can use low-pass filters that limit abrupt changes in the estimated magnetic field [181]. The first and second order finite difference regularizers are used in [1, 160] to enhance the smoothness of the solution.

After compensating for the object induced inhomogeneity using susceptibility maps, the inhomogeneous magnetic field can be modeled as a superposition of spherical harmonic bases like (6.4). Imposing this assumption not only does promote smooth solutions, but also restricts the feasible set of solutions. We separately study the estimation problem of different magnetic field types.

6.4 Static Inhomogeneous Magnetic Field

Former works, e.g., [1, 160], remove the $\mathbf{x}_k$ block and optimize a cost function of $\boldsymbol{\omega}$ via gradient-type methods. One common limitation of the above papers is that it is assumed that $|\omega_j \Delta t_i| < \pi$. This assumption prevents the existing approaches from correcting large magnetic field inhomogeneity.

To address the above limitations, we first propose an approach to solve (6.2) when $|\omega_j^k \Delta t_i| < \pi$. Next, we relax this constraint and generalize our approach. We propose a BCD method to alternatively solve (6.2) with respect to $\mathbf{x}_k$ and $\boldsymbol{\omega}^k$. We regularize the estimated field map in each iteration via a projection step. The advantage of the proposed BCD method is that it enables the decomposition of the problem across voxels, and it optimizes the phase error in each voxel separately. Since the received signal is exponentially decaying, we leverage the first Υ echo times in the initial stage to generate estimations for images and the magnetic field. Then, we leverage the rest of echoes after being filtered, based on the first estimation, to expand the number of utilized measurements. The estimated susceptibility map is leveraged to approximate the object induced magnetic field and later used to correct generated phase errors.

6.4.1 Incremental Processing of Signals from the Initial Echo Times

Consider that the field induced from the the susceptibility map in each voxel is measured accurately and denoted by $\hat{f}_j^{k,s}$. Let us denote $2\pi(f_j^m + f_j^{k,r})\Delta t_1$ by g_j, where $\Delta t_1 = t_1$ and $\Delta t_i = t_i - t_{i-1}$ for $i \geq 2$. We initialize $\mathbf{g}$ as follows:

$$g_j^{0,k} = \frac{\Delta t_1}{(\Upsilon - 1)C} \sum_{c=1}^{C} \sum_{i=2}^{\Upsilon} \frac{\log(\frac{y_{cj}^k(t_i)}{e^{(\iota 2\pi \hat{f}_j^{k,s} - R_j^k)\Delta t_i} y_{cj}^k(t_{i-1})})}{\iota(t_i - t_{i-1})}. \tag{6.7}$$

We obtain the minimizer of (6.2) with respect to x_j^k from the following equations:

$$\sum_{i=1}^{\Upsilon} \sum_{c=1}^{C} e^{-R_j^k t_i} s_c^{k*}(\mathbf{r}_j) \Big(e^{-\iota g_j^k (1 + \frac{\sum_{a=2}^{i} \Delta t_a}{\Delta t_1})} y_{cj}^k(t_i) e^{-\iota 2\pi \hat{f}_j^{k,s} t_i} - e^{-R_j^k t_i} s_c^k(\mathbf{r}_j) x_j \Big) = 0$$

$$\to x_j^k = \frac{\sum_{i=1}^{\Upsilon} \sum_{c=1}^{C} s_c^{k*}(\mathbf{r}_j) e^{-\iota g_j^k (1 + \frac{\sum_{a=2}^{i} \Delta t_a}{\Delta t_1}) - R_j^k t_i} y_{cj}^k(t_i) e^{-\iota 2\pi \hat{f}_j^{k,s} t_i}}{\sum_{i=1}^{\Upsilon} \sum_{c=1}^{C} e^{-2R_j^k t_i} |s_c^k(\mathbf{r}_j)|^2}. \tag{6.8}$$

Given that the inhomogeneity of the magnetic field is limited, we use (6.7) and obtain a subset of voxels, denoted by $\mathcal{S}^{1,k}$, where the unwrapped phase is equal to the wrapped phase. Mathematically, we have

$$\mathcal{S}^{1,k} = \Big\{ j | \omega_j^k \Delta t_1 < \pi \Big\}.$$

First, we consider positions for which $|\omega_j^k \Delta t_i| < \pi$. One can run a grid-search in the interval $[-\pi, \pi]$ to numerically obtain the global minimizer. In the t^{th} iteration of the BCD Algorithm, we update the two blocks as follows:

$$x_j^{q+1,k} = \frac{\sum_{i=1}^{\Upsilon} \sum_{c=1}^{C} s_c^{k*}(\mathbf{r}_j) e^{-\iota g_j^{q,k} (1 + \frac{\sum_{a=2}^{i} \Delta t_a}{\Delta t_1}) - R_j^k t_i} y_{cj}^k(t_i) e^{-\iota 2\pi \hat{f}_j^{k,s} t_i}}{\sum_{i=1}^{\Upsilon} \sum_{c=1}^{C} e^{-2R_j^k t_i} |s_c^k(\mathbf{r}_j)|^2}, \tag{6.9}$$

$$\tilde{g}_j^k = \arg\min_{g_j^k} \quad \sum_{c=1}^{C}\sum_{i=1}^{\Upsilon}\left| y_{cj}^k(t_i) e^{-\iota g_j^k (1+\frac{\sum_{a=2}^{i}\Delta t_a}{\Delta t_1})} e^{-\iota 2\pi \hat{f}_j^{k,s} t_i} - e^{-R_j^k t_i} s_c^k(\mathbf{r}_j) x_j^{q+1,k} \right|^2. \tag{6.10}$$

We know that after removing phase errors generated from susceptibility maps, the remaining phase errors can be obtained from a magnetic field expandable via spherical harmonics. We project the iterate $\tilde{g}_j^k$ to (6.4). We can rewrite (6.4) as follows:

$$\gamma \mathbf{E}\mathbf{c}^k = \frac{\mathbf{g}^k}{\Delta t_1}, \tag{6.11}$$

where $\mathbf{E} = \{\mathbf{e}_j\}_{j=1}^n$ and $\mathbf{e}_j = \{\{\Psi_{\mathtt{m}}^{\mathtt{n}}(\mathtt{x}_j, \mathtt{y}_j, \mathtt{z}_j, \mathbf{o})\}_{\mathtt{m}=1}^u\}_{\mathtt{n}=1}^u$. The projection step to (6.11) is as follows:

$$\mathbf{g}^{q+1,k} = \min_{\mathbf{g}^k \in \mathcal{R}^n, \mathbf{c}^k \in \mathcal{R}^{16}} \quad \left\| \mathbf{g}^k - \tilde{\mathbf{g}}^k \right\|_2^2 \quad \text{s.t.} \quad \gamma \mathbf{E}\mathbf{c}^k = \frac{\mathbf{g}^k}{\Delta t_1}, \tag{6.12}$$

We alternate between updating both blocks of variables until both converge. It is observed from the above problem that a linear combination of spherical harmonics is used to regularize estimated off-resonance frequencies. When the magnet center is unknown, in addition to spherical harmonics listed in Table 6.1, we consider one extra harmonic, either $\mathtt{x}^2$, $\mathtt{y}^2$, or $\mathtt{z}^2$, and increase the dimension of $\mathbf{c}$ to seventeen. In the following proposition, we prove that it is possible to obtain a representation for the surface of off-resonance frequencies regardless of the magnet center with seventeen spherical harmonics.

Proposition 9. *Regardless of the magnet center, one can decompose a surface made from the* 16 *harmonics in Table 6.1 across* 17 *basis functions,* 16 *harmonics listed in Table 6.1 and one of* $\mathtt{x}^2$*,* $\mathtt{y}^2$*, or* $\mathtt{z}^2$*.*

Proof. Let $\bar{c}_i$ denote the coefficient of one spherical harmonic function for the estimated magnetic field. We require to show that for every $\sum_{i=1}^u c_i \Psi_i(\mathtt{x}, \mathtt{y}, \mathtt{z}, \mathbf{o})$ there is an equal field as $\sum_{i=1}^u \bar{c}_i \Psi_i(\mathtt{x}, \mathtt{y}, \mathtt{z}, \mathbf{o})$ in all 3D positions. Therefore, we should show that there exists at least one solution $\{\bar{c}_i\}_i$ to have $\sum_{i=1}^u c_i \Psi_i(\mathtt{x}, \mathtt{y}, \mathtt{z}, \mathbf{o}) - \sum_{i=1}^u \bar{c}_i \Psi_i(\mathtt{x}, \mathtt{y}, \mathtt{z}, \mathbf{o}) = 0$. In Table 6.2, we expand and simplified the subtraction of two similar harmonics with the same order. To make sure that the two magnetic fields are equal, we use the following coefficients for the estimated surface:

$$\bar{c}_{10} = c_{10}, \bar{c}_{11} = c_{11}, \bar{c}_{12} = c_{12}, \bar{c}_{13} = c_{13},$$
$$\bar{c}_{14} = c_{14}, \bar{c}_{15} = c_{15}, \bar{c}_{16} = c_{16}.$$

Then, we can simplify the subtraction of two magnetic fields as given in Table 6.3. We expand and simplify additions given in different rows of Table 6.3. If we factor the fifth harmonic of Table 6.1, its coefficient is given in Table 6.4. This coefficient is equal to the coefficient of $\mathtt{z}^2$. We observe from Table 6.4 that regardless of $\mathbf{c}$, $\mathbf{o}$ or $\bar{\mathbf{o}}$, one can adjust $\bar{c}_5$ such that the coefficient of the fifth harmonic becomes zero.

Next, we factor the eighth harmonic and its coefficient is equal to the coefficient of $\mathtt{x}^2$. From Table 6.4, we observe that one can make the coefficient zero by adjusting $\bar{c}_8$ without any restrictions from $\mathbf{c}$, $\mathbf{o}$ or $\bar{\mathbf{o}}$.

We note that in the remaining harmonics, there is no $\mathtt{y}^2$ term to cancel the existing $\mathtt{y}^2$ function. Therefore, we add one harmonic $\mathtt{y}^2$ with a coefficient $\bar{c}_{17}$ to cancel the existing term, which is a function of $\mathtt{y}^2$. To do this, we can easily adjust $\bar{c}_{17}$. The remaining terms are only functions of $\mathtt{xz}$, $\mathtt{yz}$, $\mathtt{xy}$, $\mathtt{x}$, $\mathtt{y}$ and $\mathtt{z}$. We adjust $\bar{c}_6$, $\bar{c}_7$ and $\bar{c}_9$ such that the remaining terms which are functions of $\mathtt{xz}$, $\mathtt{yz}$, and $\mathtt{xy}$ are canceled. Moreover, we design $\bar{c}_2$, $\bar{c}_3$ and $\bar{c}_4$ in order to cancel linear terms in $\mathtt{x}$, $\mathtt{y}$ and $\mathtt{z}$. Finally, we choose $\bar{c}_1$ to cancel the constant remaining term. One can show that the above arguments are true if we add $\mathtt{x}^2$ or $\mathtt{z}^2$. □

If $\omega_j^k \Delta t_1 > \pi$ in a number of positions, using the first and second order finite difference regularizers damages the estimation. The reason is that the surface of the phase errors is no longer smooth. When $\omega_j^k \Delta t_1 > \pi$, even with optimized $e^{\iota \omega_j^k \Delta t_1}$, one cannot estimate ω_j^k, since the wrapped phase $\frac{\log(e^{\iota \omega_j^k \Delta t_1})}{\iota \Delta t_1}$ is unequal to ω_j^k.

Having multiple slices enables the approach to have a reasonable number of voxels available such that they satisfy $\omega_j^k \Delta t_1 < \pi$. First, we use this subset of voxels only and run the BCD approach. Instead of (6.2), we solve the following problem:

$$\min_{\boldsymbol{\omega}^k, \mathbf{x}^k} \quad \sum_{i=1}^{\Upsilon} \sum_{c=1}^{C} \sum_{j \in \mathcal{S}^{1,k}} \left| y_{cj}^k(t_i) - e^{\iota \omega_j^k t_i - R_j^k t_i} s_c^k(\mathbf{r}_j) x_j^k \right|^2, \tag{6.13}$$

Table 6.2: SUBTRACTION OF THE INHOMOGENEOUS MAGNETIC FIELD AND THE ESTIMATED MAGNETIC FIELD

n	m	Residual
0	0	$c_1 - \bar{c}_1$
1	0	$-o_3c_2 + \bar{o}_3\bar{c}_2 + z(c_2 - \bar{c}_2)$
1	1	$-o_1c_3 + \bar{o}_1\bar{c}_3 + x(c_3 - \bar{c}_3)$
1	1	$-o_2c_4 + \bar{o}_2\bar{c}_4 + y(c_4 - \bar{c}_4)$
2	0	$z^2(c_5 - \bar{c}_5) - 0.5x^2(c_5 - \bar{c}_5) - 0.5y^2(c_5 - \bar{c}_5) - 2z(c_5o_3 - \bar{c}_5\bar{o}_3) - y(c_5o_2 - \bar{c}_5\bar{o}_2)$ $+x(c_5o_1 - \bar{c}_5\bar{o}_1) + c_5(-0.5o_1^2 - 0.5o_2^2 + o_3^2) - \bar{c}_5(-0.5o_1^2 - 0.5o_2^2 + o_3^2)$
2	1	$xz(c_6 - \bar{c}_6) - z(-\bar{c}_6\bar{o}_1 + c_6o_1) - x(-\bar{c}_6\bar{o}_3 + c_6o_3) + c_6o_1o_3 - \bar{c}_6\bar{o}_1\bar{o}_3$
2	1	$yz(c_7 - \bar{c}_7) - z(-\bar{c}_7\bar{o}_2 + c_7o_2) - y(-\bar{c}_7\bar{o}_3 + c_7o_3) + c_7o_2o_3 - \bar{c}_7\bar{o}_2\bar{o}_3$
2	2	$x^2(c_8 - \bar{c}_8) - y^2(c_8 - \bar{c}_8) + 2x(-\bar{c}_8\bar{o}_1 + c_8o_1) - 2y(-\bar{c}_8\bar{o}_2 + c_8o_2) + c_8o_1^2 - c_8o_2^2$ $-\bar{c}_8\bar{o}_1^2 + \bar{c}_8\bar{o}_2^2$
2	2	$xy(c_9 - \bar{c}_9) - x(-\bar{c}_9\bar{o}_2 + c_9o_2) - y(-\bar{c}_9\bar{o}_1 + c_9o_1) + c_9o_1o_2 - \bar{c}_9\bar{o}_1\bar{o}_2$
3	0	$z^3(c_{10} - \bar{c}_{10}) - 1.5y^2z(c_{10} - \bar{c}_{10}) - 3z^2(c_{10}o_3 - \bar{c}_{10}\bar{o}_3) - 1.5x^2z(c_{10} - \bar{c}_{10})$ $+1.5y^2(c_{10}o_3 - \bar{c}_{10}\bar{o}_3) + 3xz(c_{10}o_1 - \bar{c}_{10}\bar{o}_1) + 3yz(c_{10}o_2 - \bar{c}_{10}\bar{o}_2) - 3y(c_{10}o_2o_3 - \bar{c}_{10}\bar{o}_2\bar{o}_3)$ $-3x(c_{10}o_1o_3 - \bar{c}_{10}\bar{o}_1\bar{o}_3) + z(-1.5c_{10}o_1^2 - 1.5c_{10}o_2^2 + 3c_{10}o_3^2 + 1.5\bar{c}_{10}\bar{o}_1^2 + 1.5\bar{c}_{10}\bar{o}_2^2 - 3\bar{c}_{10}\bar{o}_3^2)$ $+1.5c_{10}(o_1^2o_3 + o_2^2o_3 - o_3^3) - 1.5\bar{c}_{10}(\bar{o}_1^2\bar{o}_3 + \bar{o}_2^2\bar{o}_3 - \bar{o}_3^3) + 1.5x^2(c_{10}o_3 - \bar{c}_{10}\bar{o}_3)$
3	1	$-x^3(c_{11} - \bar{c}_{11}) - xy^2(c_{11} - \bar{c}_{11}) + 4xz^2(c_{11} - \bar{c}_{11}) - 4z^2(c_{11}o_1 - \bar{c}_{11}\bar{o}_1)$ $+y^2(c_{11}o_1 - \bar{c}_{11}\bar{o}_1) - 8xz(c_{11}o_3 - \bar{c}_{11}\bar{o}_3) + 2xy(c_{11}o_2 - \bar{c}_{11}\bar{o}_2) + 8z(c_{11}o_1o_3 - \bar{c}_{11}\bar{o}_1\bar{o}_3)$ $-2y(c_{11}o_1o_2 - \bar{c}_{11}\bar{o}_1\bar{o}_2) + x(-3c_{11}o_1^2 - c_{11}o_2^2 + 4c_{11}o_3^2 + 3\bar{c}_{11}\bar{o}_1^2 + \bar{c}_{11}\bar{o}_2^2 - 4\bar{c}_{11}\bar{o}_3^2)$ $+c_{11}(o_1^3 + o_1o_2^2 - 4o_1o_3^2) - \bar{c}_{11}(\bar{o}_1^3 + \bar{o}_1\bar{o}_2^2 - 4\bar{o}_1\bar{o}_3^2) + 3x^2(c_{11}o_1 - \bar{c}_{11}\bar{o}_1)$
3	1	$-y^3(c_{12} - \bar{c}_{12}) - x^2y(c_{12} - \bar{c}_{12}) + 4yz^2(c_{12} - \bar{c}_{12}) - 4z^2(c_{12}o_2 - \bar{c}_{12}\bar{o}_2)$ $+x^2(c_{12}o_2 - \bar{c}_{12}\bar{o}_2) - 8yz(c_{12}o_3 - \bar{c}_{12}\bar{o}_3) + 2xy(-c_{12}o_1 - \bar{c}_{12}\bar{o}_1) + 8z(c_{12}o_2o_3 - \bar{c}_{12}\bar{o}_2\bar{o}_3)$ $-2x(c_{12}o_1o_2 - \bar{c}_{12}\bar{o}_1\bar{o}_2) + y(-c_{12}o_1^2 - 3c_{12}o_2^2 + 4c_{12}o_3^2 + \bar{c}_{12}\bar{o}_1^2 + 3\bar{c}_{12}\bar{o}_2^2 - 4\bar{c}_{12}\bar{o}_3^2)$ $+3y^2(c_{12}o_2 - \bar{c}_{12}\bar{o}_2) + c_{12}(o_2^3 + o_1^2o_2 - 4o_2o_3^2) - \bar{c}_{12}(\bar{o}_2^3 + \bar{o}_1^2\bar{o}_2 - 4\bar{o}_2\bar{o}_3^2)$
3	2	$x^2z(c_{13} - \bar{c}_{13}) - y^2z(c_{13} - \bar{c}_{13}) - x^2(c_{13}o_3 - \bar{c}_{13}\bar{o}_3) + y^2(c_{13}o_3 - \bar{c}_{13}\bar{o}_3)$ $+2yz(c_{13}o_2 - \bar{c}_{13}\bar{o}_2) + 2x(c_{13}o_1o_3 - \bar{c}_{13}\bar{o}_1\bar{o}_3) - 2y(c_{13}o_2o_3 - \bar{c}_{13}\bar{o}_2\bar{o}_3)$ $+c_{13}(-o_1^2o_3 + o_2^2o_3) - \bar{c}_{13}(-\bar{o}_1^2\bar{o}_3 + \bar{o}_2^2\bar{o}_3) - 2xz(c_{13}o_1 - \bar{c}_{13}\bar{o}_1)$ $+z(c_{13}o_1^2 - c_{13}o_2^2 - \bar{c}_{13}\bar{o}_1^2 + \bar{c}_{13}\bar{o}_2^2)$
3	2	$xyz(c_{14} - \bar{c}_{14}) - yz(c_{14}o_1 - \bar{c}_{14}\bar{o}_1) - xz(c_{14}o_2 - \bar{c}_{14}\bar{o}_2) - xy(c_{14}o_3 - \bar{c}_{14}\bar{o}_3)$ $+z(c_{14}o_1o_2 - \bar{c}_{14}\bar{o}_1\bar{o}_2)$ $+x(c_{14}o_2o_3 - \bar{c}_{14}\bar{o}_2\bar{o}_3) + y(c_{14}o_1o_3 - \bar{c}_{14}\bar{o}_1\bar{o}_3) - c_{14}o_1o_2o_3 + \bar{c}_{14}o_1o_2o_3$
3	3	$x^3(c_{15} - \bar{c}_{15}) - 3xy^2(c_{15} - \bar{c}_{15}) - 3x^2(c_{15}o_1 - \bar{c}_{15}\bar{o}_1) + 3y^2(c_{15}o_1 - \bar{c}_{15}\bar{o}_1)$ $+3x(c_{15}o_1^2 - c_{15}o_2^2 - \bar{c}_{15}\bar{o}_1^2 + \bar{c}_{15}\bar{o}_2^2) - 6y(c_{15}o_1o_2 - \bar{c}_{15}\bar{o}_1\bar{o}_2) + c_{15}(-o_1^3 + 3o_1o_2^2)$ $+6xy(c_{15}o_2 - \bar{c}_{15}\bar{o}_2) - \bar{c}_{15}(-\bar{o}_1^3 + 3\bar{o}_1\bar{o}_2^2)$
3	3	$-y^3(c_{16} - \bar{c}_{16}) + 3x^2y(c_{16} - \bar{c}_{16}) - 3x^2(c_{16}o_2 - \bar{c}_{16}\bar{o}_2) + 3y^2(c_{16}o_2 - \bar{c}_{16}\bar{o}_2)$ $-6xy(c_{16}o_1 - \bar{c}_{16}\bar{o}_1) + 3y(c_{16}o_1^2 - c_{16}o_2^2 - \bar{c}_{16}\bar{o}_1^2 + \bar{c}_{16}\bar{o}_2^2) + 6x(c_{16}o_1o_2 - \bar{c}_{16}\bar{o}_1\bar{o}_2)$ $+c_{16}(o_2^3 - 3o_2o_1^2) - \bar{c}_{16}(\bar{o}_2^3 - 3\bar{o}_2\bar{o}_1^2)$

Table 6.3: Simplified subtraction of two magnetic fields

n	m	Residual
0	0	$c_1 - \bar{c}_1$
1	0	$-o_3 c_2 + \bar{o}_3 \bar{c}_2 + z(c_2 - \bar{c}_2)$
1	1	$-o_1 c_3 - \bar{o}_1 \bar{c}_3 + x(c_3 - \bar{c}_3)$
1	1	$-o_2 c_4 + \bar{o}_2 \bar{c}_4 + y(c_4 - \bar{c}_4)$
2	0	$z^2(c_5 - \bar{c}_5) - 0.5x^2(c_5 - \bar{c}_5) - 0.5y^2(c_5 - \bar{c}_5) - 2z(c_5 o_3 - \bar{c}_5 \bar{o}_3) + y(c_5 o_2 - \bar{c}_5 \bar{o}_2)$ $+x(c_5 o_1 - \bar{c}_5 \bar{o}_1) + c_5(-0.5o_1^2 - 0.5o_2^2 + o_3^2) - \bar{c}_5(-0.5o_1^2 - 0.5o_2^2 + o_3^2)$
2	1	$xz(c_6 - \bar{c}_6) - z(-\bar{c}_6 \bar{o}_1 + c_6 o_1) - x(-\bar{c}_6 \bar{o}_3 + c_6 o_3) + c_6 o_1 o_3 - \bar{c}_6 \bar{o}_1 \bar{o}_3$
2	1	$yz(c_7 - \bar{c}_7) - z(-\bar{c}_7 \bar{o}_2 + c_7 o_2) - y(-\bar{c}_7 \bar{o}_3 + c_7 o_3) + c_7 o_2 o_3 - \bar{c}_7 \bar{o}_2 \bar{o}_3$
2	2	$x^2(c_8 - \bar{c}_8) - y^2(c_8 - \bar{c}_8) + 2x(-\bar{c}_8 \bar{o}_1 + c_8 o_1) - 2y(-\bar{c}_8 \bar{o}_2 + c_8 o_2)$ $+c_8 o_1^2 - c_8 o_2^2 - \bar{c}_8 \bar{o}_1^2 + \bar{c}_8 \bar{o}_2^2$
2	2	$xy(c_9 - \bar{c}_9) - x(-\bar{c}_9 \bar{o}_2 + c_9 o_2) - y(-\bar{c}_9 \bar{o}_1 + c_9 o_1) + c_9 o_1 o_2 - \bar{c}_9 \bar{o}_1 \bar{o}_2$
3	0	$-3z^2(c_{10} o_3 - c_{10} \bar{o}_3) + 1.5x^2(c_{10} o_3 - c_{10} \bar{o}_3)$ $+1.5y^2(c_{10} o_3 - c_{10} \bar{o}_3) + 3xz(c_{10} o_1 - c_{10} \bar{o}_1) + 3yz(c_{10} o_2 - c_{10} \bar{o}_2)$ $-3y(c_{10} o_2 o_3 - c_{10} \bar{o}_2 \bar{o}_3) - 3x(c_{10} o_1 o_3 - c_{10} \bar{o}_1 \bar{o}_3)$ $+z(-1.5c_{10} o_1^2 - 1.5c_{10} o_2^2 + 3c_{10} o_3^2 + 1.5c_{10} \bar{o}_1^2 + 1.5c_{10} \bar{o}_2^2 - 3c_{10} \bar{o}_3^2)$ $+1.5c_{10}(o_1^2 o_3 + o_2^2 o_3 - o_3^3) - 1.5c_{10}(\bar{o}_1^2 \bar{o}_3 + \bar{o}_2^2 \bar{o}_3 - \bar{o}_3^3)$
3	1	$-4z^2(c_{11} o_1 - c_{11} \bar{o}_1) + 3x^2(c_{11} o_1 - c_{11} \bar{o}_1) + 8z(c_{11} o_1 o_3 - c_{11} \bar{o}_1 \bar{o}_3)$ $+y^2(c_{11} o_1 - c_{11} \bar{o}_1) - 8xz(c_{11} o_3 - c_{11} \bar{o}_3) + 2xy(c_{11} o_2 - c_{11} \bar{o}_2)$ $-2y(c_{11} o_1 o_2 - c_{11} \bar{o}_1 \bar{o}_2) + x(-3c_{11} o_1^2 - c_{11} o_2^2 + 4c_{11} o_3^2 + 3c_{11} \bar{o}_1^2 + c_{11} \bar{o}_2^2 - 4c_{11} \bar{o}_3^2)$ $+c_{11}(o_1^3 + o_1 o_2^2 - 4o_1 o_3^2) - c_{11}(\bar{o}_1^3 + \bar{o}_1 \bar{o}_2^2 - 4\bar{o}_1 \bar{o}_3^2)$
3	1	$-4z^2(c_{12} o_2 - c_{12} \bar{o}_2) + 3y^2(c_{12} o_2 - c_{12} \bar{o}_2) + x^2(c_{12} o_2 - c_{12} \bar{o}_2)$ $-8yz(c_{12} o_3 - c_{12} \bar{o}_3) + 2xy(c_{12} o_1 - c_{12} \bar{o}_1) + 8z(c_{12} o_2 o_3 - c_{12} \bar{o}_2 \bar{o}_3)$ $-2x(c_{12} o_1 o_2 - c_{12} \bar{o}_1 \bar{o}_2) + y(-c_{12} o_1^2 - 3c_{12} o_2^2 + 4c_{12} o_3^2 + c_{12} \bar{o}_1^2 + 3c_{12} \bar{o}_2^2 - 4c_{12} \bar{o}_3^2)$ $+c_{12}(o_2^3 + o_1^2 o_2 - 4o_2 o_3^2) - c_{12}(\bar{o}_2^3 + \bar{o}_1^2 \bar{o}_2 - 4\bar{o}_2 \bar{o}_3^2)$
3	2	$-x^2(c_{13} o_3 - c_{13} \bar{o}_3) + y^2(c_{13} o_3 - c_{13} \bar{o}_3) + z(c_{13} o_1^2 - c_{13} o_2^2 - c_{13} \bar{o}_1^2 + c_{13} \bar{o}_2^2)$ $+2yz(c_{13} o_2 - c_{13} \bar{o}_2) + 2x(c_{13} o_1 o_3 - c_{13} \bar{o}_1 \bar{o}_3) - 2y(c_{13} o_2 o_3 - c_{13} \bar{o}_2 \bar{o}_3)$ $+c_{13}(-o_1^2 o_3 + o_2^2 o_3) - c_{13}(-\bar{o}_1^2 \bar{o}_3 + \bar{o}_2^2 \bar{o}_3) - 2xz(c_{13} o_1 - c_{13} \bar{o}_1)$
3	2	$-yz(c_{14} o_1 - c_{14} \bar{o}_1) - xz(c_{14} o_2 - c_{14} \bar{o}_2) - xy(c_{14} o_3 - c_{14} \bar{o}_3) + z(c_{14} o_1 o_2 - c_{14} \bar{o}_1 \bar{o}_2)$ $+x(c_{14} o_2 o_3 - c_{14} \bar{o}_2 \bar{o}_3) + y(c_{14} o_1 o_3 - c_{14} \bar{o}_1 \bar{o}_3) - c_{14} o_1 o_2 o_3 + c_{14} o_1 o_2 o_3$
3	3	$-3x^2(c_{15} o_1 - c_{15} \bar{o}_1) + 3y^2(c_{15} o_1 - c_{15} \bar{o}_1) + 6xy(c_{15} o_2 - c_{15} \bar{o}_2)$ $+3x(c_{15} o_1^2 - c_{15} o_2^2 - c_{15} \bar{o}_1^2 + c_{15} \bar{o}_2^2) - 6y(c_{15} o_1 o_2 - c_{15} \bar{o}_1 \bar{o}_2) + c_{15}(-o_1^3 + 3o_1 o_2^2)$ $-c_{15}(-\bar{o}_1^3 + 3\bar{o}_1 \bar{o}_2^2)$
3	3	$-3x^2(c_{16} o_2 - c_{16} \bar{o}_2) + 3y^2(c_{16} o_2 - c_{16} \bar{o}_2) - 6xy(c_{16} o_1 - c_{16} \bar{o}_1)$ $+3y(c_{16} o_1^2 - c_{16} o_2^2 + c_{16} \bar{o}_1^2 - c_{16} \bar{o}_2^2) + 6x(c_{16} o_1 o_2 - c_{16} \bar{o}_1 \bar{o}_2)$ $+c_{16}(o_2^3 - 3o_2 o_1^2) - c_{16}(\bar{o}_2^3 - 3\bar{o}_2 \bar{o}_1^2)$

Table 6.4: Coefficients for different harmonics

Cartesian harmonic	Coefficient
1	straightforward to calculate
$\mathtt{z} - \mathtt{o}_3$	straightforward to calculate
$\mathtt{x} - \mathtt{o}_1$	straightforward to calculate
$\mathtt{y} - \mathtt{o}_2$	straightforward to calculate
$(\mathtt{z} - \mathtt{o}_3)^2$	$(c_5 - \bar{c}_5) - 3(c_{10}\mathtt{o}_3 - c_{10}\bar{\mathtt{o}}_3) - 4(c_{11}\mathtt{o}_1 - c_{11}\bar{\mathtt{o}}_1)$ $- \left((\mathtt{x} - \mathtt{o}_1)^2 + (\mathtt{y} - \mathtt{o}_2)^2\right)/2 - 4(c_{12}\mathtt{o}_2 - c_{12}\bar{\mathtt{o}}_2)$
$(\mathtt{x} - \mathtt{o}_1)(\mathtt{z} - \mathtt{o}_3)$	straightforward to calculate
$(\mathtt{y} - \mathtt{o}_2)(\mathtt{z} - \mathtt{o}_3)$	straightforward to calculate
$(\mathtt{x} - \mathtt{o}_1)(\mathtt{y} - \mathtt{o}_2)$	straightforward to calculate
$(\mathtt{x} - \mathtt{o}_1)^2 - (\mathtt{y} - \mathtt{o}_2)^2$	$-3(c_{16}\mathtt{o}_2 - c_{16}\bar{\mathtt{o}}_2) - 3(c_{15}\mathtt{o}_1 - c_{15}\bar{\mathtt{o}}_1) - (c_{13}\mathtt{o}_3 - c_{13}\bar{\mathtt{o}}_3)$ $+(c_{12}\mathtt{o}_2 - c_{12}\bar{\mathtt{o}}_2) - 2(c_{11}\mathtt{o}_1 - c_{11}\bar{\mathtt{o}}_1) - 2(c_{12}\mathtt{o}_2 - c_{12}\bar{\mathtt{o}}_2)$ $+3(c_{11}\mathtt{o}_1 - c_{11}\bar{\mathtt{o}}_1) + 1.5(c_{10}\mathtt{o}_3 - c_{10}\bar{\mathtt{o}}_3) + (c_8 + \bar{c}_8)$ $+0.5\left((c_5 + \bar{c}_5) - 3(c_{10}\mathtt{o}_3 - c_{10}\bar{\mathtt{o}}_3)\right)$
$\mathtt{y}^2$	$3(c_{16}\mathtt{o}_2 - c_{16}\bar{\mathtt{o}}_2) + 3(c_{15}\mathtt{o}_1 - c_{15}\bar{\mathtt{o}}_1) + (c_{13}\mathtt{o}_2 - c_{13}\bar{\mathtt{o}}_2)$ $+(c_{11}\mathtt{o}_1 - c_{11}\bar{\mathtt{o}}_1) + 1.5(c_{10}\mathtt{o}_3 - c_{10}\bar{\mathtt{o}}_3) - (c_8 + \bar{c}_8) + \bar{c}_{17}$ $3(c_{16}\mathtt{o}_2 - c_{16}\bar{\mathtt{o}}_2) - 3(c_{15}\mathtt{o}_1 - c_{15}\bar{\mathtt{o}}_1) - (c_{13}\mathtt{o}_3 - c_{13}\bar{\mathtt{o}}_3)$ $-3(c_{11}\mathtt{o}_1 - c_{11}\bar{\mathtt{o}}_1) + 1.5(c_{10}\mathtt{o}_3 - c_{10}\bar{\mathtt{o}}_3)$ $-0.5\left((c_5 + \bar{c}_5) - 3(c_{10}\mathtt{o}_3 - c_{10}\bar{\mathtt{o}}_3) - 4(c_{11}\mathtt{o}_1 - c_{11}\bar{\mathtt{o}}_1)\right)$ $-2(c_{12}\mathtt{o}_2 - c_{12}\bar{\mathtt{o}}_2) + (c_{12}\mathtt{o}_2 - c_{12}\bar{\mathtt{o}}_2) + 3(c_{12}\mathtt{o}_2 - c_{12}\bar{\mathtt{o}}_2)$

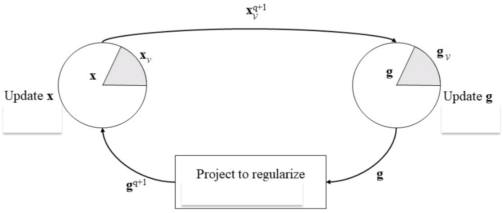


Figure 6.1: The BCD Algorithm 6.1 using a subset of voxels $\mathcal{V}$.

We update the set of voxels that correspond to the set $\mathcal{S}^{1,k}$, denoted by $\mathbf{x}^k_{\mathcal{S}^{1,k}}$, for all $j \in \mathcal{S}^{1,k}$ via (6.9). We update g^k_j as follows:

$$\tilde{\mathbf{g}}^k_{\mathcal{S}^{1,k}} = \arg\min_{\mathbf{g}^k_{\mathcal{S}^{1,k}}} \sum_{c=1}^{C}\sum_{i=1}^{\Upsilon}\sum_{j\in\mathcal{S}^{1,k}} \left| y^k_{cj}(t_i) e^{-\iota 2\pi \hat{f}_j^{k,s} t_i} e^{-\iota g_j^k (1+\frac{\sum_{a=2}^{i} \Delta t_a}{\Delta t_1})} - e^{-R_j^k t_i} s_c^k(\mathbf{r}_j) x_j^{q+1,k}\right|^2, \tag{6.14}$$

$$\begin{aligned} \mathbf{g}^{q+1,k}_{\mathcal{S}^{1,k}} = & \min_{\mathbf{g}^k_{\mathcal{S}^{1,k}} \in \mathcal{R}^{|\mathcal{S}^{1,k}|}, \mathbf{c}^k \in \mathcal{R}^{17}} \quad \left\| \mathbf{g}^k_{\mathcal{S}^{1,k}} - \tilde{\mathbf{g}}^k_{\mathcal{S}^{1,k}} \right\|_2^2 \\ & \text{s.t.} \qquad \gamma \mathbf{E}_{\mathcal{S}^{1,k}} \mathbf{c}^k = \frac{\mathbf{g}^k_{\mathcal{S}^{1,k}}}{\Delta t_1}, \end{aligned} \tag{6.15}$$

where

$$\mathbf{E}_{\mathcal{S}^{1,k}} = \{\mathbf{e}_j\}_{j\in\mathcal{S}^{1,k}}.$$

As discussed before, we alternate between updating $\mathbf{x}^k_{\mathcal{S}^{1,k}}$ and $\mathbf{g}^k_{\mathcal{S}^{1,k}}$ until both blocks converge. The utilization of a subset of voxels in the proposed BCD method is depicted in Fig. 6.1 and summarized in Algorithm 20.

