



# Research Repository UCD

<b>Title</b>	Energy Efficiency Optimization for Dense Networks
<b>Authors(s)</b>	Vu, Quang-Doanh, Juntti, Markku, Hong, Een-Kee, Tran, Le-Nam
<b>Publication date</b>	2019-02-01
<b>Publication information</b>	Vu, Quang-Doanh, Markku Juntti, Een-Kee Hong, and Le-Nam Tran. "Energy Efficiency Optimization for Dense Networks." Wiley, February 1, 2019. <a href="https://doi.org/10.1002/9781119473756.ch8">https://doi.org/10.1002/9781119473756.ch8</a> .
<b>Publisher</b>	Wiley
<b>Item record/more information</b>	<a href="http://hdl.handle.net/10197/11905">http://hdl.handle.net/10197/11905</a>
<b>Publisher's version (DOI)</b>	<a href="https://doi.org/10.1002/9781119473756.ch8">10.1002/9781119473756.ch8</a>

Downloaded 2025-12-04 22:59:37

The UCD community has made this article openly available. Please share how this access benefits you. Your story matters! (@ucd\_oa)



© Some rights reserved. For more information

# Chapter 1

## Energy Efficiency

## Optimization for Dense Networks

Quang-Doanh Vu,<sup>1</sup> Markku Juntti,<sup>1</sup> Een-Kee Hong,<sup>2</sup> and Le-Nam Tran<sup>3\*</sup>

<sup>1</sup>*Centre for Wireless Communications, University of Oulu, Finland*

<sup>2</sup>*School of Electronics and Information, Kyung Hee University, South Korea*

<sup>3</sup>*School of Electrical and Electronic Engineering, University College Dublin, Ireland*

\*Corresponding Author: Le-Nam Tran; nam.tran@ucd.ie

Dense networks open opportunities to optimize the network performance particularly from an energy efficiency perspective, since the total power consumption is a great concern as the number of devices is very large. The goal of this chapter is to provide a set of optimization tools applied in designing energy efficiency transmission for dense networks. Specifically, this chapter includes two parts. The first part introduces optimization techniques that are useful for energy efficiency optimization including concave-convex fractional programming, nontractable fractional programming, the alternating direction method of multipliers for distributed implementation. The second

part is to demonstrate how these methods can be applied to dense networks with shared spectrum and small-cell dense networks being the case studies.

## 1.1. Introduction

The two major challenges for the future wireless networks are the ever-growing data traffic and the increasing energy consumption. Both are the natural result of the popularity of wireless communications. In particular, energy consumption in wireless networks needs to be satisfactorily dealt with for sustainable economic growth. Dense network paradigm is introduced in order to solve the problem of exponential growth of data demand in communications industry. Moreover, network densification is also expected to be the key architecture for future wireless networks where wireless connectivity is pervasive. By bringing base stations near to the end users, dense small cell deployment has potential of using energy in an efficient manner, since the required transmit power for combating the path loss is reduced. However, dense small cell deployment has its own cost. Using more base stations (BSs) means that the large number of hardware elements may lead to a sharp increase in the circuit power consumption. Moreover, with full frequency reused and the small cells being close together, the interference become complicated. If not properly managed, this will become a performance bottleneck in dense networks.

In general, efficiency is defined as the magnitude to which the least amount of resources is used to achieve a certain target. Thus, efficiency is often expressed as the ratio of the obtained output to the resource consumption. For data communications, the outcome is usually the amount of received (or transmitted) information and the resource is the required power. Consequently, the

definition of energy efficiency (EE) in data communications is given by [12]

$$\mathcal{E}_{\text{eff}} = \frac{\int i(t)dt}{\int p(t)dt} \quad (1.1)$$

where  $i(t)$  and  $p(t)$  are the reliably decoded information and consumed power at a given time  $t$ , respectively. Thus, energy efficiency is measured in *bits (or nats) per Joule*. Commonly, the channels are supposed to be flat and quasi-static, then (1.1) simplifies to [13]

$$\mathcal{E}_{\text{eff}} = \frac{R}{P} \quad (1.2)$$

where  $R$  is the total data rate and  $P$  is the total power consumed in the corresponding signaling interval. The EE definition in (1.2) is mainly used in this chapter.

By the definition, the EE optimization involves fractional programs. In some special cases, e.g. quasi-convex problems, a fractional program can be solved efficiently using, e.g. Dinkelbach's algorithms or Charnes Cooper's transformation. For others, the fractional program of interest is *truly nonconvex* in the sense that there is no equivalent convex reformulation, and globally optimal solutions cannot be found in polynomial time. In such situations, to find a locally optimal solution with low-complexity is more practically useful.

In dense networks, there are possibly hundreds of nodes in a small geographic area. Thus, it is practically impossible and inefficient to coordinate the operation of all nodes with some kind of centralized mechanisms since the networks need to be reorganized frequently. The signaling for gathering relevant information may be overwhelming. Consequently, it is certain that self-organization including self-optimization is necessary. Thus, it is important

to design algorithms which can be implemented in a decentralized manner.

This chapter is organized as follows. In the first part, we discuss the fractional programs and the methods that can be used to solve these, including Dinkelbach's methods, parameter-free equivalent transformation, and successive convex approximation. The alternating direction method of multipliers (ADMM) is also introduced as a powerful approach for parallel optimization algorithms. In the second part, we illustrate the applications of these tools in two specific scenarios of dense networks.

## 1.2. Energy Efficiency Optimization Tools

### 1.2.1. Fractional Programming

An energy efficiency maximization problem can be generally formulated as a fractional program

$$\underset{\mathbf{x} \in \mathcal{K}}{\text{maximize}} \ f(\mathbf{x}) \quad (1.3)$$

where  $\mathcal{K} \in \mathbb{R}^n$  represents the design constraints, and  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  can take one of the following forms:

$$f(\mathbf{x}) \triangleq \frac{\sum_{i=1}^q g_i(\mathbf{x})}{\sum_{i=1}^q h_i(\mathbf{x})}, \quad (1.4)$$

$$f(\mathbf{x}) \triangleq \sum_{i=1}^q \frac{g_i(\mathbf{x})}{h_i(\mathbf{x})}, \quad (1.5)$$

or

$$f(\mathbf{x}) \triangleq \min_{1 \leq i \leq q} \left\{ \frac{g_i(\mathbf{x})}{h_i(\mathbf{x})} \right\} \quad (1.6)$$

with  $g_i : \mathbb{R}^n \rightarrow \mathbb{R}_+$ ,  $h_i : \mathbb{R}^n \rightarrow \mathbb{R}_{++}$  for all  $i = 1, \dots, q$ , and  $q \geq 1$ . Specifically,  $g_i(\cdot)$  and  $h_i(\cdot)$  stand for the expression of the data rate and power consump-

tion in relation to transceiver  $i$ . As we shall see, the assumptions  $g_i(\cdot) \geq 0$  and  $h_i(\cdot) > 0$  for all  $i$  are automatically satisfied in the EE problems to be considered in this chapter since the consumed power is always positive and the achievable rate is always nonnegative. The cost function in (1.4) involves a single ratio, and it arises from the problem of maximizing EE of the overall network [24]. The cost function in (1.5) is the sum of multiple ratios, which appears in the problem of weighted sum energy efficiency [22]. The objective in (1.6) results in a max-min fractional program, which occurs when the EE fairness among the nodes is concerned [18]. Generally, the EE maximization problems are not convex even if  $\mathcal{K}$  is convex. The difficulty in dealing with different classes of fractional programs is discussed below.

- For the objective in (1.4) and  $\mathcal{K}$  is convex,  $g_i$  is concave and  $h_i(\mathbf{x})$  is convex on  $\mathcal{K}$ , the resulting problem is called concave fractional program (CFP), which is a class of quasi-concave programs. Dinkelbach's method or Charnes Cooper's transformation can be used to solve this type of problems efficiently.
- For the objective in (1.6) and  $\mathcal{K}$  is convex,  $g_i$  is concave and  $h_i(\mathbf{x})$  is convex on  $\mathcal{K}$ , the obtained program is called max-min fractional program (MMFP). This problem is considered as a *generalized* convex program for which efficient solution is also possible [21].
- For the remaining cases, efficient optimal solutions for (1.3) remain open. Alternatively, one is more interested in finding the local solution for these programs.

## 1.2.2. Concave Fractional Programs

We provide two well-known approaches solving a CFP. Let  $g(\mathbf{x}) = \sum_{i=1}^p g_i(\mathbf{x})$  and  $h(\mathbf{x}) = \sum_{i=1}^p h_i(\mathbf{x})$ . Then the CFP is rewritten as

$$\max_{\mathbf{x} \in \mathcal{K}} f(\mathbf{x}) = \frac{g(\mathbf{x})}{h(\mathbf{x})}. \quad (1.7)$$

Note that the concavity of  $g(\mathbf{x})$  and convexity of  $h(\mathbf{x})$  are easily justified from that of the individual functions.

### 1.2.2.1. Parameterized Approach

The parameterized approach for solving (1.7) is to solve a series of concave programs. In particular, let us consider the following parametric problem

$$\mathcal{P}(\alpha) = \max_{\mathbf{x} \in \mathcal{K}} \{g(\mathbf{x}) - \alpha h(\mathbf{x})\} \quad (1.8)$$

where  $\alpha > 0$ . The relationship between CFP (1.7) and (1.8) is as follows [11].

**Lemma 1.1:** *A point  $\mathbf{x}^*$  is an optimal solution of (1.7) if and only if  $\mathcal{P}\left(\frac{g(\mathbf{x}^*)}{h(\mathbf{x}^*)}\right) = 0$ . In addition, there is unique  $\alpha^*$  such that  $\mathcal{P}(\alpha^*) = 0$ .*

Thus one can easily obtain  $\mathbf{x}^*$  when  $\alpha^*$  is given. Based on the relationship and the fact that  $\mathcal{P}(\alpha)$  is strictly monotonic decreasing, Dinkelbach developed an iterative procedure whose main steps are outlined in Algorithm 1.1 for solving the CFP [11]. In each iteration, a concave problem is solved and the value of  $\alpha$  is updated. It is worth mentioning that the properties of sequence  $\{\alpha^{(l)}\}_l$  obtained by Dinkelbach's algorithm are

- $\{\alpha^{(l)}\}_l$  is an increasing sequence.

- $\lim_{l \rightarrow \infty} \alpha^{(l)} = \alpha^*$ .

---

**Algorithm 1.1** Dinkelbach's procedure for solving CFP

---

- 1: **Initialization:**  $\mathbf{x}^{(0)} \in \mathcal{K}$  (or set  $\alpha^{(0)} = 0$ );  $l = 0$ ; error tolerant  $\epsilon > 0$ .
  - 2: **repeat**
  - 3:    $\alpha^{(l)} := \frac{g(\mathbf{x}^{(l)})}{h(\mathbf{x}^{(l)})}$
  - 4:    $\mathbf{x}^{(l+1)} := \arg \max_{\mathbf{x} \in \mathcal{K}} g(\mathbf{x}) - \alpha^{(l)} h(\mathbf{x})$
  - 5:    $l := l + 1$
  - 6: **until**  $(g(\mathbf{x}^{(l)}) - \alpha^{(l-1)} h(\mathbf{x}^{(l)})) < \epsilon$ .
  - 7: **Output:**  $\mathbf{x}^{(l)}$ .
- 

### 1.2.2.2. Parameter-Free Approach

Another well-known approach solving the CFP is to transform it into an equivalent convex problem. Here the equivalence means that the optimal solution of the former can be obtained from that of the latter. Such a transformation was first introduced in [6] for linear CFP. The technique was then extended to nonlinear CFP in [20]. Particularly, the CFP in (1.7) is equivalent to the following problem

$$\underset{(\mathbf{y}, \mu) \in \tilde{\mathcal{K}}}{\text{maximize}} \quad \mu g\left(\frac{\mathbf{y}}{\mu}\right) \quad (1.9)$$

where

$$\tilde{\mathcal{K}} = \left\{ \mathbf{y} \in \mathbb{R}^n, \mu | \mu > 0, \frac{\mathbf{y}}{\mu} \in \mathcal{K}, \mu h\left(\frac{\mathbf{y}}{\mu}\right) \leq 1 \right\}. \quad (1.10)$$

Clearly, (1.9) is a convex program (concave maximization to be precise) since  $\mu g\left(\frac{\mathbf{y}}{\mu}\right)$  and  $\mu h\left(\frac{\mathbf{y}}{\mu}\right)$  are concave and convex, respectively. Note that the perspective functions preserve the convexity of the original ones. The relationship between CFP (1.7) and (1.9) is stated in the following lemma [20].

