Successive Convex Approximation: Analysis and Applications

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

Meisam Razaviyayn

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

Zhi-Quan (Tom) Luo

May, 2014

Acknowledgements

First and foremost, I would like to thank my thesis advisor, Professor Tom Luo, for his continuous encouragement and guidance during this research. He has taught me, both consciously and unconsciously, how an original research idea is formed and investigated. Despite his busy schedule, he has always been extremely open to academic fruitful discussions. He is an extremely sharp thinker and our frequent academic discussions helped me a great deal in the completion of this dissertation. Tom has been exceptionally supportive and has given me freedom to pursue various research projects without any objection. I would also like to extend my sincerest appreciation to his confidence in my research abilities, true understanding of my concerns during my graduate studies, and many other reasons which cannot be expressed in the space provided.

I would like to thank my committee members, Professor Georgios B. Giannakis, Professor Mihailo R. Jovanovic, and Professor Yousef Saad. I have received invaluable comments and feedback from all of them which resulted in great improvement of the quality and the presentation of my research. I got to know Professor Giannakis from the very first days of my graduate studies and since then he has always been exceptionally kind and supportive; and shared his indispensable comments and advices on my research. I have known Professor Jovanovic also from the early days at the University of Minnesota. During our occasional chats, I found him very kind and supportive. His attitude towards students makes him a true supportive friend rather than a formal professor/advisor for almost all the graduate students at the university, including myself. I am also thankful to Professor Saad for kindly accepting to be in my committee and especially for my knowledge on matrix analysis which had a significant impact on this research. In addition to my committee members, I would also like to thank Professor

Jong-Shi Pang for providing extremely useful comments on this dissertation. The application of the successive convex approximation methods in games has been first brought to my attention by him; and the related sections of this dissertation is the direct result of our fruitful discussions during my visit at the University of Southern California.

My sincerest thanks go to Professor Maury Bramson, Professor Gennady Lyubeznik, Professor Pavlo Pylyavskyy, Professor Ravi Janardan, Professor Daniel Boley, and Professor Nihar Jindal for guiding and educating me during my graduate studies. In particular, I would like to thank Professor Gennady Lyubeznik, my M.Sc. advisor, and Professor Maury Bramson for spending long hours on research discussions and helping me getting insights on my research. I would also like to thank my B.Sc. advisor, Professor Mohammadali Khosravifard, for introducing me to the research world.

I am also grateful to my officemates Dennis Chu, Jaymes Grossman, Mingyi Hong, Bo Jiang, Lei Jiao, Mojtaba Kadkhodaie, Wei-Cheng Liao, Yingxi Liu, Yao Morin, Randall Plate, Alireza Razavi, Maziar Sanjabi, Qingjiang Shi, Ruoyu Sun, Andy Tseng, Xiangfeng Wang, and Yu Zhang. Many thanks also go to my friends Ali Ghoreyshi, Mojtaba Kadkhodaie, Morteza Mardani, Maral Mousavi, and Armin Zare for proofreading this thesis.

Last but not least, I would not be standing here without the endless love, support, and inspiration of my family: my mom Parivash, my dad Morteza, my sister Maryam, and my brother Matin.

Dedication

This dissertation is dedicated to my family for their endless love and unconditional support.

Abstract

The block coordinate descent (BCD) method is widely used for minimizing a continuous function f of several block variables. At each iteration of this method, a single block of variables is optimized, while the remaining variables are held fixed. To ensure the convergence of the BCD method, the subproblem of each block variable needs to be solved to its unique global optimal. Unfortunately, this requirement is often too restrictive for many practical scenarios. In this dissertation, we first study an alternative inexact BCD approach which updates the variable blocks by successively minimizing a sequence of approximations of f which are either locally tight upper bounds of f or strictly convex local approximations of f. Different block selection rules are considered such as cyclic (Gauss-Seidel), greedy (Gauss-Southwell), randomized, or even multiple (Parallel) simultaneous blocks. We characterize the convergence conditions and iteration complexity bounds for a fairly wide class of such methods, especially for the cases where the objective functions are either non-differentiable or non-convex. Also the case of existence of a linear constraint is studied briefly using the alternating direction method of multipliers (ADMM) idea. In addition to the deterministic case, the problem of minimizing the expected value of a cost function parameterized by a random variable is also investigated. An inexact sample average approximation (SAA) method, which is developed based on the successive convex approximation idea, is proposed and its convergence is studied. Our analysis unifies and extends the existing convergence results for many classical algorithms such as the BCD method, the difference of convex functions (DC) method, the expectation maximization (EM) algorithm, as well as the classical stochastic (sub-)gradient (SG) method for the nonsmooth nonconvex optimization, all of which are popular for large scale optimization problems involving big data.

In the second part of this dissertation, we apply our proposed framework to two practical problems: interference management in wireless networks and the dictionary learning problem for sparse representation. First, the computational complexity of these problems are studied. Then using the successive convex approximation framework, we propose novel algorithms for these practical problems. The proposed algorithms are evaluated through extensive numerical experiments on real data.

Contents

Acknowledgements **i**

Dedication **iii**

Abstract **iv**

List of Tables **viii**

List of Figures **ix**

1 Introduction **1**

1.1 Technical Preliminaries and Notations 5

2 Successive Convex Approximation **8**

2.1 Single Block Successive Convex Approximation 8

2.1.1 Problem Statement and Prior Work 8

2.1.2 Convergence Analysis 10

2.2 Multi-block Successive Convex Approximation 11

2.2.1 Prior Work 11

2.2.2 Block Successive Upper-bound Minimization Algorithm 11

2.2.3 Maximum Improvement Successive Upper-bound Minimization . 18

2.2.4 Successive Convex Approximation of a Smooth Function 19

2.2.5 Overlapping Essentially Cyclic Rule 21

2.2.6 BSUM with Linear Coupling Constraints 23

2.3 Random Parallel Successive Convex Approximation 27

2.3.1 Prior Work . . . 27
2.3.2 Algorithm Description . . . 29
2.3.3 Convergence Analysis: Asymptotic Behavior . . . 31
2.3.4 Convergence Analysis: Iteration Complexity . . . 32
2.4 Stochastic Successive Upper-bound Minimization . . . 35
2.4.1 Algorithm Description and Prior Work . . . 35
2.4.2 Asymptotic Convergence Analysis . . . 36
2.5 Successive Convex Approximation in Games . . . 42
2.5.1 Prior Work . . . 42
2.5.2 Problem Statement and Algorithm Description . . . 43
2.5.3 Gauss-Seidel Update Rule . . . 44
2.5.4 Jacobi Update Rule: . . . 48

3 Applications **52**
3.1 Interference Management in Wireless Heterogenous Networks . . . 52
3.1.1 Prior Work . . . 53
3.1.2 Beamformer Design in Multi-user Wireless Networks . . . 59
3.1.3 Joint Beamforming and Scheduling in Multi-user Networks . . . 66
3.1.4 Beamforming for Max-Min Fairness . . . 79
3.1.5 Expected Sum-Rate Maximization for Wireless Networks . . . 84
3.2 Dictionary Learning for Sparse Representation . . . 88
3.2.1 Problem Statement . . . 89
3.2.2 Prior Work . . . 89
3.2.3 Complexity Analysis . . . 90
3.2.4 Batch Dictionary Learning . . . 91
3.2.5 Online Dictionary Learning . . . 95
3.3 Other Applications . . . 97
3.3.1 Proximal Minimization Algorithm . . . 97
3.3.2 Proximal Splitting Algorithm . . . 99
3.3.3 CANDECOMP/PARAFAC Decomposition of Tensors . . . 101
3.3.4 Expectation Maximization Algorithm . . . 104
3.3.5 Concave-Convex Procedure/Difference of Convex Functions . . . 106

3.3.6 Stochastic (Sub-)Gradient Method and its Extensions 107

4 Numerical Experiments **111**

4.1 Interference Management in Wireless Networks 111

4.1.1 Beamforming in Wireless Networks 111

4.1.2 Joint Beamforming and Scheduling 114

4.1.3 Beamforming for Max-Min Fairness 119

4.1.4 Expected Sum-Rate Maximization 124

4.2 Dictionary Learning for Sparse Represnetation 127

5 Future Work **130**

References **133**

Appendix A. Proofs **156**

List of Tables

3.1 Average number of iterations for convergence 104
4.1 Achieved user rates in different groups/time slots 120
4.2 Image denoising result comparison on “Lena” 129

List of Figures

1.1 The contour plot of the function $f(z) = \|Az\|_1$ with $A = [3\ 4; 2\ 1]$. . . . 7
3.1 The dense structure of the new cellular networks. 53
3.2 BSUM convergence for tensor decomposition (small scale example) . . . 103
3.3 Convergence of different algorithms for tensor decomposition 104
4.1 WMMSE algorithm: (a) SISO-IFC (b) MIMO-IFC. 112
4.2 Average sum-rate versus SNR in the SISO IFC case. 113
4.3 Average sum-rate versus SNR in the MIMO IFC case. 113
4.4 Average CPU time versus the number of users in the MIMO IFC case. . 114
4.5 19-hexagonal wrap around cell layout 115
4.6 Rate CDF of different methods 116
4.7 Rate CDF of different methods at various iterations 117
4.8 Geometric Mean vs. Iterations 118
4.9 Rate CDF of various methods 118
4.10 Rate CDF: $K = 4, I = 3, M = 6, N = 2, d = 1$ 122
4.11 Minimum rate in the system versus transmit power 122
4.12 Rate CDF: $K = 5, I = 3, M = 3, N = 2, d = 1$ 123
4.13 Minimum rate in the system 123
4.14 WMMSE objective function while adding a User 124
4.15 Minimum rate while adding a User 124
4.16 WMMSE objective function while changing the channel 125
4.17 Minimum rate while changing the channel 125
4.18 Expected sum rate: $\eta = 6$) 127
4.19 Expected sum rate: $\eta = 12$ 127
4.20 Sample denoised images ($\sigma = 100$). 128

A.1 Channels for the clause $c_\ell : x_i + \bar{x}_j + x_k$. 188

Chapter 1

Introduction

Consider the following optimization problem

$$\begin{aligned} \min \quad & f(x_1, \ldots, x_n) \\ \text{s.t.} \quad & x_i \in \mathcal{X}_i,\ i = 1, 2, \ldots, n, \end{aligned}$$

where $\mathcal{X}_i \subseteq \mathbb{R}^{m_i}$ is a closed convex set, and $f : \prod_{i=1}^{n} \mathcal{X}_i \to \mathbb{R}$ is a continuous function. A popular approach for solving the above optimization problem is the block coordinate descent (BCD) method. At each iteration of this method, the function is minimized with respect to a single block of variables while the rest of the blocks are held fixed. More specifically, at iteration r of the algorithm, the block variable x_i is updated by solving the following subproblem

$$x_i^r = \arg\min_{y_i \in \mathcal{X}_i} \quad f(x_1^r, \ldots, x_{i-1}^r, y_i, x_{i+1}^{r-1}, \ldots, x_n^{r-1}), \quad i = 1, 2, \ldots, n. \tag{1.1}$$

Let us use $\{x^r\}$ to denote the sequence of iterates generated by this algorithm, where $x^r \triangleq (x_1^r, \ldots, x_n^r)$. Due to its particular simple and scalable implementation, the BCD method has been widely used for solving problems such as power allocation in wireless communication systems [1], clustering [2], image denoising and image reconstruction [3] and dynamic programming [4].

The updating order of the blocks in the algorithm will result in different optimization

methods. For example, the block selection choice could be cyclic (Gauss-Seidel), randomized, greedy (Gauss-Southwell); or even multiple parallel blocks could be updated at each iteration. Analytically, the convergence of the algorithm typically requires solving the subproblem (2.13) to its unique minimizer, or doing just simple gradient descent (also known as block coordinate gradient descent method [5]). On one hand, doing the simple gradient descent step might not be optimal in practical scenarios since it is only based on the first order information and ignores the higher order information; on the other hand, solving the per-block optimization problem might not be closed form; see, e.g., [6].

To overcome such difficulties, one can modify the BCD algorithm by optimizing a well-chosen *approximate* version of the objective function at each iteration. It is very hard to find the root of the classical idea of successively approximating the original objective with a sequence of convex approximations (also known as majorization-minimization [7]). Also the classical gradient descent method, for example, can be viewed as an implementation of such strategy. To illustrate, recall that the update rule of the gradient descent method is given by

$$x^{r+1} = x^r - \alpha^{r+1} \nabla f(x^r).$$

This update rule is equivalent to solving the following problem

$$x^{r+1} = \arg\min_{x} \quad g(x, x^r),$$

where

$$g(x, x^r) \triangleq f(x^r) + \nabla f(x^r)(x - x^r) + \frac{1}{2\alpha^{r+1}} \|x - x^r\|^2,$$

and yields to the block coordinate gradient descent method; see [5, 8–10]. Clearly, the function $g(x, x^r)$ is an approximation of $f(\cdot)$ around the point x^r. In fact, as we will see later in this dissertation, successively optimizing an approximate version of the original objective is the key idea of many important algorithms such as the concave-convex procedure [11], the expectation maximization (EM) algorithm [12], the proximal minimization algorithm [13], to name a few. Furthermore, this idea can be used to simplify the computation and to guarantee the convergence of the original BCD

algorithm with the Gauss-Seidel update rule (e.g. [5], [14], [15]). However, despite its wide applicability, there appears to be no general unifying convergence analysis for this class of algorithms. The only general existing result is in [16] which considers only one block of variable and no rigorous convergence analysis of the algorithm is provided. Recently and concurrently with this research, some asymptotic convergence analysis of the algorithm has been done in the context of multi-agent optimizations; see [17–21].

In this dissertation, first we provide a unified convergence analysis for a general class of inexact BCD methods in which a sequence of approximate versions of the original problem are solved successively. Two types of approximations are considered: one being a locally tight upper bound for the original objective function, the other being a convex local approximation of the objective function. In the general nonconvex nonsmooth setting, we provide asymptotic convergence analysis for these successive approximation strategies as well as for various types of updating rules, including cyclic (Gauss-Seidel), greedy (Gauss-Southwell), randomized, randomized Jacobi (Parallel), or the overlapping essentially cyclic update rule. By allowing inexact solution of subproblems, our work unifies and extends several existing algorithms and their convergence analysis, including the difference of convex functions (DC) method, the expectation maximization (EM) algorithm, as well as the alternating proximal minimization algorithm. Besides, our analysis shows that the convergence of these algorithms are guaranteed even when the variables are updated in a block coordinate manner. Moreover, in the convex scenario, we can handle the existence of linear constraints using the alternating direction method of multipliers (ADMM) idea and analyze the iteration complexity of the proposed method for different choice of block selections such as cyclic, randomized, or randomized Jacobi update rules. In addition to the deterministic scenario, we applied this idea to the stochastic optimization problems and showed that under some mild assumptions, still the asymptotic convergence of the resulting algorithm is guaranteed. The application of this idea in the non-cooperative game setting is also studied briefly.

In the third chapter of this dissertation, we apply the introduced optimization framework to different practical problems. In particular, we first consider the interference management problem in wireless networks. Due to resource sharing nature of multiuser wireless networks, a central issue in the study of these new networks is how to mitigate multiuser interference. In practice, there are several commonly used methods for

dealing with interference. First, we can treat the interference as noise and just focus on extracting the desired signals. This approach is widely used in practice because of its simplicity and ease of implementation, but is known to be non-capacity achieving in general. An alternative technique is channel orthogonalization whereby transmitted signals are chosen to be nonoverlapping either in time, frequency or space, leading to Time Division Multiple Access, Frequency Division Multiple Access, or Space Division Multiple Access respectively. While channel orthogonalization effectively eliminates multiuser interference, it can lead to inefficient use of communication resources and is also generally non-capacity achieving. Another interference management technique is to decode and remove interference. Specifically, when interference is strong relative to the desired signals, a user can decode the interference first, then subtract it from the received signal, and finally decode its own message. Due to the complexity issues and rate limitations caused by the decodability of interference, this approach is impractical and non capacity achieving. Unfortunately, none of the aforementioned interference management techniques can achieve the maximum system throughput in general. In the optimal strategy the transmission from different nodes of the network should be optimally coordinated. Such coordination can take the form of joint scheduling, joint transceiver design or even joint data processing in the base stations. In the third chapter of this dissertation, we study the application of our optimization framework in the joint beamforming and scheduling problem in the wireless networks.

As another application of our framework, we consider the dictionary learning problem for sparse representation. We first show that this problem is NP-hard and then propose an efficient dictionary learning scheme to solve several practical formulations of this problem. Our proposed algorithms are based on the successive convex approximation idea and unlike many existing algorithms in the literature, such as K-SVD [22], our proposed dictionary learning scheme is theoretically guaranteed to converge to the set of stationary points under certain mild assumptions. Finally, in the last section of this dissertation, we numerically evaluate the performance of the proposed algorithms in both interference management and sparse dictionary learning problem.

In short, the contributions of this dissertation are as follows:

- Proposing an optimization framework by combining the successive convex approximation idea and block coordinate descent method.

- Analyzing the convergence of the algorithm for various block selection rules such as cyclic, greedy, randomized, or parallel. The convergence analyses studies the asymptotic and non-asymptotic behavior of the algorithm in both convex and non-convex setup

- Extending the proposed idea and its convergence to stochastic optimization, optimizations with linear constraints, as well as the non-cooperative games.

- Investigating two practical non-convex problems: interference management in wireless networks and the dictionary learning problem for sparse representation; and studying the computational complexity of these methods.

- Proposing a novel approximation function for the sum utility maximization problem in the wireless heterogenous networks. The suggested approximation function together with the proposed optimization framework will result in a series of efficient algorithms for beamforming, power allocation, and user scheduling with theoretical convergence guarantee.

- Presenting various algorithms, resulted from our optimization framework, for the (sparse) dictionary learning problem. Unlike the existing methods in the literature, the convergence of the presented algorithms is guaranteed theoretically with our general framework.

1.1 Technical Preliminaries and Notations

Throughout the dissertation, we adopt the following notations. We use $\mathbb{R}^m$ to denote the space of m dimensional real valued vectors, which is also represented as the Cartesian product of n lower dimensional real valued vector spaces, i.e.,

$$\mathbb{R}^m = \mathbb{R}^{m_1} \times \mathbb{R}^{m_2} \times \ldots \times \mathbb{R}^{m_n},$$

where $\sum_{i=1}^{n} m_i = m$. We use the notation $(0, \ldots, d_k, \ldots, 0)$ to denote the vector of all zeros except the k-th block, with $d_k \in \mathbb{R}^{m_k}$. The following concepts/definitions are adopted in this dissertation:

- **Distance of a point from a set:** Let $\mathcal{S} \subseteq \mathbb{R}^m$ be a set and x be a point in $\mathbb{R}^m$, the distance of the point x from the set $\mathcal{S}$ is defined as

$$d(x, \mathcal{S}) = \inf_{s \in \mathcal{S}} \|x - s\|,$$

where $\|\cdot\|$ denotes the 2-norm in $\mathbb{R}^m$.

- **Directional derivative:** Let $f : \mathcal{D} \to \mathbb{R}$ be a function where $\mathcal{D} \subseteq \mathbb{R}^m$ is a convex set. The directional derivative of f at point x in direction d is defined by

$$f'(x; d) \triangleq \liminf_{\lambda \downarrow 0} \frac{f(x + \lambda d) - f(x)}{\lambda}.$$

- **Stationary points of a function:** Let $f : \mathcal{D} \to \mathbb{R}$ be a function where $\mathcal{D} \subseteq \mathbb{R}^m$ is a convex set. The point x is a stationary point of $f(\cdot)$ if $f'(x; d) \geq 0$ for all d such that $x + d \in \mathcal{D}$. In this dissertation we use the notation $\mathcal{X}^*$ to denote the set of stationary points of a function.

- **Quasi-convex function:** The function f is quasi-convex if

$$f(\theta x + (1 - \theta)y) \leq \max\{f(x), f(y)\}, \quad \forall\, \theta \in (0, 1),\ \forall\, x, y \in \text{ dom } f$$

- **Coordinatewise minimum of a function:** $z \in \text{dom } f \subseteq \mathbb{R}^m$ is coordinatewise minimum of f with respect to the coordinates in $\mathbb{R}^{m_1}, \mathbb{R}^{m_2}, \ldots, \mathbb{R}^{m_n}$, $m_1 + \ldots + m_n = m$ if

$$f(z + d_k^0) \geq f(z), \quad \forall\, d_k \in \mathbb{R}^{m_k} \quad \text{with} \quad z + d_k^0 \in \text{ dom } f, \ \ \forall k = 1, 2, \ldots, n,$$

where $d_k = (0, \ldots, d_k, \ldots, 0)$.

- **Natural history of a stochastic process:** Consider a real valued stochastic process $\{Z^r\}_{r=1}^{\infty}$. For each r, we define the natural history of the stochastic process up to time r as

$$\mathcal{F}^r \ = \ \sigma(Z^1, \ldots, Z^r),$$

where $\sigma(Z^1, \ldots, Z^r)$ denotes the σ-algebra generated by the random variables $Z^1, \ldots, Z^r$.

- **Infinity norm of a function:** Let $h : \ D \mapsto \mathbb{R}$ be a function, where $D \subseteq \mathbb{R}^n$. The infinity norm of the function $h(\cdot)$ is defined as

$$\|h\|_\infty \triangleq \sup_{x \in D} |h(x)|.$$

- **Regularity of a function at a point:** The function $f : \mathbb{R}^m \to \mathbb{R}$ is regular at the point $z \in \text{dom} f$ with respect to the coordinates $m_1, m_2, \ldots, m_n$, $m_1 + m_2 + \ldots + m_n = m$, if $f'(z; d) \geq 0$ for all $d = (d_1, d_2, \ldots, d_n)$ with $f'(z; d_k^0) \geq 0$, where $d_k^0 \triangleq (0, \ldots, d_k, \ldots, 0)$ and $d_k \in \mathbb{R}^{m_k}, \forall\ k$.

As an example, consider the function $f(z) = \|Az\|_1$, where $A = [3\ 4; 2\ 1] \in \mathbb{R}^{2\times 2}$. This function is not regular at the point $z^* = (-4, 3)$ with respect to the two standard coordinates since $f'(z^*; d) \geq 0,\ \forall d \in \{(d_1, d_2) \in \mathbb{R}^2 | d_1 d_2 = 0\}$; but $f'(z^*; d^*) < 0$ for $d^* = (4, -3)$. This fact can be also observed in the contour plot of the function in Figure 1.1.

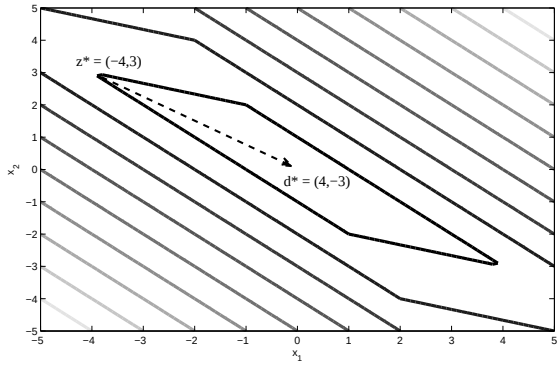


Figure 1.1: The contour plot of the function $f(z) = \|Az\|_1$ with $A = [3\ 4; 2\ 1]$.

For detailed discussion on the regularity of a function, the readers are referred to [23, Lemma 3.1].

Chapter 2

Successive Convex Approximation

The successive convex approximation (SCA) idea has been widely used in different contexts before. In this chapter, we give some preliminary results on the convergence guarantees of different algorithms developed based on this idea.

2.1 Single Block Successive Convex Approximation

2.1.1 Problem Statement and Prior Work

Consider the following optimization problem:

$$\begin{aligned} \min_{x} \quad & h_0(x) \triangleq f_0(x) + g_0(x) \\ \text{s.t.} \quad & h_i(x) \triangleq f_i(x) + g_i(x) \leq 0, \forall i = 1, \ldots, m, \end{aligned} \tag{2.1}$$

where the function $f_i(x)$ is smooth (possibly nonconvex) and g_i is convex (possibly nonsmooth), for all $i = 0, \ldots, m$. A popular practical approach for solving this problem is the successive convex approximation (also known as majorization minimization) approach where at each iteration of the method, a locally tight approximation of the original optimization problem is solved subject to a tight convex restriction of the constraint sets. More precisely, we consider the successive convex approximation method in Algorithm 1.

Algorithm 1 Successive Convex Approximation Method for Solving (2.1)

Find a feasible point x^0 in (2.1), choose a stepsize $\gamma \in (0, 1]$, and set $r = 0$

repeat

Set $r \leftarrow r + 1$

Set $\hat{x}^r$ to be a solution of the following optimization problem

$$\begin{aligned} \min_{x} \quad & \tilde{h}_0(x, x^r) \\ \text{s.t.} \quad & \tilde{h}_i(x) \leq 0, \quad \forall i = 1, \ldots, m. \end{aligned}$$

Set $x^{r+1} \leftarrow \gamma \hat{x}^r + (1 - \gamma) x^r$

until some convergence criterion is met

The approximation functions in the algorithm need to satisfy the following assumptions:

Assumption 1 *Assume the approximation functions $\tilde{h}_i(\bullet, \bullet)$, $\forall i = 0, \ldots, m$, satisfy the following assumptions:*

- *$\tilde{h}_i(x, y)$ is continuous in (x, y)*
- *$\tilde{h}_i(x, y)$ is convex in x*
- *$\tilde{h}_i(x, y) = \tilde{f}_i(x, y) + g_i(x)$, $\forall x, y$*
- *Function value consistency: $\tilde{f}_i(x, x) = f_i(x)$, $\forall x$*
- *Gradient consistency: $\nabla \tilde{f}_i(\bullet, x)(x) = \nabla f_i(x)$, $\forall x$*
- *Upper-bound: $\tilde{f}_i(x, y) \geq f_i(x)$, $\forall x, y$*

In other words, we assume that at each iteration, we approximate the original functions with some upper-bounds of them which have the same first order behavior.

To the best of our knowledge, the previous analysis of the SCA method is very limited. In fact, the classical paper [16] suggests the inner approximation algorithm (IAA) which is in many ways similar to our suggested framework. The only difference is that

the IAA algorithm is only applicable for problems with smooth objectives, while our framework algorithm is able to handle nonsmooth objectives/constraints as well. It is worth mentioning that the existing convergence result for the IAA algorithm is quite weak. In particular, [16, Theorem 1] states that if the whole sequence converges, then the algorithm should converge to a stationary point. A stronger convergence result was stated in [24, Property 3] where only smooth case is treated. In what follows, we give a simple convergence analysis of this framework which is more general than the existing ones.

2.1.2 Convergence Analysis

To state our result, we need to define the following condition:

Slater condition for SCA: Given the constraint approximation functions $\{\tilde{h}(\cdot,\cdot)\}_{i=1}^{m}$, we say that the Slater condition is satisfied at a given point $\bar{x}$ if there exists a point x in the interior of the restricted constraint sets at the point $\bar{x}$, i.e.,

$$\tilde{h}_i(x,\bar{x}) < 0, \quad \forall i = 1,\ldots,m,$$

for some x. Notice that if the approximate constraints are the same as the original constraints, then this condition will be the same as the well-known Slater condition for strong duality.

Theorem 1 *Let $\bar{x}$ be a limit point of the iterates generated by Algorithm 1. Assume Assumption 1 is satisfied and Slater condition holds at the point $\bar{x}$. Then $\bar{x}$ is a KKT point of* (2.1).

Proof See the appendix chapter for the proof.

It is worth noting that in the presence of linear constraints, the Slater condition should be considered for the relative interior of the constraint set instead of the interior. Furthermore, the Slater condition (or some other constraint qualification condition) seems to be necessary for the convergence of this simple approach. For example, when the

convex approximation at the first step is so that the restricted constraint set become a singleton, then the algorithm will stuck in a non-interesting point of the problem. In order to relax the constraint qualification condition and achieve stronger convergence results, we consider no approximation of the constraint set in the rest of this chapter.

2.2 Multi-block Successive Convex Approximation

2.2.1 Prior Work

In many practical applications, the optimization variables can be decomposed into independent blocks. Such block structure, when judiciously exploited, can lead to low-complexity algorithms that are distributedly implementable.

The asymptotic convergence behavior of the BCD algorithm is studied exhaustively in the literature; see, e.g., [23] for the general non-convex non-smooth asymptotic analysis. In general the non-asymptotic convergence analysis of the algorithm is not trivial even for the convex case (without assuming per-block strong convexity). Especially for the cyclic update rule, to the date of this dissertation no general result is known. When the objective function is strongly convex and smooth, the BCD algorithm converges globally linearly [25–27]. In addition, such linear rate is global when the feasible set is compact. This line of analysis, which is based on the error bound assumption, has recently been extended to allow certain class of nonsmooth functions in the objective [9, 28–30]. For the general convex (but not strongly convex) case, various results suggest the sublinear $\mathcal{O}(1/r)$ rate of convergence; see, e.g., [8, 10, 31–34] and the references therein. Although these results are very interesting and essential, none of them covers the block successive upper-bound minimization/successive convex approximation approach where at each iteration a general first order approximation of the objective function is minimized. In this section, we study different optimization algorithms using the SCA idea on multi-block optimization variables.

2.2.2 Block Successive Upper-bound Minimization Algorithm

Let us assume that the feasible set $\mathcal{X}$ is the cartesian product of n closed convex sets: $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_n$, with $\mathcal{X}_i \subseteq \mathbb{R}^{m_i}$ and $\sum_i m_i = m$. Accordingly, the optimization

variable $x \in \mathbb{R}^m$ can be decomposed as: $x = (x_1, x_2, \ldots, x_n)$, with $x_i \in \mathcal{X}_i,\ i = 1, \cdots, n$. We are interested in solving the problem

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in \mathcal{X}. \end{aligned} \tag{2.2}$$

Different from the SUM algorithm, the Block Successive Upper-bound Minimization (BSUM) algorithm only updates a single block of variables in each iteration. More precisely, at iteration r, the selected block (say block i) is computed by solving the following subproblem

$$\begin{aligned} \min_{x_i} \quad & u_i(x_i, x^{r-1}) \\ \text{s.t.} \quad & x_i \in \mathcal{X}_i, \end{aligned} \tag{2.3}$$

where $u_i(\cdot, x^{r-1})$ is again an approximation (in fact, a global upper-bound) of the original objective $f(\cdot)$ at the point x^{r-1}. Algorithm 2 summarizes the main steps of the BSUM algorithm. Note that although the blocks are updated following a simple cyclic rule, the algorithm and its convergence results can be easily extended to the (more general) essentially cyclic update rule as well. This point will be further elaborated in Section 2.2.5.

Algorithm 2 Pseudo code of the BSUM algorithm

Find a feasible point $x^0 \in \mathcal{X}$ and set $r = 0$

repeat

Set $r \leftarrow r + 1$, choose a block $i \in \{1, \ldots, n\}$

Let $\mathcal{X}^r = \arg\min_{x_i \in \mathcal{X}_i} u_i(x_i, x^{r-1})$

Set x_i^r to be an arbitrary element in $\mathcal{X}^r$

Set $x_k^r = x_k^{r-1}, \quad \forall\, k \neq i$

until some convergence criterion is met

Now we are ready to study the convergence behavior of the BSUM algorithm. To this end, the following regularity conditions on the function $u_i(\cdot, \cdot)$ are needed.

Assumption 2

$$u_i(y_i, y) = f(y), \quad \forall\, y \in \mathcal{X}, \forall\, i \tag{2.4}$$

$$u_i(x_i, y) \geq f(y_1, \ldots, y_{i-1}, x_i, y_{i+1}, \ldots, y_n), \quad \forall\, x_i \in \mathcal{X}_i, \forall\, y \in \mathcal{X}, \forall\, i \tag{2.5}$$

$$u_i'(x_i, y; d_i)\Big|_{x_i = y_i} = f'(y; d), \quad \forall\, d = (0, \ldots, d_i, \ldots, 0) \text{ s.t. } y_i + d_i \in \mathcal{X}_i, \forall\, i \tag{2.6}$$

$$u_i(x_i, y) \text{ is continuous in } (x_i, y), \quad \forall\, i \tag{2.7}$$

The following proposition identifies a sufficient condition to ensure (2.6).

Proposition 1 *Assume $f(x) = f_0(x) + f_1(x)$, where $f_0(\cdot)$ is differentiable and the directional derivative of $f_1(\cdot)$ exists at every point $x \in \mathcal{X}$. Consider $u_i(x_i, y) = u_{0,i}(x_i, y) + f_1(x)$ with $u_{0,i}(x_i, y)$ satisfying*

$$\begin{aligned} &u_{0,i}(x_i, x) = f_0(x), \quad \forall\, x \in \mathcal{Y}, \quad \forall\, i \\ &u_{0,i}(x_i, y) \geq f_0(y_1, \ldots, y_{i-1}, x_i, y_{i+1}, \ldots, y_n), \ \forall\, x, y \in \mathcal{Y} \quad \forall\, i, \end{aligned}$$

where $\mathcal{Y}$ is an open set containing $\mathcal{X}$. Then, (2.4)*,* (2.5)*, and* (2.6) *hold.*

Proof The proof is elementary and can be found in [35].

To have a complete algorithm, we need to identify the choice of the block selection rule in the algorithm. In this section, we consider two different types of block selection in the BSUM algorithm:

Cyclic: In the cyclic block selection rule, the blocks are chosen at each iteration according to the following rule:

$$i = (r \bmod n) + 1.$$

Randomized: In the randomized selection rule, at each iteration r, only one block is selected (independent of the previous iterations) so that

$$Pr(\text{block } i \textit{ being selected}) = p_i^r \geq p_{\min} > 0.$$

We first analyze the convergence of the BSUM algorithm with cyclic selection rule of the blocks. Our convergence results regarding to the cyclic BSUM algorithm consist of two parts. In the first part, a quasi-convexity of the objective function is assumed, which guarantees the existence of the limit points. This is in the same spirit of the classical proof of convergence for the BCD method in [13]. However, if we know that the iterates lie in a compact set, then a stronger result can be proved. Indeed, in the second part of the theorem, the convergence is obtained by relaxing the quasi-convexity assumption while imposing the compactness assumption of level sets.

Theorem 2

(a) Consider cyclic variable selection rule in the BSUM algorithm. Suppose that the function $u_i(x_i, y)$ is quasi-convex in x_i for $i = 1, \ldots, n$, and Assumption 2 holds. Furthermore, assume that the subproblem (2.3) *has a unique solution for any point $x^{r-1} \in \mathcal{X}$. Then, every limit point z of the iterates generated by the BSUM algorithm is a coordinatewise minimum of* (2.2)*. In addition, if $f(\cdot)$ is regular at z, then z is a stationary point of* (2.2)*.*

(b) Consider cyclic variable selection rule in the BSUM algorithm. Suppose the level set $\mathcal{X}^0 = \{x \mid f(x) \leq f(x^0)\}$ is compact and Assumption 2 holds. Furthermore, assume that $f(\cdot)$ is regular at any point in $\mathcal{X}^0$ and the subproblem (2.3) *has a unique solution for any point $x^{r-1} \in \mathcal{X}$ for at least $n-1$ blocks. Then, the iterates generated by the BSUM algorithm converge to the set of stationary points, i.e.,*

$$\lim_{r \to \infty} \quad d(x^r, \mathcal{X}^*) = 0.$$

Proof See the appendix chapter for the proof.

Theorem 2 extends the existing result of block coordinate descent method [13] and [23] to the BSUM case where only an approximation of the objective function is minimized at each iteration. As we will see in the next chapter, our result implies the convergence of several existing algorithms including the EM algorithm or the DC method when the Gauss-Seidel update rule is used.

A key assumption in Theorem 2 is the uniqueness of the minimizer of (2.3), while

the classical BCD method requires the uniqueness of the minimizer of (2.2) with respect to each block for convergence. This property is an advantage of BSUM over BCD since the uniqueness of the minimizer of (2.3) depends on the choice of the upperbound $u(x, y)$, while the uniqueness of the solution of (2.2) per-block depends only on the objective function. Another key assumption in Theorem 2 is the regularity of the objective function. Notice that this assumption is necessary even for the classical BCD method. To see the necessity of this assumption consider the function $f(z) = \|Az\|$, with $A = [3\ 4; 2\ 1] \in \mathbb{R}^{2\times 2}$, defined in the introduction section. Clearly, the point $z^* = (-4, 3)$ is a fixed point of the BCD method, while it is not a stationary point of $f(\cdot)$. Next we will show that For the randomized selection rule the uniqueness of the minimizer assumption is not necessary.

Theorem 3 *Consider randomized variable selection rule in the BSUM algorithm and assume that Assumption 2 holds. Furthermore, assume that $f(\cdot)$ is regular and $f(\cdot)$ is bounded from below. Then, every limit point of the iterates generated by the randomized BSUM algorithm is a stationary point with probability one.*

Proof See the appendix section.

Iteration Complexity Analysis of BSUM for Convex Case

In this subsection, the analysis of the BSUM method for the convex case is considered. In our iteration complexity analysis, we consider the following optimization problem:

$$\begin{aligned} \min_{x} \quad & f(x_1, \ldots, x_n) + \sum_{i=1}^{n} g_i(x_i) \\ \text{s.t.} \quad & x_i \in \mathcal{X}_i, \end{aligned}$$

where the function $f(x)$ is convex and smooth; and the possible nonsmooth functions $\{g_i(x_i)\}$ are separable convex. We use the BSUM algorithm for solving the above problem by assuming the following assumptions.

Assumption 3

- $u_i(x_i, y) = \tilde{f}_i(x_i, y) + g_i(x_i)$
- *Function value consistency:* $\tilde{f}_i(x_i, x) = f(x),\ \forall x$
- *Gradient consistency:* $\nabla \tilde{f}_i(\bullet, x)(x_i) = \nabla_{x_i} f(x),\ \forall x$
- *Upper-bound:* $\tilde{f}_i(x_i, y) \geq f(x),\ \forall x, y$
- $\tilde{f}_i(x_i, y)$ *is continuous in* (x_i, y) *and strongly convex in* x_i*, i.e.,*

$$\tilde{f}_i(x_i, y) \geq \tilde{f}_i(x_i', y) + \langle x_i - x_i', \nabla_{x_i} \tilde{f}_i(x_i', y)\rangle + \frac{\tau}{2}\|x_i' - x_i\|^2$$

- *For any given* y*,* $\tilde{f}_i(\cdot, y)$ *has a uniform Lipschitz continuous gradient, i.e.,*

$$\|\nabla_{x_i} \tilde{f}(x_i, y) - \nabla_{x_i} \tilde{f}(x_i', y)\| \leq L_i \|x_i - x_i'\|, \forall y \in \mathcal{X}, \forall x_i, x_i' \in \mathcal{X}_i$$

- *The function* $f(x)$ *has a Lipschitz continuous gradient, i.e.,*

$$\|\nabla f(x) - \nabla f(x')\| \leq L_f \|x - x'\|, \quad \forall x, x' \in \mathcal{X}$$

The following theorem states the iteration complexity analysis of the BSUM method.

Theorem 4 *[36, Theorem 3.1] Assume Assumption 3 is satisfied and* f^* *is the optimal objective value. Furthermore, let us assume that the level set* $\{x \mid f(x) \leq f(x^0)\}$ *is compact.*

(a) Consider cyclic variable selection rule in the BSUM algorithm. Then $f(x^r) - f^* = \mathcal{O}\left(\frac{1}{r}\right)$*.*

(b) Consider randomized variable selection rule in the BSUM algorithm and assume that the nonsmooth function $g_i(\cdot)$ *is Lipschitz continuous for all blocks, i.e.,* $|g_i(x_i) - g_i(x_i')| \leq L_g \|x_i - x_i'\|,\ \forall x_i, x_i' \in \mathcal{X}_i,\ \forall i$*. Then* $\mathbb{E}\left[f(x^r) - f^*\right] = \mathcal{O}\left(\frac{1}{r}\right)$*.*

It is worth noting that in the iteration complexity analysis of the randomized BSUM method, we assume that the nonsmooth part is Lipschitz continuous. This assumption is satisfied for many popular nonsmooth optimization problems such as Lasso or group

Lasso.

In many applications, the optimization problem is of particular form which may satisfy further assumptions. Next make more assumptions on the optimization problem to claim linear rate of convergence.

Assumption 4

- *The global minimum of the original optimization problem is attained and so is its dual optimal value. The intersection of the feasible set and the interior of the domain of the objective function is non-empty.*
- *The function $f(x)$ can be decomposed as $f(x) = \ell(Ex) + \langle b, x\rangle$, where $\ell(\cdot)$ is a strictly convex and continuously differentiable function on int, and E is some given matrix (not necessarily full column rank).*
- *Each nonsmooth function $g_i(\cdot)$ if present, has the following form*

$$g_i(x_i) = \eta_i \|x_i\|_1 + \sum_k \omega_{i,k} \|x_{i,k}\|_2,$$

  where $(x_{i,k})_k$ is a partition of x_i and $\eta_i, \omega_{i,k} \geq 0$, $\forall k, i$.
- *The feasible sets $\mathcal{X}_i$ are polyhedral sets given by $\mathcal{X}_i \triangleq \{x_i \mid C_i x_i \leq c_i\}$.*

Now we are ready to state the result:

Theorem 5 *[37, Theorem 3.1] Assume Assumption 4 and Assumption 4 are satisfied. Furthermore, let us assume that the level set $\{x \mid f(x) \leq f(x^0)\}$ is compact. Let f^* be optimum objective value.*

(a) Consider cyclic variable selection rule in the BSUM algorithm. Then $f(x^r)$ converges Q-linearly to f^*.

(b) Consider randomized variable selection rule in the BSUM algorithm and assume that $\omega_{i,k} = 0$, $\forall i, k$, and C_i has full row rank. Then $\mathbb{E}[f(x^r)]$ converges Q-linearly to f^*.

2.2.3 Maximum Improvement Successive Upper-bound Minimization

A key assumption for the BSUM algorithm is the uniqueness of the minimizer of the subproblem. This assumption is necessary even for the simple BCD method [13]. In general, by removing such assumption, the convergence is not guaranteed (see [38] for examples) unless we assume pseudo convexity in pairs of the variables [39], [23]. In this section, we explore the possibility of removing such uniqueness assumption.

Recently, Chen *et al.* [40] have proposed a related Maximum Block Improvement (MBI) algorithm, which differs from the conventional BCD algorithm only by its update schedule. More specifically, only the block that provides the *maximum improvement* is updated at each step. Remarkably, by utilizing such modified updating rule (which is similar to the well known Gauss-Southwell update rule), the per-block subproblems are allowed to have multiple solutions. Inspired by this recent development, we propose to modify the BSUM algorithm similarly by simply updating the block that gives the maximum improvement. We name the resulting algorithm the Maximum Improvement Successive Upper-bound Minimization (MISUM) algorithm, and list its main steps in Algorithm 3.

Algorithm 3 Pseudo code of the MISUM algorithm

Find a feasible point $x^0 \in \mathcal{X}$ and set $r = 0$

repeat

Set $r \leftarrow r + 1$

Let $k = \arg\min_i \min_{x_i} u_i(x_i, x^{r-1})$

Let $\mathcal{X}^r = \arg\min_{x_k \in \mathcal{X}_k} u_k(x_k, x^{r-1})$

Set x_k^r to be an arbitrary element in $\mathcal{X}^r$

Set $x_i^r = x_i^{r-1}, \quad \forall\, i \neq k$

until some convergence criterion is met

Clearly the MISUM algorithm is more general than the MBI method proposed in [40], since only an approximate version of the subproblem is solved at each iteration. Theorem 6 states the convergence result for the proposed MISUM algorithm.

Theorem 6 *Suppose that Assumption 2 is satisfied. Then, every limit point z of the iterates generated by the MISUM algorithm is a coordinatewise minimum of* (2.2). *In*

addition, if $f(\cdot)$ is regular at z, then z is a stationary point of (2.2).

Proof See the appendix chapter for the proof.

The main advantage of the MISUM algorithm over the BSUM algorithm is that its convergence does not rely on the uniqueness of the minimizer for the subproblems. On the other hand, each iteration of MISUM algorithm is more expensive than the BSUM since the minimization needs to be performed for all the blocks. Nevertheless, the MISUM algorithm is more suitable when parallel processing units are available, since the minimizations with respect to all the blocks can be carried out simultaneously.

2.2.4 Successive Convex Approximation of a Smooth Function

In the previous subsections, we have demonstrated that the stationary solutions of the studied optimization problems can be obtained by successively minimizing a sequence of upper-bounds of $f(\cdot)$. However, in practice, unless the objective $f(\cdot)$ possesses certain convexity/concavity structure, those upper-bounds may not be easily identifiable. In this section, we extend the BSUM algorithm by further relaxing the requirement that the approximation functions $\{u_i(x_i, y)\}$ must be the global upper-bounds of the original objective f.

Throughout this section, we use $h_i(.,.)$ to denote the convex approximation function for the ith block. Suppose that $h_i(x_i, x)$ is no longer a global upper-bound of $f(x)$, but only a first order approximation of $f(x)$ at each point, i.e.,

$$h_i'(y_i, x; d_i)\Big|_{y_i=x_i} = f'(x; d), \quad \forall\, d = (0, \ldots, d_i, \ldots, 0) \quad \text{with} \quad x_i + d_i \in \mathcal{X}_i. \tag{2.8}$$

In this case, simply optimizing the approximate functions in each step may not even decrease the objective function. Nevertheless, the minimizer obtained in each step can still be used to construct a good search direction, which, when combined with a proper step size selection rule, can yield a sufficient decrease of the objective value.

Suppose that at iteration r, the i-th block needs to be updated. Let $y_i^r \in \min_{y_i \in \mathcal{X}_i} h_i(y_i, x^{r-1})$ denote the optimal solution for optimizing the i-th approximation function at the point x^{r-1}. We propose to use $y_i^r - x_i^{r-1}$ as the search direction, and adopt the Armijo rule to guide the step size selection process. We name the resulting algorithm the Block

Successive Convex Approximation (BSCA) algorithm. Its main steps are given in Algorithm 4.

Algorithm 4 Pseudo code of the BSCA algorithm

Find a feasible point $x^0 \in \mathcal{X}$ and set $r = 0$

repeat

Set $r \leftarrow r + 1$, $i = (r \bmod n) + 1$

Let $\mathcal{X}^r = \arg\min_{x_i \in \mathcal{X}_i} h_i(x_i, x^{r-1})$

Set y_i^r to be an arbitrary element in $\mathcal{X}^r$ and set $y_k^r = x_k^{r-1}, \quad \forall\, k \neq i$

Set $d^r = y^r - x^{r-1}$ and choose $\sigma \in (0, 1)$

Armijo step-size rule: Choose $\alpha^{\text{init}} > 0$ and $\sigma, \beta \in (0, 1)$. Let α^r be the largest element in $\{\alpha^{\text{init}}\beta^j\}_{j=0,1,\ldots}$ satisfying:

$f(x^{r-1}) - f(x^{r-1} + \alpha^r d^r) \geq -\sigma\alpha^r f'(x^{r-1}; d^r)$

Set $x^r = x^{r-1} + \alpha^r(y^r - x^{r-1})$

until some convergence criterion is met

Note that for $d^r = (0, \ldots, d_i^r, \ldots, 0)$ with $d_i^r = y_i^r - x_i^{r-1}$, we have

$$f'(x^{r-1}; d^r) = h_i'(x_i, x^{r-1}; d_i^r)\Big|_{x_i = x_i^r} = \lim_{\lambda \downarrow 0} \frac{h_i(x_i^{r-1} + \lambda d_i^r, x^{r-1}) - h_i(x_i^{r-1}, x^{r-1})}{\lambda} \leq 0, \tag{2.9}$$

where the inequality is due to the fact that $h_i(\cdot)$ is convex and $y_i^r = x_i^{r-1} + d_i^r$ is the minimizer at iteration r. Moreover, there holds

$$f(x^{r-1}) - f(x^{r-1} + \alpha d^r) = -\alpha f'(x^{r-1}; d^r) + o(\alpha), \quad \forall\, \alpha > 0.$$

Hence the Armijo step size selection rule in Algorithm 4 is well defined when $f'(x^{r-1}; d^r) \neq 0$, and there exists $j \in \{0, 1, \ldots\}$ such that for $\alpha^r = \alpha^{\text{init}}\beta^j$,

$$f(x^r) - f(x^r + \alpha^{r+1} d^{r+1}) \geq -\sigma\alpha^{r+1} f'(x^r; d^{r+1}). \tag{2.10}$$

The following theorem states the convergence result of the proposed algorithm.

Theorem 7 *Suppose that $f(\cdot)$ is continuously differentiable and that Assumption* (2.8) *holds. Furthermore, assume that $h(x, y)$ is strictly convex in x and continuous in (x, y).*

Then every limit point of the iterates generated by the BSCA algorithm is a stationary point of (2.2).

Proof See the appendix chapter.

We remark that the proposed BSCA method is related to the coordinate gradient descent method [5], in which a strictly convex second order approximation of the objective function is minimized at each iteration. It is important to note that the convergence results of these two algorithms do not imply each other. The BSCA algorithm, although more general in the sense that the approximation function can take the form of any strictly convex function satisfying (2.8), only covers the case when the objective function is smooth. Nevertheless, the freedom provided by the BSCA to choose a more general approximation function allows one to better approximate the original function at each iteration. It is also worth noting that the idea of coordinate line search method has also appeared in [41] where the unconstrained smooth optimization problem is considered. An efficient line search algorithm is proposed so that the subproblems related to certain blocks are solved approximately. Another interesting related work is [42] where the direction d is obtained by projected gradient direction with respect to only one of the coordinates.

2.2.5 Overlapping Essentially Cyclic Rule

In both the BSUM and the BSCA algorithms considered in the previous sections, variable blocks are updated in a simple cyclic manner. In this section, we consider a very general block scheduling rule named the overlapping essentially cyclic rule and show they still ensure the convergence of the BSUM and the BSCA algorithms.

In the so called overlapping essentially cyclic rule, at each iteration r, a group ϑ^r of the variables is chosen to be updated where

$$\vartheta^r \subseteq \{1, 2, \ldots, n\} \quad \text{and} \quad \vartheta^r \neq \emptyset.$$

Furthermore, we assume that the update rule is essentially cyclic with period T, i.e.,

$$\bigcup_{i=1}^{T} \vartheta^{r+i} = \{1, 2, \ldots, n\}, \quad \forall\, r.$$

Notice that in the classical essentially cyclic rule [23], in addition to the above condition, the cardinality of each set ϑ must be one for all r; while in the overlapping essentially cyclic method, the blocks are allowed to have overlaps. Using the overlapping essentially cyclic update rule, almost all the convergence results presented so far still hold. For example, the following corollary extends the convergence of BSUM to the overlapping essentially cyclic case.

Corollary 1

(a) Assume that the function $u_i(x_i, y)$ is quasi-convex in x_i and Assumption 2 is satisfied. Furthermore, assume that the overlapping essentially cyclic update rule with period T is used and the subproblem (2.3) *has a unique solution for every block ϑ^r. Then, every limit point z of the iterates generated by the BSUM algorithm is a coordinatewise minimum of* (2.2)*. In addition, if $f(\cdot)$ is regular at z with respect to the updated blocks, then z is a stationary point of* (2.2)*.*

(b) Assume the level set $\mathcal{X}^0 = \{x \mid f(x) \leq f(x^0)\}$ is compact and Assumption 2 is satisfied. Furthermore, assume that the overlapping essentially cyclic update rule is used and the subproblem (2.3) *has a unique solution for every block ϑ^r. If $f(\cdot)$ is regular (with respect to the updated blocks), then the iterates generated by the BSUM algorithm converges to the set of stationary points, i.e.,*

$$\lim_{r\to\infty} \quad d(x^r, \mathcal{X}^*) = 0.$$

Proof The proof of both cases are similar to the proof of the BSUM algorithm with the simple cyclic update rule. Here we only present the proof for case (a). The proof of part (b) is similar.

Let $\{x^{r_j}\}$ be a convergent subsequence whose limit is denoted by z. Consider every T updating cycle along the subsequence $\{x^{r_j}\}$, namely, $\{(x^{r_j}, x^{r_j+1}, \ldots, x^{r_j+T-1})\}$. Since the number of different subblocks ϑ^r is finite, there must exist a (fixed) T tuple of variable blocks, say $(\vartheta_0, \vartheta_1, \ldots, \vartheta_{T-1})$, that has been updated in infinitely many T updating cycles. By restricting to the corresponding subsequence of $\{x^{r_j}\}$, we have

$$x_{\vartheta_i}^{r_j+i+1} = \arg\min_{x_{\vartheta_i}} \; u_{\vartheta_i}(x_{\vartheta_i}, x^{r_j+i}), \quad \forall\, i = 0, 1, 2, \ldots, T-1.$$

The rest of the proof is the same as the proof of part (a) in Theorem 2. The only difference is that the steps of the proof need to be repeated for the blocks $(\vartheta_0, \vartheta_1, \ldots, \vartheta_{T-1})$ instead of $(1, \ldots, n)$.

In the proof of Corollary 1, we first restrict ourselves to a fixed set of T variable blocks that have been updated in infinitely many consecutive T update cycles. Then, we use the same approach as in the proof of the convergence of cyclic update rule. Using the same technique, we can extend the results in Theorem 7 to the overlapping essentially cyclic update rule. More specifically, we have the following corollary.

Corollary 2 *Assume $f(\cdot)$ is smooth and the condition* (2.8) *is satisfied. Furthermore, assume that $h(x, y)$ is strictly convex in x and the overlapping essentially cyclic update rule is used in the BSCA algorithm. Then every limit point of the iterates generated by the BSCA algorithm is a stationary point of* (2.2).

Notice that the overlapping essentially cyclic rule is not applicable to the MISUM algorithm in which the update order of the variables is given by the amount of improvement. However, one can simply check that the proof of Theorem 6 still applies to the case when the blocks are allowed to have overlaps.

2.2.6 BSUM with Linear Coupling Constraints

In the previous subsections, we assume that there is no coupling constraint among the variables. A simple coupling constraint, which appears in many practical scenarios, is the linear constraint. In this subsection, we will see that the BSUM idea can be naturally combined with the Alternating Direction Method of Multipliers (ADMM) idea [43–46] to deal with the linear coupling constraints. The ADMM which combines the dual ascent method with the BCD approach is very popular in practical problems due to its distributed implementation and fast convergence; see [47–49].Consider the optimization

problem

$$\begin{aligned} \min_{x} \quad & f(x_1, \ldots, x_n) + \sum_{i=1}^{n} g_i(x_i) \\ \text{s.t.} \quad & A_1 x_1 + A_2 x_2 + \ldots + A_n x_n = b \\ & x_i \in \mathcal{X}_i, \quad \forall i, \end{aligned} \tag{2.11}$$

where $f(\cdot)$ is convex smooth and $g_i(\cdot)$ is convex and possibly nonsmooth. Here $b \in \mathbb{R}^m$, $A_i \in \mathbb{R}^{m \times n_i}$, and $x_i \in \mathbb{R}^{n_i}$ with $\sum_{i=1}^{n} n_i = n$. Problems of this form appear in many practical problems such as basis pursuit problem [50], demand-response control power in smart grids [51, 52], and dynamic spectrum management [53].

The linear coupling constraint prevents us from obtaining per-block update of the variables in the algorithm directly. A popular approach to deal with the linear coupling constraint is the Alternating Direction Method of Multipliers (ADMM) where the linear constraint is added to the objective using the augmented Lagrangian regularizer and the dual variables are updated using a gradient ascent step in the dual problem. Let us define the augmented Lagrangian function:

$$\mathcal{L}(x_1, \ldots, x_n, \lambda) = f(x_1, \ldots, x_n) + \sum_{i=1}^{n} g_i(x_i) + \langle \lambda, \sum_{i=1}^{n} A_i x_i - b \rangle + \frac{\rho}{2} \| \sum_{i=1}^{n} A_i x_i - b \|^2.$$

Then, the ADMM approach is summarized in Algorithm 5.

Algorithm 5 Alternating Direction Method of Multipliers

Find a feasible point $x^0 \in \mathcal{X} \triangleq \mathcal{X}_1 \times \ldots \times \mathcal{X}_n$; choose $\rho > 0$; set $r = 0$; and $\lambda^0 = \lambda_{\text{init}}$

repeat

  for $i = 1, 2, \ldots, n$ **do**

    $x_i^{r+1} \leftarrow \arg\min_{x_i} \mathcal{L}(x_1^{r+1}, \ldots, x_{i-1}^{r+1}, x_i, x_{i+1}^{r}, \ldots, x_n^{r}, \lambda^r)$

  end for

  Set $\lambda^{r+1} \leftarrow \lambda^r + \alpha^r (\sum_{i=1}^{n} A_i x_i^{r+1} - b)$

  Set $r \leftarrow r + 1$

until some convergence criterion is met

One drawback of the ADMM algorithm is that the primal variables' update rule could be costly in general. When the primal update rule is not closed form, a natural modification is to replace the original objective function with an approximation of it. Utilizing the BSUM idea, we can modify the ADMM algorithm to obtain the BSUM-M method described in Algorithm 6.

Algorithm 6 Block Successive Upper-bound Minimization Method of Multipliers (BSUM-M)

Find a feasible point $x^0 \in \mathcal{X}$; set $r = 0$; and $\lambda^0 = \lambda_{\text{init}}$

repeat

  for $i = 1, 2, \ldots, n$ **do**

    $x_i^{r+1} \leftarrow \arg\min_{x_i} \tilde{\mathcal{L}}_i(x_i, x_1^{r+1}, \ldots, x_{i-1}^{r+1}, x_i^r, \ldots, x_n^r, \lambda^r)$

  end for

  Set $\lambda^{r+1} \leftarrow \lambda^r + \alpha^r(\sum_{i=1}^n A_i x_i^{r+1} - b)$

  Set $r \leftarrow r + 1$

until some convergence criterion is met

The difference between the BSUM-M algorithm and the ADMM algorithm is that in the BSUM-M method, the approximation of the augmented Lagrangian is used instead of the original Lagrangian. More precisely, we define

$$\tilde{\mathcal{L}}_i(x_i, y, \lambda) \triangleq \tilde{f}_i(x_i, y_1, \ldots, y_n) + g_i(x_i) + \langle \lambda, A_i x_i \rangle + \frac{\rho}{2} \| \sum_{j \neq i} A_j y_j + A_i x_i - b \|^2.$$

In other words, instead of applying one round of block coordinate update on all the primal variables, we apply one BSUM round for updating the primal variables. Similar to the randomized BSUM idea, we can introduce the randomized BSUM-M described in Algorithm 7.