We slowly increase $\mathcal{S}^{1,k}$ size by considering voxels with $g_j^{q,k} > \pi$ that have the smallest $|g_j^{q,k} - \pi|$. By expanding the size of $\mathcal{S}^{1,k}$, we improve the accuracy of the field map estimation. Using the obtained estimation about the magnetic field, we predict the unwrapped phase in each voxel for which $\omega_j^k \Delta t_1 > \pi$ using spherical harmonics. This is due to the fact that f_j^m and $f_j^{r,k}$ can both be well-represented via spherical harmonics. Let p denote the iteration number that specifies the size of $\mathcal{S}$. We predict the number of times the phase error in one

Algorithm 20: Voxel-level BCD Algorithm to solve (6.17)

```
0. **Initialization** Use (6.7) to initialize phase errors;
**repeat**
    **for** *each* $j \in \mathcal{V}$ **do**
        1. Update $x_j^{q+1,k}$ using (6.9);
        2. Update $\tilde{g}_j^k$ using (6.18);
        3. Calculate $\breve{g}_j^k$ using (6.19a)-(6.19h);
        4. Regularize $\breve{\mathbf{g}}^k$ using (6.20) and obtain $\mathbf{g}^{q+1,k}$;
    5. $q = q + 1$;
**until** $\left\|\mathbf{x}_{\mathcal{S}^{p+1,k}}^{q} - \mathbf{x}_{\mathcal{S}^{p+1,k}}^{q-1}\right\|_2^2 + \left\|\mathbf{g}_{\mathcal{S}^{p+1,k}}^{q} - \mathbf{g}_{\mathcal{S}^{p+1,k}}^{q-1}\right\|_2^2$ *is small enough*;
```

position has exceeded $-\pi$ or π using the following equation:

$$\psi_j^{p,k} = \frac{g_j^{q,k}}{\pi}.$$

Moreover, we define

$$\delta_j^{p,k} = \begin{cases} \lfloor \psi_j^{p,k} \rfloor, & g_j^{q,k} > 0 \\ \lceil |\psi_j^{p,k}| \rceil, & \text{otherwise,} \end{cases} \tag{6.16}$$

where q in the above equation is the last iteration of the BCD Algorithm 20, as we expand $\mathcal{S}^p$ size after the convergence of Algorithm 20. In the $p+1^{\text{th}}$ iteration, we consider the following optimization:

$$\min_{\mathbf{g}^k, \mathbf{x}^k} \quad \sum_{i=1}^{\Upsilon} \sum_{c=1}^{C} \sum_{j \in \mathcal{S}^{p+1,k}} \left| y_{cj}^k(t_i) e^{-\iota 2\pi \hat{f}_j^{k,s} t_i} e^{-\iota g_j^k (1 + \frac{\sum_{a=2}^{i} \Delta t_a}{\Delta t_1})} - e^{-R_j^k t_i} s_c^k(\mathbf{r}_j) x_j^k \right|^2, \tag{6.17}$$

In each iteration, to enlarge $\mathcal{S}^{p,k}$, we choose those voxels excluded in the set $\mathcal{S}^{p,k}$ with closest unwrapped phase magnitudes to the maximum magnitude of unwrapped phase of voxels in $\mathcal{S}^{p,k}$, i.e.,

$$\mathcal{S}^{p+1,k} = \left\{ j \middle| j \in \{\mathcal{V}_k\}_{k=1}^{K}, j \notin \mathcal{S}^{p,k}, \left| |g_j^{q,k}| - \max\left(\left| \{g_a^{q,k}\}_{a \in \mathcal{S}^{p,k}} \right| \right) \right| \leq \epsilon_p \right\}, \tag{6.18}$$

$$\breve{g}_j^k = \begin{cases} \tilde{g}_j^k + \pi - \delta_j^{p,k}\pi, & \text{rem}(\delta_j^{p,k}, 2) = 1 \text{ and } \tilde{g}_j^k < 0 \text{ and } \psi_j^{p,k} < 0, \qquad (6.19a) \\ -\lfloor|\frac{\psi_j^{p,k}}{\pi}|\rceil\pi, & \text{rem}(\delta_j^{p,k}, 2) = 1 \text{ and } \tilde{g}_j^k > 0 \text{ and } \psi_j^{p,k} < 0, \qquad (6.19b) \\ \tilde{g}_j^k + \pi + \delta_j^{p,k}\pi, & \text{rem}(\delta_j^{p,k}, 2) = 1 \text{ and } \tilde{g}_j^k < 0 \text{ and } \psi_j^{p,k} > 0, \qquad (6.19c) \\ \lfloor|\frac{\psi_j^{p,k}}{\pi}|\rceil\pi, & \text{rem}(\delta_j^{p,k}, 2) = 1 \text{ and } \tilde{g}_j^k > 0 \text{ and } \psi_j^{p,k} > 0, \qquad (6.19d) \\ \tilde{g}_j^k + \delta_j^{p,k}\pi, & \text{rem}(\delta_j^{p,k}, 2) = 0 \text{ and } \tilde{g}_j^k > 0 \text{ and } \psi_j^{p,k} > 0, \qquad (6.19e) \\ \lfloor\frac{\psi_j^{p,k}}{\pi}\rceil\pi, & \text{rem}(\delta_j^{p,k}, 2) = 0 \text{ and } \tilde{g}_j^k < 0 \text{ and } \psi_j^{p,k} > 0, \qquad (6.19f) \\ \tilde{g}_j^k - \delta_j^{p,k}\pi, & \text{rem}(\delta_j^{p,k}, 2) = 0 \text{ and } \tilde{g}_j^k > 0 \text{ and } \psi_j^{p,k} < 0, \qquad (6.19g) \\ -\lfloor|\frac{\psi_j^{p,k}}{\pi}|\rceil\pi, & \text{rem}(\delta_j^{p,k}, 2) = 0 \text{ and } \tilde{g}_j^k < 0 \text{ and } \psi_j^{p,k} < 0, \qquad (6.19h) \end{cases}$$

We run Algorithm 20 again with the augmented voxel set. We update x_j^k using (6.10). We update $\tilde{g}_j^k$ as follows:

$$\tilde{\mathbf{g}}_{\mathcal{S}^{p+1,k}}^k = \arg \min_{\mathbf{g}_{\mathcal{S}^{p+1,k}}^k} \sum_{c=1}^{C} \sum_{i=1}^{\Upsilon} \sum_{j \in \mathcal{S}^{p+1,k}} \left| y_{cj}^k(t_i) e^{-\iota 2\pi \hat{f}_j^{k,s} t_i} e^{-\iota g_j^k (1 + \frac{\sum_{a=2}^{i} \Delta t_a}{\Delta t_1})} - e^{-R_j^k t_i} s_c(\mathbf{r}_j) x_j^{q+1,k} \right|^2,$$

However, we regularize $\tilde{\mathbf{g}}^k$ differently compared to (6.15). For each voxel $j \in \mathcal{S}^{p+1,k}$, we predict the unwrapped phase error using $\delta_j^{p,k}$ and optimized $\tilde{g}_j^k$ as given in (6.19a)-(6.19h).

We justify these equations as follows:

1. Equation (6.19a): In this case, the unwrapped phase lies in the interval $[-(2a+1)\pi, -2a\pi]$, $a \in \mathbb{Z}_+$, since $\text{rem}(\delta_j^{p,k}, 2) = 1$ and $\psi_j^{p,k} < 0$. Here, $\delta_j^{p,k}\pi = -(2a+1)\pi$. Given that $-\pi \leq \tilde{g}_j^k \leq 0$, $\pi + \tilde{g}_j^k$ is the amount of needed positive phase to be added $\delta_j^{p,k}\pi$ to obtain the unwrapped phase.

2. Equation (6.19b): Similar to the previous case, the unwrapped phase lies in the interval $[-(2a+1)\pi, -2a\pi]$, $a \in \mathbb{Z}_+$. However, the received signal was corrupted, which resulted in a wrapped estimated phase in the interval $[0, \pi]$. Here, we map the estimated phase to the closest point in the predicted interval, which is $-\lfloor|\frac{\psi_j^{p,k}}{\pi}|\rceil\pi$.

3. Equation (6.19c): In this case, the unwrapped phase lies in the interval $[(2a+1)\pi, (2a+2)\pi]$, $a \in \mathbb{Z}_+$, since $\text{rem}(\delta_j^{p,k}, 2) = 1$ and $\psi_j^{p,k} > 0$. Since $-\pi \leq \tilde{g}_j^k \leq 0$, the additional positive phase that needed to be added $\delta_j^{p,k}\pi$ to obtain the unwrapped phase is the complement of $|\tilde{g}_j^k|$, which is $\pi + \tilde{g}_j^k$.

4. Equation (6.19d): In this case, the unwrapped phase lies in the interval $[(2a+1)\pi, (2a+2)\pi]$, $a \in \mathbb{Z}_+$ similar to the previous case. Due to the received signal corruption, the wrapped estimated phase is in the interval $[0, \pi]$. Here, we map the estimated phase to $\lfloor | \frac{\psi_j^{p,k}}{\pi} | \rceil \pi$, which is the closest point in the predicted interval.

5. Equation (6.19e): In this case, the unwrapped phase lies in the interval $[2a\pi, (2a+1)\pi]$, $a \in \mathbb{Z}_+$, since $\text{rem}(\delta_j^{p,k}, 2) = 0$ and $\psi_j^{p,k} > 0$. As $0 \leq \tilde{g}_j^k \leq \pi$, the additional needed positive phase to be added $\delta_j^{p,k}\pi$ to obtain the unwrapped phase is $\tilde{g}_j^k$.

6. Equation (6.19f): Similar to the previous case, the unwrapped phase lies in the interval $[2a\pi, (2a+1)\pi]$, $a \in \mathbb{Z}_+$. Since the received signal is corrupted, the wrapped estimated phase is in the interval $[-\pi, 0]$. In this case, we map the estimated phase to the closest point in the predicted interval, which is $\lfloor | \frac{\psi_j^{p,k}}{\pi} | \rceil \pi$.

7. Equation (6.19e): In this case, the unwrapped phase lies in the interval $[-(2a+2)\pi, -(2a+1)\pi]$, $a \in \mathbb{Z}_+$, since $\text{rem}(\delta_j^{p,k}, 2) = 0$ and $\psi_j^{p,k} < 0$. As $0 \leq \tilde{g}_j^k \leq \pi$, the additional positive phase that needs to be added $-\delta_j^{p,k}\pi$ to obtain the unwrapped phase is $\tilde{g}_j^k$.

8. Equation (6.19f): Similar to the previous case, the unwrapped phase lies in the interval $[-(2a+2)\pi, -(2a+1)\pi]$, $a \in \mathbb{Z}_+$. Since the received signal was corrupted, the wrapped estimated phase is in the interval $[-\pi, 0]$. In this case, we map the estimated phase to the closest point in the predicted interval, which is $-\lfloor | \frac{\psi_j^{p,k}}{\pi} | \rceil \pi$.

Next, we use $\check{\mathbf{g}}_{\mathcal{S}^{p+1,k}}^k$ to estimate $\mathbf{g}_{\mathcal{S}^{p+1,k}}^{q+1,k}$ and do the regularization as follows:

$$\begin{aligned} \mathbf{g}_{\mathcal{S}^{p+1,k}}^{q+1,k} = \min_{\mathbf{g}^k \in \mathcal{R}^{|\mathcal{S}^{p+1,k}|}, \mathbf{c}^k \in \mathcal{R}^{17}} \quad & \left\| \mathbf{g}_{\mathcal{S}^{p+1,k}}^k - \check{\mathbf{g}}_{\mathcal{S}^{p+1,k}}^k \right\|_2^2 \\ \text{s.t.} \quad & \gamma \mathbf{E}_{\mathcal{S}^{p+1,k}} \mathbf{c}^k = \frac{\mathbf{g}_{\mathcal{S}^{p+1,k}}^k}{\Delta t_1}, \end{aligned} \tag{6.20}$$

Algorithm 21: Iteratively Incremental Algorithm to solve (6.2)

```
0. **Initialization** Choose voxels near the magnet center in $\mathcal{S}^{1,k}$, $p = 0$;
**repeat**
    1. $p = p + 1$;
    2. Call Algorithm 20;
    3. Enlarge the voxel set as given in (6.18);
**until** $\mathcal{S}^{p,k}$ *includes all the image voxels*;
4. $\hat{\mathbf{X}} = \mathbf{X}^q$ and $\hat{\mathbf{g}} = \mathbf{g}^q$;
```

The proposed approach to solve (6.17) is described in Algorithm 20. Step-by-step enlargement of $\mathcal{S}^{p,k}$ size enables the approach to significantly decrease unwrapped phase prediction errors. The brief description of the proposed method to solve (6.2) is given in Algorithm 21.

When an object-induced magnetic field from a susceptibility map can be well-estimated, given that the object-induced field out of the FOV is small, one can combine samples from different objects in positions that the inhomogeneity from the magnet is large. Aiming at identifying the magnet inhomogeneity, we formulate a problem as follows:

$$\min_{\boldsymbol{\omega},\mathbf{X}} \quad \sum_{i=1}^{\Upsilon}\sum_{k=1}^{K}\sum_{c=1}^{C}\sum_{j\in\mathcal{V}_l} \left| y^k_{cj}(t_i) - e^{\iota\omega_j t_i - R_j t_i} s^k_c(\mathbf{r}_j) x^k_j \right|^2,$$

where $\mathbf{X} = \{\mathbf{x}_k\}_{k=1}^{K}$ and $\mathcal{V}_l$ represent a subset of voxels in which the inhomogeneity from the magnet is large. One can solve the above optimization using a BCD approach with a projection similar to Algorithm 21.

6.4.2 Incremental Processing of Echo Times

Since the received signal is decaying over time, the signal becomes hard to distinguish from noise if the echo time number becomes large. Therefore, jointly optimizing all echo times damages estimations due to the low SNR level in last echo times. To enhance the accuracy of our estimations, we predict the magnetic field using the initial echo times. Then, based on the predicted phase errors, we filter a subset of samples. To determine this subset, we use the estimated underlying voxels and formulate the following per-voxel, per-echo optimization:

$$\bar{x}_j^k, \bar{g}_j^k(t_i) = \min_{x_j^k, g_j^k(t_i)} \quad \sum_{c=1}^{C} \left| y_{cj}^k(t_i) e^{-\iota 2\pi \hat{f}_j^{k,s} t_i} e^{-\iota g_j^k(t_i)(1+\frac{\sum_{a=2}^{i} \Delta t_a}{\Delta t_1})} - e^{-R_j^k t_i} s_c^k(\mathbf{r}_j) x_j^k \right|^2, \quad (6.21)$$

where $g_j^k(t_i) = 2\pi(f_j^m + f_j^{k,r})\Delta t_1$ and $i > \Upsilon$. To solve the above problem, we start optimizing with respect to $g_j^k(t_i)$ and we set x_j^k to its estimate from echoes $i \leq \Upsilon$. One can leverage the object-induced field information and solve the above optimization using a BCD approach where the subproblem with respect to $g_j^k(t_i)$ is solved via golden-section search nested inside a grid-search and the subproblem with respect to x_j^k has a closed-form solution. If $|\tilde{g}_j^k - \bar{g}_j^k(t_i)|$ is greater than a certain threshold, we consider $y_{cj}^k(t_i)$ as a toxic measurement and do not include that in our optimization formulation. We represent the threshold by δ_j. We define a subset of echo times which are beneficial for one voxel as follows:

$$\mathcal{L}_j^i = \{i | \delta_j \geq |\tilde{g}_j^k - \bar{g}_j^k(t_i)|\}.$$

Using $\mathcal{L}_j^i$, we formulate the following optimization problem to achieve a higher accuracy via incorporating a greater number of samples compared to the problem in the previous section:

$$\begin{aligned} \min_{\mathbf{X}, \boldsymbol{\omega}} \quad & \sum_{k=1}^{K} \sum_{c=1}^{C} \sum_{i=1}^{\Upsilon} \sum_{j \in \mathcal{V}_k} \left| y_{cj}^k(t_i) - e^{\iota \omega_j^k t_i - R_j^k t_i} s_c^k(\mathbf{r}_j) x_j^k \right|^2 \qquad (6.22) \\ & + \sum_{k=1}^{K} \sum_{c=1}^{C} \sum_{j \in \mathcal{V}_k} \sum_{i \in \mathcal{L}_j^i} \left| y_{cj}^k(t_i) - e^{\iota \omega_j^k t_i - R_j^k t_i} s_c^k(\mathbf{r}_j) x_j^k \right|^2 \\ & + \theta \left\| \hat{\mathbf{X}} - \mathbf{X} \right\|_F^2, \end{aligned}$$

where $\hat{\mathbf{X}}$ is the solution of (6.17) obtained from Algorithm 21. Since the SNR decays with the echo time number, we use estimated image voxels from the initial echo times in the proximal term to promote solutions that are close to those from the initial echo times. We iteratively solve the above optimization using Algorithm 21 with phase unwrapping similar to the previous section.

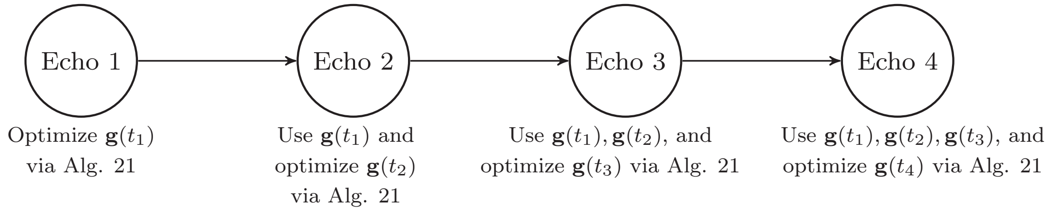

Echo 1
Echo 2
Echo 3
Echo 4
Optimize $\mathbf{g}(t_1)$ via Alg. 21
Use $\mathbf{g}(t_1)$ and optimize $\mathbf{g}(t_2)$ via Alg. 21
Use $\mathbf{g}(t_1), \mathbf{g}(t_2)$, and optimize $\mathbf{g}(t_3)$ via Alg. 21
Use $\mathbf{g}(t_1), \mathbf{g}(t_2), \mathbf{g}(t_3)$, and optimize $\mathbf{g}(t_4)$ via Alg. 21

Figure 6.2: Sequentially optimizing phase errors of different echo times.

6.5 Time-Varying Magnetic Field

Due to the eddy currents and other scanner imperfections, the magnetic field can evolve with time. We parameterize the phase errors, off-resonance frequencies, and coefficient for spherical harmonics by t, which specifies the echo time. Since the magnetic field varies with time, we cannot use all measurements from different echo times in one optimization problem. Unlike the previous section, here we process different echo times separately and optimize the phase variation from one echo time to another. Sequential processing of echo times is depicted in Fig. 6.2. In this section, we make the below assumptions:

1. The induced magnetic field from susceptibility map of an object is known;
2. The induced magnetic field out of the FOV is negligible;
3. The signal decay does not significantly hurt the first few echoes;
4. In addition to distorted images, there is a noisy, but undistorted set of images, denoted by $\breve{\mathbf{X}}$.

Let us denote the additional phase error originating from the magnet in the j^{th} voxel from echo time t_{t-1} to t_i by $g_j(t_i)$. Then, one can obtain the average off-resonance frequency for one echo time from $g_j(t_i) = 2\pi\Delta t_i f_j^m(t_i)$, where $f_j^m(t_i)$ is in the j^{th} voxel at echo time i.

We use different objects to add diversity. For one echo time, we initialize the phase errors as follows:

$$g_j^0(t_i) = \min_{g_j(t_i)} \sum_{k=1}^{K}\sum_{c=1}^{C} \left| y_{cj}^k(t_i)e^{-\iota 2\pi \hat{f}_j^{k,s} t_i} - e^{\iota \sum_{a=1}^{i-1} g_j^0(t_a) + \iota g_j(t_i) - R_j^k t_i} s_c^k(\mathbf{r}_j)\breve{x}_j^k \right|^2, \qquad (6.23)$$

where the above minimization can be solved via a golden-section search nested inside a grid search. When the magnetic field is time-varying, the phase variation in two consecutive echo times with identical length for one position are not necessarily equal. We consider a general case where phase errors in different echo times are not related to each other. We optimize phase error difference between two echo times sequentially, i.e., $\Delta\omega_j(t_i) = \omega_j(t_i) - \omega_j(t_{i-1})$. We start with the first echo time. Then, we approximate the phase errors added in next echoes. Similar to the previous section, we estimate $\mathcal{S}^1(t_i)$ for each echo time $i \leq \Upsilon$ based on a set of initial phase errors $\{\Delta\omega_j^0(t_i)\}_{j\in\mathcal{S}^1(t_i)}$, where $|\Delta\omega_j^0(t_i)t_i| < \pi$. We denote the optimized $g_j(t_i)$ variable by $\hat{g}_j(t_i)$. To optimize $\{g_j(t_i)\}_{j\in\mathcal{S}^1(t_i)}$ and $\{\{x_j^k\}_{k=1}^K\}_{j\in\mathcal{S}^1(t_i)}$, in the p^{th} iteration of expanding the set of considered voxels, we solve the following problem:

$$\begin{aligned} &\min_{\{\{x_j^k\}_{j\in\mathcal{S}^p(t_i)}\}_{k=1}^K, \{g_j(t_i)\}_{j\in\mathcal{S}^p(t_i)}} \quad \frac{\rho}{2}\left\|\mathbf{X} - \breve{\mathbf{X}}\right\|_F^2 \\ &+ \sum_{k=1}^K \sum_{c=1}^C \sum_{j\in\mathcal{S}^p(t_i)} \left| y_{cj}^k(t_i) e^{-\iota 2\pi \hat{f}_j^{k,s} t_i} - e^{\iota \sum_{a=1}^{i-1} \hat{g}_j(t_a) + \iota g_j(t_i) - R_j^k t_i} s_c^k(\mathbf{r}_j) x_j^k \right|^2, \end{aligned}$$

where $\rho > 0$. One can solve the above problem via the BCD Algorithm 20 given in the previous section. The solution for x_j^k in the $q+1^{\text{th}}$ iteration is

$$x_j^{q+1,k} = \frac{\sum_{c=1}^C s_c^{k*}(\mathbf{r}_j) e^{-\iota \sum_{a=1}^{i-1} \hat{g}_j(t_a) - \iota g_j^q(t_i) - R_j^k t_i} y_{cj}^k(t_i) e^{-\iota 2\pi \hat{f}_j^{k,s} t_i} + \rho \breve{x}_j^k}{\sum_{i=1}^{\Upsilon} \sum_{c=1}^C e^{-2R_j^k t_i} |s_c^k(\mathbf{r}_j)|^2 + \rho}. \tag{6.24}$$

The solution with respect to g_j^k can be obtained as follows:

$$\begin{aligned} \tilde{\mathbf{g}} &= \arg\min_{\mathbf{g}} \quad \sum_{k=1}^K \sum_{c=1}^C \sum_{j\in\mathcal{S}^p(t_i)} \left| y_{cj}^k(t_i) e^{-\iota 2\pi \hat{f}_j^{k,s} t_i} - e^{\iota \sum_{a=1}^{i-1} \hat{g}_j(t_a) + \iota g_j(t_i) - R_j^k t_i} s_c^k(\mathbf{r}_j) x_j^{q+1,k} \right|^2. \\ \mathbf{g}_{\mathcal{S}^p}^{q+1}(t_i) &= \min_{\mathbf{g}\in\mathcal{R}^{|\mathcal{S}^p(t_i)|}, \mathbf{c}\in\mathcal{R}^{17}} \quad \|\mathbf{g} - \tilde{\mathbf{g}}\|_2^2 \\ & \qquad \text{s.t.} \qquad \gamma \mathbf{E}_{\mathcal{S}^p}\mathbf{c} = \frac{\mathbf{g}}{\Delta t_i}. \end{aligned}$$

The set of voxels is expanded and eventually $g_j(t_i)$ for all image voxels is estimated. This is similar to Algorithm 21; however, we only consider one echo time in each iteration and collect samples from having multiple coils and objects.

After the phase errors in different positions during the initial echo times are approximated, we proceed to estimate phase errors during the latter echo time $i > \Upsilon$ as follows:

$$\begin{aligned}\min_{\mathbf{X},\mathbf{g}(t_i)} \quad & \sum_{k=1}^{K}\sum_{c=1}^{C}\sum_{j\in\mathcal{S}^p}\left|y_{cj}^{k}(t_i)e^{-\iota 2\pi \hat{f}_j^{k,s}t_i} - e^{\iota\sum_{a=1}^{i-1}\hat{g}_j(t_a)+\iota g_j(t_i)-R_j^k t_i}s_c^k(\mathbf{r}_j)x_j^k\right|^2 \\ & +\frac{1}{2}\sum_{a=1}^{\Upsilon}\theta_a\left\|\hat{\mathbf{X}}(t_a)-\mathbf{X}\right\|_F^2+\frac{\rho}{2}\left\|\mathbf{X}-\breve{\mathbf{X}}\right\|_F^2,\end{aligned}$$

where $\theta_a > 0$. We solve the above problem for all echo times larger that Υ. Similar to the previous section, we start with a small set of voxels in $\mathcal{S}^p(t_i)$ and gradually expand the size of $\mathcal{S}^p(t_i)$. For one echo time greater than Υ, we use a BCD method with the following updates:

$$\begin{aligned} x_j^{q+1,k}(t_i) &= \frac{\rho\breve{x}_j^k+\sum_{a=1}^{\Upsilon}\theta_a\hat{x}_j^k(t_i)+\sum_{c=1}^{C}s_c^{*k}(\mathbf{r}_j)e^{-\iota(g_j^q(t_i)+\sum_{a=1}^{i-1}\hat{g}_j(t_a))-R_j^k t_i}y_{cj}^k(t_i)}{\rho+\sum_{a=1}^{\Upsilon}\theta_a+e^{-2R_j^k t_i}\sum_{c=1}^{C}|s_c^{*k}(\mathbf{r}_j)|^2}, j\in\mathcal{S}^p, \\ \tilde{\mathbf{g}} &= \arg\min_{\mathbf{g}}\ \sum_{k=1}^{K}\sum_{c=1}^{C}\sum_{j\in\mathcal{S}^p}\left|y_{cj}^k(t_i)e^{-\iota 2\pi\hat{f}_j^{k,s}t_i}-e^{\iota(g_j(t_i)+\sum_{a=1}^{i-1}\hat{g}_j(t_a))-R_j^k t_i}s_c^k(\mathbf{r}_j)x_j^{q+1,k}(t_i)\right|^2, \\ \mathbf{g}_{\mathcal{S}^p}^{q+1}(t_i) &= \min_{\mathbf{g}\in\mathcal{R}^{|\mathcal{S}^p(t_i)|},\mathbf{c}\in\mathcal{R}^{17}} \quad \|\mathbf{g}-\tilde{\mathbf{g}}\|_2^2 \\ & \qquad \text{s.t.} \qquad \gamma\mathbf{E}_{\mathcal{S}^p}\mathbf{c}=\frac{\mathbf{g}}{\Delta t_i}. \end{aligned} \tag{6.25}$$

It is generally considered that the effects of eddy currents are spatially static and linear. The effects of eddy currents are related at even/odd echo times. This is due to the symmetry of the gradient coil. In this case, the phase error variation can be modeled as follows:

$$\Delta w_j(t_i) = 2\pi(f_j^m + f_j^{k,s}) + \varphi_j(z(i)),$$

where

$$z(i) = \begin{cases} 1, & \text{if } i \text{ is even}, \\ 2, & \text{otherwise}, \end{cases}$$

and

$$2\pi f_j^m + \varphi_j(z(i)) = \gamma \mathbf{E}\mathbf{w}_i,$$

where $\mathbf{w}_i$ is the vector of weights given to different spherical harmonics at echo time i, and $\varphi_j(z(i))$ is the angular frequency induced by the eddy current in an even or odd echo time.

We can use the above property to relate different echo times. In this case, we jointly optimize phase errors in even and odd echo times to enhance the accuracy of estimations. For even or odd echo times, we formulate the following optimization problem:

$$\begin{aligned}\min_{\mathbf{X},\mathbf{g}_{z(i)}} \quad & \sum_{i=z(i):2:I}\sum_{k=1}^{K}\sum_{c=1}^{C}\sum_{j}\left|y_{cj}^{k}(t_i)e^{-\iota 2\pi \hat{f}_j^{k,s}t_i} - e^{\iota\Delta\omega_j(t_i)-R_j^k t_i}s_c^k(\mathbf{r}_j)x_j^k\right|^2 + \frac{\rho}{2}\left\|\mathbf{X}-\breve{\mathbf{X}}\right\|_F^2\\ \text{s.t.} \quad & 2\pi f_j^m + \varphi_j(z(i)) = \gamma\mathbf{E}\mathbf{w}_{z(i)}.\end{aligned}$$

In the above problem, the phase errors of even and odd echo times are coupled. The reason is that the phase errors of both even and odd echo times are included in one $\Delta\omega_j(t_i)$ for $i > 1$. We use Algorithm 21 to obtain an estimate for the phase error in each position in each echo time. Let us represent phase errors in even and odd echo times by $\mathbf{g}_{z(1)} = 2\pi f_j \Delta t_1$ and $\mathbf{g}_{z(2)} = 2\pi f_j \Delta t_2$, respectively. We have:

$$\begin{aligned}\min_{\mathbf{X},\mathbf{g}_{z(1)},\mathbf{g}_{z(2)}} \quad & \sum_{i=1}^{I}\sum_{k=1}^{K}\sum_{c=1}^{C}\sum_{j}\left|y_{cj}^{k}(t_i)e^{-\iota 2\pi \hat{f}_j^{k,s}t_i},\right.\\ & \left. - e^{\iota\mathbf{g}_{z(1)}(\sum_{a=1:2:i}\frac{\Delta t_a}{\Delta t_1})}e^{\iota\mathbf{g}_{z(2)}(\sum_{a=2:2:i}\frac{\Delta t_a}{\Delta t_2})}e^{-R_j^k t_i}s_c^k(\mathbf{r}_j)x_j^k\right|^2 + \frac{\rho}{2}\left\|\mathbf{X}-\breve{\mathbf{X}}\right\|_F^2\\ \text{s.t.} \quad & 2\pi f_j^m + \varphi_j(z(i)) = \gamma\mathbf{E}\mathbf{w}_{z(i)}.\end{aligned}$$

In the $q+1^{\text{th}}$ iteration of the BCD method, we update $x_j^k, j \in \mathcal{S}^{p,k}$ as follows:

$$x_j^{q+1,k} = \frac{\rho\breve{x}_j^k + \sum_{i=1}^{I}\sum_{c=1}^{C}s_c^{*k}(\mathbf{r}_j)e^{-\iota 2\pi\hat{f}_j^{k,s}t_i}e^{-\iota\sum_{a=1}^{i}g_{z(a),j}^{q}\frac{\Delta t_a}{\Delta t_{z(a)}}-R_j^k t_i}y_{cj}^k(t_i)}{\rho + \sum_{i=1}^{I}\sum_{c=1}^{C}e^{-2R_j t_i}|s_c^k(\mathbf{r}_j)|^2}.$$

In the $q+1^{\text{th}}$ iteration, we update $\mathbf{g}_{z(1)}^{q+1}$ as follows:

$$\begin{aligned}
\tilde{\mathbf{g}}_{z(1)} = \min_{\mathbf{g}_{z(1)}} & \sum_{i=1:2:I} \sum_{k=1}^{K} \sum_{c=1}^{C} \sum_{j\in\mathcal{S}^p} \Big| y_{cj}^{k}(t_i) e^{-\iota 2\pi \hat{f}_j^{k,s} t_i} - e^{\iota \mathbf{g}_{z(1)} (\sum_{a=1:2:i} \frac{\Delta t_a}{\Delta t_1})} \\
& e^{\iota \mathbf{g}_{z(2)}^{q} (\sum_{a=2:2:i} \frac{\Delta t_a}{\Delta t_2})} e^{-R_j^k t_i} s_c^k(\mathbf{r}_j) x_j^{q+1,k} \Big|^2, \\
\mathbf{g}_{\mathcal{S}^p,z(1)}^{q+1} = & \min_{\mathbf{g}\in\mathcal{R}^{|\mathcal{S}^p|}, \mathbf{w}_{z(1)}\in\mathcal{R}^{17}} \left\| \mathbf{g} - \tilde{\mathbf{g}}_{z(1)} \right\|_2^2 \\
& \text{s.t.} \quad \gamma \mathbf{E}_{\mathcal{S}^p} \mathbf{w}_{z(1)} = \frac{\mathbf{g}}{\Delta t_{z(1)}}.
\end{aligned} \tag{6.26}$$

In the $q+1^{\text{th}}$ iteration, we update $\mathbf{g}_{z(2)}^{q+1}$ as follows:

$$\begin{aligned}
\tilde{\mathbf{g}}_{z(2)} = \min_{\mathbf{g}_{z(2)}} & \sum_{i=2:2:I} \sum_{k=1}^{K} \sum_{c=1}^{C} \sum_{j\in\mathcal{S}^p} \Big| y_{cj}^{k}(t_i) e^{-\iota 2\pi \hat{f}_j^{k,s} t_i} - e^{\iota \mathbf{g}_{z(2)} (\sum_{a=2:2:i} \frac{\Delta t_a}{\Delta t_2})} \\
& e^{\iota \mathbf{g}_{z(1)}^{q+1} (\sum_{a=1:2:i} \frac{\Delta t_a}{\Delta t_1})} e^{-R_j^k t_i} s_c^k(\mathbf{r}_j) x_j^{q+1,k} \Big|^2, \\
\mathbf{g}_{\mathcal{S}^p,z(2)}^{q+1} = & \min_{\mathbf{g}\in\mathcal{R}^{|\mathcal{S}^p|}, \mathbf{w}_{z(2)}\in\mathcal{R}^{17}} \left\| \mathbf{g} - \tilde{\mathbf{g}}_{z(2)} \right\|_2^2 \\
\text{s.t.} \quad & \gamma \mathbf{E}_{\mathcal{S}^p} \mathbf{w}_{z(2)} = \frac{\mathbf{g}}{\Delta t_{z(2)}}.
\end{aligned} \tag{6.27}$$

We continue solving with respect to $\mathbf{X}$, $\tilde{\mathbf{g}}_{z(1)}$, and $\tilde{\mathbf{g}}_{z(2)}$ until all three blocks converge.

6.6 Numerical Results

In this section, we evaluate the performance of our approach via simulations using synthetic and real data. The SNR is calculated as follows:

$$\text{SNR} = 20 \log \frac{\|\mathbf{x}\|_2}{\|\mathbf{y}(t_1) - \mathbf{x}\|_2}$$

In the first experiment, we use the NYU brain DICOM dataset with multiple slices. MATLAB is used for implementations and coil sensitivities are simulated via MIRT.

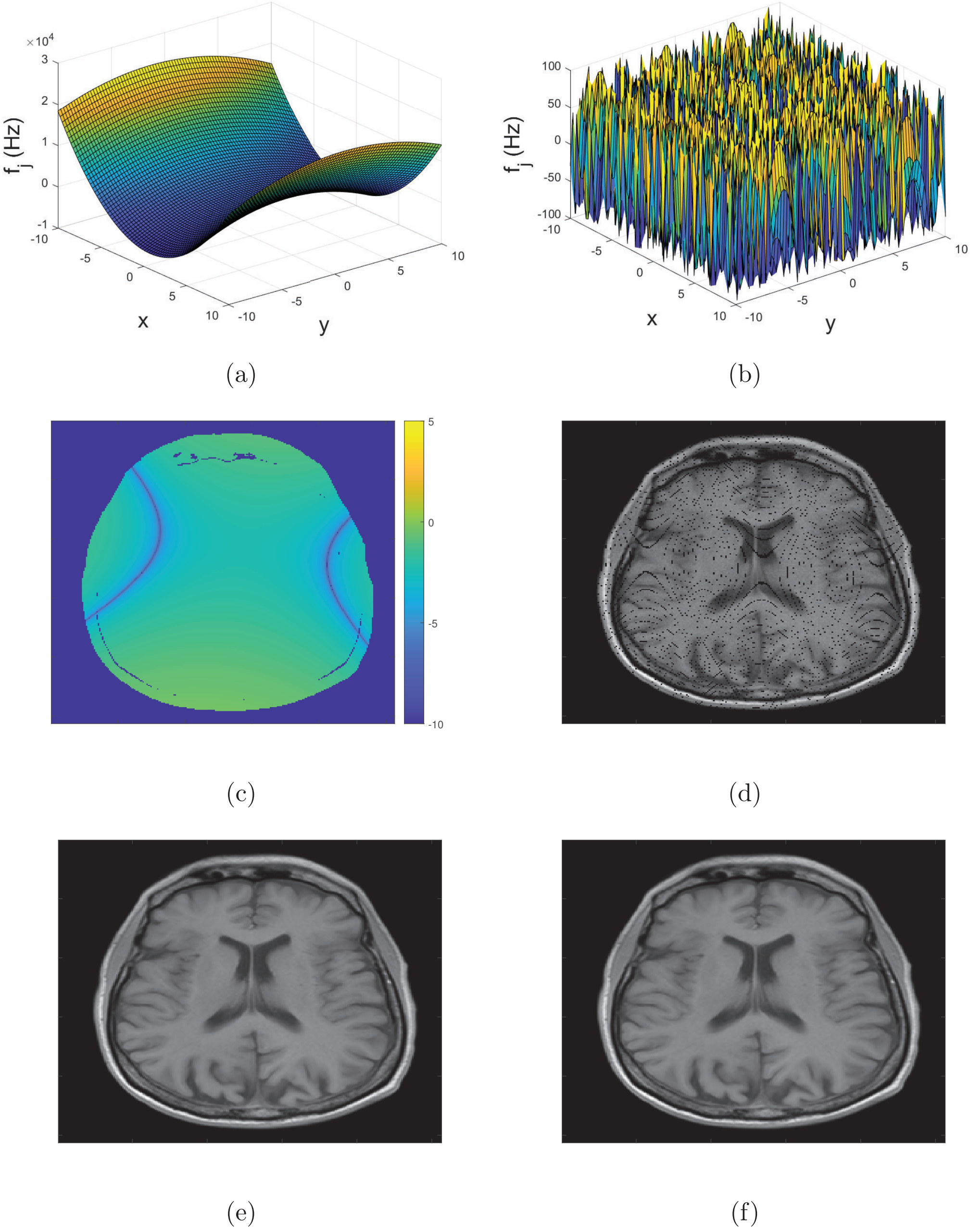
Consider that one coil conducts ten echoes with a time interval of 5 ms starting from 5 ms until 50 ms, where $R_j = 0$. The coil sensitivity is 1 in all positions for a slice of size 256×256. The surface of off-resonance frequencies is depicted in 6.3a. The maximum and minimum off-resonance frequencies are $23,081$ Hz and $-6,537$ Hz, respectively, where 16 harmonics up to the third order and listed in Table 6.1 are involved in generating the inhomogeneity. We note that off-frequencies are large such that wrapped phases are not equal to unwrapped phases, i.e., $\log(e^{\iota 2\pi f_j \Delta t_i})/\iota 2\pi \Delta t_i \neq f_j$.