**Lemma 1.2:** Suppose  $(\mathbf{y}^*, \mu^*)$  to be an optimal solution of (1.9), then  $\left(\frac{\mathbf{y}^*}{\mu^*}\right)$



is the optimal of CFP (1.7). Reversely, suppose  $\mathbf{x}^*$  to be an optimal solution of CFP (1.7), then  $\left(\frac{\mathbf{x}^*}{h(\mathbf{x}^*)}, \frac{1}{h(\mathbf{x}^*)}\right)$  is the optimal of (1.9).

The above lemma means once an optimal of one of the two problem is known, we can easily obtain that of the other. Remark that when  $h$  is linear, we can write  $\mu h\left(\frac{\mathbf{y}}{\mu}\right) = 1$ .

There exist other approaches for dealing with the CFP. The interested reader is referred to [21] for further details.

### 1.2.3. Max-Min Fractional Programs

We now discuss details on the MMFP. In fact, this class of problems can also be solved via a parameterized approach where the iterative procedure is an extension of Algorithm 1.1 [8]. The main steps are outlined in Algorithm 1.2. Similar to Algorithm 1.1, the sequence  $\{\alpha^{(l)}\}_l$  returned by Algorithm 1.2 is also increasing and converges to an optimal solution of the MMFP. However, the convergence rate of Algorithm 1.2 (normally linear rate) is slower than that of Algorithm 1.1 (superlinear convergence). Some methods for improving the convergence rate for the MMFP can be found in [9] and the references therein.

The subproblem at Step 4 in Algorithm 1.2 is a convex program since the pointwise minimum of concave functions is concave. Currently, some solvers and parsers do not accept the pointwise operations. This practical issue can be simply overcome via the epigraph of the problem, i.e. by introducing a new variable  $\beta$  we can rewrite the problem as  $\underset{(\mathbf{x}, \beta) \in \hat{\mathcal{K}}^{(l)}}{\text{maximize}} \beta$  where  $\hat{\mathcal{K}}^{(l)} = \{\mathbf{x} \in \mathcal{K}, \beta | g_i(\mathbf{x}) - \alpha^{(l)} h_i(\mathbf{x}) \geq \beta, i = 1, \dots, q\}$ .

---

**Algorithm 1.2** Generalized Dinkelbach's procedure for solving MMFP

---

- 1: **Initialization:**  $\mathbf{x}^{(0)} \in \mathcal{K}$ ,  $l = 0$ , error tolerant  $\epsilon > 0$ .
  - 2: **repeat**
  - 3:    $\alpha^{(l)} := \min_{1 \leq i \leq q} \left\{ \frac{g_i(\mathbf{x}^{(l)})}{h_i(\mathbf{x}^{(l)})} \right\}$
  - 4:    $\mathbf{x}^{(l+1)} := \arg \max_{\mathbf{x} \in \mathcal{K}} \min_{1 \leq i \leq q} \{g_i(\mathbf{x}) - \alpha^{(l)} h_i(\mathbf{x})\}$
  - 5:    $l := l + 1$
  - 6: **until**  $\min_{1 \leq i \leq q} \{g_i(\mathbf{x}^{(l)}) - \alpha^{(l-1)} h_i(\mathbf{x}^{(l)})\} < \epsilon$ .
  - 7: **Output:**  $\mathbf{x}^{(l)}$ .
- 

### 1.2.4. Generalized Nonconvex Fractional Programs

For the remaining cases, (1.3) has no hidden convexity and thus is generally difficult to find an optimal solution. In this section we present an efficient local optimization framework called successive convex approximation (SCA) which is widely applied to solve nonconvex problem in wireless communications.

By a slight abuse of notation, let us consider the general optimization problem

$$\begin{aligned} & \underset{\mathbf{x} \in \mathbb{R}^n}{\text{maximize}} && f(\mathbf{x}) \\ & \text{subject to} && g_i(\mathbf{x}) \leq 0, \ i = 1, \dots, p \\ & && h_k(\mathbf{x}) \leq 0, \ k = 1, \dots, q \end{aligned} \tag{1.11}$$

where  $f(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$  and  $g_i(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $i = 1, \dots, p$ , are differentiable convex functions,  $h_k(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $k = 1, \dots, q$  are differentiable functions, and  $\mathcal{K} \triangleq \{\mathbf{x} \in \mathbb{R}^n | g_i(\mathbf{x}) \leq 0, \ i = 1, \dots, p, h_k(\mathbf{x}) \leq 0, \ k = 1, \dots, q\}$  is a compact set. Clearly, the *nonconvex part* of the problem is due to the last  $q$  constraints. To locally solve (1.11), an iterative procedure outlined in Algorithm 1.3 was first introduced in [16], and thoroughly studied later in [1]. The idea is to suc-

cessively approximate the nonconvex set by its inner one via replacing  $\{h_k(\mathbf{x})\}_k$  by their convex upper bounds. Particularly, let us denote by  $\bar{\mathbf{x}}$  a feasible point of (1.11). Let  $\bar{\mathbf{y}}_k \triangleq \bar{h}_k(\bar{\mathbf{x}}) : \mathbb{R}^n \rightarrow \mathbb{R}^{m_k}$ ,  $m_k \geq 1$  and  $\{\tilde{h}_k(\mathbf{x}; \bar{\mathbf{y}}_k) : \mathbb{R}^n \rightarrow \mathbb{R}\}$  be the convex upper bounds functions having the following properties

$$\begin{aligned} h_k(\mathbf{x}) &\leq \tilde{h}_k(\mathbf{x}; \bar{\mathbf{y}}), \quad \forall \mathbf{x} \in \mathcal{K} \\ h_k(\bar{\mathbf{x}}) &= \tilde{h}_k(\bar{\mathbf{x}}; \bar{\mathbf{y}}) \\ \nabla_{\mathbf{x}} h_k(\mathbf{x}) &= \nabla_{\mathbf{x}} \tilde{h}_k(\mathbf{x}; \bar{\mathbf{y}}). \end{aligned} \tag{1.12}$$

The first and the second properties guarantee that the sequence of objective value obtained by the algorithm is nondecreasing while the second and the third properties ensure that the convergent points satisfy the necessary optimality (Karush-Kuhn-Tucker) conditions of (1.11).

---

**Algorithm 1.3** General Procedure of Successive Convex Approximation method

- 1: **Initialization:** generate random point  $\mathbf{x}^{(0)} \in \mathcal{K}$ . Set  $l = 0$
- 2: **repeat**
- 3:     Determine  $\bar{\mathbf{y}}_k^{(l)} \triangleq \bar{h}_k(\mathbf{x}^{(l)})$  for all  $k$
- 4:     Solve 
$$\begin{aligned} & \underset{\mathbf{x} \in \mathbb{R}^n}{\text{maximize}} && f(\mathbf{x}) \\ & \text{subject to} && g_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, p, \\ & && \tilde{h}_k(\mathbf{x}; \bar{\mathbf{y}}_k^{(l)}) \leq 0, \quad k = 1, \dots, q \end{aligned}$$
- and denote the obtained solution by  $\mathbf{x}^{(l+1)}$ .
- 5:     Update  $l := l + 1$
- 6: **until** Convergence
- 7: **Output:**  $\mathbf{x}^{(l)}$

There are several convergence results of the SCA method [16, 1, 10]. Herein we briefly summarize these results for the sake of completeness. First, if  $f(\mathbf{x})$  is coercive on the feasible set or the feasible set is bounded, then  $\{f(\mathbf{x}^{(l)})\}_l$  converges to a finite value since the sequence  $\{f(\mathbf{x}^{(l)})\}_{l=1}^\infty$  is nonincreasing. Moreover, if  $f(\mathbf{x})$  is strongly convex, the convergence of the iterates  $\{\mathbf{x}^{(l)}\}_{l=0}^\infty$

is guaranteed [1, Theorem 1]. This means, when  $f(\mathbf{x})$  is not strongly convex, Algorithm 1.3 is not guaranteed to output a Karush-Kuhn-Tucker solution. To overcome this issue, we can add a *proximal term* [10], i.e. the objective function at each iteration is modified into  $f(\mathbf{x}) + \lambda \|\mathbf{x} - \mathbf{x}^{(l)}\|_2^2$  where  $\lambda$  is some positive scalar.

### 1.2.5. Alternating Direction Method of Multipliers for Distributed Implementation

One of the popular optimization techniques for developing decentralized algorithms in wireless communications design is the dual decomposition method. The ADMM is a combination of the dual descent method and the multiplier method [4] being amenable to distributed implementations especially for large scale systems. A good introduction of ADMM can be found in [4]. To be self-contained we provide a brief review of ADMM which is necessary for the next section. Let us consider the convex problem

$$\begin{aligned} & \underset{\mathbf{x} \in \mathcal{K}_1, \mathbf{y} \in \mathcal{K}_2}{\text{minimize}} && f(\mathbf{x}) + h(\mathbf{y}) \\ & \text{subject to} && \mathbf{Ax} + \mathbf{By} = \mathbf{c} \end{aligned} \tag{1.13}$$

where  $\mathcal{K}_1 \in \mathbb{R}^{n_1}$  and  $\mathcal{K}_2 \in \mathbb{R}^{n_2}$  are convex sets,  $\mathbf{A} \in \mathbb{R}^{m \times n_1}$ ,  $\mathbf{B} \in \mathbb{R}^{m \times n_2}$  and  $\mathbf{c} \in \mathbb{R}^m$ ;  $f : \mathbb{R}^{n_1} \rightarrow \mathbb{R}$  and  $h : \mathbb{R}^{n_2} \rightarrow \mathbb{R}$  are convex. Different from the dual decomposition method, the ADMM works on the augmented Lagrangian function given by

$$L_A(\mathbf{x}, \mathbf{y}; \boldsymbol{\mu}) = f(\mathbf{x}) + h(\mathbf{y}) + \boldsymbol{\mu}^T (\mathbf{Ax} + \mathbf{By} - \mathbf{c}) + \frac{d}{2} \|\mathbf{Ax} + \mathbf{By} - \mathbf{c}\|_2^2 \tag{1.14}$$

where  $\boldsymbol{\mu}$  is the Lagrange multiplier (or dual variable) vector and  $d > 0$  is the penalty parameter.

The ADMM is an iterative procedure where the variables are successively updated. In particular, the following steps are taken place at the  $l$ th iteration

$$\begin{aligned}\mathbf{x}^{(l)} &:= \underset{\mathbf{x} \in \mathcal{K}_1}{\text{minimize}} L_A(\mathbf{x}, \mathbf{y}^{(l-1)}, \boldsymbol{\mu}^{(l-1)}) \\ \mathbf{y}^{(l)} &:= \underset{\mathbf{y} \in \mathcal{K}_2}{\text{minimize}} L_A(\mathbf{x}^{(l)}, \mathbf{y}, \boldsymbol{\mu}^{(l-1)}) \\ \boldsymbol{\mu}^{(l)} &:= \boldsymbol{\mu}^{(l-1)} + d(\mathbf{A}\mathbf{x}^{(l)} + \mathbf{B}\mathbf{y}^{(l)} - \mathbf{c})\end{aligned}\tag{1.15}$$

Although the update in (1.15) is quite similar to that in the dual decomposition method, the ADMM converges with quite mild conditions. In particular, the ADMM does not require  $f(\mathbf{x})$  and  $h(\mathbf{y})$  to be strictly convex as the strong convexity of the objective is automatically achieved via the penalty term  $\frac{d}{2} \|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} - \mathbf{c}\|_2^2$  included in the augmented Lagrangian. We will see that this property is important for developing distribution solutions to EE problems. We refer the interested reader to [4] for technical details about the convergence of the ADMM.

## 1.3. Energy Efficiency Optimization for Dense Networks: Case Studies

### 1.3.1. Multi-Radio Access Technologies

In the first case study, we apply the provided optimization tools to optimize the energy efficiency of the a dense network where multiple radio access technologies (RATs) coexist. These include wireless local area network (WLAN),

wireless metropolitan area network (WMAN), and the emerging long term evolution (LTE) for examples. Due to the evolution of smart-phone, there are more and more portable devices equipped with multiple wireless interfaces. In an ultra-dense network, it is reasonable to assume that there are many geographic areas covered by multiple radio access networks (RANs). When subscribers enter an overlapped coverage area, their devices can be configured to receive data from multiple RANs [5, 27].

This section investigates resource allocation in Multi-RATs scenario where users can simultaneously receive data from multiple RANs using multiple air interfaces. The target is optimally assigning the bandwidth and power to each user-RAN connection so as to maximize EE of the entire network subject to user specific quality of service (QoS) requirements as well as the available resource budgets. Our main focus in this section includes: first the EE maximization (EEmax) problem is formulated, and then an iterative algorithm for solving the problem based on the Dinkelbach method is presented. To facilitate a distributed implementation, we transform the EEmax problem into an equivalent convex program by applying the Charnes-Cooper transformation. Finally, we resort to the ADMM to propose a decentralized algorithm. The contents in this section are built on [23] where *the uplink* transmission is considered.