Algorithm 7 Randomized Block Successive Upper-bound Minimization Method of Multipliers (RBSUM-M)

Find a feasible point $x^0 \in \mathcal{X}$; set $r = 0$; and $\lambda^0 = \lambda_{\text{init}}$

Pick a probability vector $\{p_i\}_{i=0}^n$ with $\sum_{i=0}^n p_i = 1$

repeat

Draw a random index $i \in \{0, \ldots, n\}$ with probability p_i

If $i = 0$

$\lambda^{r+1} \leftarrow \lambda^r + \alpha^r(\sum_{i=1}^n A_i x_i^r - b)$

$x^{r+1} \leftarrow x^r$

If $i \neq 0$

$x_i^{r+1} \leftarrow \arg\min_{x_i} \tilde{\mathcal{L}}_i(x_i, x_1^{r+1}, \ldots, x_{i-1}^{r+1}, x_i^r, \ldots, x_n^r, \lambda^r)$

$x_j^{r+1} \leftarrow x_j^r,\ \forall j \neq i$

Set $r \leftarrow r + 1$

until some convergence criterion is met

Now we are ready to state the convergence result of the BSUM-M method.

Theorem 8 *[37, Theorem 2.1] Suppose that Assumption 4 and Assumption 4 are satisfied. Furthermore, let us assume that the feasible set $\mathcal{X}_i$ is compact for all i. Let us further assume that one of the following step-size selection rules are adopted: 1) for all r, $\alpha^r = \alpha$ is sufficiently small, or 2) the step-size α^r satisfies: $\sum_{r=1}^\infty \alpha^r = \infty$, $\lim_{r\to\infty} \alpha^r = 0$. Then*

(a) For the BSUM-M algorithm, $\| \sum_{i=1}^n A_i x_i - b\| \to 0$ and every limit point of (x^r, λ^r) is primal dual optimal solution.

(b) For the RBSUM-M algorithm, $\| \sum_{i=1}^n A_i x_i - b\| \to 0$ and every limit point of (x^r, λ^r) is primal dual optimal solution, with probability one.

2.3 Random Parallel Successive Convex Approximation

2.3.1 Prior Work

Consider the following optimization problem

$$\begin{aligned} \min_{x} \quad & h(x) \triangleq f(x_1, \ldots, x_n) + \sum_{i=1}^{n} g_i(x_i) \\ \text{s.t.} \quad & x_i \in \mathcal{X}_i,\ i = 1, 2, \ldots, n, \end{aligned} \tag{2.12}$$

where $\mathcal{X}_i \subseteq \mathbb{R}^{m_i}$ is a closed convex set; the function $f : \prod_{i=1}^{n} \mathcal{X}_i \to \mathbb{R}$ is a smooth function (possibly nonconvex); and $g(x) = \sum_{i=1}^{n} g_i(x_i)$ is a separable convex function (possibly nonsmooth). The above optimization problem appears in various fields such as machine learning, signal processing, wireless communication, image processing, social networks, and bioinformatics, to name just a few. These optimization problems are typically of huge size and should be solved instantaneously.

A popular approach for solving the above multi-block optimization problem is the block coordinate descent (BCD) approach, where at each iteration of BCD only one of the blocks is updated while the remaining blocks are held fixed. Since only one block is updated at each iteration, the algorithm required memory and the computational complexity per-iteration is low, which is desirable in big data problems. Furthermore, as observed in [32,54], these methods particulary perform well in numerical experiments.

With the recent progress and increasing availability of the high performance multi-core machines, it is desirable to use these technological hardware advances by designing parallel optimization schemes. One classical class of methods that could be easily parallelized is the class of (proximal) gradient methods. These methods are parallelizable in nature [5, 55–58]; however, they are typically equivalent to optimizing a quadratic approximation of the smooth part of the objective function which may not be a tight approximation; and hence suffer from low practical convergence speed [20].

In order to take the advantages of the block coordinate descent method and the parallel machines hardware simultaneously, different algorithms have been proposed in recent years for solving the multi-block optimization problems. In particular, the references [59–61] propose parallel coordinate descent minimization methods for ℓ_1-regularized convex optimization problems. Using the greedy (Gauss-Southwell) type

of update rule, the recent works [20, 62] propose parallel BCD type methods for general nonsmooth optimization problems. In contrast, references [29, 63] suggest the use of randomized block selection rule, which is more amenable to big data optimization problems, in order to parallelize the BCD method.

Motivated by [20, 35], and [32], we propose a random parallel block coordinate descent method where at each iteration of the algorithm, a random subset of the blocks is updated by minimizing locally tight approximations of the original objective function. We provide the asymptotic and non-asymptotic convergence analysis of the algorithm for both convex and nonconvex scenarios. It is also worth noting that, although parallel, our algorithm is synchronized, unlike the existing lock-free methods in [64, 65].

The contributions of this section are as follows.

- A randomized parallel block coordinate descent type method is proposed for nonconvex nonsmooth methods. To the best of our knowledge, reference [20] is the only existing algorithm in the literature for nonconvex nonsmooth methods. This reference utilizes greedy block selection rule which requires searching among all blocks and communication among processing nodes in order to find the best blocks to update. This requirement might be demanding in practical scenarios where the communication among nodes are limited or when the number of blocks are huge.

- Unlike many existing algorithms in the literature, e.g. [29, 62, 63], our algorithm utilizes the general approximation of the original function which includes the linear/proximal approximation of the objective as a special case.

- We provide iteration complexity analysis of the algorithm for both convex and nonconvex scenarios. Unlike the existing parallel methods in the literature such as [20] which only guarantees the asymptotic behavior of the algorithm, we provide non-asymptotic guarantees on the algorithm as well.

- The proposed method not only works with the constant step-size selection rule, but also with the diminishing step-size which is desirable when the Lipschitz constant of the objective function is not known.

2.3.2 Algorithm Description

As stated in the introduction section, a popular approach for solving (2.12) is the BCD method where at each iteration of this method, the function is minimized with respect to a single block of variables while the rest of the blocks are held fixed. More specifically, at iteration $r+1$ of the algorithm, the block variable x_i is updated by solving the following subproblem

$$x_i^{r+1} = \arg\min_{x_i \in \mathcal{X}_i} \quad h(x_1^r, \ldots, x_{i-1}^r, x_i, x_{i+1}^r, \ldots, x_n^r). \tag{2.13}$$

In many practical situations, the function $h(\cdot)$ might not be convex and hence the update rule (2.13) is not easy to perform. One popular approach is to replace the function $h(\cdot)$ with its convex approximation $h_i(x_i, x^r)$ in (2.13). In other words, at iteration $r+1$ of the algorithm, the block variable x_i is updated by

$$x_i^{r+1} = \arg\min_{x_i \in \mathcal{X}_i} \quad \tilde{h}_i(x_i, x^r),, \tag{2.14}$$

where $\tilde{h}_i(x_i, x^r)$ is a convex (possibly upper-bound) approximation of the function $h(\cdot)$ with respect to the i-th block around the current iteration x^r. This approach, also known as *successive convex approximation* or *successive upper-bound minimization* [35], has been widely used in different applications; see [35] for more details.

In this part of the dissertation, we assume that the approximation function $\tilde{h}_i(\cdot)$ is of the following form:

$$\tilde{h}_i(x_i, y) = \tilde{f}_i(x_i, y) + g_i(x_i). \tag{2.15}$$

Here $\tilde{f}_i(\cdot, y)$ is an approximation of the function $f(\cdot)$ around the point y with respect to the i-th block. We further assume that $\tilde{f}_i(x_i, y) : \mathcal{X}_i \times \mathcal{X} \to \mathbb{R}$ satisfies the following assumptions:

- $\tilde{f}_i(\cdot, y)$ is continuously differentiable and strongly convex with parameter τ_i for all $y \in \mathcal{X}$, i.e.,

$$\tilde{f}_i(x_i, y) \geq \tilde{f}_i(x_i', y) + \langle \nabla_{x_i} \tilde{f}_i(x_i', y), x_i - x_i' \rangle + \frac{\tau_i}{2} \|x_i - x_i'\|^2, \ \forall x_i, x_i' \in \mathcal{X}_i, \ \forall y \in \mathcal{X}$$

- *Gradient consistency assumption*:

$$\nabla_{x_i}\tilde{f}_i(x_i,x)=\nabla_{x_i}f(x),\ \forall x\in\mathcal{X} \tag{2.16}$$

- $\nabla_{x_i}\tilde{f}_i(x_i,\cdot)$ is Lipschitz continuous on $\mathcal{X}$ for all $x_i\in\mathcal{X}_i$ with constant $\tilde{L}$, i.e.,

$$\|\nabla_{x_i}\tilde{f}_i(x_i,y)-\nabla_{x_i}\tilde{f}_i(x_i,z)\|\leq\tilde{L}\|y-z\|,\quad \forall y,z\in\mathcal{X},\ \forall x_i\in\mathcal{X}_i,\ \forall i.$$

With the recent advances in the development of parallel processing machines, it is desirable to take the advantage of parallel processing by updating multiple blocks at the same time in (2.37). Unfortunately, naively updating multiple blocks using the approach (2.37) will not result in a convergent algorithm. Hence, we suggest to modify the update rule using a well chosen step size. More precisely we suggest Algorithm 8 for solving the optimization problem (2.12).

Algorithm 8 Randomized Parallel Successive Convex Approximation (RPSCA) Algorithm

find a feasible point $x^0\in\mathcal{X}$ and set $r=0$

repeat

choose a subset $S^r\subseteq\{1,\dots,n\}$

calculate $\hat{x}_i^r=\arg\min_{x_i\in\mathcal{X}_i}\ \tilde{h}_i(x_i,x^r),\ \forall i\in S^r$

set $x_i^{r+1}=x_i^r+\gamma^r(\hat{x}_i^r-x_i^r),\ \forall i\in S^r$

set $x_i^{r+1}=x_i^r,\ \forall\ i\notin S^r$

set $r=r+1$

until some convergence criterion is met

First of all, notice that when the approximation function is the standard proximal approximation, the proposed update rule is different than adaptively changing the quadratic penalization constant, which has been considered before in the literature. Secondly, in Algorithm 8, the selection of the subset S^r could be done based on different rules. A recent work [20] suggests to use a Gauss-Southwell variable selection rule where at each iteration the blocks are chosen in a greedy manner. In other words, at each iteration of the algorithm in [20], the best response of all the variables are calculated and at

the end, only the block variables with the largest amount of improvement are updated. A drawback of this approach is in the calculation of all the best responses especially when the size of the problem is huge. Unlike the work [20], we suggest a randomized variable selection rule. More precisely, at each iteration r, the set S^r is chosen randomly and independently from the previous iterations such that

$$Pr(j \in S^r \mid x^r) = p_j^r \geq p_{\min} > 0, \quad \forall j = 1, 2, \ldots, n, \ \forall r$$

2.3.3 Convergence Analysis: Asymptotic Behavior

To study the asymptotic convergence behavior of the above algorithm for the general non-convex scenario, we need to assume that $\nabla f(\cdot)$ is Lipschitz continuous with constant $L_{\nabla f}$, i.e.,

$$\|\nabla f(x) - \nabla f(y)\| \leq L_{\nabla f}\|x - y\|.$$

Let us also define $\bar{x}$ to be a *stationary point* of (2.12) if $\exists\ d \in \partial g(\bar{x})$ such that $\langle \nabla f(\bar{x}) + d, x - \bar{x} \rangle \geq 0,\ \forall x \in \mathcal{X}$, i.e., the first order optimality condition is satisfied at the point $\bar{x}$. The following lemma will help us to study the convergence of the RPSCA algorithm.

Lemma 1 *[20, Lemma 2] Define the mapping $\hat{x}(\cdot) : \mathcal{X} \mapsto \mathcal{X}$ as $\hat{x}(y) = (\hat{x}_i(y))_{i=1}^n$ with*

$$\hat{x}_i(y) = \arg\min_{x_i} \tilde{h}_i(x_i, y).$$

Then the mapping $\hat{x}(\cdot)$ is continuous Lipschitz with the constant $\hat{L} = \frac{\sqrt{n}\tilde{L}}{\tau_{\min}}$, i.e.,

$$\|\hat{x}(y) - \hat{x}(z)\| \leq \hat{L}\|y - z\|,\ \forall y, z \in \mathcal{X}$$

Proof Compared to the result of [20, Lemma 2], our result does not use the proximal regularizer. However, the proof in [20, Lemma 2] can be easily modified to handle our case when there is no proximal regularizer as well. ■

Having the above result in our hands, we are now ready to state our first result which studies the limiting behavior of the RPSCA algorithm. This result is based on the sufficient decrease of the objective function which has been also utilized in [20] for non-random choice of the variables.

Theorem 9 *Assume* $\gamma^r \in (0,1]$, $\sum_{r=1}^{\infty}\gamma^r = +\infty$, *and that* $\limsup_{r\to\infty}\gamma^r < \bar{\gamma} \triangleq \min\{\frac{\tau_{\min}}{L_{\nabla f}}, \frac{\tau_{\min}}{\tau_{\min}+\tilde{L}\sqrt{n}}\}$. *Then every limit point of the iterates is a stationary point of* (2.12) *with probability one.*

Proof See the appendix chapter.

2.3.4 Convergence Analysis: Iteration Complexity

In this section, we do iteration complexity analysis of the algorithm. The iteration complexity analysis is done for both convex and nonconvex case.

Convex Case

When the function $f(\cdot)$ is convex, the overall objective function will become convex; and as a result of Theorem 9, the proposed algorithm converges to the set of global optimal points. Let us make the following assumptions in this subsection:

- The step-size is constant with $\gamma^r = \gamma < \frac{\tau_{\min}}{L_{\nabla f}}$, $\forall r$.
- The level set $\{x \mid h(x) \leq h(x^0)\}$ is compact and the next two assumptions hold in this set.
- The nonsmooth function $g(\cdot)$ is Lipschitz continuous, i.e., $|g(x)-g(y)| \leq L_g\|x-y\|$. This assumption is satisfied in many practical problems such as Lasso or group Lasso.
- The gradient of the approximation function $\tilde{f}_i(\cdot, y)$ is uniformly Lipschitz with constant L_i:
$$\|\nabla_{x_i}\tilde{f}_i(x_i, y) - \nabla_{x_i'}\tilde{f}_i(x_i', y)\| \leq L_i\|x_i - x_i'\|.$$

Lemma 2 (Sufficient Descent) *There exists* $\hat{\beta} > 0$, *such that for all* $r \geq 1$, *we have*

$$\mathbb{E}[h(x^{r+1}) \mid x^r] \leq h(x^r) - \hat{\beta}\|\hat{x}^r - x^r\|^2.$$

Proof The above result is an immediate consequence of (A.35) for the constant choice of step-size with $\hat{\beta} \triangleq \beta\gamma p_{\min}$.

Due to the bounded level set assumption, there must exists constants $Q, R > 0$ such that

$$\|\nabla f(x^r)\| \leq Q, \tag{2.17}$$

$$\|x^r - x^*\| \leq R, \tag{2.18}$$

for all x^r. Next we use the constants Q and R to bound the cost-to-go in the algorithm.

Lemma 3 (Cost-to-go Estimate) *For all $r \geq 1$, we have*

$$\left(\mathbb{E}[h(x^{r+1}) \mid x^r] - h(x^*)\right)^2 \leq 2\left((Q + L_g)^2 + nL^2R^2\right)\|\hat{x}^r - x^r\|^2,$$

for any optimal point x^*, where $L \triangleq \max_i\{L_i\}$.

Proof The proof steps are very similar to the proof of [36][Lemma 3.2]. Let us first bound the conditional expected cost-to-go by

$$\begin{aligned}
\mathbb{E}\left[h(x^{r+1}) - h(x^*) \mid x^r\right] &\overset{\text{(i)}}{\leq} h(x^r) - h(x^*) \\
&= f(x^r) - f(x^*) + g(x^r) - g(x^*) \\
&\overset{\text{(ii)}}{\leq} \langle \nabla f(x^r), x^r - \hat{x}^r\rangle + \langle \nabla f(x^r), \hat{x}^r - x^*\rangle + L_g\|x^r - \hat{x}^r\| + g(\hat{x}^r) - g(x^*) \\
&\overset{\text{(iii)}}{\leq} (L_g + Q)\|\hat{x}^r - x^r\| + \sum_{i=1}^{n}\langle \nabla_{x_i} f(x^r) - \nabla_{x_i}\tilde{f}_i(\hat{x}_i, x^r), \hat{x}_i^r - x_i^*\rangle \\
&\quad + \sum_{i=1}^{n}\langle \nabla_{x_i}\tilde{f}_i(\hat{x}_i^r, x^r), \hat{x}_i^r - x_i^*\rangle + g(\hat{x}^r) - g(x^*) \\
&\leq (L_g + Q)\|\hat{x}^r - x^r\| + \sum_{i=1}^{n}\langle \nabla_{x_i} f(x^r) - \nabla_{x_i}\tilde{f}_i(\hat{x}_i, x^r), \hat{x}_i^r - x_i^*\rangle
\end{aligned} \tag{2.19}$$

where (i) is due to the sufficient decrease bound in Lemma 2; the inequality (ii) is due to the convexity of $f(\cdot)$ and Lipschitz continuity of $g(\cdot)$; the third inequality is due to the definition of the Q. Furthermore, the last inequality is obtained by using the first order optimality condition of the point $\hat{x}_i^r$, i.e., $\langle \nabla_{x_i}\tilde{f}_i(\hat{x}_i^r, x^r), \hat{x}_i^r - x_i^*\rangle + g_i(\hat{x}_i^r) - g_i(x_i^*) \leq 0$.

On the other hand, one can write

$$\begin{aligned}
&\left(\sum_{i=1}^{n}\langle\nabla_{x_i}f(x^r)-\nabla_{x_i}\tilde{f}_i(\hat{x}_i^r,x^r),\hat{x}_i^r-x_i^*\rangle\right)^2\\
&=\left(\sum_{i=1}^{n}\langle\nabla_{x_i}\tilde{f}_i(x_i^r,x^r)-\nabla_{x_i}\tilde{f}_i(\hat{x}_i^r,x^r),\hat{x}_i^r-x_i^*\rangle\right)^2\\
&\leq n\sum_{i=1}^{n}L_i^2\|x_i^r-\hat{x}_i^r\|^2\cdot\|\hat{x}_i^r-x_i^*\|^2\\
&\leq nL^2R^2\|x^r-\hat{x}^r\|^2. \qquad (2.20)
\end{aligned}$$

Combining (2.19) and (2.20) will conclude the proof.

Lemma 2 and Lemma 3 will yield to the iteration complexity bound in the following theorem. The proof steps of this result is the same as the ones in [36] and therefore it is omitted here.

Theorem 10 *Define* $\sigma \triangleq \frac{\hat{\beta}}{2((Q+L_g)^2+nL^2R^2)}$. *Then*

$$\mathbb{E}\left[h(x^r)\right]-h(x^*)\leq\frac{\max\{4\sigma-2,h(x^0)-h(x^*),2\}}{\sigma}\frac{1}{r}.$$

Nonconvex Case

In this subsection we study the iteration complexity of the proposed randomized algorithm for the general nonconvex function $f(\cdot)$. Since in the nonconvex scenario, the iterates may not converge to the global optimum point, the closeness to the optimal solution cannot be considered for the iteration complexity analysis. Instead, inspired by [66] where the size of the gradient of the objective function is used as a measure of optimality, we consider the proximal gradient of the objective as a measure of optimality. More precisely, we define

$$\tilde{\nabla}h(x)=x-\arg\min_{y\in\mathcal{X}}\langle\nabla f(x),y-x\rangle+g(y)+\frac{1}{2}\|y-x\|^2.$$

Clearly, $\tilde{\nabla}h(x)=0$ when x is a stationary point. Moreover, this measure coincides with the gradient of the objective if $g\equiv 0$ and $\mathcal{X}=\mathbb{R}^n$. The following theorem, which studies

the decrease rate of this measure, could be viewed as an iteration complexity analysis of the proposed algorithm.

Theorem 11 *Define T_ϵ to be the first time that $\mathbb{E}[\|\tilde{\nabla} h(x^r)\|^2] \leq \epsilon$. Then $T_\epsilon \leq \frac{\kappa}{\epsilon}$ where $\kappa \triangleq \frac{2(L^2+2L+2)(h(x^0)-h^*)}{\hat{\beta}}$ and $h^* = \min_{x\in\mathcal{X}} h(x)$.*

Proof See the appendix chapter.

Remark 1 *If we define T'_ϵ to be the first time that $\mathbb{E}\left[\|\tilde{\nabla} h(x^r)\|\right] \leq \epsilon$, then Theorem 11 combined with Jensen's inequality implies that $T'_\epsilon = \mathcal{O}\left(\frac{1}{\epsilon^2}\right)$*

2.4 Stochastic Successive Upper-bound Minimization

2.4.1 Algorithm Description and Prior Work

Consider the problem of minimizing the expected value of a cost function parameterized by a random variable. The classical sample average approximation (SAA) method for solving this problem requires minimization of an ensemble average of the objective at each step, which can be expensive. In this dissertation, we propose a stochastic successive upper-bound minimization method (SSUM) which minimizes an *approximate* ensemble average at each iteration. To be more precise, let us consider the optimization problem

$$\begin{aligned} \min \quad & \Big\{ f(x) \triangleq \mathbb{E}_\xi \left[g_1(x,\xi) + g_2(x,\xi) \right] \Big\} \\ \text{s.t.} \quad & x \in \mathcal{X}, \end{aligned} \tag{2.21}$$

where $\mathcal{X}$ is a bounded closed convex set and ξ is a random vector drawn from a set $\Xi \in \mathbb{R}^m$. We assume that the function $g_1 :\ \mathcal{X} \times \Xi \mapsto \mathbb{R}$ is a continuously differentiable (and possibly non-convex) function in x, while $g_2 :\ \mathcal{X} \times \Xi \mapsto \mathbb{R}$ is a convex continuous (and possibly non-smooth) function in x. A classical approach for solving the above optimization problem is the sample average approximation (SAA) method. At each iteration of the SAA method, a new realization of the random vector ξ is obtained and

the optimization variable x is updated by solving

$$\begin{aligned} x^r \in \ & \arg\min \ \frac{1}{r}\sum_{i=1}^{r} g_1(x,\xi^i) + g_2(x,\xi^i) \\ & \text{s.t.} \quad x \in \mathcal{X}. \end{aligned} \tag{2.22}$$

Here $\xi^1, \xi^2, \ldots$ are some independent, identically distributed realizations of the random vector ξ. We refer the readers to [67–71] for the roots of the SAA method and [72–74] for several surveys on SAA.

A main drawback of the SAA method is the complexity of each step. In general, due to the non-convexity and non-smoothness of the objective function, it may be difficult to solve the subproblems (2.22) in the SAA method. This motivates us to consider an inexact SAA method by using an approximation of the function $g(\cdot,\xi)$ in the SAA method (2.22) as follows:

$$\begin{aligned} x^r \ \leftarrow \ & \arg\min_{x} \ \frac{1}{r}\sum_{i=1}^{r} \big(\hat{g}_1(x,x^{i-1},\xi^i) + g_2(x,\xi^i)\big) \\ & \text{s.t.} \quad x \in \mathcal{X}, \end{aligned} \tag{2.23}$$

where $\hat{g}_1(x,x^{i-1},\xi^i)$ is an approximation of the function $g_1(x,\xi^i)$ around the point x^{i-1}. Table 9 summarizes the SSUM algorithm.

Algorithm 9 Stochastic Successive Convex Approximation (SSUM) Algorithm

Find a feasible point $x^0 \in \mathcal{X}$ and set $r = 0$

repeat

$x^r \leftarrow \arg\min_{x\in\mathcal{X}} \ \frac{1}{r}\sum_{i=1}^{r} \big(\hat{g}_1(x,x^{i-1},\xi^i) + g_2(x,\xi^i)\big)$

until some convergence criterion is met

2.4.2 Asymptotic Convergence Analysis

Clearly, the function $\hat{g}_1(x,y,\xi)$ should be related to the original function $g_1(x,\xi)$. Following the successive convex approximation idea from the previous sections, we assume that the approximation function $\hat{g}_1(x,y,\xi)$ satisfies the following conditions.

Assumption A:

Let $\mathcal{X}'$ be an open set containing the set $\mathcal{X}$. Suppose the approximation function $\hat{g}(x, y, \xi)$ satisfies the following

A1- $\hat{g}_1(y, y, \xi) = g_1(y, \xi), \quad \forall\ y \in \mathcal{X},\ \forall\ \xi \in \Xi$

A2- $\hat{g}_1(x, y, \xi) \geq g_1(x, \xi), \quad \forall\ x \in \mathcal{X}',\ \forall\ y \in \mathcal{X},\ \forall\ \xi \in \Xi$

A3- $\hat{g}(x, y, \xi) \triangleq \hat{g}_1(x, y, \xi) + g_2(x, \xi)$ is uniformly strongly convex in x, i.e., for all $(x, y, \xi) \in \mathcal{X} \times \mathcal{X} \times \Xi$,

$$\hat{g}(x + d, y, \xi) - \hat{g}(x, y, \xi) \geq \hat{g}'(x, y, \xi; d) + \frac{\gamma}{2}\|d\|^2, \quad \forall\ d \in \mathbb{R}^n,$$

where $\gamma > 0$ is a constant.

The assumptions A1-A2 imply that the approximation function $\hat{g}_1(\cdot, y, \xi)$ should be a locally tight approximation of the original function $g_1(\cdot, \xi)$. We point out that the above assumptions can be satisfied in many cases by the right choice of the approximation function and hence are not restrictive. For example, the approximation function $\hat{g}_1(\cdot, y, \xi)$ can be made strongly convex easily to satisfy Assumption A3 even though the function $g_1(\cdot, y)$ itself is not even convex; see Section 3 and Section 4 for some examples.

To ensure the convergence of the SSUM algorithm, we further make the following assumptions.

Assumption B:

B1- The functions $g_1(x, \xi)$ and $\hat{g}_1(x, y, \xi)$ are continuous in x for every fixed $y \in \mathcal{X}$ and $\xi \in \Xi$

B2- The feasible set $\mathcal{X}$ is bounded

B3- The functions $g_1(\cdot, \xi)$ and $\hat{g}_1(\cdot, y, \xi)$, their derivatives, and their second order derivatives are uniformly bounded. In other words, there exists a constant $K > 0$

such that for all $(x, y, \xi) \in \mathcal{X} \times \mathcal{X} \times \Xi$ we have

$$\begin{aligned} &|g_1(x,\xi)| \le K, \quad &&\|\nabla_x g_1(x,\xi)\| \le K, \quad &&\|\nabla_x^2 g_1(x,\xi)\| \le K, \\ &|\hat{g}_1(x,y,\xi)| \le K, \quad &&\|\nabla_x \hat{g}_1(x,y,\xi)\| \le K, \quad &&\|\nabla_x^2 \hat{g}_1(x,y,\xi)\| \le K, \end{aligned}$$

B4- The function $g_2(x, \xi)$ is convex in x for every fixed $\xi \in \Xi$

B5- The function $g_2(x, \xi)$ and its directional derivative are uniformly bounded. In other words, there exists $K' > 0$ such that for all $(x, \xi) \in \mathcal{X} \times \Xi$, we have $|g_2(x, \xi)| \le K'$ and

$$|g_2'(x, \xi; d)| \le K'\|d\|, \quad \forall\ d \in \mathbb{R}^n \text{ with } x + d \in \mathcal{X}.$$

B6- Let $\hat{g}(x, y, \xi) = \hat{g}_1(x, y, \xi) + g_2(x, y, \xi)$. There exists $\bar{g} \in \mathbb{R}$ such that

$$|\hat{g}(x, y, \xi)| \le \bar{g}, \quad \forall\ (x, y, \xi) \in \mathcal{X} \times \mathcal{X} \times \Xi.$$

Notice that in the assumptions B3 and B5, the derivatives are taken with respect to the x variable only. Furthermore, one can easily check that the assumption B3 is automatically satisfied if the functions $g_1(x, \xi)$ and $\hat{g}_1(x, y, \xi)$ are continuously second order differentiable with respect to (x, y, ξ) and the set Ξ is bounded; or when $g_1(x, \xi)$ and $\hat{g}_1(x, y, \xi)$ are continuous and second order differentiable in (x, y) and Ξ is finite. As will be seen later, this assumption can be easily satisfied in various practical problems. It is also worth mentioning that since the function $g_2(x, \xi)$ is assumed to be convex in x in B4, its directional derivative with respect to x in B5 can be written as

$$\begin{aligned} g_2'(x, \xi; d) &= \liminf_{t \downarrow 0} \ \frac{g_2(x + td, \xi) - g_2(x, \xi)}{t} \\ &= \inf_{t > 0} \ \frac{g_2(x + td, \xi) - g_2(x, \xi)}{t} \\ &= \lim_{t \downarrow 0} \ \frac{g_2(x + td, \xi) - g_2(x, \xi)}{t}. \end{aligned} \tag{2.24}$$

The following theorem establishes the convergence of the SSUM algorithm.

Theorem 12 *Suppose that Assumptions A and B are satisfied. Then the iterates generated by the SSUM algorithm converge to the set of stationary points of* (2.12) *almost*

surely, i.e.,

$$\lim_{r\to\infty} d(x^r, \mathcal{X}^*) = 0,$$

where $\mathcal{X}^*$ *is the set of stationary points of* (2.12)*.*

To facilitate the presentation of the proof, let us define the random functions

$$\begin{aligned}
f_1^r(x) &\triangleq \frac{1}{r}\sum_{i=1}^{r} g_1(x,\xi^i),\\
f_2^r(x) &\triangleq \frac{1}{r}\sum_{i=1}^{r} g_2(x,\xi^i),\\
\hat{f}_1^r(x) &\triangleq \frac{1}{r}\sum_{i=1}^{r} \hat{g}_1(x,x^{i-1},\xi^i),\\
f^r(x) &\triangleq f_1^r(x) + f_2^r(x),\\
\hat{f}^r(x) &\triangleq \hat{f}_1^r(x) + f_2^r(x),
\end{aligned}$$

for $r = 1, 2, \ldots$. Clearly, the above random functions depend on the realization $\xi^1, \xi^2, \ldots$ and the choice of the initial point x^0. Now we are ready to prove Theorem 12.

Proof First of all, since the iterates $\{x^r\}$ lie in a compact set, it suffices to show that every limit point of the iterates is a stationary point. To show this, let us consider a subsequence $\{x^{r_j}\}_{j=1}^{\infty}$ converging to a limit point $\bar{x}$. Note that since $\mathcal{X}$ is closed, $\bar{x} \in \mathcal{X}$ and therefore $\bar{x}$ is a feasible point. Moreover, since $|g_1(x,\xi)| < K$, $|g_2(x,\xi)| < K'$ for all $\xi \in \Xi$ (due to B3 and B5), using the strong law of large numbers [75], one can write

$$\lim_{r\to\infty} f_1^r(x) = \mathbb{E}\left[g_1(x,\xi)\right] \triangleq f_1(x), \quad \forall\, x \in \mathcal{X}, \tag{2.25}$$

$$\lim_{r\to\infty} f_2^r(x) = \mathbb{E}\left[g_2(x,\xi)\right] \triangleq f_2(x), \quad \forall\, x \in \mathcal{X}. \tag{2.26}$$

Furthermore, due to the assumptions B3, B5, and (2.24), the family of functions

$\{f_1^{r_j}(\cdot)\}_{j=1}^{\infty}$ and $\{f_2^{r_j}(\cdot)\}_{j=1}^{\infty}$ are equicontinuous and therefore by restricting to a subsequence, we have

$$\lim_{j\to\infty} f_1^{r_j}(x^{r_j}) = \mathbb{E}_\xi \left[g_1(\bar{x},\xi)\right], \tag{2.27}$$

$$\lim_{j\to\infty} f_2^{r_j}(x^{r_j}) = \mathbb{E}_\xi \left[g_2(\bar{x},\xi)\right]. \tag{2.28}$$

On the other hand, $\|\nabla_x \hat{g}(x,y,\xi)\| < K,\ \forall\, x,y,\xi$ due to the assumption B3 and therefore the family of functions $\{\hat{f}_1^r(\cdot)\}$ is equicontinuous. Moreover, they are bounded and defined over a compact set; see B2 and B4. Hence the Arzelà–Ascoli theorem [76] implies that, by restricting to a subsequence, there exists a uniformly continuous function $\hat{f}_1(x)$ such that

$$\lim_{j\to\infty} \hat{f}_1^{r_j}(x) = \hat{f}_1(x), \quad \forall\, x \in \mathcal{X}, \tag{2.29}$$

and

$$\lim_{j\to\infty} \hat{f}_1^{r_j}(x^{r_j}) = \hat{f}_1(\bar{x}), \quad \forall\, x \in \mathcal{X}. \tag{2.30}$$

Furthermore, it follows from assumption A2 that

$$\hat{f}_1^{r_j}(x) \geq f_1^{r_j}(x), \quad \forall\, x \in \mathcal{X}'.$$

Letting $j \to \infty$ and using (2.25) and (2.29), we obtain

$$\hat{f}_1(x) \geq f_1(x), \quad \forall\, x \in \mathcal{X}'. \tag{2.31}$$

On the other hand, using the update rule of the SSUM algorithm, one can show the following lemma.

Lemma 4 $\lim_{r\to\infty} \hat{f}_1^r(x^r) - f_1^r(x^r) = 0,$ almost surely.

The proof of Lemma 4 is relegated to the appendix chapter.

Combining Lemma 4 with (2.27) and (2.30) yields

$$\hat{f}_1(\bar{x}) = f_1(\bar{x}). \tag{2.32}$$

It follows from (2.31) and (2.32) that the function $\hat{f}_1(x) - f_1(x)$ takes its minimum value at the point $\bar{x}$ over the open set $\mathcal{X}'$. Therefore, the first order optimality condition implies that

$$\nabla \hat{f}_1(\bar{x}) - \nabla f_1(\bar{x}) = 0,$$

or equivalently

$$\nabla \hat{f}_1(\bar{x}) = \nabla f_1(\bar{x}). \tag{2.33}$$

On the other hand, using the update rule of the SSUM algorithm, we have

$$\hat{f}_1^{r_j}(x^{r_j}) + f_2^{r_j}(x^{r_j}) \leq \hat{f}_1^{r_j}(x) + f_2^{r_j}(x), \quad \forall\, x \in \mathcal{X}.$$

Letting $j \to \infty$ and using (2.28) and (2.30) yield

$$\hat{f}_1(\bar{x}) + f_2(\bar{x}) \leq \hat{f}_1(x) + f_2(x), \quad \forall\, x \in \mathcal{X}. \tag{2.34}$$

Moreover, the directional derivative of $f_2(\cdot)$ exists due to the bounded convergence theorem [75]. Therefore, (2.34) implies that

$$\langle \nabla \hat{f}_1(\bar{x}), d\rangle + f_2'(\bar{x}; d) \geq 0, \quad \forall\, d.$$

Combining this with (2.33), we get

$$\langle \nabla f_1(\bar{x}), d\rangle + f_2'(\bar{x}; d) \geq 0, \quad \forall\, d,$$

or equivalently

$$f'(\bar{x}; d) \geq 0, \quad \forall\, d,$$

which means that $\bar{x}$ is a stationary point of $f(\cdot)$.

Remark 2 *In Theorem 12, we assume that the set $\mathcal{X}$ is bounded. It is not hard to see*

that the result of the theorem still holds even if $\mathcal{X}$ is unbounded, so long as the iterates lie in a bounded set.

To see further non-asymptotic results on the convergence of SSUM method, the readers are referred to the concurrent works [77–79].

2.5 Successive Convex Approximation in Games

2.5.1 Prior Work

The non-cooperative games are essential in modeling the systems where selfish players are maximizing their own objectives. These systems are in nature different from optimization problems in general since the objective of different players might be contradictory to each other. A well-studied concept in these games is the Nash Equilibrium (NE) concept [80], where at the NE point, each player will not be better off by deviating from his/her equilibrium strategy while the other players keep executing their equilibrium strategies. The non-cooperative game modeling has recently become popular in different engineering contexts such as beamforming/power allocation for wireless networks and electricity market pricing. In particular, in the dynamic spectrum management problem, this modeling is popular due to the distributed nature of the system, specially in the cognitive radio scenarios; See two recent surveys [81, 82] and the references therein for more details.

Although the existence of the NE is typically easy to show, finding a convergent algorithm for finding such a NE point is not easy. In particular, the intuitive best response algorithms, which iteratively updates players' variables by the best response strategy, requires conditions on the objectives of the players to converge. Motivated by the classical paper [7], which deals with solving nonlinear equation, many researchers tried to extend it to different scenarios. For example, the references [83–85] consider iterative methods for linear complementary problems [86]. As mentioned in [86, Chapter 5], there are three typical ways of showing the convergence of such iterative methods in games: showing contraction of the iterates, proving the monotonicity of a potential, or establishing the monotonicity of the iterates. The last approach is not applicable to a

wide class of games and hence we do not consider it here. For the contraction analysis, there are recent works relating the contraction of the iterates to the spectral radius of a particular matrix depending on the utilities of the players; see [87–91]. Also when a potential function exists in the game, the convergence of different algorithms could be shown; see, e.g., [92].

When the players objectives are non-convex, the existence of the NE is not guaranteed in general. For such non-convex games, the first order NE (also known as quasi-NE [93]) can be shown to exist easily when the players' objectives are smooth and the constraint sets are compact. In this chapter, we utilize the successive convex approximation idea to find the quasi-NE of the game. The analysis of the algorithm is done based on the contraction of the iterates as well as the existence of a potential in the game.

2.5.2 Problem Statement and Algorithm Description

Consider an n-player game where each player i, $i = 1, 2, \ldots, n$, is interested in solving the following optimization problem:

$$\begin{aligned} \min_{x_i} \quad & \theta_i(x_i, x_{-i}) \\ \text{s.t.} \quad & x_i \in \mathcal{X}_i. \end{aligned} \tag{2.35}$$

Here $\mathcal{X}_i \subseteq \mathbb{R}^{m_i}$ is a closed convex set and $\theta_i(\cdot)$ is a continuous (possibly nonsmooth and nonconvex) function. A simple intuitive approach for solving the above game in a distributed manner is as follows. At each iteration, a (subset) of player try to optimize their own objective assuming the other users' strategy is fixed. More specifically, at iteration r of the algorithm, the players $\{i : i \in \mathcal{S}^r\}$ optimize their own strategy by solving the following subproblem

$$x_i^{r+1} = \arg\min_{x_i \in \mathcal{X}_i} \quad \theta_i(x_i, x_{-i}^r). \tag{2.36}$$

In many practical situations, the function $\theta_i(\cdot, x_{-i}^r)$ might not be convex and hence the update rule (2.36) is not easy to compute. One simple approach is to replace the

function $\theta_i(\cdot, x^r_{-i})$ with its convex approximation $\hat{\theta}_i(x_i, x^r)$ in (2.36). In other words, at iteration r of the algorithm, the users in set $\mathcal{S}^r$ updates their variable by

$$x_i^{r+1} = \arg\min_{x_i \in \mathcal{X}_i} \quad \hat{\theta}_i(x_i, x^r), \forall i \in \mathcal{S}^r \tag{2.37}$$

where $\hat{\theta}_i(\cdot, x^r)$ is an approximation of the function $\theta_i(\cdot, x^r_{-i})$ at the current point x_i^r. To have a concrete algorithm, we need to decide about the choice of the function $\hat{\theta}(\cdot)$ and the choice of the set S^r. Two classical ways of selecting the set S^r is the Gauss-Seidel and the Jacobi. In the rest of this section, we study these two choices separately.

2.5.3 Gauss-Seidel Update Rule

In the Gauss-Seidel choice of the players in the algorithm, at each iteration only one block i is selected to be updated. More precisely, the Gauss-Seidel approach will lead to Algorithm 10 for solving (2.35).

Algorithm 10 Gauss-Seidel Successive Upper-bound Minimization (GS-SUM) Algorithm

find a feasible point $x^0 \in \mathcal{X} \triangleq \mathcal{X}_1 \times \ldots \times \mathcal{X}_n$ and set $r = 0$

repeat

choose an index i

set $x_i^{r+1} \in \arg\min_{x_i \in \mathcal{X}_i} \hat{\theta}_i(x_i, x^r)$

set $x_j^{r+1} = x_j^r, \ \forall\, j \neq i$

set $r = r + 1$

until some convergence criterion is met

Notice that this algorithm is can be viewed as a generalization of the method in [92] where the special case of $\hat{\theta}_i(x_i, y) = \theta_i(x_i, y_{-i}) + \frac{1}{2}\|x_i - y_i\|^2$ is considered.

To study the convergence of the GS-SUM algorithm, we need to have some initial definitions:

- **Generalized potential game:** The introduced n-player game in (2.35) is said to be a *generalized potential game* if there exists a continuous function $P(\cdot) : \mathbb{R}^m \mapsto \mathbb{R}$,

$m = m_1 + \ldots + m_n$, such that for all i, all x_{-i}, and all $y_i, z_i \in \mathcal{X}_i$,

$$\theta_i(y_i, x_{-i}) > \theta_i(z_i, x_{-i})$$

implies

$$P(y_i, x_{-i}) - P(z_i, x_{-i}) \geq \sigma(\theta_i(y_i, x_{-i}) - \theta_i(z_i, x_{-i})),$$

where $\sigma : \mathbb{R}_+ \mapsto \mathbb{R}_+$ is a forcing function, i.e., $\lim_{r\to\infty} \sigma(t^r) = 0 \Rightarrow \lim_{r\to\infty} t^r = 0$.

- **Quasi-Nash equilibrium point:** The point $x^* = (x_i^*)_{i=1}^n$ is a quasi-Nash equilibrium of the game (2.35) if

$$\theta_i'(x^*; d) \geq 0, \ \forall\ d = (0, \ldots, 0, d_i, 0, \ldots, 0) \text{ with } x_i^* + d_i \in \mathcal{X}_i, \ \forall i$$

Let us further make the following assumptions on the approximation function $\hat{\theta}_i(\cdot, \cdot)$:

Assumption 5 *We assume the approximation function satisfies the following assumptions:*

- $\hat{\theta}_i(x_i, y)$ *is continuous in* (x_i, y), $\forall i$
- $\hat{\theta}_i(x_i, x) = \theta(x)$, $\forall x_i \in \mathcal{X}_i$, $\forall i$
- $\hat{\theta}_i(x_i, y) \geq \theta(y_1, \ldots, y_{i-1}, x_i, y_{i+1}, \ldots, y_n)$, $\forall y \in \mathcal{X}$, $\forall x_i \in \mathcal{X}_i$, $\forall i$
- $\hat{\theta}_i'(x_i, y; d_i)\Big|_{x_i = y_i} = \theta_i'(y; d)$, $\forall d = (0, \ldots, 0, d_i, 0, \ldots, 0)$ with $y_i + d_i \in \mathcal{X}_i$, $\forall i$

where in the last inequality, $\hat{\theta}_i'(x_i, y; d_i)$ is the directional derivative of the function $\hat{\theta}_i(\cdot, y)$ in the direction d_i.

Clearly, the choice of the player i in the algorithm could not be arbitrary. For example, if only the first player is updated at all iterations, then there is no chance for the other players to update their variables. Define i_r to be the block/player chosen in the r-th iteration. In this work, we assume the following choices of variables:

- **Essentially cyclic:** We say the choice of the updates in the algorithm is *essentially cyclic* if there exists $T \geq 1$ such that

$$\{i_r + 1, i_r + 2, \ldots, i_r + T\} = \{1, 2, \ldots, n\}, \quad \forall r$$

- **Randomized:** The choice of the updates in the algorithm is *randomized* if the players are chosen randomly at different iterations so that

$$Pr(i_r = j) = p_j > 0, \ \forall j = 1, 2, \ldots, n, \ \forall r = 1, 2, \ldots$$

with $\sum_{j=1}^{n} p_j = 1$.

Having the above assumptions/definitions in our hand, we are now ready to state Theorem 13 which studies the limit points of the GS-SUM algorithm for the essentially cyclic choice of the variables.

Theorem 13 *Assume the game* (2.35) *is generalized potential game and Assumption 5 holds. Let us further assume that the approximation function* $\hat{\theta}_i(\cdot; y)$ *is strictly convex for all fixed* $y \in \mathcal{X}$, $\forall i$*; and the choice of the variables is essentially cyclic. Then every limit point of the iterates generated by the GS-SUM algorithm is a quasi-Nash equilibrium of the game* (2.35)*.*

Proof See the appendix chapter.

Now we will analyze the randomized choice selection in the algorithm. To proceed, we need to define a merit function which is a generalization of the Nikaido-Isoda function [94]. Let $\hat{\theta}_i(\cdot, \cdot)$ be an approximation of the player i's utility which satisfies Assumption 5. Define the QNE measure of point y by $\kappa(y) = \sum_{i=1}^{n} \kappa_i(y)$ where

$$\kappa_i(y) = \hat{\theta}_i(y_i, y) - \min_{x_i \in \mathcal{X}_i} \hat{\theta}_i(x_i, y_i).$$

Clearly, $\kappa(y) \geq 0$, $\forall y \in \mathcal{X}$. Moreover, in the special case of $\hat{\theta}_i(x_i, y) = \theta_i(x_i, y_{-i})$, the above measure will exactly coincide with the Nikaido-Isoda function [94], or the Ky-Fan-function [95]. The following lemma, which can be viewed as a generalization

of [94, Lemma 3.1], sheds light on the applicability of the function $\kappa(\cdot)$ as a measure of being QNE.

Lemma 5 *Assume $-\infty < M \leq \hat{\theta}_i(x_i, y)$ for some M and for all $x_i \in \mathcal{X}_i, y \in \mathcal{X}$, $\forall i$. Then $\kappa(y)$ is a positive continuous function and $\kappa(y) = 0$ if and only if y is a QNE of the game* (2.35).

Proof The continuity of the function κ follows immediately from the continuity of the function $\hat{\theta}_i(\cdot, \cdot)$. Now consider a point y with $\kappa(y) = 0$. Since $\kappa_i(\cdot)$ is a nonnegative function for all i, we must have $\kappa_i(y) = 0$, $\forall i$. Equivalently, $y_i \in \arg\min_{x_i \in \mathcal{X}_i} \hat{\theta}(x_i, y)$. Combining the first order optimality condition and the derivative consistency assumption in Assumption5 implies

$$\theta_i'(y; d) \geq 0, \ \forall \ d = (0, \ldots, 0, d_i, 0, \ldots, 0) \text{ with } y_i + d_i \in \mathcal{X}_i, \ \forall i,$$

which implies that y is a QNE of the game. To prove the converse, we only need to take the above steps in the reverse direction.

Now we are ready to state our simple convergence analysis of the randomized algorithm.

Theorem 14 *Assume the game* (2.35) *is generalized potential game and Assumption 5 holds. Let us further assume that the choice of the variables is randomized at each iteration. Then $\lim_{r \to \infty} \kappa(x^r) = 0$, almost surely.*

Proof Check the appendix chapter.

The following corollary is an immediate consequence of Lemma 5 and Theorem 14.

Corollary 3 *Under the same set of assumptions as in Theorem 14, every limit points of the iterates generated by randomized GS-SUM method is a QNE of the game* (2.35).

Notice that in Theorem 14 there is no requirement on the approximation function to be *strictly* convex or having a unique minimizer at each step. However, this requirement appears in Theorem 13 and it is in fact necessary according to the counterexample by Powell in [38]. To the best of our knowledge, this result is new, even in the optimization context.

2.5.4 Jacobi Update Rule:

In the Jacobi selection of the players, all players update their variables in parallel at each iteration; in other words, $\mathcal{S}^r = \{1, \ldots, n\}$, $\forall r$. More precisely, the algorithm in

Algorithm 11 Jacobi Successive Upper-bound Minimization (J-SUM) Algorithm

find a feasible point $x^0 \in \mathcal{X} \triangleq \mathcal{X}_1 \times \ldots \times \mathcal{X}_n$ and set $r = 0$
repeat
  for $i = 1, \ldots, n$ **do**
    set $x_i^{r+1} \in \arg\min_{x_i \in \mathcal{X}_i} \hat{\theta}_i(x_i, x^r)$
  set $r = r + 1$
until some convergence criterion is met

In this section we assume that the cost function $\theta_i(\cdot)$ has the following form

$$\theta_i(x) = f_i(x) + g_i(x_i), \tag{2.38}$$

where the function $f_i(x)$ is smooth (possibly nonconvex) and the function $g_i(x_i)$ is a convex nonsmooth function. To study the convergence of the algorithm, we further need to make the following assumption.

Assumption 6 *Assume the approximation function satisfies the followings:*

- $\hat{\theta}_i(x_i, y) = \hat{f}_i(x_i, y) + g_i(x_i)$
- $\hat{f}_i(x_i, y)$ *is twice continuously differentiable function in* (x_i, y), $\forall i$
- $\hat{f}_i(x_i, x) = f_i(x)$, $\forall x_i \in \mathcal{X}_i$, $\forall i$
- $\nabla_{x_i} \hat{f}_i(x_i, y)\Big|_{x_i = y_i} = \nabla_{x_i} f_i(x_i, y_{-i})\Big|_{x_i = y_i}$, $\forall y \in \mathcal{X}$, $\forall i$
- $\hat{f}_i(x_i, y)$ *is uniformly strongly convex in* x_i *for all* i. *In other words, for any* i, *there exists* $\tau_i > 0$ *such that for any* $w_i, x_i \in \mathcal{X}_i$ *and* $y \in \mathcal{X}$, *we have*

$$\hat{f}_i(w_i, y) \geq \hat{f}_i(x_i, y) + \langle w_i - x_i, \nabla_{x_i} \hat{f}_i(x_i, y) \rangle + \frac{\tau_i}{2} \|x_i - w_i\|^2.$$

Similar to the classical convergence analysis of the Jacobi method based on the contraction argument [85, Chapter 3], we can obtain the following theorem:

Theorem 15 *Define*

$$\Gamma \triangleq \begin{bmatrix} \frac{\gamma_{11}}{\tau_1} & \frac{\gamma_{12}}{\tau_1} & \cdots & \frac{\gamma_{1n}}{\tau_1} \\ \frac{\gamma_{21}}{\tau_2} & \frac{\gamma_{22}}{\tau_2} & \cdots & \frac{\gamma_{2n}}{\tau_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\gamma_{n1}}{\tau_n} & \frac{\gamma_{n2}}{\tau_n} & \cdots & \frac{\gamma_{nn}}{\tau_n} \end{bmatrix},$$

where $\gamma_{ij} \triangleq \sup_{x_i,y} \| \left(\nabla_{y_j} \nabla_{x_i} \hat{f}(x_i, y) \right) \|_2$. *If* $\|\Gamma\|_2 < 1$, *then the J-SUM method in Algorithm 11 converges linearly to the unique QNE of the game* (2.35).

Proof See the appendix chapter.

It is worth noticing that the sufficient condition in Theorem 15 is more general than the one in [96, Chapter 12]. In fact, choosing the special approximation function $\hat{f}(x_i, y) = f(x_i, y_{-i}) + \frac{1}{2}\|x_i - y_i\|^2$ will yield to the bound in [96].

Randomized Jacobi Update:

In many practical scenarios, the number of available processor is less than the number of blocks. This motivates the use of Jacobi update rule over a subset of blocks at each iteration. In this subsection, we consider a randomized Jacobi update rule where at each iteration a random subset of players update their variables using the approximation function. Algorithm 12 describes the algorithm in details.

Algorithm 12 Randomized Jacobi Successive Upper-bound Minimization (RJ-SUM) Algorithm

find a feasible point $x^0 \in \mathcal{X} \triangleq \mathcal{X}_1 \times \ldots \times \mathcal{X}_n$ and set $r = 0$

repeat

choose a random subset $\mathcal{S}^r$ of players, i.e., $\mathcal{S}^r \subseteq \{1, 2, \ldots, n\}$

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

if $i \in \mathcal{S}^r$, set $x_i^{r+1} \in \arg\min_{x_i \in \mathcal{X}_i} \hat{\theta}_i(x_i, x^r)$

else set $x_i^{r+1} = x_i^r$

set $r = r + 1$

until some convergence criterion is met

Let us define R_i^r to be a Bernoulli random variable demonstrating the selection of the i-th block at iteration r, i.e., $R_i^r = 1$ iff $i \in \mathcal{S}^r$. Let us further assume that $\mathbb{E}(R_i^r) = p_i$. Similar to Theorem 15, we can have the following convergence result.

Theorem 16 *Define* $\Phi \triangleq \text{diag}(p_1, \ldots, p_n)$ *and* $\Upsilon \triangleq \mathbf{I} - \Phi + \Phi\Gamma$. *If* $\|\Upsilon\|_2 < 1$, *then* x^r *converges to a QNE almost surely. Moreover,* $\mathbf{E}[\|x^r - x^*\|]$ *converges linearly to zero, where* x^* *is the unique QNE of the problem.*

Proof Using the inequality (A.95), one can write

$$\begin{aligned}\mathbb{E}\left[\|x_i^{r+1} - x_i^*\| \mid x^r\right] &= p_i\|\hat{x}_i(x^r) - x_i^*\| + (1 - p_i)\|x_i^r - x_i^*\| \\ &\leq \frac{p_i}{\tau_i}\sum_{j=1}^{n}\gamma_{ij}\|x_j^r - x_j^*\| + (1 - p_i)\|x_i - x_i^*\|.\end{aligned}$$

Taking the expectation with respect to the whole sample space and using the definition of Υ, we obtain

$$\mathbb{E}\left[\|x_i^{r+1} - x_i^*\|\right] \leq \sum_j \Upsilon_{ij}\mathbb{E}\left[\|x_i^r - x_i^*\|\right],$$

which by writing in the matrix form implies

$$\begin{bmatrix} \mathbb{E}\left[\|x_1^{r+1} - x_1^*\|\right] \\ \mathbb{E}\left[\|x_2^{r+1} - x_2^*\|\right] \\ \vdots \\ \mathbb{E}\left[\|x_n^{r+1} - x_n^*\|\right] \end{bmatrix} \leq \begin{bmatrix} \Upsilon_{11} & \Upsilon_{12} & \dots & \Upsilon_{1n} \\ \Upsilon_{21} & \Upsilon_{22} & \dots & \Upsilon_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \Upsilon_{n1} & \Upsilon_{n2} & \dots & \Upsilon_{nn} \end{bmatrix} \begin{bmatrix} \mathbb{E}\left[\|x_1^r - x_1^*\|\right] \\ \mathbb{E}\left[\|x_2^r - x_2^*\|\right] \\ \vdots \\ \mathbb{E}\left[\|x_n^r - x_n^*\|\right] \end{bmatrix}. \tag{2.39}$$

Hence, for any i,

$$\mathbb{E}\left[\|x_i^r - x_i^*\|\right] \leq \sqrt{\sum_j \left(\mathbb{E}[\|x_j^r - x_j^*\|]\right)^2} \leq (\|\Upsilon\|_2)^r \sqrt{\sum_j \|x_j^0 - x_j^*\|^2}.$$

If $\|\Upsilon\|_2 < 1$, then we have a linear convergence of the sequence $\mathbb{E}\left[\|x_i^r - x_i^*\|\right]$ to zero. Moreover, the simple use of Markov's inequality implies

$$Prob\left(\|x_i^r - x_i^*\| > \epsilon\right) \leq \frac{\|\Upsilon\|_2^r \sqrt{\sum_j \|x_j^0 - x_j^*\|^2}}{\epsilon},$$

for any $\epsilon > 0$ and hence, the simple application of Borel-Cantelli lemma [97, 98] implies the almost sure convergence of x^r to x^*.

Chapter 3

Applications

In this chapter we will see different applications of the successive convex approximation idea on various practical problems.

3.1 Interference Management in Wireless Heterogenous Networks

The design of future wireless cellular networks is on the verge of a major paradigm change. With the proliferation of multimedia rich services as well as smart mobile devices, the demand for wireless data has been increased explosively in recent years. In order to accommodate the explosive demand for wireless data, the cell size of cellular networks is shrinking by deploying more transmitters such as macro/micro/pico/femto base stations and relays. These nodes utilize the same frequency bands, and are densely deployed to provide coverage extension for cell edge and indoor users (see Figure 3.1). Deploying more transmitters brings the transmitters and receivers closer to each other, thus we are able to provide high link quality with low transmission power [99, 100].