To better show how the phase wrapping complicates the field map estimation, suppose that off-resonance frequencies are directly estimated from wrapped phases from (6.7). It is observed from Fig. 6.3b that one cannot derive the ground-truth surface of off-resonance frequencies from wrapped phase errors via $\log(e^{\iota 2\pi f_j \Delta t_i})/\iota 2\pi \Delta t_i$. Therefore, phase unwrapping is required. We consider that the SNR level of measurements is 15 dB. Suppose that the inhomogeneity is zero in $(0, 0, 0)$. The proposed method starts from an inner square patch in the image from row 114 to 137 and from column 114 to 137. We optimize image voxels and phase errors in these positions first and approximate the coefficients for spherical harmonics. After we predict unwrapped phase errors in all positions, we slowly increase the number of voxels that the proposed algorithm processes. The proposed algorithm increases the considered set of voxels after each prediction. In particular, the length and width of the considered square increase by 20 in each iteration. The logarithm of the residual errors of estimated off-resonance frequencies via the proposed approach is depicted in Fig. 6.3c. The estimated image derived from initialized phase errors is depicted in Fig. 6.3d. The estimated image via the proposed approach is depicted in Fig. 6.3e, and the ground-truth image is depicted in Fig. 6.3f.

We compare the performance of our approach against [1] using a 2D brain dataset with nine axial 256×256 slices and one coil, where 10 echoes at $t_i = \{6, 8, 10, 12, 14, 16, 18, 20, 22, 24\}$ ms are conducted. The SNR level of measurements is 20 dB, where the noise is complex Gaussian. Since [1] does not address the phase wrapping issue, we consider that the magnetic field inhomogeneity does not cause any phase wrappings. The maximum and minimum off-resonance frequencies in the surface 6.4a are 62 Hz and -48 Hz, respectively. The logarithm of the residual error of the estimated off-resonance frequencies via initialization, the proposed method, and the method in [1] are depicted in Figs. 6.4b, 6.4c, and 6.4d. The average accuracy of returned off-resonance frequencies via [1] and our approach are 0.2091 and 0.0224, respectively, while the average accuracy of initialized off-resonance frequencies is 0.2831.

6.7 Concluding Remarks

In this chapter, we proposed a BCD algorithm for the joint optimization of 3D field map and image voxels from multiple 2D slices. We alternatively optimized voxel intensities and phase errors in a subset of voxels, which is chosen such that the unwrapped phase can be accurately estimated. We showed that the optimization subproblem with respect to the voxel intensity is strongly convex, for which solutions can be obtained via a closed-form. Furthermore, the subproblem with respect to phase errors is neither convex nor concave. However, the global solution for the wrapped phase error can be calculated via a golden-section search nested inside a grid-search. We leveraged the fact that the wrapped phase equals the unwrapped phase in voxels near the magnet center, and regularized the derived field map by 3D spherical harmonics up to the third order that model inhomogeneity. Iteratively we unwrapped phase errors and gradually expanded the size of the considered voxel subset until we include all image voxels. We demonstrated the superiority of our method relative to the existing methods via numerical simulations.



(a)

(b)

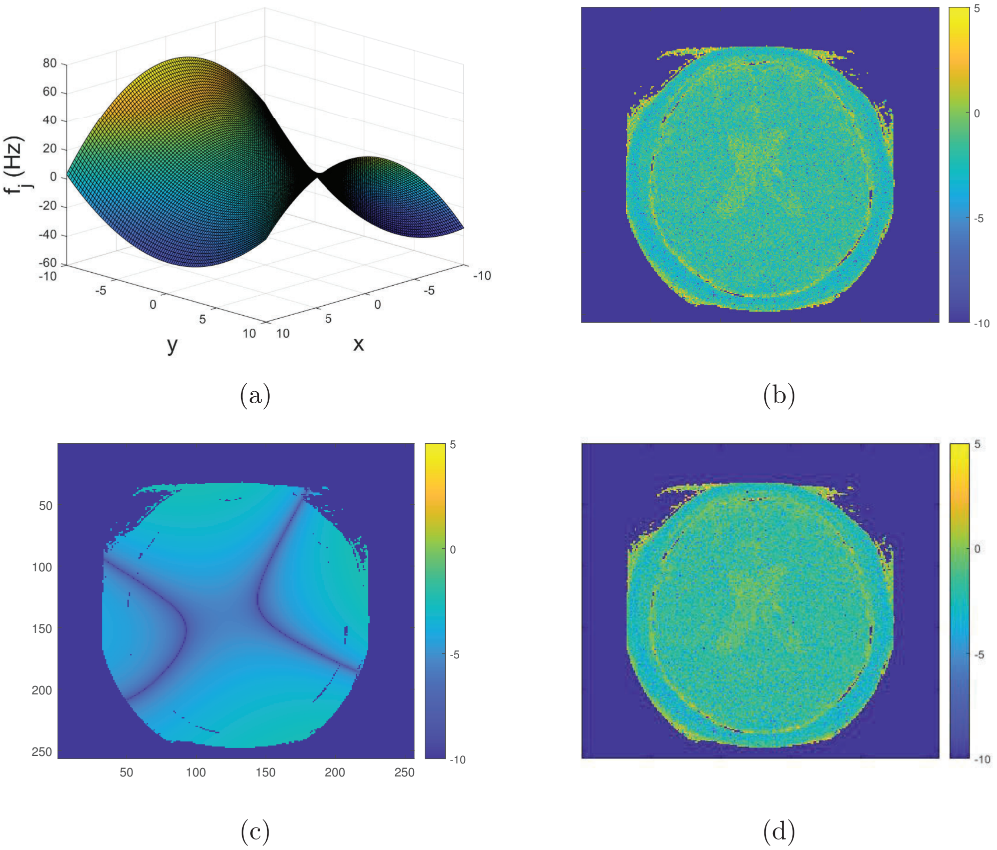
(c)

(d)

(e)

(f)

Figure 6.3: One example of joint image and field map for one 2D slice. (a) The surface of off-resonance frequencies. (b) Estimated off-resonance frequencies from wrapped phases extracted from a noise-free image. (c) The logarithm of the residual error of the off-resonance frequencies. (d) Initial image where the SNR level is 15 dB. (e) Reconstructed image. (f) The ground-truth image.



(a)

(b)

(c)

(d)

Figure 6.4: Comparison of the proposed algorithm against [1], where the ground-truth surface of off-resonance frequencies is depicted in (a). The logarithm of the residual error of the estimated off-resonance frequencies via initialization, the proposed method, and the method in [1] are depicted in (b), (c), and (d), respectively.

Chapter 7

Stochastic Gradient Descent Learns Linear Dynamical Systems from A Single Trajectory

Consider the linear time-invariant dynamical system giving rise to a single (or multiple) finite trajectory of noisy outputs $\{\mathbf{y}_t\}_t$, described by the following dynamics:

$$\mathbf{h}_{t+1} = \mathbf{A}\mathbf{h}_t + \mathbf{B}\mathbf{u}_t, \tag{7.1a}$$

$$\mathbf{y}_t = \mathbf{C}\mathbf{h}_t + \mathbf{D}\mathbf{u}_t + \boldsymbol{\zeta}_t, \tag{7.1b}$$

where $\mathbf{h}_t$, $\mathbf{u}_t$ and $\boldsymbol{\zeta}_t$ represent the hidden state, the input and the noise of the measurement at time instance t, respectively. Here, the weight matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ parameterize the system; we consider these as unknowns here.

In the system described in (7.1a)-(7.1b), the hidden state $\mathbf{h}_t$ cannot be measured. Instead, the system is indirectly measured from outputs. System identification for (7.1)–the problem of identifying the unknown weight matrices (or a set of weight matrices giving identical dynamics)–is involved in a wide variety of time-series analyses, robotics, economics, and modern control problems. Examples include text translation, time-series predictions, speech recognition, and many others [182, 183, 184, 185, 186].

Besides their vital applications in the control theory, there is recent interest from the machine learning community in linear dynamical systems due to their connections with recurrent neural networks (RNNs). Indeed, similar to linear dynamical systems, RNNs process the inputs to the system using their internal hidden states [187, 188]. Explorations of the connections between the linear dynamical systems and RNNs are fairly recent (see the aforementioned, as well as [189, 187, 190, 191]), and elucidating these connections plays a critical role in better understanding RNNs, such as long short-term memories (LSTMs) and gated recurrent units (GRUs), which have achieved significant success in different applications.

7.1 Prior Work

As a means of placing this work in proper context in the broader literature, we classify relevant work to this paper into two major domains: the papers that study linear dynamical systems and papers that study RNNs.

Linear Dynamical Systems. A rich literature exists in control and systems theory on the identification of linear dynamical systems; see, e.g., [192, 193, 194, 195]. More recent literature concentrates on data-driven approaches and provides sample complexity bounds, such as [196, 188, 2, 197, 198, 199, 200, 201, 202, 203, 204, 205]. Given noisy observations generated by a discrete linear dynamical system, a gradient projection approach is proposed in [188] to minimize the population risk of learning an unknown, stable, single-input and single-output (SISO) system at a sublinear convergence rate. If $q(z) = z^n + a_1 z^{n-1} + \cdots + a_n$ is the characteristic polynomial of the system, [188] assumes that $\{a_i\}_{i=1}^n$ are such that the real and imaginary parts of $q(z)/z^n$ satisfy $\Re(q(z)/z^n) > |\Im(q(z)/z^n)|$ for any z, where $|z| = 1$. The gradient approach in [188] fails if the $\{a_i\}_{i=1}^n$ in the characteristic polynomial of the underlying system do not satisfy the above assumption, and when n increases, the above assumption becomes more difficult to be satisfied. The SISO results are extended in [188] to multiple-input and multiple-output (MIMO) systems, where unknown transformation matrices $\mathbf{A}$ and $\mathbf{B}$ have the Brunovsky canonical form.

Learning unknown weight matrices of an observable and controllable stable linear dynamical system is studied in [2]. Unlike [188] that updates the estimation of unknown weight matrices in each iteration using a subset of samples, the approach given in [2] processes all samples at the same time. In particular, there a set of T Markov parameters of the system, denoted by $\mathbf{\Theta}_T$, are first estimated in [2]. Then, a Ho-Kalman algorithm that uses SVD is proposed to estimate the weight matrices from the estimated Hankel matrix. Although the identified weight matrices by [2] build an equivalent system that has an identical performance to the unknown system, the weight matrices are not necessarily equal to those for the underlying system. In [203], the authors provide complexity bounds for the estimated Markov parameters, where a prefiltered least squares approach is proposed to mitigate the effect of truncated Markov parameters and the measurement noise. Similar to [2], [203, 198, 201, 204, 205] use Ho-Kalman type algorithms. The drawback of these approaches is that the size of the Hankel matrix increases quadratically with the number of estimated Markov parameters, which increases the cost of the corresponding SVDs. The estimation errors decay at a rate of $1/\sqrt{N}$ in [2, 198, 201, 204, 205], where N denotes the trajectory length.

Different assumptions about the stability, system order and the number of required trajectories to excite the unknown system are made in existing papers. When the spectral radius of $\mathbf{A}$ is less than one, i.e., $\rho(\mathbf{A}) < 1$, the linear dynamical system becomes stable. It is marginally stable and unstable if $\rho(\mathbf{A}) \leq 1$ and $\rho(\mathbf{A}) > 1$, respectively. Table 7.1 summarizes different assumptions in existing papers. The approaches in the above papers require all input-output samples to be stored in the memory, which makes them (potentially) memory inefficient. Furthermore, the approaches explained above are not necessarily scalable since they simultaneously process all samples of one (or multiple) trajectory to learn weight matrices. From the last column of Table 7.1, we observe that only [188] provides conditions and guarantees for its proposed algorithm to converge to the ground truth weight matrices.

If the dynamics of a system can be fully described only by (7.1a) and the system output is generated by $\mathbf{y}_t = \mathbf{h}_{t+1}$, the system is directly measured. Unlike the above papers that address the identification of indirectly measured systems, a number of papers study directly measured systems from a single trajectory. In [200] and [197], the estimation of $\mathbf{A}$ from a system with dynamic $\mathbf{h}_{t+1} = \mathbf{A}\mathbf{h}_t + \mathbf{u}_t$ is studied. Similarly, [202] studied estimating $\mathbf{A}$ and $\mathbf{B}$ from $\mathbf{h}_{t+1} = \mathbf{A}\mathbf{h}_t + \mathbf{B}\mathbf{u}_t + \boldsymbol{\zeta}_t$ via a regression method; where error bounds are provided. The same dynamics are considered in [199], where $\mathbf{A}$ is unknown and $\mathbf{B}$ is considered to

Table 7.1: A summary of recent non-asymptotic analysis for LTI system learning

Paper	**Known order**	Meas.	Type	Stability	**# of trajectories**	Online	Inputs	Sample complexity	
								$\mathbf{\Theta}_T$	$\mathbf{A}, \mathbf{B}$ $\mathbf{C}, \mathbf{D}$
[203]	No	Indirect	MIMO	$\rho(\mathbf{A}) \leq 1$	Single	No	Gaussian	$\mathcal{O}(\frac{1}{\sqrt{N}})$	—
[2]	Yes	Indirect	MIMO	$\rho(\mathbf{A}) < 1$	Single	No	Gaussian	$\mathcal{O}(\frac{1}{\sqrt{N}})$	—
[198]	Yes	Indirect	MIMO	$\rho(\mathbf{A}) < 1$	Single	No	Gaussian	$\mathcal{O}(\frac{1}{\sqrt{N}})$	—
[201]	Yes	Indirect	MIMO	$\rho(\mathbf{A}) \leq 1$	Single	No	Gaussian	$\mathcal{O}(\frac{1}{\sqrt{N}})$	—
[188]	Yes	Indirect	MIMO	$\rho(\mathbf{A}) < 1$	Multiple	No	Gaussian	—	$\mathcal{O}(\frac{1}{\sqrt{N}})$
[204]	No	Indirect	MIMO	Any	Multiple	No	Gaussian	$\mathcal{O}(\frac{1}{\sqrt{N}})$	—
[205]	No	Indirect	MIMO	Any	Multiple	No	Gaussian	$\mathcal{O}(\frac{1}{\sqrt{N}})$	—
This paper	Yes	Indirect	MIMO	$\rho(\mathbf{A}) < 1$	Single	Yes	Gaussian	$\mathcal{O}(\frac{1}{\sqrt{N}})$	$\mathcal{O}(\frac{1}{\sqrt{N}})$
[199]	Yes	Direct	MIMO	$\rho(\mathbf{A}) < 1$	Single	No	Controlled	—	$\mathcal{O}(\frac{1}{\sqrt{N}})$
[202]	Yes	Direct	MIMO	Any	Single	No	Gaussian	—	$\mathcal{O}(\frac{1}{\sqrt{N}})$
[197]	Yes	Direct	MIMO	Any	Single	No	Gaussian	—	—
[200]	Yes	Direct	MIMO	$\rho(\mathbf{A}) \leq 1$	Single	No	Gaussian	—	$\mathcal{O}(\frac{1}{\sqrt{N}})$

be known. It is proven in [199] that the estimation of $\mathbf{A}$ can be accelerated if inputs are controlled rather than merely being Gaussian.

Recurrent Neural Networks. It is common to consider RNNs as non-linear dynamical systems. A growing number of papers have recently studied training RNNs and provided theoretical guarantees for the problem. The connections between RNNs and state equations of simple dynamical systems are characterized in [190, 187, 191], where a neural network architecture is proposed to capture long-term dependencies enabled by the stability property of its underlying differential equation. In [187], a discrete-time dynamical system controlled by the state equation is considered, and an SGD algorithm is proposed to learn weight matrices of the state equation when the output layer activation function is a leaky rectified linear unit (ReLU). The approach in [187] is extended in [191], where the noise of measurements is also considered in the recursion dynamics. In [206, 207], similar dynamical systems managed by the state equation given in [187] are studied with different activation functions. In [206], the activation function is hyperbolic tangent, however, it is differentiable and strongly convex in [207]. To prove the convergence of the proposed algorithms in the above papers for learning unknown weight matrices, it is assumed that the hidden state of the system is observable. In practice, however, large RNNs with complex state evolutions are required to increase the representation power of networks. When $\mathbf{h}_{t+1} = \tanh(\mathbf{A}\mathbf{h}_t + \mathbf{B}\mathbf{u}_t)$ is considered instead of (7.1a), a particular class of RNNs is obtained. The identification of this class via a non-linear regression is studied in [208]. With continually running the RNN and implementing a gradient method to update $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ from a non-linear regression, [208] shows that a local minima of the problem can be obtained.

7.2 Contributions

We study the identification of a stable linear dynamical system based on a single sequence of input-output pairs. We formulate a finite sum problem to efficiently learn the truncated Markov parameters of the system. The formulated problem becomes strongly convex when the system input is white Gaussian noise. We prove that the sequence length strictly decreases the Frobenius norm distance between the regression solution and the truncated ground truth Markov parameters with a rate of $1/\sqrt{N}$. However, when the trajectory length increases, the complexity of solving the finite sum problem via the pseudo-inverse method increases. We propose low iteration cost online and offline stochastic gradient descent (SGD) algorithms to efficiently learn truncated Markov parameters. The offline SGD algorithms works on a batch of input-output pairs; however, the online SGD uses the most recent input-output pair to implement a gradient step in an online streaming fashion and then discards it. Therefore, it is storage efficient as compared with the existing methods. Via novel complexity bounds, we prove that when the system input is Gaussian, the proposed offline SGD algorithm *linearly* converges in expectation to the finite sum solution. Unlike full-batch methods in [188, 2, 198, 201], an update step in our SGD algorithms is simply implemented via one input-output pair rather than a trajectory.

When the unknown weight matrices have *Brunovsky canonical form*, which is perhaps the most widely used form in control theory [209, 210, 211], we propose a novel approach to uniquely identify the ground truth weight matrices from a linear system of equations formulated based on the SGD iterates and the transfer function of the linear dynamical system. This is unlike widely used Ho-Kalman methods in [203, 2, 198, 201, 204, 205] that estimate some weight matrices to find a system with an equivalent performance. We solve the proposed linear system of equations in each iteration of the SGD algorithms. We use the derived bounds for the proposed offline SGD algorithm to develop complexity bounds for the identification of unknown weight matrices. We provide guarantees that the estimated weight matrices from the proposed linear system built from SGD iterates *linearly* converge in expectation to the ground truth values. Extensive numerical tests confirm the linear convergence of proposed approaches and demonstrate that they outperform the existing state of the art methods.

The rest of this chapter is organized as follows. The problem formulation is given in Section 7.3. Section 7.4 describes SGD algorithms to solve the formulated linear regression. In Section 7.5, we relate Markov parameters of the system to unknown weight matrices and propose a linear system of equations inside SGD methods. The online pseudo-inverse approach is summarized in Section 7.6. The simulation results are given in Section 7.7, and concluding remarks are given in Section 7.8.

7.3 Problem Setup

Notation. Bold upper-case and lower-case letters are used to denote matrices and vectors, respectively. The trace of matrix $\mathbf{M}$ is denoted by $\mathrm{Tr}(\mathbf{M})$. $\mathbf{M}'$ denotes the transpose of $\mathbf{M}$. Given a matrix $\mathbf{M}$, $\|\mathbf{M}\|_F$ denotes the Frobenius norm, and given a vector $\mathbf{m}$, $\|\mathbf{m}\|_2$ denotes the ℓ_2-norm. A diagonal matrix is denoted by $\mathbf{M}_{\mathbf{m}}$, where elements of $\mathbf{m}$ are on the diagonal. The vector of elements of $\mathbf{m}$ raised to power 2 is denoted by $\mathbf{m}^{\cdot 2}$. We denote $(i,j)^{\text{th}}$ element of $\mathbf{M}$ by $[\mathbf{M}]_{ij}$. The spectral radius of matrix $\mathbf{M}$ is denoted by $\rho(\mathbf{M})$ and its spectral norm is denoted by $\|\mathbf{M}\|_2$. The Hermitian adjoint of $\mathbf{M}$ is denoted by $\mathbf{M}^H$.

Setup. As alluded above, we consider a *time-invariant* linear dynamical system characterized by matrices $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times m}$, $\mathbf{C} \in \mathbb{R}^{p\times n}$ and $\mathbf{D} \in \mathbb{R}^{p\times m}$ as follows:

$$\begin{aligned}\mathbf{h}_{t+1} &= \mathbf{A}\mathbf{h}_t + \mathbf{B}\mathbf{u}_t,\\ \mathbf{y}_t &= \mathbf{C}\mathbf{h}_t + \mathbf{D}\mathbf{u}_t + \boldsymbol{\zeta}_t,\end{aligned}$$

where $\mathbf{u}_t$ is an external control input vector at time instance t, $\mathbf{y}_t$ is the vector of system outputs, and $\boldsymbol{\zeta}_t$ is the noise of measurement. In the above model, the hidden state is denoted by $\mathbf{h}_t$, and n is called the *order of the system.* In addition, $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ are unknown transformation matrices. We assume that the system is stable, and thus, $\rho(\mathbf{A}) < 1$. Furthermore, we assume matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ have bounded Frobenius norms. Based on one sequence of input-output pairs $\{\mathbf{u}_t, \mathbf{y}_t\}_{t=1}^N$, and assuming n is known (similar to [2, 188, 187]), we aim to learn the unknown matrices and characterize complexity bounds for the accuracy of the estimated unknowns.

Consider that T is a finite time horizon. Each $y_t, t \geq T-1$, can be expanded recursively using $\mathbf{u}_t, \dots, \mathbf{u}_{t-T+1}$ and $\mathbf{h}_{t-T+1}$ as follows:

$$\mathbf{y}_t = \textstyle\sum_{i=1}^{T-1} \mathbf{CA}^{i-1}\mathbf{B}\mathbf{u}_{t-i} + \mathbf{D}\mathbf{u}_t + \boldsymbol{\zeta}_t + \mathbf{CA}^{T-1}\mathbf{h}_{t-T+1}, \tag{7.2}$$

when $\mathbf{A}\mathbf{h}_{t-1} + \mathbf{B}\mathbf{u}_{t-1}$ is substituted for each $\mathbf{h}_t, t \in \{t, \dots, t-T+2\}$. Suppose that the input signal $\mathbf{u}_t$ at each time instance is random and follows a normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{\boldsymbol{\sigma}^{\cdot 2}})$, where $\mathbf{\Sigma}_{\boldsymbol{\sigma}^{\cdot 2}}$ is the covariance matrix. Furthermore, $\boldsymbol{\zeta}_t$ also follows a normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{\boldsymbol{\sigma}_\zeta^{\cdot 2}})$ and $\{\boldsymbol{\zeta}_t\}_t$ is independent of $\{\mathbf{u}_t\}_t$. Let $\mathbf{x}_t \in \mathbb{R}^{mT \times 1}$ denote a finite sequence of inputs with length T as follows:

$$\mathbf{x}_t = \begin{cases} [\mathbf{u}_t' \;\; \mathbf{u}_{t-1}' \;\; \mathbf{u}_{t-2}' \;\; \dots \;\; \mathbf{u}_1' \;\; \mathbf{0} \;\; \dots \;\; \mathbf{0}]', & \text{if} \quad t < T, \\ [\mathbf{u}_t' \;\; \mathbf{u}_{t-1}' \;\; \mathbf{u}_{t-2}' \;\; \dots \;\; \mathbf{u}_{t-T+1}']', & \text{if} \quad t \geq T. \end{cases} \tag{7.3}$$

Using $\mathbf{x}_t$, we rewrite (7.2) as follows:

$$\mathbf{y}_t = \underbrace{[\mathbf{D} \;\;\; \mathbf{CB} \;\;\; \mathbf{CAB} \;\;\; \dots \;\;\; \mathbf{CA}^{T-2}\mathbf{B}]}_{\mathbf{\Theta}_T} \mathbf{x}_t + \mathbf{CA}^{T-1}\mathbf{h}_{t-T+1} + \boldsymbol{\zeta}_t.$$

In the above equation, $\mathbf{h}_{t-T+1}$ is a linear combination of inputs and the initial state. In the following lemma, we bound the Frobenius norm of $\mathbf{CA}^{T-1}\mathbf{h}_{t-T+1}$.

Lemma 1. *Suppose that* $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$ *is the eigenvalue decomposition for* $\mathbf{A}$. *We bound the norm of* $\mathbf{CA}^{T-1}\mathbf{h}_{t-T+1}$ *when* $t \geq T$ *as follows:*

$$\begin{aligned} &\mathbb{E}_{\mathbf{u}}[\left\|\mathbf{CA}^{T-1}\mathbf{h}_{t-T+1}\right\|_2^2] \leq n^2\ell \left\|\mathbf{C}\right\|_F^2 \rho(\mathbf{A})^{2(T-1)} \\ &\times \left[n^2\ell\rho(\mathbf{A})^2 \left\|\mathbf{h}_0\right\|_2^2 + \frac{n^2\ell m \max(\boldsymbol{\sigma}^{\cdot 2})\rho(\mathbf{A})^2 \left\|\mathbf{B}\right\|_F^2}{1-\rho(\mathbf{A})^2} + m\max(\boldsymbol{\sigma}^{\cdot 2}) \left\|\mathbf{B}\right\|_F^2 \right], \end{aligned} \tag{7.4}$$

where $\ell = \left\|\mathbf{V}^{-1}\right\|_F$.

Proof. First, let us bound $\|\mathbf{h}_t\|_2^2$. Since $\mathbf{h}_t$ is a linear combination of system inputs, we have

$$\mathbf{h}_t = \mathbf{A}^t\mathbf{h}_0 + \sum_{i=1}^{t} \mathbf{A}^{i-1}\mathbf{B}\mathbf{u}_{t-i}. \tag{7.5}$$

We bound $\|\mathbf{h_{t+1}}\|_2^2$ using recursion as follows:

$$\begin{aligned}\mathbb{E}_{\mathbf{u}}[\|\mathbf{h}_{t+1}\|_2^2] &= \mathbb{E}_{\mathbf{u}}[\|\mathbf{A}\mathbf{h}_t + \mathbf{B}\mathbf{u}_t\|_2^2] \overset{(a)}{=} \mathbb{E}_{\mathbf{u}}[\|\mathbf{A}\mathbf{h}_t\|_2^2] + \mathbb{E}_{\mathbf{u}}[\|\mathbf{B}\mathbf{u}_t\|_2^2] \\ &\leq \mathbb{E}_{\mathbf{u}}[\|\mathbf{A}\mathbf{h}_t\|_2^2] + \mathbb{E}_{\mathbf{u}}[\|\mathbf{u}_t\|_2^2]\,\|\mathbf{B}\|_2^2 \overset{(b)}{\leq} \mathbb{E}_{\mathbf{u}}[\|\mathbf{A}\mathbf{h}_t\|_2^2] + m(\max(\boldsymbol{\sigma}^{\cdot 2}))\,\|\mathbf{B}\|_F^2\,, \end{aligned} \tag{7.6}$$

where (a) follows due to the fact that $\mathbf{h}_t$ is made of $\{\mathbf{u}_i\}_{i=0}^{t-1}$ and the initial state $\mathbf{h}_0$, which are independent from $\mathbf{u}_t$. We have (b) since $\mathbb{E}_{\mathbf{u}}[\|\mathbf{u}_t\|_2^2] = \sum_{i=1}^{m} \boldsymbol{\sigma}^{\cdot 2}(i) \leq m(\max(\boldsymbol{\sigma}^{\cdot 2}))$.

Based on the update rule for the hidden state, one can expand $\mathbb{E}_{\mathbf{u}}[\|\mathbf{A}\mathbf{h}_t\|_2^2]$ recursively as $\mathbb{E}_{\mathbf{u}}[\|\mathbf{A}\mathbf{h}_t\|_2^2] = \mathbb{E}_{\mathbf{u}}[\|\mathbf{A}(\mathbf{A}\mathbf{h}_{t-1} + \mathbf{B}\mathbf{u}_{t-1})\|_2^2]$. Consider eigendecomposition for $\mathbf{A}$ as $\mathbf{A} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{-1}$. Using recursions (7.5) and (7.6), we bound the norm of the hidden state as follows:

$$\begin{aligned}
&\mathbb{E}_{\mathbf{u}}[\|\mathbf{h}_{t+1}\|_2^2] \leq \mathbb{E}_{\mathbf{u}}\left[\left\|\mathbf{A}^{t+1}\mathbf{h}_0 + \sum_{i=1}^{t}\mathbf{A}^{i}\mathbf{B}\mathbf{u}_{t-i}\right\|_2^2\right] + m\max(\boldsymbol{\sigma}^{\cdot 2})\,\|\mathbf{B}\|_F^2 \\
&\overset{(a)}{=} \mathbb{E}_{\mathbf{u}}\left[\left\|\mathbf{V}\boldsymbol{\Lambda}^{t+1}\mathbf{V}^{-1}\mathbf{h}_0 + \sum_{i=1}^{t}\mathbf{V}\boldsymbol{\Lambda}^{i}\mathbf{V}^{-1}\mathbf{B}\mathbf{u}_{t-i}\right\|_2^2\right] + m\max(\boldsymbol{\sigma}^{\cdot 2})\,\|\mathbf{B}\|_F^2 \\
&= \mathbb{E}_{\mathbf{u}}\left[\left\|\mathbf{V}\boldsymbol{\Lambda}^{t+1}\mathbf{V}^{-1}\mathbf{h}_0 + \sum_{i=1}^{t}\mathbf{V}\boldsymbol{\Lambda}^{i}\mathbf{V}^{-1}\mathbf{B}\mathbf{u}_{t-i}\right\|_2^2\right] + m\max(\boldsymbol{\sigma}^{\cdot 2})\,\|\mathbf{B}\|_F^2 \\
&= \mathbb{E}_{\mathbf{u}}\left[\left\|\mathbf{V}\boldsymbol{\Lambda}^{t+1}\mathbf{V}^{-1}\mathbf{h}_0\right\|_2^2 + \sum_{i=1}^{t}\left\|\mathbf{V}\boldsymbol{\Lambda}^{i}\mathbf{V}^{-1}\mathbf{B}\mathbf{u}_{t-i}\right\|_2^2\right] + m\max(\boldsymbol{\sigma}^{\cdot 2})\,\|\mathbf{B}\|_F^2 \\
&\overset{(b)}{\leq} n^2\ell\rho(\mathbf{A})^{2(t+1)}\,\|\mathbf{h}_0\|_2^2 + n^2\ell m\max(\boldsymbol{\sigma}^{\cdot 2})\,\|\mathbf{B}\|_F^2\sum_{i=1}^{t}\rho(\mathbf{A})^{2i} + m\max(\boldsymbol{\sigma}^{\cdot 2})\,\|\mathbf{B}\|_F^2 \\
&= n^2\ell\rho(\mathbf{A})^{2(t+1)}\,\|\mathbf{h}_0\|_2^2 + \frac{n^2\ell m\max(\boldsymbol{\sigma}^{\cdot 2})\rho(\mathbf{A})^2(1-\rho(\mathbf{A})^{2t})\,\|\mathbf{B}\|_F^2}{1-\rho(\mathbf{A})^2} + m\max(\boldsymbol{\sigma}^{\cdot 2})\,\|\mathbf{B}\|_F^2 \\
&\leq n^2\ell\rho(\mathbf{A})^{2}\,\|\mathbf{h}_0\|_2^2 + \frac{n^2\ell m\max(\boldsymbol{\sigma}^{\cdot 2})\rho(\mathbf{A})^2\,\|\mathbf{B}\|_F^2}{1-\rho(\mathbf{A})^2} + m\max(\boldsymbol{\sigma}^{\cdot 2})\,\|\mathbf{B}\|_F^2\,,
\end{aligned} \tag{7.7}$$

where (a) follows from eigendecomposition for $\mathbf{A}$. In (b), $\ell = \left\|\mathbf{V}^{-1}\right\|_F^2$. Using the above upper-bound for $\mathbb{E}_{\mathbf{u}}[\|\mathbf{h}_{t+1}\|_2^2]$, we bound $\mathbb{E}_{\mathbf{u}}[\left\|\mathbf{C}\mathbf{A}^{T-1}\mathbf{h}_{t-T+1}\right\|_2^2]$ as follows:

$$\begin{aligned}
&\mathbb{E}_{\mathbf{u}}[\left\|\mathbf{C}\mathbf{A}^{T-1}\mathbf{h}_{t-T+1}\right\|_2^2] \leq \|\mathbf{C}\|_F^2 \left\|\mathbf{A}^{T-1}\right\|_F^2 \mathbb{E}_{\mathbf{u}}[\|\mathbf{h}_{t-T+1}\|_2^2] \\
&\leq \|\mathbf{C}\|_F^2 \left\|\mathbf{V}\boldsymbol{\Lambda}^{T-1}\mathbf{V}^{-1}\right\|_F^2 \mathbb{E}_{\mathbf{u}}[\|\mathbf{h}_{t-T+1}\|_2^2] \\
&\leq n^2\ell \|\mathbf{C}\|_F^2 \rho(\mathbf{A})^{2(T-1)}\mathbb{E}_{\mathbf{u}}[\|\mathbf{h}_{t-T+1}\|_2^2] \\
&\leq n^2\ell \|\mathbf{C}\|_F^2 \rho(\mathbf{A})^{2(T-1)} \\
&\times \underbrace{\left[n^2\ell\rho(\mathbf{A})^2 \|\mathbf{h}_0\|_2^2 + \frac{n^2\ell m \max(\boldsymbol{\sigma}^{\cdot 2})\rho(\mathbf{A})^2 \|\mathbf{B}\|_F^2}{1-\rho(\mathbf{A})^2} + m\max(\boldsymbol{\sigma}^{\cdot 2}) \|\mathbf{B}\|_F^2\right]}_{\gamma}.
\end{aligned}$$

When the truncation length T is large enough, $\lim_{T\to\infty} \rho(\mathbf{A})^{2(T-1)} = 0$ since $\rho(\mathbf{A}) < 1$. For a large T, we find $\mathbb{E}_{\mathbf{u}}\left[\left\|\mathbf{C}\mathbf{A}^{T-1}\mathbf{h}_{t-T+1}\right\|_2^2\right] \approx 0$, or equivalently, $\lim_{T\to\infty} \mathbb{E}_{\mathbf{u}}\left[\left\|\mathbf{C}\mathbf{A}^{T-1}\mathbf{h}_{t-T+1}\right\|_2^2\right] = 0$. □

From Lemma 1, one can observe that the resulted error from truncation, $\left\|\mathbf{C}\mathbf{A}^{T-1}\mathbf{h}_{t-T+1}\right\|_2^2$, decreases exponentially with the truncation length T. Thus, the error becomes very small for a large enough T. To reconstruct the system output $\mathbf{y}_t$, it is essentially enough to identify

$$\boldsymbol{\Theta}_T = [\mathbf{D} \quad \mathbf{C}\mathbf{B} \quad \mathbf{C}\mathbf{A}\mathbf{B} \quad \mathbf{C}\mathbf{A}^2\mathbf{B} \quad \ldots \quad \mathbf{C}\mathbf{A}^{T-2}\mathbf{B}],$$

where the size of the above unknown matrix is $p \times m\,T$. We notice that $\boldsymbol{\Theta}_T$ incorporates the first T Markov parameters; the first one is $\mathbf{D}$ and the rest are $\{\mathbf{C}\mathbf{A}^i\mathbf{B}\}_{i=0}^{T-2}$. To estimate $\boldsymbol{\Theta}_T$, we use a regression approach and formulate the following optimization:

$$\hat{\boldsymbol{\Theta}}_T = \arg\min_{\hat{\boldsymbol{\Theta}}_T} \lim_{N\to\infty} \frac{1}{2N} \sum_{t=1}^{N} \left\|\mathbf{y}_t - \hat{\boldsymbol{\Theta}}_T\mathbf{x}_t\right\|_2^2. \tag{7.8}$$

The above problem is strongly convex in $\hat{\boldsymbol{\Theta}}_T$ since the Hessian matrix (or the covariance of the inputs) $\lim_{N\to\infty} \frac{1}{N}\sum_{t=1}^{N} \mathbb{E}(\mathbf{x}_t\mathbf{x}_t')$ is positive definite. This means that a unique solution is attained from the above minimization problem.

Proposition 10. *The solution* $\hat{\mathbf{\Theta}}_T$ *from* (7.8) *is identical to the ground truth* $\mathbf{\Theta}_T$ *in spite of the noise of measurements and excluding the hidden state transformation* $\mathbf{CA}^{T-1}\mathbf{h}_{t-T+1}$ *in* (7.8)*.*

Proof. From the least squares problem, we have:

$$\begin{aligned}
\hat{\mathbf{\Theta}}_T =& \arg\min_{\hat{\mathbf{\Theta}}_T} \lim_{N\to\infty} \frac{1}{2N}\sum_{t=1}^{N} \left\|\mathbf{y}_t - \hat{\mathbf{\Theta}}_T\mathbf{x}_t\right\|_2^2 \\
&= \arg\min_{\hat{\mathbf{\Theta}}_T} \lim_{N\to\infty} \frac{1}{2N}\sum_{t=1}^{N} \left\|\mathbf{\Theta}_T\mathbf{x}_t + \mathbf{CA}^{T-1}\mathbf{h}_{t-T+1} + \boldsymbol{\zeta}_t - \hat{\mathbf{\Theta}}_T\mathbf{x}_t\right\|_2^2 \\
&= \arg\min_{\hat{\mathbf{\Theta}}_T} \lim_{N\to\infty} \frac{1}{2N}\sum_{t=1}^{N} \Bigg[\left\|\mathbf{\Theta}_T\mathbf{x}_t - \hat{\mathbf{\Theta}}_T\mathbf{x}_t\right\|_2^2 + \left\|\mathbf{CA}^{T-1}\mathbf{h}_{t-T+1} + \boldsymbol{\zeta}_t\right\|_2^2 \\
&+ 2\mathrm{Tr}\left(\mathbf{x}_t'(\mathbf{\Theta}_T - \hat{\mathbf{\Theta}}_T)'(\mathbf{CA}^{T-1}\mathbf{h}_{t-T+1} + \boldsymbol{\zeta}_t)\right)\Bigg] \\
&\overset{(a)}{=} \arg\min_{\hat{\mathbf{\Theta}}_T} \lim_{N\to\infty} \frac{1}{2N}\sum_{t=1}^{N} \left\|(\mathbf{\Theta}_T - \hat{\mathbf{\Theta}}_T)\mathbf{x}_t\right\|_2^2 .
\end{aligned} \tag{7.9}$$

Before justifying (a), we notice that:

1. From Lemma 1, we know that $\left\|\mathbf{CA}^{T-1}\mathbf{h}_{t-T+1}\right\|_2^2$ tends to zero when T increases.
2. We have $\lim_{N\to\infty} \frac{1}{2N}\sum_{t=1}^{N} \boldsymbol{\zeta}_t = \mathbf{0}$ with probability one due to Chebyshev's inequality [212].