### **1.3.1.1. System Model and Energy Efficiency Maximization Problem**

We consider a region covered by a set of  $B$  wireless access network base stations (BSs) denoted by  $\mathcal{B} = \{1, 2, \dots, B\}$ . In this region, there exist a set of  $U$  multi-homing users, denoted by  $\mathcal{U} = \{1, 2, \dots, U\}$ , which are able to receive data from multiple network BSs simultaneously. We assume that all BSs and users are

equipped with single-antenna. The objective is to allocate BSs' power and the available bandwidth for each user-BS link to maximize the EE of the entire network. For the connection between user  $u$  and BS  $b$ , let us denote by  $\beta_{ub}$  and  $\rho_{ub}$  the allocated bandwidth and power, respectively. Then, the data rate transmitted over the connection is

$$r_{ub} = a_b \beta_{ub} \log \left( 1 + \frac{g_{ub} \rho_{ub}}{\beta_{ub}} \right) \quad (1.16)$$

where  $a_b \in (0, 1)$  is the network efficiency depending on the network [7],  $g_{ub} \triangleq |h_{ub}|^2 / N_0$ ,  $N_0$  is the power spectral density of the Gaussian background noise, and  $h_{ub}$  is the channel coefficient. We assume that the channels are flat and the channel coefficients remain constant during a transmission time interval [7, 17]. The QoS constraint for each user  $u$  is given as

$$\sum_{b \in \mathcal{B}} r_{ub} \geq \bar{R}_u \quad (1.17)$$

where  $\bar{R}_u$  is the predefined data rate threshold. Each user may be assigned by the corresponding network provider a certain degree of priority. Let us denote by  $m_{ub}$  the priority of user  $u$  in the network of BS  $b$ . Then, the overall weighted sum rate transmitted of the entire network is

$$R = \sum_{b \in \mathcal{B}} \sum_{u \in \mathcal{U}} m_{ub} r_{ub}. \quad (1.18)$$

The total bandwidth that BS  $b$  can allocate to the users is limited, i.e.,

$$\sum_{u \in \mathcal{U}} \beta_{ub} \leq \bar{W}_b, \forall b \in \mathcal{B} \quad (1.19)$$

where  $\bar{W}_b$  is the maximum bandwidth. There is also a constraint on the total transmit power at each BS given by

$$\sum_{u \in \mathcal{U}} \rho_{ub} \leq \bar{P}_b \quad (1.20)$$

where  $\bar{P}_b$  is the maximum power budget of BS  $b$ .

Let  $p_b^{\text{ct}}$  and  $p_u^{\text{cr}}$  be the circuit power for activating BS  $b$  and user  $u$ , respectively. In addition, the data transmission between user  $u$  and BS  $b$  requires an amount of circuit power  $\rho_{ub}^{\text{ca}}$  for activating the processing blocks (e.g., mixers, filters). Then, the total power consumed in the network is

$$P = p^c + \sum_{b \in \mathcal{B}} \frac{1}{\lambda_b} \left( \sum_{u \in \mathcal{U}} \rho_{ub} \right) \quad (1.21)$$

where  $p^c = \sum_{b \in \mathcal{B}} (p_b^{\text{ct}} + \sum_{u \in \mathcal{U}} p_{ub}^{\text{ca}}) + \sum_{u \in \mathcal{U}} p_u^{\text{cr}}$  is constant, and  $\lambda_b \in (0, 1)$  is the power amplifier's efficiency at BS  $b$ . The EE of the whole network is

$$\mathcal{E}(\boldsymbol{\rho}, \boldsymbol{\beta}) \triangleq \frac{R}{P} \quad (1.22)$$

where  $\boldsymbol{\rho} \triangleq \{\rho_{ub}\}_{u,b}$  and  $\boldsymbol{\beta} \triangleq \{\beta_{ub}\}_{u,b}$ .

The design problem is to find optimal values for  $\boldsymbol{\rho}$  and  $\boldsymbol{\beta}$  to maximize



$\mathcal{E}(\boldsymbol{\rho}, \boldsymbol{\beta})$  which is stated as

$$\underset{\boldsymbol{\rho} \geq 0, \boldsymbol{\beta} \geq 0}{\text{maximize}} \mathcal{E}(\boldsymbol{\rho}, \boldsymbol{\beta}) \quad (1.23a)$$

$$\text{subject to } \sum_{u \in \mathcal{U}} \rho_{ub} \leq \bar{P}_b, \forall b \in \mathcal{B} \quad (1.23b)$$

$$\sum_{u \in \mathcal{U}} \beta_{ub} \leq \bar{W}_b, \forall b \in \mathcal{B} \quad (1.23c)$$

$$\sum_{b \in \mathcal{B}} r_{ub} \geq \bar{R}_u, \forall u \in \mathcal{U} \quad (1.23d)$$

We assume that there exists a proper scheduler guaranteeing problem (1.23) is feasible [26, 14, 15, 19]. We further assume that the system has enough resources such that the data rate of all users can be larger than thresholds, i.e. there exist  $\boldsymbol{\rho}$  and  $\boldsymbol{\beta}$  such that the constraints hold with strict inequalities.

The rate function (1.16) is the perspective function of the concave function  $a_b \log_2(1 + g_{ub}\rho_{ub})$ , thus it is jointly concave with respect to  $\beta_{ub}$  and  $\rho_{ub}$ . In addition, the feasible set of (1.23) is convex. Hence problem (1.23) is a CFP which can be optimally solved by using parameterized or parameterized-free approaches.

### 1.3.1.2. Solution via Parameterized Approach

We first solve (1.23) using the procedure in Algorithm 1.1. Let us define the function  $f_\alpha(\boldsymbol{\rho}, \boldsymbol{\beta}) \triangleq R - \alpha P$  for a fixed  $\alpha > 0$  and arrive at the parameterized

problem

$$\underset{\boldsymbol{\rho} \geq 0, \boldsymbol{\beta} \geq 0}{\text{maximize}} \quad f_{\alpha}(\boldsymbol{\rho}, \boldsymbol{\beta}) \quad (1.24a)$$

$$\text{subject to} \quad \sum_{u \in \mathcal{U}} \rho_{ub} \leq \bar{P}_b, \forall b \in \mathcal{B} \quad (1.24b)$$

$$\sum_{u \in \mathcal{U}} \beta_{ub} \leq \bar{W}_b, \forall b \in \mathcal{B} \quad (1.24c)$$

$$\sum_{b \in \mathcal{B}} r_{ub} \geq \bar{R}_u, \forall u \in \mathcal{U} \quad (1.24d)$$

In each iteration, problem (1.24) is solved and parameter  $\alpha$  is updated as  $\alpha := \mathcal{E}(\boldsymbol{\rho}, \boldsymbol{\beta})$ .

There are some interesting insights into the energy-efficient resource allocation inferred from (1.24). Recall that (1.23) is assumed to be strictly feasible, thus strong duality holds for problem (1.23) by Slater's condition and the duality gap is zero for (1.24) [3]. By maneuvering the Lagrangian function of (1.24), we can express the optimal transmit power and allocated bandwidth for a given  $\alpha$  as

$$\rho_{ub}^* = \beta_{ub}^* \left[ \frac{a_b(m_{ub} + \mu_u^*)}{(\nu_b^* + \alpha)} - \frac{1}{g_{ub}} \right]^+ \quad (1.25)$$

where  $[x]^+ \triangleq \max\{0, x\}$ ,  $\{\nu_b^*\}_{b \in \mathcal{B}} \geq 0$ , and  $\{\mu_u^*\}_{u \in \mathcal{U}} \geq 0$  are the optimal Lagrangian multipliers corresponding to (1.24b) and (1.24d), respectively. We can observe from (1.25) that user  $u$  receives data from BS  $b$  if the channel gain  $g_{ub}$  is good enough. Moreover, power  $\rho_{ub}^*$  increases with  $a_b$  and  $m_{ub}$  which intuitively means that more power should be allocated to a user having higher network efficiency and/or priority.

### 1.3.1.3. Solution via Parameter-Free Approach

We can use the Charnes-Cooper transformation to translate (1.23) into the equivalent convex problem given as

$$\underset{\tilde{\rho} \geq 0, \tilde{\beta} \geq 0, \mu > 0}{\text{maximize}} \quad \tilde{\mathcal{E}}(\tilde{\rho}, \tilde{\beta}) \triangleq \sum_{b \in \mathcal{B}} \sum_{u \in \mathcal{U}} m_{ub} a_b \tilde{\beta}_{ub} \log \left( 1 + \frac{g_{ub} \tilde{\rho}_{ub}}{\tilde{\beta}_{ub}} \right) \quad (1.26a)$$

$$\text{subject to } \mu p^c + \sum_{b \in \mathcal{B}} \frac{1}{\lambda_b} \sum_{u \in \mathcal{U}} \tilde{\rho}_{ub} = 1 \quad (1.26b)$$

$$\sum_{u \in \mathcal{U}} \tilde{\rho}_{ub} \leq \mu \bar{P}_b, \forall b \in \mathcal{B} \quad (1.26c)$$

$$\sum_{u \in \mathcal{U}} \tilde{\beta}_{ub} \leq \mu \bar{W}_b, \forall b \in \mathcal{B} \quad (1.26d)$$

$$\sum_{b \in \mathcal{B}} a_b \tilde{\beta}_{ub} \log \left( 1 + \frac{g_{ub} \tilde{\rho}_{ub}}{\tilde{\beta}_{ub}} \right) \geq \mu \bar{R}_u, \forall u \in \mathcal{U} \quad (1.26e)$$

After solving (1.26) and obtaining the optimal solution  $(\tilde{\rho}^*, \tilde{\beta}^*, \mu^*)$ , we follow the result in Lemma 1.2 to determine the optimal of (1.23) as

$$\rho^* = \frac{\tilde{\rho}^*}{\mu^*}, \beta^* = \frac{\tilde{\beta}^*}{\mu^*}. \quad (1.27)$$

### 1.3.1.4. Distributed Implementation

The parameterized approach is not suitable for decentralized implementation since it requires updating parameter  $\alpha$  which is only done when the global data rate and global consumed power are collected. Thus we focus on the convex equivalent program (1.26). The current form (1.26) is not amenable to applying the ADMM. Hence, we first translate it into a proper formulation. Let  $\tilde{\rho}_b \triangleq \{\tilde{\rho}_{ub}\}_{u \in \mathcal{U}}$  and  $\tilde{\beta}_b \triangleq \{\tilde{\beta}_{ub}\}_{u \in \mathcal{U}}$ . For the connection between user  $u$  and BS  $b$ ,  $\forall u, b$ , we introduce two local variables  $t_{ub}$  and  $t'_{ub}$  and a global variable  $v_{ub}$ . We also introduce local variables  $\tilde{\mu}_b$  for all  $b \in \mathcal{B}$ , and  $\tilde{\mu}'_u$  for all  $u \in \mathcal{U}$ .

With these new variables, we equivalently rewrite (1.26) as

$$\begin{aligned} & \underset{\substack{\tilde{\boldsymbol{\rho}}, \tilde{\boldsymbol{\beta}}, \mu > 0 \\ \mathbf{t}, \mathbf{t}', \mathbf{v} \geq 0, \tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\mu}}'}}{\text{minimize}} & - \sum_{b \in \mathcal{B}} \mathbf{m}_b^T \mathbf{t}_b \end{aligned} \quad (1.28a)$$

$$\text{subject to } \mu p^c + \sum_{b \in \mathcal{B}} \frac{1}{\lambda_b} \mathbf{1}^T \tilde{\boldsymbol{\rho}}_b = 1 \quad (1.28b)$$

$$\mathbf{1}^T \tilde{\boldsymbol{\rho}}_b \leq \tilde{\mu}_b \bar{P}_b, \forall b \in \mathcal{B} \quad (1.28c)$$

$$\mathbf{1}^T \tilde{\boldsymbol{\beta}}_b \leq \tilde{\mu}_b \bar{W}_b, \forall b \in \mathcal{B} \quad (1.28d)$$

$$R_{ub}(\tilde{\boldsymbol{\rho}}_b, \tilde{\boldsymbol{\beta}}_b) \geq t_{ub}, \forall b \in \mathcal{B}, u \in \mathcal{U} \quad (1.28e)$$

$$\mathbf{1}^T \mathbf{t}'_u \geq \tilde{\mu}'_u \bar{R}_u, \forall u \in \mathcal{U} \quad (1.28f)$$

$$t_{ub} = v_{ub}, \forall b \in \mathcal{B}, u \in \mathcal{U} \quad (1.28g)$$

$$t'_{ub} = v_{ub}, \forall b \in \mathcal{B}, u \in \mathcal{U} \quad (1.28h)$$

$$\tilde{\mu}_b = \mu, \forall b \in \mathcal{B} \quad (1.28i)$$

$$\tilde{\mu}'_u = \mu, \forall u \in \mathcal{U} \quad (1.28j)$$

where  $\mathbf{m}_b \triangleq \{m_{ub}\}_{u \in \mathcal{U}}$ ,  $\mathbf{t}_b \triangleq \{t_{ub}\}_{u \in \mathcal{U}}$ ,  $\mathbf{t}'_u \triangleq \{t'_{ub}\}_{b \in \mathcal{B}}$ , and  $R_{ub}(\tilde{\boldsymbol{\rho}}_b, \tilde{\boldsymbol{\beta}}_b) \triangleq a_b \tilde{\beta}_{ub} \log \left( 1 + \frac{g_{ub} \tilde{\rho}_{ub}}{\beta_{ub}} \right)$ . The purpose of introducing  $\tilde{\boldsymbol{\mu}} \triangleq \{\tilde{\mu}_b\}$  is to make the constraints in (1.28c) and (1.28d) handled *locally* at the BSs. Similarly, introducing  $\tilde{\boldsymbol{\mu}}' \triangleq \{\tilde{\mu}'_u\}$  is to make the constraints in (1.28f) handled *locally* at the users. The global variables  $\{v_{ub}\}$  are introduced to keep the corresponding local versions of data rate at the BSs and the users to be equal.