Unfortunately, close proximity of many transmitters and receivers introduces substantial intracell and intercell interference, which, if not properly managed, can significantly affect the system performance. In the context of multiuser cellular networks, the intracell (resp. intercell) interference refers to the interference generated from the same access point/transmitter (resp. different access points/transmitters). This huge

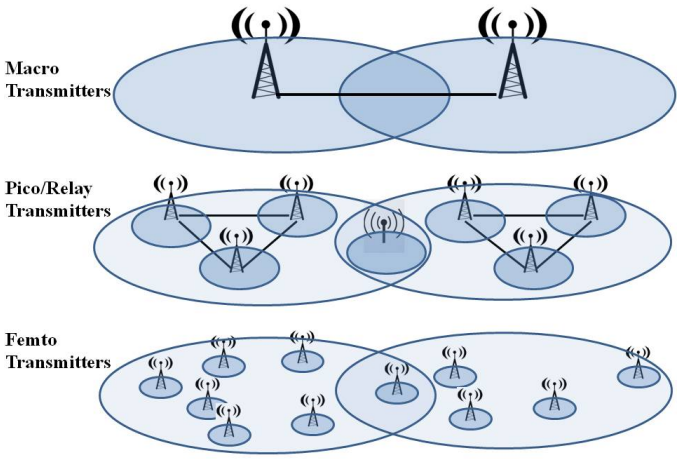


Figure 3.1: The dense structure of the new cellular networks.

amount of interference caused by resource sharing among the nodes cannot be handled by traditional ways of interference management methods such as time division multiple access, frequency division multiple access, or space division multiple access. In fact, interference is the major performance limiting factor for the modern dense cellular networks. The key challenge for interference management in the new wireless networks is to develop low-complexity schemes that mitigates the multiuser interference in the system, optimally balance the overall spectrum efficiency and user fairness. This chapter deals with various theoretical and practical aspects of interference management for multiuser cellular networks. In particular, we study the interference management in the physical and MAC layer using optimized beamforming and scheduling techniques. We utilize the successive convex approximation idea to develop algorithms for this purpose. A special consideration is given to practical issues such as parallel implementation, overhead reduction, channel estimation error, and channel aging.

3.1.1 Prior Work

Consider a MIMO interfering Broadcast Channel (IBC) in which a number of transmitters, each equipped with multiple antennas, wish to simultaneously send independent data streams to their intended receivers. As a generic model for multi-user downlink communication, MIMO-IBC can be used in the study of many practical systems such

as Digital Subscriber Lines (DSL), Cognitive Radio systems, ad-hoc wireless networks, wireless cellular communication, to name just a few. Unfortunately, despite the importance and years of intensive research, the search for optimal transmit/receive strategies that can maximize the weighted sum-rate of all users in a MIMO-IBC remains rather elusive. In fact, even for the simpler case of MIMO interference channel, the optimal strategy is still unknown. This lack of understanding of the capacity region has motivated a pragmatic approach whereby we simply treat interference as noise and maximize the weighted sum-rate by searching within the class of linear transmit/receive strategies.

Transceiver design of *Interference Channel* (IFC), which is a special case of IBC, has been a topic of intensive research in recent years. From the optimization's perspective, this problem is nonconvex and NP-hard even in the single antenna case [101]. Thus, most current research efforts have been focused on finding a high quality sub-optimal solution efficiently. For example, the works [102] and [103] proposed iterative algorithms for solving a general smooth utility maximization and the min-SINR maximization problems, respectively. The *interference-pricing* game method is another sum-utility maximization method which strives to reach a stationary point of the weighted sum-rate maximization problem. In contrast, the other game theoretic methods (e.g., [87, 104, 105]) can only find a Nash equilibrium solution, typically yielding a suboptimal sum-rate. For the SISO-IFC, an interference pricing game is proposed in [106] along with an asynchronous distributed algorithm to solve it. It is shown in [106] that the algorithm converges for a set of utility functions, which unfortunately does not include the basic Shannon rate function $\log(1 + \text{SINR})$. In [107], Shi *et al.* have modified the method in [106] and proposed a new algorithm for the SISO and MISO interference channel that can monotonically converge to a stationary point of the weighted sum-rate maximization problem. A similar algorithm for the MIMO-IFC is considered in [108] for the single data stream case. However, these algorithms only allow one user to update its power or beamformer at each time, which entails a large communication overhead needed to exchange price information. A general distributed pricing algorithm that allows simultaneous user update is proposed in [109] for the MIMO-IFC in the single stream case. Regarding the convergence of such methods, reference [106] has established the convergence of the interference pricing algorithm to a stationary point for a set of utility functions, which unfortunately does not include the standard Shannon rate function. Several extensions

and variations of the interference pricing algorithm have been proposed [107] for the SISO and MISO IFC that can monotonically converge to a stationary point of the weighted sum-rate maximization problem. A similar algorithm for the MIMO interference channel was proposed in [108] without considering multiplexing (i.e., one data stream per user). All of these algorithms allow only one user to update its beamformer at a time, which may lead to excessive communication overhead for price exchanges. A general distributed interference pricing algorithm that allows multiple users to update simultaneously was proposed in [109] for the MIMO interference channel with no multiplexing, although no convergence analysis is provided for the algorithm.

By fixing the receiver structure to any of the standard linear receivers (e.g., the MMSE or Zero-Forcing receivers), we can reduce the linear transceiver design to a transmit covariance matrix design problem. Reference [110] proposed an iterative algorithm based on the gradient projection method for the transmit covariance matrix design problem. The algorithm allows each user to update its own covariance matrix locally, provided that the channel state information and the covariance matrices of other users can be gathered. Based on a local linear approximation, reference [111] proposed a distributed algorithm which lets each user update its own covariance matrix by solving a convex optimization problem. This algorithm can be viewed as the MIMO extension version of the sequential distributed pricing algorithm in [107]. We henceforth unify the name of these algorithms as the *iterative linear approximation* (ILA) algorithm. Moreover, since these algorithms use a local tight concave lower bound approximation of the weighted sum-rate objective function, they ensure that the rates increase monotonically and that the transmit covariance matrices converge to a stationary point of the original objective function (i.e., the weighted sum-rate) [107, 112].

A different sum-rate maximization approach was proposed in [113] for the MIMO broadcast downlink channel, where the weighted sum-rate maximization problem is transformed to an equivalent weighted sum MSE minimization (WMMSE) problem with some specially chosen weight matrices that depend on the optimal beamforming matrices. Since the weight matrices are generally unknown, the authors of [113] proposed an iterative algorithm that adaptively chooses the weight matrices and updates the linear transmit/receive beamformers at each iteration. A nonconvex cost function was constructed [113] and shown to monotonically decrease as the algorithm progresses. But

the convergence of the iterates to a stationary point (or the global minimum) of the cost function has not been studied. A similar algorithm has been proposed in [114] for the interference channel where each user only transmits one data stream. Interestingly, the approach in [113] is a special for of the BSUM algorithm (introduced in the first chapter of this dissertation), and its convergence is guaranteed by BSUM framework.

Inspired by the work of [113,114] and utilizing the BSUM framework, we first propose a simple distributed linear transceiver design method, named the WMMSE algorithm, for general utility maximization in an interfering broadcast channel. This algorithm extends the existing algorithms of [113] and [114] in several directions. In particular, it can handle fairly general utility functions (which includes weighted sum-rate utility function as a special case), and works for general MIMO interfering broadcast channel (which includes MIMO broadcast channel [113] and MISO interference channel [114] as special cases). Theoretically, we show that the sequence of iterates generated by the WMMSE algorithm converges to at least a local optima of the utility maximization problem, and does so with low communication and computational complexity.

In the second subsection, we consider a joint user grouping and beamformer design problem. Throughout, the term "grouping" (or "scheduling") refers to the process of assigning users to a fixed number of time/frequency slots. In this terminology, the users that are served in the same time/frequency slot are considered as one group. In our formulation, each user is optimally scheduled to a subset of time/frequency slots (not necessarily just one slot), while its linear transceiver is simultaneously optimized across the slots. Our formulation captures all the important performance factors into a single comprehensive formulation, without any ad-hoc combination of multi-stage formulations. Using the developed WMMSE algorithm [1, 113–116], we propose an algorithm to solve this joint user grouping and beamformer design problem. This is a special case of BSUM framework and is guaranteed to converge to at least a stationary point of the original joint user grouping and transceiver design problem. Moreover, we can extend our algorithm and its convergence to further optimize the amount of time allocated across different groups. The proposed algorithm exhibits fast convergence and is amenable to distributed implementation. The simulation results in the next chapter show that the proposed formulation/algorithm can offer significantly higher system throughput than the standard multi-user MIMO techniques, while still respecting user

fairness.

In subsection 3.1.4, instead of sum utility maximization, we consider the max-min utility function, i.e., the maximization of the worst user rate. Providing max-min fairness has long been considered as an important design criterion for wireless networks. Hence various algorithms that optimize the min-rate utility in different network settings have been proposed in the literature. References [117, 118] are early works that studied the max-min signal to interference plus noise ratio (SINR) power control problem and a related SINR feasibility problem in a scalar interference channel (IC). It was shown in [117, 118] that for randomly generated scalar ICs, with probability one there exists a unique optimal solution to the max-min problem. The proposed algorithm with an additional binary search can be used to solve the max-min fairness problem efficiently. Recently reference [119] derived a set of algorithms based on nonlinear Perron-Frobenius theory for the same network setting. Differently from [117,118], the proposed algorithms can also deal with individual users' power constraints.

Apart from the scalar IC case, there have been many published results [102,120–125] on the min rate maximization problem in a multiple input single output (MISO) network, in which the BSs are equipped with multiple antennas and the users are only equipped with a single antenna. Reference [120] utilized the nonnegative matrix theory to study the related power control problem when the beamformers are known and fixed. When optimizing the transmit power and the beamformers jointly, the corresponding min-rate utility maximization problem is non-convex. Despite the lack of convexity, the authors of [121] showed that a semidefinite relaxation is tight for this problem, and the optimal solution can be constructed from the solution to a reformulated semidefinite program. Furthermore, the authors of [122] showed that this max-min problem can be solved by a sequence of second order cone programs (SOCP). Reference [125] identified an interesting uplink downlink duality property, in which the downlink min-rate maximization problem can be solved by alternating between a downlink power update and a uplink receiver update. In a related work [123], the authors made an interesting observation that in a single cell MISO network, the global optimum of this problem can be obtained by solving a (simpler) weighted sum inverse SINR problem with a set of appropriately chosen weights. However, this observation is only true when the receiver noise is negligible. The authors of [124] extended their early results [119] to the

MISO setting with a single BS and multiple users. A fixed-point algorithm that alternates between power update and beamformer updates was proposed, and the nonlinear Perron-Frobenius theory was applied to prove the convergence of the algorithm.

Unlike the MISO case, the existing work on the max-min problem for MIMO networks is rather limited; see [124] and [103]. Both of these studies consider a MIMO network in which a *single stream* is transmitted for each user. In particular, the author of [103] showed that finding the global optimal solution for this problem is intractable (NP-hard) when the number of antennas at each transmitter/receiver is at least *three*. They then proposed an efficient algorithm that alternates between updating the transmit and the receive beamformers to find a local optimal solution. The key observation is that when the users' receive beamformers are fixed, finding the set of optimal transmit beamformers can be again reduced to a sequence of SOCP and solved efficiently. In [124], an algorithm that updates the transmit beamformers and the receive beamformers in an alternating fashion is proposed. The global convergence of the proposed method is shown for the special cases of rank one channels and low SNR region. For more discussion of the max-min and its related resource allocation problems in interfering wireless networks, we refer the readers to a recent survey [126].

Here we consider optimization of the minimum rate user and first we show that in the considered general setting, when there are at least *two* antennas at each transmitters and the receivers, the min-rate maximization problem is NP-hard in the number of users. This result is a generalization of that presented in [103], in which the NP-hardness results require more than *three* antennas at the users and BSs. We further provide a reformulation of the original max-min problem by generalizing the WMMSE/BSUM framework, and design an algorithm that computes an approximate solution to the max-min problem. The proposed algorithm has the following desirable features: *i)* itis computationally efficient, as in each step a convex optimization problem whose solution can be obtained easily in a closed form is solved; *ii)* it is guaranteed to converge to a stationary solution of the original problem by the discussed convergence analysis of the successive convex approximation method.

Despite the intensive aforementioned research on the weighted sum rate maximization problem, most of the proposed methods require the perfect and complete channel state information (CSI) of all links–an assumption which is impractical due to channel

aging and channel estimation errors. More importantly, obtaining the complete CSI for all links usually requires a large amount of system overhead which is prohibitive for practical implementation. Using robust optimization techniques, different algorithms have been proposed to address this issue [127–131]. The robust optimization methods are in general designed for the worst case scenarios and therefore, due to their nature, are suboptimal when the worst cases happen with small probability. An alternative approach is to design the transceivers by optimizing the *average performance* using a stochastic optimization framework. Unfortunately, few algorithms [132, 133] have been devised using this approach, partly due to various technical challenges related to the computation of the objective function and its derivatives.

In subsection 3.1.5, we propose a simple stochastic iterative optimization algorithm for solving the ergodic sum rate maximization problem. Our approach is based on the SSUM framework and unlike the previous approach of [132] which maximizes a lower bound of the expected weighted sum rate problem, our work directly maximizes the ergodic sum rate, and is guaranteed to converge to the set of stationary points of the ergodic sum rate maximization problem. For each link of the IC, our proposed algorithm requires either the channel statistics, or the actual CSI. Moreover, our approach can adapt easily to situations when the channel statistics change over time. Although presented for sum rate maximization in an inference channel, our algorithm and its convergence can be easily extended to other system utilities and more general channel models such as interfering broadcast (IBC) networks.

3.1.2 Beamformer Design in Multi-user Wireless Networks

Consider a K cell interfering broadcast channel where the base station k, $k = 1, 2, \ldots, K$, is equipped with M_k transmit antennas and serves I_k users in cell k. Let us define i_k to be the i-th user in cell k and N_{i_k} be the number of receive antennas at receiver i_k. Let us also define $\mathcal{I}$ to be the set of all receivers, i.e.,

$$\mathcal{I} = \{i_k \mid k \in \{1, 2, \ldots, K\}, \;\; i \in \{1, 2, \ldots, I_k\}\}\,.$$

Let $\mathbf{V}_{i_k} \in \mathbb{C}^{M_k \times d_{i_k}}$ denote the beamformer that base station k uses to transmit the signal $\mathbf{s}_{i_k} \in \mathbb{C}^{d_{i_k} \times 1}$ to receiver i_k, $i = 1, 2, \ldots, I_k$, i.e.,

$$\mathbf{x}_k = \sum_{i=1}^{I_k} \mathbf{V}_{i_k} \mathbf{s}_{i_k},$$

where we assume $\mathbb{E}\left[\mathbf{s}_{i_k} \mathbf{s}_{i_k}^H\right] = \mathbf{I}$. Assuming a linear channel model, the received signal $\mathbf{y}_{i_k} \in \mathbb{C}^{N_{i_k} \times 1}$ at receiver i_k can be written as

$$\mathbf{y}_{i_k} = \underbrace{\mathbf{H}_{i_k k} \mathbf{V}_{i_k} \mathbf{s}_{i_k}}_{\text{desired signal}} + \underbrace{\sum_{m=1, m \neq i}^{I_k} \mathbf{H}_{i_k k} \mathbf{V}_{m_k} \mathbf{s}_{m_k}}_{\text{intracell interference}} + \underbrace{\sum_{j \neq k, j=1}^{K} \sum_{\ell=1}^{I_j} \mathbf{H}_{i_k j} \mathbf{V}_{\ell_j} \mathbf{s}_{\ell_j} + \mathbf{n}_{i_k}}_{\text{intercell interference plus noise}}, \ \forall\ i_k \in \mathcal{I}$$

where matrix $\mathbf{H}_{i_k j} \in \mathbb{C}^{N_{i_k} \times M_j}$ represents the channel from the transmitter j to receiver i_k, while $\mathbf{n}_{i_k} \in \mathbb{C}^{N_k \times 1}$ denotes the additive white Gaussian noise with distribution $\mathcal{CN}(0, \sigma_{i_k}^2 \mathbf{I})$. We assume that the signals for different users are independent from each other and from receiver noises. In this part, we treat interference as noise and consider linear receive beamforming strategy so that the estimated signal is given by

$$\hat{\mathbf{s}}_{i_k} = \mathbf{U}_{i_k}^H \mathbf{y}_{i_k}, \quad \forall\ i_k \in \mathcal{I}.$$

Then, the problem of interest is to find the transmit and receive beamformers[1] $\{\mathbf{V}, \mathbf{U}\}$ such that a certain utility of the system is maximized, while the power budget of each transmitter is respected:

$$\sum_{i=1}^{I_k} \mathrm{Tr}(\mathbf{V}_{i_k} \mathbf{V}_{i_k}^H) \leq P_k,$$

where P_k denotes the power budget of transmitter k.

In what follows, we consider the popular sum-rate utility function and apply the block successive upper-bound framework to the optimization problem.

[1] The notation $\mathbf{V}$ is short for $\{\mathbf{V}_{i_k}\}_{i_k \in \mathcal{I}}$, which denotes all variables $\mathbf{V}_{i_k}$ with $i_k \in \mathcal{I}$.

Weighted Sum-Rate Maximization and a Matrix-Weighted Sum-MSE Minimization

A popular utility maximization problem is the weighted sum-rate maximization which can be written as

$$\begin{aligned} \max_{\{\mathbf{V}_{i_k}\}} \quad & \sum_{k=1}^{K}\sum_{i=1}^{I_k} \alpha_{i_k} R_{i_k} \\ \text{s.t.} \quad & \sum_{i=1}^{I_k} \mathrm{Tr}(\mathbf{V}_{i_k}\mathbf{V}_{i_k}^H) \le P_k, \ \ \forall\ k = 1, 2, \ldots, K, \end{aligned} \tag{3.1}$$

where R_{i_k} is the rate of user i_k which can be written as

$$R_{i_k} \triangleq \log\det\left(\mathbf{I}+\mathbf{H}_{i_k k}\mathbf{V}_{i_k}\mathbf{V}_{i_k}^H\mathbf{H}_{i_k k}^H\left(\sum_{(\ell,j)\neq(i,k)}\mathbf{H}_{i_k j}\mathbf{V}_{\ell_j}\mathbf{V}_{\ell_j}^H\mathbf{H}_{i_k j}^H+\sigma_{i_k}^2\mathbf{I}\right)^{-1}\right). \tag{3.2}$$

The weight α_{i_k} is used to represent the priority of user i_k in the system.

Another popular utility maximization problem for MIMO-IBC is sum-MSE minimization. Under the independence assumption of $\mathbf{s}_{i_k}$'s and $\mathbf{n}_{i_k}$'s, the MSE matrix $\mathbf{E}_{i_k}$ can be written as,

$$\begin{aligned} \mathbf{E}_{i_k} &\triangleq \mathbb{E}_{\mathbf{s},\mathbf{n}}\left[(\hat{\mathbf{s}}_{i_k}-\mathbf{s}_{i_k})(\hat{\mathbf{s}}_{i_k}-\mathbf{s}_{i_k})^H\right] \\ &= (\mathbf{I}-\mathbf{U}_{i_k}^H\mathbf{H}_{i_k k}\mathbf{V}_{i_k})(\mathbf{I}-\mathbf{U}_{i_k}^H\mathbf{H}_{i_k k}\mathbf{V}_{i_k})^H + \sum_{(\ell,j)\neq(i,k)}\mathbf{U}_{i_k}^H\mathbf{H}_{i_k j}\mathbf{V}_{\ell_j}\mathbf{V}_{\ell_j}^H\mathbf{H}_{i_k j}^H\mathbf{U}_{i_k}+\sigma_{i_k}^2\mathbf{U}_{i_k}^H\mathbf{U}_{i_k}, \end{aligned} \tag{3.3}$$

and the sum-MSE minimization problem for the MIMO-IBC can be written as

$$\begin{aligned} \min_{\{\mathbf{U}_{i_k},\mathbf{V}_{i_k}\}} \quad & \sum_{k=1}^{K}\sum_{i=1}^{I_k}\mathrm{Tr}(\mathbf{E}_{i_k}) \\ \text{s.t.} \quad & \sum_{i=1}^{I_k}\mathrm{Tr}(\mathbf{V}_{i_k}\mathbf{V}_{i_k}^H) \le P_k, \quad k = 1, 2, \ldots, K. \end{aligned} \tag{3.4}$$

Fixing all the transmit beamformers $\{\mathbf{V}_{i_k}\}$ and minimizing (weighted) sum-MSE lead

to the well known MMSE receiver:

$$\mathbf{U}_{i_k}^{\text{mmse}} = \mathbf{J}_{i_k}^{-1}\mathbf{H}_{i_k k}\mathbf{V}_{i_k}, \tag{3.5}$$

where $\mathbf{J}_{i_k} \triangleq \sum_{j=1}^{K}\sum_{\ell=1}^{I_j}\mathbf{H}_{i_k j}\mathbf{V}_{\ell_j}\mathbf{V}_{\ell_j}^H\mathbf{H}_{i_k j}^H + \sigma_{i_k}^2\mathbf{I}$ is the covariance matrix of the total received signal at receiver i_k. Using this MMSE receiver, the corresponding MSE matrix is given by

$$\mathbf{E}_{i_k}^{\text{mmse}} = \mathbf{I} - \mathbf{V}_{i_k}^H\mathbf{H}_{i_k k}^H\mathbf{J}_{i_k}^{-1}\mathbf{H}_{i_k k}\mathbf{V}_{i_k}. \tag{3.6}$$

The following result establishes the equivalence between the weighted sum-rate maximization problem and a matrix-weighted sum-MSE minimization problem.

Theorem 17 *The rate of user i_k in* (3.2) *can also be represented as*

$$R_{i_k} = \max_{\mathbf{U}_{i_k},\mathbf{W}_{i_k}} \log\det\left(\mathbf{W}_{i_k}\right) - \mathrm{Tr}\left(\mathbf{W}_{i_k}\mathbf{E}_{i_k}\right) + d_{i_k}, \tag{3.7}$$

where $\mathbf{E}_{i_k}$ is the MSE value of user i_k given by (3.3) *and $\mathbf{W}_{i_k} \in \mathbb{C}^{d_{i_k}\times d_{i_k}}$ is an auxiliary optimization variable.*

Proof First, by checking the first order optimality condition of (3.7) with respect to $\mathbf{U}_{i_k}$, we get

$$\mathbf{W}_{i_k}^H\left(\mathbf{J}_{i_k}\mathbf{U}_{i_k}^* - \mathbf{H}_{i_k k}\mathbf{V}_{i_k}\right) = 0,$$

which yields to the optimum MMSE receiver $\mathbf{U}_{i_k}^* = \mathbf{U}_{i_k}^{\text{mmse}} = \mathbf{J}_{i_k}^{-1}\mathbf{H}_{i_k k}\mathbf{V}_{i_k}$ where $\mathbf{J}_{i_k} = \sigma_{i_k}^2\mathbf{I} + \sum_{\ell_j\in\mathcal{I}}\mathbf{H}_{i_k j}\mathbf{V}_{\ell_j}\mathbf{V}_{\ell_j}^H\mathbf{H}_{i_k j}^H$. By plugging in the optimal value $\mathbf{U}_{i_k}^*$ in (3.3), we obtain $\mathbf{E}_{i_k}^{\text{mmse}} = \mathbf{I} - \mathbf{V}_{i_k}^H\mathbf{H}_{i_k k}^H\mathbf{J}_{i_k}^{-1}\mathbf{H}_{i_k k}\mathbf{V}_{i_k}$. Hence plugging $\mathbf{E}_{i_k}^{\text{mmse}}$ in (3.7) yields

$$\begin{aligned}&\max_{\mathbf{U}_{i_k},\mathbf{W}_{i_k}} \quad \log\det\left(\mathbf{W}_{i_k}\right) - \mathrm{Tr}\left(\mathbf{W}_{i_k}\mathbf{E}_{i_k}\right) + d_{i_k}\\ &= \max_{\mathbf{W}_{i_k}} \quad \log\det\left(\mathbf{W}_{i_k}\right) - \mathrm{Tr}\left(\mathbf{W}_{i_k}\mathbf{E}_{i_k}^{\text{mmse}}\right) + d_{i_k}.\end{aligned} \tag{3.8}$$

The first order optimality condition of (3.8) with respect to $\mathbf{W}_{i_k}$ implies $\mathbf{W}_{i_k}^* = \left(\mathbf{E}_{i_k}^{\text{mmse}}\right)^{-1}$.

By plugging in the optimal $\mathbf{W}_{i_k}^*$ in (3.8), we can write

$$\begin{aligned}
&\max_{\mathbf{U}_{i_k},\mathbf{W}_{i_k}} \log\det\left(\mathbf{W}_{i_k}\right) - \mathrm{Tr}\left(\mathbf{W}_{i_k}\mathbf{E}_{i_k}\right) + d_{i_k} \\
&= -\log\det(\mathbf{E}_{i_k}^{\mathrm{mmse}}) \\
&= -\log\det\left(\mathbf{I} - \mathbf{H}_{i_k k}\mathbf{V}_{i_k}\mathbf{V}_{i_k}^H\mathbf{H}_{i_k k}^H\mathbf{J}_{i_k}^{-1}\right) \\
&= \log\det\left(\mathbf{J}_{i_k}\left(\mathbf{J}_{i_k} - \mathbf{H}_{i_k k}\mathbf{V}_{i_k}\mathbf{V}_{i_k}^H\mathbf{H}_{i_k k}^H\right)^{-1}\right),
\end{aligned}$$

which is the rate of user i_k in (3.2).

Combining Theorem 17 and the Danskin's theorem [13] implies that the function $d_{i_k} - \mathrm{Tr}(\mathbf{W}_{i_k}\mathbf{E}_{i_k}) + \log\det(\mathbf{W}_{i_k})$ can be viewed as a local lower-bound of R_{i_k} after fixing the the value of $\mathbf{W}$ and $\mathbf{V}$ to the current optimum values. By Theorem 17, we only need to solve the approximation function instead of the original function at each iteration. Interestingly, at each iteration, the update of transmit beamformers $\{\mathbf{V}_{i_k}\}$ for all i_k can be decoupled across transmitters, resulting in the following optimization problem:

$$\begin{aligned}
\min_{\{\mathbf{V}_{i_k}\}_{i=1}^{I_k}} \quad & \sum_{i=1}^{I_k} \mathrm{Tr}(\alpha_{i_k}\mathbf{W}_{i_k}(\mathbf{I} - \mathbf{U}_{i_k}^H\mathbf{H}_{i_k k}\mathbf{V}_{i_k})(\mathbf{I} - \mathbf{U}_{i_k}^H\mathbf{H}_{i_k k}\mathbf{V}_{i_k})^H) \\
& + \sum_{i=1}^{I_k}\sum_{(\ell,j)\neq(i,k)} \mathrm{Tr}(\alpha_{\ell_j}\mathbf{W}_{\ell_j}\mathbf{U}_{\ell_j}^H\mathbf{H}_{\ell_j k}\mathbf{V}_{i_k}\mathbf{V}_{i_k}^H\mathbf{H}_{\ell_j k}^H\mathbf{U}_{\ell_j}^H) \qquad (3.9) \\
\text{s.t.} \quad & \sum_{i=1}^{I_k} \mathrm{Tr}(\mathbf{V}_{i_k}\mathbf{V}_{i_k}^H) \leq P_k.
\end{aligned}$$

This is a convex quadratic optimization problem which can be solved by using standard convex optimization algorithms. In fact, this problem also has a closed form solution using the Lagrange multipliers method. Specifically, attaching a Lagrange multiplier μ_k to the power budget constraint of transmitter k, we get the following Lagrange function:

$$\begin{aligned}
&L(\{\mathbf{V}_{i_k}\}_{i=1}^{I_k}, \mu_k) \triangleq \sum_{i=1}^{I_k} \mathrm{Tr}\left(\alpha_{i_k}\mathbf{W}_{i_k}(\mathbf{I} - \mathbf{U}_{i_k}^H\mathbf{H}_{i_k k}\mathbf{V}_{i_k})(\mathbf{I} - \mathbf{U}_{i_k}^H\mathbf{H}_{i_k k}\mathbf{V}_{i_k})^H\right) \\
&+ \sum_{i=1}^{I_k}\sum_{(\ell,j)\neq(i,k)} \mathrm{Tr}\left(\alpha_{\ell j}\mathbf{W}_{\ell_j}\mathbf{U}_{\ell_j}^H\mathbf{H}_{\ell_j k}\mathbf{V}_{i_k}\mathbf{V}_{i_k}^H\mathbf{H}_{\ell_j k}^H\mathbf{U}_{\ell_j}^H\right) + \mu_k(\sum_{i=1}^{I_k}\mathrm{Tr}(\mathbf{V}_{i_k}\mathbf{V}_{i_k}^H) - P_k).
\end{aligned}$$

The first order optimality condition of $L(\{\mathbf{V}_{i_k}\}_{i=1}^{I_k}, \mu_k)$ with respect to each $\mathbf{V}_{i_k}$ yields

$$\mathbf{V}_{i_k}^{\text{opt}} = \left(\sum_{j=1}^{K} \sum_{\ell=1}^{I_j} \alpha_{\ell j} \mathbf{H}_{\ell_j k}^H \mathbf{U}_{\ell_j} \mathbf{W}_{\ell_j} \mathbf{U}_{\ell_j}^H \mathbf{H}_{\ell_j k} + \mu_k \mathbf{I} \right)^{-1} \alpha_{i_k} \mathbf{H}_{i_k k}^H \mathbf{U}_{i_k} \mathbf{W}_{i_k}, \quad i = 1, \ldots, I_k, \tag{3.10}$$

where $\mu_k \geq 0$ should be chosen such that the complementarity slackness condition of the power budget constraint is satisfied. Let $\mathbf{V}_{i_k}(\mu_k)$ denote the right-hand side of (3.10). When the matrix
$\sum_{j=1}^{K} \sum_{\ell=1}^{I_j} \alpha_{\ell_j} \mathbf{H}_{\ell_j k}^H \mathbf{U}_{\ell_j} \mathbf{W}_{\ell_j} \mathbf{U}_{\ell_j}^H \mathbf{H}_{\ell_j k}$ is invertible and $\sum_{i=1}^{I_k} \text{Tr}(\mathbf{V}_{i_k}(0)\mathbf{V}_{i_k}(0)^H) \leq P_k$, then $\mathbf{V}_{i_k}^{\text{opt}} = \mathbf{V}_{i_k}(0)$, otherwise we must have

$$\sum_{i=1}^{I_k} \text{Tr}\left(\mathbf{V}_{i_k}(\mu_k)\mathbf{V}_{i_k}(\mu_k)^H\right) = P_k \tag{3.11}$$

which is equivalent to

$$\text{Tr}\left((\mathbf{\Lambda} + \mu_k \mathbf{I})^{-2} \mathbf{\Phi}\right) = P_k \tag{3.12}$$

where $\mathbf{D}\mathbf{\Lambda}\mathbf{D}^H$ is the eigen-decomposition of $\sum_{j=1}^{K} \sum_{\ell=1}^{I_j} \mathbf{H}_{\ell_j k}^H \mathbf{U}_{\ell_j} \bar{\mathbf{W}}_{\ell_j} \mathbf{U}_{\ell_j}^H \mathbf{H}_{\ell_j k}$ and
$\mathbf{\Phi} = \mathbf{D}^H \left(\sum_{i=1}^{I_k} \mathbf{H}_{i_k k}^H \mathbf{U}_{i_k} \bar{\mathbf{W}}_{i_k}^2 \mathbf{U}_{i_k}^H \mathbf{H}_{i_k k} \right) \mathbf{D}$. Let $[\mathbf{X}]_{mm}$ denote the m-th diagonal element of $\mathbf{X}$, then (3.12) can be simplified as

$$\sum_{m=1}^{M_k} \frac{[\mathbf{\Phi}]_{mm}}{([\mathbf{\Lambda}]_{mm} + \mu_k)^2} = P_k. \tag{3.13}$$

Note that the optimum μ_k (denoted by μ_k^*) must be positive in this case and the left hand side of (3.13) is a decreasing function in μ_k for $\mu_k > 0$. Hence, (3.13) can be easily solved using one dimensional search techniques (e.g., bisection method). Finally, by plugging μ_k^* in (3.10), we get the solution for $\mathbf{V}_{i_k}(\mu_k^*)$, for all $i = 1, \ldots, I_k$.

Therefore, applying the BSUM framework to the sum rate maximization problem will result in the WMMSE algorithm, whic is summarized in Algorithm 13.

Algorithm 13 WMMSE algorithm (μ_k^* is determined by a bisection method)

Initialize $\mathbf{V}_{i_k}$'s such that $\mathrm{Tr}\left(\mathbf{V}_{i_k}\mathbf{V}_{i_k}^H\right) = \frac{p_k}{I_k}$

repeat

$\mathbf{W}'_{i_k} \leftarrow \mathbf{W}_{i_k}, \quad \forall\, i_k \in \mathcal{I}$

$\mathbf{U}_{i_k} \leftarrow \left(\sum_{(j,\ell)} \mathbf{H}_{i_k j}\mathbf{V}_{\ell_j}\mathbf{V}_{\ell_j}^H\mathbf{H}_{i_k j}^H + \sigma_{i_k}^2\mathbf{I}\right)^{-1}\mathbf{H}_{i_k k}\mathbf{V}_k,\ \forall\, i_k \in \mathcal{I}$

$\mathbf{W}_{i_k} \leftarrow \left(\mathbf{I} - \mathbf{U}_{i_k}^H\mathbf{H}_{i_k k}\mathbf{V}_{i_k}\right)^{-1},\ \forall\, i_k \in \mathcal{I}$

$x^{r+1} \leftarrow x^r$

$$\mathbf{V}_{i_k} \leftarrow \alpha_{i_k}\left(\sum_{(j,\ell)}\alpha_{\ell_j}\mathbf{H}_{\ell_j k}^H\mathbf{U}_{\ell_j}\mathbf{W}_{\ell_j}\mathbf{U}_{\ell_j}^H\mathbf{H}_{\ell_j k} + \mu_k^*\mathbf{I}\right)^{-1}\mathbf{H}_{i_k k}^H\mathbf{U}_{i_k}\mathbf{W}_{i_k},\ \forall\, i_k$$

until $\left|\sum_{(j,\ell)}\log\det\left(\mathbf{W}_{\ell_j}\right) - \sum_{(j,\ell)}\log\det\left(\mathbf{W}'_{\ell_j}\right)\right| \le \epsilon$

Note that the convergence of the WMMSE algorithm (to the set of stationary points) is guaranteed using the BSUM convergence result (Theorem 2). It is also worth noting that the BSUM framework has been extensively used for resource allocation in wireless networks, for example [107, 134–137], and [108]. However, the convergence of most of the algorithms was not rigorously established.

Distributed Implementation and Complexity Analysis

For the purpose of distributed implementation, we make two reasonable assumptions (similar to [138]). First, we assume that local channel state information is available for each user, namely, each transmitter k knows the local channel matrices $\mathbf{H}_{\ell_j k}$ to all receivers ℓ_j. The second assumption is that each receiver has an additional channel to feedback information (e.g., the updated beamformers or equivalent information) to the transmitters. Under these two assumptions, the WMMSE algorithm can be implemented in a distributed fashion. More specifically, each receiver i_k locally estimates the received signal covariance matrix $\mathbf{J}_{i_k}$ and updates the matrices $\mathbf{U}_{i_k}$ and $\mathbf{W}_{i_k}$. Then, it feeds back the updated $\mathbf{W}_{i_k}$ and $\mathbf{U}_{i_k}$ to the transmitters. Note that, to reduce communication overhead, user i_k only needs to feedback either the upper triangular part of the matrix $\alpha_{i_k}\mathbf{U}_{i_k}\mathbf{W}_{i_k}\mathbf{U}_{i_k}^H$ or the decomposition $\hat{\mathbf{U}}_{i_k}$ where $\hat{\mathbf{U}}_{i_k}\hat{\mathbf{U}}_{i_k}^H = \alpha_{i_k}\mathbf{U}_{i_k}\mathbf{W}_{i_k}\mathbf{U}_{i_k}^H$ (depending on the relative size of N_{i_k} and d_{i_k}). It should be pointed out that the termination criterion in Algorithm 13 may not be suitable for distributed implementation.

In practice, we suggest setting a maximum number of iterations for the algorithm or simply just do one step of the algorithm within each packet.

Note that the ILA algorithm [107, 111] allows only one user to update its transmit covariance matrix at each iteration. When one user updates its variables, each user must compute $(K-1)$ prices [107] or gradient matrices [111] for other users and then broadcast them within the network. In contrast, the WMMSE algorithm allows simultaneous update among all users since the updating steps are decoupled across users when any of the two variables in $(\{\mathbf{W}_{i_k}\}, \{\mathbf{U}_{i_k}\}, \{\mathbf{V}_{i_k}\})$ are fixed. Therefore, the WMMSE algorithm requires less CSI exchange within the network. For simplicity of complexity analysis, let $\kappa \triangleq |\mathcal{I}|$ be the total number of users in the system and T, R denote the number of antennas at each transmitter and receiver respectively. Also, since both the WMMSE algorithm and the ILA algorithm include a bisection step which generally takes few iterations, we ignore this bisection step in the complexity analysis. Under these assumptions, each iteration of the ILA algorithm involves only the computation of the price matrices in [111] (i.e., $\mathbf{A}_i$'s in equation (10) of [111]). To determine the price matrices in the ILA algorithm, we need to first calculate the covariance matrix of interference at all users and then compute their sum, yielding a complexity of $\mathcal{O}(\kappa^2)$ per user. As a result, the per-iteration complexity of the ILA algorithm is $\mathcal{O}(\kappa^3 T^2 R + \kappa^3 R^2 T + \kappa^2 R^3)$. By a similar analysis, the per-iteration complexity of the WMMSE algorithm can be shown to be $\mathcal{O}(\kappa^2 T R^2 + \kappa^2 R T^2 + \kappa^2 T^3 + \kappa R^3)$. Here an iteration of the WMMSE or the ILA algorithm means one round of updating all users' beamformers or covariance matrices.

3.1.3 Joint Beamforming and Scheduling in Multi-user Networks

Consider the wireless system described in subsection 3.1.2. Assume that we group the users into G groups, with different groups served in an orthogonal manner. In this way, when a base station serves users in one group, it causes no interference to the users in other groups. For example, these G groups may represent different time slots so that the users in group g ($g \in \mathcal{G} \triangleq \{1, 2, \ldots, G\}$) are served in the time slot g. Furthermore, we assume that the channel matrices remain constant while different groups are served. In addition, to keep the decoding and encoding process simple, we assume no correlated signaling across different groups. Under these assumptions and considering linear channel model between the transceivers, the received signal of user i_k in group/time slot g

can be written as

$$\mathbf{y}_{i_k}^g = \underbrace{\mathbf{H}_{i_k k}\mathbf{x}_{i_k}^g}_{\text{desired signal}} + \underbrace{\sum_{\ell\neq i,\ell=1}^{I_k} \mathbf{H}_{i_k k}\mathbf{x}_{\ell_k}^g}_{\text{intracell interference}} + \underbrace{\sum_{j\neq k,j=1}^{K}\sum_{\ell=1}^{I_j} \mathbf{H}_{i_k j}\mathbf{x}_{\ell_j}^g + \mathbf{n}_{i_k}^g}_{\text{intercell interference plus noise}},\ \forall i_k \in \mathcal{I},$$

where $\mathbf{x}_{i_k}^g \in \mathbb{R}^{M_k\times 1}$ and $\mathbf{y}_{i_k}^g \in \mathbb{R}^{N_{i_k}\times 1}$ are respectively the transmitted and received signal of user i_k while it is served in group g. The matrix $\mathbf{H}_{i_k j} \in \mathbb{R}^{N_{i_k}\times M_j}$ represents the channel response from the transmitter j to receiver i_k, while $\mathbf{n}_{i_k}^g \in \mathbb{R}^{N_{i_k}\times 1}$ denotes the additive white Gaussian noise with distribution $\mathcal{N}(0, \sigma_{i_k,g}^2\mathbf{I})$. We assume that the signals of different users are independent of each other and the noise. Moreover, we restrict ourselves to linear beamforming strategies where base station k deploys a beamformer $\mathbf{V}_{i_k}^g \in \mathbb{R}^{M_k\times d_{i_k}}$ to modulate d_{i_k} number of data stream for user i_k in group g, while user i_k estimates the transmitted signal in group g using a linear beamforming matrix $\mathbf{U}_{i_k}^g \in \mathbb{R}^{N_{i_k}\times d_{i_k}}$. That is, we have

$$\mathbf{x}_{i_k}^g = \mathbf{V}_{i_k}^g \mathbf{s}_{i_k}^g, \qquad \hat{\mathbf{s}}_{i_k}^g = \mathbf{U}_{i_k}^{g\ T}\mathbf{y}_{i_k}^g,$$

where $\mathbf{s}_{i_k} \in \mathbb{R}^{d_{i_k}\times 1}$ is the data vector of user i_k with a normalized power $\mathbb{E}[\mathbf{s}_{i_k}\mathbf{s}_{i_k}^T] = \mathbf{I}$. Note that d_{i_k} is the number of data streams of user i_k and should be no more than the number of antennas at the transmitter and receiver side.

Let us define the group association variables $\{\alpha_{i_k}^g\}$ where $\alpha_{i_k}^g \in \{0,1\}$ is a binary variable with $\alpha_{i_k}^g = 1$ signifying the user i_k is served in group g. Since the receiver of user i_k only receives the signal in the associated time slot/group, the rate of user i_k when it is served in group g is given by

$$\mathcal{R}_{i_k}^g = \alpha_{i_k}^g \log\det\Bigg(\mathbf{I} + \mathbf{H}_{i_k k}\mathbf{V}_{i_k}^g(\mathbf{V}_{i_k}^g)^T\mathbf{H}_{i_k k}^T\Bigg(\sigma_{i_k}^2\mathbf{I} + \sum_{(j,\ell)\neq(k,i)} \mathbf{H}_{i_k j}\mathbf{V}_{\ell_j}^g(\mathbf{V}_{\ell_j}^g)^T\mathbf{H}_{i_k j}^T\Bigg)^{-1}\Bigg). \tag{3.14}$$

Let β_g denote the fraction of the resources allocated to the users in group g. For example, if we serve different users in different time slots (TDMA), then β_g denotes the fraction of time that is allocated to the users in group g. With an appropriate normalization, we can assume that $\sum_{g=1}^{G} \beta_g = 1$. Under these assumptions, the rate of user i_k is the weighted sum of the rates that it can get in each group, i.e., $\mathcal{R}_{i_k} = \sum_{g=1}^{G} \beta_g \mathcal{R}_{i_k}^g$. Employing a system utility function $\mathcal{U}(\cdot)$, we are led to the following joint user grouping and transceiver design problem:

$$\begin{aligned} \max_{\boldsymbol{\alpha},\boldsymbol{\beta},\mathbf{V}} \quad & \mathcal{U}\left(\{\mathcal{R}_{i_k}\}_{i_k \in \mathcal{I}}\right) \\ \text{s.t.} \quad & \sum_{i=1}^{I_k} \mathrm{Tr}\left(\mathbf{V}_{i_k}^g \mathbf{V}_{i_k}^{g\,T}\right) \le P_k, \ \ \forall\, k \in \mathcal{K}, \ \forall\, g \in \mathcal{G} \\ & \sum_{g=1}^{G} \beta_g = 1, \ \beta_g \ge 0, \quad \forall\, g \in \mathcal{G} \\ & \alpha_{i_k}^g \in \{0,1\}, \quad \forall i_k \in \mathcal{I}, \ \ \forall\, g \in \mathcal{G}, \end{aligned} \tag{3.15}$$

where $R_{i_k}^g$ is defined by (3.14).

In many cases, the utility function $\mathcal{U}(\cdot)$ can be decomposed as the sum of utilities of individual users. If so, the optimization problem (3.15) can be rewritten as

$$\begin{aligned} \max_{\boldsymbol{\alpha},\mathbf{V},\boldsymbol{\beta}} \quad & \sum_{k=1}^{K} \sum_{i=1}^{I_k} u_{i_k} \left(\sum_{g=1}^{G} \beta_g \mathcal{R}_{i_k}^g \right) \\ \text{s.t.} \quad & \sum_{i=1}^{I_k} \mathrm{Tr}\left(\mathbf{V}_{i_k}^g \mathbf{V}_{i_k}^{g\,T}\right) \le P_k, \ \ \forall\, k \in \mathcal{K}, \ \forall\, g \in \mathcal{G} \\ & \sum_{g=1}^{G} \beta_g = 1, \ \beta_g \ge 0, \quad \forall\, g \in \mathcal{G} \\ & \alpha_{i_k}^g \in \{0,1\}, \quad \forall\, i_k \in \mathcal{I}, \ \ \forall\, g \in \mathcal{G}. \end{aligned} \tag{3.16}$$

One of the major difficulties in handling the above problem is the presence of discrete variables $\{\alpha_{i_k}^g\}$. We can overcome this difficulty by the following observation: if the utility function $u_{i_k}(\cdot)$ is non-decreasing for all $i_k \in \mathcal{I}$, then there exists an optimal solution of (3.16) for which $\alpha_{i_k}^g = 1, \ \forall\ i_k \in \mathcal{I}, \ \forall\ g \in \mathcal{G}$. The reason is because by

increasing the value of α_{i_k} from zero to one, the objective value will not decrease in (3.16). Using this simple observation, we can set $\alpha_{i_k}^g = 1$ for all i_k and g and solve the following equivalent optimization problem:

$$\begin{aligned} \max_{\mathbf{V},\boldsymbol{\beta}} \quad & \sum_{k=1}^{K}\sum_{i=1}^{I_k} u_{i_k}\left(\sum_{g=1}^{G}\beta_g R_{i_k}^g\right) \\ \text{s.t.} \quad & \sum_{i=1}^{I_k}\mathrm{Tr}\left(\mathbf{V}_{i_k}^g \mathbf{V}_{i_k}^{g\ T}\right) \le P_k, \ \ \forall\, k\in\mathcal{K},\ \forall\, g\in\mathcal{G} \\ & \sum_{g=1}^{G}\beta_g = 1,\ \beta_g \ge 0, \quad \forall\, g\in\mathcal{G}, \end{aligned} \tag{3.17}$$

where $R_{i_k} = \sum_{g=1}^{G} R_{i_k}^g$ with

$$R_{i_k}^g = \log\det\left(\mathbf{I} + \mathbf{H}_{i_k k}\mathbf{V}_{i_k}^g(\mathbf{V}_{i_k}^g)^T\mathbf{H}_{i_k k}^T\left(\sigma_{i_k}^2\mathbf{I} + \sum_{(j,\ell)\neq(k,i)}\mathbf{H}_{i_k j}\mathbf{V}_{\ell_j}^g(\mathbf{V}_{\ell_j}^g)^T\mathbf{H}_{i_k j}^T\right)^{-1}\right).$$

After solving (3.17) and obtaining the optimal solution, we can group/schedule the users by simply checking the optimal value $\{\mathbf{V}_{i_k}^{g*}\}$. In particular, the following simple rule can be used:

$$\alpha_{i_k}^g = \begin{cases} 1 & \text{if} \quad \|\mathbf{V}_{i_k}^{g*}\| > 0, \\ 0 & \text{if} \quad \|\mathbf{V}_{i_k}^{g*}\| = 0. \end{cases}$$

In practice, due to rounding errors and in order to reduce the transmission complexity, one may use a relaxed rule to group users:

$$\alpha_{i_k}^g = \begin{cases} 1 & \text{if} \quad \|\mathbf{V}_{i_k}^{g*}\| > \epsilon, \\ 0 & \text{if} \quad \|\mathbf{V}_{i_k}^{g*}\| \le \epsilon, \end{cases} \tag{3.18}$$

where ϵ is a suitable small number. After adjusting the variables $\{\alpha_{i_k}^g\}$, one can reduce the transmission process complexity by the update rule $\mathbf{V}_{i_k}^g = \alpha_{i_k}^g\mathbf{V}_{i_k}^g$. Notice that by doing so, we reduce the transmission complexity by not transmitting in the groups with very small gains.

It is important to note that in our formulation (3.17) each user can be served in more than one group. This is in contrast to the traditional orthogonal partitioning based user scheduling/grouping whereby each user is to be served in only one time slot. We provide below two simple examples to illustrate the benefits of our new user grouping formulation. In both of these two examples, the harmonic mean maximization problem is considered: max $\mathcal{U}(\{\mathcal{R}_{i_k}\}) \triangleq \frac{|\mathcal{I}|}{\sum_{i_k \in \mathcal{I}} \mathcal{R}_{i_k}^{-1}}$. Notice that although this utility function is not decomposable across the users, the equivalent formulation max $-\sum_{i_k \in \mathcal{I}} \mathcal{R}_{i_k}^{-1}$ is decomposable across the different users.

Example 1 (Grouping vs. no grouping) Consider a SISO system with one base station serving two users. The channels to the users are given by

$$H_{1_1 1} = H_{2_1 1} = 1.$$

Assume the noise power $\sigma^2 = 1$ and the power budget $P_1 = 1$. Consider the harmonic mean objective function:

$$\mathcal{U}(\mathcal{R}_1, \mathcal{R}_2) = \frac{2}{\mathcal{R}_1^{-1} + \mathcal{R}_2^{-1}}.$$

If no grouping is allowed, i.e., $G = 1$, the maximum system utility is

$$\mathcal{U}(\mathcal{R}_1, \mathcal{R}_2) = \frac{2}{\mathcal{R}_1^{-1} + \mathcal{R}_2^{-1}} = \frac{2}{2(\log \frac{4}{3})^{-1}} = \log \frac{4}{3}.$$

which is achieved at $V_1 = V_2 = \sqrt{0.5}$. On the other hand, if grouping is allowed, by putting each user in one group, the classical TDMA approach results in the harmonic mean of

$$\mathcal{U}(\mathcal{R}_1, \mathcal{R}_2) = \frac{2}{\mathcal{R}_1^{-1} + \mathcal{R}_2^{-1}} = \frac{2}{2(\frac{1}{2}\log 2)^{-1}} = \frac{1}{2} > \log \frac{4}{3} \approx 0.415.$$

Therefore, our user grouping strategy can improve the overall system performance. This example shows that we can broaden our design space by introducing the grouping variables; and therefore one can achieve performance gain by grouping the users. As we will see in the simulation section, this gain is substantial for practical systems.

Example 2 (Multiple groups per user vs. single group per user) Consider a

system with two cells. The first cell is similar to the one in Example 1, i.e., base station 1 serves two users with channels given in Example 1. The second base station serves one user with channel $H_{1_2 2} = 1$. We also assume no inter-cell interference, i.e.,

$$H_{i_k j} = 0, \quad \forall\, j \neq k,\ \forall\, i \in \mathcal{I}_k.$$

Assume the harmonic mean utility function $\mathcal{U}(\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3) = \frac{3}{\mathcal{R}_1^{-1} + \mathcal{R}_2^{-1} + \mathcal{R}_3^{-1}}$ is used. In this example, user 1_2 can be served in all time slots/groups without causing interference to the other two users in cell 1. This example sheds light on why our nonorthogonal user grouping method can yield a higher system utility than the partitioning based orthogonal user grouping.

Our goal in the rest of this subsection is to design an efficient algorithm to solve (3.17). To facilitate the presentation of ideas, we first consider in Section 3.1.3 a fixed value of $\boldsymbol{\beta}$ and present the ideas for this case. Then, in Section 3.1.3, we consider the grouping and time allocation problem by treating $\boldsymbol{\beta}$ as an optimization variable.

Joint User Grouping and Beamformer Design

In this section, we use the ideas behind the WMMSE algorithm to develop a joint user grouping and beamformer design algorithm for the case when the fraction of resources time allocated to each group is fixed, i.e., $\beta_g = 1/G,\ \forall\, g \in \mathcal{G}$. The goal is to maximize the system throughput while considering fairness in the system. More specifically, we are interested in solving (3.17) for a fixed value of $\beta_g = \frac{1}{G}$, i.e.,

$$\begin{aligned} \max_{\mathbf{V}} \quad & \sum_{k=1}^{K} \sum_{i=1}^{I_k} u_{i_k} \left(\sum_{g=1}^{G} \frac{1}{G} R_{i_k}^{g} \right) \\ \text{s.t.} \quad & \sum_{i=1}^{I_k} \mathrm{Tr}\left(\mathbf{V}_{i_k}^{g} \mathbf{V}_{i_k}^{g\ T} \right) \leq P_k, \ \ \forall\, k \in \mathcal{K},\ \forall\, g \in \mathcal{G}. \end{aligned} \tag{3.19}$$

It is worth noting the differences between the single group formulation in subsection 3.1.2 and the multi-group formulation (3.19). In particular, in the multi-group formulation different rates of the same user are summed in the utility function and hence the utility

function is not decomposable across different groups. This difference makes the algorithm design and the ensuing analysis significantly more challenging than those in the single group case. Let us define $\mathbf{E}_{i_k}^g$ to be the MMSE value of user i_k when it is served in group g, i.e.,

$$\begin{aligned}\mathbf{E}_{i_k}^g &\triangleq \mathbb{E}_{\mathbf{s},\mathbf{n}}\left[(\hat{\mathbf{s}}_{i_k}^g - \mathbf{s}_{i_k}^g)(\hat{\mathbf{s}}_{i_k}^g - \mathbf{s}_{i_k}^g)^T\right] \\ &= (\mathbf{I} - (\mathbf{U}_{i_k}^g)^T \mathbf{H}_{i_k k} \mathbf{V}_{i_k}^g)(\mathbf{I} - (\mathbf{U}_{i_k}^g)^T \mathbf{H}_{i_k k} \mathbf{V}_{i_k}^g)^T \\ &+ \sum_{(\ell,j)\neq(i,k)} (\mathbf{U}_{i_k}^g)^T \mathbf{H}_{i_k j} \mathbf{V}_{\ell_j}^g (\mathbf{V}_{\ell_j}^g)^T \mathbf{H}_{i_k j}^T \mathbf{U}_{i_k}^g + \sigma_{i_k}^2 (\mathbf{U}_{i_k}^g)^T \mathbf{U}_{i_k}^g.\end{aligned}$$

Using the relation between the rate and the MSE value, the rate of user i_k can be written as

$$R_{i_k} = \sum_{g=1}^{G} \frac{1}{G} R_{i_k}^g = \max_{\mathbf{U}} \quad -\sum_{g=1}^{G} \frac{1}{G} \log\det\left(\mathbf{E}_{i_k}^g\right).$$

Assuming $u_{i_k}(\cdot)$ is an increasing function of R_{i_k}, one can rewrite the optimization problem (3.19) as

$$\begin{aligned}\max_{\mathbf{U},\mathbf{V}} \quad & \sum_{k=1}^{K} \sum_{i=1}^{I_k} u_{i_k}\left(-\sum_{g=1}^{G} \frac{1}{G} \log\det \mathbf{E}_{i_k}^g\right) \\ \text{s.t.} \quad & \sum_{i=1}^{I_k} \mathrm{Tr}\left(\mathbf{V}_{i_k}^g \mathbf{V}_{i_k}^{g\,T}\right) \leq P_k, \;\; \forall\, k \in \mathcal{K}, \; \forall\, g \in \mathcal{G}.\end{aligned} \tag{3.20}$$

The following lemma, whose proof is relegated to the appendix, is the key step in reformulating (3.20) as an equivalent higher dimensional optimization problem which is amenable to block coordinate minimization.

Lemma 6 *Let $f_i : \mathbb{R}^{m_i} \mapsto \mathbb{R}$, $i = 1, 2, \ldots, n$, be strictly concave and twice continuously differentiable functions. Furthermore, assume the mappings $\mathbf{h}_i : \mathbb{R}^p \mapsto \mathbb{R}^{m_i}$, $i = 1, 2, \ldots, n$, are continuously differentiable. Then, the mapping $\nabla f_i(\cdot)$ is invertible*

for all i, and the optimization problem

$$\begin{aligned} \min_{\mathbf{x}} \quad & \sum_{i=1}^{n} f_i(\mathbf{h}_i(\mathbf{x})) \\ \text{s.t.} \quad & \mathbf{x} \in \mathbb{X}, \end{aligned} \tag{3.21}$$

is equivalent to

$$\begin{aligned} \min_{\mathbf{x},\mathbf{y}} \quad & \sum_{i=1}^{n} \left(\mathbf{y}_i^T \mathbf{h}_i(\mathbf{x}) + f_i\left(\boldsymbol{\phi}_i(\mathbf{y}_i)\right) - \mathbf{y}_i^T \boldsymbol{\phi}_i(\mathbf{y}_i) \right) \\ \text{s.t.} \quad & \mathbf{x} \in \mathbb{X}, \end{aligned} \tag{3.22}$$

where $\boldsymbol{\phi}_i(\cdot) : \mathbb{R}^{m_i} \mapsto \mathbb{R}^{m_i}$ is the inverse map of the gradient map $\nabla f_i(\cdot)$. Moreover, the objective function of (3.22) *is convex with respect to each $\mathbf{y}_i$. If in addition, we assume that the set $\mathbb{X}$ is convex, then there is an one-to-one correspondence between the set of stationary points of* (3.21) *and* (3.22)*. In other words, $\mathbf{x}^*$ is a stationary point of* (3.21) *if and only if $(\mathbf{x}^*, \mathbf{y}^*)$ is a stationary point of* (3.22) *where $\mathbf{y}_i^* = \nabla f(\mathbf{h}_i^*)$ with $\mathbf{h}_i^* = \mathbf{h}_i(\mathbf{x}^*)$.*

Let us assume that $c_{i_k} = -u_{i_k}\left(-\sum_{g=1}^{G} \frac{1}{G} \log\det(\mathbf{E}_{i_k}^g)\right)$ is strictly concave in $\mathbf{E}_{i_k} \triangleq (\mathbf{E}_{i_k}^1, \mathbf{E}_{i_k}^2, \ldots, \mathbf{E}_{i_k}^G)$. According to Lemma 6, one can rewrite (3.20) as

$$\begin{aligned} \min_{\mathbf{U},\mathbf{V},\mathbf{W}} \quad & \sum_{k=1}^{K} \sum_{i=1}^{I_k} \left[\mathrm{Tr}\left(\mathbf{W}_{i_k} \mathbf{E}_{i_k}\right) + c_{i_k}\left(\boldsymbol{\gamma}_{i_k}(\mathbf{W}_{i_k})\right) - \mathrm{Tr}\left(\mathbf{W}_{i_k} \boldsymbol{\gamma}_{i_k}(\mathbf{W}_{i_k})\right) \right] \\ \text{s.t.} \quad & \sum_{i=1}^{I_k} \mathrm{Tr}\left(\mathbf{V}_{i_k}^g \mathbf{V}_{i_k}^{g\,T}\right) \le P_k, \ \ \forall\, k \in \mathcal{K},\ \forall\, g \in \mathcal{G}, \end{aligned} \tag{3.23}$$

where $\boldsymbol{\gamma}_{i_k}(\cdot)$ is the inverse map of the gradient map $\nabla_{\mathbf{E}_{i_k}} c_{i_k}(\cdot)$.