In light of the above arguments, $\lim_{N\to\infty} \frac{1}{2N}\sum_{t=1}^{N} \mathrm{Tr}\left(\mathbf{x}_t'(\mathbf{\Theta}_T - \hat{\mathbf{\Theta}}_T)'(\mathbf{CA}^{T-1}\mathbf{h}_{t-T+1} + \boldsymbol{\zeta}_t)\right)$ can be made as small as desired.

The Hessian matrix for $\lim_{N\to\infty} \frac{1}{2N}\sum_{t=1}^{N} \left\|(\mathbf{\Theta}_T - \hat{\mathbf{\Theta}}_T)\mathbf{x}_t\right\|_2^2$ is $\lim_{N\to\infty} \frac{1}{2N}\sum_{t=1}^{N} \mathbf{x}_t\mathbf{x}_t'$, which is positive definite with probability one due to Chebyshev's inequality, and thus, the solution for $\mathbf{\Theta}_T$ in (7.9) is unique. Based on the above arguments, we observe that $\hat{\mathbf{\Theta}}_T = \mathbf{\Theta}_T$. □

Remark 9. *The Markov parameters of the system can be learned from* (7.8) *if the process noise is considered in* (7.1a)*.*

Proof. Instead of (7.1a), consider the following dynamics:

$$\begin{aligned}\mathbf{h}_{t+1} &= \mathbf{A}\mathbf{h}_t + \mathbf{B}\mathbf{u}_t + \boldsymbol{\varsigma}_t,\\ \mathbf{y}_t &= \mathbf{C}\mathbf{h}_t + \mathbf{D}\mathbf{u}_t + \boldsymbol{\zeta}_t,\end{aligned}$$

where $\boldsymbol{\varsigma}_t$ is the process noise at time instance t. We have:

$$\begin{aligned}\hat{\boldsymbol{\Theta}}_T =& \arg\min_{\hat{\boldsymbol{\Theta}}_T} \lim_{N\to\infty} \frac{1}{2N}\sum_{t=1}^{N} \left\|\mathbf{y}_t - \hat{\boldsymbol{\Theta}}_T\mathbf{x}_t\right\|_2^2\\ &= \arg\min_{\hat{\boldsymbol{\Theta}}_T} \lim_{N\to\infty} \frac{1}{2N}\sum_{t=1}^{N} \left\|\boldsymbol{\Theta}_T\mathbf{x}_t + \mathbf{C}\mathbf{A}^{T-1}\mathbf{h}_{t-T+1} + \boldsymbol{\zeta}_t + \mathbf{C}\sum_{i=1}^{T-1}\mathbf{A}^{i-1}\boldsymbol{\varsigma}_{t-i} - \hat{\boldsymbol{\Theta}}_T\mathbf{x}_t\right\|_2^2\\ &= \arg\min_{\hat{\boldsymbol{\Theta}}_T} \lim_{N\to\infty} \frac{1}{2N}\sum_{t=1}^{N} \Bigg[\left\|\boldsymbol{\Theta}_T\mathbf{x}_t - \hat{\boldsymbol{\Theta}}_T\mathbf{x}_t\right\|_2^2 + \left\|\mathbf{C}\mathbf{A}^{T-1}\mathbf{h}_{t-T+1} + \boldsymbol{\zeta}_t + \mathbf{C}\sum_{i=1}^{T-1}\mathbf{A}^{i-1}\boldsymbol{\varsigma}_{t-i}\right\|_2^2\\ &+ 2\mathrm{Tr}\left(\mathbf{x}_t'(\boldsymbol{\Theta}_T - \hat{\boldsymbol{\Theta}}_T)'(\mathbf{C}\mathbf{A}^{T-1}\mathbf{h}_{t-T+1} + \boldsymbol{\zeta}_t + \mathbf{C}\sum_{i=1}^{T-1}\mathbf{A}^{i-1}\boldsymbol{\varsigma}_{t-i})\right)\Bigg]\\ &= \arg\min_{\hat{\boldsymbol{\Theta}}_T} \lim_{N\to\infty} \frac{1}{2N}\sum_{t=1}^{N} \left\|(\boldsymbol{\Theta}_T - \hat{\boldsymbol{\Theta}}_T)\mathbf{x}_t\right\|_2^2. \qquad (7.10)\end{aligned}$$

Since the process noise is independent of the inputs, we observe that one can learn $\hat{\boldsymbol{\Theta}}_T$ through a regression given in (7.8). Although $\mathbf{C}\sum_{i=1}^{T-1}\mathbf{A}^{i-1}\boldsymbol{\varsigma}_{t-i}$ accumulates in the hidden state of the system, the norm of $\mathbf{C}\mathbf{A}^{i-1}\boldsymbol{\varsigma}_{t-i}$ becomes small for a large i. The reason is that $\rho(\mathbf{A}) < 1$ (similar to (7.7)). □

7.4 Regression Approach and Convergence Analysis

This section is concerned with solving (7.8). Overall, it is difficult to solve since an infinite sum of squared Frobenius norms are to be minimized. In practice, it is impossible to solve, as one cannot wait for an infinite number of input-output pairs. We solve the following problem instead:

$$\hat{\boldsymbol{\Theta}}_T = \arg\min_{\hat{\boldsymbol{\Theta}}_T} \frac{1}{2N}\sum_{t=1}^{N} \left\|\mathbf{y}_t - \hat{\boldsymbol{\Theta}}_T\mathbf{x}_t\right\|_2^2. \qquad (7.11)$$

Based on the finite collected input-output pairs, we estimate $\mathbf{\Theta}_T$. Due to the strong convexity of (7.11) when $N \geq 2T$ (i.e., $\frac{1}{N}\sum_{t=1}^{N}\mathbb{E}(\mathbf{x}_t\mathbf{x}_t') \succ \mathbf{0}$), increasing the number of samples N strictly decreases the Frobenius norm distance between the minimizer of (7.11) and $\mathbf{\Theta}_T$. In the following theorem, we characterize the maximum Frobenius norm distance between $\mathbf{\Theta}_T$ and the minimizer of (7.11) as a function of N, the truncation length T, the covariance of inputs, and the measurement noise level.

Theorem 1. *For any given $N \geq 2T$, the maximum Frobenius norm distance between the first-order stationary solution to* (7.11) *and $\mathbf{\Theta}_T$ is upper-bounded as follows:*

$$\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_{\boldsymbol{\zeta}}\left[\left\|\hat{\mathbf{\Theta}}_T - \mathbf{\Theta}_T\right\|_F^2\right]\right] \leq \frac{n^2\ell\,\|\mathbf{C}\|_F^2\,\rho(\mathbf{A})^{2(T-1)}m^3T^2(\max(\boldsymbol{\sigma}^{\cdot 2}))^2\,\|\mathbf{B}\|_F^2}{(N-T+1)\,(\min(\boldsymbol{\sigma}^{\cdot 2}))^2}$$
$$+\frac{pm^2T^2\max(\boldsymbol{\sigma}_{\zeta}^{\cdot 2})\max(\boldsymbol{\sigma}^{\cdot 2})}{(N-T+1)\,(\min(\boldsymbol{\sigma}^{\cdot 2}))^2} + \frac{n^4\ell^2m^2T^2\rho(\mathbf{A})^{2T}\,\|\mathbf{C}\|_F^2\left(\max(\boldsymbol{\sigma}^{\cdot 2})\right)^2\iota}{(N-T+1)\,(\min(\boldsymbol{\sigma}^{\cdot 2}))^2} = \chi_N^2, \qquad (7.12)$$

where $\iota = \|\mathbf{h}_0\|_2^2 + \frac{m\max(\boldsymbol{\sigma}^{\cdot 2})\|\mathbf{B}\|_F^2}{1-\rho(\mathbf{A})^2}$.

Proof. Consider the following problem:

$$\hat{\mathbf{\Theta}}_T = \arg\min_{\hat{\mathbf{\Theta}}_T}\frac{1}{2(N-T+1)}\sum_{t=T}^{N}\left\|\mathbf{y}_t - \hat{\mathbf{\Theta}}_T\mathbf{x}_t\right\|_2^2.$$

The starting point in the above summation is $t = T$. The reason is that at this point, all the elements of the vector $\mathbf{x}_t$ are filled with random numbers (see (7.3)). Once we bound the distance between $\hat{\mathbf{\Theta}}_T$ for the above problem and $\mathbf{\Theta}_T$, it will an upper-bound for $\hat{\mathbf{\Theta}}_T$ in (7.11). The Hessian matrix for the above problem is $\frac{1}{N-T+1}\sum_{t=T}^{N}\mathbf{x}_t\mathbf{x}_t'$. When $N \geq 2\,T$, the Hessian matrix is full rank [213]. Since the Hessian matrix is positive definite, we can solve the above problem using the first order optimality condition:

$$\nabla_{\hat{\mathbf{\Theta}}_T}\frac{1}{2(N-T+1)}\sum_{t=T}^{N}\left\|\mathbf{y}_t - \hat{\mathbf{\Theta}}_T\mathbf{x}_t\right\|_2^2 = \frac{1}{N-T+1}\sum_{t=T}^{N}\left(\mathbf{y}_t - \hat{\mathbf{\Theta}}_T\mathbf{x}_t\right)\mathbf{x}_t'$$
$$= \frac{1}{N-T+1}\sum_{t=T}^{N}\left(\mathbf{\Theta}_T\mathbf{x}_t + \boldsymbol{\zeta}_t + \mathbf{C}\mathbf{A}^{T-1}\mathbf{h}_{t-T+1} - \hat{\mathbf{\Theta}}_T\mathbf{x}_t\right)\mathbf{x}_t' = \mathbf{0}.$$

Using the above equation, we find:

$$\left(\hat{\mathbf{\Theta}}_T - \mathbf{\Theta}_T\right)\left(\sum_{t=T}^{N} \mathbf{x}_t\mathbf{x}_t'\right) = \sum_{t=T}^{N} \boldsymbol{\zeta}_t\mathbf{x}_t' + \sum_{t=T}^{N} \mathbf{C}\mathbf{A}^{T-1}\mathbf{h}_{t-T+1}\mathbf{x}_t'.$$

Since when $N \geq 2\,T$, the scatter matrix $\frac{1}{2(N-T+1)}\sum_{t=T}^{N} \mathbf{x}_t\mathbf{x}_t'$ is full rank and invertible. We find the difference between the optimal solution and the current point as follows:

$$\hat{\mathbf{\Theta}}_T - \mathbf{\Theta}_T = \left(\sum_{t=T}^{N} \boldsymbol{\zeta}_t\mathbf{x}_t'\right)\left(\sum_{t=T}^{N} \mathbf{x}_t\mathbf{x}_t'\right)^{-1} + \left(\sum_{t=T}^{N} \mathbf{C}\mathbf{A}^{T-1}\mathbf{h}_{t-T+1}\mathbf{x}_t'\right)\left(\sum_{t=T}^{N} \mathbf{x}_t\mathbf{x}_t'\right)^{-1}. \quad (7.13)$$

We bound the expected Frobenius norm distance to the global optimal solution as follows:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{\zeta}}\left[\mathbb{E}_{\mathbf{u}}\left[\left\|\hat{\mathbf{\Theta}}_T - \mathbf{\Theta}_T\right\|_F^2\right]\right] \\
&= \mathbb{E}_{\boldsymbol{\zeta}}\left[\mathbb{E}_{\mathbf{u}}\left[\left\|\left(\sum_{t=T}^{N} \boldsymbol{\zeta}_t\mathbf{x}_t'\right)\left(\sum_{t=T}^{N} \mathbf{x}_t\mathbf{x}_t'\right)^{-1} + \left(\sum_{t=T}^{N} \mathbf{C}\mathbf{A}^{T-1}\mathbf{h}_{t-T+1}\mathbf{x}_t'\right)\left(\sum_{t=T}^{N} \mathbf{x}_t\mathbf{x}_t'\right)^{-1}\right\|_F^2\right]\right] \\
&\leq \mathbb{E}_{\boldsymbol{\zeta}}\left[\mathbb{E}_{\mathbf{u}}\left[\left\|\left(\sum_{t=T}^{N} \boldsymbol{\zeta}_t\mathbf{x}_t'\right)\left(\sum_{t=T}^{N} \mathbf{x}_t\mathbf{x}_t'\right)^{-1}\right\|_F^2\right]\right] \\
&+ \mathbb{E}_{\boldsymbol{\zeta}}\left[\mathbb{E}_{\mathbf{u}}\left[\left\|\left(\sum_{t=T}^{N} \mathbf{C}\mathbf{A}^{T-1}\mathbf{h}_{t-T+1}\mathbf{x}_t'\right)\left(\sum_{t=T}^{N} \mathbf{x}_t\mathbf{x}_t'\right)^{-1}\right\|_F^2\right]\right] \\
&\overset{(a)}{\leq} \mathbb{E}_{\boldsymbol{\zeta}}\left[\mathbb{E}_{\mathbf{u}}\left[\left\|\sum_{t=T}^{N} \boldsymbol{\zeta}_t\mathbf{x}_t'\right\|_F^2\left\|\left(\sum_{t=T}^{N} \mathbf{x}_t\mathbf{x}_t'\right)^{-1}\right\|_F^2\right]\right] \\
&+ \mathbb{E}_{\mathbf{u}}\left[\left\|\sum_{t=T}^{N} \mathbf{C}\mathbf{A}^{T-1}\mathbf{h}_{t-T+1}\mathbf{x}_t'\right\|_F^2\left\|\left(\sum_{t=T}^{N} \mathbf{x}_t\mathbf{x}_t'\right)^{-1}\right\|_F^2\right], \\
&\leq \mathbb{E}_{\boldsymbol{\zeta}}\left[\mathbb{E}_{\mathbf{u}}\left[\left\|\sum_{t=T}^{N} \boldsymbol{\zeta}_t\mathbf{x}_t'\right\|_F^2\right]\right]\mathbb{E}_{\mathbf{u}}\left[\left\|\left(\sum_{t=T}^{N} \mathbf{x}_t\mathbf{x}_t'\right)^{-1}\right\|_F^2\right] \\
&+ \mathbb{E}_{\mathbf{u}}\left[\left\|\sum_{t=T}^{N} \mathbf{C}\mathbf{A}^{T-1}\mathbf{h}_{t-T+1}\mathbf{x}_t'\right\|_F^2\right]\mathbb{E}_{\mathbf{u}}\left[\left\|\left(\sum_{t=T}^{N} \mathbf{x}_t\mathbf{x}_t'\right)^{-1}\right\|_F^2\right], \qquad (7.14)
\end{aligned}$$

where (a) follows because of the Cauchy-Schwarz inequality. Before simplifying the above inequality, let us bound the norm of $\left(\frac{1}{N-T+1}\sum_{t=T}^{N}\mathbf{x}_t\mathbf{x}_t'\right)^{-1}$ as follows:

$$\begin{aligned}&\mathbb{E}_{\mathbf{u}}\left[\left\|\left(\frac{1}{N-T+1}\sum_{t=T}^{N}\mathbf{x}_t\mathbf{x}_t'\right)^{-1}\right\|_F^2\right]\\&\leq mT\mathbb{E}_{\mathbf{u}}\left[\left\|\left(\frac{1}{N-T+1}\sum_{t=T}^{N}\mathbf{x}_t\mathbf{x}_t'\right)^{-1}\right\|_2^2\right]=\frac{mT}{\left(\min(\boldsymbol{\sigma}^{\cdot 2})\right)^2}.\end{aligned}$$

We know that $\frac{1}{(N-T+1)}\left(\sum_{t=T}^{N}\mathbf{x}_t\mathbf{x}_t'\right)^{-1}$ is the unbiased estimator of the covariance matrix [213]. Therefore, we have

$$\mathbb{E}_{\mathbf{u}}\left[\left\|\left(\sum_{t=T}^{N}\mathbf{x}_t\mathbf{x}_t'\right)^{-1}\right\|_F^2\right]\leq mT\mathbb{E}_{\mathbf{u}}\left[\left\|\left(\sum_{t=T}^{N}\mathbf{x}_t\mathbf{x}_t'\right)^{-1}\right\|_2^2\right]=\frac{mT}{(N-T+1)^2\left(\min(\boldsymbol{\sigma}^{\cdot 2})\right)^2}.$$

We bound $\mathbb{E}_{\boldsymbol{\zeta}}\left[\mathbb{E}_{\mathbf{u}}\left[\left\|\sum_{t=T}^{N}\boldsymbol{\zeta}_t\mathbf{x}_t'\right\|_F^2\right]\right]$ as follows:

$$\mathbb{E}_{\boldsymbol{\zeta}}\left[\mathbb{E}_{\mathbf{u}}\left[\left\|\sum_{t=T}^{N}\boldsymbol{\zeta}_t\mathbf{x}_t'\right\|_F^2\right]\right]\leq\sum_{t=T}^{N}\mathbb{E}_{\boldsymbol{\zeta}}\left[\mathbb{E}_{\mathbf{u}}\left[\|\boldsymbol{\zeta}_t\|_2^2\left\|\mathbf{x}_t'\right\|_2^2\right]\right]\leq(N-T+1)\,pmT\max(\boldsymbol{\sigma}_{\zeta}^{\cdot 2})\max(\boldsymbol{\sigma}^{\cdot 2}).$$

We bound $\mathbb{E}_{\mathbf{u}}\left[\left\|\sum_{t=T}^{N}\mathbf{C}\mathbf{A}^{T-1}\mathbf{h}_{t-T+1}\mathbf{x}_t'\right\|_F^2\right]$ as follows:

$$\begin{aligned}&\mathbb{E}_{\mathbf{u}}\left[\left\|\sum_{t=T}^{N}\mathbf{C}\mathbf{A}^{T-1}\mathbf{h}_{t-T+1}\mathbf{x}_t'\right\|_F^2\right]\leq\sum_{t=T}^{N}\|\mathbf{C}\|_F^2\left\|\mathbf{A}^{T-1}\right\|_F^2\mathbb{E}_{\mathbf{u}}\left[\left\|\mathbf{x}_t'\right\|_2^2\right]\mathbb{E}_{\mathbf{u}}\left[\|\mathbf{h}_{t-T+1}\|_2^2\right]\\&\leq\sum_{t=T}^{N}n^2\ell\|\mathbf{C}\|_F^2\,\rho(\mathbf{A})^{2(T-1)}\mathbb{E}_{\mathbf{u}}\left[\left\|\mathbf{x}_t'\right\|_2^2\right]\mathbb{E}_{\mathbf{u}}\left[\|\mathbf{h}_{t-T+1}\|_2^2\right]\\&\overset{(a)}{\leq}n^2\ell mT\|\mathbf{C}\|_F^2\,\rho(\mathbf{A})^{2(T-1)}\max(\boldsymbol{\sigma}^{\cdot 2})\sum_{t=T}^{N}\left(m\max(\boldsymbol{\sigma}^{\cdot 2})\|\mathbf{B}\|_F^2\right)\\&+n^2\ell mT\|\mathbf{C}\|_F^2\,\rho(\mathbf{A})^{2(T-1)}\max(\boldsymbol{\sigma}^{\cdot 2})\sum_{t=T}^{N}\left(n^2\ell\rho(\mathbf{A})^2\|\mathbf{h}_0\|_2^2+\frac{n^2\ell m\max(\boldsymbol{\sigma}^{\cdot 2})\rho(\mathbf{A})^2\|\mathbf{B}\|_F^2}{1-\rho(\mathbf{A})^2}\right)\end{aligned}$$

$$
\begin{aligned}
&\leq n^2\ell \left\|\mathbf{C}\right\|_F^2 \rho(\mathbf{A})^{2(T-1)} m^2 T (\max(\boldsymbol{\sigma}^{\cdot 2}))^2 (N-T+1) \left\|\mathbf{B}\right\|_F^2 \\
&+ n^4\ell^2 m T \left\|\mathbf{C}\right\|_F^2 \rho(\mathbf{A})^{2T} \left(\max(\boldsymbol{\sigma}^{\cdot 2})\right)^2 (N-T+1) \times \underbrace{\left[\left\|\mathbf{h}_0\right\|_2^2 + \frac{m \max(\boldsymbol{\sigma}^{\cdot 2}) \left\|\mathbf{B}\right\|_F^2}{1-\rho(\mathbf{A})^2}\right]}_{\iota},
\end{aligned}
\tag{7.15}
$$

where in (a) we use (7.7) to bound $\left\|\mathbf{h}_{t-T+1}\right\|_2^2$. We use the above inequalities to simplify (7.14) as follows:

$$
\begin{aligned}
&\mathbb{E}_{\boldsymbol{\zeta}}\left[\mathbb{E}_{\mathbf{u}}\left[\left\|\hat{\boldsymbol{\Theta}}_T - \boldsymbol{\Theta}_T\right\|_F^2\right]\right] \\
&\leq \mathbb{E}_{\boldsymbol{\zeta}}\left[\mathbb{E}_{\mathbf{u}}\left[\left\|\sum_{t=T}^{N} \boldsymbol{\zeta}_t \mathbf{x}_t'\right\|_F^2\right]\right] \mathbb{E}_{\boldsymbol{\zeta}}\left[\mathbb{E}_{\mathbf{u}}\left[\left\|\left(\sum_{t=T}^{N} \mathbf{x}_t\mathbf{x}_t'\right)^{-1}\right\|_F^2\right]\right] \\
&+ \mathbb{E}_{\mathbf{u}}\left[\left\|\sum_{t=T}^{N} \mathbf{C}\mathbf{A}^{T-1}\mathbf{h}_{t-T+1}\mathbf{x}_t'\right\|_F^2\right] \mathbb{E}_{\mathbf{u}}\left[\left\|\left(\sum_{t=T}^{N} \mathbf{x}_t\mathbf{x}_t'\right)^{-1}\right\|_F^2\right] \\
&\leq \frac{p m^2 T^2 \max(\boldsymbol{\sigma}_{\zeta}^{\cdot 2}) \max(\boldsymbol{\sigma}^{\cdot 2})}{(N-T+1)\left(\min(\boldsymbol{\sigma}^{\cdot 2})\right)^2} + \frac{n^2\ell \left\|\mathbf{C}\right\|_F^2 \rho(\mathbf{A})^{2(T-1)} m^3 T^2 (\max(\boldsymbol{\sigma}^{\cdot 2}))^2 \left\|\mathbf{B}\right\|_F^2}{(N-T+1)\left(\min(\boldsymbol{\sigma}^{\cdot 2})\right)^2} \\
&+ \frac{n^4\ell^2 m^2 T^2 \rho(\mathbf{A})^{2T} \left\|\mathbf{C}\right\|_F^2 \left(\max(\boldsymbol{\sigma}^{\cdot 2})\right)^2 \iota}{(N-T+1)\left(\min(\boldsymbol{\sigma}^{\cdot 2})\right)^2} = \chi_N^2.
\end{aligned}
$$

□

Based on the above theorem, increasing the trajectory length N drives the solution of (7.11) closer to $\boldsymbol{\Theta}_T$. Although the Frobenius norm distance between the solution of (7.11) and the ground truth strictly decreases with N, solving (7.11) globally by the pseudo-inverse method (e.g., [2, 201, 204]), second-order methods (e.g., log barrier), and gradient descent methods are costly and challenging. The reason is that when N and T are large numbers, the calculation and inversion of $\frac{1}{N}\sum_{t=1}^{N} \mathbf{x}_t\mathbf{x}_t'$, which is $mT \times mT$ dimensional becomes expensive. Therefore, a computationally faster and more cost-efficient approach is desired.

Algorithm 22: Offline SGD algorithm to learn $\mathbf{\Theta}_T$

```
**Initialization**: Assign small value to $\hat{\mathbf{\Theta}}_{0,T}$
**Input**: $\{\mathbf{x}_t, \mathbf{y}_t\}_{t=1}^N$, learning rate $\eta$
**Output**: Estimation of $\mathbf{\Theta}_T$
**for** $\tau$ *from* 1 *to END* **do**
    Uniformly at random choose $t \in \{T, T+1, \dots, N\}$
    $\hat{\mathbf{\Theta}}_{\tau,T} = \hat{\mathbf{\Theta}}_{\tau-1,T} - \eta(\hat{\mathbf{\Theta}}_{\tau-1,T}\mathbf{x}_t - \mathbf{y}_t)\mathbf{x}_t'$
**end**
```

7.4.1 Offline SGD

To alleviate the computational cost of solving (7.11), we propose a low iteration cost SGD algorithm, which works based on a fixed batch of input-output pairs. Since this algorithm uses a fixed batch size, we name it offline SGD. The τ^{th} iteration of the offline SGD is described in the following step:

$$\hat{\mathbf{\Theta}}_{\tau,T} = \hat{\mathbf{\Theta}}_{\tau-1,T} - \eta(\hat{\mathbf{\Theta}}_{\tau-1,T}\mathbf{x}_t - \mathbf{y}_t)\mathbf{x}_t', \tag{7.16}$$

where η is a constant learning rate and $t \in \{T, T+1, \dots, N\}$ is chosen with probability $\frac{1}{N-T+1}$. When we use the offline SGD instead of the traditional gradient descent to solve (7.11) , the complexity of solving the problem in each iteration reduces from $\mathcal{O}(NpmT)$ to $\mathcal{O}(pmT)$, which is a significant improvement if N is large. The offline SGD is summarized in Algorithm 22. In the following theorem, we bound the maximum expected distance between the offline SGD iterate $\hat{\mathbf{\Theta}}_{\tau,T}$ and $\mathbf{\Theta}_T$ as a function of the number of iterations, T, the covariance of inputs, N, and noise levels.

Lemma 2. *For an arbitrary μ-strongly convex function $f(\mathbf{x})$ with an L-Lipschitz continuous gradient, we have*

$$\langle \nabla f(\mathbf{x}) - \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle \geq \mu \left\| \mathbf{x} - \mathbf{y} \right\|_2^2, \tag{7.17}$$

$$\langle \nabla f(\mathbf{x}) - \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle \geq \frac{1}{L} \left\| \nabla f(\mathbf{x}) - \nabla f(\mathbf{y}) \right\|_2^2, \tag{7.18}$$

where (7.17) *follows from [214, eq. 2.1.11] and* (7.18) *follows from [214, eq. 2.1.8]. From* *the Bergstrom's inequality, we have*

$$2\left\|\mathbf{x}\right\|_2^2 + 2\left\|\mathbf{y}\right\|_2^2 \geq \left\|\mathbf{x}+\mathbf{y}\right\|_2^2. \tag{7.19}$$

Theorem 2. *Let $\boldsymbol{\phi}_\tau$ denote the difference between $\hat{\boldsymbol{\Theta}}_{\tau,T}$ (in the τ^{th} iteration) and ground truth $\boldsymbol{\Theta}_T$ as $\boldsymbol{\phi}_\tau = \hat{\boldsymbol{\Theta}}_{\tau,T} - \boldsymbol{\Theta}_T$, and $\boldsymbol{w}_0 = \hat{\boldsymbol{\Theta}}_{0,T} - \hat{\boldsymbol{\Theta}}_T$. Consider that the offline SGD minimizes* (7.11) *with a batch of size $N \geq 2T$, where each iteration is implemented based on* (7.16) *with $\eta \leq \frac{1}{m\,T\max(\boldsymbol{\sigma}^{\cdot 2})}$. Then, $\mathbb{E}_{\mathbf{u}}[\mathbb{E}_{\boldsymbol{\zeta}}[\|\boldsymbol{\phi}_\tau\|_F^2]]$ in the τ^{th} iteration of the offline SGD can be upper-bounded as follows:*

$$\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_{\boldsymbol{\zeta}}[\|\boldsymbol{\phi}_\tau\|_F^2]\right] < \|\boldsymbol{\omega}_0\|_F^2\left(1 - 2\eta m\,T\min(\boldsymbol{\sigma}^{\cdot 2}) + 2\eta^2 m^2\,T^2\min(\boldsymbol{\sigma}^{\cdot 2})\max(\boldsymbol{\sigma}^{\cdot 2})\right)^{\tau} + \Delta_N + \chi_N^2, \tag{7.20}$$

where Δ_N is given in (7.27)*, which depends on different problem parameters such as χ_N, η, and T.*

Proof. Let us assume that the initial state of the system is denoted by $\mathbf{h}_0$. Each iteration of the offline SGD is as follows:

$$\hat{\boldsymbol{\Theta}}_{\tau,T} = \hat{\boldsymbol{\Theta}}_{\tau-1,T} - \eta(\hat{\boldsymbol{\Theta}}_{\tau-1,T}\mathbf{x}_t - \mathbf{y}_t)\mathbf{x}_t',$$

where t is randomly chosen from $\{T, T+1, \ldots, N\}$ with uniform probability. We let the first-order solution obtained from (7.11) be denoted by $\hat{\boldsymbol{\Theta}}_T$ and the ground truth solution is represented by $\boldsymbol{\Theta}_T$. The difference between $\hat{\boldsymbol{\Theta}}_T$ and the ground truth solution is denoted by $\boldsymbol{\nu}$ and defined as $\boldsymbol{\nu} = \hat{\boldsymbol{\Theta}}_T - \boldsymbol{\Theta}_T$. In Theorem 1, the Frobenius norm of the difference is bounded as $\mathbb{E}_{\mathbf{u}}[\mathbb{E}_{\boldsymbol{\zeta}}[\|\boldsymbol{\nu}\|]]_F^2 \leq \chi_N^2$, where N is the batch size. Let $\boldsymbol{\omega}_\tau$ denote the difference between $\hat{\boldsymbol{\Theta}}_{\tau,T}$ (in τ^{th} iteration) and $\hat{\boldsymbol{\Theta}}_T$ as $\boldsymbol{\omega}_\tau = \hat{\boldsymbol{\Theta}}_{\tau,T} - \hat{\boldsymbol{\Theta}}_T$. Based on the definition of $\boldsymbol{\omega}_\tau$, the update rule for $\boldsymbol{\omega}_\tau$ is $\boldsymbol{\omega}_{\tau+1} = \boldsymbol{\omega}_\tau - \eta(\hat{\boldsymbol{\Theta}}_{\tau,T}\mathbf{x}_t - \mathbf{y}_t)\mathbf{x}_t'$. We have

$$\begin{aligned}
&\hat{\boldsymbol{\Theta}}_{\tau,T}\mathbf{x}_t - \mathbf{y}_t = \hat{\boldsymbol{\Theta}}_{\tau,T}\mathbf{x}_t - \boldsymbol{\Theta}_T\mathbf{x}_t - \boldsymbol{\zeta}_t - \mathbf{C}\mathbf{A}^{T-1}\mathbf{h}_{t-T+1}\\
&= \hat{\boldsymbol{\Theta}}_{\tau,T}\mathbf{x}_t - \hat{\boldsymbol{\Theta}}_T\mathbf{x}_t + \boldsymbol{\nu}\mathbf{x}_t - \boldsymbol{\zeta}_t - \mathbf{C}\mathbf{A}^{T-1}\mathbf{h}_{t-T+1}\\
&= (\hat{\boldsymbol{\Theta}}_{\tau,T} - \hat{\boldsymbol{\Theta}}_T)\mathbf{x}_t + \boldsymbol{\nu}\mathbf{x}_t - \boldsymbol{\zeta}_t - \mathbf{C}\mathbf{A}^{T-1}\mathbf{h}_{t-T+1} = \boldsymbol{\omega}_\tau\mathbf{x}_t + \boldsymbol{\nu}\mathbf{x}_t - \boldsymbol{\zeta}_t - \mathbf{C}\mathbf{A}^{T-1}\mathbf{h}_{t-T+1}.
\end{aligned}$$

We bound the optimality gap as follows:

$$\begin{aligned}\|\boldsymbol{\omega}_{\tau+1}\|_F^2 &= \left\|\boldsymbol{\omega}_\tau - \eta(\hat{\boldsymbol{\Theta}}_{\tau,T}\mathbf{x}_t - \mathbf{y}_t)\mathbf{x}_t'\right\|_F^2 = \|\boldsymbol{\omega}_\tau\|_F^2 - 2\eta\mathbf{x}_t'\boldsymbol{\omega}_\tau'(\hat{\boldsymbol{\Theta}}_{\tau,T}\mathbf{x}_t - \mathbf{y}_t)\\ &+ \eta^2\left\|(\hat{\boldsymbol{\Theta}}_{\tau,T}\mathbf{x}_t - \mathbf{y}_t)\mathbf{x}_t'\right\|_F^2.\end{aligned}$$

We take expectations with respect to t and $\mathbf{u}$ and obtain

$$\mathbb{E}_{\mathbf{u}}[\mathbb{E}_t[\|\boldsymbol{\omega}_{\tau+1}\|_F^2]] = \|\boldsymbol{\omega}_\tau\|_F^2 - 2\eta\mathbb{E}_{\mathbf{u}}[\mathbb{E}_t[\mathbf{x}_t'\boldsymbol{\omega}_\tau'(\hat{\boldsymbol{\Theta}}_{\tau,T}\mathbf{x}_t - \mathbf{y}_t)]] + \eta^2\mathbb{E}_{\mathbf{u}}[\mathbb{E}_t[\left\|(\hat{\boldsymbol{\Theta}}_{\tau,T}\mathbf{x}_t - \mathbf{y}_t)\mathbf{x}_t'\right\|_F^2]]. \tag{7.21}$$

To simplify (7.21), we obtain a lower-bound for $\mathbb{E}_{\mathbf{u}}[\mathbb{E}_t[\mathbf{x}_t'\boldsymbol{\omega}_\tau'(\hat{\boldsymbol{\Theta}}_{\tau,T}\mathbf{x}_t - \mathbf{y}_t)]]$. Using (7.17), we have:

$$\begin{aligned}&\mathrm{Tr}\left[\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_t[(\hat{\boldsymbol{\Theta}}_{\tau,T}\mathbf{x}_t - \mathbf{y}_t)\mathbf{x}_t' - (\hat{\boldsymbol{\Theta}}_T\mathbf{x}_t - \mathbf{y}_t)\mathbf{x}_t']\right]\boldsymbol{\omega}_\tau'\right] \geq m\,T\min(\boldsymbol{\sigma}^{\cdot 2})\left\|\hat{\boldsymbol{\Theta}}_{\tau,T} - \hat{\boldsymbol{\Theta}}_T\right\|_F^2,\\ \Rightarrow&\mathrm{Tr}\left[\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_t[(\hat{\boldsymbol{\Theta}}_{\tau,T}\mathbf{x}_t - \mathbf{y}_t)\mathbf{x}_t']\boldsymbol{\omega}_\tau'\right]\right]\\ &- \mathrm{Tr}\left[\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_t[(\hat{\boldsymbol{\Theta}}_T\mathbf{x}_t - (\hat{\boldsymbol{\Theta}}_T - \boldsymbol{\nu})\mathbf{x}_t - \boldsymbol{\zeta}_t - \mathbf{C}\mathbf{A}^{T-1}\mathbf{h}_{t-T+1})\mathbf{x}_t']\boldsymbol{\omega}_\tau'\right]\right]\\ &\geq m\,T\min(\boldsymbol{\sigma}^{\cdot 2})\left\|\hat{\boldsymbol{\Theta}}_{\tau,T} - \hat{\boldsymbol{\Theta}}_T\right\|_F^2,\\ \Rightarrow&\mathrm{Tr}\left[\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_t[(\hat{\boldsymbol{\Theta}}_{\tau,T}\mathbf{x}_t - \mathbf{y}_t)\mathbf{x}_t']\boldsymbol{\omega}_\tau'\right]\right] - \mathrm{Tr}\left[\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_t[\boldsymbol{\nu}\mathbf{x}_t\mathbf{x}_t']\boldsymbol{\omega}_\tau'\right]\right]\\ &- \frac{1}{N-T+1}\sum_{t=2T}^{N}\mathrm{Tr}[\mathbb{E}_{\mathbf{u}}[\mathbf{C}\mathbf{A}^{T-1}\mathbf{h}_{t-T+1}\mathbf{x}_t'\boldsymbol{\omega}_\tau']] \geq m\,T\min(\boldsymbol{\sigma}^{\cdot 2})\left\|\hat{\boldsymbol{\Theta}}_{\tau,T} - \hat{\boldsymbol{\Theta}}_T\right\|_F^2,\\ \Rightarrow&\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_t[\mathbf{x}_t'\boldsymbol{\omega}_\tau'(\hat{\boldsymbol{\Theta}}_{\tau,T}\mathbf{x}_t - \mathbf{y}_t)]\right] \geq \mathrm{Tr}\left[\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_t[\boldsymbol{\nu}\mathbf{x}_t\mathbf{x}_t']\boldsymbol{\omega}_\tau'\right]\right] + m\,T\min(\boldsymbol{\sigma}^{\cdot 2})\left\|\hat{\boldsymbol{\Theta}}_{\tau,T} - \hat{\boldsymbol{\Theta}}_T\right\|_F^2,\end{aligned} \tag{7.22}$$