For the ease of presentation, let  $\boldsymbol{\theta}_b \triangleq [\tilde{\mu}_b, \mathbf{t}_b^T]^T$ ,  $\boldsymbol{\theta}'_u \triangleq [\tilde{\mu}'_u, \mathbf{t}'_u^T]^T$ ,  $\mathbf{v}_b \triangleq \{v_{ub}\}_{u \in \mathcal{U}}$ ,  $\mathbf{v}'_u \triangleq \{v_{ub}\}_{b \in \mathcal{B}}$ ,  $\mathbf{z}_b \triangleq [\mu, \mathbf{v}_b^T]^T$ , and  $\mathbf{z}'_u \triangleq [\mu, \mathbf{v}'_u^T]^T$ . We note that  $\{\mathbf{z}_b\}$  and  $\{\mathbf{z}'_u\}$  are derived from *the same set* of variables  $\{v_{ub}\}$  and  $\mu$ . For each

BS  $b$  we define a local feasible set  $\mathcal{K}_b$  as

$$\mathcal{K}_b = \left\{ \left( \tilde{\mu}_b, \tilde{\boldsymbol{\rho}}_b, \tilde{\boldsymbol{\beta}}_b, \mathbf{t}_b \right) \left| \mathbf{1}^T \tilde{\boldsymbol{\beta}}_b \leq \tilde{\mu}_b \bar{W}_b, \mathbf{1}^T \tilde{\boldsymbol{\rho}}_b \leq \tilde{\mu}_b \bar{P}_b, R_{ub} \left( \tilde{\boldsymbol{\rho}}_b, \tilde{\boldsymbol{\beta}}_b \right) \geq t_{ub}, \forall u \in \mathcal{U} \right. \right\} \quad (1.29)$$

Similarly, for each user  $u$  we define a local feasible set  $\mathcal{K}'_u$  as

$$\mathcal{K}'_u = \left\{ \left( \tilde{\mu}'_u, \mathbf{t}'_u \right) \left| \mathbf{1}^T \mathbf{t}'_u \geq \tilde{\mu}'_u \bar{R}_u, \right. \right\}. \quad (1.30)$$

Then, we can rewrite (1.28) in a more compact form as follows

$$\begin{aligned} & \underset{\substack{\{\tilde{\boldsymbol{\rho}}_b\}, \{\tilde{\boldsymbol{\beta}}_b\}, \mu > 0 \\ \{\mathbf{t}_b\}, \{\mathbf{t}'_u\}, \mathbf{v} \geq 0, \tilde{\mu}_b, \tilde{\mu}'_u, \mathbf{x}}}{\text{minimize}} & - \sum_{b \in \mathcal{B}} \mathbf{m}_b^T \mathbf{t}_b \end{aligned} \quad (1.31a)$$

$$\text{subject to } \mathbf{g}^T \boldsymbol{\theta}_b - \frac{1}{B} = x_b, \forall b \in \mathcal{B} \quad (1.31b)$$

$$\sum_{b \in \mathcal{B}} x_b = 0 \quad (1.31c)$$

$$\boldsymbol{\theta}_b = \mathbf{z}_b, \forall b \in \mathcal{B} \quad (1.31d)$$

$$\boldsymbol{\theta}'_u = \mathbf{z}'_u, \forall u \in \mathcal{U} \quad (1.31e)$$

$$\left( \tilde{\mu}_b, \tilde{\boldsymbol{\rho}}_b, \tilde{\boldsymbol{\beta}}_b, \mathbf{t}_b \right) \in \mathcal{K}_b, \forall b \in \mathcal{B} \quad (1.31f)$$

$$\left( \tilde{\mu}'_u, \mathbf{t}'_u \right) \in \mathcal{K}'_u, \forall u \in \mathcal{U}. \quad (1.31g)$$

where  $\mathbf{g} \triangleq \left[ \frac{p^c}{B}, \frac{1}{\lambda_b} \mathbf{1}^T, \mathbf{0}^T \right]^T$ . Here (1.28b) is equivalently rewritten into the two constraints (1.31b) and (1.31c) due to the equality in (1.28i).

We now form the partial augmented Lagrangian function of (1.31) given

by

$$\begin{aligned}
L_A(\boldsymbol{\theta}, \boldsymbol{\theta}', \mathbf{z}, \mathbf{x}; \boldsymbol{\tau}, \boldsymbol{\zeta}, \boldsymbol{\zeta}', \xi) = & - \sum_{b \in \mathcal{B}} \mathbf{m}_b^T \mathbf{t}_b \\
& + \sum_{b \in \mathcal{B}} \left( \tau_b \left( \mathbf{g}^T \boldsymbol{\theta}_b - x_b - \frac{1}{B} \right) + \frac{d}{2} \left( \mathbf{g}^T \boldsymbol{\theta}_b - x_b - \frac{1}{B} \right)^2 + \boldsymbol{\zeta}_b^T (\boldsymbol{\theta}_b - \mathbf{z}_b) + \frac{d}{2} \|\boldsymbol{\theta}_b - \mathbf{z}_b\|^2 \right) \\
& + \sum_{u \in \mathcal{U}} \left( \boldsymbol{\zeta}'_u{}^T (\boldsymbol{\theta}'_u - \mathbf{z}'_u) + \frac{d}{2} \|\boldsymbol{\theta}'_u - \mathbf{z}'_u\|^2 \right) + \xi \sum_{b \in \mathcal{B}} x_b + \frac{d}{2} \left( \sum_{b \in \mathcal{B}} x_b \right)^2 \quad (1.32)
\end{aligned}$$

where  $d > 0$  is the penalty parameter and  $\xi, \boldsymbol{\tau} \triangleq \{\tau_b\}_{b \in \mathcal{B}}, \boldsymbol{\zeta} \triangleq \{\boldsymbol{\zeta}_b\}_{b \in \mathcal{B}}, \boldsymbol{\zeta}' \triangleq \{\boldsymbol{\zeta}'_u\}_{u \in \mathcal{U}}$  are the Lagrangian multipliers.

In what follows, we detail the variable update at iteration  $(l+1)$  based on the ADMM to solve (1.31). First  $\{\boldsymbol{\theta}_b\}_{b \in \mathcal{B}}$  and  $\{\boldsymbol{\theta}'_u\}_{u \in \mathcal{U}}$  are updated as

$$\left( \boldsymbol{\theta}^{(l+1)}, \boldsymbol{\theta}'^{(l+1)} \right) = \arg \min_{\boldsymbol{\theta}, \boldsymbol{\theta}'} L_A \left( \boldsymbol{\theta}, \boldsymbol{\theta}', \mathbf{z}^{(l)}, \mathbf{x}^{(l)}; \boldsymbol{\tau}^{(l)}, \boldsymbol{\zeta}^{(l)}, \boldsymbol{\zeta}'^{(l)}, \xi^{(l)} \right). \quad (1.33)$$

The update in (1.33) can be carried out independently at each BS and user.

Particularly, BS  $b$  solves the following convex subproblem

$$\begin{aligned}
& \underset{(\tilde{\mu}_b, \tilde{\rho}_b, \tilde{\beta}_b, \mathbf{t}_b) \in \mathcal{K}_b}{\text{minimize}} \quad - \mathbf{m}_b^T \mathbf{t}_b + \tau_b^{(l)} \left( \mathbf{g}^T \boldsymbol{\theta}_b - x_b^{(l)} - \frac{1}{B} \right) \\
& \quad + \frac{d}{2} \left( \mathbf{g}^T \boldsymbol{\theta}_b - x_b^{(l)} - \frac{1}{B} \right)^2 + \boldsymbol{\zeta}_b^{(l)T} (\boldsymbol{\theta}_b - \mathbf{z}_b^{(l)}) + \frac{d}{2} \|\boldsymbol{\theta}_b - \mathbf{z}_b^{(l)}\|^2 \quad (1.34)
\end{aligned}$$

and user  $u$  solves the following quadratic program

$$\underset{(\tilde{\mu}'_u, \mathbf{t}'_u) \in \mathcal{K}'_u}{\text{minimize}} \quad \boldsymbol{\zeta}'_u{}^{(l)T} (\boldsymbol{\theta}'_u - \mathbf{z}'_u) + \frac{d}{2} \|\boldsymbol{\theta}'_u - \mathbf{z}'_u\|^2. \quad (1.35)$$

Next, the global variables  $\mathbf{x}$  and  $\mathbf{z}$  are updated as

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \xi^{(l)} \sum_{b \in \mathcal{B}} x_b + \frac{d}{2} \left( \sum_{b \in \mathcal{B}} x_b \right)^2 \\ & + \sum_{b \in \mathcal{B}} \left( \tau_b^{(l)} \left( \mathbf{g}^T \boldsymbol{\theta}_b^{(l+1)} - x_b - \frac{1}{B} \right) + \frac{d}{2} \left( \mathbf{g}^T \boldsymbol{\theta}_b^{(l+1)} - x_b - \frac{1}{B} \right)^2 \right) \end{aligned} \quad (1.36)$$

$$\begin{aligned} \underset{\mathbf{z}}{\text{minimize}} \quad & \sum_{b \in \mathcal{B}} \left( \boldsymbol{\zeta}_b^{(l)T} \left( \boldsymbol{\theta}_b^{(l+1)} - \mathbf{z}_b \right) + \frac{d}{2} \left\| \boldsymbol{\theta}_b^{(l+1)} - \mathbf{z}_b \right\|_2^2 \right) \\ & + \sum_{u \in \mathcal{U}} \left( \boldsymbol{\zeta}'_u{}^{(l)T} \left( \boldsymbol{\theta}'_u{}^{(l+1)} - \mathbf{z}'_u \right) + \frac{d}{2} \left\| \boldsymbol{\theta}'_u{}^{(l+1)} - \mathbf{z}'_u \right\|_2^2 \right). \end{aligned} \quad (1.37)$$