Now we use the block coordinate descent approach (see [13]) to solve (3.23). If we fix the value of $\{\mathbf{U}_{i_k}, \mathbf{V}_{i_k}\}$, the optimal $\mathbf{W}_{i_k}$ is given by $\nabla c_{i_k}(\cdot)$ (see the proof of Lemma 6). Furthermore, one can easily see that by fixing $\{\mathbf{W}_{i_k}, \mathbf{V}_{i_k}\}$, the optimum receiver is given by

$$\mathbf{U}_{i_k}^{g*} = (\mathbf{J}_{i_k}^g)^{-1} \mathbf{H}_{i_k k} \mathbf{V}_{i_k}^g, \ \forall i_k \ \in \mathcal{I},$$

where $\mathbf{J}_{i_k}^g \triangleq \sum_{(j,\ell)} \mathbf{H}_{i_k j} \mathbf{V}_{\ell_j}^g (\mathbf{V}_{\ell_j}^g)^T \mathbf{H}_{i_k j}^T + \sigma_{i_k}^2 \mathbf{I}$ is the received signal covariance matrix at receiver i_k. Finally, if we fix $\{\mathbf{U}_{i_k}, \mathbf{W}_{i_k}\}$, we need to solve the following weighted sum MSE minimization problem

$$\begin{aligned} \min_{\mathbf{V}} \quad & \sum_{k=1}^{K} \sum_{i=1}^{I_k} \sum_{g=1}^{G} \mathrm{Tr}(\mathbf{W}_{i_k}^g \mathbf{E}_{i_k}^g) \\ \text{s.t.} \quad & \sum_{i=1}^{I_k} \mathrm{Tr}\left(\mathbf{V}_{i_k}^g \mathbf{V}_{i_k}^{g\,T}\right) \leq P_k, \ \ \forall\, k \in \mathcal{K},\ \forall\, g \in \mathcal{G}, \end{aligned} \tag{3.24}$$

where $\mathbf{W}_{i_k}^g \in \mathbb{R}^{d_{i_k} \times d_{i_k}}$ is the part of $\mathbf{W}_{i_k}$ which corresponds to $\mathbf{E}_{i_k}^g$. Notice that problem (3.24) is decomposable across the base stations and groups. Using the Lagrange multipliers, we can develop closed form updates for $\mathbf{V}$ by solving (3.24). The resulting block coordinate descent algorithm is summarized in Algorithm 14. Since in the block coordinate descent method every limit point of the iterates is a stationary point [13,23], it is not hard to see that in the proposed method in algorithm 14, every limit point of the iterates is a stationary point of (3.23). Moreover, due to Lemma 6, if $(\mathbf{U}^*, \mathbf{V}^*, \mathbf{W}^*)$ is a stationary point of (3.23), then $(\mathbf{U}^*, \mathbf{V}^*)$ is a stationary point of (3.20). Therefore, the proposed method in Algorithm 14 generates a sequence converging to a stationary point of (3.20).

Algorithm 14 The proposed algorithm with no time allocation

initialize $\mathbf{V}_{i_k}^g$'s randomly such that $\mathrm{Tr}\left(\mathbf{V}_{i_k}^g (\mathbf{V}_{i_k}^g)^T\right) = \frac{p_k}{I_k}$
repeat
$\mathbf{U}_{i_k}^g \leftarrow \left(\sum_{(j,\ell)} \mathbf{H}_{i_k j} \mathbf{V}_{\ell_j}^g \mathbf{V}_{\ell_j}^{g^T} \mathbf{H}_{i_k j}^T + \sigma_{i_k}^2 \mathbf{I}\right)^{-1} \mathbf{H}_{i_k k} \mathbf{V}_{i_k}^g, \ \ \forall\, i_k \in \mathcal{I}, \ \ \forall\, g \in \mathcal{G}$
$\mathbf{W}_{i_k}^g \leftarrow \nabla_{\mathbf{E}_{i_k}^g} c_{i_k}(\cdot), \ \ \forall\, i_k \in \mathcal{I}, \ \ \forall\, g \in \mathcal{G}$
$\mathbf{V}_{i_k}^g \leftarrow \left(\sum_{(j,\ell)} \mathbf{H}_{\ell_j k}^T \mathbf{U}_{\ell_j}^g \mathbf{W}_{\ell_j}^g \mathbf{U}_{\ell_j}^{g^T} \mathbf{H}_{\ell_j k} + \mu_k^* \mathbf{I}\right)^{-1} \mathbf{H}_{i_k k}^T \mathbf{U}_{i_k}^g \mathbf{W}_{i_k}^g, \ \forall\, i_k \in \mathcal{I}, \ \ \forall\, g \in \mathcal{G}$
until

Three remarks about the proposed algorithm are in order.

1. The same WMMSE algorithm can be used in the multi-cell *uplink* scenario, where the base stations update the auxiliary variables $\mathbf{W}$ and their receive beamformers using the MMSE receiver. Users update their transmit beamformers according to the weighted MSE minimization rule similar to the downlink case.

2. The per iteration complexity of our proposed algorithm becomes $\mathcal{O}(G\kappa^2 MN^2 + G\kappa^2 NM^2 + G\kappa^2 M^3 + G\kappa N^3)$, which remains quadratic in terms of the number of users (same as the no grouping case in subsection 3.1.2). Note that here one iteration means one complete round of updating the variables for all users.

3. The distributed implementation of the proposed method is similar to the implementation of the WMMSE algorithm in subsection 3.1.2.

User Grouping and Time Allocation

Another degree of freedom in the design of optimal transmit strategy is the fraction of time allocated to each group of users. In other words, we can consider the parameter $\{\beta_g\}_{g=1}^G$ as additional optimization variables. In this case, the corresponding joint user grouping and transceiver design problem becomes the following optimization problem:

$$\begin{aligned} \max_{\mathbf{V},\boldsymbol{\beta}} \quad & \sum_{k=1}^{K} \sum_{i=1}^{I_k} u_{i_k} \left(\sum_{g=1}^{G} \beta_g R_{i_k}^g \right) \\ \text{s.t.} \quad & \sum_{i=1}^{I_k} \mathrm{Tr}\left(\mathbf{V}_{i_k}^g \mathbf{V}_{i_k}^{g\,T} \right) \leq P_k, \ \ \forall\, k \in \mathcal{K}, \ \ \forall\, g \in \mathcal{G} \\ & \sum_{g=1}^{G} \beta_g = 1, \ \beta_g \geq 0, \quad \forall\, g \in \mathcal{G}. \end{aligned} \tag{3.25}$$

Let us again assume that $u_{i_k}(\cdot)$ is a strictly increasing function of R_{i_k}. Defining

$$c_{i_k}\left(\boldsymbol{\beta}, \mathbf{E}_{i_k}\right) \triangleq -u_{i_k}\left(-\sum_{g=1}^{G} \beta_g \log\det(\mathbf{E}_{i_k}^g) \right),$$

one can rewrite (3.25) as the following equivalent optimization problem

$$\begin{aligned} \min_{\mathbf{U},\mathbf{V},\boldsymbol{\beta}} \quad & \sum_{k=1}^{K}\sum_{i=1}^{I_k} c_{i_k}\left(\boldsymbol{\beta}, \mathbf{E}_{i_k}\right) \\ \text{s.t.} \quad & \sum_{i=1}^{I_k} \mathrm{Tr}\left(\mathbf{V}_{i_k}^{g}\,\mathbf{V}_{i_k}^{g\,T}\right) \leq P_k, \;\; \forall\, k \in \mathcal{K}, \;\; \forall\, g \in \mathcal{G} \\ & \sum_{g=1}^{G} \beta_g = 1, \;\; \beta_g \geq 0, \quad \forall\, g \in \mathcal{G}. \end{aligned} \tag{3.26}$$

For a fixed value of β, the problem is similar to the one in the previous section. Hence the update rules of $\mathbf{U}, \mathbf{V}, \mathbf{W}$, derived in the previous section, can be used in this problem as well. To update the variable β, we can fix all other variables and solve

$$\begin{aligned} \min_{\boldsymbol{\beta}} \quad & \sum_{k=1}^{K}\sum_{i=1}^{I_k} c_{i_k}\left(\boldsymbol{\beta}, \mathbf{E}_{i_k}\right) \\ \text{s.t.} \quad & \sum_{g=1}^{G} \beta_g = 1, \;\; \beta_g \geq 0, \quad \forall\, g \in \mathcal{G}. \end{aligned} \tag{3.27}$$

Note that when $u_{i_k}(\cdot)$ is concave for all $i_k \in \mathcal{I}$, e.g., $u_{i_k} = \log(R_{i_k})$ (which represents the proportional fairness utility function), the objective function $c_{i_k}\left(\boldsymbol{\beta}, \mathbf{E}_{i_k}\right) = -u_{i_k}\left(-\sum_{g=1}^{G} \beta_g \log\det(\mathbf{E}_{i_k}^{g})\right)$ is convex in $\boldsymbol{\beta}$ and the above problem can be solved efficiently. Moreover, problem (3.27) does not need the knowledge of channel coefficients and therefore can be solved in a centralized manner in the MAC layer. The overall proposed algorithm is summarized in Algorithm 15.

We emphasize that Algorithm 15 is not the standard block coordinate descent (BCD) method. The reason is that for updating the variables $(\mathbf{U}, \mathbf{V}, \mathbf{W})$, we consider the objective function in (3.23), while for updating the variable $\boldsymbol{\beta}$, the objective function in (3.27) is considered. This type of update rule prevents us from applying the classical convergence result of the BCD method. In fact, we need to use another interpretation of the algorithm for studying its convergence. Our next result shows that the proposed method in Algorithm 15 converges to a stationary point of (3.26) if $c_{i_k}(\boldsymbol{\beta}, \mathbf{E}_{i_k})$ is a strictly concave function of $\mathbf{E}_{i_k}$ for all $i_k \in \mathcal{I}$.

Algorithm 15 The proposed algorithm when $\boldsymbol{\beta}$ is a design variable

initialize $\mathbf{V}_{i_k}^g$'s randomly such that $\mathrm{Tr}\left(\mathbf{V}_{i_k}^g(\mathbf{V}_{i_k}^g)^T\right) = \frac{p_k}{I_k}$
initialize $\boldsymbol{\beta}$ with $\beta_g = \frac{1}{G},\ \forall g$
repeat
$\mathbf{W}_{i_k}^g \leftarrow \nabla_{\mathbf{E}_{i_k}^g} c_{i_k}(\cdot),\ \ \forall\, i_k \in \mathcal{I},\ \ \forall\, g \in \mathcal{G}$
$\mathbf{U}_{i_k}^g \leftarrow \left(\sum_{(j,\ell)} \mathbf{H}_{i_k j}\mathbf{V}_{\ell_j}^g \mathbf{V}_{\ell_j}^{g^T}\mathbf{H}_{i_k j}^T + \sigma_{i_k}^2\mathbf{I}\right)^{-1}\mathbf{H}_{i_k k}\mathbf{V}_{i_k}^g,\ \ \forall\, i_k \in \mathcal{I},\ \ \forall\, g \in \mathcal{G}$
$\mathbf{V}_{i_k}^g \leftarrow \left(\sum_{(j,\ell)} \mathbf{H}_{\ell_j k}^T\mathbf{U}_{\ell_j}^g \mathbf{W}_{\ell_j}^g \mathbf{U}_{\ell_j}^{g^T}\mathbf{H}_{\ell_j k} + \mu_k^*\mathbf{I}\right)^{-1}\mathbf{H}_{i_k k}^T\mathbf{U}_{i_k}^g\mathbf{W}_{i_k}^g,\ \forall\, i_k \in \mathcal{I},\ \ \forall\, g \in \mathcal{G}$
update $\{\beta_g\}_{g=1}^G$ by solving (3.27)
until convergence

Theorem 18 *Assume the optimal value of $\boldsymbol{\beta}$ in* (3.27) *is unique and positive. Moreover, suppose that $c_{i_k}(\boldsymbol{\beta}, \mathbf{E}_{i_k})$ is strict concave as a function of $\mathbf{E}_{i_k}$ for any fixed positive value of $\boldsymbol{\beta}$. Then every limit point of the proposed algorithm in Algorithm 15 is a stationary point of* (3.26).

Proof To prove this theorem, we use the convergence result of Block Successive Upper-bound Minimization (BSUM) method; see Theorem 2. Hence, we only need to show that at each step of updating the variables $\boldsymbol{\beta}, \mathbf{U}$, and $\mathbf{V}$, we minimize a convex upper bound of the objective function which is tight at the current step.

Fix $\boldsymbol{\beta}$. Since $c_{i_k}(\boldsymbol{\beta}, \mathbf{E}_{i_k})$ is a strictly concave function of $\mathbf{E}_{i_k}$, the first order Taylor expansion of $c_{i_k}(\boldsymbol{\beta}, \cdot)$ around any point $\mathbf{E}_{i_k}^0$ is an upper bound of the function $c_{i_k}(\boldsymbol{\beta}, \cdot)$, i.e.,

$$c_{i_k}(\boldsymbol{\beta}, \mathbf{E}_{i_k}) \le c_{i_k}(\boldsymbol{\beta}, \mathbf{E}_{i_k}^0) + \mathrm{Tr}\left(\nabla_{\mathbf{E}_{i_k}} c_{i_k}(\boldsymbol{\beta}, \mathbf{E}_{i_k}^0)\left(\mathbf{E}_{i_k} - \mathbf{E}_{i_k}^0\right)\right), \quad \forall\, \mathbf{E}_{i_k}. \tag{3.28}$$

Clearly, the MSE value $\mathbf{E}_{i_k}$ is a function of all receive beamformers $\{\mathbf{U}_{j\ell}\}_{j\ell\in\mathcal{I}}$ as well as all transmit beamformers $\{\mathbf{V}_{j\ell}\}_{j\ell\in\mathcal{I}}$. Therefore, one can consider the right hand side of (3.28) as a function of $(\mathbf{U}, \mathbf{V})$. By summing up (3.28) across all the users, we can define the function

$$g(\mathbf{U}, \mathbf{V}; \mathbf{E}^0, \boldsymbol{\beta}) \triangleq \sum_{k=1}^{K}\sum_{i=1}^{I_k}\left(c_{i_k}(\boldsymbol{\beta}, \mathbf{E}_{i_k}^0) + \mathrm{Tr}\left(\nabla_{\mathbf{E}_{i_k}} c_{i_k}(\boldsymbol{\beta}, \mathbf{E}_{i_k}^0)\left(\mathbf{E}_{i_k}(\mathbf{U}, \mathbf{V}) - \mathbf{E}_{i_k}^0\right)\right)\right),$$

for any fixed value of $\mathbf{E}^0$ and $\boldsymbol{\beta}$. Notice that the function $g(\mathbf{U}, \mathbf{V}; \mathbf{E}^0, \boldsymbol{\beta})$ is convex

individually in the variables $\mathbf{U}$ and $\mathbf{V}$. Moreover, it is an upper bound of the objective function in (3.26), i.e.,

$$\sum_{k=1}^{K}\sum_{i=1}^{I_k} c_{i_k}(\boldsymbol{\beta}, \mathbf{E}_{i_k}) \leq g(\mathbf{U}, \mathbf{V}; \mathbf{E}^0, \boldsymbol{\beta}),$$

where this upper bound is tight at the point $(\mathbf{U}^0, \mathbf{V}^0)$ at which $\mathbf{E}(\mathbf{U}^0, \mathbf{V}^0) = \mathbf{E}^0$. Now, we claim that the steps of updating the beamformers $\mathbf{U}, \mathbf{V}$ in the proposed algorithm are equivalent to minimizing the convex upper bound $g(\cdot)$. To see the this, let us consider the update rule of the receive beamformer $\mathbf{U}$. First of all, this update rule is derived by solving the following optimization problem

$$\min_{\mathbf{U}} \quad \sum_{k=1}^{K}\sum_{i=1}^{I_k} \mathrm{Tr}\left(\mathbf{W}_{i_k}\mathbf{E}_{i_k}\right), \tag{3.29}$$

where $\mathbf{W}_{i_k} = \nabla_{\mathbf{E}_{i_k}} c_{i_k}(\boldsymbol{\beta}, \mathbf{E}^0_{i_k})$ and $\mathbf{E}^0_{i_k}$ is the MSE value at the previous iteration. Clearly, (3.29) is equivalent to

$$\min_{\mathbf{U}} \quad g(\mathbf{U}, \mathbf{V}; \mathbf{E}^0, \boldsymbol{\beta}).$$

Thus, for updating the receive beamformer $\mathbf{U}$ in the proposed algorithm in Algorithm 15, we minimize a locally tight strictly convex upper bound of the objective function of (3.26). Similarly, we can argue that the step of updating the transmit beamformer $\mathbf{V}$ corresponds to minimizing a locally tight strictly convex upper bound of the objective function.

It follows that at steps of updating $\mathbf{U}$, $\mathbf{V}$, and $\boldsymbol{\beta}$ in the proposed algorithm, we update the variables by minimizing upper bounds of the objective function of (3.26). Moreover, these upper bounds are convex, tight at the current iteration, and have unique minimizers for $\mathbf{U}, \boldsymbol{\beta}$. Therefore, the BSUM convergence result (Theorem 2) implies that every limit point of the iterates generated by the algorithm is a stationary point of (3.26).

3.1.4 Beamforming for Max-Min Fairness

Consider the system model of subsection 3.1.2. Here our focus is on the max-min utility function, i.e., we are interested in solving the following problem

$$\begin{aligned} \max_{\{\mathbf{V}_{i_k}\}_{i_k\in\mathcal{I}}} \quad & \min_{i_k\in\mathcal{I}} \quad R_{i_k}(\mathbf{V}) \\ \text{s.t.} \quad & \sum_{i=1}^{I_k} \mathrm{Tr}(\mathbf{V}_{i_k}\mathbf{V}_{i_k}^H) \le P_k, \quad \forall\, k \in \mathcal{K}. \end{aligned} \tag{P}$$

Similar to [122], one can solve (P) by solving a series of problems of the following type for different values of γ:

$$\begin{aligned} \min_{\{\mathbf{V}_{i_k}\}_{i_k\in\mathcal{I}}} \quad & \sum_{k=1}^{K}\sum_{i=1}^{I_k} \mathrm{Tr}(\mathbf{V}_{i_k}\mathbf{V}_{i_k}^H) \\ \text{s.t.} \quad & R_{i_k}(\mathbf{V}) \ge \gamma, \quad \forall\, i_k \in \mathcal{I} \\ & \sum_{i=1}^{I_k} \mathrm{Tr}(\mathbf{V}_{i_k}\mathbf{V}_{i_k}^H) \le P_k, \quad \forall\, k \in \mathcal{K}. \end{aligned} \tag{3.30}$$

The above problem is to minimize the total power consumption in the network subject to quality of service (QoS) constraints. In what follows, we first study the complexity status of problem (P) and (3.30). Then, we propose an efficient algorithm for designing the beamformers based on the maximization of the worst user performance in the system.

NP-Hardness of Optimal Beamformer Design

Here we analyze the complexity status of problem (P) and (3.30). In the single input single output (SISO) case where $M_k = N_{i_k} = 1, \forall\, k \in \mathcal{K}, \forall\, i_k \in \mathcal{I}$, it has been shown that problem (P) and problem (3.30) can be solved in polynomial time, see [101] and the references therein. Furthermore, it is shown that in the multiple input single output (MISO) case where $M_k > N_{i_k} = 1, \forall\, k \in \mathcal{K}, \forall\, i_k \in \mathcal{I}$, both problems are still polynomial time solvable [139,140]. In this section, we consider the MIMO case where $M_k \ge 2$, and $N_{i_k} \ge 2$. We show that unlike the above mentioned special cases, both problems (P) and (3.30) are NP-hard.

In fact, it is sufficient to show that for a *simpler* MIMO IC network with K transceiver pairs and with each node equipped with at least two antennas, solving the max-min

problem (P) and the min-power problem (3.30) are both NP-hard. For convenience, we rewrite the max-min beamformer design problem in this K user MIMO IC as an equivalent[2] covariance maximization form

$$\begin{array}{ll} \max\limits_{(\lambda,\mathbf{Q})} & \lambda \\ \text{s.t.} & \lambda \le R_k(\mathbf{Q}),\ \mathrm{Tr}(\mathbf{Q}_k) \le P_k, \mathbf{Q}_k \succeq 0, \quad \forall\ k = 1, \cdots, K. \end{array} \tag{3.31}$$

where $R_k(\mathbf{Q}) = \log\det\left(\mathbf{I} + \mathbf{H}_{kk}\mathbf{Q}_k\mathbf{H}_{kk}^H(\sigma_k^2\mathbf{I} + \sum_{j\neq k}\mathbf{H}_{kj}\mathbf{Q}_j\mathbf{H}_{kj}^H)^{-1}\right)$. Note that λ is the slack variable that is introduced to represent the objective value of the problem. The first step towards proving the desired complexity result is to recognize certain special structures in the optimal solutions of the problem (3.31). More specifically, since most of the well-known NP-hard problems are discrete, in order to relate (3.31) to the well-known NP-hard problems, we need to find problem instances for which the solution set of (3.31) has some discrete structure. To find such problem instances, let us consider a 3-user MIMO IC with two antennas at each node. Suppose $\sigma_k^2 = P_k = 1$ for all k and the channels are given as

$$\mathbf{H}_{ii} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \forall\, i = 1,2,3 \quad \text{and} \quad \mathbf{H}_{im} = \begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix}, \quad \forall\, i \neq m, \quad i,m = 1,2,3. \tag{3.32}$$

Our first result characterizes the global optimal solutions for problem (3.31) in this special network.

Lemma 7 *Suppose $K = 3$ and the channels are given as* (3.32)*. Let $\mathcal{S} = \{(\lambda^*, \mathbf{Q}_1^*, \mathbf{Q}_2^*, \mathbf{Q}_3^*)\}$ denote the set of optimal solutions of the problem* (3.31)*. Then $\mathcal{S}$ can be expressed as*

$$\mathcal{S} = \{(1, \mathbf{Q}_a^*, \mathbf{Q}_a^*, \mathbf{Q}_a^*), (1, \mathbf{Q}_b^*, \mathbf{Q}_b^*, \mathbf{Q}_b^*), (1, \mathbf{Q}_c^*, \mathbf{Q}_c^*, \mathbf{Q}_c^*), (1, \mathbf{Q}_d^*, \mathbf{Q}_d^*, \mathbf{Q}_d^*)\}, \tag{3.33}$$

where $\mathbf{Q}_a^* = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$, $\mathbf{Q}_b^* = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$, $\mathbf{Q}_c^* = \begin{bmatrix} 0.5 & 0.5j \\ -0.5j & 0.5 \end{bmatrix}$, *and* $\mathbf{Q}_d^* = \begin{bmatrix} 0.5 & -0.5j \\ 0.5j & 0.5 \end{bmatrix}$.

The proof of this lemma can be found in the Appendix. Next we proceed to consider a 5-user interference channel with two antennas at each node. Again suppose $\sigma_k^2 = 1,\ \forall k$

[2] The equivalence is in the sense that for every optimal solution $\{\mathbf{V}^*\}$ of (P) with $M_k = d_k$, there exists $\lambda^* \geq 0$ so that by defining $\mathbf{Q}_k^* = \mathbf{V}_k^*\mathbf{V}_k^{*H}, \forall\, k$, the point $\{\lambda^*, \mathbf{Q}^*\}$ is an optimal solution of (3.31). Conversely, if $\{\lambda^*, \mathbf{Q}^*\}$ is an optimal solution of (3.31) and $\mathbf{Q}_k^* = \mathbf{V}_k^*\mathbf{V}_k^{*H}, \forall\, k$, then $\mathbf{V}^*$ is an optimal solution of (P).

and the channels are given as

$$\mathbf{H}_{ii} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \forall\, i = 1,2,3 \quad \text{and} \quad \mathbf{H}_{im} = \begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix}, \quad \forall\, i \neq m, \quad i,m = 1,2,3; \quad (3.34)$$

$$\mathbf{H}_{ii} = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}, \forall\, i = 4,5, \quad \mathbf{H}_{4m} = \begin{bmatrix} 1 & j \\ 0 & 0 \end{bmatrix}, \forall\, m = 1,2,3, \quad \mathbf{H}_{5m} = \begin{bmatrix} j & 1 \\ 0 & 0 \end{bmatrix}, \forall\, m = 1,2,3; \quad (3.35)$$

$$\mathbf{H}_{im} = 0,\ \forall\, i = 1,2,3,\ \forall\, m = 4,5; \quad \mathbf{H}_{im} = 0,\ \forall\, i \neq m,\ i,m = 4,5. \quad (3.36)$$

Our next result characterizes the global optimal solutions for the problem (3.31) for this special case.

Lemma 8 *Suppose $K = 5$ and the channels are given as* (3.34)–(3.36)*. Let $\mathbf{Q}_a^*,\ \mathbf{Q}_b^*$ be defined in Lemma 7. Denote the set of optimal solutions of the problem* (3.31) *as $\mathcal{T}$. Then $\mathcal{T}$ can be expressed as*

$$\mathcal{T} = \{(1, \mathbf{Q}_a^*, \mathbf{Q}_a^*, \mathbf{Q}_a^*, \mathbf{Q}_a^*, \mathbf{Q}_a^*), (1, \mathbf{Q}_b^*, \mathbf{Q}_b^*, \mathbf{Q}_b^*, \mathbf{Q}_a^*, \mathbf{Q}_a^*)\}\,. \quad (3.37)$$

Proof First of all, it is not hard to see that by selecting each of the values in the optimal set $\mathcal{T}$, we get the objective value of $\lambda^* = 1$. Therefore, it suffices to show that for any other feasible point, we get lower objective value. To show this, we first notice that the first three users form an interference channel which is exactly the same as the one in Lemma 7. Therefore, in order to get the minimum rate of one, we need to use one of the optimal solutions in $\mathcal{S}$ in Lemma 7 for $(\mathbf{Q}_1, \mathbf{Q}_2, \mathbf{Q}_3)$. Furthermore, it is not hard to see that using either $(\mathbf{Q}_1, \mathbf{Q}_2, \mathbf{Q}_3) = (\mathbf{Q}_c^*, \mathbf{Q}_c^*, \mathbf{Q}_c^*)$ or $(\mathbf{Q}_1, \mathbf{Q}_2, \mathbf{Q}_3) = (\mathbf{Q}_d^*, \mathbf{Q}_d^*, \mathbf{Q}_d^*)$ would cause high interference to either user 4 or user 5 and prevent them from achieving the communication rate of one. Therefore, the only optimal solutions are the ones in the set $\mathcal{T}$.

Using Lemma 8, we can discretize the variables in the max-min problem and use it to prove the NP-hardness of the problem. In fact, for any 5 users similar to the ones in Lemma 8, there are only two possible strategies that can maximize the minimum rate of communication: either we should transmit on the first antenna or transmit on the

second antenna. This observation will be crucial in establishing our NP-hardness result.

Theorem 19 *For a K-cell MIMO interference channel where each transmit/receive node is equipped with at least two antennas, the problem of designing covariance matrices to achieve max-min fairness is NP-hard in K. More specifically, solving the following problem is NP-hard*

$$\begin{aligned} \max_{\{\mathbf{Q}_i\}_{i=1}^K} \quad & \min_k \quad \log\det\left(\mathbf{I}+\mathbf{H}_{kk}\mathbf{Q}_k\mathbf{H}_{kk}^H\big(\sigma_k^2\mathbf{I}+\sum_{j\neq k}\mathbf{H}_{kj}\mathbf{Q}_j\mathbf{H}_{kj}^H\big)^{-1}\right) \\ \text{s.t.} \quad & \mathrm{Tr}(\mathbf{Q}_k)\leq P_k, \mathbf{Q}_k\succeq 0,\ k=1,\cdots,K. \end{aligned} \tag{3.38}$$

This theorem is proved based on a polynomial time reduction from the 3-satisfiability (3-SAT) problem which is known to be NP-complete [141]. The 3-SAT problem is described as follows. Consider M disjunctive clauses $c_1,\ldots,c_M$ defined on N Boolean variables $x_1,\ldots,x_N$ and their negations $\bar{x}_1,\ldots,\bar{x}_N$. More specifically, let $c_m = y_{m1}\vee y_{m2}\vee y_{m3}$ where $\vee$ denotes the *Boolean OR* operation and $y_{mi}\in\{x_1,\ldots,x_N,\bar{x}_1,\ldots,\bar{x}_N\}$. The 3-SAT problem is to check whether there exists a truth assignment for the Boolean variables such that all the clauses are satisfied simultaneously. The details of the proof of the theorem can be found in Appendix.

Corollary 4 *Under the same set up as in Theorem 19, problem* (3.30) *is NP-hard.*

To see why the above corollary holds, we assume the contrary. Then a binary search procedure for λ would imply a polynomial time algorithm for (P), which would contradict the NP-hardness result of Theorem 19.

The Proposed Algorithm

The complexity results established in the previous section suggest that it is generally not possible to solve the max-min problem (P) to its global optimality in a time that grows polynomially in K. Guided by this insight, we reset our goal to that of designing computationally efficient algorithms that can compute a high quality solution for (P). To this end, we first provide an equivalent reformulation of problem (P), which will be used later for our algorithm design.

Introducing a slack variable λ, the problem (P) can be equivalently written as

$$\begin{aligned} \max_{\{\mathbf{V}_{i_k}\}_{i_k\in\mathcal{I}},\lambda} \quad & \lambda \\ \text{s.t.} \quad & R_{i_k}(\mathbf{V}) \geq \lambda, \forall\ i_k \in \mathcal{I} \\ & \sum_{i\in\mathcal{I}_k} \mathrm{Tr}[\mathbf{V}_{i_k}\mathbf{V}_{i_k}^H] \leq P_k,\ \forall\ k \in \mathcal{K}. \end{aligned} \tag{P1}$$

Using the observation in Theorem 17, we can apply Algorithm 1 by successively approximating the constraints in (P1). Then at each iteration of the algorithm, solving the subproblem is equivalent to solving a problem of the following form

$$\begin{aligned} \max_{\mathbf{V},\lambda} \quad & \lambda \\ \text{s.t.} \quad & \mathrm{Tr}[\mathbf{W}_{i_k}\mathbf{E}_{i_k}] - \log\det(\mathbf{W}_{i_k}) - d_{i_k} \leq -\lambda,\ \forall\ i_k \in \mathcal{I} \\ & \sum_{i_k\in\mathcal{I}_k} \mathrm{Tr}[\mathbf{V}_{i_k}\mathbf{V}_{i_k}^H] \leq P_k,\ \forall\ k \in \mathcal{K}. \end{aligned} \tag{Q1}$$

The overall algorithm is summarized in Algorithm 16; see [142] for details. Notice that the convergence of the algorithm to the set of KKT points is guaranteed by Theorem 1.

Algorithm 16 The Proposed Max-Min Fairness Algorithm

Initialize $\mathbf{V}_{i_k}$'s such that $\mathrm{Tr}\left(\mathbf{V}_{i_k}\mathbf{V}_{i_k}^H\right) = \frac{p_k}{I_k}$

repeat

$\mathbf{W}'_{i_k} \leftarrow \mathbf{W}_{i_k}, \quad \forall\ i_k \in \mathcal{I}$

$\mathbf{U}_{i_k} \leftarrow \left(\sum_{(j,\ell)} \mathbf{H}_{i_k j}\mathbf{V}_{\ell_j}\mathbf{V}_{\ell_j}^H\mathbf{H}_{i_k j}^H + \sigma_{i_k}^2\mathbf{I}\right)^{-1}\mathbf{H}_{i_k k}\mathbf{V}_k,\ \forall\ i_k \in \mathcal{I}$

$\mathbf{W}_{i_k} \leftarrow \left(\mathbf{I} - \mathbf{U}_{i_k}^H\mathbf{H}_{i_k k}\mathbf{V}_{i_k}\right)^{-1},\ \forall\ i_k \in \mathcal{I}$

$x^{r+1} \leftarrow x^r$

Update $\{\mathbf{V}_{i_k}\}$ by solving (Q1)

until $\left|\sum_{(j,\ell)} \log\det\left(\mathbf{W}_{\ell_j}\right) - \sum_{(j,\ell)} \log\det\left(\mathbf{W}'_{\ell_j}\right)\right| \leq \epsilon$

3.1.5 Expected Sum-Rate Maximization for Wireless Networks

The *ergodic/stochastic* transceiver design problem is a long standing problem in the signal processing and communication area, and yet no efficient algorithm has been developed to date which can deliver good practical performance. In contrast, substantial progress has been made in recent years for the *deterministic* counterpart of this problem; see [1, 17, 105, 108, 111, 112, 126, 139, 143, 144]. That said, it is important to point out that most of the proposed methods require the perfect and full channel state information (CSI) of all links – an assumption that is clearly impractical due to channel aging and channel estimation errors. More importantly, obtaining the full CSI for all links would inevitably require a prohibitively large amount of training overhead and is therefore practically infeasible.

One approach to deal with the channel aging and the full CSI problem is to use the robust optimization methodology. To date, various robust optimization algorithms have been proposed to address this issue [127–131, 145]. However, these methods are typically rather complex compared to their non-robust counterparts. Moreover, they are mostly designed for the worst case scenarios and therefore, due to their nature, are suboptimal when the worst cases happen with small probability. An alternative approach is to design the transceivers by optimizing the *average performance* using a stochastic optimization framework which requires only the *statistical channel knowledge* rather than the full instantaneous CSI. In what follows, we propose a simple iterative algorithm for ergodic/stochastic sum rate maximization problem using the SSUM framework. Unlike the previous approach of [132] which maximizes a lower bound of the expected weighted sum rate problem, our approach directly maximizes the ergodic sum rate and is guaranteed to converge to the set of stationary points of the ergodic sum rate maximization problem. Furthermore, the proposed algorithm is computationally simple, fully distributed, and has a per-iteration complexity comparable to that of the deterministic counterpart [1].

Consider the expected sum rate maximization problem:

$$\begin{aligned} \max_{\mathbf{V}} \ & \mathbb{E}_{\mathbf{H}} \left\{ \sum_{k=1}^{K} \sum_{i=1}^{I_k} \max_{\mathbf{U}_{i_k}} \left\{ R_{i_k}(\mathbf{U}_{i_k}, \mathbf{V}, \mathbf{H}) \right\} \right\} \\ \text{s.t.} \ & \sum_{i=1}^{I_k} \mathrm{Tr}(\mathbf{V}_{i_k} \mathbf{V}_{i_k}^H) \leq P_k, \ \forall \ k = 1, \cdots, K. \end{aligned} \tag{3.39}$$

To be consistent with the SSUM section, let us rewrite (3.39) as a minimization problem:

$$\begin{aligned} \min_{\mathbf{V}} \ & \mathbb{E}_{\mathbf{H}} \{ g_1(\mathbf{V}, \mathbf{H}) \} \\ \text{s.t.} \ & \sum_{i=1}^{I_k} \mathrm{Tr}(\mathbf{V}_{i_k} \mathbf{V}_{i_k}^H) \leq P_k, \ \forall \ k = 1, \cdots, K, \end{aligned} \tag{3.40}$$

where

$$g_1(\mathbf{V}, \mathbf{H}) \ = \ \sum_{k=1}^{K} \sum_{i=1}^{I_k} \min_{\mathbf{U}_{i_k}} \left\{ -R_{i_k}(\mathbf{U}_{i_k}, \mathbf{V}, \mathbf{H}) \right\}. \tag{3.41}$$

It can be checked that g_1 is smooth but non-convex in $\mathbf{V}$ [1]. In practice, due to other design requirements, one might be interested in adding some convex non-smooth regularizer to the above objective function. For example, the authors of [146] added a convex group sparsity promoting regularizer term to the objective for the purpose of joint base station assignment and beamforming optimization. In such a case, since the non-smooth part is convex, the SSUM algorithm is still applicable. For simplicity, we consider only the simple case of $g_2 \equiv 0$ in this section.

In order to utilize the SSUM algorithm, we need to find a convex tight upper-bound approximation of $g_1(\mathbf{V}, \mathbf{H})$. To do so, let us introduce a set of variables $\mathbf{P} \triangleq (\mathbf{W}, \mathbf{U}, \mathbf{Z})$, where $\mathbf{W}_{i_k} \in \mathbb{C}^{d_{i_k} \times d_{i_k}}$ (with $\mathbf{W}_{i_k} \succeq \mathbf{0}$) and $\mathbf{Z}_{i_k} \in \mathbb{C}^{M_k \times d_{i_k}}$ for any $i = 1, \cdots, I_k$ and for all $k = 1, \cdots, K$. Furthermore, define

$$\begin{aligned} \hat{R}_{i_k}(\mathbf{W}_{i_k}, \mathbf{Z}_{i_k}, \mathbf{U}_{i_k}, \mathbf{V}, \mathbf{H}) \ \triangleq \ & - \log\det(\mathbf{W}_{i_k}) \ + \ \mathrm{Tr}(\mathbf{W}_{i_k} \mathbf{E}_{i_k}(\mathbf{U}_{i_k}, \mathbf{V})) \ + \\ & \frac{\rho}{2} \| \mathbf{V}_{i_k} - \mathbf{Z}_{i_k} \|^2 - d_{i_k}, \end{aligned} \tag{3.42}$$

for some fixed $\rho > 0$ and

$$\mathcal{G}_1(\mathbf{V}, \mathbf{P}, \mathbf{H}) \triangleq \sum_{k=1}^{K} \sum_{i=1}^{I_k} \hat{R}_{i_k}(\mathbf{W}_{i_k}, \mathbf{Z}_{i_k}, \mathbf{U}_{i_k}, \mathbf{V}, \mathbf{H}). \tag{3.43}$$

Using the first order optimality condition, we can check that

$$g_1(\mathbf{V}, \mathbf{H}) = \min_{\mathbf{P}} \mathcal{G}_1(\mathbf{V}, \mathbf{P}, \mathbf{H}).$$

Now, let us define

$$\hat{g}_1(\mathbf{V}, \bar{\mathbf{V}}, \mathbf{H}) = \mathcal{G}_1(\mathbf{V}, \mathcal{P}(\bar{\mathbf{V}}, \mathbf{H}), \mathbf{H}),$$

where

$$\mathcal{P}(\bar{\mathbf{V}}, \mathbf{H}) = \arg\min_{\mathbf{P}} \mathcal{G}_1(\bar{\mathbf{V}}, \mathbf{P}, \mathbf{H}).$$

Clearly, we have

$$g_1(\bar{\mathbf{V}}, \mathbf{H}) = \min_{\mathbf{P}} \mathcal{G}_1(\bar{\mathbf{V}}, \mathbf{P}, \mathbf{H}) = \mathcal{G}_1(\bar{\mathbf{V}}, \mathcal{P}(\bar{\mathbf{V}}, \mathbf{H}), \mathbf{H}) = \hat{g}_1(\bar{\mathbf{V}}, \bar{\mathbf{V}}, \mathbf{H}),$$

and

$$g_1(\mathbf{V}, \mathbf{H}) = \min_{\mathbf{P}} \mathcal{G}_1(\mathbf{V}, \mathbf{P}, \mathbf{H}) \leq \mathcal{G}_1(\mathbf{V}, \mathcal{P}(\bar{\mathbf{V}}, \mathbf{H}), \mathbf{H}) = \hat{g}_1(\mathbf{V}, \bar{\mathbf{V}}, \mathbf{H}).$$

Furthermore, $\hat{g}_1(\mathbf{V}, \bar{\mathbf{V}}, \mathbf{H})$ is strongly convex in $\mathbf{V}$ with parameter ρ due to the quadratic term in (3.42). Hence $\hat{g}_1(\mathbf{V}, \bar{\mathbf{V}}, \mathbf{H})$ satisfies the assumptions A1-A3. In addition, if the channels lie in a bounded subset with probability one and the noise power $\sigma_{i_k}^2$ is strictly positive for all users, then it can be checked that $g_1(\mathbf{V}, \mathbf{H})$ and $\hat{g}_1(\mathbf{V}, \bar{\mathbf{V}}, \mathbf{H})$ satisfy the assumptions B1-B6. Consequently, we can apply the SSUM algorithm to solve (3.40).

Define $\mathbf{H}^r$ to be the r-th channel realization. Let us further define

$$\mathbf{P}^r \triangleq \arg\min_{\mathbf{P}} \mathcal{G}_1(\mathbf{V}^{r-1}, \mathbf{P}, \mathbf{H}^r), \tag{3.44}$$

where $\mathbf{V}^{r-1}$ denotes the transmit beamformer at iteration $r-1$. Notice that $\mathbf{P}^r$ is well defined since the optimizer of (3.44) is unique. With these definitions, the update rule

of the SSUM algorithm becomes

$$\begin{aligned}\mathbf{V}^r \leftarrow \arg\min_{\mathbf{V}} \; & \frac{1}{r}\sum_{i=1}^{r} \hat{g}_1(\mathbf{V}, \mathbf{V}^{i-1}, \mathbf{H}^i) \\ \text{s.t.} \quad & \sum_{i=1}^{I_k} \mathrm{Tr}(\mathbf{V}_{i_k}\mathbf{V}_{i_k}^H) \le P_k, \quad \forall\ k\end{aligned}$$

or equivalently

$$\begin{aligned}\mathbf{V}^r \leftarrow \arg\min \; & \frac{1}{r}\sum_{i=1}^{r} \mathcal{G}_1(\mathbf{V}, \mathbf{P}^i, \mathbf{H}^i) \\ \text{s.t.} \quad & \sum_{i=1}^{I_k} \mathrm{Tr}(\mathbf{V}_{i_k}\mathbf{V}_{i_k}^H) \le P_k, \quad \forall\ k.\end{aligned} \tag{3.45}$$

In order to make sure that the SSUM algorithm can efficiently solve (3.40), we need to confirm that the update rules of the variables $\mathbf{V}$ and $\mathbf{P}$ can be performed in a computationally efficient manner in (3.44) and (3.45). Checking the first order optimality condition of (3.44), it can be shown that the updates of the variable $\mathbf{P} = (\mathbf{W}, \mathbf{U}, \mathbf{Z})$ can be done in closed form; see Algorithm 17. Moreover, for updating the variable $\mathbf{V}$, we need to solve a simple quadratic problem in (3.45). Using the Lagrange multipliers, the update rule of the variable $\mathbf{V}$ can be performed using a one dimensional search method over the Lagrange multiplier [1]. Algorithm 17 summarizes the SSUM algorithm applied to the expected sum rate maximization problem; we name this algorithm as stochastic weighted mean square error minimizations (stochastic WMMSE) algorithm. Notice that although in the SSUM algorithm the update of the precoder $\mathbf{V}_{i_k}$ depends on all the past realizations, Algorithm 17 shows that all the required information (for updating $\mathbf{V}_{i_k}$) can be encoded into two matrices $\mathbf{A}_{i_k}$ and $\mathbf{B}_{i_k}$, which are updated recursively.

Remark 3 *Similar to the deterministic WMMSE algorithm [1] which works for the general α-fairness utility functions, the Stochastic WMMSE algorithm can also be extended to maximize the expected sum of such utility functions; see [1] for more details on the derivations of the respective update rules.*

Algorithm 17 Stochastic WMMSE Algorithm for sum rate maximization

Initialize $\mathbf{V}$ randomly such that $\sum_{i=1}^{I_k} \mathrm{Tr}\left(\mathbf{V}_{i_k}\mathbf{V}_{i_k}^H\right) = P_k,\ \forall\ k$ and set $r = 0$

repeat

$r \leftarrow r + 1$

Obtain the new channel estimate/realization $\mathbf{H}^r$

$\mathbf{U}_{i_k} \leftarrow \left(\sum_{j=1}^{K}\sum_{l=1}^{I_j} \mathbf{H}_{i_kj}^r \mathbf{V}_{l_j}\mathbf{V}_{l_j}^H (\mathbf{H}_{i_kj}^r)^H + \sigma_{i_k}^2\mathbf{I}\right)^{-1} \mathbf{H}_{i_kk}^r \mathbf{V}_{i_k},\ \forall\ k,\ i = 1, \cdots, I_k$

$\mathbf{W}_{i_k} \leftarrow \left(\mathbf{I} - \mathbf{U}_{i_k}^H \mathbf{H}_{i_kk}^r \mathbf{V}_{i_k}\right)^{-1},\ \forall\ k,\ i = 1, \cdots, I_k$

$\mathbf{Z}_{i_k} \leftarrow \mathbf{V}_{i_k},\ \forall\ k,\ i = 1, \cdots, I_k$

$\mathbf{A}_{i_k} \leftarrow \mathbf{A}_{i_k} + \rho\mathbf{I} + \sum_{j=1}^{K}\sum_{l=1}^{I_j} (\mathbf{H}_{l_jk}^r)^H \mathbf{U}_{l_j}\mathbf{W}_{l_j}\mathbf{U}_{l_j}^H \mathbf{H}_{l_jk}^r,\ \forall\ k,\ i = 1, \cdots, I_k$

$\mathbf{B}_{i_k} \leftarrow \mathbf{B}_{i_k} + \rho\mathbf{Z}_{i_k} + (\mathbf{H}_{i_kk}^r)^H \mathbf{U}_{i_k}\mathbf{W}_{i_k},\ \forall\ k,\ i = 1, \cdots, I_k$

$\mathbf{V}_{i_k} \leftarrow (\mathbf{A}_{i_k} + \mu_k^*\mathbf{I})^{-1}\mathbf{B}_{i_k},\ \forall\ k,\ i = 1, \cdots, I_k$, where μ_k^* is the optimal Lagrange multiplier for the constraint $\sum_{i=1}^{I_k} \mathrm{Tr}(\mathbf{V}_k\mathbf{V}_k^H) \leq P_k$ which can be found using bisection.

until some convergence criterion is met.

3.2 Dictionary Learning for Sparse Representation

In this section, we consider the dictionary learning problem for sparse representation. We first state the problem and establish the NP-hardness of this problem. Then we consider different formulations of the dictionary learning problem and propose several efficient algorithms to solve this problem. In particular, we consider both the batched and online dictionary learning algorithm and see how the idea of successive convex approximation could be helpful for this particular problem. In contrast to the existing dictionary training algorithms [22, 147, 148], our methods neither solve Lasso-type subproblems nor find the active support of the sparse representation vector at each step; instead, they require only simple inexact updates in closed form. Furthermore, unlike most of the existing methods in the literature, e.g., [22, 148], the iterates generated by the proposed dictionary learning algorithms are theoretically guaranteed to converge to the set of stationary points under certain mild assumptions.

3.2.1 Problem Statement

Given a set of training signals $\mathrm{Y} = \{\mathbf{y}_i \in \mathbb{R}^n \mid i = 1, 2, \ldots, N\}$, our task is to find a dictionary $\mathrm{A} = \{\mathbf{a}_i \in \mathbb{R}^n \mid i = 1, 2, \ldots, k\}$that can sparsely represent the training signals in the set Y. Let $\mathbf{x}_i \in \mathbb{R}^k,\ i = 1, \ldots, N$, denote the coefficients of sparse representation of the signal $\mathbf{y}_i$, i.e., $\mathbf{y}_i = \sum_{j=1}^{k} \mathbf{a}_j x_{ij}$, where x_{ij} is the j-th component of signal $\mathbf{x}_i$. By concatenating all the training signals, the dictionary elements, and the coefficients, we can define the matrices $\mathbf{Y} \triangleq [\mathbf{y}_1, \ldots, \mathbf{y}_N]$, $\mathbf{A} \triangleq [\mathbf{a}_1, \ldots, \mathbf{a}_k]$, and $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$. Having these definitions in our hands, the dictionary learning problem for sparse representation can be stated as

$$\min_{\mathbf{A},\mathbf{X}} \ \mathrm{d}(\mathbf{Y}, \mathbf{A}, \mathbf{X}) \quad \text{s.t. } \mathbf{A} \in \mathcal{A},\ \mathbf{X} \in \mathcal{X}, \tag{3.46}$$

where $\mathcal{A}$ and $\mathcal{X}$ are two constraint sets. The function $\mathrm{d}(\cdot, \cdot, \cdot)$ measures our model goodness of fit.

3.2.2 Prior Work

The idea of representing a signal with few samples/observations dates back to the classical result of Kotelnikon, Nyquist, Shannon, and Whittaker [149–153]. This idea has evolved over time, and culminated to the *compressive sensing* concept in recent years [154, 155]. The *compressive sensing* or *sparse recovery* approach relies on the observation that many practical signals can be sparsely approximated in a suitable over-complete basis (i.e., a dictionary). In other words, the signal can be approximately written as a linear combination of only a few components (or *atoms*) of the dictionary. This observation is a key to many lossy compression methods such as JPEG and MP3.

Theoretically, the exact sparse recovery is possible with high probability under certain conditions. More precisely, it is demonstrated that if the linear measurement matrix satisfies some conditions such as null space property (NSP) or restricted isometry property (RIP), then the exact recovery is possible [154, 155]. These conditions are satisfied with high probability for different matrices such as Gaussian random matrices, Bernoulli random matrices, and partial random Fourier matrices.

In addition to the theoretical advances, compressive sensing has shown great potential in various applications. For example, in the nuclear magnetic resonance (NMR) imaging application, compressive sensing can help reduce the radiation time [156, 157].

Moreover, the compressive sensing technique has been successfully applied to many other practical scenarios including sub-Nyquist sampling [158, 159], compressive imaging [160, 161], and compressive sensor networks [162, 163], to name just a few.

In some of the aforementioned applications, the sensing matrix and dictionary are pre-defined using application domain knowledge. However, in most applications, the dictionary is not known a-priori and must be learned using a set of training signals. It has been observed that learning a good dictionary can substantially improve the compressive sensing performance, see [22,148,164–168]. In these applications, dictionary learning is the most crucial step affecting the performance of the compressive sensing approach.

To determine a high quality dictionary, various learning algorithms have been proposed; see, e.g., [22, 147, 148, 169]. These algorithms are typically composed of two major steps: 1) finding an approximate sparse representation of the training signals 2) updating the dictionary using the sparse representation.

In this subsection, we consider the dictionary learning problem for sparse representation. We first establish the NP-hardness of this problem. Then we consider different formulations of the dictionary learning problem and propose several efficient algorithms to solve this problem based on the successive convex approximation framework. In contrast to the existing dictionary training algorithms [22, 147, 148], our methods neither solve Lasso-type subproblems nor find the active support of the sparse representation vector at each step; instead, they require only simple inexact updates in closed form. Furthermore, unlike most of the existing methods in the literature, e.g., [22, 148], the iterates generated by the proposed dictionary learning algorithms are theoretically guaranteed to converge to the set of stationary points under certain mild assumptions.

3.2.3 Complexity Analysis

In this section, we analyze the computational complexity of one of the most popular forms of problem (3.46). Consider a special case of problem (3.46) by choosing the distance function to be the Frobenius norm and imposing sparsity by considering the constraint set $\mathcal{X} = \{\mathbf{X} \in \mathbb{R}^{k\times N} \mid \|\mathbf{x}_i\|_0 \leq s\}$. Then the optimization problem (3.46) can be re-written as

$$\min_{\mathbf{A},\mathbf{X}} \ \|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F^2, \ \text{ s.t. } \|\mathbf{x}_i\|_0 \leq s, \ \forall\, i = 1, \ldots, N. \tag{3.47}$$

This formulation is very popular and is considered in different studies; see, e.g., [22,170]. The following theorem characterizes the computational complexity of (3.47) by showing its NP-hardness. In particular, we show that even for the simple case of $s = 1$ and $k = 2$, problem (3.47) is NP-hard. To state our result, let us define the following concept: let $(\mathbf{A}^*, \mathbf{X}^*)$ be a solution of (3.47). For $\epsilon > 0$, we say a point $(\tilde{\mathbf{A}}, \tilde{\mathbf{X}})$ is an *ϵ-optimal solution* of (3.47) if $\|\mathbf{Y} - \tilde{\mathbf{A}}\tilde{\mathbf{X}}\|_F^2 \leq \|\mathbf{Y} - \mathbf{A}^*\mathbf{X}^*\|_F^2 + \epsilon$.

Theorem 20 *Assume $s = 1$ and $k = 2$. Then finding an ϵ-optimal algorithm for solving* (3.47) *is NP-hard. More precisely, there is no polynomial time algorithm in $N, n, \lceil \frac{1}{\epsilon} \rceil$ that can solve* (3.47) *to ϵ-optimality, unless $P = NP$.*

Proof See the appendix chapter.

Remark 4 *Note that in the above NP-hardness result, the input size of $\lceil \frac{1}{\epsilon} \rceil$ is considered instead of $\lceil \log(\frac{1}{\epsilon}) \rceil$. This in fact implies a stronger result that there is no quasi-polynomial time algorithm for solving* (3.47)*; unless P=NP.*

It is worth noting that the above NP-hardness result is different from (and is not a consequence of) the compressive sensing NP-hardness result in [171]. In fact, for a fixed sparsity level s, the compressive sensing problem is no longer NP-hard, while the dictionary learning problem considered herein remains NP-hard (see Theorem 20).

3.2.4 Batch Dictionary Learning

Optimizing the goodness of fit

In this section, we assume that the function $\mathrm{d}(\cdot)$ is composed of a smooth part and a non-smooth part for promoting sparsity, i.e., $\mathrm{d}(\mathbf{Y}, \mathbf{A}, \mathbf{X}) = \mathrm{d}_1(\mathbf{Y}, \mathbf{A}, \mathbf{X}) + \mathrm{d}_2(\mathbf{X})$, where d_1 is smooth and d_2 is continuous and possibly non-smooth. Let us further assume that the sets $\mathcal{A}, \mathcal{X}$ are closed and convex. Our approach to solve (3.46) is to apply the general block successive upper-bound minimization framework developed in [35]. More specifically, we propose to alternately update the variables $\mathbf{A}$ and $\mathbf{X}$. Let $(\mathbf{A}^r, \mathbf{X}^r)$ be the point obtained by the algorithm at iteration r. Then, we select one of the following methods to update the dictionary variable $\mathbf{A}$ at iteration $r + 1$:

(a) $\mathbf{A}^{r+1} \leftarrow \arg\min_{\mathbf{A} \in \mathcal{A}} \mathrm{d}(\mathbf{Y}, \mathbf{A}, \mathbf{X}^r)$

(b) $\mathbf{A}^{r+1} \leftarrow \arg\min_{\mathbf{A}\in\mathcal{A}} \langle \nabla_{\mathbf{A}} \mathrm{d}_1(\mathbf{Y}, \mathbf{A}^r, \mathbf{X}^r), \mathbf{A}\rangle + \frac{\tau_a^r}{2}\|\mathbf{A}-\mathbf{A}^r\|_F^2 = \mathcal{P}_{\mathcal{A}}\left(\mathbf{A}^r - \frac{1}{\tau_a^r}\nabla_{\mathbf{A}} \mathrm{d}_1(\mathbf{Y}, \mathbf{A}^r, \mathbf{X}^r)\right)$

and we update the variable $\mathbf{X}$ by

- $\mathbf{X}^{r+1} \leftarrow \arg\min_{\mathbf{X}\in\mathcal{X}} \langle \nabla_{\mathbf{X}} \mathrm{d}_1(\mathbf{Y}, \mathbf{A}^{r+1}, \mathbf{X}^r), \mathbf{X}\rangle + \frac{\tau_x^r}{2}\|\mathbf{X} - \mathbf{X}^r\|_F^2 + \mathrm{d}_2(\mathbf{X})$.

Here the operator $\langle\cdot,\cdot\rangle$ denotes the inner product; the superscript r represents the iteration number; the notation $\mathcal{P}_{\mathcal{A}}(\cdot)$ is the projection operator to the convex set $\mathcal{A}$; and the constants $\tau_a^r \triangleq \tau_a(\mathbf{Y}, \mathbf{A}^r, \mathbf{X}^r)$ and $\tau_x^r \triangleq \tau_x(\mathbf{Y}, \mathbf{A}^{r+1}, \mathbf{X}^r)$ are chosen such that

$$\begin{aligned}\mathrm{d}_1(\mathbf{Y}, \mathbf{A}, \mathbf{X}^r) \le \mathrm{d}_1(\mathbf{Y}, \mathbf{A}^r, \mathbf{X}^r) + \langle \nabla_{\mathbf{A}} \mathrm{d}_1(\mathbf{Y}, \mathbf{A}^r, \mathbf{X}^r), \mathbf{A} - \mathbf{A}^r\rangle \\ + \frac{\tau_a^r}{2}\|\mathbf{A} - \mathbf{A}^r\|_F^2,\ \forall\ \mathbf{A} \in \mathcal{A}\end{aligned}$$

and

$$\begin{aligned}\mathrm{d}(\mathbf{Y}, \mathbf{A}^{r+1}, \mathbf{X}) \le \mathrm{d}_1(\mathbf{Y}, \mathbf{A}^{r+1}, \mathbf{X}^r) + \mathrm{d}_2(\mathbf{X}) + \frac{\tau_x^r}{2}\|\mathbf{X} - \mathbf{X}^r\|_F^2 \\ + \langle \nabla_{\mathbf{X}} \mathrm{d}_1(\mathbf{Y}, \mathbf{A}^{r+1}, \mathbf{X}^r), \mathbf{X} - \mathbf{X}^r\rangle,\ \forall\ \mathbf{X} \in \mathcal{X}. \qquad (3.48)\end{aligned}$$

It should be noted that each step of the algorithm requires solving an optimization problem. For the commonly used objective functions and constraint sets, the solution to these optimization problems is often in closed form. In addition, the update rule (b) is the classical gradient projection step which can be viewed as an approximate version of (a). As we will see later, for some special choices of the function $\mathrm{d}(\cdot)$ and the set $\mathcal{A}$, using (b) leads to a closed form update rule, while (a) does not. In the sequel, we specialize this framework to different popular choices of the objective functions and the constraint sets.