In the above chain of inequalities, we have $\frac{1}{N-T+1}\sum_{t=2T}^{N}\mathrm{Tr}[\mathbb{E}_{\mathbf{u}}[\mathbf{C}\mathbf{A}^{T-1}\mathbf{h}_{t-T+1}\mathbf{x}_t'\boldsymbol{\omega}_\tau']] = \mathbf{0}$ similar to (7.9). Before we bound $\mathbb{E}_{\mathbf{u}}[\mathbb{E}_t[\left\|(\hat{\boldsymbol{\Theta}}_{\tau,T}\mathbf{x}_t - \mathbf{y}_t)\mathbf{x}_t'\right\|_F^2]]$, first, we demonstrate that $\mathbb{E}_{\mathbf{u}}[\left\|\hat{\boldsymbol{\Theta}}_{\tau,T}\mathbf{x}_t - \mathbf{y}_t\right\|_2^2]$ has a Lipschitz continuous gradient as follows:

$$\mathbb{E}_{\mathbf{u}}\left[\left\|(\hat{\boldsymbol{\Theta}}_{\tau,T}\mathbf{x}_t - \mathbf{y}_t)\mathbf{x}_t' - (\hat{\boldsymbol{\Theta}}_T\mathbf{x}_t - \mathbf{y}_t)\mathbf{x}_t'\right\|_F^2\right] \leq \mathbb{E}_{\mathbf{u}}\left[\left\|(\hat{\boldsymbol{\Theta}}_{\tau,T}\mathbf{x}_t - \hat{\boldsymbol{\Theta}}_T\mathbf{x}_t)\mathbf{x}_t'\right\|_F^2\right]$$

$$\le \mathbb{E}_{\mathbf{u}}\left[\left\|\hat{\boldsymbol{\Theta}}_{\tau,T}-\hat{\boldsymbol{\Theta}}_T\right\|_F^2\right]\mathbb{E}_{\mathbf{u}}\left[\left\|\mathbf{x}_t\mathbf{x}_t'\right\|_F^2\right] \le m^2\,T^2(\max(\boldsymbol{\sigma}^{\cdot 2}))^2\mathbb{E}_{\mathbf{u}}\left[\left\|\hat{\boldsymbol{\Theta}}_{\tau,T}-\hat{\boldsymbol{\Theta}}_T\right\|_F^2\right], \quad (7.23)$$

where $m\,T\max(\boldsymbol{\sigma}^{\cdot 2})$ is the Lipschitz constant. We have

$$\begin{aligned}
&\mathbb{E}_{\mathbf{u}}[\mathbb{E}_t[\left\|(\hat{\boldsymbol{\Theta}}_{\tau,T}\mathbf{x}_t-\mathbf{y}_t)\mathbf{x}_t'\right\|_F^2]] \le \mathbb{E}_{\mathbf{u}}[\mathbb{E}_t[\left\|(\hat{\boldsymbol{\Theta}}_{\tau,T}\mathbf{x}_t-\mathbf{y}_t)\mathbf{x}_t'-(\hat{\boldsymbol{\Theta}}_T\mathbf{x}_t-\mathbf{y}_t)\mathbf{x}_t'+(\hat{\boldsymbol{\Theta}}_T\mathbf{x}_t-\mathbf{y}_t)\mathbf{x}_t'\right\|_F^2]]\\
&\overset{(a)}{\le} 2\mathbb{E}_{\mathbf{u}}[\mathbb{E}_t[\left\|(\hat{\boldsymbol{\Theta}}_T\mathbf{x}_t-\mathbf{y}_t)\mathbf{x}_t'-(\hat{\boldsymbol{\Theta}}_{\tau,T}\mathbf{x}_t-\mathbf{y}_t)\mathbf{x}_t'\right\|_F^2]]+2\mathbb{E}_{\mathbf{u}}[\mathbb{E}_t[\left\|(\hat{\boldsymbol{\Theta}}_T\mathbf{x}_t-\mathbf{y}_t)\mathbf{x}_t'\right\|_F^2]]\\
&\overset{(b)}{\le} 2m\,T\max(\boldsymbol{\sigma}^{\cdot 2})\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_t[\mathbf{x}_t'\boldsymbol{\omega}_\tau'(\hat{\boldsymbol{\Theta}}_{\tau,T}\mathbf{x}_t-\mathbf{y}_t-\hat{\boldsymbol{\Theta}}_T\mathbf{x}_t+\mathbf{y}_t)]\right]\\
&+2\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_t[\left\|(\boldsymbol{\nu}\mathbf{x}_t-\mathbf{C}\mathbf{A}^{T-1}\mathbf{h}_{t-T+1}-\boldsymbol{\zeta}_t)\mathbf{x}_t'\right\|_F^2]\right]\\
&\le 2m\,T\max(\boldsymbol{\sigma}^{\cdot 2})\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_t[\mathbf{x}_t'\boldsymbol{\omega}_\tau'(\hat{\boldsymbol{\Theta}}_{\tau,T}\mathbf{x}_t-\mathbf{y}_t)]\right]-2m\,T\max(\boldsymbol{\sigma}^{\cdot 2})\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_t[\mathbf{x}_t'\boldsymbol{\omega}_\tau'(\hat{\boldsymbol{\Theta}}_T\mathbf{x}_t-\mathbf{y}_t)]\right]\\
&+2\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_t[\left\|(\boldsymbol{\nu}\mathbf{x}_t-\mathbf{C}\mathbf{A}^{T-1}\mathbf{h}_{t-T+1}-\boldsymbol{\zeta}_t)\mathbf{x}_t'\right\|_F^2]\right]\\
&\le 2m\,T\max(\boldsymbol{\sigma}^{\cdot 2})\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_t[\mathbf{x}_t'\boldsymbol{\omega}_\tau'(\hat{\boldsymbol{\Theta}}_{\tau,T}\mathbf{x}_t-\mathbf{y}_t)]\right]\\
&-2m\,T\max(\boldsymbol{\sigma}^{\cdot 2})\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_t[\mathbf{x}_t'\boldsymbol{\omega}_\tau'(\boldsymbol{\nu}\mathbf{x}_t-\mathbf{C}\mathbf{A}^{T-1}\mathbf{h}_{t-T+1}-\boldsymbol{\zeta}_t)]\right]\\
&+2\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_t[\left\|(\boldsymbol{\nu}\mathbf{x}_t-\mathbf{C}\mathbf{A}^{T-1}\mathbf{h}_{t-T+1}-\boldsymbol{\zeta}_t)\mathbf{x}_t'\right\|_F^2]\right]\\
&\le 2m\,T\max(\boldsymbol{\sigma}^{\cdot 2})\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_t[\mathbf{x}_t'\boldsymbol{\omega}_\tau'(\hat{\boldsymbol{\Theta}}_{\tau,T}\mathbf{x}_t-\mathbf{y}_t)]\right]\\
&-2m\,T\max(\boldsymbol{\sigma}^{\cdot 2})\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_t[\mathbf{x}_t'\boldsymbol{\omega}_\tau'\boldsymbol{\nu}\mathbf{x}_t]\right]\\
&+2\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_t[\left\|(\boldsymbol{\nu}\mathbf{x}_t-\mathbf{C}\mathbf{A}^{T-1}\mathbf{h}_{t-T+1}-\boldsymbol{\zeta}_t)\mathbf{x}_t'\right\|_F^2]\right],
\end{aligned}$$

where (a) follows from (7.19) and (b) follows from (7.18). We simplify (7.21) using (7.22) and the above inequality as follows:

$$\begin{aligned}
&\mathbb{E}_{\mathbf{u}}[\mathbb{E}_t[\|\boldsymbol{\omega}_{\tau+1}\|_F^2]] \le \|\boldsymbol{\omega}_\tau\|_F^2-2\eta\mathbb{E}_{\mathbf{u}}[\mathbb{E}_t[\mathbf{x}_t'\boldsymbol{\omega}_\tau'(\hat{\boldsymbol{\Theta}}_{\tau,T}\mathbf{x}_t-\mathbf{y}_t)]]\\
&+2\eta^2 m\,T\max(\boldsymbol{\sigma}^{\cdot 2})\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_t[\mathbf{x}_t'\boldsymbol{\omega}_\tau'(\hat{\boldsymbol{\Theta}}_{\tau,T}\mathbf{x}_t-\mathbf{y}_t)]\right]\\
&+2\eta^2\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_t[\left\|(\boldsymbol{\nu}\mathbf{x}_t-\mathbf{C}\mathbf{A}^{T-1}\mathbf{h}_{t-T+1}-\boldsymbol{\zeta}_t)\mathbf{x}_t'\right\|_F^2]\right]-2\eta^2 m\,T\max(\boldsymbol{\sigma}^{\cdot 2})\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_t[\mathbf{x}_t'\boldsymbol{\omega}_\tau'\boldsymbol{\nu}\mathbf{x}_t]\right]\\
&\le \|\boldsymbol{\omega}_\tau\|_F^2+(-2\eta+2\eta^2 m\,T\max(\boldsymbol{\sigma}^{\cdot 2}))\mathbb{E}_{\mathbf{u}}[\mathbb{E}_t[\mathbf{x}_t'\boldsymbol{\omega}_\tau'(\hat{\boldsymbol{\Theta}}_{\tau,T}\mathbf{x}_t-\mathbf{y}_t)]]\\
&+2\eta^2\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_t[\left\|(\boldsymbol{\nu}\mathbf{x}_t-\mathbf{C}\mathbf{A}^{T-1}\mathbf{h}_{t-T+1}-\boldsymbol{\zeta}_t)\mathbf{x}_t'\right\|_F^2]\right]-2\eta^2 m\,T\max(\boldsymbol{\sigma}^{\cdot 2})\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_t[\mathbf{x}_t'\boldsymbol{\omega}_\tau'\boldsymbol{\nu}\mathbf{x}_t]\right]\\
&\overset{(a)}{\le} \|\boldsymbol{\omega}_\tau\|_F^2\,(1-2\eta m\,T\min(\boldsymbol{\sigma}^{\cdot 2})+2\eta^2 m^2\,T^2\min(\boldsymbol{\sigma}^{\cdot 2})\max(\boldsymbol{\sigma}^{\cdot 2}))
\end{aligned}$$

$$
\begin{aligned}
&+ (-2\eta + 2\eta^2 m\, T \max(\boldsymbol{\sigma}^{\cdot 2}))\mathrm{Tr}\left[\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_t[\boldsymbol{\nu}\mathbf{x}_t\mathbf{x}_t']\boldsymbol{\omega}_\tau'\right]\right] \\
&- 2\eta^2 m\, T \max(\boldsymbol{\sigma}^{\cdot 2})\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_t[\mathbf{x}_t'\boldsymbol{\omega}_\tau'\boldsymbol{\nu}\mathbf{x}_t]\right] \\
&+ 2\eta^2\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_t[\left\|(\boldsymbol{\nu}\mathbf{x}_t - \mathbf{C}\mathbf{A}^{T-1}\mathbf{h}_{t-T+1} - \boldsymbol{\zeta}_t)\mathbf{x}_t'\right\|_F^2]\right] \\
&\overset{(b)}{\le} \|\boldsymbol{\omega}_\tau\|_F^2 (1 - 2\eta m\, T \min(\boldsymbol{\sigma}^{\cdot 2}) + 2\eta^2 m^2\, T^2 \min(\boldsymbol{\sigma}^{\cdot 2}) \max(\boldsymbol{\sigma}^{\cdot 2})) \\
&+ 2\eta m T \max(\boldsymbol{\sigma}^{\cdot 2})\, \|\boldsymbol{\nu}\|_F\, \|\boldsymbol{\omega}_0\|_F + 2\eta^2\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_t[\left\|(\boldsymbol{\nu}\mathbf{x}_t - \mathbf{C}\mathbf{A}^{T-1}\mathbf{h}_{t-T+1} - \boldsymbol{\zeta}_t)\mathbf{x}_t'\right\|_F^2]\right] \\
&+ 2\eta^2 m^2\, T^2(\max(\boldsymbol{\sigma}^{\cdot 2}))^2|\mathrm{Tr}[\boldsymbol{\nu}\boldsymbol{\omega}_\tau']| \\
&\le \|\boldsymbol{\omega}_\tau\|_F^2 (1 - 2\eta m\, T \min(\boldsymbol{\sigma}^{\cdot 2}) + 2\eta^2 m^2\, T^2 \min(\boldsymbol{\sigma}^{\cdot 2}) \max(\boldsymbol{\sigma}^{\cdot 2})) \\
&+ (\eta m T \max(\boldsymbol{\sigma}^{\cdot 2}) + \eta^2 m^2\, T^2(\max(\boldsymbol{\sigma}^{\cdot 2}))^2)(\|\boldsymbol{\nu}\|_F^2 + \|\boldsymbol{\omega}_0\|_F^2) \\
&+ 2\eta^2\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_t[\left\|(\boldsymbol{\nu}\mathbf{x}_t - \mathbf{C}\mathbf{A}^{T-1}\mathbf{h}_{t-T+1} - \boldsymbol{\zeta}_t)\mathbf{x}_t'\right\|_F^2]\right], \qquad (7.24)
\end{aligned}
$$

where (a) follows from (7.22) and the fact that $\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_t[\mathbf{x}_t'\boldsymbol{\omega}_\tau'\boldsymbol{\nu}\mathbf{x}_t]\right] \le mT\max(\boldsymbol{\sigma}^{\cdot 2})\mathrm{Tr}[\boldsymbol{\nu}\boldsymbol{\omega}_\tau]$. Furthermore, (b) follows from Von Neumann's trace inequality and also the assumption that $-2\eta + 2\eta^2 m\, T \max(\boldsymbol{\sigma}^{\cdot 2}) < 0$. To simplify (7.24), we consider the following bound:

$$
\begin{aligned}
&\mathbb{E}_{\boldsymbol{\zeta}}\left[\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_t[\left\|(\boldsymbol{\nu}\mathbf{x}_t - \mathbf{C}\mathbf{A}^{T-1}\mathbf{h}_{t-T+1} - \boldsymbol{\zeta}_t)\mathbf{x}_t'\right\|_F^2]\right]\right] \\
&\le \|\boldsymbol{\nu}\|_F^2\, \mathbb{E}_{\mathbf{u}}[\left\|\mathbf{x}_t\mathbf{x}_t'\right\|_F^2] + \|\mathbf{C}\|_F^2\, \left\|\mathbf{A}^{T-1}\right\|_F^2 \mathbb{E}_{\mathbf{u}}[\|\mathbf{h}_{t-T+1}\|_2^2]\mathbb{E}_{\mathbf{u}}[\|\mathbf{x}_t\|_2^2] + \mathbb{E}_{\mathbf{u}}[\|\mathbf{x}_t\|_2^2]\mathbb{E}_{\boldsymbol{\zeta}}[\|\boldsymbol{\zeta}_t\|_2^2] \\
&\le m^2\, T^2(\max(\boldsymbol{\sigma}^{\cdot 2}))^2\chi_N^2 + n^2 mT\max(\boldsymbol{\sigma}^{\cdot 2})\ell\rho(\mathbf{A})^{2(T-1)}\gamma\,\|\mathbf{C}\|_F^2 + pmT\max(\boldsymbol{\sigma}^{\cdot 2})\max(\boldsymbol{\sigma}_\zeta^{\cdot 2}).
\end{aligned}
$$

Using the above inequality, we continue (7.24) as follow:

$$
\begin{aligned}
&\mathbb{E}_{\boldsymbol{\zeta}}\left[\mathbb{E}_{\mathbf{u}}[\mathbb{E}_t[\|\boldsymbol{\omega}_{\tau+1}\|_F^2]]\right] \le \|\boldsymbol{\omega}_\tau\|_F^2 (1 - 2\eta m\, T \min(\boldsymbol{\sigma}^{\cdot 2}) + 2\eta^2 m^2\, T^2 \min(\boldsymbol{\sigma}^{\cdot 2}) \max(\boldsymbol{\sigma}^{\cdot 2})) \\
&+ (\eta m T \max(\boldsymbol{\sigma}^{\cdot 2}) + \eta^2 m^2\, T^2(\max(\boldsymbol{\sigma}^{\cdot 2}))^2)(\|\boldsymbol{\nu}\|_F^2 + \|\boldsymbol{\omega}_0\|_F^2) \\
&+ 2\eta^2\left[m^2\, T^2(\max(\boldsymbol{\sigma}^{\cdot 2}))^2\chi_N^2 + n^2 mT\max(\boldsymbol{\sigma}^{\cdot 2})\ell\rho(\mathbf{A})^{2(T-1)}\gamma\,\|\mathbf{C}\|_F^2\right. \\
&\left.+ pmT\max(\boldsymbol{\sigma}^{\cdot 2})\max(\boldsymbol{\sigma}_\zeta^{\cdot 2})\right]. \qquad (7.25)
\end{aligned}
$$

We observe if $1-2\eta mT\min(\boldsymbol{\sigma}^{\cdot 2})+2\eta^2 m^2 T^2\min(\boldsymbol{\sigma}^{\cdot 2})\max(\boldsymbol{\sigma}^{\cdot 2}) \le 1$, we obtain $\mathbb{E}_t[\mathbb{E}_{\mathbf{u}}[\|\boldsymbol{\omega}_{\tau+1}\|_F^2]] \le \|\boldsymbol{\omega}_\tau\|_F^2$. Therefore, η should satisfy

$$
\eta \le \frac{1}{m\, T\max(\boldsymbol{\sigma}^{\cdot 2})}. \qquad (7.26)
$$

To make the additive constant terms in (7.25) small enough, we can choose η close to zero although very small η makes the coefficient of $\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_{\boldsymbol{\zeta}}\left[\|\boldsymbol{\omega}_\tau\|_F^2\right]\right]$ close to one and makes the convergence rate slow. Therefore, more iterations are required to reach a certain error.

From recursion, we bound the Frobenius norm distance between the solution of (7.11) and the initial point as follows:

$$\begin{aligned}
&\mathbb{E}_{\mathbf{t}}\left[\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_{\boldsymbol{\zeta}}\left[\|\boldsymbol{\omega}_\tau\|_F^2\right]\right]\right] < \|\boldsymbol{\omega}_0\|_F^2\left(1-2\eta m\,T\min(\boldsymbol{\sigma}^{\cdot 2})+2\eta^2m^2\,T^2\min(\boldsymbol{\sigma}^{\cdot 2})\max(\boldsymbol{\sigma}^{\cdot 2})\right)^\tau\\
&+\sum_{i=0}^{\tau-1}\left(1-2\eta m\,T\min(\boldsymbol{\sigma}^{\cdot 2})+2\eta^2m^2\,T^2\min(\boldsymbol{\sigma}^{\cdot 2})\max(\boldsymbol{\sigma}^{\cdot 2})\right)^i\,\Big[2\eta^2m^2\,T^2(\max(\boldsymbol{\sigma}^{\cdot 2}))^2\chi_N^2\\
&+2n^2\eta^2mT\max(\boldsymbol{\sigma}^{\cdot 2})\ell\rho(\mathbf{A})^{2(T-1)}\gamma\,\|\mathbf{C}\|_F^2+2\eta^2pmT\max(\boldsymbol{\sigma}^{\cdot 2})\max(\boldsymbol{\sigma}_\zeta^{\cdot 2})\\
&+(\eta mT\max(\boldsymbol{\sigma}^{\cdot 2})+\eta^2m^2\,T^2(\max(\boldsymbol{\sigma}^{\cdot 2}))^2)(\|\boldsymbol{\nu}\|_F^2+\|\boldsymbol{\omega}_0\|_F^2)\Big]\\
&<\|\boldsymbol{\omega}_0\|_F^2\left(1-2\eta m\,T\min(\boldsymbol{\sigma}^{\cdot 2})+2\eta^2m^2\,T^2\min(\boldsymbol{\sigma}^{\cdot 2})\max(\boldsymbol{\sigma}^{\cdot 2})\right)^\tau\\
&+\frac{2\eta^2m^2\,T^2(\max(\boldsymbol{\sigma}^{\cdot 2}))^2\chi_N^2+(\eta mT\max(\boldsymbol{\sigma}^{\cdot 2})+\eta^2m^2\,T^2(\max(\boldsymbol{\sigma}^{\cdot 2}))^2)(\chi_N^2+\|\boldsymbol{\omega}_0\|_F^2)}{1-2\eta m\,T\min(\boldsymbol{\sigma}^{\cdot 2})+2\eta^2m^2\,T^2\min(\boldsymbol{\sigma}^{\cdot 2})\max(\boldsymbol{\sigma}^{\cdot 2})}\\
&+\frac{2n^2\eta^2mT\max(\boldsymbol{\sigma}^{\cdot 2})\ell\rho(\mathbf{A})^{2(T-1)}\gamma\,\|\mathbf{C}\|_F^2+2\eta^2pmT\max(\boldsymbol{\sigma}^{\cdot 2})\max(\boldsymbol{\sigma}_\zeta^{\cdot 2})}{1-2\eta m\,T\min(\boldsymbol{\sigma}^{\cdot 2})+2\eta^2m^2\,T^2\min(\boldsymbol{\sigma}^{\cdot 2})\max(\boldsymbol{\sigma}^{\cdot 2})}.
\end{aligned}$$

The additive terms can become as small as desired by adjusting η. With smaller step-size, the proposed offline SGD requires additional iterations to reach a certain neighborhood of the ground truth solution. Suppose that the distance between $\hat{\boldsymbol{\Theta}}_{\tau,T}$ and the ground truth $\boldsymbol{\Theta}_T$ is denoted by $\boldsymbol{\phi}_\tau$. Then, we bound $\mathbb{E}_{\mathbf{t}}\left[\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_{\boldsymbol{\zeta}}[\boldsymbol{\phi}_\tau]\right]\right]$ as follows:

$$\begin{aligned}
&\mathbb{E}_{\mathbf{t}}\left[\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_{\boldsymbol{\zeta}}\left[\|\boldsymbol{\phi}_\tau\|_F^2\right]\right]\right]=\mathbb{E}_{\mathbf{t}}\left[\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_{\boldsymbol{\zeta}}\left[\left\|\hat{\boldsymbol{\Theta}}_{\tau,T}-\boldsymbol{\Theta}_T\right\|_F^2\right]\right]\right]\\
&=\mathbb{E}_{\mathbf{t}}\left[\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_{\boldsymbol{\zeta}}\left[\left\|\hat{\boldsymbol{\Theta}}_{\tau,T}-\hat{\boldsymbol{\Theta}}_T+\hat{\boldsymbol{\Theta}}_T-\boldsymbol{\Theta}_T\right\|_F^2\right]\right]\right]\\
&=\mathbb{E}_{\mathbf{t}}\left[\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_{\boldsymbol{\zeta}}\left[\|\boldsymbol{\omega}_\tau+\boldsymbol{\nu}\|_F^2\right]\right]\right]\leq\mathbb{E}_{\mathbf{t}}\left[\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_{\boldsymbol{\zeta}}\left[\|\boldsymbol{\omega}_\tau\|_F^2\right]\right]\right]+\mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_{\boldsymbol{\zeta}}\left[\|\boldsymbol{\nu}\|_F^2\right]\right]\\
&<\|\boldsymbol{\omega}_0\|_F^2\left(1-2\eta m\,T\min(\boldsymbol{\sigma}^{\cdot 2})+2\eta^2m^2\,T^2\min(\boldsymbol{\sigma}^{\cdot 2})\max(\boldsymbol{\sigma}^{\cdot 2})\right)^\tau\\
&\underbrace{+\frac{2\eta^2m^2\,T^2(\max(\boldsymbol{\sigma}^{\cdot 2}))^2\chi_N^2+(\eta mT\max(\boldsymbol{\sigma}^{\cdot 2})+\eta^2m^2\,T^2(\max(\boldsymbol{\sigma}^{\cdot 2}))^2)(\chi_N^2+\|\boldsymbol{\omega}_0\|_F^2)}{1-2\eta m\,T\min(\boldsymbol{\sigma}^{\cdot 2})+2\eta^2m^2\,T^2\min(\boldsymbol{\sigma}^{\cdot 2})\max(\boldsymbol{\sigma}^{\cdot 2})}\cdots}
\end{aligned}$$

$$\underbrace{\cdots + \frac{2n^2\eta^2 mT\max(\boldsymbol{\sigma}^{\cdot 2})\ell\rho(\mathbf{A})^{2(T-1)}\gamma\left\|\mathbf{C}\right\|_F^2 + 2\eta^2 pmT\max(\boldsymbol{\sigma}^{\cdot 2})\max(\boldsymbol{\sigma}_\zeta^{\cdot 2})}{1-2\eta m\,T\min(\boldsymbol{\sigma}^{\cdot 2}) + 2\eta^2 m^2\,T^2\min(\boldsymbol{\sigma}^{\cdot 2})\max(\boldsymbol{\sigma}^{\cdot 2})}}_{\Delta_N} + \chi_N^2. \tag{7.27}$$

□

Corollary 1. *The cost function in* (7.11) *is in expectation* $mT\min(\boldsymbol{\sigma}^{\cdot 2})$*-strongly convex and the Lipschitz constant for its gradient is* $m\,T\max(\boldsymbol{\sigma}^{\cdot 2})$*. When the step-size is* $\eta = \frac{1}{2m\,T\max(\boldsymbol{\sigma}^{\cdot 2})}$*, the fastest convergence rate is obtained. This convergence rate is equal to that given in the state of the art method [215, Theorem 3.1]. Compared to [187, Appendix A],* (7.20) *is tighter since in* (7.20) *the third term in parenthesis depends linearly on the Lipschitz constant, while the dependence is quadratic in [187, Appendix A]. The dependence of the SGD error bound on the batch size and the truncation length is not studied in any of the aforementioned papers.*

Theorem 2 states that Algorithm 22 linearly converges up to the sum of two additive constant terms, which are calibrated by $\boldsymbol{\sigma}_\zeta^{\cdot 2}$, $\mathbf{h}_0$, T, η, $\boldsymbol{\sigma}^{\cdot 2}$, and the batch size N as given in (7.27). With a small enough Δ_N, Algorithm 22 linearly converges to a region with a maximum distance of $\mathcal{O}(\frac{1}{\sqrt{N}})$ to the ground truth Markov parameters since χ_N decreases with rate $\mathcal{O}(\frac{1}{\sqrt{N}})$. We observe from (7.27) that one can make the two additive terms as small as desired by increasing N, which decreases χ_N^2, and picking a smaller η, which slows down the convergence rate of Algorithm 22. One drawback of full-batch methods in [187, 188, 191] is that they simultaneously require all the samples to be stored and processed. Although Algorithm 22 decreases the cost of computation by utilizing one input-output pair in each iteration, it requires all samples to be stored. We proved in Theorem 1 that increasing the batch size helps to reach a closer neighborhood of the ground truth solution. However, storing a large batch of input-output pairs is challenging and storage inefficient. Therefore, we propose an online SGD that does not require samples to be stored.

7.4.2 Online SGD

```
Algorithm 23: Online SGD algorithm to learn $\mathbf{\Theta}_T$
Initialization: Assign small value to $\hat{\mathbf{\Theta}}_{0,T}$, $t = 1$
Input: $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{t}$, learning rate $\eta$
Output: Estimation of $\mathbf{\Theta}_T$
if a new input-output pair arrives then
    $\hat{\mathbf{\Theta}}_{t,T} = \hat{\mathbf{\Theta}}_{t-1,T} - \eta(\hat{\mathbf{\Theta}}_{t-1,T}\mathbf{x}_t - \mathbf{y}_t)\mathbf{x}_t'$
    $t = t + 1$
end
```

Motivated by Theorem 1, we propose an algorithm which utilizes newly arrived samples and discards the old ones. We develop an SGD algorithm to learn $\mathbf{\Theta}_T$ in an online streaming fashion. The online SGD algorithm implements the descent on the loss function (7.11) in each iteration using a gradient obtained from the most recent input-output pair at time instance t as follows:

$$\hat{\mathbf{\Theta}}_{t,T} = \hat{\mathbf{\Theta}}_{t-1,T} - \eta(\hat{\mathbf{\Theta}}_{t-1,T}\mathbf{x}_t - \mathbf{y}_t)\mathbf{x}_t'. \tag{7.28}$$

In each time instance t, one iteration is implemented. The proposed online SGD algorithm is summarized in Algorithm 23; we provide a corresponding convergence guarantee below.

7.5 Transfer Function Estimation and Recovery of Weight Matrices

The transfer function of a linear dynamical system is obtained by taking z-transformation of the impulse response of the system and is computed as follows [216, p. 267–p. 268]:

$$\mathbf{G}(z) = \sum_{t=1}^{\infty} z^{-t}\mathbf{C}\mathbf{A}^{t-1}\mathbf{B} + \mathbf{D} = \mathbf{C}(z\mathbf{I}_{n\times n} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}.$$

We can rewrite the above transfer function as follows:

$$\mathbf{G}(z) = \sum_{t=1}^{T-1} z^{-t}\mathbf{C}\mathbf{A}^{t-1}\mathbf{B} + \mathbf{D} + \mathbf{E}_{z,T}, \tag{7.29}$$

where $\mathbf{E}_{z,T} = \sum_{t=T}^{\infty} z^{-t}\mathbf{C}\mathbf{A}^{t-1}\mathbf{B}$. Given a large enough T, the Frobenius norm of $\mathbf{E}_{z,T}$ becomes close to zero as shown in the following lemma.

Lemma 3. *The truncation error in computing the transfer function is upper-bounded as follows:*

$$\|\mathbf{E}_{z,T}\|_F^2 = \left\|\sum_{t=T}^{\infty} z^{-t}\mathbf{C}\mathbf{A}^{t-1}\mathbf{B}\right\|_F^2 \leq \frac{n^2\ell\, \|\mathbf{C}\|_F^2 \, \|\mathbf{B}\|_F^2 \, \rho(\mathbf{A})^{2(T-1)}}{1-\rho(\mathbf{A})^2|z|^{-2}}. \tag{7.30}$$

Give a large T, the RHS of (7.30) *tends to zero and the LHS is enforced to be very small.*

When $\|\mathbf{E}_{z,T}\|_F^2$ is small enough, we can efficiently approximate $\mathbf{G}(z)$ using T Markov parameters: $\{\mathbf{C}\mathbf{A}^{t-1}\mathbf{B}\}_{t=1}^{T-1}$ and $\mathbf{D}$, which are learned by Algorithms 22 and 23. Upon the convergence of $\hat{\mathbf{\Theta}}_{t,T}$ (or $\hat{\mathbf{\Theta}}_{\tau,T}$), the first Markov parameter, $\mathbf{D}$, is learned and needs no further processing. To recover $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ from the estimated transfer function, we assume $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ have Brunovsky canonical form [217], which is perhaps the most popular canonical form [209, 210, 211]. In Brunovsky canonical form, we have:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_{m\times m} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{m\times m} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I}_{m\times m} \\ -a_n\mathbf{I}_{m\times m} & -a_{n-1}\mathbf{I}_{m\times m} & -a_{n-2}\mathbf{I}_{m\times m} & \cdots & -a_1\mathbf{I}_{m\times m} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{I}_{m\times m} \end{bmatrix}, \tag{7.31}$$

where $\mathbf{A} \in \mathbb{R}^{nm\times nm}$, $\mathbf{B} \in \mathbb{R}^{nm\times m}$, and $\mathbf{C} \in \mathbb{R}^{p\times nm}$. To recover matrix $\mathbf{A}$, it is enough to find $\{a_i\}_{i=1}^n$. To recover $\mathbf{C}$, all elements should be estimated. In Brunovsky canonical form, $\mathbf{B}$ is known as given in (7.31). The above special forms for $\mathbf{A}$ and $\mathbf{B}$ matrices help to find unknowns. If $\mathbf{A}$ and $\mathbf{B}$ are in Brunovsky canonical form, $\mathbf{G}(z)$ is obtained as follows:

$$\mathbf{G}(z) = \mathbf{C}\mathbf{S}(z) + \mathbf{D}, \tag{7.32}$$

where $\mathbf{S}(z) = (z\mathbf{I}_{nm\times nm} - \mathbf{A})^{-1}\mathbf{B}$ and can be rewritten as follows [188, Lemma B.1]:

$$\mathbf{S}(z) = \frac{1}{z^n + a_1 z^{n-1} + \cdots + a_n} \underbrace{\begin{bmatrix} \mathbf{I}_{m\times m} & z\mathbf{I}_{m\times m} & \ldots & z^{n-1}\mathbf{I}_{m\times m} \end{bmatrix}'}_{\mathbf{W}},$$

where $\mathbf{W} \in \mathbb{C}^{nm\times m}$. The denominator of $\mathbf{S}(s)$ is called *characteristic polynomial* and is denoted by $q(z)$. If $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ have Brunovsky canonical form, the transfer function (7.32) is uniquely realized by the state-space representation [188]. When the linear dynamical system is SISO, i.e., $m = p = 1$, Brunovsky canonical form reduces to the *controllable canonical form*. We match (7.29) and (7.32) as follows:

$$\sum_{t=1}^{T-1} z^{-t}\mathbf{C}\mathbf{A}^{t-1}\mathbf{B} = \mathbf{C}\mathbf{S}(z) - \mathbf{E}_{z,T}. \tag{7.33}$$

The LHS of the above equation can be efficiently estimated using $\hat{\boldsymbol{\Theta}}_{\tau,T}$ (or $\hat{\boldsymbol{\Theta}}_{t,T}$) from the regression problem. In Brunovsky canonical form, there are n and pnm unknown elements in $\mathbf{A}$ and $\mathbf{C}$, respectively. We need at least $n + pnm$ equations to identify unknowns. To find $n + pnm$ equations, we match both sides of (7.33) in $n + pnm$ complex frequencies. In particular, we choose z such that it does not yield $\left\| \sum_{t=1}^{T-1} z^{-t}\mathbf{C}\mathbf{A}^{t-1}\mathbf{B} \right\|_F = 0$ or make it unbounded. For example, one can choose frequencies on the unit circle $z_k = e^{j\frac{\pi(k-1)}{n+pnm}}$, $k \in \{1, \ldots, n + pnm\}$ if none of them is a pole or zero of $\sum_{t=1}^{T-1} z^{-t}\mathbf{C}\mathbf{A}^{t-1}\mathbf{B}$. By choosing $|z_k| = 1$, one can avoid the linear system of equations built using (7.33) from becoming ill-conditioned. When $|z_k| \neq 1$ and n is large, $\{z_k^{n-i}\}_{i=1}^{n}$, which are coefficients of $\{a_i\}_{i=1}^{n}$, become very different in terms of their absolute value, and the linear system of equations becomes ill-conditioned. Each side of (7.33) is a $p \times m$ matrix and yields pm equations in each frequency. Therefore, having $n+pnm$ frequencies yields an over-determined consistent system. To represent the LHS of (7.33) in a compact form, we define $\boldsymbol{\vartheta}$ as follows:

$$\boldsymbol{\vartheta} = \begin{bmatrix} \mathbf{0}_{m\times m} & z^{-1}\mathbf{I}_{m\times m} & z^{-2}\mathbf{I}_{m\times m} & \ldots & z^{-T+1}\mathbf{I}_{m\times m} \end{bmatrix}'.$$

Suppose $\boldsymbol{\vartheta}_k = \boldsymbol{\vartheta}|_{z=z_k}$ and $\mathbf{W}_k = \mathbf{W}|_{z=z_k}$. The linear system of equations is obtained as follows:

$$\hat{\boldsymbol{\Theta}}_{\tau,T}\, \boldsymbol{\vartheta}_k q(z_k) = \mathbf{C}\mathbf{W}_k - q(z_k)\mathbf{E}_{z_k,T}, \qquad \forall k \in \{1, \ldots, n + pnm\}, \tag{7.34}$$

where $\hat{\boldsymbol{\Theta}}_{\tau,T}\, \boldsymbol{\vartheta}_k$ is numerically computed by Algorithm 22, and $\mathbf{E}_{z_k,T}$ is treated as noise when $\|\mathbf{E}_{z_k,T}\|_F$ is small enough. The unknowns are embedded in $q(z)$ and $\mathbf{C}$. In general, solving a linear system is easier when compared to the SVD-based methods in [2, 198, 201].

Algorithm 24: Offline SGD combined with the linear system

```
Input: $\{\mathbf{x}_t, \mathbf{y}_t\}_{t=1}^{N}$, learning rate $\eta$, $\tau = 1$
Output: Estimation of $\mathbf{A}$, $\mathbf{C}$ and $\mathbf{D}$
for $\tau$ from 1 to END do
    Uniformly at random choose $t \in \{T, T+1, \ldots, N\}$;
    $\hat{\mathbf{\Theta}}_{\tau,T} = \hat{\mathbf{\Theta}}_{\tau-1,T} - \eta(\hat{\mathbf{\Theta}}_{\tau-1,T}\mathbf{x}_t - \mathbf{y}_t)\mathbf{x}_t'$
    Find $\mathbf{\Gamma}_{\tau,T}$ and $\varkappa_\tau$
    $\hat{\boldsymbol{\varrho}}_\tau = (\mathbf{\Gamma}_{\tau,T}^H \mathbf{\Gamma}_{\tau,T})^{-1} \mathbf{\Gamma}_{\tau,T}^H \varkappa_\tau$
end
```

Algorithm 25: Online SGD combined with the linear system

```
Input: $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{t}$, learning rate $\eta$, $t = 1$
Output: Estimation of $\mathbf{A}$, $\mathbf{C}$ and $\mathbf{D}$
if a new input-output pair arrives then
    $\hat{\mathbf{\Theta}}_{t,T} = \hat{\mathbf{\Theta}}_{t-1,T} - \eta(\hat{\mathbf{\Theta}}_{t-1,T}\mathbf{x}_t - \mathbf{y}_t)\mathbf{x}_t'$
    Find $\mathbf{\Gamma}_{t,T}$ and $\varkappa_t$
    $\hat{\boldsymbol{\varrho}}_t = (\mathbf{\Gamma}_{t,T}^H \mathbf{\Gamma}_{t,T})^{-1} \mathbf{\Gamma}_{t,T}^H \varkappa_t$
    $t = t + 1$
end
```

We guarantee a unique solution for (7.34). Let the vector of unknowns be denoted by $\boldsymbol{\varrho} = \{\{a_i\}_{i=1}^{n}, \{c_{i,j}\}_{i=1:p,j=1:mn}\}$. Then, one can rewrite (7.34) in the standard form of linear system of equations easily, as will be explained later, as follows:

$$\mathbf{\Gamma}_{\tau,T}\boldsymbol{\varrho}_\tau = \varkappa_\tau, \tag{7.35}$$

where $\mathbf{\Gamma}_{\tau,T}$ and $\varkappa_\tau$ are calculated using $\hat{\mathbf{\Theta}}_{\tau,T}$. The above equation can be solved either by the pseudo-inverse method or iterative methods, e.g., [218, 219, 220]. Consider in each iteration of Algorithm 22, we solve (7.35) by the pseudo-inverse method as given in Algorithm 24. Theorem 3 ensures the linear convergence of the vector of unknowns returned by Algorithm 24 upon the convergence of $\hat{\mathbf{\Theta}}_{\tau,T}$. We can combine Algorithm 23 with the above system of equations and obtain an online method to derive weight matrices as given in Algorithm 25.

Theorem 3. *The linear convergence of $\hat{\mathbf{\Theta}}_{\tau,T}$ imposes a linear convergence to $\hat{\boldsymbol{\varrho}}_\tau$.*

Proof. When $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are in Brunovsky canonical form, the transfer function (7.32) is uniquely realized by the state-space representation [188]. Due to [188, Lemma B.1], one can observe that the LHS of (7.32) uniquely realizes the RHS of (7.32) and vice versa.