Problem (1.36) admits a closed form solution given as

$$x_b^{(l+1)} := X_b^{(l)} - \frac{B \bar{X}^{(l)}}{B+1} \quad (1.38)$$

where  $X_b^{(l)} = \mathbf{g}^T \boldsymbol{\theta}_b^{(l+1)} - \frac{1}{B} - \frac{\xi^{(l)} - \tau_b^{(l)}}{d}$  and  $\bar{X}^{(l)} = \frac{\sum_{b \in \mathcal{B}} X_b^{(l)}}{B}$ . Thus, for updating  $x_b^{(l+1)}$ , we can perform an average consensus algorithm among the BSs to compute  $\bar{X}^{(l)}$  [25]. We turn to updating  $\mathbf{z}$  in (1.37). For decomposing (1.37), let us define  $\zeta_{b,\mu} \triangleq [1 \ \mathbf{0}] \boldsymbol{\zeta}_b$ ,  $\zeta'_{u,\mu} \triangleq [1 \ \mathbf{0}] \boldsymbol{\zeta}'_u$ ,  $\zeta_{b,\mathbf{t}} \triangleq [\mathbf{0} \ \mathbf{I}_{|\zeta_b-1|}] \boldsymbol{\zeta}_b$ . Then (1.37) is rewritten as

$$\begin{aligned}
& \underset{(\mu, \mathbf{v}_b)}{\text{minimize}} \sum_{b \in \mathcal{B}} \zeta_{b,\mu}^{(l)} \left( \mu_b^{(l+1)} - \mu \right) + \frac{d}{2} \left\| \mu_b^{(l+1)} - \mu \right\|_2^2 + \zeta_{b,\mathbf{t}}^{(l)T} \left( \mathbf{t}_b^{(l+1)} - \mathbf{v}_b \right) \\
& + \frac{d}{2} \left\| \mathbf{t}_b^{(l+1)} - \mathbf{v}_b \right\|_2^2 + (\tilde{\zeta}'_b)^{(l)T} \left( (\tilde{\mathbf{t}}'_b)^{(l+1)} - \mathbf{v}_b \right) + \frac{d}{2} \left\| (\tilde{\mathbf{t}}'_b)^{(l+1)} - \mathbf{v}_b \right\|_2^2 \\
& \sum_{u \in \mathcal{U}} \zeta'_{u,\mu}{}^{(l)} \left( \mu'_u{}^{(l+1)} - \mu \right) + \frac{d}{2} \left\| \mu'_u{}^{(l+1)} - \mu \right\|_2^2 \quad (1.39)
\end{aligned}$$

where  $\tilde{\mathbf{t}}'_b \triangleq \{t'_{ub}\}_{u \in \mathcal{B}}$  and  $\tilde{\zeta}'_b$  are the corresponding elements in  $\{\zeta'_u\}_u$ . The objective in (1.39) is a quadratic function, then the update is simple as

$$\mathbf{v}_b^{(l+1)} := 0.5 \left( \mathbf{t}_b^{(l+1)} + (\tilde{\mathbf{t}}'_b)^{(l+1)} + \frac{\zeta_{b,\mathbf{t}}^{(l)} + (\tilde{\zeta}'_b)^{(l)}}{d} \right) \quad (1.40)$$

$$\mu^{(l+1)} := \frac{\sum_{b \in \mathcal{B}} \phi_b^{(l)} + \sum_{u \in \mathcal{U}} \phi'_u{}^{(l)}}{d(B + U)} \quad (1.41)$$

where  $\phi_b^{(l)} \triangleq \left( \zeta_{b,\mu}^{(l)} + c\mu_b^{(l+1)} \right)$  and  $\phi'_u{}^{(l)} \triangleq \left( \zeta'_{u,\mu}{}^{(l)} + d\mu'_u{}^{(l+1)} \right)$ . Updating  $\mathbf{z}$  can be done independently at each BS as follows. BS  $b$  receives information from its connected users and forms  $(\tilde{\mathbf{t}}'_b)^{(l+1)}$  and  $(\tilde{\zeta}'_b)^{(l)}$  to update  $\mathbf{v}_b^{(l+1)}$ . The update of  $\mu$  can be implemented by means of consensus among all BSs and users by exchanging  $\phi_b, \phi'_u$ . After a consensus is reached,  $\mu^{(l+1)}$  is recovered by dividing the consensus value by  $d$ . After the  $\mathbf{z}$ -update finished at all BSs, the users gather the required information to form  $\mathbf{z}'^{(l+1)}$ .



---

**Algorithm 1.4** Decentralized algorithm solving (1.26)

---

**Initialization:** Set  $l = 0$  and choose initial values for  $\mathbf{x}^{(0)}, \mathbf{z}^{(0)}, \mathbf{z}'^{(0)}, \xi^{(0)}, \tau^{(0)}, \zeta^{(0)}, \zeta'^{(0)}$ .

**repeat**

**for**  $b \in \mathcal{B}$  **and**  $u \in \mathcal{U}$  **do**

    BS  $b$  updates  $\theta_b$  using (1.34); user  $u$  updates  $\theta'_u$  using (1.35).

    BS  $b$  calculates  $X_b$ , exchanges this value with other BSs to compute  $\bar{X}^{(l)}$ , and then updates  $x_b$  as in (1.38).

    BS  $b$  receives  $\theta'_{u'}, \zeta'_{u'}$  from user  $u'$  for all  $u' \in \mathcal{U}$ , and forms  $(\hat{\mathbf{t}}_b^{(l+1)}, (\tilde{\zeta}'_b)^{(l)})$  and  $\phi'_{u'}$ . Then BS  $b$  exchanges  $\phi_b, \phi'_u$  with other BSs.

    BS  $b$  updates  $\mathbf{v}_b$  using (1.40), and  $\mu$  using (1.41).

    User  $u$  receives  $v_{ub'}$  and  $\mu$  from BS  $b'$  for all  $b' \in \mathcal{B}$  to form  $\mathbf{z}'_u$ .

    BS  $b$  updates  $\xi, \tau_b, \zeta_b$  while user  $u$  updates  $\zeta'_u$ .

**end for**

$l := l + 1$ .

**until** Convergence

---

The last step of the ADMM is to update the Lagrangian multipliers as

$$\xi^{(l+1)} = \xi^{(l)} + \frac{dB\bar{X}^{(l)}}{B+1} \quad (1.42)$$

$$\tau_b^{(l+1)} = \tau_b^{(l)} + d \left( \mathbf{g}^T \theta_b^{(l+1)} - x_b^{(l+1)} - \frac{1}{B} \right) \quad (1.43)$$

$$\zeta_b^{(l+1)} = \zeta_b^{(l)} + d \left( \theta_b^{(l+1)} - \mathbf{z}_b^{(l+1)} \right) \quad (1.44)$$

$$\zeta'_u{}^{(l+1)} = \zeta'_u{}^{(l)} + d \left( \theta'_u{}^{(l+1)} - \mathbf{z}'_u{}^{(l+1)} \right). \quad (1.45)$$

Since  $\bar{X}^{(l)}$  is available at all BSs,  $\xi$  can be updated without requiring extra gathered information. Updating  $\tau_b$  and  $\zeta_b$  is carried out at BS  $b$ , while user  $u$  updates  $\zeta'_u$ . In summary, the decentralized procedure is outlined in Algorithm 1.4.

### 1.3.1.5. Numerical Examples

We consider a simple simulation model as follows. In the coverage area with radius 300m centered at origin, there are four BSs ( $B = 4$ ) placed at (200m,0),

(0,200m), (-200m,0) and (0,-200m). There are  $U = 7$  users whose positions are randomly generated inside the area. We use the modified Okumura-Hata urban model for path loss which is given as [7]

$$PL(\text{dB}) = \begin{cases} 122 + 38 \log(x) & \text{if } x \geq 0.05 \text{ km,} \\ 122 + 38 \log(0.05) & \text{otherwise} \end{cases} \quad (1.46)$$

where  $x$  is the distance in kilometers. The standard deviation of lognormal distribution shadowing is 8 dB. The noise power density is  $N_0 = -174$  dBm/Hz. The maximum bandwidth at the BSs are  $\bar{W}_1 = 2.4$  MHz,  $\bar{W}_2 = 1.2$  MHz,  $\bar{W}_3 = 1.8$  MHz and  $\bar{W}_4 = 2.5$  MHz. The maximum transmission power at the BSs are  $\bar{P}_1 = 35$  dBm,  $\bar{P}_2 = 32$  dBm,  $\bar{P}_3 = 26$  dBm and  $\bar{P}_4 = 37$  dBm. The power amplifier's efficiency at the BSs are  $\lambda_1 = 0.95$ ,  $\lambda_2 = 0.91$ ,  $\lambda_3 = 0.88$  and  $\lambda_4 = 0.9$ . Without loss of generality, we set  $a_b = 1, \forall b \in \mathcal{B}$ , and  $m_{ub} = 1, \forall u \in \mathcal{U}, b \in \mathcal{B}$ . We set  $p_u^{\text{cr}} = p^{\text{cr}}$ ,  $p_b^{\text{ct}} = p^{\text{ct}}$ , and  $p_{ub}^{\text{ca}} = p^{\text{ca}}$ ,  $\forall u \in \mathcal{U}, b \in \mathcal{B}$ . The QoSs for the users are  $\bar{\mathbf{R}} = [2, 1.4, 1.1, 2.2, 3.1, 2.5, 3.7]$  Mbits/s where  $\bar{\mathbf{R}} \triangleq \{\bar{R}_u\}$ .

Figure 1.1 plots the convergence behavior of the Dinkelbach algorithm over two random channel realizations. We can observe that, with superlinear convergence, the algorithm outputs the optimal solutions after a few iterations.

Figure 1.2 shows the convergence performance of Algorithm 1.4 over a random channel realization with two different values of penalty parameter  $d$ . Specifically, Figure 1.2(a) plots the objective function and Figure 1.2(b) plots the term  $\|\mathbf{t} - \mathbf{t}'\|_2$  which represents the difference between the local versions of data rate at the users and the BSs. The expected observation is that the iterative procedure converges to the centralized solution with all cases of  $d$ . We

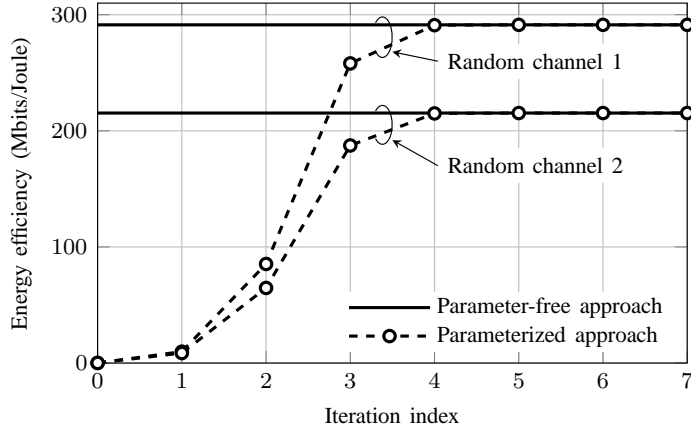


Figure 1.1: Convergence behavior of Dinkelbach's procedure for two random channel realizations. The circuit power parameters are taken as  $p^{\text{ct}} = p^{\text{cr}} = 10$  dBm and  $p^{\text{ca}} = 5$  dBm.

can also see that the convergence rate depends on  $d$ .

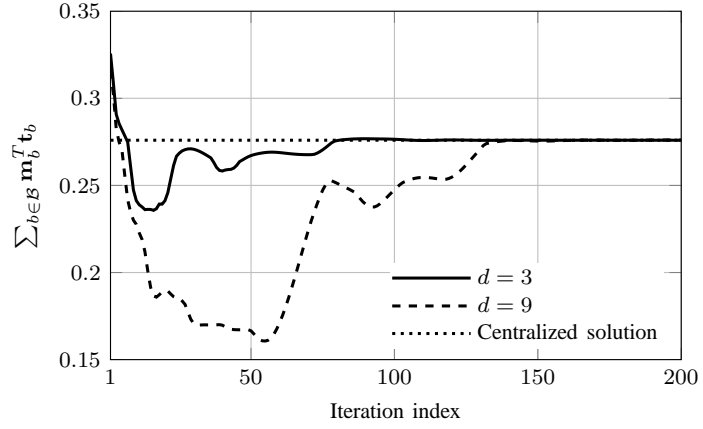
We study the average EE performance in Figure 1.3. For comparison purpose, we also consider the schemes sum rate maximization (SRmax) and power minimization (Powermin) which are mathematically stated as

$$\text{SRmax} \triangleq \max \left\{ \sum_{b \in \mathcal{B}} \sum_{u \in \mathcal{U}} m_{ub} r_{ub} \mid (1.23\text{b}), (1.23\text{c}), (1.23\text{d}) \right\} \quad (1.47)$$

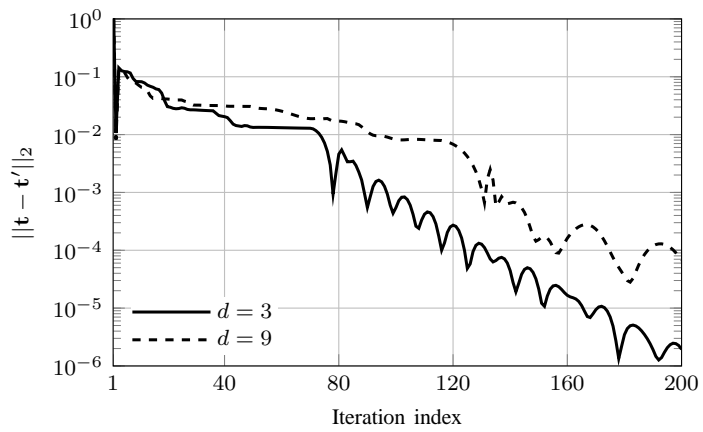
and

$$\text{Powermin} \triangleq \min \left\{ \sum_{b \in \mathcal{B}} \frac{\sum_{u \in \mathcal{U}} \rho_{ub}}{\lambda_b} \mid (1.23\text{b}), (1.23\text{c}), (1.23\text{d}) \right\}. \quad (1.48)$$

As expected, EEmax always outperforms the others in terms of EE. The other observation is that the EE performances of all schemes reduce when  $p^c$  increases. For SRmax and Powermin,  $p^c$  has no impact on the optimal solution of (1.47) and (1.48); thus when  $p^c$  increases, the total consumption powers increase while the sum rates do not change.