Case I: Constraining the total dictionary norm
For any $\beta > 0$, we consider the following optimization problem

$$\min_{\mathbf{A},\mathbf{X}} \ \frac{1}{2}\|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F^2 + \lambda\|\mathbf{X}\|_1 \quad \text{s.t. } \|\mathbf{A}\|_F^2 \le \beta, \qquad (3.49)$$

where λ denotes the regularization parameter. By simple calculations, we can check that all the steps of the proposed algorithm can be done in closed form. More specifically, using the dictionary update rule (a) will lead to Algorithm 18. In this algorithm, $\sigma_{\max}(\cdot)$ denotes the maximum singular value; $\theta \ge 0$ is the Lagrange multiplier of the constraint $\|\mathbf{A}\|_F^2 \le \beta$ which can be found using one dimensional search algorithms such as bisection

Algorithm 18 The proposed algorithm for solving (3.49)

initialize $\mathbf{A}$ randomly such that $\|\mathbf{A}\|_F^2 \leq \beta$
repeat
  $\tau_a \leftarrow \sigma_{\max}^2(\mathbf{X})$
  $\mathbf{X} \leftarrow \mathbf{X} - \mathcal{S}_{\frac{\lambda}{\tau_a}}(\mathbf{X} - \frac{1}{\tau_a}\mathbf{A}^T(\mathbf{A}\mathbf{X} - \mathbf{Y}))$
  $\mathbf{A} \leftarrow \mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T + \theta\mathbf{I})^{-1}$
until some convergence criterion is met

or Newton. The notation $\mathcal{S}(\cdot)$ denotes the component-wise soft shrinkage operator, i.e., $\mathbf{B} = \mathcal{S}_\gamma(\mathbf{C})$ if

$$\mathbf{B}_{ij} = \begin{cases} \mathbf{C}_{ij} - \gamma & \text{if } \mathbf{C}_{ij} > \gamma \\ 0 & \text{if } -\gamma \leq \mathbf{C}_{ij} \leq \gamma \\ \mathbf{C}_{ij} + \gamma & \text{if } \mathbf{C}_{ij} < -\gamma \end{cases}$$

where $\mathbf{B}_{ij}$ and $\mathbf{C}_{ij}$ denote the (i,j)-th component of the matrices $\mathbf{B}$ and $\mathbf{C}$, respectively.

Case II: Constraining the norm of each dictionary atom
In many applications, it is of interest to constrain the norm of each dictionary atom, i.e., the dictionary is learned by solving:

$$\min_{\mathbf{A},\mathbf{X}} \ \frac{1}{2}\|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F^2 + \lambda\|\mathbf{X}\|_1 \quad \text{s.t. } \|\mathbf{a}_i\|_F^2 \leq \beta_i, \ \forall\, i \tag{3.50}$$

In this case, the dictionary update rule (a) cannot be expressed in closed form; as an alternative, we can use the update rule (b), which is in closed form, in place of (a). This gives Algorithm 19. In this algorithm, the set $\mathcal{A}$ is defined as $\mathcal{A} \triangleq \{\mathbf{A} \mid \|\mathbf{a}_i\|_F^2 \leq \beta_i, \ \forall\, i\}$

Algorithm 19 The proposed algorithm for solving (3.50) and (3.51)

For solving (3.50): initialize $\mathbf{A}$ randomly s.t. $\|\mathbf{a}_i\|_F^2 \leq \beta_i, \ \forall\, i$
For solving (3.51): initialize $\|\mathbf{A}\|_F^2 \leq \beta$ and $\mathbf{A} \geq 0$
repeat
  $\tau_x \leftarrow \sigma_{\max}^2(\mathbf{A})$
  For solving (3.50): $\mathbf{X} \leftarrow \mathbf{X} - \mathcal{S}_{\frac{\lambda}{\tau_x}}(\mathbf{X} - \frac{1}{\tau_x}\mathbf{A}^T(\mathbf{A}\mathbf{X} - \mathbf{Y}))$
  For solving (3.51): $\mathbf{X} \leftarrow \mathcal{P}_{\mathcal{X}}\left(\mathbf{X} - \frac{1}{\tau_x}\mathbf{A}^T(\mathbf{A}\mathbf{X} - \mathbf{Y}) - \lambda\right)$
  $\tau_a \leftarrow \sigma_{\max}^2(\mathbf{X})$
  $\mathbf{A} \leftarrow \mathcal{P}_{\mathcal{A}}\left(\mathbf{A} - \frac{1}{\tau_a}(\mathbf{A}\mathbf{X} - \mathbf{Y})\mathbf{X}^T\right)$
until some convergence criterion is met

Case III: Non-negative dictionary learning with the total norm constraint
Consider the non-negative dictionary learning problem for sparse representation:

$$\min_{\mathbf{A},\mathbf{X}} \ \frac{1}{2}\|\mathbf{Y}-\mathbf{A}\mathbf{X}\|_F^2+\lambda\|\mathbf{X}\|_1 \quad \text{s.t. } \|\mathbf{A}\|_F^2\leq\beta,\ \mathbf{A},\mathbf{X}\geq 0 \tag{3.51}$$

Utilizing the update rule (b) leads to Algorithm 19. Note that in this case, projections to the sets $\mathcal{X}=\{\mathbf{X}\mid\mathbf{X}\geq 0\}$ and $\mathcal{A}=\{\mathbf{A}\mid\|\mathbf{A}\|_F^2\leq\beta,\mathbf{A}\geq 0\}$ are simple. In particular, to project to the set $\mathcal{A}$, we just need to first project to the set of nonnegative matrices first and then project to the set $\tilde{\mathcal{A}}=\{\mathbf{A}\mid\|\mathbf{A}\|_F^2\leq\beta\}$.

It is worth noting that Algorithm 19 can also be applied to the case where $\mathcal{A}=\{\mathbf{A}\mid\mathbf{A}\geq 0,\ \|\mathbf{a}_i\|_F^2\leq\beta_i,\ \forall\ i\}$, since the projection to the constraint set still remains simple.

Case IV: Sparse non-negative matrix factorization
In some applications, it is desirable to have a sparse non-negative dictionary; see, e.g., [172–174]. In such cases, we can formulate the dictionary learning problem as:

$$\min_{\mathbf{A},\mathbf{X}} \ \frac{1}{2}\|\mathbf{Y}-\mathbf{A}\mathbf{X}\|_F^2+\lambda\|\mathbf{X}\|_1 \text{ s.t. } \|\mathbf{a}_i\|_1\leq\theta,\ \forall\ i,\ \mathbf{A},\mathbf{X}\geq 0 \tag{3.52}$$

It can be checked that we can again use the essentially same steps of the algorithm in case III to solve (3.52). The only required modification is in the projection step since the projection should be onto the set $\mathcal{A}=\{\mathbf{A}\mid\mathbf{A}\geq 0,\|\mathbf{a}_i\|_1\leq\theta,\ \forall\ i\}$. This step can be performed in a column-wise manner by updating each column $\mathbf{a}_i$ to $[\mathbf{a}_i-\rho_i\mathbf{1}]_+$, where $[\cdot]_+$ denotes the projection to the set of nonnegative matrices and $\rho_i\in\mathbb{R}^+$ is a constant that can be determined via one dimensional bisection. The resulting algorithm is very similar (but not identical) to the one in [172]. However, unlike the algorithm in [172], all of our proposed algorithms are theoretically guaranteed to converge, as shown in Theorem 21.

Theorem 21 *The iterates generated by the algorithms in cases I-IV converge to the set of stationary points of the corresponding optimization problems.*

Proof: Each of the proposed algorithms in cases I-IV is a special case of the block successive upper-bound minimization (BSUM) approach [35]. Therefore, Theorem 2 guarantees the convergence of the proposed methods.

Constraining the goodness of fit

In some practical applications, the goodness of fit level may be known *a-priori*. In these cases, we may be interested in finding the sparsest representation of the data for a given goodness of fit level. In particular, for a given $\alpha > 0$, we consider

$$\min_{\mathbf{A},\mathbf{X}} \ \|\mathbf{X}\|_1 \quad \text{s.t. } \mathrm{d}(\mathbf{Y},\mathbf{A},\mathbf{X}) \leq \alpha,\ \mathbf{A} \in \mathcal{A},\ \mathbf{X} \in \mathcal{X}. \tag{3.53}$$

For example, when the noise level is known, the goodness of fit function can be set as $\mathrm{d}(\mathbf{Y},\mathbf{A},\mathbf{X}) = \|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F^2$. We propose an efficient method (Algorithm 20) to solve (3.53), where the constant τ_x is chosen according to criterion in (3.48).

The convergence of Algorithm 20 is guaranteed in light of the following theorem.

Algorithm 20 The proposed algorithm for solving (3.53)

initialize $\mathbf{A}$ randomly s.t. $\mathbf{A} \in \mathcal{A}$ and find a feasible $\mathbf{X}$
repeat
  $\bar{\mathbf{X}} \leftarrow \mathbf{X}$
  $\mathbf{X} \leftarrow \arg\min_{\mathbf{X}\in\mathcal{X}} \ \|\mathbf{X}\|_1 \ \text{ s.t. } \mathrm{d}_1(\mathbf{Y},\mathbf{A},\bar{\mathbf{X}}) + \langle \nabla_{\mathbf{X}} \mathrm{d}_1(\mathbf{Y},\mathbf{A},\bar{\mathbf{X}}), \mathbf{X} - \bar{\mathbf{X}}\rangle + \frac{\tau_x}{2}\|\mathbf{X} - \bar{\mathbf{X}}\|_F^2 + \mathrm{d}_2(\mathbf{X}) \leq \alpha$
  $\mathbf{A} \leftarrow \arg\min_{\mathbf{A}\in\mathcal{A}} \ \mathrm{d}(\mathbf{Y},\mathbf{A},\mathbf{X})$
until some convergence criterion is met

Theorem 22 *Assume that* $(\bar{\mathbf{X}},\bar{\mathbf{A}})$ *is a limit point of the iterates generated by Algorithm 20. Furthermore, assume that the subproblem for updating* $\mathbf{X}$ *is strictly feasible at* $(\bar{\mathbf{X}},\bar{\mathbf{A}})$*, i.e., there exists* $\tilde{\mathbf{X}} \in \mathcal{X}$ *such that* $\mathrm{d}_1(\mathbf{Y},\bar{\mathbf{A}},\bar{\mathbf{X}}) + \langle \nabla_{\mathbf{X}} \mathrm{d}_1(\mathbf{Y},\bar{\mathbf{A}},\bar{\mathbf{X}}), \tilde{\mathbf{X}} - \bar{\mathbf{X}}\rangle + \frac{\tau_x}{2}\|\tilde{\mathbf{X}} - \bar{\mathbf{X}}\|_F^2 + \mathrm{d}_2(\tilde{\mathbf{X}}) < \alpha$*. Then* $(\bar{\mathbf{X}},\bar{\mathbf{A}})$ *is a stationary point of* (3.53).

This theorem is the result of Theorem 1.

3.2.5 Online Dictionary Learning

Consider the online/stochastic dictionary learning problem [175]: Given a random signal $\mathbf{y} \in \mathbb{R}^n$ drawn from a distribution $P_Y(\mathbf{y})$, we are interested in finding a dictionary $\mathbf{A} \in \mathbb{R}^{n\times k}$ so that the empirical cost function

$$f(\mathbf{A}) \triangleq \mathbb{E}_{\mathbf{y}}\left[g(\mathbf{A},\mathbf{y})\right]$$

is minimized over the feasible set $\mathcal{A}$; see [22,166,175]. The loss function $g(\mathbf{A}, \mathbf{y})$ measures the fitting error of the dictionary $\mathbf{A}$ to the signal $\mathbf{y}$. Most of the classical and modern loss functions can be represented in the form of

$$g(\mathbf{A}, \mathbf{y}) \triangleq \min_{\mathbf{x} \in \mathcal{X}} \; h(\mathbf{x}, \mathbf{A}, \mathbf{y}), \tag{3.54}$$

where $\mathcal{X} \subseteq \mathbb{R}^k$ and $h(\mathbf{x}, \mathbf{A}, \mathbf{y})$ is a convex function in $\mathbf{x}$ and $\mathbf{A}$ separately. For example, by choosing $h(\mathbf{x}, \mathbf{A}, \mathbf{y}) = \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1$, we obtain the sparse dictionary learning problem; see [175]. Notice that this problem is different than the online sparse recovery problem where the dictionary atoms are known but the overall problem is stochastic; see [176–178].

In order to apply the SSUM framework to the online dictionary learning problem, we need to choose an appropriate approximation function $\hat{g}(\cdot)$. To this end, let us define

$$\hat{g}(\mathbf{A}, \bar{\mathbf{A}}, \mathbf{y}) = h(\bar{\mathbf{x}}, \mathbf{A}, \mathbf{y}) + \frac{\gamma}{2}\|\mathbf{A} - \bar{\mathbf{A}}\|_2^2,$$

where

$$\bar{\mathbf{x}} \triangleq \arg\min_{\mathbf{x} \in \mathcal{X}} h(\mathbf{x}, \mathbf{A}, \mathbf{y}).$$

Clearly, we have

$$\hat{g}(\bar{\mathbf{A}}, \bar{\mathbf{A}}, \mathbf{y}) = h(\bar{\mathbf{x}}, \bar{\mathbf{A}}, \mathbf{y}) = \min_{\mathbf{x} \in \mathcal{X}} h(\mathbf{x}, \bar{\mathbf{A}}, \mathbf{y}) = g(\bar{\mathbf{A}}, \mathbf{y}),$$

and

$$\hat{g}(\mathbf{A}, \bar{\mathbf{A}}, \mathbf{y}) \geq h(\bar{\mathbf{x}}, \mathbf{A}, \mathbf{y}) \geq g(\mathbf{A}, \mathbf{y}).$$

Furthermore, if we assume that the solution of (3.54) is unique, the function $g(\cdot)$ is smooth due to Danskin's Theorem [13]. Moreover, the function $\hat{g}(\mathbf{A}, \bar{\mathbf{A}}, \mathbf{y})$ is strongly convex in $\mathbf{A}$. In addition, if we assume that the feasible set $\mathcal{A}$ is bounded and the signal vector $\mathbf{y}$ lies in a bounded set $\mathcal{Y}$, the assumptions of the SSUM algorithm are satisfied as well. Hence the SSUM algorithm is applicable to the online dictionary learning problem.

Remark 5 *Choosing* $h(\mathbf{x}, \mathbf{A}, \mathbf{y}) = \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1$ *and* $\gamma = 0$ *leads to the online*

sparse dictionary learning algorithm in [175]. Notice that the authors of [175] had to assume the uniform strong convxity of $\frac{1}{2}\|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2$ *for all* $\mathbf{x}\in\mathcal{X}$ *since they did not consider the quadratic proximal term* $\gamma\|\mathbf{A}-\bar{\mathbf{A}}\|^2$.

3.3 Other Applications

3.3.1 Proximal Minimization Algorithm

The classical proximal minimization algorithm (see, e.g., [85, Section 3.4.3]) obtains a solution of the problem $\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$ by solving an equivalent problem

$$\min_{\mathbf{x}\in\mathcal{X},\mathbf{y}\in\mathcal{X}} f(\mathbf{x})+\frac{1}{2c}\|\mathbf{x}-\mathbf{y}\|_2^2, \tag{3.55}$$

where $f(\cdot)$ is a convex function, $\mathcal{X}$ is a closed convex set, and $c>0$ is a scalar parameter. The equivalent problem (3.55) is attractive in that it is strongly convex in both x and y (but not jointly) so long as $f(x)$ is convex. This problem can be solved by performing the following two steps in an alternating fashion

$$\mathbf{x}^{r+1}=\arg\min_{\mathbf{x}\in\mathcal{X}}\left\{f(x)+\frac{1}{2c}\|\mathbf{x}-\mathbf{y}^r\|_2^2\right\} \tag{3.56}$$

$$\mathbf{y}^{r+1}=\mathbf{x}^{r+1}. \tag{3.57}$$

Equivalently, let $u(\mathbf{x};\mathbf{x}^r)\triangleq f(\mathbf{x})+\frac{1}{2c}\|\mathbf{x}-\mathbf{x}^r\|_2^2$, then the iteration (3.56)–(3.57) can be written as

$$\mathbf{x}^{r+1}=\arg\min_{\mathbf{x}\in\mathcal{X}} u(\mathbf{x},\mathbf{x}^r). \tag{3.58}$$

It can be straightforwardly checked that for all $\mathbf{x},\mathbf{x}^r\in\mathcal{X}$, the function $u(\mathbf{x},\mathbf{x}^r)$ serves as an upper bound for the function $f(\mathbf{x})$. It is not hard to check that the the convergence of the proximal minimization procedure can be obtained from Theorem 2.

The proximal minimization algorithm can be generalized in the following way. Consider the problem

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}_1, \cdots, \mathbf{x}_n) \\ \text{s.t.} \quad & \mathbf{x}_i \in \mathcal{X}_i,\ i = 1, \cdots, n, \end{aligned} \tag{3.59}$$

where $\{\mathcal{X}_i\}_{i=1}^n$ are closed convex sets, $f(\cdot)$ is convex in each of its block components, but not necessarily strictly convex. A straightforward application of the BCD procedure may fail to find a stationary solution for this problem, as the per-block subproblems may contain multiple solutions. Alternatively, we can consider an *alternating proximal minimization* algorithm [14, 179], in each iteration of which the following subproblem is solved

$$\begin{aligned} \min_{\mathbf{x}_i} \quad & f(\mathbf{x}_1^r, \ldots, \mathbf{x}_{i-1}^r, \mathbf{x}_i, \mathbf{x}_{i+1}^r, \ldots, \mathbf{x}_n^r) + \frac{1}{2c}\|\mathbf{x}_i - \mathbf{x}_i^r\|_2^2 \\ \text{s.t.} \quad & \mathbf{x}_i \in \mathcal{X}_i. \end{aligned} \tag{3.60}$$

It is not hard to see that this subproblem always admits a unique solution, as the objective is a strictly convex function of $\mathbf{x}_i$. Let $u_i(\mathbf{x}_i, \mathbf{x}^r) \triangleq f(\mathbf{x}_1^r, \cdots, \mathbf{x}_i, \cdots \mathbf{x}_n^r) + \frac{1}{2c}\|\mathbf{x}_i - \mathbf{x}_i^r\|_2^2$. Again for each $\mathbf{x}_i \in \mathcal{X}_i$ and $\mathbf{x}^r \in \prod_j \mathcal{X}_j$, the function $u_i(\mathbf{x}_i, \mathbf{x}^r)$ is an upper bound of the original objective $f(\mathbf{x})$. Moreover, all the conditions in Assumption 2 are satisfied. Utilizing Theorem 2, we conclude that the alternating proximal minimization algorithm must converge to a stationary solution of the problem (3.59). Moreover, our result extends those in [14] to the case of nonsmooth objective function as well as the case with iteration-dependent coefficient c. The latter case, which was also studied in the contemporary work [15], will be demonstrated in an example for tensor decomposition shortly. It is also worth noting that the convergence of the alternating proximal minimization algorithm is also studied in [180] for Kurdyka-Lojasiewicz functions.

3.3.2 Proximal Splitting Algorithm

The proximal splitting algorithm (see, e.g., [181]) for nonsmooth optimization is also a special case of the BSUM algorithm. Consider the following problem

$$\min_{\mathbf{x}\in\mathcal{X}} f_1(\mathbf{x}) + f_2(\mathbf{x}) \tag{3.61}$$

where $\mathcal{X}$ is a closed and convex set. Furthermore, f_1 is convex and lower semicontinuous; f_2 is convex and has Lipschitz continuous gradient, i.e., $\|\nabla f_2(\mathbf{x}) - \nabla f_2(\mathbf{y})\| \leq \beta\|\mathbf{x}-\mathbf{y}\|$, $\forall\ \mathbf{x},\mathbf{y} \in \mathcal{X}$ and for some $\beta > 0$.

Define the proximity operator $\text{prox}_{f_i} : \mathcal{X} \to \mathcal{X}$ as

$$\text{prox}_{f_i}(\mathbf{x}) = \arg\min_{\mathbf{y}\in\mathcal{X}} f_i(\mathbf{y}) + \frac{1}{2}\|\mathbf{x}-\mathbf{y}\|^2. \tag{3.62}$$

The following forward-backward splitting iteration can be used to obtain a solution for problem (3.61) [181, 182]:

$$\mathbf{x}^{r+1} = \underbrace{\text{prox}_{\gamma f_1}}_{\text{backward step}} \underbrace{(\mathbf{x}^r - \gamma\nabla f_2(\mathbf{x}^r))}_{\text{forward step}} \tag{3.63}$$

where $\gamma \in [\epsilon, 2/\beta - \epsilon]$ with $\epsilon \in]0, \min\{1, 1/\beta\}[$. Define

$$u(\mathbf{x}, \mathbf{x}^r) \triangleq f_1(\mathbf{x}) + \frac{1}{2\gamma}\|\mathbf{x}-\mathbf{x}^r\|^2 + \langle \mathbf{x}-\mathbf{x}^r, \nabla f_2(\mathbf{x}^r)\rangle + f_2(\mathbf{x}^r). \tag{3.64}$$

We first show that the iteration (3.63) is equivalent to the following iteration

$$\mathbf{x}^{r+1} = \arg\min_{\mathbf{x}\in\mathcal{X}} u(\mathbf{x}, \mathbf{x}^r). \tag{3.65}$$

From the definition of the prox operation, we have

$$\begin{aligned}
\text{prox}_{\gamma f_1}(\mathbf{x}^r - \gamma\nabla f_2(\mathbf{x}^r)) &= \arg\min_{\mathbf{x}\in\mathcal{X}} \gamma f_1(\mathbf{x}) + \frac{1}{2}\|\mathbf{x}-\mathbf{x}^r+\gamma\nabla f_2(\mathbf{x}^r)\|_2^2 \\
&= \arg\min_{\mathbf{x}\in\mathcal{X}} f_1(\mathbf{x}) + \frac{1}{2\gamma}\|\mathbf{x}-\mathbf{x}^r\|_2^2 + \langle \mathbf{x}-\mathbf{x}^r, \nabla f_2(\mathbf{x}^r)\rangle \\
&= \arg\min_{\mathbf{x}\in\mathcal{X}} u(\mathbf{x}, \mathbf{x}^r).
\end{aligned}$$

We then show that $u(\mathbf{x}, \mathbf{x}^r)$ is an upper bound of the original function $f_1(\mathbf{x}) + f_2(\mathbf{x})$, for all $\mathbf{x}, \mathbf{x}^r \in \mathcal{X}$. Note that from the well known Descent Lemma [13, Proposition A.32], we have that

$$\begin{aligned} f_2(\mathbf{x}) &\leq f_2(\mathbf{x}^r) + \frac{\beta}{2}\|\mathbf{x} - \mathbf{x}^r\|^2 + \langle \mathbf{x} - \mathbf{x}^r, \nabla f_2(\mathbf{x}^r)\rangle \\ &\leq f_2(\mathbf{x}^r) + \frac{1}{2\gamma}\|\mathbf{x} - \mathbf{x}^r\|^2 + \langle \mathbf{x} - \mathbf{x}^r, \nabla f_2(\mathbf{x}^r)\rangle \end{aligned}$$

where the second inequality is from the definition of γ. This result implies that $u(\mathbf{x}, \mathbf{y}) \geq f_1(\mathbf{x}) + f_2(\mathbf{x}),\ \forall\ \mathbf{x}, \mathbf{y} \in \mathcal{X}$. Moreover, we can again verify that all the other conditions BSUM are true. Consequently, we conclude that the forward-backward splitting algorithm is a special case of the BSUM algorithm.

Similar to the previous example, we can generalize the forward-backward splitting algorithm to the problem with multiple block components. Consider the following problem

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} f_i(\mathbf{x}_i) + f_{n+1}(\mathbf{x}_1, \cdots, \mathbf{x}_n) \\ \text{s.t.} \quad & \mathbf{x}_i \in \mathcal{X}_i, i = 1, \cdots, n \end{aligned} \tag{3.66}$$

where $\{\mathcal{X}_i\}_{i=1}^n$ are a closed and convex sets. Each function $f_i(\cdot)$, $i = 1, \cdots n$ is convex and lower semicontinuous w.r.t. $\mathbf{x}_i$; $f_{n+1}(\cdot)$ is convex and has Lipschitz continuous gradient w.r.t. each of the component $\mathbf{x}_i$, i.e., $\|\nabla f_{n+1}(\mathbf{x}) - \nabla f_{n+1}(\mathbf{y})\| \leq \beta_i \|\mathbf{x}_i - \mathbf{y}_i\|$, $\forall\ \mathbf{x}_i, \mathbf{y}_i \in \mathcal{X}_i, i = 1, \cdots, n$. Then the following block forward-backward splitting algorithm can be shown as a special case of the BSUM algorithm, and consequently converges to a stationary solution of the problem (3.66)

$$\mathbf{x}_i^{r+1} = \text{prox}_{\gamma f_i}(\mathbf{x}_i^r - \gamma \nabla_{\mathbf{x}_i} f_{n+1}(\mathbf{x}^r)), \quad i = 1, 2, ..., n,$$

where $\gamma \in [\epsilon_i, 2/\beta_i - \epsilon_i]$ with $\epsilon_i \in]0, \min\{1, 1/\beta_i\}[$. To the best of our knowledge, the convergence of the block forward-backward splitting method has not been studied before for non-smooth non-convex problems.

3.3.3 CANDECOMP/PARAFAC Decomposition of Tensors

Another application of the proposed method is in CANDECOMP/PARAFAC (CP) decomposition of tensors, which is useful in various practical problems; see, e.g., [183–185]. Given a tensor $\mathfrak{X} \in \mathbb{R}^{m_1 \times m_2 \times \ldots \times m_n}$ of order n, the idea of CP decomposition is to write the tensor as the sum of rank-one tensors:

$$\mathfrak{X} = \sum_{r=1}^{R} \mathfrak{X}_r,$$

where $\mathfrak{X}_r = a_{1r} \circ a_{2r} \circ \ldots \circ a_{nr}$ and $a_{ir} \in \mathbb{R}^{m_i}$. Here the notation " $\circ$ " denotes the outer product. It is also worth noting that unlike the matrices, the CP decompositions of tensors are often unique; see [186–191].

In general, finding the CP decomposition of a given tensor is NP-hard [192]. In practice, one of the most widely accepted algorithms for computing the CP decomposition of a tensor is the Alternating Least Squares (ALS) algorithm [193–195]. The ALS algorithm proposed in [196, 197] is in essence a BCD method. For ease of presentation, we will present the ALS algorithm only for tensors of order three.

Let $\mathfrak{X} \in \mathbb{R}^{I \times J \times K}$ be a third order tensor. Let $(A; B; C)$ represent the following decomposition

$$(A; B; C) \triangleq \sum_{r=1}^{R} a_r \circ b_r \circ c_r,$$

where a_r (resp. b_r and c_r) is the r-th column of A (resp. B and C). The ALS algorithm minimizes the difference between the original and the reconstructed tensors

$$\min_{A,B,C} \quad \|\mathfrak{X} - (A; B; C)\|, \tag{3.67}$$

where $A \in \mathbb{R}^{I \times R}$, $B \in \mathbb{R}^{J \times R}$, $C \in \mathbb{R}^{K \times R}$, and R is the rank of the tensor.

The ALS approach is a special case of the BCD algorithm in which the three blocks of variables A, B, and C are cyclically updated. In each step of the computation when two blocks of variables are held fixed, the subproblem becomes the quadratic least squares

problem and admits closed form updates (see [193]).

One of the well-known drawbacks of the ALS algorithm is the *swamp* effect where the objective value remains almost constant for many iterations before starting to decrease again. Navasca *et al.* in [198] observed that adding a proximal term in the algorithm could help reducing the swamp effect. More specifically, at each iteration r the algorithm proposed in [198] solves the following problem for updating the variables:

$$\|\mathfrak{X} - (A; B; C)\|^2 + \lambda\|A - A^r\|^2 + \lambda\|B - B^r\|^2 + \lambda\|C - C^r\|^2, \tag{3.68}$$

where $\lambda \in \mathbb{R}$ is a positive constant. As discussed before, this proximal term has been considered in different optimization contexts and its convergence has been already showed in [14]. An interesting numerical observation in [198] is that decreasing the value of λ during the algorithm can noticeably improve the convergence of the algorithm. Such iterative decrease of λ can be accomplished in a number of different ways. Our numerical experiments show that the following simple approach to update λ can significantly improve the convergence of the ALS algorithm and substantially reduce the swamp effect:

$$\lambda^r = \lambda_0 + \lambda_1 \frac{\|\mathfrak{X} - (A^r; B^r; C^r)\|}{\|\mathfrak{X}\|}, \tag{3.69}$$

where λ^r is the proximal coefficient λ at iteration r. Theorem 2 implies the convergence is guaranteed even with this update rule of λ, whereas the convergence result of [14] does not apply in this case since the proximal coefficient is changing during the iterations.

Figure 3.2 shows the performance of different algorithms for the example given in [198] where the tensor $\mathfrak{X}$ is obtained from the decomposition

$$A = \begin{bmatrix} 1 & \cos\theta & 0 \\ 0 & \sin\theta & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 3 & \sqrt{2}\cos\theta & 0 \\ 0 & \sin\theta & 1 \\ 0 & \sin\theta & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

To generate the plots, we have used a MATLAB script running on a PC with 4GB RAM memory and a dual core 2.7 GHz CPU. In Figure 3.2, the vertical axis is the value of the

objective function where the horizontal axis is the iteration number. In this plot, *ALS* is the classical alternating least squares algorithm. The curve for *Constant Proximal* shows the performance of the BSUM algorithm when we use the objective function in (3.68) with $\lambda = 0.1$. The curve for *Diminishing Proximal* shows the performance of block coordinate descent method on (3.68) where the weight λ decreases iteratively according to (3.69) with $\lambda_0 = 10^{-7}, \lambda_1 = 0.1$. The other two curves *MBI* and *MISUM* correspond to the maximum block improvement algorithm and the MISUM algorithm. In the implementation of the MISUM algorithm, the proximal term is of the form in (3.68) and the weight λ is updated based on (3.69).

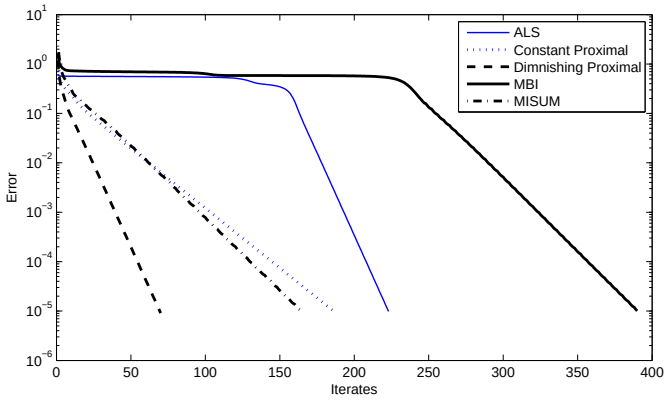


Figure 3.2: BSUM convergence for tensor decomposition (small scale example)

Table 3.1 represents the average number of iterations required to get an objective value less than $\epsilon = 10^{-5}$ for different algorithms. The average is taken over 1000 Monte-Carlo runs over different initializations. The initial points are generated randomly where the components of the variables A, B, and C are drawn independently from the uniform distribution over the unit interval $[0, 1]$. As it can be seen, adding a diminishing proximal term significantly improves the convergence speed of the ALS algorithm.

Figure 3.3 illustrates the performance of different algorithms for the case of $I = J = K = R = 100$. In this experiment, we set $\lambda_0 = 10^{-1}$ and $\lambda_1 = 1$. As it can be seen from the figure, adding the proximal terms reduces the swamp effect.

Algorithm	Average number of iterations for convergence
ALS	277
Constant Proximal	140
Diminishing Proximal	78
MBI	572
MISUM	175

Table 3.1: Average number of iterations for convergence

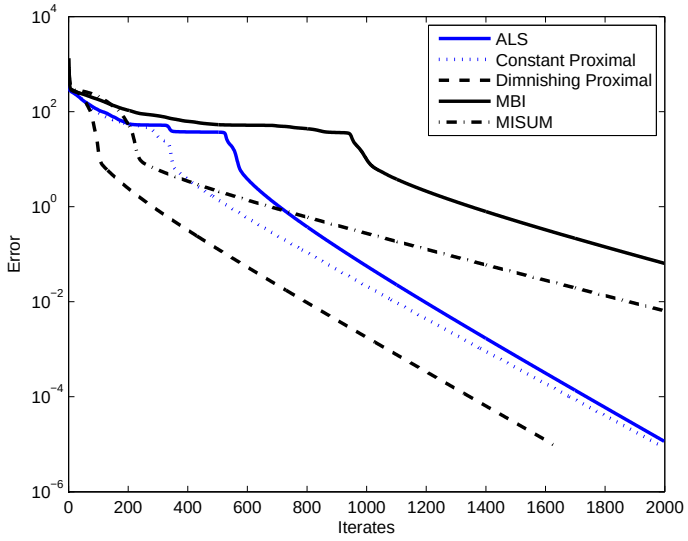


Figure 3.3: Convergence of different algorithms for tensor decomposition

3.3.4 Expectation Maximization Algorithm

The expectation maximization algorithm (EM) in [12] is an iterative procedure for maximum likelihood estimation when some of the random variables are unobserved/hidden. Let w be the observed random vector which is used for estimating the value of θ. The maximum likelihood estimate of θ can be given as

$$\hat{\theta}_{\mathrm{ML}} = \arg\max_{\theta} \ln p(w|\theta). \tag{3.70}$$

Let the random vector z be the hidden/unobserved variable. The EM algorithm starts from an initial estimate θ^0 and generates a sequence $\{\theta^r\}$ by repeating the following steps:

- E-Step: Calculate $g(\theta, \theta^r) \triangleq \mathbb{E}_{z|w,\theta^r}\{\ln p(w, z|\theta)\}$
- M-Step: $\theta^{r+1} = \arg\max_\theta \ g(\theta, \theta^r)$

The EM-algorithm can be viewed as a special case of SUM algorithm [199]. In fact, we are interested in solving the following optimization problem

$$\min_\theta \quad -\ln p(w|\theta).$$

The objective function could be written as

$$\begin{aligned}
-\ln p(w|\theta) &= -\ln \ \mathbb{E}_{z|\theta}\, p(w|z,\theta) \\
&= -\ln \ \mathbb{E}_{z|\theta}\left[\frac{p(z|w,\theta^r)p(w|z,\theta)}{p(z|w,\theta^r)}\right] \\
&= -\ln \ \mathbb{E}_{z|w,\theta^r}\left[\frac{p(z|\theta)p(w|z,\theta)}{p(z|w,\theta^r)}\right] \\
&\leq -\mathbb{E}_{z|w,\theta^r} \ln\left[\frac{p(z|\theta)p(w|z,\theta)}{p(z|w,\theta^r)}\right] \\
&= -\mathbb{E}_{z|w,\theta^r} \ln p(w,z|\theta) + \mathbb{E}_{z|w,\theta^r} \ln p(z|w,\theta^r) \\
&\triangleq u(\theta,\theta^r),
\end{aligned}$$

where the inequality is due to the Jensen's inequality and the third equality follows from a simple change of the order of integration for the expectation. Since $\mathbb{E}_{z|w,\theta^r} \ln p(z|w,\theta^r)$ is not a function of θ, the M-step in the EM-algorithm can be written as

$$\theta^{r+1} = \arg\max_\theta u(\theta,\theta^r).$$

Furthermore, it is not hard to see that $u(\theta^r,\theta^r) = -\ln p(w|\theta^r)$. Therefore, under the smoothness assumption, it is not hard to check that the BSUM assumptions are satisfied. As an immediate consequence, the EM-algorithm is a special case of the SUM algorithm. Therefore, our result implies not only the convergence of the EM-algorithm, but also the convergence of the EM-algorithm with Gauss-Seidel/coordinatewise update rule (under the assumptions of Theorem 2). In fact in the block coordinate EM-algorithm (BEM), at each M-step, only one block is updated. More specifically, let $\theta = (\theta_1, \ldots, \theta_n)$ be the unknown parameter. Assume w is the observed vector and z is the hidden/unobserved

variable as before. The BEM algorithm starts from an initial point $\theta^0 = (\theta_1^0, \ldots, \theta_n^0)$ and generates a sequence $\{\theta^r\}$ according to the algorithm in Figure 21.

Algorithm 21 Pseudo code of the BEM algorithm

Initialize with θ^0 and set $r = 0$

repeat

$r = r + 1$, $i = r \bmod n + 1$

E-Step: $g_i(\theta_i, \theta^r) = \mathbb{E}_{z|w,\theta^r}\{\ln p(w, z|\theta_1^r, \ldots, \theta_{i-1}^r, \theta_i, \theta_{i+1}^r, \ldots, \theta_n^r)\}$

M-Step: $\theta_i^{r+1} = \arg\max_{\theta_i} \; g_i(\theta_i, \theta^r)$

until some convergence criterion is met

The motivation behind using the BEM algorithm instead of the EM algorithm could be the difficulties in solving the M-step of EM for the entire set of variables, while solving the same problem per block of variables is easy. As an example, consider the mixture of Gaussian model [200] where different Gaussian distributions have the same mean but different variances. It can be checked that the update rule of EM algorithm [12] cannot be done in closed form. However, fixing the variance, the update rule of the mean could be done in closed form. Furthermore, by fixing the mean, the variance could be updated in closed form and hence the BEM algorithm can be applied. To the best of our knowledge, the BEM algorithm and its convergence behavior have not been analyzed before.

3.3.5 Concave-Convex Procedure/Difference of Convex Functions

A popular algorithm for solving unconstrained problems, which also belongs to the class of successive upper-bound minimization, is the Concave-Convex Procedure (CCCP) introduced in [11]. In CCCP, also known as the difference of convex functions (DC) programming, we consider the unconstrained problem

$$\min_{x \in \mathbb{R}^m} \; f(x),$$

where $f(x) = f_{cve}(x) + f_{cvx}(x), \forall\, x \in \mathbb{R}^m$; where $f_{cve}(\cdot)$ is a concave function and $f_{cvx}(\cdot)$ is convex. The CCCP generates a sequence $\{x^r\}$ by solving the following equation:

$$\nabla f_{cvx}(x^{r+1}) = -\nabla f_{cve}(x^r),$$

which is equivalent to

$$x^{r+1} = \arg\min_x \; g(x, x^r), \tag{3.71}$$

where $g(x, x^r) \triangleq f_{cvx}(x) + (x - x^r)^T \nabla f_{cve}(x^r) + f_{cve}(x^r)$. Clearly, $g(x, x^r)$ is a tight convex upper-bound of $f(x)$ and hence CCCP is a special case of the SUM algorithm and its convergence is guaranteed by the convergence of BSUM under certain assumptions. Furthermore, if the updates are done in a block coordinate manner, the algorithm becomes a special case of BSUM whose convergence is guaranteed by Theorem 2. To the best of our knowledge, the general block coordinate version of CCCP algorithm and its convergence have not been studied before. For applications of the block coordinate version of the constrained CCCP method in various practical problems, the readers are referred to [15, 19, 107, 108, 134–137, 201–211].

3.3.6 Stochastic (Sub-)Gradient Method and its Extensions

In this section, we show that the classical SG method, the incremental gradient method and the stochastic sub-gradient method are special cases of the SSUM method. We also present an extension of these classical methods using the SSUM framework.

To describe the SG method, let us consider a special (unconstrained smooth) case of the optimization problem (2.12), where $g_2 \equiv 0$ and $\mathcal{X} = \mathbb{R}^n$. One of the popular algorithms for solving this problem is the stochastic gradient (also known as stochastic approximation) method. At each iteration r of the stochastic gradient (SG) algorithm, a new realization ξ^r is obtained and x is updated based on the following simple rule [73, 212–214]:

$$x^r \;\leftarrow\; x^{r-1} - \gamma^r \nabla_x g_1(x^{r-1}, \xi^r). \tag{3.72}$$

Here γ^r is the step size at iteration r. Due to its simple update rule, the SG algorithm has been widely used in various applications such as data classification [215, 216], training

multi-layer neural networks [217–220], the expected risk minimization [221], solving least squares in statistics [222], and distributed inference in sensor networks [85, 223, 224]. Also the convergence of the SG algorithm is well-studied in the literature; see, e.g., [73, 214, 225].

The popular incremental gradient method [219–222, 226] can be viewed as a special case of the SG method where the set Ξ is finite. In the incremental gradient methods, a large but finite set of samples Ξ is available and the objective is to minimize the empirical expectation

$$\hat{\mathbb{E}}\{g(x,\xi)\} \;=\; \frac{1}{|\Xi|}\sum_{\xi\in\Xi} g(x,\xi). \tag{3.73}$$

At each iteration r of the incremental gradient method (with random updating order), a new realization $\xi^r \in \Xi$ is chosen randomly and uniformly, and then (3.72) is used to update x. This is precisely the SG algorithm applied to the minimization of (3.73). In contrast to the batch gradient algorithm which requires computing $\sum_{\xi\in\Xi} \nabla_x g(x,\xi)$, the updates of the incremental gradient algorithm are computationally cheaper, especially if $|\Xi|$ is very large.

In general, the convergence of the SG method depends on the proper choice of the step size γ^r. It is known that for the constant step size rule, the SG algorithm might diverge even for a convex objective function; see [219] for an example. There are many variants of the SG algorithm with different step size rules [227]. In the following, we introduce a special form of the SSUM algorithm that can be interpreted as the SG algorithm with diminishing step sizes. Let us define

$$\hat{g}_1(x,y,\xi) = g_1(y,\xi) + \langle \nabla g_1(y,\xi), x-y\rangle + \frac{\alpha}{2}\|x-y\|^2, \tag{3.74}$$

where α is a function of y and is chosen so that $\hat{g}_1(x,y,\xi) \geq g_1(x,\xi)$. One simple choice is $\alpha^r = L$, where L is the Lipschitz constant of $\nabla_x g_1(x,\xi)$. Choosing $\hat{g}_1$ in this way, the assumptions A1-A3 are clearly satisfied. Moreover, the update rule of the SSUM

algorithm becomes

$$x^r \leftarrow \arg\min_x \frac{1}{r}\sum_{i=1}^{r} \hat{g}_1(x, x^{i-1}, \xi^i). \tag{3.75}$$

Checking the first order optimality condition of (3.75), we obtain

$$x^r \leftarrow \frac{1}{\sum_{i=1}^r \alpha^i}\left(\sum_{i=1}^{r}(\alpha^i x^{i-1} - \nabla_x g_1(x^{i-1}, \xi^i))\right). \tag{3.76}$$

Rewriting (3.76) in a recursive form yields

$$x^r \leftarrow x^{r-1} - \frac{1}{\sum_{i=1}^r \alpha^i}\nabla_x g_1(x^{r-1}, \xi^r), \tag{3.77}$$

which can be interpreted as the stochastic gradient method (3.72) with $\gamma^r = \frac{1}{\sum_{i=1}^r \alpha_i}$. Notice that the simple constant choice of $\alpha^i = L$ yields $\gamma^r = \frac{1}{rL}$, which gives the most popular diminishing step size rule of the SG method.

Remark 6 *When $\mathcal{X}$ is bounded and using the approximation function in* (3.74)*, we see that the SSUM algorithm steps become*

$$z^r = \frac{1}{\sum_{i=1}^r \alpha^i}\left(\sum_{i=1}^{r-1}\alpha^i z^{r-1} + \alpha^r x^{r-1} - \nabla_x g_1(x^{r-1}, \xi^r)\right),$$
$$x^r = \Pi_{\mathcal{X}}(z^r),$$

where $\Pi_{\mathcal{X}}(\cdot)$ signifies the projection operator to the constraint set $\mathcal{X}$. Notice that this update rule is different from the classical SG method as it requires generating the auxiliary iterates $\{z^r\}$ which may not lie in the feasible set $\mathcal{X}$.

It is also worth noting that in the presence of the non-smooth part of the objective function, the SSUM algorithm becomes different from the classical stochastic subgradient method [73, 212–214]. To illustrate the ideas, let us consider a simple deterministic nonsmooth function $g_2(x)$ to be added to the objective function. The resulting

optimization problem becomes

$$\min_{x} \quad \mathbb{E}\left[g_1(x,\xi)\right] + g_2(x).$$

Using the approximation introduced in (3.74), the SSUM update rule can be written as

$$x^r \leftarrow \arg\min_{x} \quad \frac{1}{r}\sum_{i=1}^{r} \hat{g}_1(x, x^{i-1}, \xi^i) + g_2(x). \tag{3.78}$$

Although this update rule is similar to the (regularized) dual averaging method [228,229] for convex problems, its convergence is guaranteed even for the nonconvex nonsmooth objective function under the assumptions of Theorem 12. Moreover, similar to the (regularized) dual averaging method, the steps of the SSUM algorithm are computationally cheap for some special nonsmooth functions. As an example, let us consider the special non-smooth function $g_2(x) \triangleq \lambda\|x\|_1$. Setting $\alpha^r = L$, the first order optimality condition of (3.78) yields the following update rule:

$$\begin{aligned} z^{r+1} &\leftarrow \frac{rz^r + x^r - \frac{1}{L}\nabla g_1(x^r, \xi^{r+1})}{r+1}, \\ x^{r+1} &\leftarrow \mathrm{shrink}_{\frac{\lambda}{L}}(z^{r+1}), \end{aligned} \tag{3.79}$$

where $\{z^{r+1}\}_{r=1}^{\infty}$ is an auxiliary variable sequence and $\mathrm{shrink}_{\tau}(z)$ is the soft shrinkage operator defined as

$$\mathrm{shrink}_{\tau}(z) = \begin{cases} z-\tau & z \geq \tau \\ 0 & \tau \geq z \geq -\tau \\ z+\tau & z \leq -\tau \end{cases}.$$

Notice that the algorithm obtained in (3.79) is different from the existing stochastic subgradient algorithm and the stochastic proximal gradient algorithm [8, 226]; furthermore, if the conditions in Theorem 12 is satisfied, its convergence is guaranteed even for nonconvex objective functions.

To see other applications of the SSUM framework, see [230, 231].

Chapter 4

Numerical Experiments

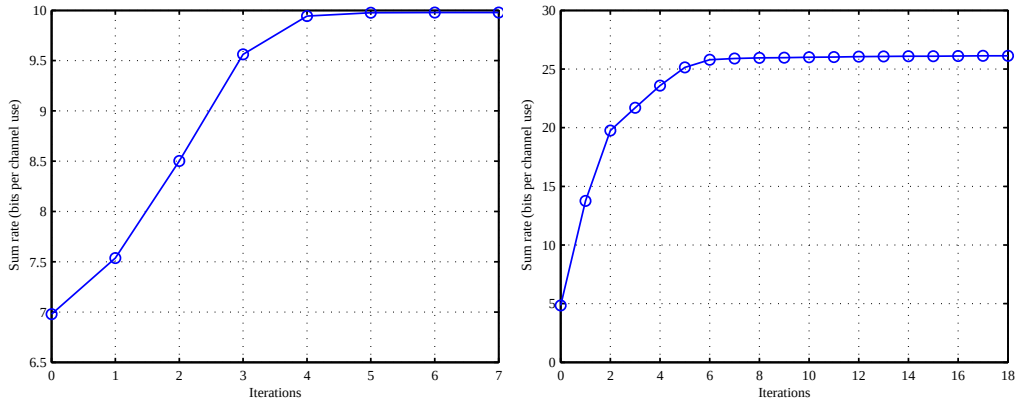
In this chapter, we evaluate the numerical performance of the proposed algorithms based on the successive convex approximation idea. A special emphasize will be given to the interference management problem in wireless communication and dictionary learning problem for sparse recovery.

4.1 Interference Management in Wireless Networks

4.1.1 Beamforming in Wireless Networks

In this subsection section, we numerically evaluate the performance of the proposed WMMSE algorithm introduced in subsection 3.1.2. For ease of comparison with existing algorithms, all simulations are conducted for MIMO interference channel (the degenerate MIMO-IBC case with one receiver per cell). The weights $\{\alpha_{i_k}\}$ and noise powers $\{\sigma_{i_k}^2\}$ are set equally for all users. The transmit power budget is set to P for all transmitters, where $P = 10^{\frac{\text{SNR}}{10}}$. Moreover, all transmitters (or receivers) are assumed to have the same number of antennas, denoted by T (or R). We use uncorrelated fading channel model with channel coefficients generated from the complex Gaussian distribution $\mathcal{CN}(0, 1)$.

Fig. 4.1(a) and Fig. 4.1(b) illustrate the convergence behavior of the WMMSE algorithm for the case of SNR = 25(dB). These plots show that the WMMSE algorithm converges in few steps and it does so monotonically. Figure 4.1(a) uses the parameters



(a) (b)

Figure 4.1: WMMSE algorithm: (a) SISO-IFC (b) MIMO-IFC.

$K = 3$, $\epsilon = 1e-3$ in a SISO channel, while figure 4.1(b) is for MIMO interference channel with $K = 4$, $T = 3$, $R = 2$, $\epsilon = 1e-2$.

Fig. 4.2 plots the average sum-rate versus the SNR for the SISO interference channel case. Each curve is averaged over 100 random channel realizations. The term "WMMSE" represents running the WMMSE algorithm once while "WMMSE_10rand_int" means running the WMMSE algorithm 10 times with different initialization and then keeping the best result. The terms "ILA" and "ILA_10rand_int" are similarly defined. It can be observed that the WMMSE algorithm and the ILA algorithm yield almost the same performance. The performance of the brute force search method (exponential complexity) is provided in the three users case as a benchmark. We can see that the gap between the performance of the WMMSE algorithm and the optimal performance is small and slowly increasing with SNR. However, repeating the WMMSE algorithm ten times can close this performance gap.

Similar observations can be made for the MIMO interference channel case, as Fig. 4.3 illustrates. As a comparison, we also provide the performance of the MMSE algorithm [112] which has been shown to perform better than the interference alignment method [232]. Obviously, the WMMSE algorithm significantly outperforms the MMSE algorithm in terms of the achieved sum-rate. This is due to the use of iterative weighting matrices $\{\mathbf{W}_{i_k}\}$.

Although the ILA algorithm yields almost the same performance as the WMMSE

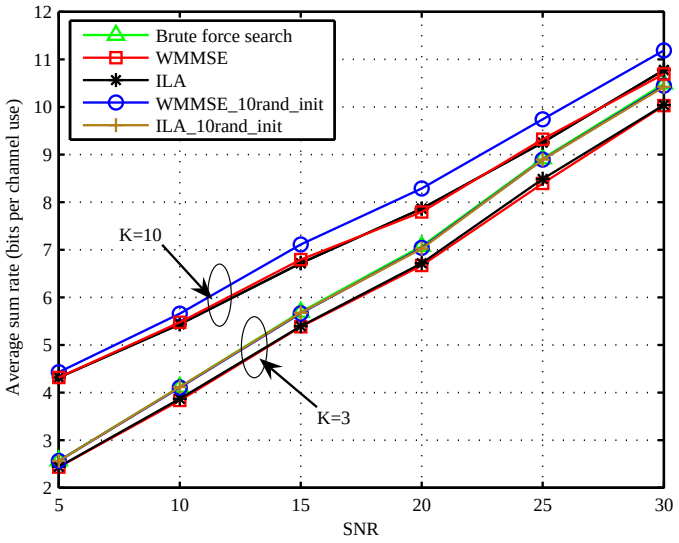


Figure 4.2: Average sum-rate versus SNR in the SISO IFC case.

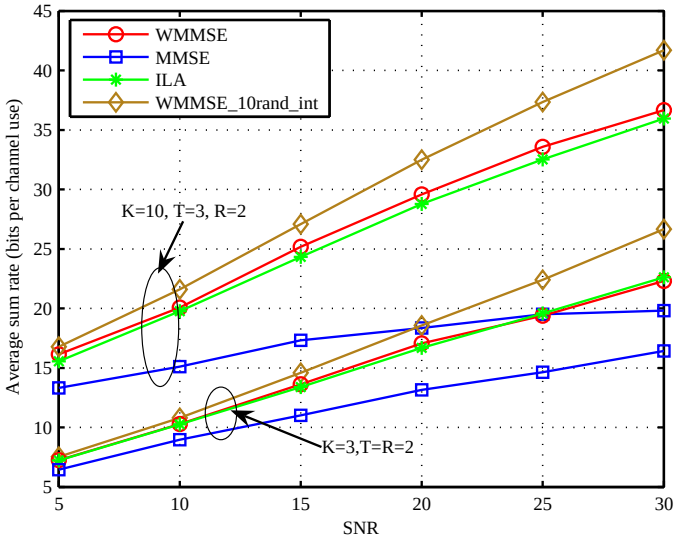


Figure 4.3: Average sum-rate versus SNR in the MIMO IFC case.

algorithm in terms of the sum rate, it has higher complexity. Figure 4.4 represents the average CPU time comparison of the two algorithms under the same termination criterion. The number of transmit antennas is 3, while the number of receive antennas is 2. It can be observed that the WMMSE algorithm significantly outperform the ILA algorithm when the number of users is large.

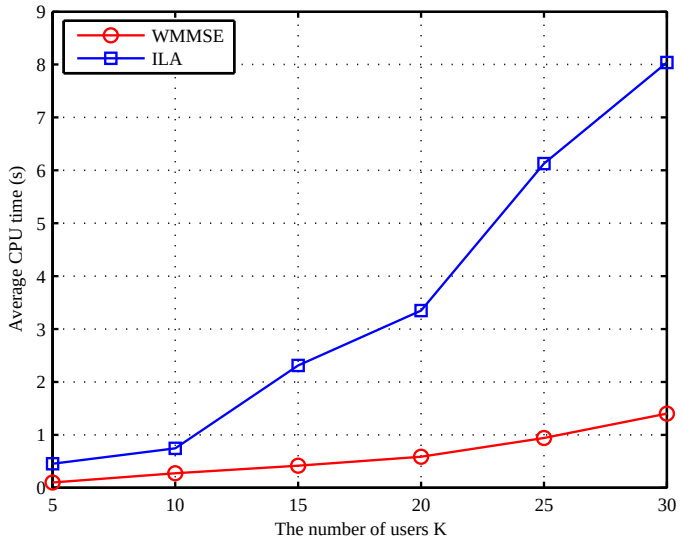


Figure 4.4: Average CPU time versus the number of users in the MIMO IFC case.

4.1.2 Joint Beamforming and Scheduling

Here we present some simulations comparing different beamforming/scheduling methods with the one proposed in subsection 3.1.3. In our numerical experiments, the path loss of the channel coefficients are generated using the 3GPP (TR 36.814) evaluation methodology [233], with the additional standard Rayleigh fading. As such, the channel taps are drawn randomly from appropriately scaled Rayleigh distributions. Our first sets of numerical experiments are obtained via 5 rounds of channel realizations (5 Monte Carlo runs of channel generation). We consider a 19-hexagonal wrap-around cell layout (see Figure 4.5). Each base station has three sectors, i.e., essentially $19 \times 3 = 57$ base stations. These base stations serve a total of 285 users in the system. Each base station is equipped with M antennas while each user is equipped with N antennas. We consider the thermal noise figure of 8.3dB and the bandwidth of 15KHz for each tone. Therefore,

the total noise power per tone is -124dBm/Hz/tone. The transmit power is 46dBm for 600 tones, i.e., 18.21dBm/tone and we run our algorithm on a single frequency tone of the OFDM system. We initialize our algorithm with a random transmit and receive beamformer. Furthermore, we adopt the geometric mean utility function (i.e., proportional fairness) in our experiments, that is, $u_{i_k} = \log(\cdot)$ in (3.25).

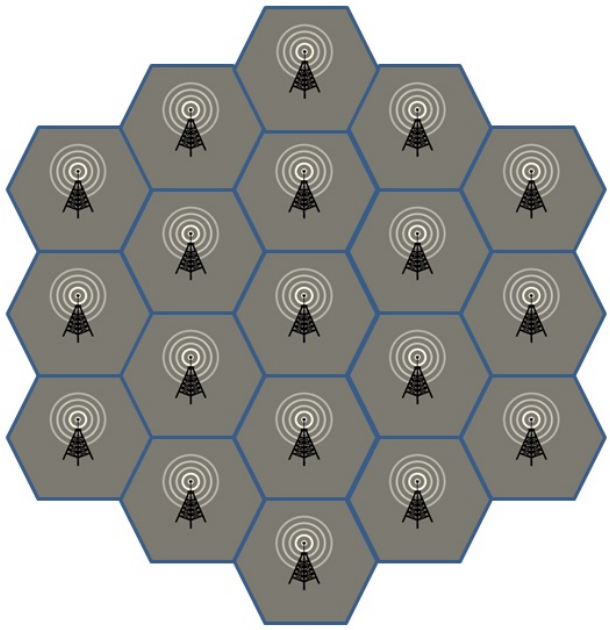


Figure 4.5: 19-hexagonal wrap around cell layout

In the first numerical experiment, we consider $N = 2$ antennas at the receivers and $M = 4$ antennas at the base stations; and we compare different transceiver design algorithms for this cellular system. The "No Grouping" approach is the WMMSE algorithm with no grouping [1] while the "Proposed Grouping Approach" represents the results of performing Algorithm 14 with $G = 3$ groups. Thus, the "No Grouping" approach serves all the users simultaneously in a single group, while the grouping approach arranges the users into three (possibly overlapping) groups which are then served in the TDMA fashion. In the "SVD-MMSE-TDMA" approach, each base station serves its own users in a TDMA fashion by using the SVD (singular value decomposition) precoder and the users deploy MMSE receivers. The base stations transmit simultaneously. The "Correlated Signaling" refers to the WMMSE algorithm [1] applied to the extended channel

over G time slots (defined by block diagonal channel matrices of size $MG \times NG$) to determine the transmit (resp. receive) beamformers of size $MG \times d$ (resp. $NG \times d$). In the "Random Grouping" approach, we first partition the users randomly into 3 different groups and then we use WMMSE algorithm for beamformer design within each group and we also use equal time allocated to all groups. Figure 4.6 represents the rate CDF (Cumulative Distribution Function) comparison between these methods. The $x-$axis corresponds to the rate values and the $y-$axis is the percentage of the users having rates smaller than the value on the $x-$axis. As can be seen from Figure 4.6, the proposed grouping method achieves a substantially higher and, at the same time, more fair rate distribution than the standard multi-user MIMO strategy (namely, "SVD-MMSE-TDMA"). Furthermore, there is a small additional gain if Algorithm 2 is used to further optimize time allocation (i.e., update $\boldsymbol{\beta}$ using the method in Section 3.1.3). This figure also shows that correlated signaling does not provide any material improvement in either the system throughput or user fairness over the proposed grouping approach.