We notice that the linear system in (7.34) is consistent. The reason is that (7.33) holds in all frequencies including the chosen z_k. Using $\mathbf{W}_k$, we rewrite (7.34) in the form of a linear system of equations as follows:

$$\underbrace{\begin{bmatrix} -(\boldsymbol{\Theta}_T\,\boldsymbol{\vartheta}_1+\mathbf{E}_{z_1,T})'z_1^{n-1} & \cdots & -(\boldsymbol{\Theta}_T\,\boldsymbol{\vartheta}_1+\mathbf{E}_{z_1,T})' & \mathbf{W}_1' \\ -(\boldsymbol{\Theta}_T\,\boldsymbol{\vartheta}_2+\mathbf{E}_{z_2,T})'z_2^{n-1} & \cdots & -(\boldsymbol{\Theta}_T\,\boldsymbol{\vartheta}_2+\mathbf{E}_{z_2,T})' & \mathbf{W}_2' \\ \vdots & \ddots & \vdots & \vdots \\ -(\boldsymbol{\Theta}_T\,\boldsymbol{\vartheta}_{n+pnm}+\mathbf{E}_{z_{n+pnm},T})'z_{n+pnm}^{n-1} & \cdots & -(\boldsymbol{\Theta}_T\,\boldsymbol{\vartheta}_{n+pnm}+\mathbf{E}_{z_{n+pnm},T})' & \mathbf{W}_{n+pnm}' \end{bmatrix}}_{\boldsymbol{\Psi}_T}$$

$$\underbrace{\begin{bmatrix} a_1 \\ \vdots \\ a_n \\ \mathbf{C}' \end{bmatrix}}_{\boldsymbol{\varphi}} = \underbrace{\begin{bmatrix} (\boldsymbol{\Theta}_T\,\boldsymbol{\vartheta}_1 z_1^n)' \\ (\boldsymbol{\Theta}_T\,\boldsymbol{\vartheta}_2 z_2^n)' \\ \vdots \\ (\boldsymbol{\Theta}_T\,\boldsymbol{\vartheta}_{n+pnm} z_{n+pnm}^n)' \end{bmatrix}}_{\boldsymbol{\kappa}}. \tag{7.36}$$

In $\boldsymbol{\Psi}_T$, in each block row, all $\{(\boldsymbol{\Theta}_T\,\boldsymbol{\vartheta}_k+\mathbf{E}_{z_k,T})'z_k^{n-v}\}_{v=1}^n$ blocks are $m\times p$ matrix blocks except the last one, $\mathbf{W}_k$, which is an $m\times nm$ matrix block. The first n blocks, $\{(\boldsymbol{\Theta}_T\,\boldsymbol{\vartheta}_k+\mathbf{E}_{z_k,T})'z_k^{n-v}\}_{v=1}^n$, in each row are multiplied by $\{a_v\}_{v=1}^n$ elements in $\boldsymbol{\varphi}$, and $\mathbf{W}_k$ is multiplied by $\mathbf{C}'$. In the above equation, $\boldsymbol{\kappa}$ is made of $n+pnm$ block matrices stacked vertically, each with the dimension $m\times p$. We note that (7.36) is made of $n+pnm$ block rows. The k^{th} block row is as follows:

$$-\sum_{v=1}^{n} a_v(\boldsymbol{\Theta}_T\,\boldsymbol{\vartheta}_k+\mathbf{E}_{z_k,T})'z_k^{n-v}+\mathbf{W}_k'\mathbf{C}'=(\boldsymbol{\Theta}_T\,\boldsymbol{\vartheta}_k z_k^n)', \qquad \forall k\in\{1,n+nmp\}. \tag{7.37}$$

The above block equation is $m\times p$. In the $(i,j)^{\text{th}}$ equation of block equation (7.37), the variables are $\{a_i\}_{i=1}^n$ and $\{c_{j,i+vm}\}_{v=0}^{n-1}$. The $(i,j)^{\text{th}}$ equation from the above block is as follows:

$$-\sum_{v=1}^{n} a_v[(\boldsymbol{\Theta}_T\,\boldsymbol{\vartheta}_k + \mathbf{E}_{z_k,T})']_{ij} z_k^{n-v} + \sum_{v=0}^{n-1} c_{j,i+vm} z_k^v = [(\boldsymbol{\Theta}_T\,\boldsymbol{\vartheta}_k z_k^n)']_{ij}, \tag{7.38}$$

We stack (7.38) for all frequencies, $\{z_k\}_{k=1}^{n+nmp}$, and form the following linear system:

$$\underbrace{\begin{bmatrix} -[(\boldsymbol{\Theta}_T\,\boldsymbol{\vartheta}_1 + \mathbf{E}_{z_1,T})']_{ij} z_1^{n-1} & -[(\boldsymbol{\Theta}_T\,\boldsymbol{\vartheta}_1 + \mathbf{E}_{z_1,T})']_{ij} z_1^{n-2} & \dots & -[(\boldsymbol{\Theta}_T\,\boldsymbol{\vartheta}_1 + \mathbf{E}_{z_1,T})']_{ij} & 1 & z_1 & \dots & z_1^{n-1} \\ -[(\boldsymbol{\Theta}_T\,\boldsymbol{\vartheta}_2 + \mathbf{E}_{z_2,T})']_{ij} z_2^{n-1} & -[(\boldsymbol{\Theta}_T\,\boldsymbol{\vartheta}_2 + \mathbf{E}_{z_2,T})']_{ij} z_2^{n-2} & \dots & -[(\boldsymbol{\Theta}_T\,\boldsymbol{\vartheta}_2 + \mathbf{E}_{z_2,T})']_{ij} & 1 & z_2 & \dots & z_2^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -[(\boldsymbol{\Theta}_T\,\boldsymbol{\vartheta}_{n+pnm} + \mathbf{E}_{z_{n+pnm},T})']_{ij} z_{n+nmp}^{n-1} & -[(\boldsymbol{\Theta}_T\,\boldsymbol{\vartheta}_{n+pnm} + \mathbf{E}_{z_{n+pnm},T})']_{ij} z_{n+nmp}^{n-2} & \dots & -[(\boldsymbol{\Theta}_T\,\boldsymbol{\vartheta}_{n+nmp} + \mathbf{E}_{z_{n+nmp,T}})']_{ij} & 1 & z_{n+nmp} & \dots & z_{n+nmp}^{n-1} \end{bmatrix}}_{\mathbf{R}_{ij}} \begin{bmatrix} \{a_v\}_{v=1}^{n} \\ \{c_{j,i+vm}\}_{v=0}^{n-1} \end{bmatrix}$$

$$= \underbrace{\begin{bmatrix} [(\boldsymbol{\Theta}_T\,\boldsymbol{\vartheta}_1 z_1^n)']_{ij} \\ [(\boldsymbol{\Theta}_T\,\boldsymbol{\vartheta}_2 z_2^n)']_{ij} \\ \vdots \\ [(\boldsymbol{\Theta}_T\,\boldsymbol{\vartheta}_{n+nmp} z_{n+nmp}^n)']_{ij} \end{bmatrix}}_{\mathbf{r}_{ij}}. \tag{7.39}$$

We notice that the exponential of a frequency, denoted by z^v, is different from the frequency z. We investigate the linear dependency of the columns in the above matrix, where the k^{th} row corresponds to the k^{th} frequency. We note that in $\mathbf{R}_{ij}$, each element in a row incorporates a particular set of frequencies. The set of frequencies embedded in $\boldsymbol{\vartheta}_k$ is $\{z_k^{-1}, \dots, z_k^{-T+1}\}$. Given that $\mathbf{E}_{z_k,T}$ is negligible, we list the sets of frequencies incorporated in the coefficients of each unknown in a row of (7.39) as follows:

$$c_{j,i+vm} \longrightarrow z_k^v, \qquad v \in \{0, \dots, n-1\}, \tag{7.40a}$$

$$a_v \longrightarrow \{z_k^{n-v-1}, z_k^{n-v-2}, \dots, z_k^{n-v-T+1}\}, \qquad v \in \{1, \dots, n\}. \tag{7.40b}$$

The set of frequencies in the coefficient of a_1 is $\{z_k^{n-2}, z_k^{n-3}, \ldots, z_k^{n-T}\}$. The set of frequencies in the coefficient of a_2 is $\{z_k^{n-3}, z_k^{n-4}, \ldots, z_k^{n-1-T}\}$. We observe that the frequency z_k^{n-1-T} does not exist in the set of frequencies in the coefficient of a_1. Similarly, the coefficient of a_v incorporates frequency $z_k^{n-v-T+1}$ that does not exist in the coefficients of $\{a_1, \ldots, a_{v-1}\}$. Furthermore, we observe from (7.40a) that the coefficient of $c_{j,i+vm}$ is z_k^v. The frequencies in the coefficients of $\{c_{j,i+vm}\}_{v=0}^{n-1}$ are separate and do not overlap. Given that $T \geq n+1$, we obtain $n - v - T + 1 < 0$. There is at least one frequency $z^{n-v-T+1}$ in the coefficient of a_v that do not appear in the frequencies incorporated in the coefficients of $\{c_{j,i+vm}\}_{v=0}^{n-1}$ since $n - v - T + 1 < 0$.

Let us suppose that the columns of the coefficient matrix in (7.39) are linearly dependent. In this case, we have:

$$-\sum_{v=1}^{n}[(\mathbf{\Theta}_T\,\boldsymbol{\vartheta}_k + \mathbf{E}_{z_k,T})']_{ij} z_k^{n-v}\alpha_a^v + \sum_{v=1}^{n} z_k^{v-1}\alpha_c^v = 0, \tag{7.41}$$

where α_a^v and α_c^v are given coefficients to the v^{th} and $v+n^{\text{th}}$, $v \in \{1, \ldots, n\}$, columns of $\mathbf{R}_{ij}$, respectively, to ensure the linear dependency of columns. We consider two possibilities:

1. Suppose that each α_a^v is zero, i.e., $\{\alpha_a^v\}_{v=1}^n = \{0\}$. Then, (7.41) implies that $\sum_{v=1}^n z_k^{v-1}\alpha_c^v = 0$. When at least two different α_c^v are non-zero, $\sum_{v=1}^n z_k^{v-1}\alpha_c^v = 0$ yields a polynomial of z, in which z_k is a root. However, we note that z_k is arbitrarily chosen by us. It is impossible that a polynomial of z with a finite degree has infinite roots. Therefore, by contradiction, we conclude that the columns $v+1$ to $v+n$ of $\mathbf{R}_{ij}$ are linearly independent of each other.

2. Suppose that at least one α_a^v, $\forall v \in \{1, \ldots, n\}$, is not zero. Then, we consider $[(\mathbf{\Theta}_T\boldsymbol{\vartheta}_k + \mathbf{E}_{z_k,T})']_{ij} z_k^{n-v}\alpha_a^v$. We choose the largest v such that $\alpha_a^v \neq 0$. From (7.40b), we see that the frequency $z_k^{n-v-T+1}$ appears only in $[(\mathbf{\Theta}_T\,\boldsymbol{\vartheta}_k + \mathbf{E}_{z_k,T})']_{ij} z_k^{n-v}\alpha_a^v$ and it does not exist in the other elements of the k^{th} row, which are given in (7.41). The reason is that we picked the largest v and $n - v - T + 1$ is the least exponent for z_k in (7.41) . Hence, due to its uniqueness, $z_k^{n-v-T+1}$ cannot be removed by the linear combination of different elements in (7.41). Based on this fact, (7.41) is always at least a polynomial of $z^{n-v-T+1}$, in which an arbitrary z_k is a root. It is impossible that

Algorithm 26: Transforming (7.34) to the standard form of linear system of equations

```
Initialization: $\mathbf{\Gamma}_T = \mathbf{0}_{mp(n+nmp)\times(n+nmp)}$, $\varkappa = \mathbf{0}_{mp(n+nmp)\times 1}$
Input: (7.34) for all $(i,j) \in \{(i,j) \mid i \in \{1,\dots,m\}, j \in \{1,\dots,p\}\}$
Output: $\mathbf{\Gamma}_T$, $\varkappa$
for all $(i,j) \in \{(i,j) \mid i \in \{1,\dots,m\}, j \in \{1,\dots,p\}\}$ do
    Find $\mathbf{R}_{ij}$ and $\mathbf{r}_{ij}$ from (7.39)
    $\mathbf{\Gamma}_T((n+nmp)((i-1)p+j-1)+1 : (n+nmp)((i-1)p+j), 1:n) = \mathbf{R}_{ij}(:,1:n)$
    for $v \in \{0,\dots,n-1\}$ do
        $\mathbf{\Gamma}_T((n+nmp)((i-1)p+j-1)+1 :$
          $(n+nmp)((i-1)p+j), n+(j-1)nm+i+vm) = \mathbf{R}_{ij}(:,n+v+1)$
    end
    $\varkappa((n+nmp)((i-1)p+j-1)+1 : (n+nmp)((i-1)p+j), 1) = \mathbf{r}_{ij}$
end
Return: $\mathbf{\Gamma}_T$, $\varkappa$
```

a polynomial of z with a finite degree has infinite roots. Therefore, by contradiction, we conclude that the columns of $\mathbf{R}_{ij}$ are linearly independent of each other.

Since the coefficient matrix for the linear system (7.39) is full-rank, one can identify both blocks of variables, i.e., $\{a_v\}_{v=1}^{n}$ and $\{c_{j,i+vm}\}_{v=0}^{n-1}$. By changing i in the range $\{1,\dots,m\}$ and j in the range $\{1,\dots,p\}$, one can identify all elements of $\mathbf{C}$. We can rewrite (7.36) in the standard form of a linear system of equations $\mathbf{\Gamma}_T\boldsymbol{\varrho} = \varkappa$ by using Algorithm 26. Algorithm 26 stacks (7.36) for different i and j one after the other, while it includes all coefficients for all elements of $\mathbf{C}$.

In general, $\mathbf{\Psi}_T$ is a tall matrix. The system $\mathbf{\Psi}_T\boldsymbol{\varrho} = \boldsymbol{\kappa}$ can be solved by different numerical approaches (e.g., [218, 219, 220]).

We observe that both $\mathbf{\Psi}_T$ and $\boldsymbol{\kappa}$ are linearly parameterized by $\mathbf{\Theta}_T\,\boldsymbol{\vartheta}_k$. In addition, $\mathbf{E}_{z_k,T} = \sum_{t=T}^{\infty} z_k^{-t}\mathbf{C}\mathbf{A}^{t-1}\mathbf{B}$ appears in $\mathbf{\Psi}_T$. Let us represent the matrix $\mathbf{\Psi}_T$ by $\mathbf{\Psi}_{\tau,T}$ when 1) $\mathbf{\Psi}_T$ is parameterized by $\hat{\mathbf{\Theta}}_{\tau,T}\boldsymbol{\vartheta}_k$; and 2) $\mathbf{E}_{z_k,T} = \mathbf{0}$. Moreover, $\boldsymbol{\kappa}_t$ is parameterized by $\hat{\mathbf{\Theta}}_{\tau,T}$. Based on this, we demonstrate that the linear convergence of $\hat{\mathbf{\Theta}}_{\tau,T}$ enforces the solution of $\mathbf{\Psi}_{\tau,T}\hat{\boldsymbol{\varphi}}_t = \boldsymbol{\kappa}_\tau$ to linearly converge to the ground truth values. We assume $|z_k| = 1$. We have:

$$\left.\begin{aligned}\mathbf{\Psi}_{\tau,T}\hat{\boldsymbol{\varphi}}_\tau &= \boldsymbol{\kappa}_\tau\\ \mathbf{\Psi}_T\boldsymbol{\varphi} &= \boldsymbol{\kappa}\end{aligned}\right\} \to \mathbf{\Psi}_{\tau,T}\hat{\boldsymbol{\varphi}}_\tau - \mathbf{\Psi}_T\boldsymbol{\varphi} = \boldsymbol{\kappa}_\tau - \boldsymbol{\kappa}.$$

We expand the above equation as follows:

$$\begin{aligned}
&\boldsymbol{\Psi}_{\tau,T}\hat{\boldsymbol{\varphi}}_\tau - \boldsymbol{\Psi}_{\tau,T}\boldsymbol{\varphi} + \boldsymbol{\Psi}_{\tau,T}\boldsymbol{\varphi} - \boldsymbol{\Psi}_T\boldsymbol{\varphi} = \boldsymbol{\kappa}_\tau - \boldsymbol{\kappa}, \\
&\Rightarrow \left\| \boldsymbol{\Psi}_{\tau,T}(\hat{\boldsymbol{\varphi}}_\tau - \boldsymbol{\varphi}) + (\boldsymbol{\Psi}_{\tau,T} - \boldsymbol{\Psi}_T)\boldsymbol{\varphi} \right\|_F^2 = \left\| \boldsymbol{\kappa}_\tau - \boldsymbol{\kappa} \right\|_F^2, \\
&\overset{(a)}{\Rightarrow} \left\| \boldsymbol{\Psi}_{\tau,T}(\hat{\boldsymbol{\varphi}}_\tau - \boldsymbol{\varphi}) \right\|_F^2 - \left\| (\boldsymbol{\Psi}_{\tau,T} - \boldsymbol{\Psi}_T)\boldsymbol{\varphi} \right\|_F^2 \leq \left\| \boldsymbol{\kappa}_\tau - \boldsymbol{\kappa} \right\|_F^2, \\
&\overset{(b)}{\Rightarrow} (\min(\delta(\boldsymbol{\Psi}_{\tau,T})))^2 \left\| \hat{\boldsymbol{\varphi}}_\tau - \boldsymbol{\varphi} \right\|_F^2 \leq \left\| \boldsymbol{\kappa}_\tau - \boldsymbol{\kappa} \right\|_F^2 + \left\| (\boldsymbol{\Psi}_{\tau,T} - \boldsymbol{\Psi}_T) \right\|_F^2 \left\| \boldsymbol{\varphi} \right\|_F^2, \\
&\overset{(c)}{\Rightarrow} \mathbb{E}_{\mathbf{u}}[\mathbb{E}_{\boldsymbol{\zeta}}[\left\| \hat{\boldsymbol{\varphi}}_\tau - \boldsymbol{\varphi} \right\|_F^2]] \\
&\quad \leq \mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_{\boldsymbol{\zeta}}\left[\frac{m(T-1)(n+nmp)\left\| \hat{\boldsymbol{\Theta}}_{\tau,T} - \hat{\boldsymbol{\Theta}}_T \right\|_F^2 + nm(n+nmp)(T-1)\left\| \hat{\boldsymbol{\Theta}}_{\tau,T} - \hat{\boldsymbol{\Theta}}_T \right\|_F^2 \left\| \boldsymbol{\varphi} \right\|_F^2}{(\min(\delta(\boldsymbol{\Psi}_{\tau,T})))^2} \right] \right] \\
&\quad + \frac{n^3(n+nmp)\ell \left\| \mathbf{C} \right\|_F^2 \left\| \mathbf{B} \right\|_F^2 \rho(\mathbf{A})^{2(T-1)} \left\| \boldsymbol{\varphi} \right\|_F^2}{(1-\rho(\mathbf{A})^2)(\min(\delta(\boldsymbol{\Psi}_{\tau,T})))^2} \\
&\quad = \frac{(m(T-1)(n+nmp) + nm(n+nmp)(T-1)\left\| \boldsymbol{\varphi} \right\|_F^2)}{(\min(\delta(\boldsymbol{\Psi}_{\tau,T})))^2} \mathbb{E}_{\mathbf{u}}\left[\mathbb{E}_{\boldsymbol{\zeta}}\left[\left\| \hat{\boldsymbol{\Theta}}_{\tau,T} - \hat{\boldsymbol{\Theta}}_T \right\|_F^2 \right] \right] \\
&\quad + \underbrace{\frac{n^3(n+nmp)\ell \left\| \mathbf{C} \right\|_F^2 \left\| \mathbf{B} \right\|_F^2 \rho(\mathbf{A})^{2(T-1)} \left\| \boldsymbol{\varphi} \right\|_F^2}{(1-\rho(\mathbf{A})^2)(\min(\delta(\boldsymbol{\Psi}_{\tau,T})))^2}}_{s_1}.
\end{aligned} \tag{7.42}$$

In the above series of inequalities, we have (a) due to the triangle difference inequality. Moreover, (b) follows due to the fact that 1) $\left\| (\boldsymbol{\Psi}_{\tau,T} - \boldsymbol{\Psi}_T)\boldsymbol{\varphi} \right\|_F^2 \leq \left\| (\boldsymbol{\Psi}_{\tau,T} - \boldsymbol{\Psi}_T) \right\|_F^2 \left\| \boldsymbol{\varphi} \right\|_F^2$; and 2) $(\min(\delta(\boldsymbol{\Psi}_{\tau,T})))^2 \left\| \hat{\boldsymbol{\varphi}}_\tau - \boldsymbol{\varphi} \right\|_F^2 \leq \left\| \boldsymbol{\Psi}_{\tau,T}(\hat{\boldsymbol{\varphi}}_\tau - \boldsymbol{\varphi}) \right\|_F^2$, where $\min(\delta(\boldsymbol{\Psi}_{\tau,T}))$ is the minimum non-zero singular value of $\boldsymbol{\Psi}_{\tau,T}$. In (c), we notice that $\boldsymbol{\Psi}_{\tau,T} - \boldsymbol{\Psi}_T$ is only a function of $\hat{\boldsymbol{\Theta}}_{\tau,T} - \boldsymbol{\Theta}_T$ and $\{\sum_{t=T}^{\infty} z_k^{-t}\mathbf{C}\mathbf{A}^{t-1}\mathbf{B}\}_{k=1}^{n+nmp}$ as follows:

$$\boldsymbol{\Psi}_{\tau,T} - \boldsymbol{\Psi}_T = \underbrace{\begin{bmatrix} -((\hat{\boldsymbol{\Theta}}_{\tau,T} - \boldsymbol{\Theta}_T)\,\boldsymbol{\vartheta}_1)' z_1^{n-1} & \cdots & -((\hat{\boldsymbol{\Theta}}_{\tau,T} - \boldsymbol{\Theta}_T)\,\boldsymbol{\vartheta}_1)' & \mathbf{0} \\ -((\hat{\boldsymbol{\Theta}}_{\tau,T} - \boldsymbol{\Theta}_T)\,\boldsymbol{\vartheta}_2)' z_2^{n-1} & \cdots & -((\hat{\boldsymbol{\Theta}}_{\tau,T} - \boldsymbol{\Theta}_T)\,\boldsymbol{\vartheta}_2)' & \mathbf{0} \\ \vdots & \ddots & \vdots & \vdots \\ -((\hat{\boldsymbol{\Theta}}_{\tau,T} - \boldsymbol{\Theta}_T)\,\boldsymbol{\vartheta}_{n+pnm})' z_{n+pnm}^{n-1} & \cdots & -((\hat{\boldsymbol{\Theta}}_{\tau,T} - \boldsymbol{\Theta}_T)\,\boldsymbol{\vartheta}_{n+pnm})' & \mathbf{0} \end{bmatrix}}_{\diamondsuit}$$

$$+\underbrace{\begin{bmatrix} (\sum_{t=T}^{\infty} z_1^{-t}\mathbf{C}\mathbf{A}^{t-1}\mathbf{B})' z_1^{n-1} & \cdots & (\sum_{t=T}^{\infty} z_1^{-t}\mathbf{C}\mathbf{A}^{t-1}\mathbf{B})' & \mathbf{0} \\ (\sum_{t=T}^{\infty} z_2^{-t}\mathbf{C}\mathbf{A}^{t-1}\mathbf{B})' z_2^{n-1} & \cdots & (\sum_{t=T}^{\infty} z_2^{-t}\mathbf{C}\mathbf{A}^{t-1}\mathbf{B})' & \mathbf{0} \\ \vdots & \ddots & \vdots & \vdots \\ (\sum_{t=T}^{\infty} z_{n+pnm}^{-t}\mathbf{C}\mathbf{A}^{t-1}\mathbf{B})' z_{n+pnm}^{n-1} & \cdots & (\sum_{t=T}^{\infty} z_{n+pnm}^{-t}\mathbf{C}\mathbf{A}^{t-1}\mathbf{B})' & \mathbf{0} \end{bmatrix}}_{\clubsuit},$$

where $\diamondsuit$ can be factorized into $\hat{\mathbf{\Theta}}_{\tau,T} - \mathbf{\Theta}_T$ and a constant matrix whose norm is denoted by $nm(n + nmp)(T-1)$. When $|z| = 1$, the norm of $\clubsuit$ is bounded using Lemma 3 as follows:

$$\|\clubsuit\|_F^2 \leq \frac{n^3(n+nmp)\ell \|\mathbf{C}\|_F^2 \|\mathbf{B}\|_F^2 \rho(\mathbf{A})^{2(T-1)}}{1-\rho(\mathbf{A})^2}. \tag{7.43}$$

Due to its structure, $\boldsymbol{\kappa}_\tau - \boldsymbol{\kappa}$ can be factorized into $\hat{\mathbf{\Theta}}_{\tau,T} - \mathbf{\Theta}_T$ and a constant matrix whose norm is denoted by $m(T-1)(n+nmp)$. In (c), we decompose $\mathbf{\Psi}_{\tau,T} - \mathbf{\Psi}_\tau$ and $\boldsymbol{\kappa}_\tau - \boldsymbol{\kappa}$ and use the Cauchy–Schwarz inequality.

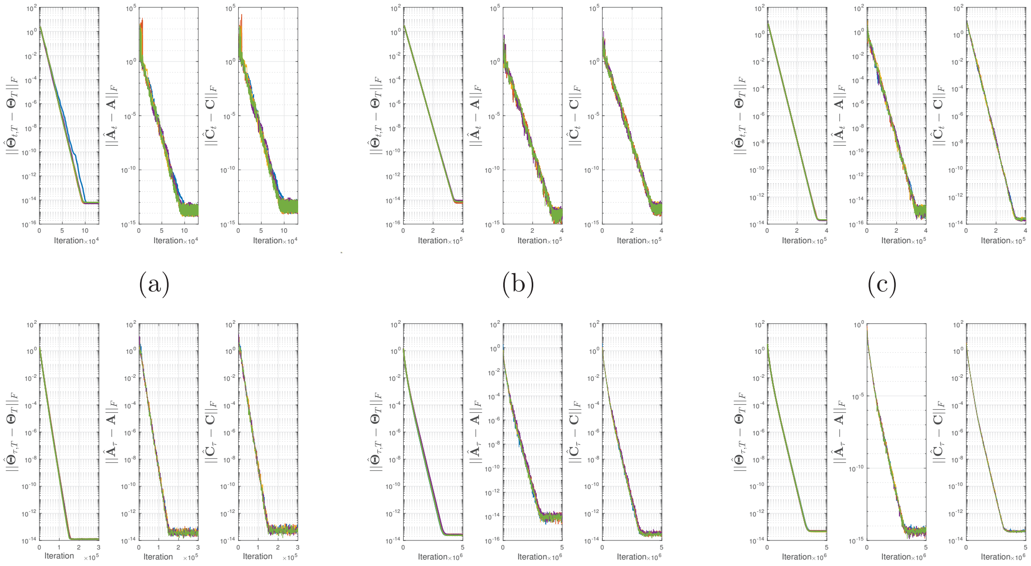
Since we have already shown that Algorithm 24 decreases $\left\|\hat{\mathbf{\Theta}}_{\tau,T} - \mathbf{\Theta}_T\right\|_F^2$ exponentially, we observe from (7.42) that $\|\hat{\boldsymbol{\varphi}}_\tau - \boldsymbol{\varphi}\|_2^2$ is enforced to be decreased at least exponentially when T is large enough to make s_1 very small. This concludes the linear convergence in expectation for the unknown parameters $\{a_i\}_{i=1}^n$ and $\{c_{i,j}\}_{i=1:p,j=1:mn}$ in $\hat{\boldsymbol{\varphi}}_\tau$ to $\boldsymbol{\varphi}$ when $\mathbf{\Psi}_{\tau,T}\hat{\boldsymbol{\varphi}}_\tau = \boldsymbol{\kappa}_\tau$ is solved in each iteration of Algorithm 24. Since $\|\hat{\boldsymbol{\varphi}}_\tau\|_F = \|\hat{\boldsymbol{\varrho}}_\tau\|_2$, Algorithm 24 linearly converges in expectation.

□

7.6 Online pseudo-inverse based method

If the computation cost of pseudo-inversion to solve (7.11) is not high, one can solve (7.11) by the pseudo-inverse method when each new input-output pair arrives. We solve (7.11) with $N = t$, where t is the number observed input-output pairs. We extract the estimated weight matrices from the linear system of equations (7.34) if $\hat{\mathbf{\Theta}}_T$ is used instead of $\hat{\mathbf{\Theta}}_{t,T}$. This approach is summarized in Algorithm 27. Although Algorithm 27 is more computationally expensive compared to Algorithm 24, it is more robust to noise. The reason is that it simultaneously uses all the available samples to estimate the Markov parameters, and due

```
Algorithm 27: Online pseudo-inverse based method
  Input: $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{t}$, learning rate $\eta$, $t = 1$
  Output: Estimation of $\mathbf{A}$, $\mathbf{C}$ and $\mathbf{D}$
  if a new input-output pair arrives then
      Update (7.11)
      Solve (7.11) by the pseudo-inverse method
      Find $\mathbf{\Gamma}_{t,T}$ and $\varkappa_t$
      $\hat{\boldsymbol{\varrho}}_t = (\mathbf{\Gamma}_{t,T}^H \mathbf{\Gamma}_{t,T})^{-1} \mathbf{\Gamma}_{t,T}^H \varkappa_t$
      $t = t + 1$
  end
```



(a)

(b)

(c)

(d)

(e)

(f)

Figure 7.1: The top row depicts the convergence of Algorithm 25. The bottom row depicts the convergence of Algorithm 24. In (a), (d) the underlying system is SISO with $n = 30$, $m = 1$, $p = 1$. In (b), (e) the considered system is MISO with $n = 5$, $m = 6$, $p = 1$. In (c), (f) the test system is MIMO with $n = 5$, $m = 6$, $p = 4$.

to averaging, the aggregate noise is attenuated.

7.7 Numerical Tests

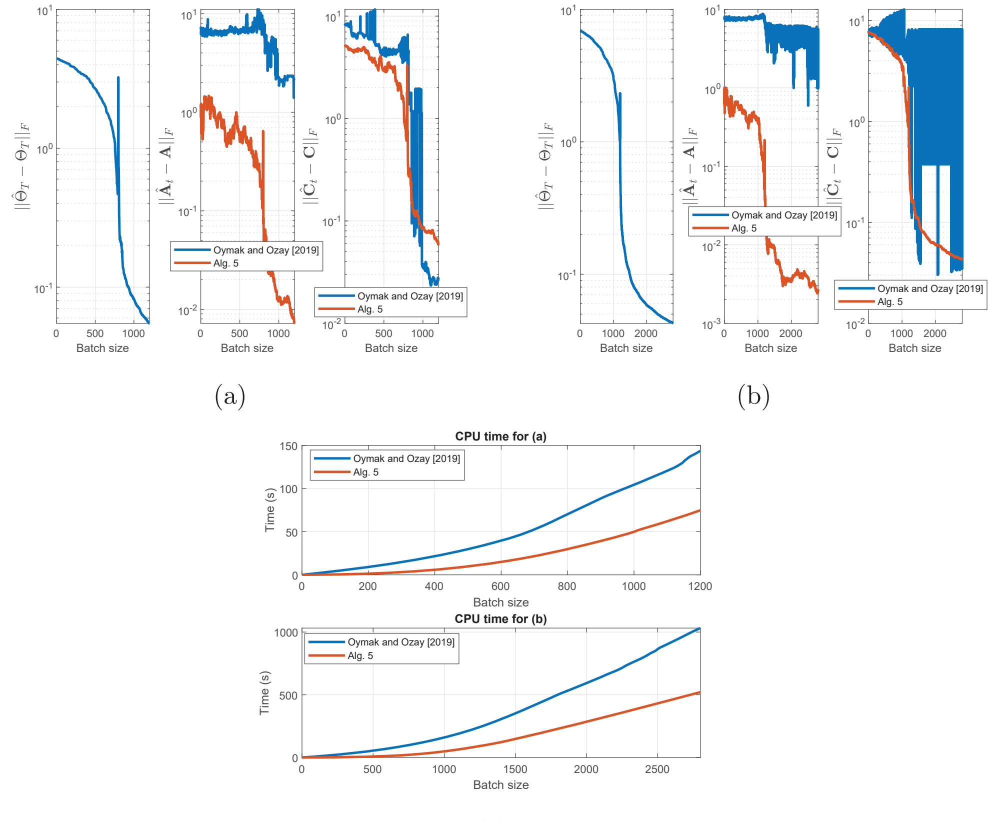
In this section, we evaluate the performance of the proposed approaches. The matrix $\mathbf{A}$ is randomly generated by choosing the conjugate pairs of roots of the characteristic polynomial inside a circle with a maximum radius of $\rho(\mathbf{A}) = 0.975$. Elements of the matrices $\mathbf{C}$ and $\mathbf{D}$ are independently drawn from a standard Gaussian distribution $\mathcal{N}(0, 1)$ for each experiment. The initial state of the system is zero. The performance measure in the experiments is the Frobenius norm distance between the estimated solution and the ground truth solution. We repeat each experiment 5 times and each curve corresponds to one independent realization. The spectral radius of $\mathbf{A}$ for SISO, multi-input single-output (MISO), and MIMO systems is 0.975, 0.70, and 0.64, respectively. In experiments, as $\rho(\mathbf{A})$ is close to one, transfer function has a heavy tail and a large T is required.

The convergence of Algorithm 25 for SISO, MISO, and MIMO systems is depicted in Figs. 7.1a-7.1c, where the measurement noise is zero. The hidden state dimension for considered systems is 30. The convergence of Algorithm 24 for identical systems is depicted in Figs. 7.1d-7.1f, when the batch size is $10,000$. In each iteration of Algorithm 24, one input-output pair is chosen uniformly at random, and the gradient is implemented based on that sample. It is observed that the number of iterations required by Algorithm 25 is fewer compared to Algorithm 24. The reason for this difference is that Algorithm 25 has access to a greater number of input-output pairs. The numerical tests confirm that the system identification error can be as small as desired via adjusting the learning rate and the truncation length.

We compare the performance of Algorithm 25 against [188]. The gradient projection algorithm in [188] implements gradient steps for $\mathbf{A}$ and $\mathbf{C}$ based on extracted information from a trajectory. After the gradients are implemented, the estimation of $\mathbf{A}$ is projected to a convex set. This set is characterized by $\Re(q(z)/z^n) > |\Im(q(z)/z^n)|$. The initialization of $\hat{\mathbf{A}}_0$ is critical for [188]. If the initial $\hat{\mathbf{A}}_0$ is unstable, the system blows up after one trajectory is fed to the system and the gradients cannot be computed. We compare the performance of the gradient projection algorithm in [188] against Algorithm 25 based on the number of input-output pairs that are fed to both approaches, where samples are discarded after the gradient implementations. In each iteration, the length of each trajectory fed to the gradient projection algorithm in [188] is 500. For the comparisons, we consider two SISO systems. In the first system, we have $m = 1$, $n = 20$, and $p = 1$. Moreover, for the second system, we set $m = 1$, $n = 30$, and $p = 1$. We observe that Algorithm 25 outperforms the gradient projection algorithm in [188] given an identical number of input-output pairs. The reason is that the gradients for $\mathbf{A}$ and $\mathbf{C}$ are extracted from a non-linear non-convex regression in [188] and also the gradient implementation in [188] requires a greater number of input-output samples compared to Algorithm 25.

The Ho-Kalman algorithm in [2] estimates the Hankel matrix $\mathbf{H}$, which is built using the Markov parameters of the system. Consider that the last $mT/2$ columns of the Hankel matrix are denoted by $\mathbf{H}^+$, where $T = 200$. The Ho-Kalman algorithm finds the rank-n-approximation of the Hankel matrix. Next, the rank-n-approximation, denoted by $\mathbf{L}$, is decomposed into the observability and controllability matrices. This decomposition is carried out using SVD. Therefore, if the rank-n-approximation of the Hankel matrix has an SVD decomposition like $\mathbf{L} = \mathbf{U\Sigma V}'$, the observability matrix is $\mathbf{O} = \mathbf{U\Sigma}^{1/2}$ and the controllability matrix is $\mathbf{Q} = \mathbf{\Sigma}^{1/2}\mathbf{V}'$. Then, the estimated $\hat{\mathbf{C}}$ matrix is the first p rows of the observability matrix. Furthermore, the estimated $\hat{\mathbf{A}}$ matrix is $(\hat{\mathbf{O}}'\hat{\mathbf{O}})^{-1}\hat{\mathbf{O}}\hat{\mathbf{H}}^+(\hat{\mathbf{Q}}'\hat{\mathbf{Q}})^{-1}\hat{\mathbf{Q}}$. We consider two MIMO systems for the comparisons. In the first system, the hidden state dimension is 20, $m = 4$, $n = 5$, and $p = 4$. In the second system, the hidden state dimension is 30, $m = 6$, $n = 5$, and $p = 6$. Our numerical simulations confirm that if the ground truth Hankel matrix is given to the Ho-Kalman algorithm, the estimated matrices are not identical to the ground truth weight matrices. To help the Ho-Kalman algorithm to find the underlying weight matrices, we give the optimal transformation $\mathcal{T}$ to the Ho-Kalman algorithm such that $\mathbf{U\Sigma}^{1/2}\mathcal{T}$ becomes the ground truth observability matrix, and $\mathcal{T}^{-1}\mathbf{\Sigma}^{1/2}\mathbf{V}'$ becomes the ground truth controllability matrix of the underlying system. Furthermore, we consider that the standard deviation of measurement noise is 0.1.