(a) Convergence on objective value.



(b) The gap between local versions of data rate at the BSs and the users.

Figure 1.2: Convergence behavior of the ADMM procedure over a random channel. The circuit power parameters are taken as  $p^{\text{ct}} = p^{\text{cr}} = 10$  dBm and  $p^{\text{ca}} = 5$  dBm.

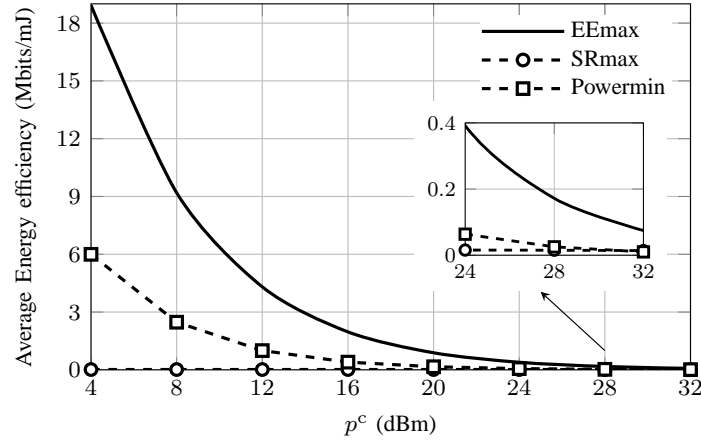


Figure 1.3: Average energy efficiency performance of EEmax, SRmax and Powermin with different total circuit power  $p^c$ .

### 1.3.2. Dense Small-Cell Networks

In this section, we apply the provided optimization tools in designing EE beamforming vectors for a dense network scenario where macro-cell and small-cell BSs cooperate in transmitting data to users. The group of BSs using the same bandwidth to simultaneously serve multiple users, i.e. interference channels, are considered. Here we are interested in noncoherent joint transmission in which the information for a specific user is encoded independently at the BSs [2]. This transmit technique requires less strict synchronization compared to the joint transmission coordinated multi-point (or coherent transmission). Thus it is more easy to implement in practice.

The EEmax problem is cast as a generalized nonconvex fractional program, thus we use SCA to locally solve the problem. Specifically, the main works are follows. We first transform the problem into a formulation amenable to the SCA. We then present how to arrive at the convex approximation subproblem solved at each iteration of the SCA procedure. Finally, we implement the solution in decentralized manner with the ADMM.

### 1.3.2.1. System Model

We consider a region covered by a macro-cell BS and a set of  $B$  small-cell BSs denoted by  $\mathcal{B} = \{1, 2, \dots, B\}$ . For notational convenience, we refer to the macro BS as BS 0, and let  $\bar{\mathcal{B}} = \mathcal{B} \cup \{0\}$ . Let  $K_b$  denote the number of antennas equipped at BS  $b$ . In the region, there is a set of  $U$  single-antenna users, denoted by  $\mathcal{U} = \{1, 2, \dots, U\}$ , simultaneously served by the BSs under the same frequency band. We assume that linear precoding is used at all BSs and the users receive information from all BSs under noncoherent transmission [2]. Particularly, let  $\mathbf{h}_{ub} \in \mathbb{C}^{1 \times K_b}$  (row vector) denote the channels between user  $u$  and BS  $b$ ; let  $x_{ub}$  and  $\mathbf{w}_{ub} \in \mathbb{C}^{K_b \times 1}$  be the normalized symbol and the beamforming vector at BS  $b$  for user  $u$ , respectively. Suppose the channels to be flat. The received signal at user  $u$  is

$$r_u = \underbrace{\mathbf{h}_{u0}\mathbf{w}_{u0}x_{u0} + \sum_{b \in \mathcal{B}} \mathbf{h}_{ub}\mathbf{w}_{ub}x_{ub}}_{\text{from small-cell BSs}} + \underbrace{\sum_{b \in \bar{\mathcal{B}}} \sum_{m \in \mathcal{U} \setminus u} \mathbf{h}_{ub}\mathbf{w}_{mb}x_{mb}}_{\text{interference}} + n_u \quad (1.49)$$

where  $n_u \sim \mathcal{N}(0, \sigma_u^2)$  is the additive white Gaussian noise at user  $u$ . We further assume that users treat signal of other users as noise. The signal-to-interference-plus-noise ratio (SINR) at user  $u$  is

$$\gamma_u = \frac{|\mathbf{h}_{u0}\mathbf{w}_{u0}|^2 + \sum_{b \in \mathcal{B}} |\mathbf{h}_{ub}\mathbf{w}_{ub}|^2}{\sum_{b \in \bar{\mathcal{B}}} \sum_{m \in \mathcal{U} \setminus u} |\mathbf{h}_{ub}\mathbf{w}_{mb}|^2 + \sigma_u^2}. \quad (1.50)$$

We note that (1.50) is due to the noncoherent transmission which does not require phase synchronization between the BSs. For notational convenience, let  $\mathbf{w}_u \triangleq [\mathbf{w}_{u0}^T, \mathbf{w}_{u1}^T, \dots, \mathbf{w}_{uB}^T]^T \in \mathbb{C}^{\sum_{b \in \bar{\mathcal{B}}} K_b \times 1}$  and  $\bar{\mathbf{H}}_u \triangleq \text{blkdiag}\{\mathbf{H}_{u0}, \mathbf{H}_{u1}, \dots, \mathbf{H}_{uB}\}$

where  $\mathbf{H}_{ub} \triangleq \mathbf{h}_{ub}^H \mathbf{h}_{ub}$ . Then the SINR can be rewritten as

$$\gamma_u = \frac{\mathbf{w}_u^H \bar{\mathbf{H}}_u \mathbf{w}_u}{\sum_{m \in \mathcal{U} \setminus u} \mathbf{w}_m^H \bar{\mathbf{H}}_u \mathbf{w}_m + \sigma_u^2}. \quad (1.51)$$

The power consumption model similar to that in Section 1.3.1 is considered. Specifically, let us denote by  $p_b^{\text{cirTx}}$  and  $p_u^{\text{cirRx}}$  the circuit power of the idle mode of BS  $b$  and user  $u$ , respectively. Also, let  $p_{ub}^{\text{cirCo}}$  denote extra circuit consumed power when BS  $b$  transmits data to user  $u$ . Then, the total circuit power is

$$P^{\text{cir}} = \sum_{b \in \bar{\mathcal{B}}} p_b^{\text{cirTx}} + \sum_{u \in \mathcal{U}} p_u^{\text{cirRx}} + \sum_{b \in \bar{\mathcal{B}}} \sum_{u \in \mathcal{U}} p_{ub}^{\text{cirCo}}. \quad (1.52)$$

The total consumption power is

$$P^{\text{Total}} = \sum_{b \in \bar{\mathcal{B}}} \frac{1}{\lambda_b} \sum_{u \in \mathcal{U}} \mathbf{w}_u^H \mathbf{E}_b \mathbf{w}_u + P^{\text{cir}} \quad (1.53)$$

where  $\lambda_b \in (0, 1)$  is the amplifier's efficiency at BS  $b$ , and  $\mathbf{E}_b = \text{blkdiag}\{\mathbf{0}_{\sum_{b'=0}^{b-1} K_{b'}}, \mathbf{I}_{K_b}, \mathbf{0}_{\sum_{b'=b+1}^B K_{b'}}\}$ .

The aim is to design beamforming vectors  $\{\mathbf{w}_u\}_u$  so that the overall EE is maximized under the constraints of transmit power budget at the BSs and QoS for the users. Mathematically, the interest problem reads

$$\underset{\{\mathbf{w}_u\}}{\text{maximize}} \quad \frac{\sum_{u \in \mathcal{U}} a_u \log(1 + \gamma_u)}{P^{\text{Total}}} \quad (1.54a)$$

$$\text{subject to } \log(1 + \gamma_u) \geq \bar{R}_u, \forall u \in \mathcal{U}, \quad (1.54b)$$

$$\sum_{u \in \mathcal{U}} \mathbf{w}_u^H \mathbf{E}_b \mathbf{w}_u \leq P_b, \forall b \in \mathcal{B}, \quad (1.54c)$$

where coefficient  $a_u$  represents the priority of user  $u$ ,  $\bar{R}_u$  is the predefined data

rate threshold representing the QoS for user  $u$ , and  $P_b$  is the maximum total transmit power at BS  $b$ . Problem (1.54) is a generalized nonconvex fractional program due to the SINR terms  $\{\gamma_k\}$ .

### 1.3.2.2. Centralized Solution via Successive Convex Approximation

In what follows, we present how to locally solve (1.54) using the SCA framework. First, we transform the problem into an equivalent formulation as follows

$$\begin{aligned} & \underset{\{\mathbf{w}_u\}, \{t_u\}, \{\mu_u\}, \{\pi_b\}}{\text{maximize}} \quad \frac{\sum_{u \in \mathcal{U}} a_u t_u^2}{\sum_{b \in \bar{\mathcal{B}}} \frac{\pi_b}{\lambda_b} + P_{\text{cir}}} \end{aligned} \quad (1.55a)$$

$$\text{subject to } \frac{\mathbf{w}_u^H \bar{\mathbf{H}}_u \mathbf{w}_u}{\sum_{m \in \mathcal{U} \setminus u} \mathbf{w}_m^H \bar{\mathbf{H}}_u \mathbf{w}_m + \sigma_u^2} \geq \mu_u, \forall u \in \mathcal{U}, \quad (1.55b)$$

$$\log(1 + \mu_u) \geq t_u^2, \forall u \in \mathcal{U}, \quad (1.55c)$$

$$t_u \geq \sqrt{\bar{R}_u}, \forall u \in \mathcal{U}, \quad (1.55d)$$

$$\sum_{u \in \mathcal{U}} \mathbf{w}_u^H \mathbf{E}_b \mathbf{w}_u \leq \pi_b, \forall b \in \bar{\mathcal{B}} \quad (1.55e)$$

$$\pi_b \leq P_b, \forall b \in \bar{\mathcal{B}}, \quad (1.55f)$$

where  $\{t_u\}$ ,  $\{\mu_u\}$  and  $\{\pi_b\}$  are newly introduced slack variables. Problem (1.55) is equivalent to (1.54) in sense of the optimal set; it is easy to justify that the constraints in (1.55b), (1.55c) and (1.55e) are active at the optimal points. We can see that the nonconvex parts lie in (1.55a) and (1.55b).

The objective in (1.55) is a quadratic-over-linear function which is convex. A common approach arriving at a concave approximate of the function is to



use the first-order Taylor series. Particularly, let  $(\{\mathbf{w}_u^{(l)}\}, \{t_u^{(l)}\}, \{\mu_u^{(l)}\}, \{\pi_b^{(l)}\})$  be a feasible point of (1.55). A concave approximate of the objective function is

$$\sum_{u \in \mathcal{U}} a_u \left( \frac{2t_u^{(l)}}{\left(\sum_{b \in \bar{\mathcal{B}}} \frac{\pi_b^{(l)}}{\lambda_b} + P^{\text{cir}}\right)} t_u - \frac{(t_u^{(l)})^2}{\left(\sum_{b \in \bar{\mathcal{B}}} \frac{\pi_b^{(l)}}{\lambda_b} + P^{\text{cir}}\right)^2} \left(\sum_{b \in \bar{\mathcal{B}}} \frac{\pi_b}{\lambda_b}\right) \right). \quad (1.56)$$

We now turn to the nonconvex constraints in (1.55b) which can be rewritten for each  $u$  as

$$\frac{\mathbf{w}_u^H \bar{\mathbf{H}}_u \mathbf{w}_u}{\mu_u} \geq \sum_{m \in \mathcal{U} \setminus u} \mathbf{w}_m^H \bar{\mathbf{H}}_u \mathbf{w}_m + \sigma_u^2. \quad (1.57)$$