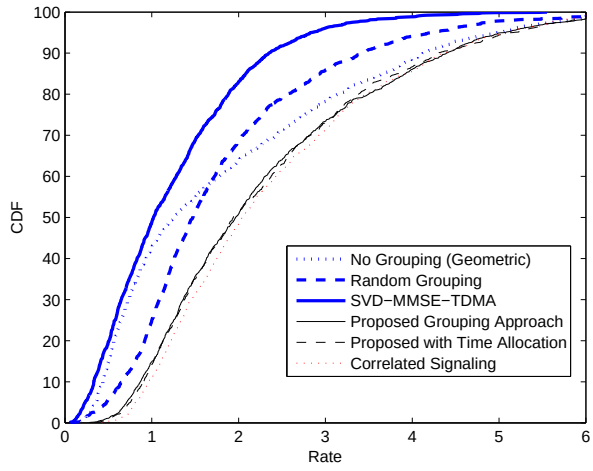


Figure 4.6: Rate CDF of different methods

Figure 4.7 depicts the convergence speed of the proposed algorithm (with no time allocation). As can be seen, the algorithm appears to converge in a few iterations where one *iteration* consists of one round of updating all the transmit and receive beamformers. This fast convergence property makes the algorithm well-suited for distributed implementation with low system overhead.

Figure 4.8 represents the tradeoff between the performance and the convergence speed while changing the number of groups in the system. We plot the value of the system utility versus iteration number in the proposed algorithm. As expected, when the number of groups increases, the convergence speed slows down, but the system performance improves.

In all above simulations, the number of users is much more than the number of groups. In the next simulation experiment, we examine the performance of the algorithm in the scenario that the number of groups is comparable to the number of users. We consider a small system with 3 cells where there are 2 users in each cell. We consider 2 antennas at the transmit side and one antenna at the receive side. The channel model is the same as the above channel model and we averaged the results over 50 Monte Carlo runs with independently generated channels. The results are illustrated in Figure 4.9. We also observe that when the number of groups is comparable to the number of users, adjusting the time of each group could improve the performance of the cell-edge users. In fact, in this simulation, time allocation results in 21% improvement of the cell-edge users' rate.

Finally, we use a small example to illustrate how the groups are formed by the proposed mehtod. We randomly select 7 base stations in the system and in each cell, we choose 4 random users. Hence, there are a total of 28 users in the system. Table 4.1 represents the rate allocation of different users in different groups. As can be seen from the table, although we do not have any discrete variables for user grouping in our final

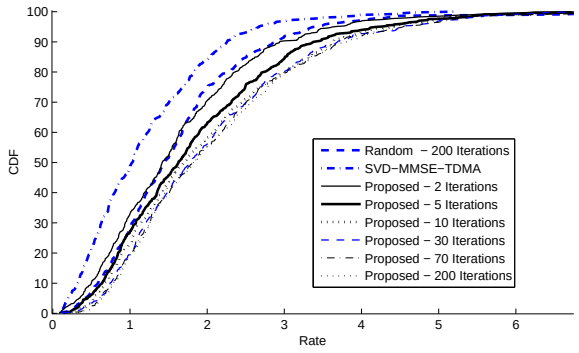


Figure 4.7: Rate CDF of different methods at various iterations

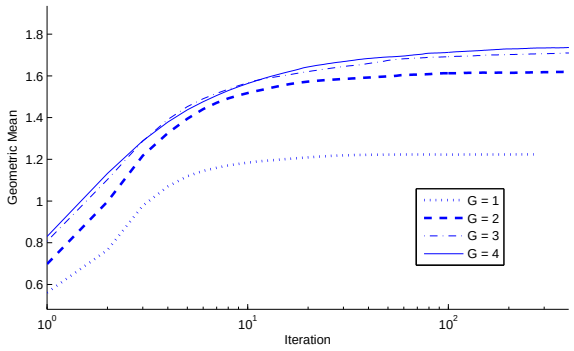


Figure 4.8: Geometric Mean vs. Iterations

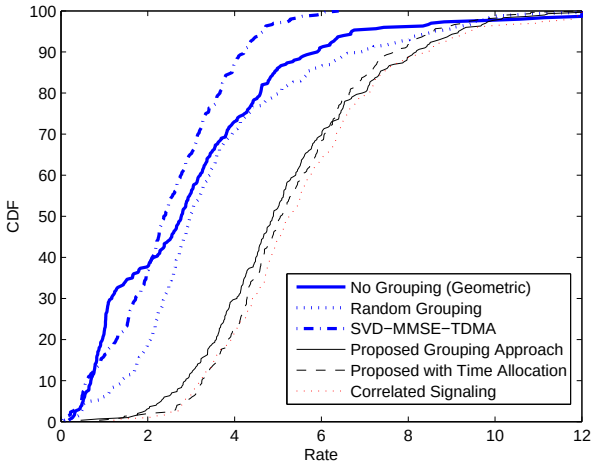


Figure 4.9: Rate CDF of various methods

problem formulation, the resulting rate allocation divides the users in different groups. Furthermore, the algorithm have the *group sharing* property, i.e., some users are served in multiple groups. This property, which was explained in Example 2 (subsection 3.1.3), is the result of our problem formulation (3.16).

4.1.3 Beamforming for Max-Min Fairness

Now we present our numerical experiments comparing four different approaches for the beamformer design in the interfering broadcast channel related to the problem in subsection 3.1.4. The first approach for designing the beamformers is the simple "*WMMSE*" algorithm proposed in [1] for maximizing the weighted sum rate of the system. Since the sum rate utility function is not a fair utility function among the users, we also consider the proportional fairness (geometric mean) utility function of the users. We use the framework in [1], [6] for maximizing the geometric mean utility function of the system and the resulting plots are denoted by the label "*GWMMSE*".

Another way of designing the beamformers for maximizing the performance of the worst user in the system is to approximate the max-min utility function. One proposed approximation for the max-min utility function could be (see [234]): $\min_{i_k} R_{i_k} \approx \log\left(\sum_{i_k \in \mathcal{I}} \exp(-R_{i_k})\right)$. Therefore instead of solving problem (P), we may maximize the above approximation of the objective by solving the following optimization problem

$$\begin{aligned} \max_{\mathbf{V}} \quad & \sum_{i_k \in \mathcal{I}} \exp(-R_{i_k}) \\ \text{s.t.} \quad & \sum_{i=1}^{I_k} \mathrm{Tr}(\mathbf{V}_{i_k} \mathbf{V}_{i_k}^H) \leq P_k, \quad \forall\, k \in \mathcal{K}. \end{aligned} \tag{4.1}$$

If we restrict ourselves to the case of $d_{i_k} = 1, \forall\, i_k \in \mathcal{I}$, then the MSE matrix $\mathbf{E}_{i_k}$ becomes a scalar and thus we can denote it by e_{i_k}. Using the relation (3.7) and plugging in the optimal value for the matrix $\mathbf{W}_{i_k}$ yields $R_{i_k} = \log(e_{i_k}^{-1})$. Plugging in this relation

Table 4.1: Achieved user rates in different groups/time slots

–	Achieved rate in group 1	Achieved rate in group 2	Achieved rate in group 3
User 1	0	2.6669	0
User 2	0	4.7570	0
User 3	3.6187	0	0
User 4	4.5252	0	0
User 5	0	0	5.6320
User 6	1.1291	2.3090	0
User 7	1.9406	3.8585	0
User 8	0	0	11.0470
User 9	0	3.7778	4.1621
User 10	0.9279	0	0
User 11	2.5982	0	0
User 12	0	3.4498	0
User 13	0	0	1.7782
User 14	0.8661	0	0
User 15	0	0	3.1569
User 16	0	3.1501	0
User 17	0	3.9681	0
User 18	0	3.1423	0
User 19	7.8421	0	0
User 20	0	0	4.9356
User 21	0	2.3733	0
User 22	0	8.4049	0
User 23	0	0	2.3800
User 24	4.9645	0	0
User 25	6.2302	0	7.4342
User 26	0	4.0770	0
User 27	0	8.7246	0
User 28	3.3389	0	6.0817

in (4.1), we obtain the equivalent optimization form of (4.1):

$$\begin{aligned} \min_{\mathbf{V}} \quad & \sum_{i_k \in \mathcal{I}} e_{i_k} \\ \text{s.t.} \quad & \sum_{i=1}^{I_k} \mathrm{Tr}(\mathbf{V}_{i_k} \mathbf{V}_{i_k}^H) \leq P_k, \quad \forall\, k, \end{aligned} \tag{4.2}$$

which is the well-known sum MSE minimization problem and we use the algorithm in [235] to solve (4.2). The corresponding plots of this method are labeled by "*MMSE*" in our figures.

In our simulations, the first four plots are averaged over 50 channel realizations. In each channel realization, the channel coefficients are drawn from the zero mean unit variance i.i.d. Gaussian distribution.

In the first numerical experiment, we consider $K = 4$ BSs, each equipped with $M = 6$ antennas. There are $I = 3$ users in each cell where each of them is equipped with $N = 2$ antennas. Figure 4.10 and Figure 4.11 respectively represent the rate cumulative rate distribution function and the minimum rate in the system. The power level P_k is set to 20dB for all BSs in Figure 4.10. As these figures show, our proposed method yields substantially more fair rate allocation in the system.

In our second set of numerical experiments in Figure 4.12 and Figure 4.13, we explore the system with $K = 5$ cells where each BS serves $I = 3$ users. The number of transmit and receive antennas are respectively $M = 3$ and $N = 2$. As Figure 4.11 and Figure 4.13 show, WMMSE and MMSE algorithms could shut off some users and lead to zero objective function.

Figure 4.14 and Figure 4.15 show the convergence rate of the algorithm while a user is joining the system. In these plots, there are 5 cells and 2 users in each cell initially and at iteration 11, another user is added to one of the cells. When the extra user is added to the system, the power for the users in the same cell is reduced by a factor of $\frac{2}{3}$ and the rest of the power is used to serve the joined user initially. The precoder of the joined user is initialized randomly. Figure 4.14 shows the objective function of (Q) during the iterations while Figure 4.15 demonstrates the minimum rate of the users in the system versus the iteration number.

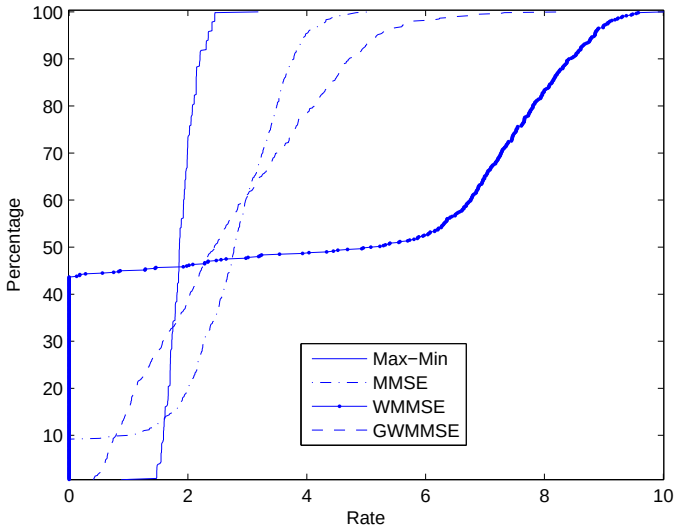


Figure 4.10: Rate CDF: $K = 4, I = 3, M = 6, N = 2, d = 1$

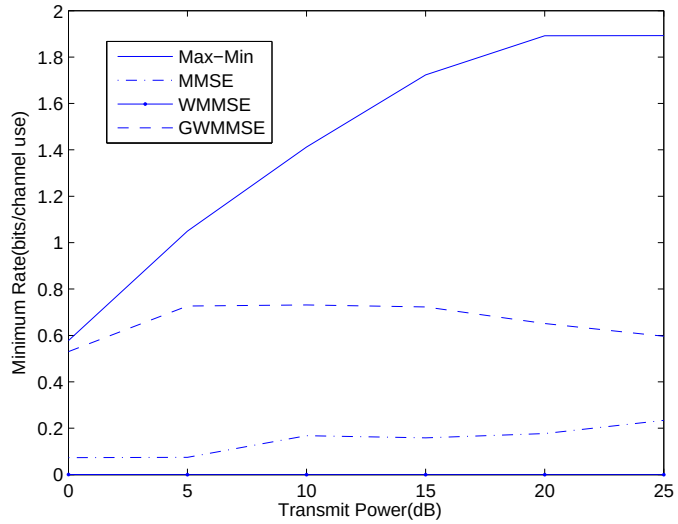


Figure 4.11: Minimum rate in the system versus transmit power

Figure 4.16 and Figure 4.17 represent the performance and the convergence rate of the algorithm when the channel is changing during the iterations. At iteration 15, the channel is changed by a Rayleigh fade with power 0.1. As it can be seen from the plots,

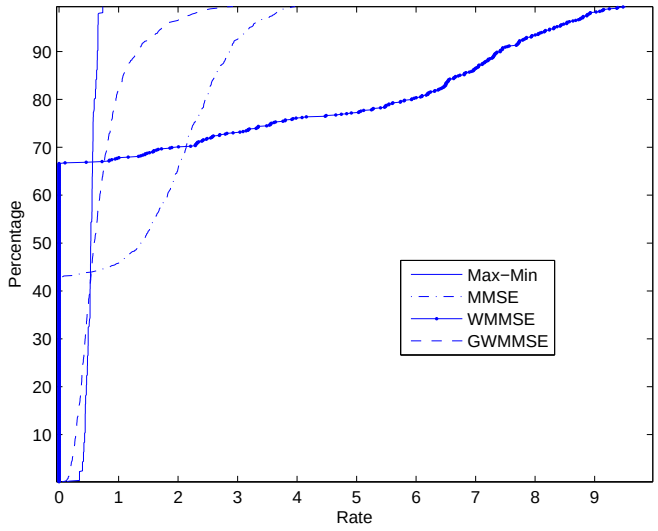


Figure 4.12: Rate CDF: $K = 5, I = 3, M = 3, N = 2, d = 1$

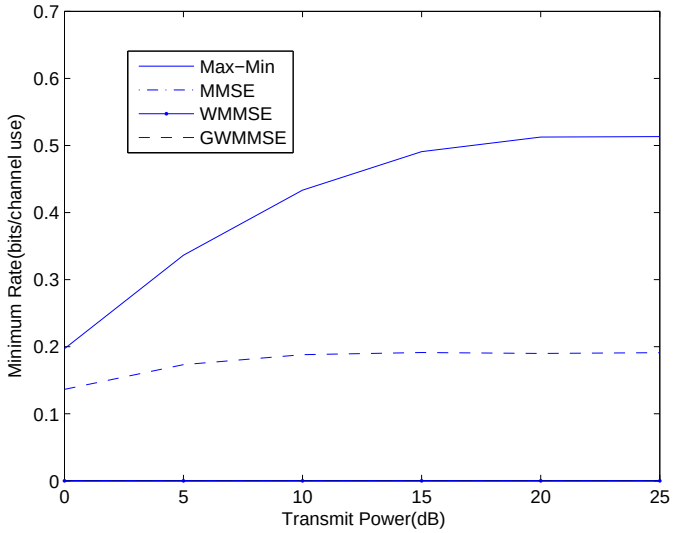


Figure 4.13: Minimum rate in the system

the algorithm converges fast and it adapts to the new channel after a few iterations.

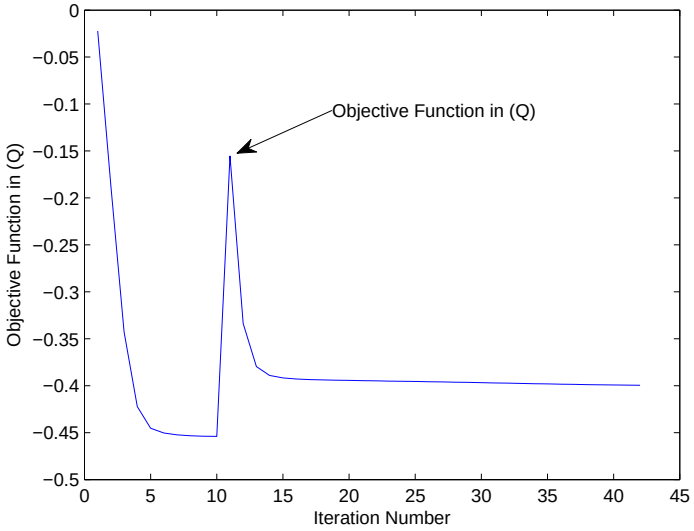


Figure 4.14: WMMSE objective function while adding a User

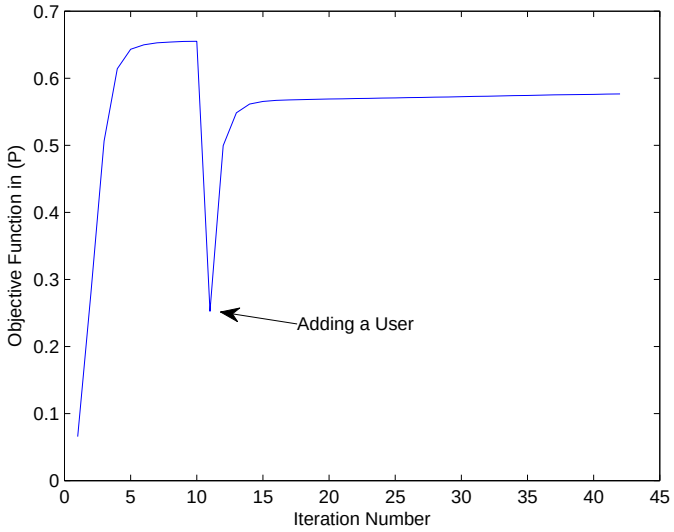


Figure 4.15: Minimum rate while adding a User

4.1.4 Expected Sum-Rate Maximization

In this subsection we numerically evaluate the performance of the SSUM algorithm for maximizing the expected sum-rate in a wireless network. In our simulations, we

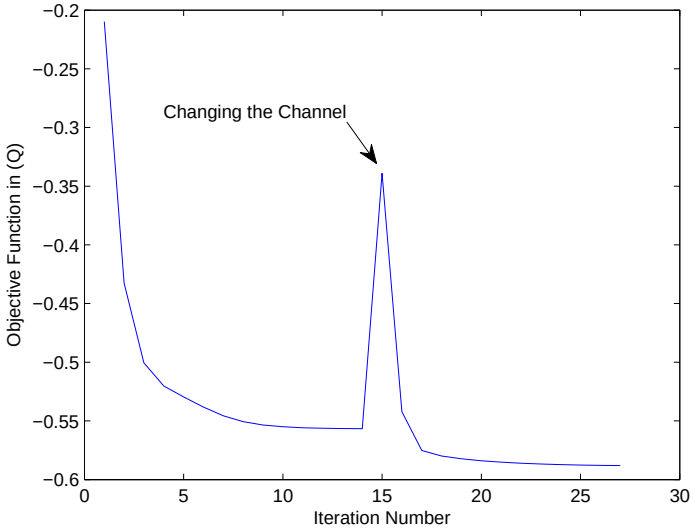


Figure 4.16: WMMSE objective function while changing the channel

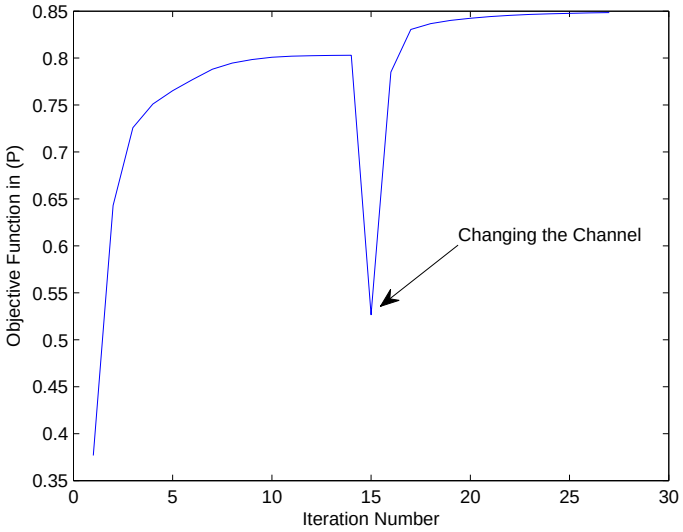


Figure 4.17: Minimum rate while changing the channel

consider $K = 57$ base stations each equipped with $M = 4$ antennas and serve a two antenna user in its own cell. The path loss and the power budget of the transmitters are generated using the 3GPP (TR 36.814) evaluation methodology [233]. We assume that

partial channel state information is available for some of the links. In particular, each user estimates only its direct link, plus the interfering links whose powers are at most η (dB) below its direct channel power. For these estimated links, we assume a channel estimation error model in the form of $\hat{h} = h + z$, where h is the actual channel; $\hat{h}$ is the estimated channel, and z is the estimation error. Given a MMSE channel estimate $\hat{h}$, we can determine the distribution of h as $\mathcal{CN}(\hat{h}, \frac{\sigma_l^2}{1+\gamma \mathrm{SNR}})$ where γ is the effective signal to noise ratio (SNR) coefficient depending on the system parameters (e.g. the number of pilot symbols used for channel estimation) and σ_l is the path loss. Moreover, for the channels which are not estimated, we assume the availability of estimates of the path loss σ_l and use them to construct statistical models (Rayleigh fading is considered on top of the path loss).

We compare the performance of four different algorithms: *one sample WMMSE*, *mean WMMSE*, *stochastic gradient*, and *Stochastic WMMSE*. In "one sample WMMSE" and "mean WMMSE", we apply the WMMSE algorithm [1] on one realization of all channels and mean channel matrices respectively. In the SG method, we apply the stochastic gradient method with diminishing step size rule to the ergodic sum rate maximization problem; see Section 3.3.6. Figure 4.18 shows our simulation results when each user only estimates about 3% of its channels, while the others are generated synthetically according to the channel distributions. The expected sum rate in each iteration is approximated in this figure by a Monte-Carlo averaging over 500 independent channel realizations. As can be seen from Figure 4.18, the Stochastic WMMSE algorithm significantly outperforms the rest of the algorithms. Although the stochastic gradient algorithm with diminishing step size (of order $1/r$) is guaranteed to converge to a stationary solution, its convergence speed is sensitive to the step size selection and is usually slow. We have also experimented the SG method with different constant step sizes in our numerical simulations, but they typically led to divergence.

In Figure 4.18, we set $\eta = 6$, $\gamma = 1$ and consequently only 3% of the channel matrices are estimated, while the rest are generated by their path loss coefficients plus Rayleigh fading. The signal to noise ratio is set SNR = 15 (dB). Figure 4.19 illustrates the performance of the algorithms for $\eta = 12$ whereby about 6% of the channels are estimated.

Figure 4.18: Expected sum rate: $\eta = 6$)

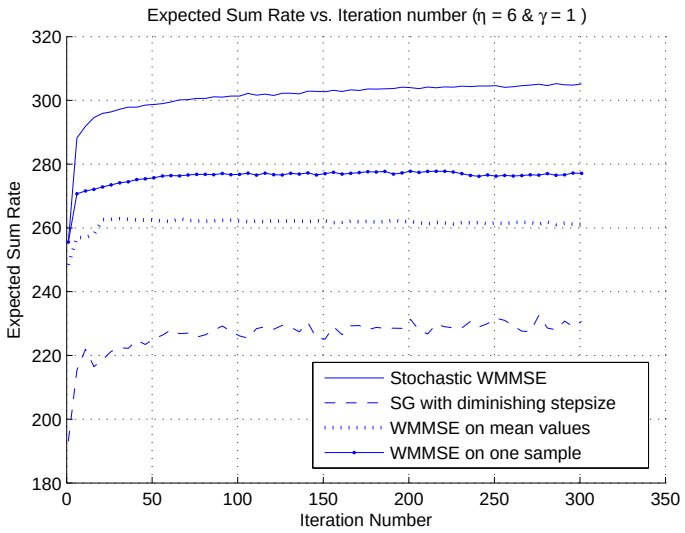
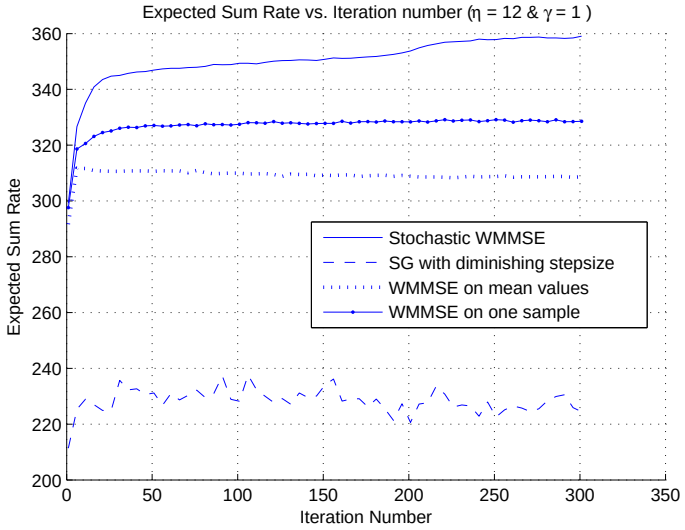


Figure 4.19: Expected sum rate: $\eta = 12$



4.2 Dictionary Learning for Sparse Represnetation

In this section, we apply the proposed sparse dictionary learning methods in subsection 3.2.4, namely algorithm 19, to the image denoising application; and compare its

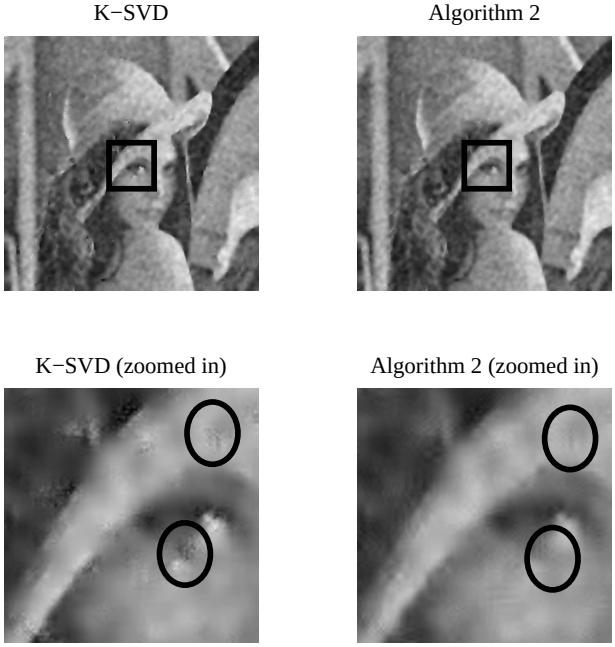


Figure 4.20: Sample denoised images ($\sigma = 100$).

performance with that of the K-SVD algorithm proposed in [165] (and summarized in Algorithm 22). As a test case, we use the image of Lena corrupted by additive Gaussian noise with various variances (σ^2).

In Algorithm 22, $\mathbf{R}_{i,j}\mathbf{S}$ denotes the image patch centered at (i, j) coordinate. In step 2, dictionary $\mathbf{A}$ is trained to sparsely represent *noisy* image patches by using either K-SVD algorithm or Algorithm 19. The term $\mathbf{x}_{i,j}$ denotes the sparse representation coefficient of the patch (i, j). In K-SVD, it (approximately) solves ℓ_0-norm regularized problem (4.3) by using orthogonal matching pursuit (OMP) to update $\mathbf{X}$. In our approach, we use Algorithm 19 with $\mathcal{A} = \{\mathbf{A} \mid \|\mathbf{a}_i\| \leq 1, \forall\ i = 1, \cdots, N\}$ to solve the ℓ_1-penalized dictionary learning formulation (4.4). We set $\mu_{i,j} = c(0.0015\sigma + 0.2),\ \forall\ i, j$, in (4.4) with $c = \frac{1}{I \times J} \sum_{i,j} \|\mathbf{R}_{i,j} S\|_2$, and $I \times J$ denotes the total number of image patches. This choice of the parameter μ_{ij} intuitively means that we emphasize on sparsity more in the presence of stronger noise. Numerical values $(0.0015, 0.2)$ are determined experimentally. The final denoised image $\mathbf{S}$ is obtained by (4.5) and setting $\beta = 30/\sigma$, as suggested in [165].

σ/PSNR	DCT	K-SVD	Algorithm 19
20/22.11	32	**32.38**	30.88
60/12.57	26.59	**26.86**	26.37
100/8.132	24.42	24.45	**24.46**
140/5.208	22.96	22.93	**23.11**
180/3.025	21.73	21.69	**21.96**

Table 4.2: Image denoising result comparison on "Lena"

Algorithm 22 Image denoising using K-SVD or algorithm 19

Require: noisy image $\mathbf{Y}$, noise variance σ^2
Ensure: denoised image $\mathbf{S}$
1: Initialization: $\mathbf{S} = \mathbf{Y}$, $\mathbf{A}$ = overcomplete DCT dictionary
2: Dictionary learning:
K-SVD:

$$\min_{\mathbf{A},\mathbf{X}} \sum_{i,j} \mu_{ij} \|\mathbf{x}_{i,j}\|_0 + \sum_{i,j} \|\mathbf{A}\mathbf{x}_{i,j} - \mathbf{R}_{i,j}\mathbf{S}\|^2 \tag{4.3}$$

Algorithm 19:

$$\min_{\mathbf{A}\in\mathcal{A},\mathbf{X}} \sum_{i,j} \mu_{ij} \|\mathbf{x}_{i,j}\|_1 + \sum_{i,j} \|\mathbf{A}\mathbf{x}_{i,j} - \mathbf{R}_{i,j}\mathbf{S}\|^2 \tag{4.4}$$

3: $\mathbf{S}$ update:

$$\mathbf{S} = (\beta\mathbf{I} + \sum_{i,j} \mathbf{R}_{i,j}^T \mathbf{R}_{i,j})^{-1} (\beta\mathbf{Y} + \sum_{i,j} \mathbf{R}_{i,j}^T \mathbf{A}\mathbf{x}_{i,j}) \tag{4.5}$$

The final peak signal-to-noise ratio (PSNR) comparison is summarized in Table 4.2; and sample images are presented in Figure 4.20. As can be seen in Table 4.2, the resulting PSNR values of the proposed algorithm are comparable with the ones obtained by K-SVD, where the results are averaged over 10 Monte-Carlo runs. However, visually, K-SVD produces more noticeable artifacts (see the circled spot in Figure 4.20) than our proposed algorithm. The artifacts may be due to the use of OMP in K-SVD which is less robust to noise than the ℓ_1-regularizer used in Algorithm 19. As for the CPU time, the two algorithms perform similarly in the numerical experiments.

Chapter 5

Future Work

Here we briefly outline some of the possible future directions of this research:

- **Parallel methods for stochastic optimization:** Big data problems typically requires data-fetching since accessing to the whole data is not possible. Is it possible to extend our proposed parallel framework to the stochastic problems in order to solve the big data problems with data-fetching?
- **Solving the optimization problems over the network of computing nodes:** Our proposed parallel processing framework assumes that the computing nodes are fully connected. What happens if they are not fully connected?
- **Sparse dictionary learning problem with parallel processing:** It is very natural to apply the RPSUM framework to the discussed sparse dictionary learning problem. The performance of such an algorithm should be evaluated numerically and on real data.
- **Detailed computational complexity analysis of the dictionary learning problem:** Our NP-hardness result of the sparse dictionary learning problem requires that both the number of samples and the data dimension to increase. What is the computational complexity status of the problem when one of these variables is fixed?
- **Generalization of the SSUM algorithm to the Markov chain scenario:** In many practical scenarios, the random samples of the stochastic optimization

problem is not independent. For example, in the beamformer design problem, when the samples are obtained using the estimation of the channels, it is more reasonable to model the samples as a Markov chain rather than i.i.d samples. How does the SSUM framework perform on this setup?

- **Dealing with users joining/leaving the network in the modern heterogenous networks:** In a wireless heterogenous network, the users may join or leave the network at any time. How should a beamfoming algorithm respond to such changes?

- **Joint beamfroming, scheduling, base-station assignment, and traffic engineering in the heterogenous networks:** Here we proposed an algorithm for the joint beamforming and scheduling problem. However, in the modern heterogenous networks, each user may connect to different base stations and also the packet of each user may be routed in different ways. What is the optimal strategy for transmitting a packet from the cloud center to the users in the system?

- **Non-asymptotic convergence analysis of the deterministic parallel successive upper-bound minimization algorithm:** In this reseach, we analyzed the iteration complexity of the RPSUM algorithm. What happens if the blocks are chosen based on the essentially coverable update rule?

- **Iteration complexity analysis of the diminishing step-size selection rule:** In all the iteration complexity analyses of this dissertation, the step-size is constant and fixed. It is interesting to study the diminishing step-size selection rule as well. The result of this study could shed light on the choice of the diminishing step-size selection rule.

- **Convergence of non-convex ADMM/BSUM-M framework:** In many practical optimization problems, the objective function is non-convex. What can we say about the convergence of ADMM/BSUM-M framework when it is applied to non-convex problems?

- **Accelerated versions of the proposed methods:** It is well-known that the gradient descent method and many other first order algorithms could be accelerated to obtain $\mathcal{O}(1/r^2)$ convergence rate. Is it possible to accelerate the BSUM

framework algorithms?

- **Does the randomization help to solve the non-cooperative games:** Consider a simple scenario of solving a system of linear equations with Gauss-Seidel method. As discussed in this dissertation, in order for the Gauss-Seidel method to converge, it is sufficient and necessary that certain linear mapping (which depends on the coefficients matrix) be contraction. In the randomized setup, the matrix which is multiplied to the iterates is no longer fixed. What is the necessary and sufficient condition for the randomized method to converge? This problem seems to be related to the largest Lyapunov exponent of the random matrices product.

References

[1] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He. An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel. *IEEE Transactions on Signal Processing*, 59(9):4331–4340, 2011.

[2] J. A. Hartigan and M. A. Wong. K-means clustering algorithm. *Journal of the Royal Statistical Society, Series C (Applied Statistics)*, 28:100–108, 1979.

[3] Y. Censor and S. A. Zenios. *Parallel Optimization: Theory, Algorithm, and Applications*. Oxford University Press, Oxford, United Kingdom, 1997.

[4] H. R. Howson and N. G. F. Sancho. A new algorithm for the solution of multistate dynamic programming problems. *Mathematical Programming*, 8:104–116, 1975.

[5] P. Tseng and S. Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, 117(1-2):387–423, 2009.

[6] M. Razaviyayn, H. Baligh, A. Callard, and Z.-Q. Luo. Joint user grouping and transceiver design in a MIMO interfering broadcast channel. *IEEE Transactions on Signal Processing*, 2013.

[7] J. M. Ortega and W. C. Rheinboldt. *Iterative Solutions of Nonlinear Equations in Several Variables*. New York: Academic, 1970.

[8] S. Shalev-Shwartz and A. Tewari. Stochastic methods for ℓ_1-regularized loss minimization. *The Journal of Machine Learning Research*, 12:1865–1892, 2011.

[9] H. Zhang, J. Jiang, and Z.-Q. Luo. On the linear convergence of a proximal gradient method for a class of nonsmooth convex minimization problems. *Journal of the Operations Research Society of China*, 1(2):163–186, 2013.

[10] A. Beck and L. Tetruashvili. On the convergence of block coordinate descent type methods. *SIAM Journal on Optimization*, 23(4):2037–2060, 2013.

[11] A. L. Yuille and A. Rangarajan. The concave-convex procedure. *Neural Computation*, 15:915–936, 2003.

[12] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B*, 39:1–38, 1977.

[13] D. P. Bertsekas. *Nonlinear Programming*. Athena-Scientific, second edition, 1999.

[14] L. Grippo and M. Sciandrone. On the convergence of the block nonlinear Gauss-Seidel method under convex constraints. *Operations Research Letters*, 26:127–136, 2000.

[15] Y. Xu and W. Yin. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. *SIAM Journal on Imaging Sciences*, 6(3):1758–1789, 2013.

[16] B. R. Marks and G. P. Wright. A general inner approximation algorithm for nonconvex mathematical programs. *Operations Research*, 26:681–683, 1978.

[17] G. Scutari, F. Facchinei, P. Song, D. P. Palomar, and J.-S. Pang. Decomposition by partial linearization: Parallel optimization of multi-agent systems. *arXiv preprint arXiv:1302.0756*, 2013.

[18] G. Scutari, F. Facchinei, P. Song, D. P. Palomar, and J.-S. Pang. Decomposition by partial linearization in multiuser systems. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4424–4428. IEEE, 2013.

[19] A. Alvarado, G. Scutari, and J.=S. Pang. A new decomposition method for multiuser DC-programming and its applications. *arXiv preprint arXiv:1308.3521*, 2013.

[20] F. Facchinei, S. Sagratella, and G. Scutari. Flexible parallel algorithms for big data optimization. *arXiv preprint arXiv:1311.2444*, 2013.

[21] J. Mairal. Optimization with first-order surrogate functions. *arXiv preprint arXiv:1305.3120*, 2013.

[22] M. Aharon, M. Elad, and A. Bruckstein. K-svd: Design of dictionaries for sparse representation. *Proceedings of SPARS*, 5:9–12, 2005.

[23] P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109:475–494, 2001.

[24] L. J. Hong, Y. Yang, and L. Zhang. Sequential convex approximations to joint chance constrained programs: A monte carlo approach. *Operations Research*, 59(3):617–630, 2011.

[25] Z.-Q. Luo and P. Tseng. Error bounds and convergence analysis of feasible descent methods: a general approach. *Annals of Operations Research*, 46(1):157–178, 1993.

[26] Z.-Q. Luo and P. Tseng. On the convergence of the coordinate descent method for convex differentiable minimization. *Journal of Optimization Theory and Applications*, 72(1):7–35, 1992.

[27] Z.-Q. Luo and P. Tseng. On the linear convergence of descent methods for convex essentially smooth minimization. *SIAM Journal on Control and Optimization*, 30(2):408–425, 1992.

[28] P. Tseng. Approximation accuracy, gradient methods, and error bound for structured convex optimization. *Mathematical Programming*, 125(2):263–295, 2010.

[29] I. Necoara and D. Clipici. Distributed coordinate descent methods for composite minimization. *arXiv preprint arXiv:1312.5302*, 2013.

[30] M. Kadkhodaie, M. Sanjabi, and Z.-Q. Luo. On the linear convergence of the approximate proximal splitting method for non-smooth convex optimization. *arXiv preprint arXiv:1404.5350*, 2014.

[31] A. Saha and A. Tewari. On the nonasymptotic convergence of cyclic coordinate descent methods. *SIAM Journal on Optimization*, 23(1):576–601, 2013.

[32] Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

[33] P. Richtárik and M. Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, pages 1–38, 2012.

[34] Z. Lu and L. Xiao. On the complexity analysis of randomized block-coordinate descent methods. *arXiv preprint arXiv:1305.4723*, 2013.

[35] 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.

[36] M. Hong, X. Wang, M. Razaviyayn, and Z.-Q. Luo. Iteration complexity analysis of block coordinate descent methods. *arXiv preprint arXiv:1310.6957*, 2013.

[37] M. Hong, T. Chang, X. Wang, M. Razaviyayn, S. Ma, and Z.-Q. Luo. A block successive upper bound minimization method of multipliers for linearly constrained convex optimization. *arXiv preprint arXiv:1401.7079*, 2014.

[38] M. J. D. Powell. On search directions for minimization algorithms. *Mathematical Programming*, 4(1):193–201, 1973.

[39] N. Zadeh. A note on the cyclic coordinate ascent method. *Management Science*, 16:642–644, 1970.

[40] B. Chen, S. He, Z. Li, and S. Zhang. Maximum block improvement and polynomial optimization. *SIAM Journal on Optimization*, 22:87–107, 2012.

[41] L. Grippo and M. Sciandrone. Globally convergent block-coordinate techniques for unconstrained optimization. *Optimization methods and software*, 10:587–637, 1999.

[42] S. Bonettini. Inexact block coordinate descent methods with application to non-negative matrix factorization. *IMA journal of numerical analysis*, 31:1431–1452, 2011.

[43] D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1):17–40, 1976.

[44] D. Gabay. Applications of the method of multipliers to variational inequalities. *Studies in mathematics and its applications*, 15:299–331, 1983.

[45] 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.

[46] M. Hong and Z.-Q. Luo. On the linear convergence of the alternating direction method of multipliers. *arXiv preprint arXiv:1208.3922*, 2012.

[47] I. D. Schizas, G. Mateos, and G. B. Giannakis. Distributed LMS for consensus-based in-network adaptive processing. *IEEE Transactions on Signal Processing*, 57(6):2365–2382, 2009.

[48] G. Mateos, I. D. Schizas, and G. B. Giannakis. Performance analysis of the consensus-based distributed LMS algorithm. *EURASIP Journal on Advances in Signal Processing*, 2009:68, 2009.

[49] G. Mateos, I. D. Schizas, and G. B. Giannakis. Distributed recursive least-squares for consensus-based in-network adaptive estimation. *IEEE Transactions on Signal Processing*, 57(11):4583–4588, 2009.

[50] S. Chenand, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM journal on scientific computing*, 20(1):33–61, 1998.

[51] M. Alizadeh, X. Li, Z. Wang, A. Scaglione, and R. Melton. Demand-side management in the smart grid: Information processing for the power switch. *IEEE Signal Processing Magazine*, 29(5):55–67, 2012.

[52] N. Li, L. Chen, and S. H. Low. Optimal demand response based on utility maximization in power networks. In *IEEE Power and Energy Society General Meeting*, pages 1–8. IEEE, 2011.

[53] Q. Zhao and B. M. Sadler. A survey of dynamic spectrum access. *IEEE Signal Processing Magazine*, 24(3):79–89, 2007.

[54] P. Richtárik and M. Takáč. Efficient serial and parallel coordinate descent methods for huge-scale truss topology design. In *Operations Research Proceedings*, pages 27–32. Springer, 2012.

[55] I. Necoara and D. Clipici. Efficient parallel coordinate descent algorithm for convex optimization problems with separable constraints: application to distributed MPC. *Journal of Process Control*, 23(3):243–253, 2013.

[56] Y. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

[57] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[58] S. J. Wright, R. D. Nowak, and M. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, 57(7):2479–2493, 2009.

[59] J. K. Bradley, A. Kyrola, D. Bickson, and C. Guestrin. Parallel coordinate descent for ℓ_1-regularized loss minimization. *arXiv preprint arXiv:1105.5379*, 2011.

[60] C. Scherrer, A. Tewari, M. Halappanavar, and D. Haglin. Feature clustering for accelerating parallel coordinate descent. In *NIPS*, pages 28–36, 2012.

[61] C. Scherrer, M. Halappanavar, A. Tewari, and D. Haglin. Scaling up coordinate descent algorithms for large ℓ_1 regularization problems. *arXiv preprint arXiv:1206.6409*, 2012.

[62] Z. Peng, M. Yan, and W. Yin. Parallel and distributed sparse optimization. *preprint*, 2013.

[63] P. Richtárik and M. Takáč. Parallel coordinate descent methods for big data optimization. *arXiv preprint arXiv:1212.0873*, 2012.

[64] F. Niu, B. Recht, C. Ré, and S. J. Wright. Hogwild!: A lock-free approach to parallelizing stochastic gradient descent. *Advances in Neural Information Processing Systems*, 24:693–701, 2011.

[65] J. Liu, S. J. Wright, C. Ré, and V. Bittorf. An asynchronous parallel stochastic coordinate descent algorithm. *arXiv preprint arXiv:1311.1873*, 2013.

[66] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer, 2004.

[67] E. L. Plambeck, B.-R. Fu, S. M. Robinson, and R. Suri. Sample-path optimization of convex stochastic performance functions. *Mathematical Programming*, 75(2):137–176, 1996.

[68] S. M. Robinson. Analysis of sample-path optimization. *Mathematics of Operations Research*, 21(3):513–528, 1996.

[69] K. Healy and L. W. Schruben. Retrospective simulation response optimization. In *Proceedings of the 23rd conference on Winter simulation*, pages 901–906. IEEE Computer Society, 1991.

[70] R. Y. Rubinstein and A. Shapiro. Optimization of static simulation models by the score function method. *Mathematics and Computers in Simulation*, 32(4):373–392, 1990.

[71] R. Y. Rubinstein and A. Shapiro. *Discrete event systems: Sensitivity analysis and stochastic optimization by the score function method*, volume 346. Wiley New York, 1993.

[72] A. Shapiro. Monte carlo sampling methods. *Handbooks in operations research and management science*, 10:353–426, 2003.

[73] A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on stochastic programming: modeling and theory*, volume 9. Society for Industrial and Applied Mathematics, 2009.

[74] S. Kim, R. Pasupathy, and S. Henderson. A guide to sample-average approximation. 2011.

[75] B. E. Fristedt and L. F. Gray. *A modern approach to probability theory*. Birkhuser Boston, 1996.

[76] N. Dunford and J. T. Schwartz. *Linear Operators. Part 1: General Theory.* Interscience Publ. New York, 1958.

[77] J. Mairal. Incremental majorization-minimization optimization with application to large-scale machine learning. *arXiv preprint arXiv:1402.4419*, 2014.

[78] J. Mairal. Stochastic majorization-minimization algorithms for large-scale optimization. In *Advances in Neural Information Processing Systems*, pages 2283–2291, 2013.

[79] M. Razaviyayn, M. Sanjabi, and Z.-Q. Luo. A stochastic successive minimization method for nonsmooth nonconvex optimization with applications to transceiver design in wireless communication networks. *arXiv preprint arXiv:1307.4457*, 2013.

[80] J. Nash. Non-cooperative games. *Annals of mathematics*, pages 286–295, 1951.

[81] F. Facchinei and C. Kanzow. Generalized nash equilibrium problems. *Annals of Operations Research*, 175(1):177–211, 2010.

[82] F. Facchinei and J.-S. Pang. Nash equilibria: the variational approach. *Convex optimization in signal processing and communications*, pages 443–493, 2009.

[83] J.-S. Pang. Asymmetric variational inequality problems over product sets: applications and iterative methods. *Mathematical Programming*, 31(2):206–219, 1985.

[84] J.-S. Pang and D. Chan. Iterative methods for variational and complementarity problems. *Mathematical programming*, 24(1):284–313, 1982.

[85] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods.* Athena-Scientific, second edition, 1999.

[86] R. W. Cottle, J.-S. Pang, and R. E. Stone. *The linear complementarity problem*, volume 60. Siam, 2009.

[87] Z.-Q. Luo and J.-S. Pang. Analysis of iterative waterfilling algorithm for multiuser power control in digital subscriber lines. *EURASIP Journal on Advances in Signal Processing*, 2006, 2006.

[88] J.-S. Pang, G. Scutari, F. Facchinei, and C. Wang. Distributed power allocation with rate constraints in gaussian parallel interference channels. *IEEE Transactions on Information Theory*, 54(8):3471–3489, 2008.

[89] J.-S. Pang, G. Scutari, D. P. Palomar, and F. Facchinei. Design of cognitive radio systems under temperature-interference constraints: A variational inequality approach. *IEEE Transactions on Signal Processing*, 58(6):3251–3271, 2010.

[90] J.-S. Pang and G. Scutari. Joint sensing and power allocation in nonconvex cognitive radio games: Quasi-Nash equilibria. volume 61, pages 2366–2382. IEEE, 2013.

[91] G. Scutari and J.-S. Pang. Joint sensing and power allocation in nonconvex cognitive radio games: Nash equilibria and distributed algorithms. *IEEE Transactions on Information Theory*, 59(6):4626–4661, 2013.

[92] F. Facchinei, V. Piccialli, and M. Sciandrone. Decomposition algorithms for generalized potential games. *Computational Optimization and Applications*, 50(2):237–262, 2011.

[93] J.-S. Pang and G. Scutari. Nonconvex games with side constraints. *SIAM Journal on Optimization*, 21(4):1491–1522, 2011.

[94] H. Nikaido and K. Isoda. Note on noncooperative convex games. *Pacific Journal of Mathematics*, 5:807–815, 1955.

[95] S. D. Flam and A. Ruszczynski. Noncooperative convex games: computing equilibrium by partial regularization. Technical report, Department of Economics, University of Bergen, 2000.

[96] D. P. Palomar and Y. C. Eldar. *Convex optimization in signal processing and communications*. Cambridge university press, 2010.

[97] M. É. Borel. Les probabilités dénombrables et leurs applications arithmétiques. *Rendiconti del Circolo Matematico di Palermo (1884-1940)*, 27(1):247–271, 1909.

[98] F. P. Cantelli. Sulla probabilita come limite della frequenza. *Atti Reale Accademia Nazionale Lincei*, 26:39–45, 1917.

[99] A. Damnjanovic, J. Montojo, Y. Wei, T. Ji, T. Luo, M. Vajapeyam, T. Yoo, O. Song, and D. Malladi. A survey on 3GPP heterogeneous networks. *IEEE Wireless Communications*, 18(3):10 –21, June 2011.

[100] V. Chandrasekhar and J.G. Andrews. Femtocell networks: A survey. *IEEE Communicatios Magazine*, pages 59–67, sept 2008.

[101] Z.-Q. Luo and S. Zhang. Dynamic spectrum management: Complexity and duality. *IEEE Journal of Selected Topics in Signal Processing*, 2(1):57–73, 2008.

[102] Y.-F. Liu, Y.-H. Dai, and Z.-Q. Luo. Coordinated beamforming for MISO interference channel: Complexity analysis and efficient algorithms. *IEEE Transactions on Signal Processing*, 59(3):1142–1157, 2011.

[103] Y.-F. Liu, Y.-H. Dai, and Z.-Q. Luo. Max-min fairness linear transceiver design for a multi-user MIMO interference channel. In *IEEE International Conference on Communications (ICC), 2011*, pages 1–5. IEEE, 2011.

[104] G. Scutari, D. Palomar, and S. Barbarossa. The MIMO iterative waterfilling algorithm. *IEEE Transactions on Signal Processing*, 57(5):1917–1935, 2009.

[105] M. Razaviyayn, Z.-Q. Luo, P. Tseng, and J.-S. Pang. A Stackelberg game approach to distributed spectrum management. *Mathematical programming*, 129(2):197–224, 2011.

[106] J. Huang, R. A. Berry, and M. L. Honig. Distributed interference compensation for wireless networks. *IEEE Journal on Selected Areas in Communications*, 24(5):1074–1084, 2006.

[107] C. Shi, R. A. Berry, and M. L. Honig. Monotonic convergence of distributed interference pricing in wireless networks. In *IEEE International Symposium on Information Theory, ISIT*, pages 1619–1623. IEEE, 2009.

[108] C. Shi, R. A. Berry, and M. L. Honig. Local interference pricing for distributed beamforming in MIMO networks. In *IEEE Military Communications Conference, MILCOM*, pages 1–6. IEEE, 2009.

[109] Z. K. M. Ho and D. Gesbert. Balancing egoism and altruism on interference channel: The MIMO case. In *IEEE International Conference on Communications (ICC), 2010*, pages 1–5. IEEE, 2010.

[110] S. Ye and R. S. Blum. Optimized signaling for MIMO interference systems with feedback. *IEEE Transactions on Signal Processing*, 51(11):2839–2848, 2003.

[111] S.-J. Kim and G. B. Giannakis. Optimal resource allocation for MIMO ad hoc cognitive radio networks. *IEEE Transactions on Information Theory*, 57(5):3117–3131, 2011.

[112] M. Razaviyayn, M. Sanjabi, and Z.-Q. Luo. Linear transceiver design for interference alignment: Complexity and computation. *IEEE Transactions on Information Theory*, 58(5):2896–2910, 2012.

[113] S. S. Christensen, R. Agarwal, E. Carvalho, and J. M. Cioffi. Weighted sum-rate maximization using weighted MMSE for MIMO-BC beamforming design. *IEEE Transactions on Wireless Communications*, 7(12):4792–4799, 2008.

[114] D. A. Schmidt, C. Shi, R. A. Berry, M. L. Honig, and W. Utschick. Minimum mean squared error interference alignment. In *Forty-Third Asilomar Conference on Signals, Systems and Computers, 2009*, pages 1106–1110. IEEE, 2009.

[115] F. Negro, S. P. Shenoy, I. Ghauri, and D. Slock. On the MIMO interference channel. In *Information Theory and Applications Workshop (ITA), 2010*, pages 1–9. IEEE, 2010.

[116] J. Shin and J. Moon. Weighted sum rate maximizing transceiver design in MIMO interference channel. In *IEEE Global Telecommunications Conference (GLOBECOM 2011)*, pages 1–5. IEEE, 2011.

[117] J. Zander. Performance of optimum transmitter power control in cellular radio systems. *IEEE Transactions on Vehicular Technology*, 41(1):57–62, 1992.

[118] J. Zander. Distributed cochannel interference control in cellular radio systems. *IEEE Transactions on Vehicular Technology*, 41(3):305–311, 1992.

[119] C. W. Tan, M. Chiang, and R. Srikant. Fast algorithms and performance bounds for sum rate maximization in wireless networks. *IEEE/ACM Transactions on Networking*, 21(3):706–719, 2013.

[120] W. Yang and G. Xu. Optimal downlink power assignment for smart antenna systems. In *IEEE International Conference on Acoustics, Speech and Signal Processing, 1998*, volume 6, pages 3337–3340. IEEE, 1998.

[121] M. Bengtsson and B. Ottersten. Handbook of antennas in wireless communications. *Optimal and Suboptimal Transmit Beamforming*, 2001.

[122] A. Wiesel, Y. C. Eldar, and S. Shamai. Linear precoding via conic optimization for fixed MIMO receivers. *IEEE Transactions on Signal Processing*, 54(1):161–176, 2006.

[123] H. Boche and M. Schubert. Resource allocation in multiantenna systems-achieving max-min fairness by optimizing a sum of inverse SIR. *IEEE Transactions on Signal Processing*, 54(6):1990–1997, 2006.

[124] D. W. H. Cai, T. Q. S. Quek, and C. W. Tan. A unified analysis of max-min weighted SINR for MIMO downlink system. *IEEE Transactions on Signal Processing*, 59(8):3850–3862, 2011.

[125] M. Schubert and H. Boche. Solution of the multiuser downlink beamforming problem with individual SINR constraints. *IEEE Transactions on Vehicular Technology*, 53(1):18–28, 2004.

[126] M. Hong and Z.-Q. Luo. Signal processing and optimal resource allocation for the interference channel. *arXiv preprint arXiv:1206.5144*, 2012.

[127] I. Wajid, Y. C. Eldar, and A. Gershman. Robust downlink beamforming using covariance channel state information. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, pages 2285–2288, 2009.

[128] N. Vucic and H. Boche. Downlink precoding for multiuser MISO systems with imperfect channel knowledge. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, pages 3121–3124, 2008.

[129] E. Song, Q. Shi, M. Sanjabi, R. Sun, and Z.-Q. Luo. Robust SINR-constrained MISO downlink beamforming: When is semidefinite programming relaxation tight? In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, pages 3096–3099, 2011.

[130] A. Tajer, N. Prasad, and X. Wang. Robust linear precoder design for multi-cell downlink transmission. In *IEEE Transactions on Signal Processing*, volume 59, pages 235–251, 2011.

[131] M. Shenouda and T. N. Davidson. On the design of linear transceivers for multiuser systems with channel uncertainty. In *IEEE Journal on Selected Areas in Communications*, volume 26, pages 1015–1024, 2008.

[132] F. Negro, I. Ghauri, and D. Slock. Sum rate maximization in the noisy MIMO interfering broadcast channel with partial CSIT via the expected weighted MSE. In *International Symposium on Wireless Communication Systems, ISWCS*, pages 576–580, 2012.

[133] E.A. Jorswieck, A. Sezgin, H. Boche, and E. Costa. Multiuser MIMO MAC with statistical CSI and MMSE receiver: Feedback strategies and transmitter optimization. In *Proceedings of the 2006 international conference on Wireless communications and mobile computing*, pages 455–460. ACM, 2006.

[134] S. J. Kim and G. B. Giannakis. Optimal resource allocation for mimo ad-hoc cognitive radio networks. *Proceedings of Allerton Conference on Communication, Control, and Computing*, 2008.

[135] M. Chiang, C. W. Tan, D. P. Palomar, D. O'Neill, and D. Julian. Power control by geometric programming. *IEEE Transactions Wireless Communications*, 6(7):2640 –2651, july 2007.

[136] C. T. K. Ng and H. Huang. Linear precoding in cooperative MIMO cellular networks with limited coordination clusters. *IEEE Journal on Selected Areas in Communications*, 28(9):1446 –1454, december 2010.

[137] M. Hong and Z.-Q. Luo. Joint linear precoder optimization and base station selection for an uplink mimo network: A game theoretic approach. In *the Proceedings of the IEEE ICASSP*, 2012.

[138] Z. K. M. Ho and D. Gesbert. Balancing egoism and altruism on MIMO interference channel. *arXiv preprint arXiv:0910.1688*, 2009.

[139] M. Bengtsson and B. Ottersten. Optimal and suboptimal transmit beamforming. *Handbook of Antennas in Wireless Communications*, 2001. CRC Press.

[140] F. Rashid-Farrokhi, K.J.R. Liu, and L. Tassiulas. Transmit beamforming and power control for cellular wireless systems. *IEEE Journal on Selected Areas in Communications*, 16(8):1437 –1450, oct 1998.