Since the Ho-Kalman Algorithm in [2] solves (7.11) by the pseudo-inverse method, we use Algorithm 27 for comparisons. We assume that input-output pairs arrive in an online streaming fashion and the batch size increases gradually. Both approaches share an identical estimation for the set of Markov parameters. From Figs. 7.2a and 7.2b, we observe that Algorithm 27 outperforms the Ho-Kalman approach in the estimation of $\mathbf{A}$. The reason is that Algorithm 27 directly extracts $\mathbf{A}$ from the Markov parameters. However,



(a)

(b)

(c)

Figure 7.2: Comparison of the performance of Algorithm 27 with [2]: (a) $m = 4$, $n = 5$, and $p = 4$; and (b) $m = 6$, $n = 5$, and $p = 6$; and (c) the CPU time.

the Ho-Kalman approach estimates $\mathbf{H}$, $\mathbf{O}$ and $\mathbf{Q}$ first and based on these matrices, $\mathbf{A}$ is recovered. Therefore, the errors of estimations for $\mathbf{O}$, $\mathbf{Q}$ and $\mathbf{H}$ are added to each other in the estimation of $\mathbf{A}$. Furthermore, we observe from Fig. 7.2c that the required CPU time by Algorithm 27 is significantly less than that by the Ho-Kalman algorithm as computing SVD is more costly compared to solving a linear system.

Next, we continue to evaluate the performance of our online and offline SGD algorithms on noisy and noisy-free linear dynamical systems when the system is SISO, single-input multi-output (SIMO), MISO, and MIMO. For each case, we consider three different hidden state dimensions and evaluate proposed algorithms in noisy and noise-free scenarios. The initial state of the system is zero.

7.7.1 SISO

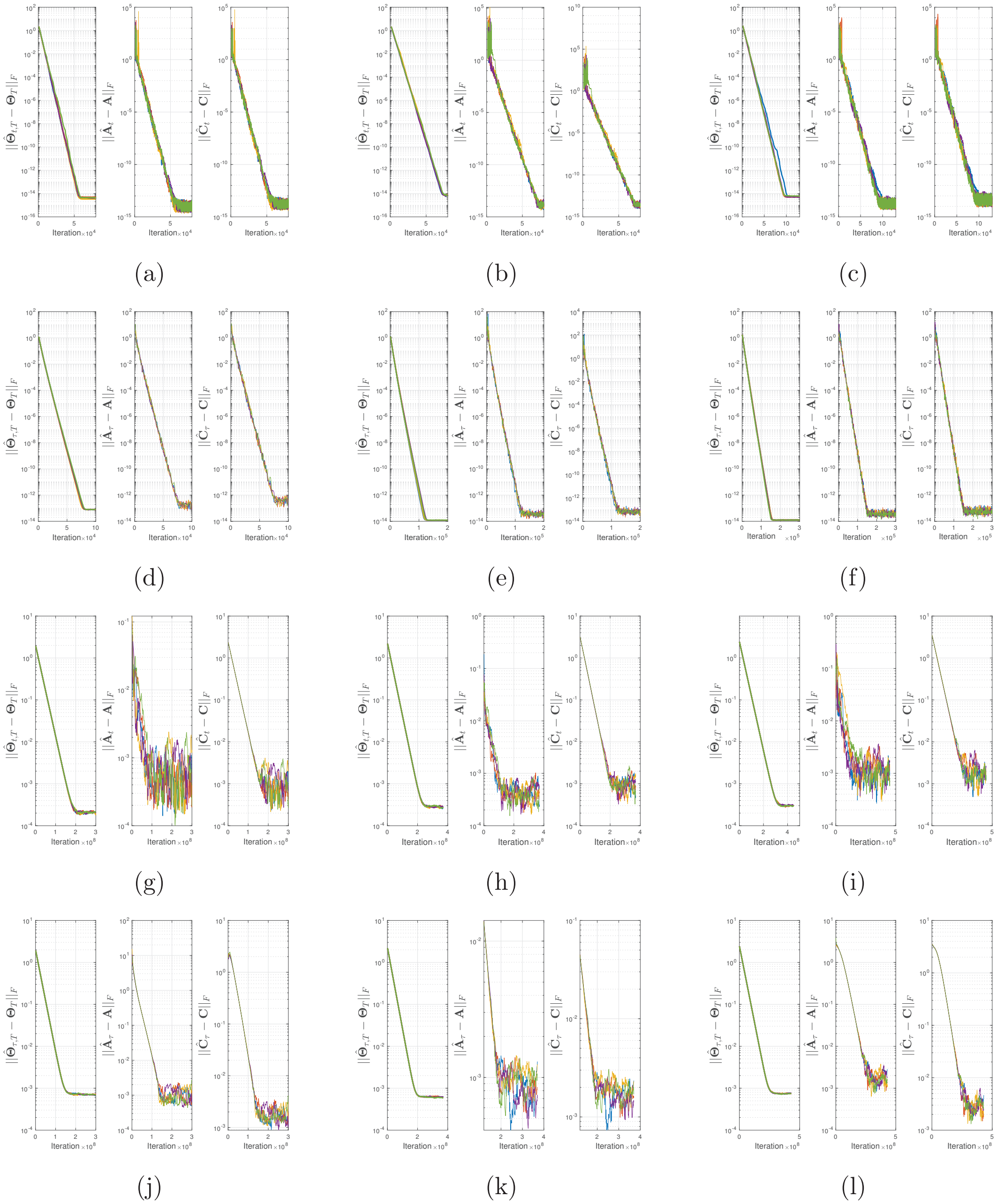
We consider three different hidden state dimensions, 20, 25 and 30 for the SISO system. It is observed from Figs. 7.3a, 7.3b and 7.3c that in all three cases Algorithm 25 learns the unknown parameters $\mathbf{\Theta}_T$, $\mathbf{A}$ and $\mathbf{C}$ at a linear convergence rate. The convergence of Algorithm 24 for identical systems is depicted in Figs. 7.3d-7.3f. Each iteration of either approach is implemented based on the obtained gradient from one input-output sample while the measurement noise is zero. We observe that when the hidden state dimension increases, the required iterations by both algorithms to reach a certain residual error increase. In Fig. 7.3a, we have $\rho(\mathbf{A}) = 0.93$, and for Figs. 7.3b and 7.3c we have $\rho(\mathbf{A}) = 0.975$. To tackle the heavy-tail issue of the transfer function, we increase T and also decrease the learning rate when the size of the hidden state increases. For Algorithms 24 and 25, we have $(T, \eta) = \{(800, 3 \times 10^{-4}), (1300, 3 \times 10^{-4}), (1600, 2 \times 10^{-4})\}$, when the $n = \{20, 25, 30\}$, respectively. For Algorithm 24, the batch size is $10,000$. The performance of Algorithms 25 and 24 for previously described systems is depicted in Figs. 7.3g-7.3i and 7.3j-7.3l, respectively, when the measurement noise follows a normal distribution with zero mean and standard deviation 0.1. The input to the system is Gaussian noise with zero mean and standard deviation 1. For both approaches, we set $(T, \eta) = \{(170, 5 \times 10^{-8}), (400, 4 \times 10^{-8}), (600, 3 \times 10^{-8})\}$ for the three considered systems. The batch size is 10^7 for Algorithm 24. In both noisy and noise-free systems, we observe that Algorithm 24 requires a greater number of iterations compared to Algorithm 25 to reach a certain residual error.

7.7.2 SIMO

We consider three different SIMO systems where $(n, p) = (20, 4)$, $(n, p) = (25, 5)$ and $(n, p) = (30, 6)$. We observe from Figs. 7.4a, 7.4b and 7.4c that when the measurement noise is zero, in all three cases Algorithm 25 reaches very close to the machine epsilon. For the above three dimensions, the necessary truncation length and the learning rate do not change significantly when the hidden state dimension and output size increase. The spectral radius of $\mathbf{A}$ in Figs. 7.4a, 7.4b and 7.4c is 0.93, 0.95 and 0.96, respectively. The convergence of Algorithm 24 for the three considered systems is depicted in Figs. 7.4d-7.4f, when the batch size is $10,000$. For both algorithms, we have $(T, \eta) = \{(800, 10^{-5}), (800, 10^{-5}), (800, 10^{-5})\}$. For the above systems, we consider measurement noise with zero mean and standard deviation 0.1. For both approaches, we set $(T, \eta) = \{(300, 4 \times 10^{-8}), (500, 3 \times 10^{-8}), (700, 3 \times 10^{-8})\}$. The batch size is 10^7 for Algorithm 24. The convergence of Algorithm 25 for noisy systems is depicted in Figs. 7.4g-7.4i, and the convergence of Algorithm 24 is depicted in Figs. 7.4j-7.4l. The input to the system is Gaussian noise with zero mean and standard deviation 1, and system noise has the standard deviation 0.1.

7.7.3 MISO

We consider three different MISO systems for which the hidden state dimensions are 20, 25 and 30. For these three systems, the input sizes are 4, 5 and 6, respectively. As depicted in Figs. 7.5a, 7.5b and 7.5c, Algorithm 25 learns the unknown parameters at a linear convergence rate. In Fig. 7.5a, we have $\rho(\mathbf{A}) = 0.75$ and for Figs. 7.5b and 7.5c, we have $\rho(\mathbf{A}) = 0.70$. In Figs. 7.5a-7.5c, we have $(T, \eta) = \{(800, 10^{-5}), (800, 10^{-5}), (800, 10^{-5})\}$. Identical truncation length and the learning rate are considered for Algorithm 24 in 7.5d-7.5f. The batch size for Algorithm 24 is $10,000$. For noisy systems, the measurement noise is white, and its mean is zero and its variance is 0.01. The standard deviation of the input signal is 0.1. When the measurement noise is considered for the above systems, the convergence of Algorithm 25 is depicted in Figs. 7.5g-7.5i. Moreover, Figs. 7.5j-7.5l show the convergence of Algorithm 24 for the three noisy systems. The truncation length and



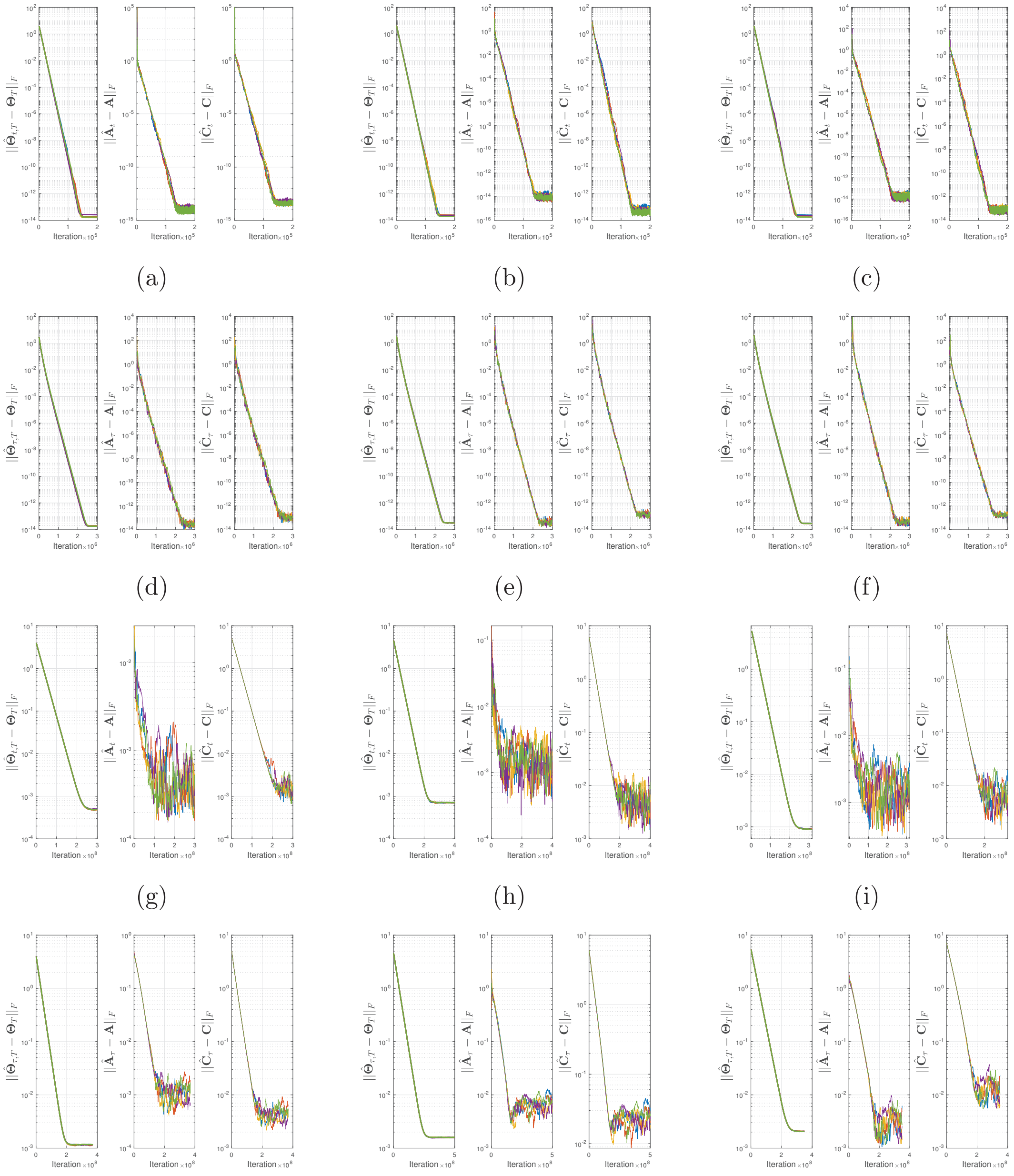
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 7.3: Results for SISO systems. In (a)-(f), the systems are noise-free. In (g)-(l), the systems are noisy. In (a), (d), (g) and (j), $n = 20$, $m = 1$, $p = 1$. In (b), (e), (h) and (k), $n = 25$, $m = 1$, $p = 1$. In (c), (f), (i) and (l), $n = 30$, $m = 1$, $p = 1$.



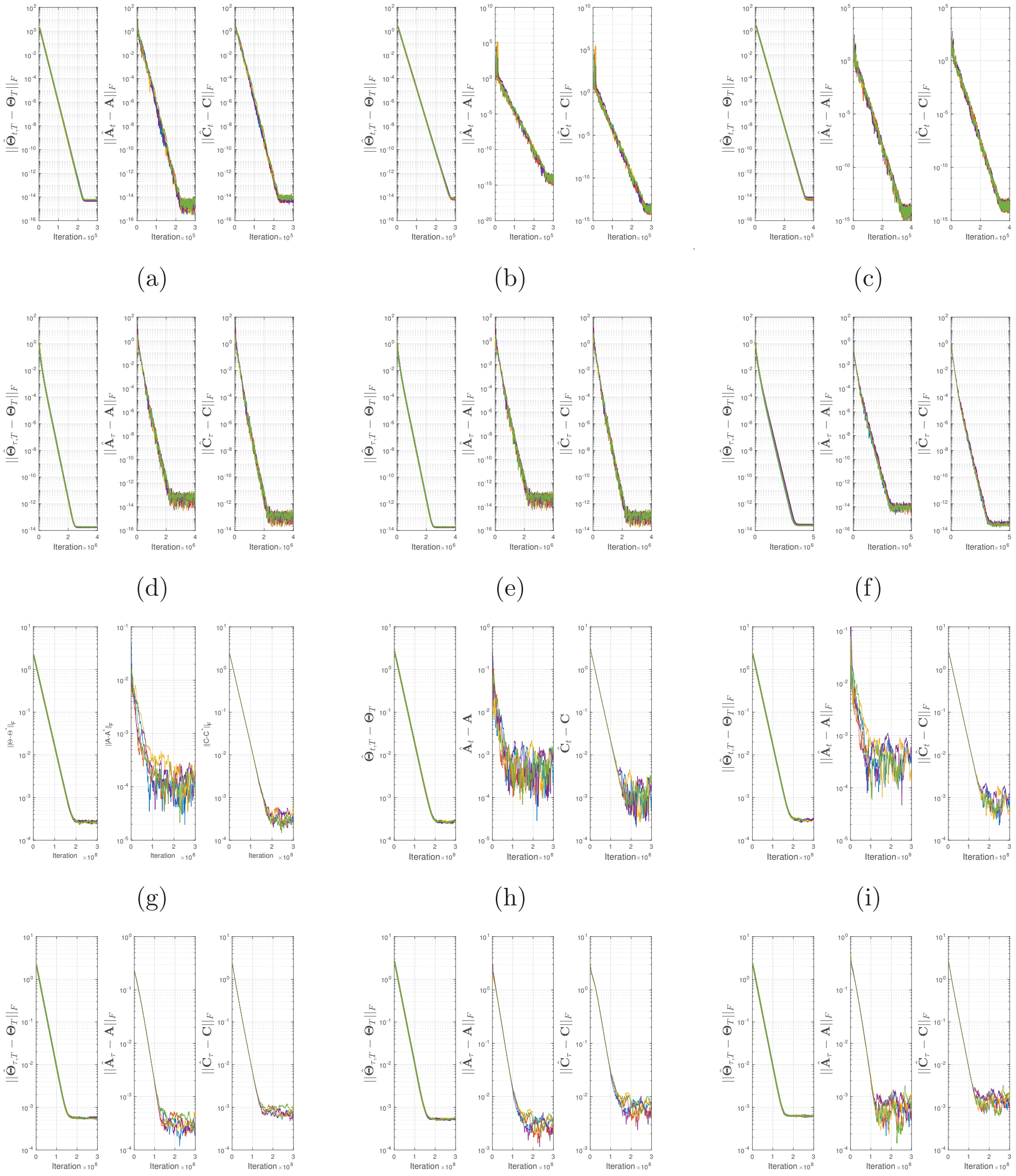
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 7.4: Results for SIMO systems. In (a)-(f), the systems are noise-free. In (g)-(l), the systems are noisy. In (a), (d), (g) and (j), $n = 20$, $m = 1$, $p = 4$. In (b), (e), (h) and (k), $n = 25$, $m = 1$, $p = 5$. In (c), (f), (i) and (l), $n = 30$, $m = 1$, $p = 6$.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 7.5: Results for MISO systems. In (a)-(f), the systems are noise-free. In (g)-(l), the systems are noisy. In (a), (d), (g) and (j), $n = 5$, $m = 4$, $p = 1$. In (b), (e), (h) and (k), $n = 5$, $m = 5$, $p = 1$. In (c), (f), (i) and (l), $n = 5$, $m = 6$, $p = 1$.

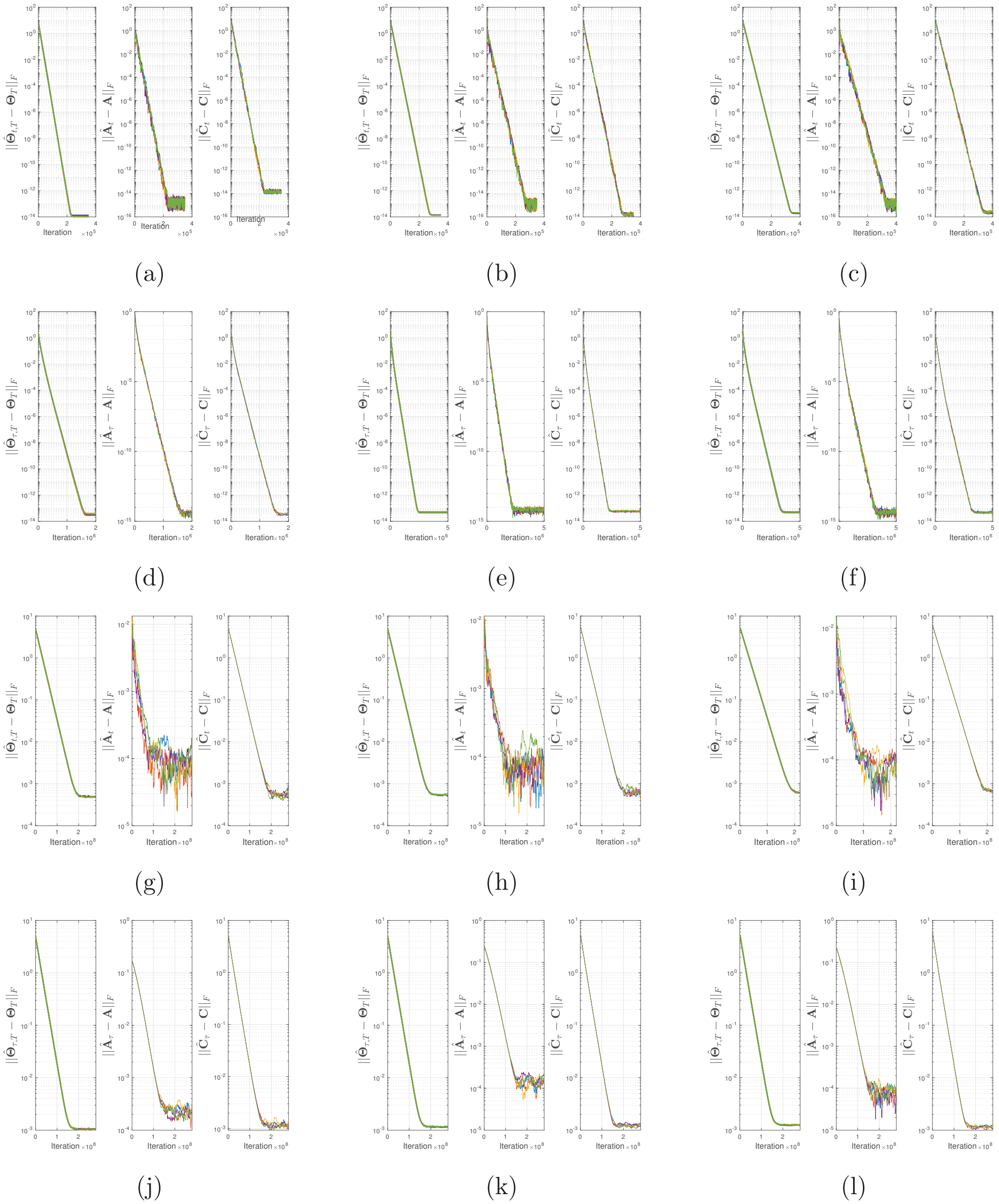
step-size for both algorithms are $(T, \eta) = \{(60, 5 \times 10^{-8}), (60, 5 \times 10^{-8}), (60, 5 \times 10^{-8})\}$ for different systems. The batch size for Algorithm 24 is 10^7.

7.7.4 MIMO

Three different MIMO systems with hidden state dimensions 20, 25 and 30 are considered. In particular, for the three systems, we have $(n, m, p) = (5, 4, 4)$, $(n, m, p) = (5, 5, 4)$ and $(n, m, p) = (5, 6, 4)$. As the hidden state dimension increases, the number of required samples to reach a certain distance from the ground truth unknowns increases. The spectral radius of $\mathbf{A}$ in Figs. 7.6a, 7.6b and 7.6c is 0.55, 0.75 and 0.64, respectively. The convergence of Algorithm 25 for three noise-free systems is depicted in Figs. 7.6a-7.6c, where $(T, \eta) = \{(800, 10^{-5}), (800, 10^{-5}), (800, 10^{-5})\}$. Figs. 7.6d-7.6f depict the convergence of Algorithm 24 with the same truncation lengths and learning rates. For Algorithm 24, the batch size is 10^4. The convergence of Algorithm 25 for the three systems after the addition of white noise with standard deviation 0.1 is given Figs. 7.6g-7.6i. Furthermore, the convergence of Algorithm 24 for identical systems is depicted Figs. 7.6j-7.6l. For both approaches, we have $(T, \eta) = \{(60, 5 \times 10^{-8}), (60, 5 \times 10^{-8}), (60, 5 \times 10^{-8})\}$. The batch size is 10^7. The control signal is white noise with zero mean and standard deviation 1.

7.8 Concluding Remarks

This paper presents a novel approach to learn unknown transformation matrices of a certain class of stable linear dynamical systems from a single, noisy sequence of input-output pairs. We proposed online and offline SGD algorithms, proved that they efficiently learn the Markov parameters of the system at a linear convergence rate, and provide novel complexity bounds. When the unknown transformation matrices of the system have Brunovsky canonical form, we draw connections between Markov parameters and unknown transformation matrices using the transfer function of the system. We proved that the linear convergence of the Markov parameters enforces a linear convergence rate for unknown matrices to converge to their ground truth weights. We demonstrated the performance of our methods against state of the art methods through numerical simulations. It would be interesting to see whether our proposed approaches could be extended to the identification of periodic and Markov jump linear systems as well, as such systems are structurally more similar to multi-layer perceptron-type neural architectures.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 7.6: Results for MIMO systems. In (a)-(f), the systems are noise-free. In (g)-(l), the systems are noisy. In (a), (d), (g) and (j), $n = 5$, $m = 4$, $p = 4$. In (b), (e), (h) and (k), $n = 5$, $m = 5$, $p = 4$. In (c), (f), (i) and (l), $n = 5$, $m = 6$, $p = 4$.

Chapter 8

Future Directions

Here, we briefly outline some of the possible future directions of this research:

1. **Joint software-defined resource allocation and RAN optimization:** Although the software-defined resource allocation in service-oriented networks is well-studied, the problem of joint flow processing and wirelessly transmitting processed flows to users is not studied in the literature. The joint optimization of flow processing and wireless transmission of data flows enables the end-to-end resource allocation, which prevents the resources in the service-oriented network and RAN from negatively affecting each other. Moreover, it ensures that proportional resources are allocated in both networks, which result in boosting the reliablity in serving users. The joint optimization has proven to be an efficient tool to enhance the quality of service for users.

2. **Uplink and downlink wireless flows in wireless networks:** Uplink and downlink information flows occur simultaneously in wireless networks. One interesting problem is to consider the resource allocation in RAN for both downlink transmission and uplink reception from users. For this practical problem, in addition to downlink channels, uplink channels should also be considered in the model. Similar to the user downlink rates, the expectation of user uplink rates can be calculated. Once the resource consumption rule in access points for signal reception is identified, we can extend our formulations to address the joint problem of wireless reception and

transmission in access points.

3. **Dynamic slice creation and management**: One of the major goals of operators in 5G and beyond networks is to allocate network resources so as to maximize their revenues. In order to be able to provide the maximum number of services aiming at maximizing revenues, virtual network functions and network resources should be rapidly configured to build network slices. Efficient network slicing protocols are required for dynamic creation and scaling of slices based on the time-varying service load. Furthermore, network slices are typically managed by operators; however, some slices may still need to deploy self-controlled. For these scenarios, the management of slices needs to be implemented in a distributed and automated fashion. Distributed methods with low overhead are of interest to address the above problems.

4. **Share-based network slicing:** One of the key components of network slicing is resource allocation among slices. Network slicing addresses how to assign the underlying infrastructure resources to each slice at each point in time. Two different types of approaches have appeared in the literature to implement resource allocations. The first type is reservation-based and the second type is share-based. In this dissertation, we studied the reservation-based network slicing. In share-based resource allocation, a network tenant does not make reservation requests for different type of resources. However, it acquires a share of the end-to-end network. Such share is later dynamically mapped to different allocations of several resources depending on tenants' needs at each time. Share-based network slicing is an interesting research direction, since it can more rapidly adapt resource allocations to time-varying demands of network slices.

5. **Magnetic resonance imaging with large field inhomogeneity:** Our work, of course, has its limitations. We have assumed images in the first dataset are taken to be at the same physical location within the magnet, and are affected by the same distribution of off-resonance frequencies. This means that we identify the field map on a given *physical* slice; of course, our method could be repeated at each physical slice to estimate field maps for a prescribed volume. Moreover, the number of dataset images used in the proposed approach must be greater than or equal to the number

of image voxels. Therefore, NYU DICOM that includes a sufficient number of images was used to implement the proposed method.

It was assumed in our work that images in both datasets have the same size. To reduce the acquisition time and enhance the SNR level of images in the second dataset, it might be possible to lower the resolution of images in the second dataset. However, this may make the optimization more complicated compared to the model we consider here. Moreover, we assume here that there is a single off-resonance frequency describing each voxel, while in reality the spatial distribution of off-resonance frequencies may not be uniform in each voxel. Increasing the resolution of the images, or adopting a more general field map model, could each be useful in addressing such issues. Finally, reducing the amount of data required for this type of inference would, of course, make it more amenable to practice.

6. **Learning linear time-variant systems:** Linear, time-variant dynamical systems are attracting attention due to their ability to model a wide range of dynamical systems and their resemblance to non-linear systems. Among different time-variant systems, periodic and Markov jump systems have attracted more interest due to their wide applications. In each of these two systems, there are a finite number of modes, which are visited by the system based on a certain rule. If the mode of the system is repeated deterministically and regularly, the system is periodic. The behavior of linear dynamical systems is periodic in various applications, e.g., power generation, multi-mode energy consumption, financial time-series, and maneuvering targets. Even though the analysis and control of periodic systems is well-studied, the identification of periodic linear systems is still a challenging problem due to the variation of weight matrices with time.

   Unlike periodic systems in which the system mode changes deterministically, in Markov jump systems, the mode of the system is randomized from a finite set of values, based on the law of a Markov chain. The mode switches of linear dynamical systems occur for many reasons, such as environmental disturbances, component failures, or changes in subsystem interconnections. Applications of Markov jump linear systems include, but are not limited to, power generation in central receiver solar thermal power plants, robotic manipulators, economics, and aircraft control. Due to the randomness of the mode of Markov jump systems, their identification is harder than that for the periodic systems, where the mode of the system is deterministic.

We will study the identification of periodic and Markov jump linear systems based on time-domain data under three assumptions: a) the mode transitions of systems are observable; b) the order of the system is known; and c) weight matrices have Brunovsky canonical forms. We collect the input-output pairs to form finite sum problems and subsequently estimate the Markov parameters for each mode. Since the required number of input-output pairs to reach a desired accuracy can be large, we propose storage-efficient SGD algorithms to solve finite sum problems. When a mode persists for $2n$ time instances, where n is the order of the system, we prove that the z- transformation of the first $2n$ Markov parameters of each mode can be written as a subtraction of the transfer functions of two LTI dynamical systems. We use this fact to formulate a novel linear system of equations, which is proven to uniquely identify the ground-truth unknown weight matrices for the underlying mode.

Bibliography

[1] C. Y. Lin and J. A. Fessler. Efficient regularized field map estimation in 3D MRI. *IEEE Trans. Comput. Imag.*, 6:1451–1458, Oct. 2020.

[2] S. Oymak and N. Ozay. Non-asymptotic identification of LTI systems from a single trajectory. In *Proc. Amer. Control Conf. (ACC)*, pages 5655–5661, 2019.

[3] W.-C. Liao, M. Hong, H. Farmanbar, and Z.-Q. Luo. A distributed semiasynchronous algorithm for network traffic engineering. *IEEE Trans. Signal Inf. Process. Netw.*, 4(3):436–450, Sep. 2018.

[4] W.-C. Liao, M. Hong, H. Farmanbar, X. Li, Z.-Q. Luo, and H. Zhang. Min flow rate maximization for software defined radio access networks. *IEEE J. Sel. Areas Commun.*, 32(6):1282–1294, Jun. 2014.

[5] W.-C. Liao, M. Hong, Y.-F. Liu, and Z.-Q. Luo. Base station activation and linear transceiver design for optimal resource management in heterogeneous networks. *IEEE Trans. Signal Process.*, 62(15):3939–3952, Jul. 2014.

[6] Z. Wu, Z. Fei Y. Zesong, and Z. Han. Toward optimal remote radio head activation, user association, and power allocation in C-RANs using benders decomposition and ADMM. *IEEE Trans. Commun.*, 67(7):5008–5023, Jul. 2019.

[7] N. Moehle, X. Shen, Z.-Q. Luo, and S. Boyd. A distributed method for optimal capacity reservation. *Journal of Optimization Theory and Applications*, 182(3):1130–1149, May 2019.

[8] D. Ma, B. Sheng, S. Jin, X. Ma, and P. Gao. Short-term traffic flow forecasting by selecting appropriate predictions based on pattern matching. *IEEE Access*, 6:75629–75638, Nov. 2018.

[9] M. Yan, G. Feng, J. Zhou, Y. Sun, and Y.-C. Liang. Intelligent resource scheduling for 5G radio access network slicing. *IEEE Trans. Veh. Technol.*, 68(8):7691–7703, Jun. 2019.

[10] Q. He, A. Moayyedi, G. Dán, G. P. Koudouridis, and P. Tengkvist. A meta-learning scheme for adaptive short-term network traffic prediction. *IEEE J. Sel. Areas Commun.*, Jun. 2020.

[11] L. U. Khan, I. Yaqoob, N.-H. Tran, Z. Han, and C.-S. Hong. Network slicing: Recent advances, taxonomy, requirements, and open research challenges. *IEEE Access*, 8:36009–36028, Feb. 2020.

[12] Y. Li, J. Liu, B. Cao, and C. Wang. Joint optimization of radio and virtual machine resources with uncertain user demands in mobile cloud computing. *IEEE Trans. Multimedia*, 20(9):2427–2438, Sep. 2018.

[13] P. Kemmer, A. Strauss, and T. Winter. Dynamic simultaneous fare proration for large-scale network revenue management. *Journal of the Operational Research Society*, 63(10):1336–1350, 2012.

[14] D. Applegate and E. Cohen. Making routing robust to changing traffic demands: algorithms and evaluation. *IEEE/ACM Trans. Netw.*, 14(6):1193–1206, Dec. 2006.

[15] M. Mullen and M. Garwood. Contemporary approaches to high-field magnetic resonance imaging with large field inhomogeneity. *Prog. Nucl. Magn. Reson. Spectrosc.*, 120:95–108, 2020.

[16] Y. Lim, Y. Bliesener, S. Narayanan, and K. S. Nayak. Deblurring for spiral real-time MRI using convolutional neural networks. *Magn. Reson. Med.*, 84(6):3438–3452, 2020.

[17] B. P. Sutton, D. C. Noll, and J. A. Fessler. Fast, iterative image reconstruction for MRI in the presence of field inhomogeneities. *IEEE Trans. Med. Imag.*, 22(2):178–188, Feb. 2003.

[18] A. Maeda, K. Sano, and T. Yokoyama. Reconstruction by weighted correlation for MRI with time-varying gradients. *IEEE Trans. Med. Imag.*, 7(1):26–31, Mar. 1988.

[19] M. Razaviyayn, M. Hong, and Z.-Q. Luo. A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM Journal on Optimization*, 23(2):1126–1153, 2013.

[20] NGMN Alliance. Description of network slicing concept. *NGMN 5G P*, 1, 2016.

[21] L. Liang, Y. Wu, G. Feng, X. Jian, and Y. Jia. Online auction-based resource allocation for service-oriented network slicing. *IEEE Trans. Veh. Technol.*, 68(8):8063–8074, Aug. 2019.

[22] D. Drutskoy, E. Keller, and J. Rexford. Scalable network virtualization in software-defined networks. *IEEE Internet Comput.*, 17(2):20–27, Apr. 2012.

[23] S. Q. Zhang, Q. Zhang, H. Bannazadeh, and A. Leon-Garcia. Routing algorithms for network function virtualization enabled multicast topology on SDN. *IEEE Transactions on Network and Service Management*, 12(4):580–594, Dec. 2015.

[24] A. Basta, A. Blenk, K. Hoffmann, H. J. Morper, M. Hoffmann, and W. Kellerer. Towards a cost optimal design for a 5G mobile core network based on SDN and NFV. *IEEE Trans. Netw. Service Manag.*, 14(4):1061–1075, Dec. 2017.

[25] C. Pham, N. Tran, S. Ren, W. Saad, and C. Hong. Traffic-aware and energy-efficient vNF placement for service chaining: Joint sampling and matching approach. *IEEE Trans. on Services Comput.*, Feb. 2017.

[26] H. Hawilo, M. Jammal, and A. Shami. Network function virtualization-aware orchestrator for service function chaining placement in the cloud. *IEEE J. Sel. Areas Commun.*, Mar. 2019.

[27] Y. Zhang, N. Beheshti, L. Beliveau, G. Lefebvre, R. Manghirmalani, R. Mishra, R. Patneyt, M. Shirazipour, R. Subrahmaniam, C. Truchan, and M. Tatipamula. StEERING: A software-defined networking for inline service chaining. In *Proc. IEEE ICNP*, pages 1–10, Oct. 2013.

[28] D. Cotroneo, L. De Simone, and R. Natella. NFV-Bench: A dependability benchmark for network function virtualization systems. *IEEE Trans. Netw. Service Manag.*, 14(4):934–948, Dec. 2017.

[29] C. Sun, J. Bi, Z. Zheng, and H. Hu. HYPER: A hybrid high-performance framework for network function virtualization. *IEEE J. Sel. Areas Commun.*, 35(11):2490–2500, Nov. 2017.

[30] V. Eramo, E. Miucci, M. Ammar, and F. G. Lavacca. An approach for service function chain routing and virtual function network instance migration in network function virtualization architectures. *IEEE/ACM Trans. Netw.*, 25(4):2008–2025, Aug. 2017.

[31] J. Liu, W. Lu, F. Zhou, P. Lu, and Z. Zhu. On dynamic service function chain deployment and readjustment. *IEEE Trans. Netw. Service Manag.*, 14(3):543–553, Sept. 2017.

[32] N. Zhang, Y. F. Liu, H. Farmanbar, T. H. Chang, M. Hong, and Z.-Q. Luo. Network slicing for service-oriented networks under resource constraints. *IEEE J. Sel. Areas Commun.*, 35(11):2512–2521, Nov. 2017.

[33] B. Kar, E. H. K. Wu, and Y. D. Lin. Energy cost optimization in dynamic placement of virtualized network function chains. *IEEE Trans. Netw. Service Manag.*, 15(1):372–386, Mar. 2018.

[34] I. Afolabi, J. Prados, M. Bagaa, T. Taleb, and P. Ameigeiras. Dynamic resource provisioning of a scalable E2E network slicing orchestration system. *IEEE Trans. Mobile Comput.*, pages 1–1, Jul. 2019.

[35] M. Jia, W. Liang, M. Huang, Z. Xu, and Y. Ma. Routing cost minimization and throughput maximization of NFV-enabled unicasting in software-defined networks. *IEEE Trans. Netw. Service Manag.*, PP(99):1–1, Mar. 2018.

[36] M. Huang, W. Liang, Z. Xu, and S. Guo. Efficient algorithms for throughput maximization in software-defined networks with consolidated middleboxes. *IEEE Trans. Netw. Service Manag.*, 14(3):631–645, Sept. 2017.

[37] H. Cao, Y. Zhu, G. Zheng, and L. Yang. A novel optimal mapping algorithm with less computational complexity for virtual network embedding. *IEEE Trans. Netw. Service Manag.*, 15(1):356–371, Mar. 2018.

[38] O. Soualah, N. Aitsaadi, and I. Fajjari. A novel reactive survivable virtual network embedding scheme based on game theory. *IEEE Trans. Netw. Service Manag.*, 14(3):569–585, Sept. 2017.