Again, the left side of the inequality is a quadratic-over-linear function. Then, an approximate of (1.55b) is

$$\frac{2\text{Re}\{(\mathbf{w}_u^{(l)})^H \bar{\mathbf{H}}_u \mathbf{w}_u\}}{\mu_u^{(l)}} - \frac{(\mathbf{w}_u^{(l)})^H \bar{\mathbf{H}}_u \mathbf{w}_u^{(l)}}{(\mu_u^{(l)})^2} \mu_u \geq \sum_{m \in \mathcal{U} \setminus u} \mathbf{w}_m^H \bar{\mathbf{H}}_u \mathbf{w}_m + \sigma_u^2. \quad (1.58)$$

where  $\text{Re}\{\cdot\}$  denotes the real part. It is easy to justify that (1.56) and (1.58) satisfy the three conditions posted in (1.12). Finally, the convex approximation

subproblem solved at the  $(l + 1)$ th iteration of the SCA procedure is

$$\begin{aligned} & \underset{\{\mathbf{w}_u\}, \{t_u\}, \{\mu_u\}, \{\pi_b\}}{\text{maximize}} && \sum_{u \in \mathcal{U}} A_u^{(l)} t_u - \sum_{b \in \bar{\mathcal{B}}} B_b^{(l)} \pi_b \end{aligned} \quad (1.59a)$$

$$\begin{aligned} & \text{subject to} && \frac{2\text{Re}\{(\mathbf{w}_u^{(l)})^H \bar{\mathbf{H}}_u \mathbf{w}_u^{(l)}\}}{\mu_u^{(l)}} - \frac{(\mathbf{w}_u^{(l)})^H \bar{\mathbf{H}}_u \mathbf{w}_u^{(l)}}{(\mu_u^{(l)})^2} \mu_u \\ & && \geq \sum_{m \in \mathcal{U} \setminus u} \mathbf{w}_m^H \bar{\mathbf{H}}_u \mathbf{w}_m + \sigma_u^2, \forall u \in \mathcal{U}, \end{aligned} \quad (1.59b)$$

$$\log(1 + \mu_u) \geq t_u^2, \forall u \in \mathcal{U}, \quad (1.59c)$$

$$t_u \geq \sqrt{\bar{R}_u}, \forall u \in \mathcal{U}, \quad (1.59d)$$

$$\sum_{u \in \mathcal{U}} \mathbf{w}_u^H \mathbf{E}_b \mathbf{w}_u \leq \pi_b, \forall b \in \bar{\mathcal{B}}, \quad (1.59e)$$

$$\pi_b \leq P_b, \forall b \in \bar{\mathcal{B}}, \quad (1.59f)$$

$$\text{where } A_u^{(l)} = \frac{2a_u t_u^{(l)}}{\left(\sum_{b \in \bar{\mathcal{B}}} \frac{\pi_b^{(l)}}{\lambda_b} + P^{\text{cir}}\right)} \text{ and } B_b^{(l)} = \left( \frac{\sum_{u \in \mathcal{U}} a_u (t_u^{(l)})^2}{\lambda_b \left(\sum_{b \in \bar{\mathcal{B}}} \frac{\pi_b^{(l)}}{\lambda_b} + P^{\text{cir}}\right)^2} \right).$$

A feasible point is required to start the SCA algorithm. For (1.55), this requirement might be a challenge due to the QoS constraints. Here, we show a simple heuristic method to overcome this practical issue. By introducing nonnegative slack variables  $\{\varphi_u\}$ , we arrive at a regularization formulation of (1.55) given as

$$\begin{aligned} & \underset{\{\mathbf{w}_u\}, \{t_u\}, \{\mu_u\}, \{\pi_b\}, \{\varphi_u\}}{\text{maximize}} && \frac{\sum_{u \in \mathcal{U}} a_u t_u^2}{\sum_{b \in \bar{\mathcal{B}}} \frac{\pi_b}{\lambda_b} + P^{\text{cir}}} - \beta \sum_{u \in \mathcal{U}} \varphi_u \end{aligned} \quad (1.60a)$$

$$\text{subject to } t_u + \varphi_u \geq \sqrt{\bar{R}_u}, \forall u \in \mathcal{U}, \quad (1.60b)$$

$$(1.55b), (1.55c), (1.55e), (1.55f) \quad (1.60c)$$

where parameter  $\beta > 0$ . A feasible point of (1.60) is easy to find as follows. We first randomly generate beamforming vectors  $\{\mathbf{w}_u^{(0)}\}_u$  so that (1.54c) is satisfied. Then,  $\mu_u^{(0)}$ ,  $t_u^{(0)}$ , and  $\pi_b^{(0)}$  are determined according to (1.55b), (1.55c), and (1.55e), respectively. Constraint (1.60b) is automatically satisfied with sufficient large  $\varphi_u^{(0)}$ . By using the SCA to solve 1.60, it is expected that  $\varphi_u \rightarrow 0$  for all  $u$  due to the term  $\beta \sum_{u \in \mathcal{U}} \varphi_u$  at the objective. Once  $\varphi_u = 0$  for all  $u$ , we yield a feasible point of (1.55) which could be use to initialize the SCA algorithm solving (1.55).

### 1.3.2.3. Distributed Implementation

We now implement the SCA solution in decentralized manner. The core idea is to use ADMM to solve the convex approximation subproblem (1.59). For the

purpose, we first equivalently rewrite (1.59) as

$$\begin{aligned} & \underset{\substack{\{\mathbf{w}_{ub}\}, \{t_u\}, \{\mu_u\}, \\ \{\pi_b\}, \{y_{ub}\}, \{\tilde{y}_{ub}\}, \\ \{\tilde{\mu}_{ub}\}, \{s_{ub}\}, \{z_u\}}}{\text{maximize}} \quad \sum_{u \in \mathcal{U}} A_u^{(l)} t_u - \sum_{b \in \bar{\mathcal{B}}} B_b^{(l)} \pi_b \end{aligned} \quad (1.61a)$$

$$\text{subject to } f_{ub}^{(l)}(\{\mathbf{w}_{ub}\}_u, \tilde{\mu}_{ub}; \mathbf{w}_{ub}^{(l)}, \mu_u^{(l)}) \geq \tilde{y}_{ub}, \forall u \in \mathcal{U}, b \in \bar{\mathcal{B}} \quad (1.61b)$$

$$\sum_{b \in \bar{\mathcal{B}}} y_{ub} - \sigma_u^2 \geq 0, \forall u \in \mathcal{U}, \quad (1.61c)$$

$$\log(1 + \mu_u) \geq t_u^2, \forall u \in \mathcal{U}, \quad (1.61d)$$

$$t_u \geq \sqrt{\bar{R}_u}, \forall u \in \mathcal{U}, \quad (1.61e)$$

$$\sum_{u \in \mathcal{U}} \mathbf{w}_{ub}^H \mathbf{w}_{ub} \leq \pi_b, \forall b \in \bar{\mathcal{B}}, \quad (1.61f)$$

$$\pi_b \leq P_b, \forall b \in \bar{\mathcal{B}}, \quad (1.61g)$$

$$\tilde{y}_{ub} = s_{ub}, \forall u \in \mathcal{U}, b \in \bar{\mathcal{B}} \quad (1.61h)$$

$$y_{ub} = s_{ub}, \forall u \in \mathcal{U}, b \in \bar{\mathcal{B}} \quad (1.61i)$$

$$\tilde{\mu}_{ub} = z_u, \forall u \in \mathcal{U}, b \in \bar{\mathcal{B}} \quad (1.61j)$$

$$\mu_u = z_u, \forall u \in \mathcal{U} \quad (1.61k)$$

where  $\{y_{ub}\}$ ,  $\{\tilde{y}_{ub}\}$ ,  $\{\tilde{\mu}_{ub}\}$ ,  $\{s_{ub}\}$ , and  $\{z_u\}$  are newly introduced variables;

$$f_{ub}^{(l)}(\{\mathbf{w}_{ub}\}_u, \tilde{\mu}_{ub}; \mathbf{w}_{ub}^{(l)}, \mu_u^{(l)}) \triangleq \frac{2\text{Re}\{(\mathbf{w}_{ub}^{(l)})^H \mathbf{H}_{ub} \mathbf{w}_{ub}\}}{\mu_u^{(l)}} - \frac{(\mathbf{w}_{ub}^{(l)})^H \mathbf{H}_{ub} \mathbf{w}_{ub}^{(l)}}{(\mu_u^{(l)})^2} \tilde{\mu}_{ub}$$

$-\sum_{m \in \mathcal{U} \setminus u} \mathbf{w}_{mb}^H \mathbf{H}_{ub} \mathbf{w}_{mb}$ . The introduction of  $\{y_{ub}\}$ ,  $\{\tilde{y}_{ub}\}$  and  $\{\tilde{\mu}_{ub}\}$  is to de-

compose (1.59b) into (1.61b) and (1.61c) which will be handled locally at BS

$b$  and user  $u$ , respectively. The introduction of global variables  $\{s_{ub}\}$  and  $\{z_u\}$

is to guarantee the corresponding local versions of the variables are equal with

the others via the constraints in (1.61h)–(1.61k). Particularly, problem (1.61)

can be handled distributively as follows.

For each user  $u \in \mathcal{U}$ , let us denote by  $\mathbf{v}_u \triangleq \{t_u, \mu_u, \{y_{ub}\}_b\}$  the local

variables, and define the local feasible set as

$$\mathcal{K}_u = \{\mathbf{v}_u | \sum_{b \in \bar{\mathcal{B}}} y_{ub} - \sigma_u^2 \geq 0, \log(1 + \mu_u) \geq t_u^2, t_u \geq \sqrt{\bar{R}_u}\}. \quad (1.62)$$

Similarly, for each BS  $b \in \bar{\mathcal{B}}$ , let  $\tilde{\mathbf{v}}_b \triangleq (\{\mathbf{w}_{ub}\}_u, \{\tilde{\mu}_{ub}\}_u, \{\tilde{y}_{ub}\}_u, \pi_b)$  be the local variables, and the local feasible set is defined as

$$\begin{aligned} \tilde{\mathcal{K}}_b = \{ \tilde{\mathbf{v}}_b | f_{ub}^{(l)}(\{\mathbf{w}_{ub}\}_u, \tilde{\mu}_{ub}; \mathbf{w}_{ub}^{(l)}, \mu_u^{(l)}) \geq \tilde{y}_{ub} \forall u \in \mathcal{U}, \\ \sum_{u \in \mathcal{U}} \mathbf{w}_{ub}^H \mathbf{w}_{ub} \leq \pi_b \leq P_b \}. \end{aligned} \quad (1.63)$$

With these definitions, the problem (1.61) is rewritten as

$$\underset{\{\tilde{\mathbf{v}}_b\}, \{\mathbf{v}_u\}, \{s_{ub}\}, \{z_u\}}{\text{minimize}} \quad - \sum_{u \in \mathcal{U}} A_u^{(l)} t_u + \sum_{b \in \bar{\mathcal{B}}} B_b^{(l)} \pi_b \quad (1.64a)$$

$$\text{subject to } \tilde{\mathbf{v}}_b \in \tilde{\mathcal{K}}_b, \forall b \in \bar{\mathcal{B}}, \quad (1.64b)$$

$$\mathbf{v}_u \in \mathcal{K}_u, \forall u \in \mathcal{U}, \quad (1.64c)$$

$$\tilde{\boldsymbol{\theta}}_b = \tilde{\boldsymbol{\phi}}_b, \forall b \in \bar{\mathcal{B}}, \quad (1.64d)$$

$$\boldsymbol{\theta}_u = \boldsymbol{\phi}_u, \forall u \in \mathcal{U} \quad (1.64e)$$

where  $\tilde{\boldsymbol{\theta}}_b \triangleq \{\{\tilde{y}_{ub}\}_u, \{\tilde{\mu}_{ub}\}_u\}$ ,  $\boldsymbol{\theta}_u \triangleq \{\{y_{ub}\}_b, \mu_u\}$ ,  $\tilde{\boldsymbol{\phi}}_b$  and  $\boldsymbol{\phi}_u$  are the rearranged vectors from the same set of variables  $(\{s_{ub}\}, \{z_u\})$ . The augmented Lagrangian

function of (1.64) is

$$\begin{aligned}
L_A(\{\tilde{\mathbf{v}}_b\}, \{\mathbf{v}_u\}, \{s_{ub}\}, \{z_u\}; \{\boldsymbol{\xi}_u\}, \{\boldsymbol{\rho}_b\}) = \\
\sum_{u \in \mathcal{U}} \left( \boldsymbol{\xi}_u^T (\boldsymbol{\theta}_u - \boldsymbol{\phi}_u) + \frac{d}{2} \|\boldsymbol{\theta}_u - \boldsymbol{\phi}_u\|_2^2 - A_u^{(l)} t_u \right) \\
+ \sum_{b \in \bar{\mathcal{B}}} \left( B_b^{(l)} x_b + \boldsymbol{\rho}_b^T (\tilde{\boldsymbol{\theta}}_b - \tilde{\boldsymbol{\phi}}_b) + \frac{d}{2} \|\tilde{\boldsymbol{\theta}}_b - \tilde{\boldsymbol{\phi}}_b\|_2^2 \right) \quad (1.65)
\end{aligned}$$

where  $\{\boldsymbol{\xi}_u\}$  and  $\{\boldsymbol{\rho}_b\}$  are the Lagrangian multipliers and  $d$  is the penalty parameter.