[141] M. R. Garey and D. S. Johnson. *Computers and Intractability: A guide to the Theory of NP-completeness*. W. H. Freeman and Company, San Francisco, U.S.A, 1979.

[142] M. Razaviyayn, M. Hong, and Z.-Q. Luo. Linear transceiver design for a MIMO interfering broadcast channel achieving max-min fairness. In *Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, pages 1309–1313. IEEE, 2011.

[143] E. Larsson and E. Jorswieck. Competition versus cooperation on the MISO interference channel. In *IEEE Journal on Selected Areas in Communications*, volume 26, pages 1059–1069, 2008.

[144] G. Scutari, D. P. Palomar, F. Facchinei, and J.-S. Pang. Distributed dynamic pricing for mimo interfering multiuser systems: A unified approach. In *5th International Conference on Network Games, Control and Optimization (NetGCooP)*, pages 1–5. IEEE, 2011.

[145] Wei-Chiang Li, Tsung-Hui Chang, Che Lin, and Chong-Yung Chi. Coordinated beamforming for multiuser miso interference channel under rate outage constraints. 2011.

[146] M. Hong, R. Sun, H. Baligh, and Z.-Q. Luo. Joint base station clustering and beamformer design for partial coordinated transmission in heterogeneous networks. *IEEE Journal on Selected Areas in Communications*, 31(2):226–240, 2013.

[147] K. Engan, S. O. Aase, and J. Hakon Husoy. Method of optimal directions for frame design. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 2443–2446. IEEE, 1999.

[148] H. Lee, A. Battle, R. Raina, and A. Ng. Efficient sparse coding algorithms. In *Advances in neural information processing systems*, pages 801–808, 2006.

[149] V. A. Kotelnikov. On the carrying capacity of the ether and wire in telecommunications. In *Material for the First All-Union Conference on Questions of Communication, Izd. Red. Upr. Svyazi RKKA, Moscow*, 1933.

[150] H. Nyquist. Certain topics in telegraph transmission theory. *Transactions of the American Institute of Electrical Engineers*, 47(2):617–644, 1928.

[151] C. E. Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, 1949.

[152] E. T. Whittaker. *On the functions which are represented by the expansions of the interpolation-theory.* Edinburgh University, 1915.

[153] R. Prony. Essai experimental et analytique sur les lois de la dilatabilite des fluides elastiques et sur celles de la force expansive de la vapeur de leau et de la vapeur de lalkool, r differentes temperatures. *Journal Polytechnique ou Bulletin du Travail fait r Lecole Centrale des Travaux Publics, Paris, Premier Cahier*, pages 24–76, 1995.

[154] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[155] E. J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.

[156] M. Lustig, D. L. Donoho, and J. M. Pauly. Rapid MR imaging with compressed sensing and randomly under-sampled 3DFT trajectories. In *Proceedings of 14th Annual Meeting of ISMRM*. Citeseer, 2006.

[157] M. Lustig, J. H. Lee, D. L. Donoho, and J. M. Pauly. Faster imaging with randomly perturbed, under-sampled spirals and l1 reconstruction. In *Proceedings of the 13th Annual Meeting of ISMRM, Miami Beach*, page 685, 2005.

[158] K. Gedalyahu and Y. C. Eldar. Time-delay estimation from low-rate samples: A union of subspaces approach. *IEEE Transactions on Signal Processing*, 58(6):3017–3031, 2010.

[159] M. Mishali and Y. C. Eldar. Blind multiband signal reconstruction: Compressed sensing for analog signals. *IEEE Transactions on Signal Processing*, 57(3):993–1009, 2009.

[160] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk. Single-pixel imaging via compressive sampling. *IEEE Signal Processing Magazine*, 25(2):83–91, 2008.

[161] R. F. Marcia, Z. T. Harmany, and R. M. Willett. Compressive coded aperture imaging. In *IS&T/SPIE Electronic Imaging*, pages 72460G–72460G. International Society for Optics and Photonics, 2009.

[162] M. F. Duarte, S. Sarvotham, D. Baron, M. B. Wakin, and R. G. Baraniuk. Distributed compressed sensing of jointly sparse signals. In *Asilomar Conference on Signals, Systems, and Computing*, pages 1537–1541, 2005.

[163] M. A. Davenport, C. Hegde, M. F. Duarte, and R. G. Baraniuk. Joint manifolds for data fusion. *IEEE Transactions on Image Processing*, 19(10):2580–2594, 2010.

[164] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T.-W. Lee, and T. J. Sejnowski. Dictionary learning algorithms for sparse representation. *Neural computation*, 15(2):349–396, 2003.

[165] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, 2006.

[166] M. S. Lewicki and T. J. Sejnowski. Learning overcomplete representations. *Neural computation*, 12(2):337–365, 2000.

[167] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. *arXiv preprint arXiv:0809.3083*, 2008.

[168] R. Rubinstein, A. M. Bruckstein, and M. Elad. Dictionaries for sparse representation modeling. *Proceedings of the IEEE*, 98(6):1045–1057, 2010.

[169] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997.

[170] Z. Jiang, G. Zhang, and L. S. Davis. Submodular dictionary learning for sparse coding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3418–3425. IEEE, 2012.

[171] B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2):227–234, 1995.

[172] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *The Journal of Machine Learning Research*, 5:1457–1469, 2004.

[173] V. K. Potluru, S. M. Plis, J. L. Roux, B. A. Pearlmutter, V. D. Calhoun, and T. P. Hayes. Block coordinate descent for sparse nmf. *arXiv preprint arXiv:1301.3527*, 2013.

[174] J. Kim and H. Park. Sparse nonnegative matrix factorization for clustering. 2008.

[175] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *The Journal of Machine Learning Research*, 11:19–60, 2010.

[176] D. Angelosante, G. B. Giannakis, and E. Grossi. Compressed sensing of time-varying signals. In *16th International Conference on Digital Signal Processing*, pages 1–8. IEEE, 2009.

[177] D. Angelosante and G. B. Giannakis. RLS-weighted lasso for adaptive estimation of sparse signals. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, pages 3245–3248. IEEE, 2009.

[178] D. Angelosante, J. A. Bazerque, and G. B. Giannakis. Online adaptive estimation of sparse signals: Where RLS meets the ℓ_1-norm. *IEEE Transactions on Signal Processing*, 58(7):3436–3447, 2010.

[179] A. Auslender. Asymptotic properties of the fenchel dual functional and applications to decomposition problems. *Journal of optimization theory and applications*, 73:427–449, 1992.

[180] H. Attouch, J. Bolte, and B. F. Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forwardbackward splitting, and regularized gaussseidel methods. *Mathematical Programming*, pages 1–39, 2011.

[181] P. L. Combettes and J.-C Pesquet. Proximal splitting methods in signal processing. 2009. Available online at: arxiv.org.

[182] P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling and Simulation*, 4:1168–1200, 2005.

[183] L. De Lathauwer and J. Castaing. Tensor-based techniques for the blind separation of DS–CDMA signals. *Signal Processing*, 87(2):322–336, 2007.

[184] N. D. Sidiropoulos, R. Bro, and G. B. Giannakis. Parallel factor analysis in sensor array processing. *IEEE Transactions on Signal Processing*, 48(8):2377–2388, 2000.

[185] N. D. Sidiropoulos and R. S. Budampati. Khatri-Rao space-time codes. *IEEE Transactions on Signal Processing*, 50(10):2396–2407, 2002.

[186] N. D. Sidiropoulos and R. Bro. On the uniqueness of multilinear decomposition of N-way arrays. *Journal of chemometrics*, 14(3):229–239, 2000.

[187] T. Jiang and N. D. Sidiropoulos. Kruskal's permutation lemma and the identification of CANDECOMP/PARAFAC and bilinear models with constant modulus constraints. *IEEE Transactions on Signal Processing*, 52(9):2625–2636, 2004.

[188] J. B. Kruskal. Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear algebra and its applications*, 18(2):95–138, 1977.

[189] A. Stegeman and N. D. Sidiropoulos. On Kruskals uniqueness condition for the Candecomp/Parafac decomposition. *Linear Algebra and its applications*, 420(2):540–552, 2007.

[190] J. M. ten Berge. The typical rank of tall three-way arrays. *Psychometrika*, 65(4):525–532, 2000.

[191] J. M. ten Berge and N. D. Sidiropoulos. On uniqueness in CANDECOMP/PARAFAC. *Psychometrika*, 67(3):399–409, 2002.

[192] J. Hastad. Tensor rank is NP-complete. *Journal of Algorithms*, 11:644–654, 1990.

[193] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51:455–500, 2009.

[194] N. K. M. Faber, R. Bro, and P. K. Hopke. Recent developments in CANDECOMP/PARAFAC algorithms: A critical review. *Chemometrics and Intelligent Laboratory Systems*, 65:119–137, 2003.

[195] G. Tomasi and R. Bro. A comparison of algorithms for fitting the parafac model. *Computational Statistics and Data Analysis*, 50:17001734, April 2006.

[196] J. D. Carroll and J. J. Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35:283–319, 1970.

[197] R. A. Harshman. Foundations of the parafac procedure: Models and conditions for an explanatory" multi-modal factor analysis. *UCLA working papers in phonetics*, 16:1–84, 1970.

[198] C. Navasca, L. De Lathauwer, and S. Kindermann. Swamp reducing technique for tensor decomposition. *Proc. 16th European Signal Processing Conference (EUSIPCO)*, August 2008.

[199] S. Borman. The expectation maximization algorithm - a short tutorial. *Unpublished paper.*

[200] V. Hasselblad. Estimation of parameters for a mixture of normal distributions. *Technometrics*, 8:431–444, 1966.

[201] M. Chiang, C. W. Tan, D. P. Palomar, D. O'Neill, and D. Julian. Power control by geometric programming. *IEEE Transactions on Wireless Communications*, 6:2640–2651, 2007.

[202] J. Papandriopoulos and J. S. Evans. Low-complexity distributed algorithms for spectrum balancing in multi-user dsl networks. *IEEE International Conference on Communications (ICC)*, 7:3270–3275, 2006.

[203] Y. Zhang, E. Dall'Anese, and G. B. Giannakis. Distributed robust beamforming for MIMO cognitive networks. *IEEE International Conference on In Acoustics, Speech and Signal Processing (ICASSP)*, 59:2953–2956. 2012.

[204] M. Hong, Q. Li, Y.-F. Liu, and Z.-Q. Luo. Decomposition by successive convex approximation: A unifying approach for linear transceiver design in interfering heterogeneous networks. *arXiv preprint arXiv:1210.1507*, 2012.

[205] J. A. Bazerque, G. Mateos, and G. B. Giannakis. Rank regularization and bayesian inference for tensor completion and extrapolation. *IEEE Transactions on Signal Processing*, 61(22):5689–5703, 2013.

[206] S. Shen, W.-C. Li, and T.-H. Chang. Wireless information and energy transfer in multi-antenna interference channel. *arXiv preprint arXiv:1308.2838*, 2013.

[207] J. A. Bazerque, G. Mateos, and G. B. Giannakis. Inference of poisson count processes using low-rank tensor data. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5989–5993. IEEE, 2013.

[208] Q. Sun, G. Zhu, C. Shen, X. Li, and Z. Zhong. Joint beamforming design and time allocation for wireless powered communication networks. *arXiv preprint arXiv:1403.4492*, 2014.

[209] B. Baingana and G. B. Giannakis. Centrality-constrained graph embedding. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3113–3117. IEEE, 2013.

[210] P. Tsiaflakis, F. Glineur, and M. Moonen. Iterative convex approximation based real-time dynamic spectrum management in multi-user multi-carrier communication systems. 2014.

[211] B. Baingana and G. B. Giannakis. Embedding graphs under centrality constraints for network visualization. *arXiv preprint arXiv:1401.4408*, 2014.

[212] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951.

[213] J. Kiefer and J. Wolfowitz. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 23(3):462–466, 1952.

[214] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

[215] S. Amari. A theory of adaptive pattern classifiers. *IEEE Transactions on Electronic Computers*, (3):299–307, 1967.

[216] R. Wijnhoven and P. H. N. D. With. Fast training of object detection using stochastic gradient descent. In *Proc. IEEE International Conference on Pattern Recognition (ICPR)*, pages 424–427, 2010.

[217] L. Grippo. Convergent on-line algorithms for supervised learning in neural networks. *IEEE Transactions on Neural Networks*, 11(6):1284–1299, 2000.

[218] O. L. Mangasarian and M. V. Solodov. Serial and parallel backpropagation convergence via nonmonotone perturbed minimization. *Optimization Methods and Software*, 4(2):103–116, 1994.

[219] Z.-Q. Luo. On the convergence of the LMS algorithm with adaptive learning rate for linear feedforward networks. *Neural Computation*, 3(2):226–245, 1991.

[220] Z.-Q. Luo and P. Tseng. Analysis of an approximate gradient projection method with applications to the backpropagation algorithm. *Optimization Methods and Software*, 4(2):85–101, 1994.

[221] L. Bottou. Online learning and stochastic approximations. *On-line learning in neural networks*, 17:9, 1998.

[222] D. P. Bertsekas. A new class of incremental gradient methods for least squares problems. *SIAM Journal on Optimization*, 7(4):913–926, 1997.

[223] J. Tsitsiklis, D. P. Bertsekas, and M. Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, 31(9):803–812, 1986.

[224] D. P. Bertsekas. Distributed asynchronous computation of fixed points. *Mathematical Programming*, 27(1):107–120, 1983.

[225] Y. M. Ermol'ev and V. I. Norkin. Stochastic generalized gradient method for nonconvex nonsmooth stochastic optimization. *Cybernetics and Systems Analysis*, 34(2):196–215, 1998.

[226] D. P. Bertsekas. Incremental gradient, subgradient, and proximal methods for convex optimization: a survey. *Optimization for Machine Learning*, page 85, 2011.

[227] P. Tseng. An incremental gradient (-projection) method with momentum term and adaptive stepsize rule. *SIAM Journal on Optimization*, 8(2):506–531, 1998.

[228] Y. Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical programming*, 120(1):221–259, 2009.

[229] L. Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *The Journal of Machine Learning Research*, 11:2543–2596, 2010.

[230] M. Mardani, G. Mateos, and G. B. Giannakis. Dynamic anomalography: Tracking network anomalies via sparsity and low rank. *IEEE Journal of Selected Topics in Signal Processing*, 7(1):50–66, 2013.

[231] M. Mardani, G. Mateos, and G. B. Giannakis. Subspace learning and imputation for streaming big data matrices and tensors. *arXiv preprint arXiv:1404.4667*, 2014.

[232] K. Gomadam, V. R. Cadambe, and S. A. Jafar. Approaching the capacity of wireless networks through distributed interference alignment. In *Global Telecommunications Conference, 2008*, pages 1–6. IEEE, 2008.

[233] 3GPP TR 36.814. In *http://www.3gpp.org/ftp/specs/ archive/36_series/36.814/*.

[234] E. Polak, J. O. Royset, and R. S. Womersley. Algorithms with adaptive smoothing for finite minimax problems. *Journal of Optimization Theory and Applications*, 119(3):459–484, 2003.

[235] S. Serbetli and A. Yener. Transceiver optimization for multiuser MIMO systems. *IEEE Transactions on Signal Processing*, 52(1):214–226, 2004.

[236] D. P. Bertsekas and J. N. Tsitsiklis. Neuro-dynamic programming. 1996.

[237] D. L. Fisk. Quasi-martingales. *Transactions of the American Mathematical Society*, 120:369–389, 1965.

[238] A. W. Van der Vaart. *Asymptotic statistics (Vol. 3)*. Cambridge university press, 2000.

[239] E. Hewitt and L. J. Savage. Symmetric measures on cartesian products. *Transactions of the American Mathematical Society*, 80:470–501, 1955.

[240] J. F. Bonnans and A. Shapiro. *Perturbation analysis of optimization problems*. Springer Verlag, 2000.

[241] D. Aloise, A. Deshpande, P. Hansen, and P. Popat. Np-hardness of euclidean sum-of-squares clustering. *Machine Learning*, 75(2):245–248, 2009.

Appendix A

Proofs

Proof of Theorem 1: First of all since the approximate functions are upper-bounds of the original functions, all the iterates are feasible in the algorithm. Moreover, due to the upper-bound and function value consistency assumptions, it is not hard to see that

$$h_0(x^{r+1}) \leq \tilde{h}_0(x^{r+1}, x^r) \leq \gamma \tilde{h}_0(\hat{x}^r, x^r) + (1-\gamma)\tilde{h}_0(x^r, x^r) \leq \tilde{h}_0(x^r, x^r) = h_0(x^r),$$

where the second inequality is the result of convexity of $\tilde{h}_0(\cdot, x^r)$. Hence, the objective value is nonincreasing and we must have

$$\lim_{r\to\infty} h_0(x^r) = h_0(\bar{x}), \tag{A.1}$$

and

$$\lim_{r\to\infty} \tilde{h}_0(\hat{x}^r, x^r) = h_0(\bar{x}). \tag{A.2}$$

Let $\{x^{r_j}\}_{j=1}^{\infty}$ be the subsequence converging to the limit point $\bar{x}$. Consider any fixed point x' satisfying

$$\tilde{h}_i(x', \bar{x}) < 0, \ \ \forall i = 1, 2, \ldots, m. \tag{A.3}$$

Then for j sufficiently large, we must have

$$\tilde{h}_i(x', x^{r_j}) < 0, \ \ \forall i = 1, 2, \ldots, m,$$

i.e., x' is a strictly feasible point at the iteration r_j. Therefore,

$$\tilde{h}_0(\hat{x}^{r_j}, x^{r_j}) \leq \tilde{h}_0(x', x^{r_j}),$$

due to the definition of $\hat{x}^{r_j}$. Letting $j \to \infty$ and using (A.2), we have

$$\tilde{h}_0(\bar{x}, \bar{x}) \leq \tilde{h}_0(x', \bar{x}).$$

Notice that this inequality holds for any x' satisfying (A.3). Combining this fact with the convexity of $\tilde{h}_i(\cdot, \bar{x})$ and the Slater condition implies that

$$\begin{aligned} \bar{x} \in \arg\min_{x} \ & \tilde{h}_0(x, \bar{x}) \\ \text{s.t.} \ \ & \tilde{h}_i(x, \bar{x}) \leq 0, \ \forall i = 1, \ldots, m. \end{aligned}$$

Since the Slater condition is satisfied, using the gradient consistency assumption, the KKT condition of the above optimization problem implies that there exist $\lambda_1, \ldots, \lambda_m \geq 0$ such that

$$\begin{aligned} & 0 \in \nabla f_0(\bar{x}) + \partial g_0(\bar{x}) + \sum_{i=1}^{m} \lambda_i \left(\nabla f_i(\bar{x}) + \partial g_i(\bar{x}) \right) \\ & \tilde{f}_i(\bar{x}, \bar{x}) + g_i(\bar{x}) \leq 0, \ \forall i = 1, \ldots, m, \\ & \lambda_i \left(\tilde{f}_i(\bar{x}, \bar{x}) + g_i(\bar{x}) \right) = 0, \ \forall i = 1, \ldots, m. \end{aligned}$$

Using the upper-bound and the objective value consistency assumptions, we have

$$\begin{aligned} & 0 \in \nabla f_0(\bar{x}) + \partial g_0(\bar{x}) + \sum_{i=1}^{m} \lambda_i \left(\nabla f_i(\bar{x}) + \partial g_i(\bar{x}) \right) \\ & f_i(\bar{x}) + g_i(\bar{x}) \leq 0, \ \forall i = 1, \ldots, m, \\ & \lambda_i \left(f_i(\bar{x}) + g_i(\bar{x}) \right) = 0, \ \forall i = 1, \ldots, m, \end{aligned}$$

which completes the proof. ■

Proof of Theorem 2: The proof of part (a) is similar to the one in [13] for block coordinate descent approach. First of all, since a locally tight upper bound of $f(\cdot)$ is minimized at each iteration, we have

$$f(x^0) \geq f(x^1) \geq f(x^2) \geq \dots. \tag{A.4}$$

Consider a limit point z. Combining (A.4) with the continuity of $f(\cdot)$ implies

$$\lim_{r\to\infty} f(x^r) = f(z). \tag{A.5}$$

Let us consider the subsequence $\{x^{r_j}\}$ converging to the limit point z. Since the number of blocks is finite, there exists a block which is updated infinitely often in the subsequence $\{r_j\}$. Without loss of generality, we assume that block n is updated infinitely often. Thus, by further restricting to a subsequence, we can write

$$x_n^{r_j} = \arg\min_{x_n} \quad u_n(x_n, x^{r_j-1}).$$

Now we prove that $x^{r_j+1} \to z$, in other words, we will show that $x_1^{r_j+1} \to z_1$. Assume the contrary that $x_1^{r_j+1}$ does not converge to z_1. Therefore by further restricting to a subsequence, there exists $\bar{\gamma} > 0$ such that

$$\bar{\gamma} \leq \gamma^{r_j} = \|x_1^{r_j+1} - x_1^{r_j}\|, \ \forall\ r_j.$$

Let us normalize the difference between $x_1^{r_j}$ and $x_1^{r_j+1}$, i.e.,

$$s^{r_j} \triangleq \frac{x_1^{r_j+1} - x_1^{r_j}}{\gamma^{r_j}}.$$

Notice that $\|s^{r_j}\| = 1$, thus s^{r_j} belongs to a compact set and it has a limit point $\bar{s}$. By

further restricting to a subsequence that converges to $\bar{s}$, using (2.4) and (2.5), we obtain

$$\begin{aligned} f(x^{r_j+1}) &\leq u_1(x_1^{r_j+1}, x^{r_j}) && \text{(A.6)} \\ &= u_1(x_1^{r_j} + \gamma^{r_j} s^{r_j}, x^{r_j}) && \text{(A.7)} \\ &\leq u_1(x_1^{r_j} + \epsilon\bar{\gamma} s^{r_j}, x^{r_j}), \quad \forall\, \epsilon \in [0,1] && \text{(A.8)} \\ &\leq u_1(x_1^{r_j}, x^{r_j}) && \text{(A.9)} \\ &= f(x^{r_j}), && \text{(A.10)} \end{aligned}$$

where (A.6) and (A.10) hold due to (2.4) and (2.5). The inequalities (A.8) and (A.9) are the result of quasi-convexity of $u(\cdot, x^{r_j})$. Letting $j \to \infty$ and combining (A.6), (A.8), (A.5), and (A.10) imply

$$f(z) \leq u_1(z_1 + \epsilon\bar{\gamma}\bar{s}, z) \leq f(z), \quad \forall\, \epsilon \in [0,1],$$

or equivalently

$$f(z) = u_1(z_1 + \epsilon\bar{\gamma}\bar{s}, z), \quad \forall\, \epsilon \in [0,1]. \tag{A.11}$$

Furthermore,

$$\begin{aligned} u_1(x_1^{r_{j+1}}, x^{r_{j+1}}) &= f(x^{r_{j+1}}) \leq f(x^{r_j+1}) \\ &\leq u_1(x_1^{r_j+1}, x^{r_j}) \leq u_1(x_1, x^{r_j}), \quad \forall\, x_1 \in \mathcal{X}_1. \end{aligned}$$

Letting $j \to \infty$, we obtain

$$u_1(z_1, z) \leq u_1(x_1, z), \quad \forall\, x_1 \in \mathcal{X}_1,$$

which further implies that z_1 is the minimizer of $u_1(\cdot, z)$. On the other hand, we assume that the minimizer is unique, which contradicts (A.11). Therefore, the contrary assumption is not true, i.e., $x^{r_j+1} \to z$.

Since $x_1^{r_j+1} = \arg\min_{x_1 \in \mathcal{X}_1} \quad u_1(x_1, x^{r_j})$, we get

$$u_1(x_1^{r_j+1}, x^{r_j}) \leq u_1(x_1, x^{r_j}) \quad \forall\, x_1 \in \mathcal{X}_1.$$

Taking the limit $j \to \infty$ implies

$$u_1(z_1, z) \leq u_1(x_1, z) \quad \forall\, x_1 \in \mathcal{X}_1,$$

which further implies

$$u_1'(x_1, z; d_1)\Big|_{x_1 = z_1} \geq 0, \quad \forall\, d_1 \in \mathbb{R}^{m_1} \quad \text{with} \quad z_1 + d_1 \in \mathcal{X}_1.$$

Similarly, by repeating the above argument for the other blocks, we obtain

$$u_k'(x_k, z; d_k)\Big|_{x_k = z_k} \geq 0, \quad \forall\, d_k \in \mathbb{R}^{m_k} \quad \text{with} \quad d_k + z_k \in \mathcal{X}_k, \qquad \forall\, k = 1, \ldots, n. \tag{A.12}$$

Combining (2.6) and (A.12) implies

$$f'(z; d) \geq 0, \quad \forall\, d = (0, \ldots, d_k, \ldots, 0) \quad \text{s.t.} \quad d + z \in \mathcal{X},\ \forall\, k$$

in other words, z is the coordinatewise minimum of $f(\cdot)$.

Now we prove part (b) of the theorem. Without loss of generality, let us assume that (2.3) has a unique solution at every point x^{r-1} for $i = 1, 2, \ldots, n-1$. Since the iterates lie in a compact set, we only need to show that every limit point of the iterates is a stationary point of $f(\cdot)$. To do so, let us consider a subsequence $\{x^{r_j}\}$ which converges to a limit point $z \in \mathcal{X}^0 \subseteq \mathcal{X}$. Since the number of blocks is finite, there exists a block i which is updated infinitely often in the subsequence $\{x^{r_j}\}$. By further restricting to a subsequence, we can assume that

$$x_i^{r_j} \in \arg\min_{x_i} \quad u_i(x_i, x^{r_j - 1}).$$

Since all the iterates lie in a compact set, we can further restrict to a subsequence such that

$$\lim_{j \to \infty} \quad x^{r_j - i + k} = z^k, \quad \forall\, k = 0, 1, \ldots, n,$$

where $z^k \in \mathcal{X}^0 \subseteq \mathcal{X}$ and $z^i = z$. Moreover, due to the update rule in the algorithm, we

have

$$u_k(x_k^{r_j-i+k}, x^{r_j-i+k-1}) \leq u_k(x_k, x^{r_j-i+k-1}), \quad \forall\, x_k \in \mathcal{X}_k, \qquad k = 1, 2, \ldots, n.$$

Taking the limit $j \to \infty$, we obtain

$$u_k(z_k^k, z^{k-1}) \leq u_k(x_k, z^{k-1}), \quad \forall\, x_k \in \mathcal{X}_k, \qquad k = 1, 2, \ldots, n. \tag{A.13}$$

This, plus (2.4) and (2.5), implies

$$f(z^k) \leq u_k(z_k^k, z^{k-1}) \leq u_k(z_k^{k-1}, z^{k-1}) = f(z^{k-1}), \quad k = 1, \ldots, n. \tag{A.14}$$

On the other hand, the objective function is non-increasing in the algorithm and it has a limit. Thus, due to the continuity of $f(\cdot)$, we have

$$f(z^0) = f(z^1) = \ldots = f(z^n). \tag{A.15}$$

Using (A.14), (A.15), and (A.13), we obtain

$$f(z) = u_k(z_k^k, z^{k-1}) \leq u_k(x_k, z^{k-1}),\ \forall\, x_k \in \mathcal{X}_k, \quad k = 1, 2, \ldots, n. \tag{A.16}$$

Furthermore, $f(z) = f(z^{k-1}) = u_k(z_k^{k-1}, z^{k-1})$ and therefore,

$$u_k(z_k^{k-1}, z^{k-1}) \leq u_k(x_k, z^{k-1}),\ \forall\, x_k \in \mathcal{X}_k, \quad k = 1, 2, \ldots, n. \tag{A.17}$$

The inequalities (A.16) and (A.17) imply that z_k^{k-1} and z_k^k are both the minimizer of $u_k(\cdot, z^{k-1})$. However, according to our assumption, the minimizer is unique for $k = 1, 2, \ldots, n-1$ and therefore,

$$z^0 = z^1 = z^2 = \ldots = z^{n-1} = z$$

Plugging the above relation in (A.13) implies

$$u_k(z_k, z) \leq u_k(x_k, z),\ \forall\, x_k \in \mathcal{X}_k, \quad k = 1, 2, \ldots, n-1. \tag{A.18}$$

Moreover, by setting $k = n$ in (A.17), we obtain

$$u_n(z_n, z) \leq u_n(x_n, z), \quad \forall\, x_n \in \mathcal{X}_n. \tag{A.19}$$

The inequalities (A.18) and (A.19) imply that

$$u_k'(x_k, z; d_k)\Big|_{x_k = z_k} \geq 0, \quad \forall\, d_k \in \mathbb{R}^{m_k} \text{ with } z_k + d_k \in \mathcal{X}_k, \quad k = 1, 2, \ldots, n.$$

Combining this with (2.6) yields

$$f'(z; d) \geq 0, \quad \forall\, d = (0, \ldots, d_k, \ldots, 0) \;\text{ with }\; z_k + d_k \in \mathcal{X}_k, \quad k = 1, 2, \ldots, n,$$

which implies the stationarity of the point z due to the regularity of $f(\cdot)$. ■

Proof of Theorem 3: First of all, due to update rule of the algorithm and the upper-bound assumption, one can write

$$\begin{aligned}\mathbb{E}\left[f(x^{r+1}) \mid x^r\right] &\leq \sum_{i=1}^{n} p_i^r \min_{x_i} u_i(x_i, x^r) \\ &= f(x^r) - \sum_{i=1}^{n} p_i^r \left(f(x^r) - \min_{x_i} u_i(x_i, x^r)\right),\end{aligned}$$

which implies that $f(x^r)$ is a supermartingale; therefore $f(x^r)$ converges [236, Proposition 4.2], and

$$\sum_{r=1}^{\infty} \sum_{i=1}^{n} p_i^r \left(f(x^r) - \min_{x_i} u_i(x_i, x^r)\right) < \infty, \text{ almost surely.}$$

Since $p_i^r \geq p_{\min} > 0,\ \forall i, r$, we must have that

$$\lim_{r \to \infty} \left((x^r) - \min_{x_i} u_i(x_i, x^r)\right) = 0,\ \forall i, \quad \text{almost surely.} \tag{A.20}$$

Now let us restrict our analysis to the set of realizations for which the above result holds. Consider a limit point $\bar{x}$ with $\{x^{r_j}\}$ converging to $\bar{x}$. Since $\lim_{r \to \infty} f(x^r) = f(\bar{x})$,

from (A.20) we obtain

$$\lim_{j\to\infty} \min_{x_i} u_i(x_i, x^{r_j}) = f(\bar{x}),\ \forall i. \tag{A.21}$$

Furthermore, clearly, one can write

$$\min_{x_i} u_i(x_i, x^{r_j}) \leq u_i(y_i, x^{r_j}),\ \forall y_i \in \mathcal{X}_i,\ \forall i. \tag{A.22}$$

Combining (A.21) and (A.22), we obtain

$$f(\bar{x}) \leq u_i(y_i, \bar{x}),\ \forall y_i \in \mathcal{X}_i,\ \forall i,$$

or in other words, due to the function value consistency assumption, we have

$$u_i(\bar{x}_i, \bar{x}) \leq u_i(y_i, \bar{x}),\ \forall y_i \in \mathcal{X}_i,\ \forall i.$$

Checking the first order optimality condition combined with the gradient consistency assumption will complete the proof. ■

Proof of Theorem 6: Let us define $R_i(y)$ to be the minimum objective value of the i-th subproblem at a point y, i.e.,

$$R_i(y) \triangleq \min_{x_i} \quad u_i(x_i, y).$$

Using a similar argument as in Theorem 2, we can show that the sequence of the objective function values are non-increasing, that is

$$f(x^r) = u_i(x_i^r, x^r) \geq R_i(x^r) \geq f(x^{r+1}).$$

Let $\{x^{r_j}\}$ be the subsequence converging to a limit point z. For every fixed block index

$i = 1, 2, \ldots, n$ and every $x_i \in \mathcal{X}_i$, we have the following series of inequalities

$$\begin{aligned} u_i(x_i, x^{r_j}) &\geq R_i(x^{r_j}) \\ &\geq u_k(x_k^{r_j+1}, x^{r_j}) \\ &\geq f(x^{r_j+1}) \\ &\geq f(x^{r_{j+1}}) \\ &= u_i(x_i^{r_{j+1}}, x^{r_{j+1}}), \end{aligned}$$

where we use k to index the block that provides the maximum improvement at iteration $r_j + 1$. The first and the second inequalities are due to the definition of the function $R_i(\cdot)$ and the MISUM update rule, respectively. The third inequality is implied by the upper bound assumption (2.5), while the last inequality is due to the non-increasing property of the objective values.

Letting $j \to \infty$, we obtain

$$u_i(x_i, z) \geq u_i(z_i, z), \quad \forall\, x_i \in \mathcal{X}_i, \quad i = 1, 2, \ldots, n.$$

The first order optimality condition implies

$$u_i'(x_i, z; d_i)\Big|_{x_i = z_i} \geq 0, \quad \forall\, d_i \ \text{ with } \ z_i + d_i \in \mathcal{X}_i, \quad \forall\, i = 1, 2, \ldots, n.$$

Combining this with (2.6) yields

$$f'(z; d) \geq 0, \quad \forall\, d = (0, \ldots, d_i, \ldots, 0) \ \text{ with } \ z_i + d_i \in \mathcal{X}_i, \quad i = 1, 2, \ldots, n.$$

In other words, z is the coordinatewise minimum of $f(\cdot)$. ■

Proof of Theorem 7: First of all, due to the use of Armijo step size selection rule, we have

$$f(x^r) - f(x^{r+1}) \geq -\sigma \alpha^r f'(x^r; d^r) \geq 0. \tag{A.23}$$

Consider a limit point z and a subsequence $\{x^{r_j}\}_j$ converging to z. Since $\{f(x^r)\}$ is a

monotonically decreasing sequence, it follows that

$$\lim_{r\to\infty} f(x^r) = f(z).$$

Moreover, (A.23) implies

$$\lim_{r\to\infty} \alpha^r f'(x^r; d^r) = 0. \tag{A.24}$$

By further restricting to a subsequence if necessary, we can assume without loss of generality that in the subsequence $\{x^{r_j}\}_j$ the first block is updated. We first claim that we can restrict to a further subsequence if necessary so that

$$\lim_{j\to\infty} d^{r_j+1} = 0. \tag{A.25}$$

We prove this by contradiction. Let us assume the contrary so that there exists a $\delta,\ 0 < \delta < 1$ and an $\ell \in \{1, 2, \ldots\}$ with

$$\|d^{r_j+1}\| \geq \delta,\ \forall\ j \geq \ell. \tag{A.26}$$

Define $p^{r_j+1} = \frac{d^{r_j+1}}{\|d^{r_j+1}\|}$. The equation (A.24) implies $\alpha^{r_j+1}\|d^{r_j+1}\| f'(x^{r_j}; p^{r_j+1}) \to 0$. We consider the following two cases:

Case A: $f'(x^{r_j}; p^{r_j+1}) \to 0$ along a subsequence of $\{r_j\}$. Let us restrict ourselves to that subsequence. Since $\|p^{r_j+1}\| = 1$, there exists a limit point $\bar{p}$. By further restricting to a subsequence and using the smoothness of $f(\cdot)$, we obtain

$$f'(z; \bar{p}) = 0. \tag{A.27}$$

Furthermore, due to the strict convexity of $h_1(\cdot, z)$,

$$h_1(z_1 + \delta\bar{p}_1, z) > h_1(z_1, z) + \delta h_1'(x_1, z; \bar{p}_1)\Big|_{x_1=z_1} \geq h_1(z_1, z), \tag{A.28}$$

where $\bar{p}_1$ is the first block of $\bar{p}$ and the last step is due to (A.27) and (2.8). On the other hand, since $x_1^{r_j+1} + \delta p_1^{r_j}$ lies between $x_1^{r_j}$ and $y_1^{r_j}$, we have (from the convexity of $h_1(\cdot, x^{r_j})$)

$$h_1(x_1^{r_j} + \delta p_1^{r_j+1}, x^{r_j}) \leq h_1(x_1^{r_j}, x^{r_j}).$$

Letting $j \to \infty$ along the subsequence, we obtain

$$h_1(z_1 + \delta\bar{p}_1, z) \leq h_1(z_1, z), \tag{A.29}$$

which contradicts (A.28).

Case B: $\alpha^{r_j+1}\|d^{r_j+1}\| \to 0$ along a subsequence. Let us restrict ourselves to that subsequence. Due to the hypothesis (A.26),

$$\lim_{j\to\infty} \alpha^{r_j+1} = 0,$$

which further implies that there exists $j_0 \in \{1, 2, \ldots\}$ such that

$$f(x^{r_j} + \frac{\alpha^{r_j+1}}{\beta} d^{r_j+1}) - f(x^{r_j}) > \sigma \frac{\alpha^{r_j+1}}{\beta} f'(x^{r_j}; d^{r_j+1}), \quad \forall\, j \geq j_0.$$

Rearranging the terms, we obtain

$$\frac{f(x^{r_j} + \frac{\alpha^{r_j+1}}{\beta}\|d^{r_j+1}\|p^{r_j+1}) - f(x^{r_j})}{\frac{\alpha^{r_j+1}}{\beta}\|d^{r_j+1}\|} > \sigma f'(x^{r_j}; p^{r_j+1}), \quad \forall\, j \leq j_0.$$

Letting $j \to \infty$ along the subsequence that $p^{r_j+1} \to \bar{p}$, we obtain

$$f'(z; \bar{p}) \geq \sigma f'(z; \bar{p}),$$

which implies $f(z; \bar{p}) \geq 0$ since $\sigma < 1$. Therefore, using an argument similar to the previous case, (A.28) and (A.29) hold, which is a contradiction. Thus, the assumption (A.26) must be false and the condition (A.25) must hold. On the other hand, $y_1^{r_j+1}$ is the minimizer of $h_1(\cdot, x^{r_j})$; thus,

$$h_1(y_1^{r_j+1}, x^{r_j}) \leq h_1(x_1, x^{r_j}), \quad \forall\, x_1 \in \mathcal{X}_1. \tag{A.30}$$

Note that $y_1^{r_j+1} = x_1^{r_j} + d_1^{r_j+1}$. Combining (A.25) and (A.30) and letting $j \to \infty$ yield

$$h_1(z_1, z) \leq h_1(x_1, z), \quad \forall\, x_1 \in \mathcal{X}_1.$$

The first order optimality condition and assumption (2.8) imply

$$f'(z;d) \geq 0,\ \forall\ d = (d_1, 0, \ldots, 0) \quad \text{with} \quad z_1 + d_1 \in \mathcal{X}_1.$$

On the other hand, since $d^{r_j+1} \to 0$, it follows that

$$\lim_{j\to\infty} x^{r_j+1} = z.$$

Therefore, by restricting ourselves to the subsequence that $d^{r_j+1} \to 0$ and repeating the above argument n times, we obtain

$$f'(z;d) \geq 0, \quad \forall\ d = (0, \ldots, d_k, \ldots, 0) \quad \text{with} \quad z_k + d_k \in \mathcal{X}_k;\ k = 1, \ldots, n.$$

Using the regularity of $f(\cdot)$ at point z completes the proof. ■

Proof of Theorem 9: We will first prove that $\lim_{r\to\infty} \|\hat{x}^r - x^r\| = 0$, with probability one. To show this, let us first bound the change in the objective value in the consecutive steps of the algorithm:

$$\begin{aligned}
h(x^{r+1}) &= f(x^{r+1}) + \sum_i g_i(x_i^{r+1}) \\
&= f(x^{r+1}) + \sum_{i\notin S^r} g_i(x_i^r) + \sum_{i\in S^r} g_i\left(x_i^r + \gamma^r(\hat{x}_i^r - x_i^r)\right) \\
&\leq f(x^{r+1}) + \sum_i g_i(x_i^r) + \gamma^r \sum_{i\in S^r} \left(g_i(\hat{x}_i^r) - g_i(x_i^r)\right) \\
&\leq f(x^r) + \gamma^r \langle \nabla_x f(x^r), \hat{x}^r - x^r\rangle_{S^r} + \frac{(\gamma^r)^2 L_{\nabla F}}{2}\|\hat{x}^r - x^r\|_{S^r}^2 \\
&\quad + \sum_i g_i(x_i^r) + \gamma^r \sum_{i\in S^r} \left(g_i(\hat{x}_i^r) - g_i(x_i^r)\right) \\
&= h(x^r) + \frac{(\gamma^r)^2 L_{\nabla f}}{2}\|\hat{x}^r - x^r\|_{S^r}^2 \\
&\quad + \gamma^r \left(\langle \nabla_x f(x^r), \hat{x}^r - x^r\rangle_{S^r} + \sum_{i\in S^r} \left(g_i(\hat{x}_i^r) - g_i(x_i^r)\right)\right),
\end{aligned} \tag{A.31}$$

where the first inequality is due to convexity of $g(\cdot)$; the second inequality is due to the

Lipschitz continuity of $\nabla f(\cdot)$; and we have also use the notation $\langle a, b\rangle_S \triangleq \sum_{i\in S}\langle a_i, b_i\rangle$ and $\|a\|_S^2 \triangleq \langle a, a\rangle$. In order to get a typical sufficient decrease bound, we next need to bound the last term in (A.31) by noticing that $\tilde{h}_i$ is strongly convex and therefore using the definition of $\hat{x}_i^r$, we have

$$\tilde{h}_i(x_i^r, x^r) \geq \tilde{h}_i(\hat{x}_i^r, x^r) + \frac{\tau_{\min}}{2}\|\hat{x}_i^r - x_i^r\|^2,\ \forall i \in S^r,$$

where $\tau_{\min} \triangleq \min_i \tau_i$. Substituting the definition of $\tilde{h}_i$ and multiplying both sides by minus one imply

$$-\tilde{f}_i(x_i^r, x^r) - g_i(x_i^r) \leq -\tilde{f}_i(\hat{x}_i^r, x^r) - g_i(\hat{x}_i^r) - \frac{\tau_{\min}}{2}\|\hat{x}_i^r - x_i^r\|^2.$$

Linearizing the smooth part and using the gradient consistency assumption (2.16) lead to

$$\langle \nabla_{x_i} f(x^r), \hat{x}_i^r - x_i^r\rangle + g_i(\hat{x}_i^r) - g_i(x_i^r) \leq -\frac{\tau_{\min}}{2}\|\hat{x}_i^r - x_i^r\|^2.$$

Summing up the above inequality over all $i \in S^r$, we obtain

$$\langle \nabla_x f(x^r), \hat{x}^r - x^r\rangle_{S^r} + \sum_{i\in S^r}(g_i(\hat{x}_i^r) - g_i(x_i^r)) \leq -\frac{\tau_{\min}}{2}\|\hat{x}^r - x^r\|_{S^r}^2, \tag{A.32}$$

where $\hat{x}^r \triangleq (\hat{x}_i^r)_{i=1}^n$. Combining (A.31) and (A.32) leads to

$$h(x^{r+1}) \leq h(x^r) + \frac{\gamma^r(-\tau_{\min} + \gamma^r L_{\nabla f})}{2}\|\hat{x}^r - x^r\|_{S^r}^2.$$

Since $\limsup_{r\to\infty}\gamma^r < \bar{\gamma}$, for sufficiently large r, there exists $\beta > 0$ such that

$$h(x^{r+1}) \leq h(x^r) - \beta\gamma^r\|\hat{x}^r - x^r\|_{S^r}^2. \tag{A.33}$$

Taking the conditional expectation from both sides implies

$$\mathbb{E}[h(x^{r+1}) \mid x^r] \leq h(x^r) - \beta\gamma^r\mathbb{E}[\sum_{i=1}^n R_i^r\|\hat{x}_i^r - x_i^r\|^2 \mid x^r], \tag{A.34}$$

where R_i^r is a Bernoulli random variable which is one if $i \in S^r$ and it is zero otherwise.

Clearly, $\mathbb{E}[R_i^r \mid x^r] = p_i^r$ and therefore,

$$\mathbb{E}[h(x^{r+1}) \mid x^r] \leq h(x^r) - \beta\gamma^r p_{\min}\|\hat{x}^r - x^r\|^2, \ \ \forall r, \tag{A.35}$$

and therefore $\{h(x^r)\}$ is a supermartingale and by the supermartingale convergence theorem [236, Proposition 4.2], $h(x^r)$ converges and we have

$$\sum_{r=1}^{\infty} \gamma^r \|\hat{x}^r - x^r\|^2 < \infty, \quad \text{almost surely.} \tag{A.36}$$

Let us now restrict our analysis to the set of probability one for which $h(x^r)$ converges and $\sum_{r=1}^{\infty} \gamma^r \|\hat{x}^r - x^r\|^2 < \infty$. Fix a realization in that set. The equation (A.36) simply implies that, in the considered set of realizations, $\liminf_{r\to\infty} \|\hat{x}^r - x^r\| = 0$, since $\sum_r \gamma^r = \infty$. Next we strengthen this result by proving that $\lim_{r\to\infty} \|\hat{x}^r - x^r\| = 0$ over the considered set of probability one. Consider the contrary that there exists $\delta > 0$ such that $\Delta^r \triangleq \|\hat{x}^r - x^r\| \geq 2\delta$ infinitely often. Since $\liminf_{r\to\infty} \Delta^r = 0$, there exists a subset of indices $\mathcal{K}$ and $\{i_r\}$ such that for any $r \in \mathcal{K}$,

$$\Delta^r < \delta \tag{A.37}$$

$$2\delta < \Delta^{i_r} \tag{A.38}$$

$$\delta \leq \Delta^j \leq 2\delta, \ \forall j = r+1, \ldots, i_r - 1. \tag{A.39}$$

Clearly,

$$\begin{aligned}
\delta - \Delta^r &\overset{\text{(i)}}{\leq} \Delta^{r+1} - \Delta^r = \|\hat{x}^{r+1} - x^{r+1}\| - \|\hat{x}^r - x^r\| \\
&\overset{\text{(ii)}}{\leq} \|\hat{x}^{r+1} - \hat{x}^r\| + \|x^{r+1} - x^r\| \\
&\overset{\text{(iii)}}{\leq} (1 + \hat{L})\|x^{r+1} - x^r\| \\
&\overset{\text{(iv)}}{=} (1 + \hat{L})\gamma^r \|\hat{x}^r - x^r\| \leq (1 + \hat{L})\gamma^r \delta,
\end{aligned} \tag{A.40}$$

where (i) and (ii) are due to (A.39) and the triangle inequality, respectively. The inequality (iii) is the result of Lemma 1 with $\hat{L} \triangleq \frac{\sqrt{n}\tilde{L}}{\tau_{\min}}$ defined in Lemma 1; and (iv) is followed from the iteration update rule of the algorithm. Since $\limsup_{r\to\infty} \gamma^r < \frac{1}{1+\hat{L}}$,

the above inequality implies that for r large enough, there exists an $\alpha > 0$ such that

$$\Delta^r > \alpha. \tag{A.41}$$

Furthermore, since the chosen realization satisfies (A.36), we have that

$$\lim_{r\to\infty} \sum_{t=r}^{i_r-1} \gamma^t (\Delta^t)^2 = 0. \tag{A.42}$$

Combining (A.39), (A.41), and (A.42), we obtain

$$\lim_{r\to\infty} \sum_{t=r}^{i_r-1} \gamma^t = 0. \tag{A.43}$$

On the other hand, using the similar reasoning as in above, we have

$$\begin{aligned}
\delta < \Delta^{i_r} - \Delta^r &= \|\hat{x}^{i_r} - x^{i_r}\| - \|\hat{x}^r - x^r\| \\
&\le \|\hat{x}^{i_r} - \hat{x}^r\| + \|x^{i_r} - x^r\| \\
&\le (1+\hat{L}) \sum_{t=r}^{i_r-1} \gamma^t \|\hat{x}^t - x^t\| \\
&\le 2\delta(1+\hat{L}) \sum_{t=r}^{i_r-1} \gamma^t,
\end{aligned}$$

and hence $\liminf_{r\to\infty} \sum_{t=r}^{i_r-1} \gamma^t > 0$, which contradicts (A.43). Therefore the contrary assumption does not hold and we must have $\lim_{r\to\infty} \|\hat{x}^r - x^r\| = 0,$ almost surely. Consider a limit point $\bar{x}$ with the subsequence $\{x^{r_j}\}_{j=1}^{\infty}$ converging to $\bar{x}$. Using the definition of $\hat{x}^{r_j}$, we have

$$\lim_{j\to\infty} \tilde{h}_i(\hat{x}_i^{r_j}, x^{r_j}) \le \tilde{h}_i(x_i, x^{r_j}), \quad \forall x_i \in \mathcal{X}_i,\ \forall i.$$

Therefore, by letting $j \to \infty$ and using the fact that $\lim_{r\to\infty} \|\hat{x}^r - x^r\| = 0,$ almost surely, we obtain

$$\tilde{h}_i(\bar{x}_i, \bar{x}) \le \tilde{h}_i(x_i, \bar{x}), \quad \forall x_i \in \mathcal{X}_i,\ \forall i,\ \text{almost surely},$$

which in turn, using the gradient consistency assumption, implies

$$\langle \nabla f(\bar{x}) + d, x - \bar{x} \rangle \geq 0, \ \forall x \in \mathcal{X},$$

for some $d \in \partial g(\bar{x})$, i.e., $\bar{x}$ is a stationary point of (2.12) with probability one. ■

Proof of Theorem 11:

To simplify the presentation of the proof, let us define

$$\tilde{y}_i^r \triangleq \arg\min_{y_i \in \mathcal{X}_i} \langle \nabla_{x_i} f(x^r), y_i - x_i^r \rangle + g_i(y_i) + \frac{1}{2}\|y_i - x_i^r\|^2.$$

Clearly, $\tilde{\nabla} h(x^r) = (x_i^r - \tilde{y}_i^r)_{i=1}^n$. The first order optimality condition of the above optimization problem implies

$$\langle \nabla_{x_i} f(x^r) + \tilde{y}_i^r - x_i^r, x_i - \tilde{y}_i^r \rangle + g_i(x_i) - g_i(\tilde{y}_i^r) \geq 0, \quad \forall x_i \in \mathcal{X}_i. \tag{A.44}$$

Furthermore, based on the definition of $\hat{x}_i^r$, we have

$$\langle \nabla_{x_i} \tilde{f}_i(\hat{x}_i^r, x^r), x_i - \hat{x}_i^r \rangle + g_i(x_i) - g_i(\hat{x}_i^r) \geq 0, \quad \forall x_i \in \mathcal{X}_i. \tag{A.45}$$

Plugging in the points $\hat{x}_i^r$ and $\tilde{y}_i^r$ in (A.44) and (A.45); and summing up the two equations will yield to

$$\langle \nabla_{x_i} \tilde{f}_i(\hat{x}_i^r, x^r) - \nabla_{x_i} f(x^r) + x_i^r - \tilde{y}_i^r, \tilde{y}_i^r - \hat{x}_i^r \rangle \geq 0.$$

Using the gradient consistency assumption, we can write

$$\langle \nabla_{x_i} \tilde{f}_i(\hat{x}_i^r, x^r) - \nabla_{x_i} \tilde{f}_i(x_i^r, x^r) + x_i^r - \hat{x}_i^r + \hat{x}_i^r - \tilde{y}_i^r, \tilde{y}_i^r - \hat{x}_i^r \rangle \geq 0,$$

or equivalently,

$$\langle \nabla_{x_i} \tilde{f}_i(\hat{x}_i^r, x^r) - \nabla_{x_i} \tilde{f}_i(x_i^r, x^r) + x_i^r - \hat{x}_i^r, \tilde{y}_i^r - \hat{x}_i^r \rangle \geq \|\hat{x}_i^r - \tilde{y}_i^r\|^2.$$

Applying the Cauchy-Schwarz and the triangle inequality will yield to

$$\Big(\|\nabla_{x_i}\tilde{f}_i(\hat{x}_i^r, x^r) - \nabla_{x_i}\tilde{f}_i(x_i^r, x^r)\| + \|x_i^r - \hat{x}_i^r\|\Big)\,\|\tilde{y}_i^r - \hat{x}_i^r\| \geq \|\hat{x}_i^r - \tilde{y}_i^r\|^2.$$

Since the function $\tilde{f}_i(\cdot, x)$ is Lipschitz, we must have

$$\|\hat{x}_i^r - \tilde{y}_i^r\| \leq (1 + L_i)\|x_i^r - \hat{x}_i^r\| \tag{A.46}$$

Using the inequality (A.46), the norm of the proximal gradient of the objective can be bounded by

$$\begin{aligned}
\|\tilde{\nabla} h(x^r)\|^2 &= \sum_{i=1}^{n} \|x_i^r - \tilde{y}_i^r\|^2 \\
&\leq 2\sum_{i=1}^{n} \left(\|x_i^r - \hat{x}_i^r\|^2 + \|\hat{x}_i^r - \tilde{y}_i^r\|^2\right) \\
&\leq 2\sum_{i=1}^{n} \left(\|x_i^r - \hat{x}_i^r\|^2 + (1 + L_i)^2\|x_i^r - \hat{x}_i^r\|^2\right) \\
&\leq 2(2 + 2L + L^2)\|\hat{x}^r - x^r\|^2.
\end{aligned}$$

Combining the above inequality with the sufficient decrease bound in (A.34), one can write

$$\begin{aligned}
\sum_{r=0}^{T} \mathbb{E}\left[\|\tilde{\nabla} h(x^r)\|^2\right] &\leq \sum_{r=1}^{T} 2(2 + 2L + L^2)\mathbb{E}\left[\|\hat{x}^r - x^r\|^2\right] \\
&\leq \sum_{r=0}^{T} \frac{2(2 + 2L + L^2)}{\hat{\beta}}\mathbb{E}\left[h(x^r) - h(x^{r+1})\right] \\
&\leq \frac{2(2 + 2L + L^2)}{\hat{\beta}}\mathbb{E}\left[h(x^0) - h(x^{T+1})\right] \\
&\leq \frac{2(2 + 2L + L^2)}{\hat{\beta}}\left[h(x^0) - h^*\right] = \kappa,
\end{aligned}$$

which implies that $T_\epsilon \leq \frac{\kappa}{\epsilon}$. ■

Proof of Lemma 4:

The proof requires the use of quasi martingale convergence theorem [237], much like the convergence proof of online learning algorithms [175, Proposition 3]. In particular, we will show that the sequence $\{\hat{f}^r(x^r)\}_{r=1}^{\infty}$ converges almost surely. Notice that

$$\begin{aligned}
&\hat{f}^{r+1}(x^{r+1}) - \hat{f}^r(x^r) \\
&= \hat{f}^{r+1}(x^{r+1}) - \hat{f}^{r+1}(x^r) + \hat{f}^{r+1}(x^r) - \hat{f}^r(x^r) \\
&= \hat{f}^{r+1}(x^{r+1}) - \hat{f}^{r+1}(x^r) + \frac{1}{r+1}\sum_{i=1}^{r+1}\hat{g}(x^r, x^{i-1}, \xi^i) - \frac{1}{r}\sum_{i=1}^{r}\hat{g}(x^r, x^{i-1}, \xi^i) \\
&= \hat{f}^{r+1}(x^{r+1}) - \hat{f}^{r+1}(x^r) - \frac{1}{r(r+1)}\sum_{i=1}^{r}\hat{g}(x^r, x^{i-1}, \xi^i) + \frac{1}{r+1}\hat{g}(x^r, x^r, \xi^{r+1}) \\
&= \hat{f}^{r+1}(x^{r+1}) - \hat{f}^{r+1}(x^r) - \frac{\hat{f}^r(x^r)}{r+1} + \frac{1}{r+1}g(x^r, \xi^{r+1}) \\
&\leq \frac{-\hat{f}^r(x^r) + g(x^r, \xi^{r+1})}{r+1},
\end{aligned}$$

where the last equality is due to the assumption A1 and the inequality is due to the update rule of the SSUM algorithm. Taking the expectation with respect to the natural history yields

$$\begin{aligned}
\mathbb{E}\left[\hat{f}^{r+1}(x^{r+1}) - \hat{f}^r(x^r)\middle|\mathcal{F}^r\right] &\leq \mathbb{E}\left[\frac{-\hat{f}^r(x^r) + g(x^r, \xi^{r+1})}{r+1}\middle|\mathcal{F}^r\right] \\
&= \frac{-\hat{f}^r(x^r)}{r+1} + \frac{f(x^r)}{r+1} \\
&= \frac{-\hat{f}^r(x^r) + f^r(x^r)}{r+1} + \frac{f(x^r) - f^r(x^r)}{r+1} \qquad \text{(A.47)} \\
&\leq \frac{f(x^r) - f^r(x^r)}{r+1} \qquad \text{(A.48)} \\
&\leq \frac{\|f - f^r\|_\infty}{r+1}, \qquad \text{(A.49)}
\end{aligned}$$

where (A.48) is due to the assumption A2 and (A.49) follows from the definition of $\|\cdot\|_\infty$. On the other hand, the Donsker theorem (see [175, Lemma 7] and [238, Chapter 19]) implies that there exists a constant k such that

$$\mathbb{E}\left[\|f - f^r\|_\infty\right] \leq \frac{k}{\sqrt{r}}. \qquad \text{(A.50)}$$

Combining (A.49) and (A.50) yields

$$\mathbb{E}\left[\left(\mathbb{E}\left[\hat{f}^{r+1}(x^{r+1})-\hat{f}^{r}(x^{r})\middle|\mathcal{F}^{r}\right]\right)_{+}\right]\leq\frac{k}{r^{3/2}}, \tag{A.51}$$

where $(a)_+ \triangleq \max\{0,a\}$ is the projection to the non-negative orthant. Summing (A.51) over r, we obtain

$$\sum_{r=1}^{\infty}\mathbb{E}\left[\left(\mathbb{E}\left[\hat{f}^{r+1}(x^{r+1})-\hat{f}^{r}(x^{r})\middle|\mathcal{F}^{r}\right]\right)_{+}\right]\leq M<\infty, \tag{A.52}$$

where $M \triangleq \sum_{r=1}^{\infty}\frac{k}{r^{3/2}}$. The equation (A.52) combined with the quasi-martingale convergence theorem (see [237] and [175, Theorem 6]) implies that the stochastic process $\{\hat{f}^{r}(x^{r})+\bar{g}\}_{r=1}^{\infty}$ is a quasi-martingale with respect to the natural history $\{\mathcal{F}^{r}\}_{r=1}^{\infty}$ and $\hat{f}^{r}(x^{r})$ converges. Moreover, we have

$$\sum_{r=1}^{\infty}\left|\mathbb{E}\left[\hat{f}^{r+1}(x^{r+1})-\hat{f}^{r}(x^{r})\middle|\mathcal{F}^{r}\right]\right|<\infty,\quad\text{almost surely.} \tag{A.53}$$

Next we use (A.53) to show that $\sum_{r=1}^{\infty}\frac{\hat{f}^{r}(x^{r})-f^{r}(x^{r})}{r+1}<\infty$, almost surely. To this end, let us rewrite (A.47) as

$$\frac{\hat{f}^{r}(x^{r})-f^{r}(x^{r})}{r+1}\leq\mathbb{E}\left[-\hat{f}^{r+1}(x^{r+1})+\hat{f}^{r}(x^{r})\middle|\mathcal{F}^{r}\right]+\frac{f(x^{r})-f^{r}(x^{r})}{r+1}. \tag{A.54}$$

Using the fact that $\hat{f}^{r}(x^{r})\geq f^{r}(x^{r}),\ \forall\ r$ and summing (A.54) over all values of r, we have

$$\begin{aligned}0&\leq\sum_{r=1}^{\infty}\frac{\hat{f}^{r}(x^{r})-f^{r}(x^{r})}{r+1}\\&\leq\sum_{r=1}^{\infty}\left|\mathbb{E}\left[-\hat{f}^{r+1}(x^{r+1})+\hat{f}^{r}(x^{r})\middle|\mathcal{F}^{r}\right]\right|+\sum_{r=1}^{\infty}\frac{\|f-f^{r}\|_{\infty}}{r+1}.\end{aligned} \tag{A.55}$$

Notice that the first term in the right hand side is finite due to (A.53). Hence in order to show $\sum_{r=1}^{\infty}\frac{\hat{f}^{r}(x^{r})-f^{r}(x^{r})}{r+1}<\infty$, almost surely, it suffices to show that $\sum_{r=1}^{\infty}\frac{\|f-f^{r}\|_{\infty}}{r+1}<\infty$, almost surely. To show this, we use the Hewitt-Savage zero-one law; see [239,

Theorem 11.3] and [75, Chapter 12, Theorem 19]. Let us define the event

$$\mathcal{A} \triangleq \left\{ (\xi^1, \xi^2, \ldots) \mid \sum_{r=1}^{\infty} \frac{\|f^r - f\|_\infty}{r+1} < \infty \right\}.$$

It can be checked that the event $\mathcal{A}$ is permutable, i.e., any finite permutation of each element of $\mathcal{A}$ is inside $\mathcal{A}$; see [239, Theorem 11.3] and [75, Chapter 12, Theorem 19]. Therefore, due to the Hewitt-Savage zero-one law [239], probability of the event $\mathcal{A}$ is either zero or one. On the other hand, it follows from (A.50) that there exists $M' > 0$ such that

$$\mathbb{E}\left[\sum_{r=1}^{\infty} \frac{\|f^r - f\|_\infty}{r+1} \right] \leq M' < \infty. \tag{A.56}$$

Using Markov's inequality, (A.56) implies that

$$Pr\left(\sum_{r=1}^{\infty} \frac{\|f^r - f\|_\infty}{r+1} > 2M' \right) \leq \frac{1}{2}.$$

Hence combining this result with the result of the Hewitt-Savage zero-one law, we obtain $Pr(\mathcal{A}) = 1$; or equivalently

$$\sum_{r=1}^{\infty} \frac{\|f^r - f\|_\infty}{r+1} < \infty, \quad \text{almost surely.} \tag{A.57}$$

As a result of (A.55) and (A.57), we have

$$0 \leq \sum_{r=1}^{\infty} \frac{\hat{f}^r(x^r) - f^r(x^r)}{r+1} < \infty, \quad \text{almost surely.} \tag{A.58}$$

On the other hand, it follows from the triangle inequality that

$$\begin{aligned} &\left| \hat{f}^{r+1}(x^{r+1}) - f^{r+1}(x^{r+1}) - \hat{f}^r(x^r) + f^r(x^r) \right| \\ &\quad \leq \left| \hat{f}^{r+1}(x^{r+1}) - \hat{f}^r(x^r) \right| + \left| f^{r+1}(x^{r+1}) - f^r(x^r) \right| \end{aligned} \tag{A.59}$$

and

$$\begin{aligned}
&\left|\hat{f}^{r+1}(x^{r+1}) - \hat{f}^{r}(x^{r})\right| \\
&\leq \left|\hat{f}^{r+1}(x^{r+1}) - \hat{f}^{r+1}(x^{r})\right| + \left|\hat{f}^{r+1}(x^{r}) - \hat{f}^{r}(x^{r})\right| \\
&\leq \kappa\|x^{r+1} - x^{r}\| + \left|\frac{1}{r+1}\sum_{i=1}^{r+1}\hat{g}(x^{r}, x^{i-1}, \xi^{i}) - \frac{1}{r}\sum_{i=1}^{r}\hat{g}(x^{r}, x^{i-1}, \xi^{i})\right| \qquad \text{(A.60)} \\
&\leq \kappa\|x^{r+1} - x^{r}\| + \left|\frac{1}{r(r+1)}\sum_{i=1}^{r}\hat{g}(x^{r}, x^{i-1}, \xi^{i}) + \frac{\hat{g}(x^{r}, x^{r}, \xi^{r+1})}{r+1}\right| \\
&\leq \kappa\|x^{r+1} - x^{r}\| + \frac{2\bar{g}}{r+1} \qquad \text{(A.61)} \\
&= \mathcal{O}\left(\frac{1}{r}\right), \qquad \text{(A.62)}
\end{aligned}$$

where (A.60) is due to the assumption B3 (with $\kappa = (K + K')$); (A.61) follows from the assumption B6, and (A.62) will be shown in Lemma 9. Similarly, one can show that

$$|f^{r+1}(x^{r+1}) - f^{r}(x^{r})| = \mathcal{O}\left(\frac{1}{r}\right). \qquad \text{(A.63)}$$

It follows from (A.59), (A.62), and (A.63) that

$$\left|\hat{f}^{r+1}(x^{r+1}) - f^{r+1}(x^{r+1}) - \hat{f}^{r}(x^{r}) + f^{r}(x^{r})\right| = \mathcal{O}\left(\frac{1}{r}\right). \qquad \text{(A.64)}$$

Let us fix a random realization $\{\xi^{r}\}_{r=1}^{\infty}$ in the set of probability one for which (A.58) and (A.64) hold. Define

$$\alpha^{r} \triangleq \hat{f}^{r}(x^{r}) - f^{r}(x^{r}).$$

Clearly, $\alpha^{r} \geq 0$ and $\sum_{r} \frac{\alpha^{r}}{r} < \infty$ due to (A.58). Moreover, it follows from (A.64) that $|\alpha^{r+1} - \alpha^{r}| < \frac{\tau}{r}$ for some constant $\tau > 0$. Hence Lemma 10 implies that

$$\lim_{r\to\infty} \alpha^{r} = 0,$$

which is the desired result. ■

Lemma 9 $\|x^{r+1} - x^r\| = \mathcal{O}(\frac{1}{r})$.