[39] M. Ghaznavi, N. Shahriar, R. Ahmed, and R. Boutaba. Distributed service function chaining. *IEEE J. Sel. Areas Commun.*, 35(11):2479–2489, Nov. Oct. 2017.

[40] A. Gupta, B. Jaumard, M. Tornatore, and B. Mukherjee. A scalable approach for service chain mapping with multiple SC instances in a wide-area network. *IEEE J. Sel. Areas Commun.*, 36(3):529–541, Mar. 2018.

[41] M. Otokura, K. Leibnitz, Y. Koizumi, D. Kominami, T. Shimokawa, and M. Murata. Evolvable virtual network function placement method: Mechanism and performance evaluation. *IIEEE Trans. Netw. Service Manag.*, 16(1):27–40, Mar. 2019.

[42] N. Zhao, X. Zhang, F. Yu, and V. Leung. To align or not to align: Topology management in asymmetric interference networks. *IEEE Trans. Veh. Technol.*, 66(8):7164–7177, Aug. 2017.

[43] Y. Zhang and T. Zhou. Controllability analysis for a networked dynamic system with autonomous subsystems. *IEEE Trans. Autom. Control*, 62(7):3408–3415, Jul. 2017.

[44] T. Mano, T. Inoue, D. Ikarashi, K. Hamada, K. Mizutani, and O. Akashi. Efficient virtual network optimization across multiple domains without revealing private information. *IEEE Trans. Netw. Service Manag.*, 13(3):477–488, Sept. 2016.

[45] G. Sun, Y. Li, D. Liao, and V. Chang. Service function chain orchestration tiple domains: A full mesh aggregation approach. *IEEE Trans. Netw. Service Manag.*, 15(3):1175–1191, Jul. 2018.

[46] J. Halpern and C. Pignataro. Service function chaining (SFC) architecture. Technical report, 2015.

[47] H. K. Nguyen, Y. Zhang, Z. Chang, and Z. Han. Parallel and distributed resource allocation with minimum traffic disruption for network virtualization. *IEEE Trans. Commun.*, 65(3):1162–1175, Mar. 2017.

[48] D. Acemoglu and A. Ozdaglar. Competition in parallel-serial networks. *IEEE J. Sel. Areas Commun.*, 25(6):1180–1192, Aug. 2007.

[49] L. Kleinrock. *Communication nets: Stochastic message flow and delay.* Courier Corporation, 2007.

[50] P. Liu, Y.-F. Liu, and J. Li. An iterative reweighted minimization framework for joint channel and power allocation in the OFDMA system. In *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, pages 3068–3072, Apr. 2015.

[51] B. Jiang, Y.-F. Liu, and Z. Wen. l_p-norm regularization algorithms for optimization over permutation matrices. *SIAM Journal on Optimization*, 26(4):2284–2313, 2016.

[52] C. Beckman and G. Smith. Shared networks: making wireless communications affordable. *IEEE wireless communications*, 12(2):78–85, Apr. 2005.

[53] J. Markendahl, A. Ghanbari, and B. G. Mölleryd. Network cooperation between mobile operators-why and how competitors cooperate. In *Proc. IEEE IMP*, 2013.

[54] D.P. Bertsekas. *Nonlinear Programming.* Athena Scientific, third edition, 2016.

[55] M. Hong, M. Razaviyayn, Z. Luo, and J. Pang. A unified algorithmic framework for block-structured optimization involving big data: With applications in machine learning and signal processing. *IEEE Signal Processing Magazine*, 33(1):57–77, Jan. 2016.

[56] R. Zhao, J. Wang, K. Lu, X. Chang, J. Jia, and S. Zhang. Optimal transmission topology construction and secure linear network coding design for virtual-source multicast with integral link rates. *IEEE Trans. Multimedia*, 20(11):3069–3083, Nov. 2018.

[57] A. Craig, B. Nandy, I. Lambadaris, and P. Ashwood-Smith. Load balancing for multicast traffic in sdn using real-time link cost modification. In *Proc. IEEE Int. Conf. Commun. (ICC)*, pages 5789–5795, Jun. 2015.

[58] T. Cui and T. Ho. Minimum cost integral network coding. In *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, pages 2736–2740, Jun. 2007.

[59] L. Xiao, M. Johansson, and S. Boyd. Simultaneous routing and resource allocation via dual decomposition. *IEEE Trans. Commun.*, 52(7):1136–1144, Jul. 2004.

[60] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[61] M. Charikar, Y. Naamad, J. Rexford, and K. Zou. Multi-commodity flow with in-network processing. *Available Online: , www.cs.princeton.edu/~jrex/papers/mopt14.pdf*, 2014.

[62] E. Candes, M. Wakin, and S. Boyd. Enhancing sparsity by reweighted ℓ_1 minimization. *Journal of Fourier analysis and applications*, 14(5-6):877–905, 2008.

[63] S. Albasheir and M. Kadoch. Enhanced control for adaptive resource reservation of guaranteed services in LTE networks. *IEEE Internet Things J.*, 3(2):179–189, Apr. 2015.

[64] K. Kaur, A. Dua, A. Jindal, N. Kumar, M. Singh, and A. Vinel. A novel resource reservation scheme for mobile PHEVs in V2G environment using game theoretical approach. *IEEE Trans. Veh. Technol.*, 64(12):5653–5666, Dec. 2015.

[65] E. Van Den Berg, T. Zhang, J. Chennikara, P. Agrawal, and T. Kodama. Time series-based localized predictive resource reservation for handoff in multimedia wireless networks. In *Proc. IEEE Int. Conf. Commun.*, volume 2, pages 346–350, 2001.

[66] N. Reyhanian and B. Maham. Statistical slice selection in multi-tenant networks with maximum isolation of reserved resources. In *Proc. 54th Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, Nov. 2020.

[67] N. Reyhanian, H. Farmanbar, and Z.-Q. Luo. Data-driven adaptive network resource slicing for multi-tenant networks. In *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, pages 4715–4719, Jun. 2021.

[68] A. Al-Khatib and A. Khelil. Priority- and reservation-based slicing for future vehicular networks. In *Proc. IEEE Conf. Netw. Soft. (NetSoft)*, pages 36–42, Jun. 2020.

[69] M. Alsenwi, N.-H. Tran, M. Bennis, S. R. Pandey, A. K. Bairagi, and C. C. Hong. Intelligent resource slicing for eMBB and URLLC coexistence in 5G and beyond: A deep reinforcement learning based approach. *IEEE Trans. Wireless Commun.*, 20(7):4585–4600, Jul. 2021.

[70] N. Reyhanian and Z.-Q. Luo. Data-driven optimized slice activation in multi-tenant 5G networks. In *Proc. 55th Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, Nov. 2021.

[71] N. Reyhanian and Z.-Q. Luo. Data-driven adaptive network slicing for multi-tenant networks. *IEEE J. Sel. Topics Signal Process.*, 16(1):113–128, Jan. 2022.

[72] A. Mesodiakaki, E. Zola, and A. Kassler. Joint user association and backhaul routing for green 5G mesh millimeter wave backhaul networks. In *Proc. ACM Int. Conf. Modelling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, pages 179–186, Nov. 2017.

[73] S. Matoussi, I. Fajjari, S. Costanzo, N. Aitsaadi, and R. Langar. 5G RAN: Functional split orchestration optimization. *IEEE J. Sel. Areas Commun.*, 38(7):1448–1463, Jul. 2020.

[74] L.-D. Nguyen, H.-D. Tuan, T.-Q. Duong, V. Poor, and L. Hanzo. Energy-efficient multi-cell massive MIMO subject to minimum user-rate constraints. *IEEE Trans. Commun.*, 69(2):914–928, Feb. 2020.

[75] S. Matoussi, I. Fajjari, S. Costanzo, N. Aitsaadi, and R. Langar. 5G RAN: Functional split orchestration optimization. *IEEE J. Sel. Areas Commun.*, 38(7):1448–1463, Jul. 2020.

[76] Y. Zhang, J. Liu, M. Sheng, Y. Shi, and J. Li. Leveraging the coupling of radio access network and mmwave backhaul network: Modeling and optimization. *IEEE Trans. Veh. Technol.*, May. 2021.

[77] N. Reyhanian, V. Shah-Mansouri, B. Maham, and C. Yuen. Renewable energy distribution in cooperative cellular networks with energy harvesting. In *IEEE 26th Annu. Int. Symp. Pers. Indoor Mobile Radio Commun (PIMRC)*, pages 1617–1621. IEEE, Aug. 2015.

[78] N. Reyhanian, B. Maham, V. Shah-Mansouri, W. Tushar, and C. Yuen. Game-theoretic approaches for energy cooperation in energy harvesting small cell networks. *IEEE Trans. Veh. Technol.*, 66(8):7178–7194, Jan. 2017.

[79] N. Reyhanian, B. Maham, V. Shah-Mansouri, and C. Yuen. A matching-game-based energy trading for small cell networks with energy harvesting. In *IEEE 26th Annu. Int. Symp. Pers. Indoor Mobile Radio Commun (PIMRC)*, pages 1579–1583. IEEE, Aug. 2015.

[80] R. Gomory and T. Hu. An application of generalized linear programming to network flows. *Journal of the Society for Industrial and Applied Mathematics*, 10(2):260–283, 1962.

[81] R. Dai, L. Li, S. Wang, and X. Zhang. Planning traffic-oblivious survivable WDM networks using differentiated reliable partial SRLG-disjoint protection. In *Proc. Symposium on Photonics and Optoelectronics*, pages 1–7, Jun. 2010.

[82] P. Kumar, Y. Yuan, C. Yu, N. Foster, R. Kleinberg, and R. Soulé. Kulfi: Robust traffic engineering using semi-oblivious routing. *arXiv preprint arXiv:1603.01203*, 2016.

[83] C. Cicconetti, V. Gardellin, L. Lenzini, E. Mingozzi, and A. Erta. End-to-end bandwidth reservation in IEEE 802.16 mesh networks. In *Proc. IEEE Int. Conf. Mobile Adhoc and Sensor Syst.*, pages 1–6, Oct. 2007.

[84] J. Prados-Garzon, A. Laghrissi, M. Bagaa, T. Taleb, and J. Lopez-Soler. A complete LTE mathematical framework for the network slice planning of the EPC. *IEEE Trans. Mobile Comput.*, 19(1):1–14, Jan. 2020.

[85] T. Hößler, P. Schulz, E. A. Jorswieck, M. Simsek, and G. P. Fettweis. Stable matching for wireless URLLC in multi-cellular, multi-user systems. *IEEE Trans. Commun.*, 68(8):5228–5241, Aug. 2020.

[86] D.P. Bertsekas. *Linear network optimization: algorithms and codes.* MIT press, 1991.

[87] J. Tsitsiklis and D. Bertsekas. Distributed asynchronous optimal routing in data networks. *IEEE Trans. Autom. Control*, 31(4):325–332, Apr. 1986.

[88] D.P. Bertsekas. *Network optimization: continuous and discrete models.* Athena Scientific Belmont, MA, 1998.

[89] H. Zhang and V. W. S. Wong. A two-timescale approach for network slicing in C-RAN. *IEEE Trans. Veh. Technol.*, 69(6):6656–6669, Jun. 2020.

[90] L. Li and A. Goldsmith. Capacity and optimal resource allocation for fading broadcast channels. II. outage capacity. *IEEE Trans. Inf. Theory*, 47(3):1103–1127, Mar. 2001.

[91] X. Liao, J. Shi, Z. Li, L. Zhang, and B. Xia. A model-driven deep reinforcement learning heuristic algorithm for resource allocation in ultra-dense cellular networks. *IEEE Trans. Veh. Technol.*, 69(1):983–997, Jan. 2019.

[92] V. Sciancalepore, X. Costa-Perez, and A. Banchs. RL-NSB: Reinforcement learning-based 5G network slice broker. *IEEE/ACM Trans. Netw.*, 27(4):1543–1557, Aug. 2019.

[93] Y. Liang and V. Veeravalli. Gaussian orthogonal relay channels: Optimal resource allocation and capacity. *IEEE Trans. Inf. Theory*, 51(9):3284–3289, Sep. 2005.

[94] S. Kim and G. Giannakis. Optimal resource allocation for MIMO ad–hoc cognitive radio networks. *IEEE Trans. Inf. Theory*, 57(5):3117–3131, May 2011.

[95] A. El-Sherif and A. Mohamed. Joint routing and resource allocation for delay minimization in cognitive radio based mesh networks. *IEEE Trans. Wireless Commun.*, 13(1):186–197, Jan. 2014.

[96] K. Wang, K. Yang, and C. Magurawalage. Joint energy minimization and resource allocation in C-RAN with mobile cloud. *IEEE Trans. on Cloud Comput.*, 6(3):760–770, Sep. 2018.

[97] J. Liu, Y. Pang, H. Ding, L. Cai, H. Zhang, and Y. Fang. Optimizing IoT energy efficiency on edge (EEE): a cross-layer design in a cognitive mesh network. *arXiv preprint arXiv:1901.05494*, 2019.

[98] H. Kordbacheh, H. Oskouei, and N. Mokari. Robust cross-layer routing and radio resource allocation in massive multiple antenna and OFDMA-based wireless Ad-Hoc networks. *IEEE Access*, 7:36527–36539, Mar. 2019.

[99] K. Karakayali, J. Kang, M. Kodialam, and K. Balachandran. Joint resource allocation and routing for OFDMA-based broadband wireless mesh networks. In *Proc. IEEE Int. Conf. Commun. (ICC)*, pages 5088–5092, Jun. 2007.

[100] S. Choudhury and J. D. Gibson. Information transmission over fading channels. In *Proc. IEEE Global Commun. Conf.*, pages 3316–3321, Nov. 2007.

[101] D. Wu and R. Negi. Effective capacity: a wireless link model for support of quality of service. *IEEE Trans. Wireless Commun.*, 2(4):630–643, Jul. 2003.

[102] O. Ertug. Asymptotic ergodic capacity of multidimensional vector-sensor array MIMO channels. *IEEE Trans. Wireless Commun.*, 7(9):3297–3300, Sep. 2008.

[103] H. K. Nguyen, Y. Zhang, Z. Chang, and Z. Han. Parallel and distributed resource allocation with minimum traffic disruption for network virtualization. *IEEE Trans. Commun.*, 65(3):1162–1175, Mar. 2017.

[104] N. Reyhanian, H. Farmanbar, S. Mohajer, and Z.-Q. Luo. Resource provisioning for virtual network function deployment with in-subnetwork processing. In *Proc. IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 1–5, May 2020.

[105] N. Reyhanian, H. Farmanbar, S. Mohajer, and Z.-Q. Luo. Joint resource allocation and routing for service function chaining with in-subnetwork processing. In *IEEE*

International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 4990–4994, May 2020.

[106] D. P. Palomar and Mung Chiang. A tutorial on decomposition methods for network utility maximization. *IEEE J. Sel. Areas Commun.*, 24(8):1439–1451, Aug. 2006.

[107] A. Goldsmith. *Wireless communications.* Cambridge university press, 2005.

[108] S. Boyd and L. Vandenberghe. *Convex optimization.* Cambridge university press, 2004.

[109] R. L. Burden, J. D. Faires, and A. M. Burden. *Numerical analysis.* Cengage learning, 2015.

[110] N. Reyhanian and B. Maham. Energy-aware scheduling for broadcast erasure channels with two energy harvesting receivers. *IEEE Trans. Green Commun. Netw.*, 4(4):1174–1187, Dec. 2020.

[111] D. P. Bertsekas. *Nonlinear Programming.* Athena Scientific, 3rd ed., 2016.

[112] M. Hong, X. Wang, M. Razaviyayn, and Z.-Q. Luo. Iteration complexity analysis of block coordinate descent methods. *Math. Program.*, 163(1-2):85–114, May 2017.

[113] M. Razaviyayn, M. Hong, Z.-Q. Luo, and J.-S. Pang. Parallel successive convex approximation for nonsmooth nonconvex optimization. In *Proc. Neural Inf. Process. (NIPS)*, volume 27, Dec. 2014.

[114] C. Wolverton and T.V. Wagner. Asymptotically optimal discriminant functions for pattern classification. *IEEE Trans. on Inf. Theory*, 15(2):258–265, Mar. 1969.

[115] F. Comte and N. Marie. Bandwidth selection for the Wolverton-Wagner estimator. *arXiv preprint arXiv:1902.00734*, 2019.

[116] A.E. Zonouz, L. Xing, V.M. Vokkarane, and Y. Sun. Reliability-oriented single-path routing protocols in wireless sensor networks. *IEEE Sensors J.*, 14(11):4059–4068, Nov. 2014.

[117] A. Erickson, A. Kiasari, J. Navaridas, and I.A. Stewart. An optimal single-path routing algorithm in the datacenter network DPillar. *IEEE Trans. Parallel Distrib. Syst.*, 28(3):689–703, Mar. 2016.

[118] J. Hou, L. Sun, T. Shu, Y. Xiao, and M. Krunz. Economics of strategic network infrastructure sharing: A backup reservation approach. *IEEE/ACM Trans. Netw.*, 29(2):665–680, Dec. 2020.

[119] R. Naha and S. Garg. Multi-criteria–based dynamic user behaviour–aware resource allocation in fog computing. *ACM Trans. Internet of Things*, 2(1):1–31, Feb. 2021.

[120] S. Sharma, R. Miller, and A. Francini. A cloud-native approach to 5G network slicing. *IEEE Commun. Mag.*, 55(8):120–127, Aug. 2017.

[121] P. Rost, C. Mannweiler, D. S. Michalopoulos, C. Sartori, V. Sciancalepore, N. Sastry, O. Holland, S. Tayade, B. Han, D. Bega, D. Aziz, and H. Bakker. Network slicing to enable scalability and flexibility in 5G mobile networks. *IEEE Commun. Mag.*, 55(5):72–79, May 2017.

[122] G. Wang, G. Feng, T. Q. S. Quek, S. Qin, R. Wen, and W. Tan. Reconfiguration in network slicing—optimizing the profit and performance. *IEEE Trans. Netw. Service Manag.*, 16(2):591–605, Jun. 2019.

[123] X. Li, M. Samaka, H. A. Chan, D. Bhamare, L. Gupta, C. Guo, and R. Jain. Network slicing for 5G: Challenges and opportunities. *IEEE Internet Comput.*, 21(5):20–27, Sep. 2017.

[124] R. Wen, G. Feng, J. Tang, T. Q. S. Quek, G. Wang, W. Tan, and S. Qin. On robustness of network slicing for next-generation mobile networks. *IEEE Trans. Commun.*, 67(1):430–444, Jan. 2019.

[125] A. Baumgartner, T. Bauschert, A. M. C. A. Koster, and V. S. Reddy. Optimisation models for robust and survivable network slice design: A comparative analysis. In *Proc. IEEE Global Commun. Conf.*, pages 1–7, Dec. 2017.

[126] S. Paris, A. Destounis, L. Maggi, G. S. Paschos, and J. Leguay. Controlling flow reconfigurations in SDN. In *Proc. IEEE Conf. Comput. Commun.*, pages 1–9, May 2016.

[127] P. Caballero, A. Banchs, G. de Veciana, and X. Costa-Pérez. Multi-tenant radio access network slicing: Statistical multiplexing of spatial loads. *IEEE/ACM Trans. Netw.*, 25(5):3044–3058, Oct. 2017.

[128] P. Caballero, A. Banchs, G. De Veciana, and X. Costa-Pérez. Network slicing games: Enabling customization in multi-tenant mobile networks. *IEEE/ACM Trans. Netw.*, 27(2):662–675, Apr. 2019.

[129] M. Pozza, P. K. Nicholson, D. F. Lugones, A. Rao, H. Flinck, and S. Tarkoma. On reconfiguring 5G network slices. *IEEE J. Sel. Areas Commun.*, pages 1542–1554, Jul. 2020.

[130] D. Gligoroski and K. Kralevska. Expanded combinatorial designs as tool to model network slicing in 5G. *IEEE Access*, 7:54879–54887, Apr. 2019.

[131] S. Vittal, M. K. Singh, and A. Antony Franklin. Adaptive network slicing with multi-site deployment in 5G core networks. In *Proc. IEEE Conf. Netw.k Soft. (NetSoft)*, pages 227–231, Jun. 2020.

[132] D. H. Oliveira, T. de Araujo, and R. L. Gomes. An adaptive forecasting model for slice allocation in softwarized networks. *IEEE Trans. Netw. Service Manag.*, pages 94–103, Mar. 2021.

[133] J. Gang and V. Friderikos. Inter-tenant resource sharing and power allocation in 5G virtual networks. *IEEE Trans. Veh. Technol.*, 68(8):7931–7943, Aug. 2019.

[134] F. Wei, G. Feng, Y. Sun, Y. Wang, S. Qin, and Y. Liang. Network slice reconfiguration by exploiting deep reinforcement learning with large action space. *IEEE Trans. Netw. Service Manag.*, pages 2197–2211, Aug. 2020.

[135] H. Yang, A. Yu, J. Zhang, J. Nan, B. Bao, Q. Yao, and M. Cheriet. Data-driven network slicing from core to ran for 5G broadcasting services. *IEEE Trans. Broadcast.*, pages 23–32, Mar. 2021.

[136] X. Zhang, M. Peng, S. Yan, and Y. Sun. Deep-reinforcement-learning-based mode selection and resource allocation for cellular V2X communications. *IEEE Internet Things J.*, 7(7):6380–6391, Jul. 2020.

[137] Y. Zhang, S. Bi, and Y. A. Zhang. Joint spectrum reservation and on-demand request for mobile virtual network operators. *IEEE Trans. Commun.*, 66(7):2966–2977, Jul. 2018.

[138] X. Lyu, C. Ren, W. Ni, H. Tian, R. P. Liu, and Y. J. Guo. Multi-timescale decentralized online orchestration of software-defined networks. *IEEE J. Sel. Areas Commun.*, 36(12):2716–2730, Sep. 2018.

[139] P. Caballero, A. Banchs, G. De Veciana, X. Costa-Pérez, and A. Azcorra. Network slicing for guaranteed rate services: Admission control and resource allocation games. *IEEE Trans. Wireless Commun.*, 17(10):6419–6432, Oct. 2018.

[140] G. Zhao, S. Qin, and G. Feng. Network slice selection in softwarization based mobile networks. In *Proc. IEEE Global Commun Conf. (GLOBECOM)*, pages 1–7, Feb. 2018.

[141] D. Lugones, J. A. Aroca, Y. Jin, A. Sala, and V. Hilt. AidOps: a data-driven provisioning of high-availability services in cloud. In *Proc. Symp. Cloud Comput.*, pages 466–478, Sep. 2017.

[142] H.-T. Chien, Y.-D. Lin, C.-L. Lai, and C.-T. Wang. End-to-end slicing with optimized communication and computing resource allocation in multi-tenant 5G systems. *IEEE Trans. Veh. Technol.*, Feb. 2020.

[143] M. K. Lee and C. S. Hong. Efficient slice allocation for novel 5G services. In *Proc. Int. Conf. on Ubiquitous and Future Netw. (ICUFN)*, pages 625–629, Jul. 2018.

[144] A. Rostami. Private 5G networks for vertical industries: Deployment and operation models. In *Proc. IEEE 2nd 5G World Forum (5GWF)*, pages 433–439, Oct. 2019.

[145] Bundesnetzagentur versteigert frequenzen für den weg in die gigabit-gesellshaft. [Online]. Available: `https://www.bundesnetzagentur.de/DE/Allgemeines/Presse/Reden/5G.pdf?__blob=publicationFile&v=1`.

[146] R. Chartrand. Exact reconstruction of sparse signals via nonconvex minimization. *IEEE Signal Process. Lett.*, 14(10):707–710, Oct. 2007.

[147] J. Peng, S. Yue, and H. Li. NP/CMP equivalence: A phenomenon hidden among sparsity models l_0 minimization and l_p minimization for information processing. *IEEE Trans. Inf. Theory*, 61(7):4028–4033, Jul. 2015.

[148] M.-J. Lai and J. Wang. An unconstrained ℓ_q minimization with $0 < \ell_q$ for sparse solution of underdetermined linear systems. *SIAM J. Optim.*, 21(1):82–101, 2011.

[149] M. Hong, M. Razaviyayn, Z.-Q. Luo, and J.-S. Pang. A unified algorithmic framework for block-structured optimization involving big data: With applications in machine learning and signal processing. *IEEE Signal Process. Mag.*, 33(1):57–77, Dec. 2015.

[150] N. Reyhanian, H. Farmanbar, and Z.-Q. Luo. Resource reservation in backhaul and radio access network with uncertain user demands. In *Proc. IEEE Signal Process. Adv. Wireless Commun. (SPAWC)*, pages 1–5, May 2020.

[151] N. Reyhanian, H. Farmanbar, and Z.-Q. Luo. Resource reservation in backhaul and radio access network with uncertain user demands. *IEEE Trans. Veh. Technol.*, Oct. 2022.

[152] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2010.

[153] M. Jaggi and S. Lacoste-Julien. On the global linear convergence of frank-wolfe optimization variants. In *Proc. Neural Inf. Process. (NIPS)*, Dec. 2015.

[154] D. Lee, S. Zhou, X. Zhong, Z. Niu, X. Zhou, and H. Zhang. Spatial modeling of the traffic density in cellular networks. *IEEE Wireless Commun.*, 21(1):80–88, Mar. 2014.

[155] J. Ding, R. Xu, Y. Li, P. Hui, and D. Jin. Measurement-driven modeling for connection density and traffic distribution in large-scale urban mobile networks. *IEEE Trans. Mobile Comput.*, 17(5):1105–1118, Sep. 2018.

[156] K. Sekihara, S. Matsui, and H. Kohno. NMR imaging for magnets with large nonuniformities. *IEEE Trans. Med. Imag.*, 4(4):193–199, Dec. 1985.

[157] H. Eggers, T. Knopp, and D. Potts. Field inhomogeneity correction based on gridding reconstruction for magnetic resonance imaging. *IEEE Trans. Med. Imag.*, 26(3):374–384, Mar. 2007.

[158] M. V. W. Zibetti and A. R. De Pierro. A new distortion model for strong inhomogeneity problems in echo-planar MRI. *IEEE Trans. Med. Imag.*, 28(11):1736–1753, May 2009.

[159] J. A. Fessler, S. Lee, V. T. Olafsson, H. R. Shi, and D. C. Noll. Toeplitz-based iterative image reconstruction for MRI with correction for magnetic field inhomogeneity. *IEEE Trans. Signal Process.*, 53(9):3393–3402, Sep. 2005.

[160] A. Funai, J. A. Fessler, D. T. Yeo, V. T. Olafsson, and D. C. Noll. Regularized field map estimation in MRI. *IEEE Trans. Med. Imag.*, 27(10):1484–1494, Oct. 2008.

[161] W. Lu and B. A. Hargreaves. Multiresolution field map estimation using golden section search for water-fat separation. *Magn. Reson. Med.*, 60(1):236–244, 2008.

[162] A. Matakos and J. A. Fessler. Joint estimation of image and fieldmap in parallel MRI using single-shot acquisitions. In *Proc. IEEE Int. Symp. Biomed. Imaging from Nano to macro*, pages 984–987, Apr. 2010.

[163] M. Usman, L. Kakkar, A. Matakos, A. Kirkham, S. Arridge, and D. Atkinson. Joint B_0 and image estimation integrated with model based reconstruction for field map update and distortion correction in prostate diffusion MRI. *J. Magn. Reson. Imag.*, 65:90–99, 2020.

[164] D. Y. Zeng, J. Shaikh, S. Holmes, R. L. Brunsing, J. M. Pauly, D. G. Nishimura, S. S. Vasanawala, and J. Y. Cheng. Deep residual network for off-resonance artifact correction with application to pediatric body MRA with 3D cones. *Magn. Reson. Med.*, 82(4):1398–1411, 2019.

[165] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 770–778, Jun. 2016.

[166] J. Zbontar, F. Knoll, A. Sriram, and T. Murrell, et al. fastMRI: An open dataset and benchmarks for accelerated MRI. *arXiv preprint arXiv:1811.08839*, 2018.

[167] M. Halse, J. Rioux, S. Romanzetti, J. Kaffanke, B. MacMillan, I. Mastikhin, N. J. Shah, E. Aubanel, and B. J. Balcom. Centric scan SPRITE magnetic resonance imaging: optimization of SNR, resolution, and relaxation time mapping. *Magn. Reson. Med.*, 169(1):102–117, 2004.

[168] S. Emid and J. H. N. Creyghton. High resolution NMR imaging in solids. *Physica B+ C*, 128(1):81–83, 1985.

[169] M. Halse, D. J. Goodyear, B. MacMillan, P. Szomolanyi, D. Matheson, and B. J. Balcom. Centric scan SPRITE magnetic resonance imaging. *Magn. Reson. Med.*, 165(2):219–229, 2003.

[170] J. A. Fessler. Optimization methods for magnetic resonance image reconstruction: Key models and optimization algorithms. *IEEE Signal Process. Mag.*, 37(1):33–40, Jan. 2020.

[171] R. Gruetter and C. Boesch. Fast, noniterative shimming of spatially localized signals in vivo analysis of the magnetic field along axes. *J. Magn. Reson.*, 96(2):323–334, Feb. 1992.

[172] S. Patz, S. T. S. Wong, and M. S. Roos. Missing pulse steady-state free precession. *Magn. Reson. Med.*, 10(2):194–209, 1989.

[173] M. Uecker, P. Lai, M. Murphy, P. Virtue, M. Elad, J. M. Pauly, S. S. Vasanawala, and M. Lustig. ESPIRiT—an eigenvalue approach to autocalibrating parallel MRI: where SENSE meets GRAPPA. *Magn. Reson. Med.*, 71(3):990–1001, 2014.

[174] H. Yu, S. Reeder, A. Shimakawa, J. H. Brittain, and N. J. Pelc. Field map estimation with a region growing scheme for iterative 3-point water-fat decomposition. *Magn. Reson. Med.*, 54(4):1032–1039, 2005.

[175] D. Hernando, J. P. Haldar, B. P. Sutton, J. Ma, P. Kellman, and Z. P. Liang. Joint estimation of water/fat images and field inhomogeneity map. *Magn. Reson. Med.*, 59(3):571–580, 2008.

[176] D. Hernando, P. Kellman, J. P. Haldar, and Z. P. Liang. Robust water/fat separation in the presence of large field inhomogeneities using a graph cut algorithm. *Magn. Reson. Med.*, 63(1):79–90, 2010.

[177] C. Hu, S. Reeves, D. C. Peters, and D. Twieg. An efficient reconstruction algorithm based on the alternating direction method of multipliers for joint estimation of R_2^* and off-resonance in fMRI. *IEEE Trans. on Med. Imag.*, 36(6):1326–1336, Jun. 2017.

[178] K. Liu, X. Li, Z. Li, Y. Chen, H. Xiong, F. Chen, Q. Bao, and C. Liu. Robust water–fat separation based on deep learning model exploring multi-echo nature of mGRE. *Magn. Reson. Med.*, 85(5):2828–2841, 2021.

[179] M. N. Diefenbach, S. Ruschke, H. Eggers, J. Meineke, E. Rummeny, and D. C. Karampinos. Improving chemical shift encoding-based water–fat separation based on a detailed consideration of magnetic field contributions. *Magn. Reson. Med.*, 80(3):990–1004, 2018.

[180] S. D. Sharma, N. S. Artz, D. Hernando, D. E. Horng, and S. B. Reeder. Improving chemical shift encoded water–fat separation using object-based information of the magnetic field inhomogeneity. *Magn. Reson. Med.*, 73(2):597–604, 2015.

[181] C. Windischberger, S. Robinson, A. Rauscher, M. Barth, and E. Moser. Robust field map generation using a triple-echo acquisition. *J. Magn. Reson.*, 20(4):730–734, 2004.

[182] A. Graves, A.-R. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *Proc. of IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 6645–6649, 2013.

[183] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[184] Z. Liu and M. Hauskrecht. A regularized linear dynamical system framework for multivariate time series analysis. In *Proc. of AAAI Conference on Artificial Intelligence*, 2015.

[185] P. Tu. *Dynamical systems: an introduction with applications in economics and biology.* Springer Science & Business Media, 2012.

[186] M. Thieffry, A. Kruszewski, T.-M. Guerra, and C. Duriez. Trajectory tracking control design for large-scale linear dynamical systems with applications to soft robotics. *IEEE Trans. Control Syst. Technol.*, pages 1–11, 2019.

[187] S. Oymak. Stochastic gradient descent learns state equations with nonlinear activations. In *Proc. of Conference on Learning Theory (COLT)*, 2019.

[188] M. Hardt, T. Ma, and B. Recht. Gradient descent learns linear dynamical systems. *The Journal of Machine Learning Research*, 19(1):1025–1068, 2018.

[189] T. Laurent and J. Von Brecht. A recurrent neural network without chaos. In *Proc. of International Conference on Machine Learning (ICML)*, 2017.

[190] B. Chang, M. Chen, E. Haber, and E. Chi. Antisymmetricrnn: A dynamical system view on recurrent neural networks. *arXiv preprint arXiv:1902.09689*, 2019.

[191] Y. Sattar and S. Oymak. Non-asymptotic and accurate learning of nonlinear dynamical systems. *arXiv preprint arXiv:2002.08538*, 2020.

[192] L. Ljung. *System Identification: Theory for the User*. Prentice Hall information and system sciences series. Prentice Hall PTR, 1999.

[193] B. Ho and R. Kálmán. Effective construction of linear state-variable models from input/output functions. *at-Automatisierungstechnik*, 14(1-12):545–548, 1966.

[194] S. Venkatesh and M. Dahleh. On system identification of complex systems from finite data. *IEEE Trans. Autom. Control*, 46(2):235–257, 2001.

[195] K. Åström and P. Eykhoff. System identification—a survey. *Automatica*, 7(2):123–162, 1971.

[196] M. Pereira, M. Ibrahimi, and A. Montanari. Learning networks of stochastic differential equations. In *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, pages 172–180, 2010.

[197] M. Faradonbeh, A. Tewari, and G. Michailidis. Finite time identification in unstable linear systems. *Automatica*, 96:342–353, 2018.

[198] T. Sarkar, A. Rakhlin, and M. Dahleh. Nonparametric finite time LTI system identification. *arXiv preprint arXiv:1902.01848*, 2019.

[199] A. Wagenmaker and K. Jamieson. Active learning for identification of linear dynamical systems. In *Proc. of Conference on Learning Theory (COLT)*, pages 3487–3582, 2020.

[200] M. Simchowitz, H. Mania, S. Tu, M. Jordan, and B. Recht. Learning without mixing: Towards a sharp analysis of linear system identification. In *Proc. of Conference On Learning Theory (COLT)*, pages 439–473, 2018.

[201] A. Tsiamis and G. Pappas. Finite sample analysis of stochastic system identification. In *Proc. of Conference on Decision and Control (CDC)*, pages 3648–3654, 2019.

[202] T. Sarkar. and A. Rakhlin. How fast can linear dynamical systems be learned? *arXiv preprint arXiv:1812.01251*, 2018.

[203] M. Simchowitz, R. Boczar, and B. Recht. Learning linear dynamical systems with semi-parametric least squares. In *Proc. of Conference on Learning Theory (COLT)*, pages 2714–2802, 2019.

[204] Y. Zheng and N. Li. Non-asymptotic identification of linear dynamical systems using multiple trajectories. *arXiv preprint arXiv:2009.00739*, 2020.

[205] Y. Sun, S. Oymak, and M. Fazel. Finite sample system identification: Improved rates and the role of regularization. In *Proc. of Learning for Dynamics and Control*, 2020.

[206] C. Tallec and Y. Ollivier. Can recurrent neural networks warp time? In *Proc. of International Conference on Learning Representations (ICLR)*, 2018.

[207] S. Bahmani and J. Romberg. Convex programming for estimation in nonlinear recurrent models. *Journal of Machine Learning Research*, 21(235):1–20, 2020.

[208] N. Vural, S. Yilmaz, F. Ilhan, and S. Kozat. RNN-based online learning: An efficient first-order optimization algorithm with a convergence guarantee. *arXiv preprint arXiv:2003.03601*, 2020.

[209] C. Martin and R. Hermann. Applications of algebraic geometry to systems theory: The McMillan degree and ronecker indices of transfer functions as topological and holomorphic system invariants. *SIAM Journal on Control and Optimization*, 16(5):743–755, 1978.

[210] M. Hazewinkel and C. Martin. Representations of the symmetric groups, the specialization order, systems and grassmann manifolds. *EUR Report*, (353), 1983.

[211] X. Liu. A global canonical form for nonlinear singular control systems. *International Journal of Systems, Control and Communications*, 1(1):82–97, 2008.

[212] J. Saw, M. Yang, and T. Mo. Chebyshev inequality with estimated mean and variance. *The American Statistician*, 38(2):130–132, 1984.

[213] T. Tsiligkaridis and A. Hero. Covariance estimation in high dimensions via kronecker product expansions. *IEEE Trans. Signal Process*, 61(21):5347–5360, 2013.

[214] Y. Nesterov. Introductory lectures on convex programming volume i: Basic course. *Lecture notes*, 3(4):5, 1998.

[215] R. Gower Mansel, N. Loizou, X. Qian, A. Sailanbayev, E. Shulgin, and P. Richtárik. SGD: General analysis and improved rates. In *Proc. of International Conference on Machine Learning (ICML)*, pages 5200–5209, 2019.

[216] D. Luenberger. Introduction to dynamic systems; theory, models, and applications. Technical report, 1979.

[217] P. Brunovskỳ. A classification of linear controllable systems. *Kybernetika*, 6(3):173–188, 1970.

[218] M. Razaviyayn, M. Hong, N. Reyhanian, and Z.-Q. Luo. A linearly convergent doubly stochastic Gauss–Seidel algorithm for solving linear equations and a certain class of over-parameterized optimization problems. *Mathematical Programming*, 176(1-2):465–496, 2019.

[219] J. Liu and S. Wright. An accelerated randomized Kaczmarz algorithm. *Mathematics of Computation*, 85(297):153–178, 2016.

[220] A. Ma, D. Needell, and A. Ramdas. Convergence properties of the randomized extended Gauss–Seidel and Kaczmarz methods. *SIAM Journal on Matrix Analysis and Applications*, 36(4):1590–1604, 2015.