Proof The proof of this lemma is similar to the proof of [175, Lemma 1]; see also [240, Proposition 4.32]. First of all, since x^r is the minimizer of $\hat{f}^r(\cdot)$, the first order optimality condition implies

$$\hat{f}^r(x^r; d) \geq 0, \quad \forall\, d \in \mathbb{R}^n.$$

Hence, it follows from the assumption A3 that

$$\hat{f}^r(x^{r+1}) - \hat{f}^r(x^r) \geq \frac{\gamma}{2}\|x^{r+1} - x^r\|^2. \tag{A.65}$$

On the other hand,

$$\begin{aligned}
\hat{f}^r(x^{r+1}) - \hat{f}^r(x^r) &\leq \hat{f}^r(x^{r+1}) - \hat{f}^{r+1}(x^{r+1}) + \hat{f}^{r+1}(x^r) - \hat{f}^r(x^r) && \text{(A.66)}\\
&\leq \frac{1}{r(r+1)} \sum_{i=1}^{r} |\hat{g}(x^{r+1}, x^{i-1}, \xi^i) - \hat{g}(x^r, x^{i-1}, \xi^i)| \\
&\quad + \frac{1}{r+1}|\hat{g}(x^{r+1}, x^r, \xi^{r+1}) - \hat{g}(x^r, x^r, \xi^{r+1})| \\
&\leq \frac{\theta}{r+1}\|x^{r+1} - x^r\|, && \text{(A.67)}
\end{aligned}$$

where (A.66) follows from the fact that x^{r+1} is the minimizer of $\hat{f}^{r+1}(\cdot)$, the second inequality is due to the definitions of $\hat{f}^r$ and $\hat{f}^{r+1}$, while (A.67) is the result of the assumptions B3 and B5. Combining (A.65) and (A.67) yields the desired result.

Lemma 10 *Assume* $\alpha^r > 0$ *and* $\sum_{r=1}^{\infty} \frac{\alpha^r}{r} < \infty$. *Furthermore, suppose that* $|\alpha^{r+1} - \alpha^r| \leq \tau/r$ *for all* r. *Then* $\lim_{r\to\infty} \alpha^r = \infty$.

Proof Since $\sum_{r=1}^{\infty} \frac{\alpha^r}{r} < \infty$, we have $\liminf_{r\to\infty} \alpha^r = 0$. Now, we prove the result using contradiction. Assume the contrary so that

$$\limsup_{r\to\infty} \alpha^r > \epsilon, \tag{A.68}$$

for some $\epsilon > 0$. Hence there should exist subsequences $\{m_j\}$ and $\{n_j\}$ with $m_j \leq n_j <$

$m_{j+1}, \forall\ j$ so that

$$\frac{\epsilon}{3} < \alpha^r \qquad m_j \le r < n_j, \tag{A.69}$$

$$\alpha^r \le \frac{\epsilon}{3} \qquad n_j \le r < m_{j+1}. \tag{A.70}$$

On the other hand, since $\sum_{r=1}^{\infty} \frac{\alpha^r}{r} < \infty$, there exists an index $\bar{r}$ such that

$$\sum_{r=\bar{r}}^{\infty} \frac{\alpha^r}{r} < \frac{\epsilon^2}{9\tau}. \tag{A.71}$$

Therefore, for every $r_0 \ge \bar{r}$ with $m_j \le r_0 \le n_j - 1$, we have

$$\begin{aligned} |\alpha^{n_j} - \alpha^{r_0}| &\le \sum_{r=r_0}^{n_j-1} |\alpha^{r+1} - \alpha^r| \\ &\le \sum_{r=r_0}^{n_j-1} \frac{\tau}{r} \qquad (A.72) \\ &\le \frac{3}{\epsilon} \sum_{r=r_0}^{n_j-1} \frac{\tau}{r} \alpha^r \qquad (A.73) \\ &\le \frac{3\tau\epsilon^2}{9\epsilon\tau} = \frac{\epsilon}{3}, \qquad (A.74) \end{aligned}$$

where the equation (A.73) follows from (A.69), and (A.74) is the direct consequence of (A.71). Hence the triangle inequality implies

$$\alpha^{r_0} \le \alpha^{n_j} + |\alpha^{n_j} - \alpha^{r_0}| \le \frac{\epsilon}{3} + \frac{\epsilon}{3} = \frac{2\epsilon}{3},$$

for any $r_0 \ge \bar{r}$, which contradicts (A.68), implying that

$$\limsup_{r \to \infty} \alpha^r = 0.$$

■

Proof of Theorem 13:

Consider a limit point $\bar{x}$ with the subsequence $\{x^{r_j}\}$ converging to $\bar{x}$. First of all, it

is not hard to see that

$$\theta_{i_r}(x^{r+1}) \leq \hat{\theta}_{i_r}(x_{i_r}^{r+1}, x^r) \leq \hat{\theta}_{i_r}(x_{i_r}^r, x^r) = \theta_{i_r}(x^r),\ \forall r, \tag{A.75}$$

where the first inequality and the last equality is due to the properties of the approximation function $\hat{\theta}(\cdot)$ in Assumption 5; and the second inequality is due to the update rule of the algorithm. Due to strict convexity of the function $\hat{\theta}(\cdot, x^r)$, the above inequality implies that either $\theta_{i_r}(x^{r+1}) < \theta_{i_r}(x^r)$, or $x^{r+1} = x^r$. Clearly in both cases we have

$$P(x^{r+1}) \leq P(x^r),\ \forall r, \tag{A.76}$$

and therefore

$$\lim_{r\to\infty} P(x^r) = P(\bar{x}),$$

due to continuity of the potential function $P(\cdot)$. On the other hand, since the essentially cyclic update rule is chosen, by restricting to a subsequence, we can assume that there exists $(\alpha_1, \ldots, \alpha_T)$ such that

$$(i_{r_j}, i_{r_j}+1, \ldots, i_{r_j}+T-1) = (\alpha_1, \alpha_2, \ldots, \alpha_T),\ \forall j$$

with $\alpha_t \in \{1, \ldots, n\},\ \forall t = 1, \ldots, T$ and $\{\alpha_1, \alpha_2, \ldots, \alpha_T\} = \{1, 2, \ldots, n\}$. Next, we will show that

$$\lim_{j\to\infty} \theta_{\alpha_1}(x^{r_j+1}) = \theta_{\alpha_1}(\bar{x}), \tag{A.77}$$

by using contradiction argument. First, let us rewrite (A.75) for the subsequence of interest

$$\theta_{\alpha_1}(x^{r_j+1}) \leq \hat{\theta}_{\alpha_1}(x_{\alpha_1}^{r_j+1}, x^{r_j}) \leq \hat{\theta}_{\alpha_1}(x_{\alpha_1}^{r_j}, x^{r_j}) = \theta_{\alpha_1}(x^{r_j}).$$

Thus, $\limsup_{j\to\infty} \theta_{\alpha_1}(x^{r_j+1}) \leq \theta_{\alpha_1}(\bar{x})$. Combining this fact with the contrary of (A.77) implies

$$\theta_{\alpha_1}(x^{r_j}+1) \leq \theta_{\alpha_1}(\bar{x}) - \beta, \tag{A.78}$$

for some $\beta > 0$ and for all j large enough. Therefore, for large enough indices j, we

have

$$P(x^{r_j}+1) \leq P(\bar{x}) - \sigma(\theta_{\alpha_1}(\bar{x}) - \theta_{\alpha_1}(x^{r_j}+1)). \tag{A.79}$$

Clearly, $\liminf_{j\to\infty} \sigma(\theta_{\alpha_1}(\bar{x}) - \theta_{\alpha_1}(x^{r_j}+1)) > 0$ due to (A.78). Therefore, by letting $j \to \infty$ in (A.79), we have

$$P(\bar{x}) < P(\bar{x}),$$

which is a contradiction and therefore the contrary assumption does not hold and (A.77) must hold true. Next, we show that

$$\lim_{j\to\infty} x^{r_j+1} = \bar{x}. \tag{A.80}$$

Assume the contrary. Hence by restricting to a subsequence, there exists $\bar{\gamma} > 0$ such that

$$\|x_{\alpha_1}^{r_j+1} - x_{\alpha_1}^{r_j}\| \triangleq \gamma^{r_j} \geq \bar{\gamma},\ \forall j.$$

Define $S^{r_j} \triangleq \frac{x_{\alpha_1}^{r_j+1} - x_{\alpha_1}^{r_j}}{\gamma^{r_j}}$. One can write,

$$\begin{aligned}
\theta_{\alpha_1}(x^{r_j+1}) &\leq \hat{\theta}_{\alpha_1}(x_{\alpha_1}^{r_j+1}, x^{r_j}) && \text{(A.81)}\\
&= \hat{\theta}_{\alpha_1}(x_{\alpha_1}^{r_j} + \gamma^{r_j} S^{r_j}, x^{r_j}) && \text{(A.82)}\\
&\leq \hat{\theta}_{\alpha_1}(x_{\alpha_1}^{r_j} + \epsilon\bar{\gamma} S^{r_j}, x^{r_j}),\ \forall \epsilon \in [0,1] && \text{(A.83)}\\
&\leq \hat{\theta}_{\alpha_1}(x_{\alpha_1}^{r_j}, x^{r_j}) && \text{(A.84)}\\
&= \theta_{\alpha_1}(x^{r_j}), && \text{(A.85)}
\end{aligned}$$

where (A.81) and (A.85) are due to the properties of the approximation function; in the equations (A.82), (A.83), and (A.84), we use the update rule of the algorithm and the convexity of the function $\hat{\theta}_{\alpha_1}(\cdot, x^r)$. Since S^{r_j} is in a compact ball, it has a limit point $\bar{S}$. Hence by restricting to a subsequence, letting $j \to \infty$, and using (A.77), we can rewrite the above inequality as

$$\theta_{\alpha_1}(\bar{x}) \leq \hat{\theta}_{\alpha_1}(\bar{x}_{\alpha_1} + \epsilon\bar{\gamma}\bar{S}, \bar{x}) \leq \theta_{\alpha_1}(\bar{x}),\ \forall \epsilon \in [0,1],$$

which contradicts the strict convexity of $\hat{\theta}(\cdot, \bar{x})$. Therefore, the contrary assumption is

not true and (A.80) holds true.
On the other hand, due to the update rule of the algorithm, we have

$$\hat{\theta}_{\alpha_1}(x_{\alpha_1}^{r_j+1}, x^{r_j}) \leq \hat{\theta}(x_{\alpha_1}, x^{r_j}),\ \forall x_{\alpha_1} \in \mathcal{X}_{\alpha_1}.$$

Letting $j \to \infty$, we get

$$\hat{\theta}_{\alpha_1}(\bar{x}_{\alpha_1}, \bar{x}) \leq \hat{\theta}(x_{\alpha_1}, \bar{x}),\ \forall x_{\alpha_1} \in \mathcal{X}_{\alpha_1}.$$

The first order optimality condition implies

$$\hat{\theta}'_{\alpha_1}(\bar{x}_{\alpha_1}, \bar{x}; d_{\alpha_1}) \geq 0,\ \forall d_{\alpha_1} \text{ with } \bar{x}_{\alpha_1} + d_{\alpha_1} \in \mathcal{X}_{\alpha_1}.$$

Using the directional derivative property of the approximation function, we have

$$\theta'_{\alpha_1}(\bar{x}; d) \geq 0,\ \forall d = (0, \ldots, 0, d_{\alpha_1}, 0, \ldots, 0) \text{ with } \bar{x}_{\alpha_1} + d_{\alpha_1} \in \mathcal{X}_{\alpha_1}.$$

Repeating the above argument for other players $\alpha_2, \ldots, \alpha_T$ will complete the proof. ■

Proof of Theorem 14: First of all, similar to (A.76), we can show the decrease of the potential function at each iteration and therefore the objective value converges for any realization of the random choices. In other words, for any realization, we must have

$$\lim_{r \to \infty} P(x^r) - P(x^{r+1}) = 0. \tag{A.86}$$

Let $\hat{x}_i^r$ denote one of the possible optimal points at iteration r if block i is chosen, i.e.,

$$\hat{x}_i^r \in \arg\min_{x_i \in \mathcal{X}_i} \hat{\theta}_i(x_i, x^r).$$

Then similar to (A.75), we have

$$\theta_i(\hat{x}_i^r, x_{-i}^r) \leq \hat{\theta}_i(\hat{x}_i^r, x^r) \leq \hat{\theta}_i(x_i^r, x^r) = \theta_i(x^r),\ \forall i = 1, \ldots, n,\ \forall r. \tag{A.87}$$

Therefore, due to the existence of the generalized potential function, we have

$$P(x^r) - P(\hat{x}_i^r, x_{-i}^r) \geq \sigma\left(\theta_i(x^r) - \theta_i(\hat{x}_i^r, x_{-i}^r)\right), \ \forall i = 1, \ldots, n, \ \forall r,$$

which combined with the randomized choice of players implies

$$\begin{aligned}\mathbb{E}\left[P(x^r) - P(x^{r+1}) \mid x^r\right] &\geq \sum_{i=1}^{n} p_i \sigma\left(\theta_i(x^r) - \theta_i(\hat{x}_i^r, x_{-i}^r)\right) \\ &\geq p_{\min} \sum_{i=1}^{n} \sigma\left(\theta_i(x^r) - \theta_i(\hat{x}_i^r, x_{-i}^r)\right),\end{aligned}$$

where $p_{\min} \triangleq \min_i p_i$. By re-arranging the terms, we can write

$$\mathbb{E}\left[P(x^{r+1}) \mid x^r\right] \leq P(x^r) - p_{\min} \sum_{i=1}^{n} \sigma\left(\theta_i(x^r) - \theta_i(\hat{x}_i^r, x_{-i}^r)\right).$$

Clearly the process $\{P(x^r)\}_{r=1}^{\infty}$ is a supermartingale and by the supermartingale convergence theorem [236, Proposition 4.2], we have

$$p_{\min} \sum_{r=1}^{\infty} \sum_{i=1}^{n} \sigma\left(\theta_i(x^r) - \theta_i(\hat{x}_i^r, x_{-i}^r)\right) < \infty,$$

with probability one, which in turn implies that

$$\lim_{r\to\infty} \sigma\left(\kappa_i(x^r)\right) = 0, \ \text{ almost surely, } \forall i;$$

and since σ is a forcing function, we have

$$\lim_{r\to\infty} \kappa_i(x^r) = 0, \ \text{ almost surely, } \forall i,$$

which completes the proof. ■

Proof of Theorem 15: Let us define $\hat{x}_i(y) = \arg\min_{x_i} \hat{\theta}(x_i, y)$. Consider the two

points $y, w \in \mathcal{X}$. Due to the first order optimality condition of $\hat{x}_i(y), \hat{x}_i(w)$, we have

$$0 \leq \langle z_i - \hat{x}_i(y), \nabla_{x_i} \hat{f}_i \left(\hat{x}_i(y), y\right) + \vartheta_y \rangle, \ \forall z_i \in \mathcal{X}_i, \tag{A.88}$$

$$0 \leq \langle z_i - \hat{x}_i(w), \nabla_{x_i} \hat{f}_i \left(\hat{x}_i(w), w\right) + \vartheta_w \rangle, \ \forall z_i \in \mathcal{X}_i, \tag{A.89}$$

for some $\vartheta_y \in \partial g_i(\hat{x}_i(y))$ and $\vartheta_w \in \partial g_i(\hat{x}_i(w))$. Plugging $\hat{x}_i(w)$, $\hat{x}_i(w)$ in z_i and summing up the above equations, we obtain

$$\langle \hat{x}_i(w) - \hat{x}_i(y), \nabla_{x_i} \hat{f}_i \left(\hat{x}_i(y), y\right) - \nabla_{x_i} \hat{f} \left(\hat{x}_i(w), w\right) + \vartheta_y - \vartheta_w \rangle \geq 0. \tag{A.90}$$

On the other hand, due to the definition of the subgradients ϑ_y and ϑ_w, we have

$$g_i(\hat{x}_i(w)) \geq g_i(\hat{x}_i(y)) + \langle \vartheta_y, \hat{x}_i(w) - \hat{x}_i(y) \rangle \tag{A.91}$$

$$g_i(\hat{x}_i(y)) \geq g_i(\hat{x}_i(w)) + \langle \vartheta_w, \hat{x}_i(y) - \hat{x}_i(w) \rangle \tag{A.92}$$

Summing up (A.90), (A.92), and (A.91) implies

$$\langle \hat{x}_i(w) - \hat{x}_i(y), \nabla_{x_i} \hat{f} \left(\hat{x}_i(y), y\right) - \nabla_{x_i} \hat{f} \left(\hat{x}_i(w), w\right) \rangle \geq 0. \tag{A.93}$$

Applying the mean value theorem to the one dimensional function $\varpi(t) = \langle \hat{x}_i(w) - \hat{x}_i(y), \nabla_{x_i} \hat{f} \left(t\hat{x}_i(y) + (1-t)\hat{x}_i(w), ty + (1-t)w\right) \rangle$ on the interval $[0, 1]$, we can write

$$\begin{aligned} \varpi(1) - \varpi(0) = \Bigg\langle \hat{x}_i(w) - \hat{x}_i(y) \ , \ \nabla^2_{x_i x_i} \hat{f}(x_i, y)\Big|_{x_i = v_i, y = z} (\hat{x}_i(y) - \hat{x}_i(w)) \\ + \sum_{j=1}^{n} \nabla^2_{y_j x_i} \hat{f}(x_i, y)\Big|_{x_i = v_i, y = z} (y_j - w_j) \Bigg\rangle, \end{aligned} \tag{A.94}$$

for some v_i in the line segment $[\hat{x}_i(w), \hat{x}_i(y)]$; and for some z in the line segment $[y, w]$. Plugging (A.94) in (A.93) and using the fact that $\nabla^2_{x_i x_i} \hat{f}(x_i, y) \geq \tau_i \mathbf{I}$, $\forall x_i, y$, we obtain

$$\left\langle \hat{x}_i(w) - \hat{x}_i(y) \ , \ \sum_{j=1}^{n} \nabla^2_{y_j x_i} \hat{f}(x_i, y)\Big|_{x_i = v_i, y = z} (y_j - w_j) \right\rangle \geq \tau_i \|\hat{x}_i(w) - \hat{x}_i(y)\|^2.$$

Expanding the left hand side of the inequality combined with the Cauchy-Schwarz inequality implies

$$\sum_{j=1}^{n} \gamma_{ij} \|y_j - w_j\| \geq \tau_i \|\hat{x}_i(w) - \hat{x}_i(y)\|. \tag{A.95}$$

Writing in a matrix form, we obtain

$$\begin{bmatrix} \|\hat{x}_1(y) - \hat{x}_1(w)\| \\ \|\hat{x}_2(y) - \hat{x}_2(w)\| \\ \vdots \\ \|\hat{x}_n(y) - \hat{x}_n(w)\| \end{bmatrix} \leq \begin{bmatrix} \frac{\gamma_{11}}{\tau_1} & \frac{\gamma_{12}}{\tau_1} & \dots & \frac{\gamma_{1n}}{\tau_1} \\ \frac{\gamma_{21}}{\tau_2} & \frac{\gamma_{22}}{\tau_2} & \dots & \frac{\gamma_{2n}}{\tau_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\gamma_{n1}}{\tau_n} & \frac{\gamma_{n2}}{\tau_n} & \dots & \frac{\gamma_{nn}}{\tau_n} \end{bmatrix} \begin{bmatrix} \|y_1 - w_1\| \\ \|y_2 - w_2\| \\ \vdots \\ \|y_n - w_n\| \end{bmatrix} \tag{A.96}$$

Clearly, when $\|\Gamma\|_2 < 1$, we have a contraction mapping and the iterates converge linearly. ■

Proof of Lemma 7: First of all, it can be observed that choosing $\mathbf{Q}_1 = \mathbf{Q}_2 = \mathbf{Q}_3 = \mathbf{Q}_a^*$ yields an objective value of $\lambda^* = 1$; the same result holds for the case of $\mathbf{Q}_1 = \mathbf{Q}_2 = \mathbf{Q}_3 = \mathbf{Q}_b^*$, $\mathbf{Q}_1 = \mathbf{Q}_2 = \mathbf{Q}_3 = \mathbf{Q}_c^*$, and $\mathbf{Q}_1 = \mathbf{Q}_2 = \mathbf{Q}_3 = \mathbf{Q}_d^*$.

Let $(\lambda, \mathbf{Q}_1, \mathbf{Q}_2, \mathbf{Q}_3) \in \mathcal{S}$ be an optimal solution. Clearly, at least one of the users must transmit with full power, for otherwise we could simultaneously scale $(\mathbf{Q}_1, \mathbf{Q}_2, \mathbf{Q}_3)$ to get a better objective function. Without loss of generality, let us assume that user 1 is transmitting with full power, i.e., $\mathrm{Tr}(\mathbf{Q}_1) = 1$. Using eigenvalue decomposition of $\mathbf{Q}_1$, we can write $\mathbf{Q}_1 = \alpha \mathbf{a}\mathbf{a}^H + \beta \mathbf{b}\mathbf{b}^H$, where $\mathbf{a}$ and $\mathbf{b}$ are the orthonormal eigenvectors of $\mathbf{Q}_1$ and the scalars $\alpha, \beta \geq 0$ are the eigenvalues of $\mathbf{Q}_1$ with $\alpha + \beta = 1$. Since canceling the interference results in higher rate of communication, we have

$$\begin{aligned} R_2 &= \log\det\left(\mathbf{I} + \mathbf{Q}_2\left(\mathbf{I} + \sum_{m\neq 2} \mathbf{H}_{2m}\mathbf{Q}_m\mathbf{H}_{2m}^H\right)^{-1}\right) \\ &\leq \log\det\left(\mathbf{I} + \mathbf{Q}_2\left(\mathbf{I} + \mathbf{H}_{21}(\alpha\mathbf{a}\mathbf{a}^H + \beta\mathbf{b}\mathbf{b}^H)\mathbf{H}_{21}^H\right)^{-1}\right) \\ &= \log\det\left(\mathbf{I} + \mathbf{Q}_2\left(\mathbf{I} + 4\alpha\ \underline{\mathbf{a}}\ \underline{\mathbf{a}}^H + 4\beta\ \underline{\mathbf{b}}\ \underline{\mathbf{b}}^H\right)^{-1}\right) \\ &= \log\det\left(\mathbf{I} + \mathbf{Q}_2\left(\frac{1}{1+4\alpha}\ \underline{\mathbf{a}}\ \underline{\mathbf{a}}^H + \frac{1}{1+4\beta}\ \underline{\mathbf{b}}\ \underline{\mathbf{b}}^H\right)\right) \\ &\leq \log\det\left(\mathbf{I} + \frac{1}{\mathrm{Tr}(\mathbf{Q}_2)}\mathbf{Q}_2\left(\frac{1}{1+4\alpha}\ \underline{\mathbf{a}}\ \underline{\mathbf{a}}^H + \frac{1}{1+4\beta}\ \underline{\mathbf{b}}\ \underline{\mathbf{b}}^H\right)\right), \end{aligned} \tag{A.97}$$

where $\underline{\mathbf{a}} = \frac{1}{2}\mathbf{H}_{21}\mathbf{a}$ and $\underline{\mathbf{b}} = \frac{1}{2}\mathbf{H}_{21}\mathbf{b}$. The last inequality is due to the fact that $\mathrm{Tr}(\mathbf{Q}_2) \leq 1$. Clearly, $\underline{\mathbf{a}}^H\underline{\mathbf{b}} = 0$ and $\|\underline{\mathbf{a}}\| = \|\underline{\mathbf{b}}\| = 1$.

Let us use the eigenvalue decomposition $\frac{\mathbf{Q}_2}{\mathrm{Tr}(\mathbf{Q}_2)} = \theta\mathbf{c}\mathbf{c}^H + (1-\theta)\mathbf{d}\mathbf{d}^H$, for some $\theta \in [0,1]$ and some orthonormal vectors $\mathbf{c}$ and $\mathbf{d}$. Utilizing the fact that determinant is the product of the eigenvalues and trace is the sum of the eigenvalues, we can further simplify the inequality in (A.97) as

$$\begin{aligned}
R_2 &\leq \log\Bigg\{1 + \mathrm{Tr}\left[\left(\theta\mathbf{c}\mathbf{c}^H + (1-\theta)\mathbf{d}\mathbf{d}^H\right)\left(\frac{1}{1+4\alpha}\underline{\mathbf{a}}\,\underline{\mathbf{a}}^H + \frac{1}{1+4\beta}\underline{\mathbf{b}}\,\underline{\mathbf{b}}^H\right)\right] \\
&\qquad + \det\left[\left(\theta\mathbf{c}\mathbf{c}^H + (1-\theta)\mathbf{d}\mathbf{d}^H\right)\left(\frac{1}{1+4\alpha}\underline{\mathbf{a}}\,\underline{\mathbf{a}}^H + \frac{1}{1+4\beta}\underline{\mathbf{b}}\,\underline{\mathbf{b}}^H\right)\right]\Bigg\} \\
&= \log\left[1 + \frac{\theta x}{1+4\alpha} + \frac{\theta(1-x)}{1+4\beta} + \frac{(1-\theta)(1-x)}{1+4\alpha} + \frac{(1-\theta)x}{1+4\beta} + \frac{\theta(1-\theta)}{(1+4\alpha)(1+4\beta)}\right] \\
&\leq \max_{(x,\theta,\alpha,\beta)\in\mathcal{Y}} \log\left[1 + \frac{\theta x}{1+4\alpha} + \frac{\theta(1-x)}{1+4\beta} + \frac{(1-\theta)(1-x)}{1+4\alpha} + \frac{(1-\theta)x}{1+4\beta} + \frac{\theta(1-\theta)}{(1+4\alpha)(1+4\beta)}\right],
\end{aligned} \tag{A.98}$$

where $x \triangleq |\mathbf{c}^H\mathbf{a}|^2$, $\mathcal{Y} \triangleq \{(x,\theta,\alpha,\beta) \mid \alpha+\beta = 1,\ 0 \leq \alpha,\beta,x \leq 1\}$. Since the function in (A.98) is linear in x, it suffices to only check the boundary points $x = 0$ and $x = 1$ in order to find the maximum. The claim is that the maximum in (A.98) takes the value of 1, and it is achieved at both boundary points.

First consider the boundary point $x = 1$. We have

$$R_2 \leq \max_{(\theta,\alpha,\beta)\in\mathcal{X}} \quad f(\theta,\alpha,\beta), \tag{A.99}$$

where $\mathcal{X} \triangleq \{(\theta,\alpha,\beta) \mid \alpha+\beta = 1,\ 0 \leq \alpha,\beta\}$ and

$$f(\theta,\alpha,\beta) \triangleq \log\left(1 + \frac{\theta}{1+4\alpha} + \frac{1-\theta}{1+4\beta} + \frac{\theta(1-\theta)}{(1+4\alpha)(1+4\beta)}\right) \tag{A.100}$$

We are interested in finding the set of optimal solutions of (A.100). In particular, we want to characterize $\mathcal{S}_1 = \{(\theta^*,\alpha^*,\beta^*)\}$ defined by $\mathcal{S}_1 \triangleq \arg\max_{(\theta,\alpha,\beta)\in\mathcal{X}} \ f(\theta,\alpha,\beta)$.

In what follows, we will prove that $\mathcal{S}_1 = \{(0,1,0),(1,0,1)\}$. First we observe that $f(0,1,0) = f(1,0,1) = 1$. Now, we show that $f(\theta,\alpha,\beta) < 1$, for all $(\theta,\alpha,\beta) \in \mathcal{X}$ such

that $0 < \theta < 1$. Assume the contrary that there exists an optimal point $(\theta^*, \alpha^*, \beta^*)$ such that $0 < \theta^* < 1$. Using the first order optimality condition $\frac{\partial}{\partial \theta} f(\theta^*, \alpha^*, \beta^*) = 0$, we obtain $\theta^* = \frac{4\beta^* - 4\alpha^* + 1}{2}$. Combining with $0 < \theta^* < 1$ yields

$$-\frac{1}{4} < \beta^* - \alpha^* < \frac{1}{4}. \tag{A.101}$$

Plugging in the value of optimal $\theta^* = \frac{4\beta^* - 4\alpha^* + 1}{2}$ in $f(\cdot)$ and simplifying the equations, we obtain

$$f(\theta^*, \alpha^*, \beta^*) = \log\left(1 + \frac{13 + 16(\beta^* - \alpha^*)^2}{4(1 + 4\alpha^*)(1 + 4\beta^*)}\right).$$

Combining with (A.101) yields

$$\begin{aligned} f(\theta^*, \alpha^*, \beta^*) &\leq \log\left(1 + \frac{14}{4(1 + 4\alpha^*)(1 + 4\beta^*)}\right) \\ &\leq \log\left(1 + \frac{14}{4(1 + 4\alpha^* + 4\beta^*)}\right) \\ &= \log\left(1 + \frac{14}{20}\right) < 1, \end{aligned}$$

which contradicts the fact that $\max_{(\theta,\alpha,\beta)\in\mathcal{X}} f(\theta, \alpha, \beta) = 1$. Therefore, the optimal θ only happens at the boundary and we have $\{(0, 1, 0), (1, 0, 1)\} = \arg\max_{(\theta,\alpha,\beta)\in\mathcal{X}} f(\theta, \alpha, \beta)$. Similarly, for the case when $x = 0$, we can see that the optimal solution set is $\{(0, 0, 1), (1, 1, 0)\}$. Using these optimal values yields $R_2 \leq 1$. Note that in order to have equality $R_2 = 1$, we must have $\mathrm{Tr}(\mathbf{Q}_2) = 1$ and

$$(x, \theta, \alpha, \beta) \in \{(1, 0, 1, 0), (1, 1, 0, 1), (0, 0, 0, 1), (0, 1, 1, 0)\}.$$

Let us choose the optimal solution $(x, \theta, \alpha, \beta) = (1, 0, 1, 0)$. Therefore,

$$\mathbf{Q}_1 = \mathbf{a}\mathbf{a}^H, \quad \mathbf{Q}_2 = \mathbf{d}\mathbf{d}^H, \quad x = |\mathbf{c}^H\mathbf{a}|^2 = 1,$$

which yields $\underline{\mathbf{a}}^H\mathbf{d} = 0$. Repeating the above argument for user 2 and user 3, we get $\mathbf{Q}_3 = \mathbf{g}\mathbf{g}^H$ with $\underline{\mathbf{a}}^H\mathbf{g} = 0$. Since $\mathbf{d}$ and $\mathbf{g}$ are both orthogonal to $\underline{\mathbf{a}}$, we obtain $\mathbf{d} = \exp^{j\phi_d}\mathbf{g}$. Repeating the above argument for the other pair of users yields

$$\mathbf{a} = \exp^{j\phi_a}\mathbf{g} \quad \text{and} \quad \underline{\mathbf{a}}^H\mathbf{a} = 0,$$

where the last relations imply that $\mathbf{a}, \mathbf{d}$, and $\mathbf{g}$ are the same up to the phase rotation and they belong to the following set (after the proper phase rotation)

$$\mathbf{a} \in \left\{ [1 \quad 0]^H, [0 \quad 1]^H, \frac{1}{\sqrt{2}}[j \quad 1]^H, \frac{1}{\sqrt{2}}[1 \quad j]^H \right\}.$$

Each of these points gives us one of the optimal covariance matrices in (3.33). ■

Proof of Theorem 19: The proof is based on a polynomial time reduction from the 3-satisfiability (3-SAT) problem which is known to be NP-complete. We first consider an instance of the 3-SAT problem with n variables $x_1, x_2, \ldots, x_n$ and m clauses $c_1, c_2, \ldots, c_m$. For each variable x_i, we consider 5 users $\mathcal{X}_{1i}, \mathcal{X}_{2i}, \ldots, \mathcal{X}_{5i}$ in our interference channel. Each user is equipped with two antennas, and the channels between the users are specified as in (3.34)–(3.36). For each clause c_j, $j = 1, 2, \ldots, m$, we consider one user $\mathcal{C}_j$ in the system with two antennas. In summary, we totally have $5n + m$ users in the system. Set the noise power $\sigma_k^2 = 1$, $\forall k$, and the power budget $P_k = 1$ for all users. We define the channel between the users $\mathcal{C}_i$ and $\mathcal{C}_j$ to be zero for all $j \neq i$. Furthermore, we assume that the channel between the transmitter and receiver of user $\mathcal{C}_i$ is given by

$$\mathbf{H}_{\mathcal{C}_i \mathcal{C}_i} = \sqrt{3} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

Let us also assume that *i)* there is no interference among the blocks of users that correspond to different variables and *ii)* there is no interference from the transmitter of user $\mathcal{C}_j$ to the receivers of users $\mathcal{X}_{1i}, \ldots, \mathcal{X}_{5i}$ for all $i = 1, 2, \ldots, n$; $j = 1, 2, \ldots, m$. Consider a clause $c_j : y_{j1} + y_{j2} + y_{j3}$, where $y_{j1}, y_{j2}, y_{j3} \in \{x_1, x_2, \ldots, x_n, \overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n\}$ with $\overline{x}_i$ denoting the negation of x_i. We use the following rules to define the channels from the transmitter of user $\mathcal{X}_{ki}$ to the receiver of user $\mathcal{C}_j$:

- If the variable x_i appears in c_j, we define the channel from the transmitter of $\mathcal{X}_{1i}$ to the receiver of $\mathcal{C}_j$ to be $\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$.
- If the variable $\bar{x}_i$ appears in c_j, we define the channel from the transmitter of $\mathcal{X}_{1i}$ to the receiver of $\mathcal{C}_j$ to be $\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$.

- If x_i does not appear in c_j, we define the channel from the transmitter of $\mathcal{X}_{1i}$ to the receiver of $\mathcal{C}_j$ to be zero.
- The channel from transmitters of users $\mathcal{X}_{2i}, \mathcal{X}_{3i}, \mathcal{X}_{4i}, \mathcal{X}_{5i}$ to the receiver of user $\mathcal{C}_j$ is zero for all $i = 1, \ldots, n$ and $j = 1, 2, \ldots, m$.

As an example, Figure A.1 shows the channels for the clause $c_\ell : x_i + \bar{x}_j + x_k$.

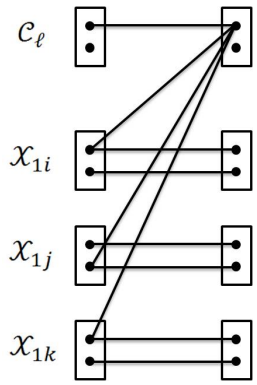


Figure A.1: Channels for the clause $c_\ell : x_i + \bar{x}_j + x_k$.

Now we claim that the 3-SAT problem is satisfiable if and only if solving the problem (3.38) for the corresponding interference channel leads to the optimum value of one. To prove this fact, let us assume that the optimum value of (3.38) is one. According to the Lemma 8, the only way to get the rate of one for users $\mathcal{X}_{kj}$, $k = 1, \ldots, 5$, $j = 1, \ldots, n$, is to transmit with full power either on the first antenna or on the second antenna. Now, based on the optimal solution of (3.38), we can determine the solution of the 3-SAT problem. In particular, if user $\mathcal{X}_{1i}$ is transmitting on the first antenna, we set $x_i = 0$. Otherwise, if it transmits on the second antenna, we set $x_i = 1$. By assigning values to all the variables in this way, we claim that all clauses are satisfied. We prove by contradiction. Assume the contrary that there exists a clause c_j that is not satisfied, i.e., all the corresponding variables are zero. Therefore, user $\mathcal{C}_j$ gets interference on the first receive antenna from all three users corresponding to the variables appearing in $\mathcal{C}_j$. As the result, the interference power is 3. Since the noise power is one and the received signal power is 3, the SINR level for user $\mathcal{C}_j$ is $\frac{3}{1+3}$ which contradicts the fact that the

minimum rate in the system is one.

Now we prove the other direction. Let us assume that the 3-SAT problem is satisfiable. We claim that the optimal value of (3.38) is one. Since in each block of 5 users the optimum value is one, it suffices to show that the objective value of one is achievable. Now, we design the covariance matrices based on the solution of the 3-SAT problem. If $x_i = 0$, we transmit with full power on the first antenna of users $\mathcal{X}_{1i}, \mathcal{X}_{2i}, \ldots, \mathcal{X}_{5i}$. If $x_i = 1$, we allocate full power for transmission on the second antenna of users $\mathcal{X}_{1i}, \mathcal{X}_{2i}, \ldots, \mathcal{X}_{5i}$. With this allocation, each user $\mathcal{X}_{ki}$, $k = 1, \ldots, 5$, $i = 1, \ldots, n$, gets the rate of one. For all users $\mathcal{C}_j$, $j = 1, 2, \ldots, m$, we transmit with full power on the first antenna. Since 3-SAT problem is satisfiable with the given boolean allocation of the variables, for each clause $\mathcal{C}_j$ at least one of the corresponding variables is one. Therefore, the interference level at the receiver of user $\mathcal{C}_j$ is at most 2. Since the received signal power at the receiver of user $\mathcal{C}_j$ is 3, the SINR level is at least $\frac{3}{1+2} = 1$ which yields the rate of communication $R_{\mathcal{C}_j} \geq 1$. Thus, all users $\mathcal{C}_j, j = 1, \ldots, m$, have rate at least one; which completes the proof of our claim. As the result, checking whether the objective value of one is achievable for (3.38) is equivalent to solving the instance of 3-SAT problem. Thus, problem (3.38) is NP-hard. ■

Proof of Theorem 20: The proof is based on the polynomial time reduction of the densest cut problem. The densest cut problem can be stated as follows:

Densest Cut Problem: Given a graph $\mathcal{G} = (V, E)$, the goal is to maximize the ratio $\frac{|E(P,Q)|}{|P| \cdot |Q|}$ over all the bipartitions (P, Q) of the vertices of the graph $\mathcal{G}$. Here $E(P, Q)$ denotes the set of edges between the two partitions and the operator $|\cdot|$ returns the cardinality of a set.

Given an undirected graph $\mathcal{G}$, we put an arbitrary directions on it and we define $\mathbf{Y}'$ to be the incidence transpose matrix of the directed graph. In other words, $\mathbf{Y}' \in \mathbb{R}^{|E| \times |V|}$ with

- $\mathbf{Y}'_{ij} = 1$ if edge i leaves vertex j
- $\mathbf{Y}'_{ij} = -1$ if edge i enters vertex j
- $\mathbf{Y}'_{ij} = 0$ otherwise

Now let us consider the following optimization problem:

$$\min_{\mathbf{A}',\mathbf{X}} \ \|\mathbf{Y}' - \mathbf{A}'\mathbf{X}'\|_F^2 \quad \text{s.t. } \|\mathbf{x}_i'\|_0 \leq s,\ \mathbf{1}^T\mathbf{x}_i' = 1,\ \forall i \tag{A.102}$$

with $s = 1$ and $k = 2$.

Claim 1: Problem (A.102) is equivalent to the densest cut problem over the graph $\mathcal{G}$ [241].

Claim 2: Consider two different feasible points $\mathbf{X}_1'$ and $\mathbf{X}_2'$ in problem (A.102). Let $\mathbf{A}_1'$ (resp. $\mathbf{A}_2'$) be the optimal solution of (A.102) after fixing the variable $\mathbf{X}'$ to $\mathbf{X}_1'$ (resp. $\mathbf{X}_2'$). Let us further assume that $\|\mathbf{Y}' - \mathbf{A}_1'\mathbf{X}_1\| \neq \|\mathbf{Y}' - \mathbf{A}_2'\mathbf{X}_2\|$. Then, $|\ \|\mathbf{Y}' - \mathbf{A}_1'\mathbf{X}_1\| - \|\mathbf{Y}' - \mathbf{A}_2'\mathbf{X}_2\|\ | \geq \frac{16}{N^3}$.

The proof of claims 1 and 2 are relegated to the appendix section. Clearly, problem (A.102) is different from (3.47); however the only difference is in the existence of the extra linear constraint in (A.102). To relate these two problems, let us define the following problem:

$$\min_{\mathbf{A},\mathbf{X}} \ \|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F^2 \quad \text{s.t. } \|\mathbf{x}_i\|_0 \leq s,\ \forall i. \tag{A.103}$$

where $\mathbf{X}$ is of the same dimension as $\mathbf{X}'$, but the matrices $\mathbf{Y}$ and $\mathbf{A}$ have one more row than $\mathbf{Y}'$ and $\mathbf{A}'$. Here the matrices $\mathbf{Y}$ and $\mathbf{A}$ have the same number of columns as $\mathbf{Y}'$ and $\mathbf{A}'$, respectively. By giving a special form to the matrix $\mathbf{Y}$, we will relate the optimization problem (A.103) to (A.102). More specifically, each column of $\mathbf{Y}$ is defined as follows:

$$\mathbf{y}_i = \begin{bmatrix} M \\ \mathbf{y}_i' \end{bmatrix}$$

with $M = 6N^7$. Clearly, the optimization problem (A.103) is of the form (3.47). Let $(\mathbf{A}^*, \mathbf{X}^*)$ denote the optimizer of (A.103). Then it is not hard to see that the first row of the matrix $\mathbf{A}^*$ should be nonzero and hence by a proper normalization of the matrices $\mathbf{A}^*$ and $\mathbf{X}^*$, we can assume that the first row of the matrix $\mathbf{A}^*$ is M, i.e., $a_{11}^* = a_{12}^* = M$. Define $h(\mathbf{A}, \mathbf{X}) \triangleq \|\mathbf{Y}' - \mathbf{A}\mathbf{X}\|_F^2$. Let $\mathbf{w}' = (\mathbf{A}'^*, \mathbf{X}'^*)$ denote the minimizer of (A.102). Similarly, define $\mathbf{w} \triangleq (\tilde{\mathbf{A}}^*, \mathbf{X}^*)$ where $\tilde{\mathbf{A}}^* \triangleq \mathbf{A}_{2:n,:}^*$ is the minimizer of (A.103), excluding the first row. Furthermore, define $\mathbf{w}_+ \triangleq \left(\tilde{\mathbf{A}}^*, \mathbf{X}_+^*\right)$, where $\mathbf{X}_+^*$ is obtained by replacing the nonzero entries of $\mathbf{X}^*$ with one. Having these definitions in our hands, the following claim will relate the two optimization problems (A.102) and (A.103).

Claim 3: $h(\mathbf{w}) \leq h(\mathbf{w}') \leq h(\mathbf{w}_+) \leq h(\mathbf{w}) + \frac{28}{3N^3}$.
The proof of this claim can be found in the appendix section.
Now set $\epsilon = \frac{28}{3N^3}$. If we can solve the optimization problem (A.103) to the ϵ-accuracy, then according to Claim 3, we have the optimal value of problem (A.102) with accuracy $\epsilon = \frac{28}{3N^3}$. Noticing that $\frac{16}{N^3} > \frac{28}{3N^3}$ and using Claim 2, we can further conclude that the exact optimal solution of (A.102) is known; which implies that the optimal value of the original densest cut problem is known (according to Claim 1). The NP-hardness of the densest cut problem will complete the proof. ■

Proof of Claim 1: This proof is exactly the same as the proof in [241]. Here we restate the proof since some parts of the proof is necessary for the proof of Claim 2. Consider a feasible point (A', X') of problem (A.102). Clearly, in any column of the matrix X', either the first component is zero, or the second one. This gives us a partition of the columns of the matrix X' (which is equivalent to a partition over the nodes of the graph). Let P (resp. Q) be the set of columns of X' for which the first (resp. the second) component is nonzero at the optimality. Define $p \triangleq |P|$ and $q = |Q|$. Then the optimal value of the matrix $\mathbf{A} = [\mathbf{a}_1 \mathbf{a}_2]$ is given by:

- $a_{j1} = \pm\frac{1}{p}$, $a_{j2} = \mp\frac{1}{q}$ if $j \in E(P, Q)$
- $a_{j1} = a_{j2} = 0$ if $j \notin E(P, Q)$

where a_{ji} is the j-th component of column i in matrix $\mathbf{A}$. Plugging in the optimal value of the matrix $\mathbf{A}$, the objective function of (A.102) can be rewritten as:

$$\begin{aligned}
\|\mathbf{Y}' - \mathbf{A}'\mathbf{X}'\|_F^2 &= \sum_{i \in P} \|\mathbf{y}'_i - \mathbf{a}'_1\|^2 + \sum_{i \in Q} \|\mathbf{y}'_i - \mathbf{a}'_2\|^2 \\
&= \sum_{j \notin E(P,Q)} 2 + \sum_{j \in E(P,Q)} \left[(1 - \frac{1}{p})^2 + \frac{p-1}{p^2} + (1 - \frac{1}{q})^2 + \frac{q-1}{q^2} \right] \\
&= 2\left(|E| - |E(P,Q)|\right) + |E(P,Q)|(\frac{p-1}{p} + \frac{q-1}{q}) \\
&= 2|E| - |E(P,Q)|(\frac{1}{p} + \frac{1}{q}) \\
&= 2|E| - |V|\frac{|E(P,Q)|}{p \cdot q} = 2n - N\frac{|E(P,Q)|}{p.q}. \qquad \text{(A.104)}
\end{aligned}$$

Hence, clearly, solving (A.102) is equivalent to solving the densest cut problem on graph $\mathcal{G}$. ■

Proof of Claim 2: According to the proof of Claim 1, we can write

$$\begin{aligned}\left|\|\mathbf{Y}' - \mathbf{A}_1'\mathbf{X}_1'\|_F^2 - \|\mathbf{Y}' - \mathbf{A}_2'\mathbf{X}_2'\|_F^2\right| &= N\left|\frac{|E(P_1,Q_1)|}{p_1q_1} - \frac{|E(P_2,Q_2)|}{p_2q_2}\right| \\ &\geq \frac{N}{p_1(N-p_1)p_2(N-p_2)} \\ &\geq \frac{N}{(N/2)^2} = \frac{16}{N^3}.\end{aligned}$$

■

Proof of Claim 3: First of all, notice that the point

$$\mathbf{X} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 0 & 0 & \cdots & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{A} = \begin{bmatrix} M & M \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix}$$

is feasible and it should have a higher objective value than the optimal one. Therefore,

$$\sum_{i=1}^{N}(M - M(x_{1i}^* + x_{2i}^*))^2 + h(\mathbf{w}) \leq \|\mathbf{Y}'\|_F^2 = 2|E| \leq 2N^2$$

which in turn implies that

$$\max_i\{|1 - x_{1i}^* - x_{2i}^*|\} \leq \frac{\sqrt{2}N}{M} = \frac{1}{3N^6} \triangleq \delta, \tag{A.105}$$

since $h(\mathbf{w}) \geq 0$. Clearly, $\delta < \frac{1}{2}$ and moreover notice that for each i only one of the elements x_{1i}^* and x_{2i}^* is nonzero. Therefore, any nonzero element x_{ij}^* should be larger than $\frac{1}{2}$. On the other hand, due to the way that we construct $\mathbf{Y}'$, we have $|y_{ij}'| \leq 1,\ \forall i, j$.

This implies that $|\tilde{a}_{ij}| \leq 2,\ \forall i,j$, leading to

$$\|\tilde{\mathbf{a}}_1\|^2, \|\tilde{\mathbf{a}}_2\|^2 \leq 4N, \tag{A.106}$$

where $\tilde{\mathbf{a}}_1$ and $\tilde{\mathbf{a}}_2$ are the first and the second column of matrix $\tilde{\mathbf{A}}$. Having these simple bounds in our hands, we are now able to bound $h(\mathbf{w}_+)$:

$$\begin{aligned}
h(\mathbf{w}_+) =& \sum_{i\in P} \|\mathbf{y}'_i - \tilde{\mathbf{a}}_1\|^2 + \sum_{i\in Q} \|\mathbf{y}'_i - \tilde{\mathbf{a}}_2\|^2 \\
&= \sum_{i\in P} \|\mathbf{y}'_i - \tilde{\mathbf{a}}_1 x_{1i}\|^2 + \sum_{i\in P} \|\mathbf{a}_1\|^2(1-x_{1i})^2 + 2\sum_{i\in P}\langle \mathbf{y}'_i - \tilde{\mathbf{a}}_1 x_{1i}, (x_{1i}-1)\tilde{\mathbf{a}}_1\rangle \\
&+ \sum_{i\in Q} \|\mathbf{y}'_i - \tilde{\mathbf{a}}_2 x_{2i}\|^2 + \sum_{i\in Q} \|\mathbf{a}_2\|^2(1-x_{2i})^2 + 2\sum_{i\in Q}\langle \mathbf{y}'_i - \tilde{\mathbf{a}}_2 x_{2i}, (x_{2i}-1)\tilde{\mathbf{a}}_2\rangle \\
&\leq h(\mathbf{w}) + \sum_i 4N^2\delta^2 + 2\sum_{i\in P}(\|\mathbf{y}'_i\| + x_{1i}\|\tilde{\mathbf{a}}_1\|)\cdot\|\tilde{\mathbf{a}}_1\|\cdot|1-x_{1i}| \\
&+ 2\sum_{i\in Q}(\|\mathbf{y}'_i\| + x_{2i}\|\tilde{\mathbf{a}}_2\|)\cdot\|\tilde{\mathbf{a}}_2\|\cdot|1-x_{2i}| \\
&\leq h(\mathbf{w}) + 4N^3\delta^2 + 2\sum_{i\in P}(\|\mathbf{y}'_i\| + 4N)2N\delta + 2\sum_{i\in Q}(\|\mathbf{y}'_i\| + 4N)2N\delta \\
&\leq h(\mathbf{w}) + 4N^3\delta^2 + 4N\delta(\sqrt{N}\|\mathbf{Y}'\|_F) + 16N^3\delta \\
&\leq h(\mathbf{w}) + 4N^3\delta^2 + 4N\delta(\sqrt{N}\|\mathbf{Y}'\|_F) + 16N^3\delta \\
&\leq h(\mathbf{w}) + 28N^3\delta \leq h(\mathbf{w}) + \frac{28}{3N^3}.
\end{aligned} \tag{A.107}$$

Furthermore, since $\mathbf{w}_+$ is a feasible point for (A.102) and due to the optimality of w', we have

$$h(\mathbf{w}') \leq h(\mathbf{w}_+). \tag{A.108}$$

On the other hand,

$$h(\mathbf{w}) \leq h(\mathbf{w}'); \tag{A.109}$$

otherwise, we can add the row $[M \quad M]$ on top of $\mathbf{A}'$ and get a lower objective for (A.103). Combining (A.107), (A.108), and (A.109) will conclude the proof. ■