In what follows, we present the variable updates at iteration  $(k + 1)$  of the ADMM procedure. The global variables  $\{s_{ub}\}$  and  $\{z_u\}$  are updated via solving the following problem extracted from (1.65)

$$\begin{aligned}
\underset{\{s_{ub}\}, \{z_u\}}{\text{minimize}} \quad & \sum_{b \in \bar{\mathcal{B}}} \sum_{u \in \mathcal{U}} \left( [\boldsymbol{\rho}_b^{(k)}]_{s_{ub}} (\tilde{y}_{ub}^{(k)} - s_{ub}) + \frac{d}{2} (\tilde{y}_{ub}^{(k)} - s_{ub})^2 \right. \\
& \left. + [\boldsymbol{\xi}_u^{(k)}]_{s_{ub}} (y_{ub}^{(k)} - s_{ub}) + \frac{d}{2} (y_{ub}^{(k)} - s_{ub})^2 \right) \\
& + \sum_{u \in \mathcal{U}} \left( \sum_{b \in \bar{\mathcal{B}}} \left( [\boldsymbol{\rho}_b^{(k)}]_{z_u} (\tilde{\mu}_{ub}^{(k)} - z_u) + \frac{d}{2} (\tilde{\mu}_{ub}^{(k)} - z_u)^2 \right) \right. \\
& \left. + [\boldsymbol{\xi}_u^{(k)}]_{z_u} (\mu_u^{(k)} - z_u) + \frac{d}{2} (\mu_u^{(k)} - z_u)^2 \right) \quad (1.66)
\end{aligned}$$

where  $[\boldsymbol{\rho}_b^{(k)}]_{s_{ub}}$  is the element in  $\boldsymbol{\rho}_b^{(k)}$  corresponding to constraint  $\tilde{y}_{ub} = s_{ub}$ ; similar definition is applied to  $[\boldsymbol{\rho}_b^{(k)}]_{z_u}$ ,  $[\boldsymbol{\xi}_u^{(k)}]_{s_{ub}}$  and  $[\boldsymbol{\xi}_u^{(k)}]_{z_u}$ . Problem (1.66) has

the close-form solution as follows

$$s_{ub}^{(k+1)} = \frac{([\boldsymbol{\rho}_b^{(k)}]_{s_{ub}} + d\tilde{y}_{ub}^{(k)}) + ([\boldsymbol{\xi}_u^{(k)}]_{s_{ub}} + dy_{ub}^{(k)})}{2d} \quad (1.67)$$

$$z_u^{(k+1)} = \frac{([\boldsymbol{\xi}_u^{(k)}]_{z_u} + d\mu_u^{(k)}) + \sum_{b \in \bar{\mathcal{B}}} ([\boldsymbol{\rho}_b^{(k)}]_{z_u} + d\tilde{\mu}_{ub}^{(k)})}{(B+2)d}. \quad (1.68)$$

Updating  $s_{ub}^{(k+1)}$  can be done at BS  $b$  or user  $u$ ; if BS  $b$  updates  $s_{ub}^{(k+1)}$ , it requires the term  $[\boldsymbol{\xi}_u^{(k)}]_{s_{ub}} + dy_{ub}^{(k)}$  from user  $u$ . Variable  $z_u^{(k+1)}$  can be updated at user  $u$ ; for this, user  $u$  requires the term  $[\boldsymbol{\rho}_b^{(k)}]_{z_u} + d\tilde{\mu}_{ub}^{(k)}$  from BS  $b$  for all  $b \in \bar{\mathcal{B}}$ .

BS  $b$  updates its local variables  $\tilde{\mathbf{v}}_b$  by solving the following QCQP problem

$$\underset{\tilde{\mathbf{v}}_b \in \tilde{\mathcal{K}}_b}{\text{minimize}} \quad B_b^{(l)} x_b + (\boldsymbol{\rho}_b^{(k)})^T (\tilde{\boldsymbol{\theta}}_b - \tilde{\boldsymbol{\phi}}_b^{(k+1)}) + \frac{d}{2} \|\tilde{\boldsymbol{\theta}}_b - \tilde{\boldsymbol{\phi}}_b^{(k+1)}\|_2^2. \quad (1.69)$$

Similarly, user  $u$  updates local variable  $\mathbf{v}_u$  by solving the convex problem

$$\underset{\mathbf{v}_u \in \mathcal{K}_u}{\text{minimize}} \quad (\boldsymbol{\xi}_u^{(k)})^T (\boldsymbol{\theta}_u - \boldsymbol{\phi}_u^{(k+1)}) + \frac{d}{2} \|\boldsymbol{\theta}_u - \boldsymbol{\phi}_u^{(k+1)}\|_2^2 - A_u^{(l)} t_u. \quad (1.70)$$

Thus, for the local variable update, user  $u$  receives  $s_{ub}^{(k+1)}$  from BS  $b$ , for all  $b$ , to form  $\boldsymbol{\phi}_u^{(k+1)}$ , and BS  $b$  receives  $z_u^{(k+1)}$  from user  $u$ , for all  $u$ , to form  $\tilde{\boldsymbol{\phi}}_b^{(k+1)}$ .

Finally, the Lagrangian multipliers are updated as follows

$$\boldsymbol{\xi}_u^{(k+1)} = \boldsymbol{\xi}_u^{(k)} + d(\boldsymbol{\theta}_u^{(k+1)} - \boldsymbol{\phi}_u^{(k+1)}) \quad (1.71)$$

$$\boldsymbol{\rho}_b^{(k+1)} = \boldsymbol{\rho}_b^{(k)} + d(\tilde{\boldsymbol{\theta}}_b^{(k+1)} - \tilde{\boldsymbol{\phi}}_b^{(k+1)}). \quad (1.72)$$

Since  $\boldsymbol{\phi}_u^{(k+1)}$  and  $\tilde{\boldsymbol{\phi}}_b^{(k+1)}$  are already available at user  $u$  and BS  $b$ , respectively, updating  $\boldsymbol{\xi}_u^{(k+1)}$  and  $\boldsymbol{\rho}_b^{(k+1)}$  does not require additional exchanged information. For summary, the main steps of the distributed procedure is outlined in

---

**Algorithm 1.5** Decentralized procedure solving (1.54)

---

- 1: **Initialization:** Set  $l := 0$  and  $k := 0$ , then choose initial values for  $(\{\mathbf{w}_u^{(0)}\}, \{t_u^{(0)}\}, \{\mu_u^{(0)}\}, \{\pi_b^{(0)}\})$  and  $(\tilde{\mathbf{v}}_b^{(0)}, \mathbf{v}_u^{(0)}; \boldsymbol{\rho}_b^{(0)}, \boldsymbol{\xi}_u^{(0)})$ .
  - 2: **repeat** {Outer loop (SCA procedure)}
  - 3:   BS  $b$  receives  $t_u^{(l)}$  from user  $u \in \mathcal{U}$  to determine  $B_b^{(l)}$ ; user  $u$  receives  $\pi_b^{(l)}$  from BS  $b \in \bar{\mathcal{B}}$  to determine  $A_u^{(l)}$ .
  - 4:   **repeat** {Inner loop (ADMM procedure)}
  - 5:     **for**  $b \in \bar{\mathcal{B}}$  **and**  $u \in \mathcal{U}$  **do**
  - 6:       BS  $b$  updates  $s_{ub}^{(k+1)}$  using (1.67); user  $u$  updates  $z_u^{(k+1)}$  using (1.68)
  - 7:       BS  $b$  updates  $\tilde{\mathbf{v}}_b^{(k+1)}$  using (1.69); user  $u$  updates  $\mathbf{v}_u^{(k+1)}$  using (1.70)
  - 8:       BS  $b$  updates  $\boldsymbol{\rho}_b^{(k+1)}$  using (1.72); user  $u$  updates  $\boldsymbol{\xi}_u^{(k+1)}$  using (1.71)
  - 9:     **end for**
  - 10:     $k := k + 1$ .
  - 11:   **until** ADMM convergence
  - 12:   Obtain  $(\{\mathbf{w}_u^*\}, \{t_u^*\}, \{\mu_u^*\}, \{\pi_b^*\}; \boldsymbol{\xi}_u^*, \boldsymbol{\rho}_b^*)$ , the solution from the ADMM
  - 13:    $l := l + 1$ ;  $k := 0$
  - 14:    $(\{\mathbf{w}_u^{(l)}\}, \{t_u^{(l)}\}, \{\mu_u^{(l)}\}, \{\pi_b^{(l)}\}) \quad := \quad (\{\mathbf{w}_u^*\}, \{t_u^*\}, \{\mu_u^*\}, \{\pi_b^*\});$   
 $(\tilde{\mathbf{v}}_b^{(0)}, \mathbf{v}_u^{(0)}; \boldsymbol{\rho}_b^{(0)}, \boldsymbol{\xi}_u^{(0)}) := (\tilde{\mathbf{v}}_b^*, \mathbf{v}_u^*; \boldsymbol{\rho}_b^*, \boldsymbol{\xi}_u^*)$
  - 15: **until** SCA convergence
- 

Algorithm 1.5.

### 1.3.2.4. Numerical Examples

We consider an area with radius 500m centered at the origin where the macro-cell BS is placed. There are four small-cell BSs ( $B = 4$ ) placed at (300m,0), (0,300m), (-300m,0) and (0,-300m). The number of antennas equipped at the BSs are  $K_0 = 8$  and  $K_b = 6$  for all  $b \in \mathcal{B}$ . There are  $U = 6$  users randomly generated inside the area. We use the modified Okumura-Hata urban, i.e. (1.46), for path loss and the standard deviation of the shadowing is set as 8 dB. The noise power density is  $N_0 = -174$  dBm/Hz and the operation bandwidth is 1 MHz. The maximum transmission power at BSs are  $P_0 = 40$  dBm and  $P_b = 35$  dBm for all  $b \in \mathcal{B}$ ; we take  $p_{ub}^{\text{cirCo}} = p^{\text{cirCo}}$  for all  $u, b$ ; the power amplifier's



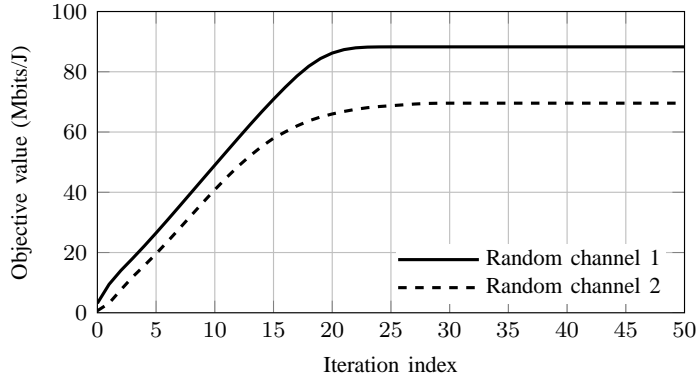


Figure 1.4: Convergence behavior of the SCA procedure over two random channel realizations. The total circuit power  $P^{\text{cir}}$  is 30 dBm.

efficiency at the BSs are  $\lambda_0 = 0.96$  and  $\lambda_b = 0.93$  for all  $b \in \mathcal{B}$ ; we take the QoS thresholds as  $\bar{\mathbf{r}} \triangleq \{\bar{R}_u\} = [1.1; 0.91; 0.75; 1.3; 1.22; 1.15]$  Mbits/s.

Figure 1.4 plots the convergence performance of the SCA procedure solving (1.55) over two random channel realizations. We can observe from the figure that the algorithm converges within a few iterations (less than 30 iterations) in all cases of the considered channels.

Figure 1.5 depicts the average energy efficiency performance of the non-coherent transmission and the coordinated transmission as the functions of  $p^{\text{cirCo}}$ . Here, for the coordinated transmission, each of the users only receives information from the nearest BS; and the circuit power for this scheme is  $P^{\text{cir}} = \sum_{b \in \mathcal{B}} p_b^{\text{cirTx}} + \sum_{u \in \mathcal{U}} p_u^{\text{cirRx}} + \sum_{u \in \mathcal{U}} p_{ub_u}^{\text{cirCo}}$  where  $b_u$  is the BS serving user  $u$ . The main observation is that when  $p^{\text{cirCo}}$  is small the noncoherent transmission outperforms the coordinated transmission in terms of EE due to the gain of the joint processing transmission. As  $p^{\text{cirCo}}$  increases, the EE performance of both schemes decrease; however, the reduction speed of the noncoherent transmission is faster since its circuit consumption power increase faster compared to the coordinated transmission scheme. As a result, when  $p^{\text{cirCo}}$  is large

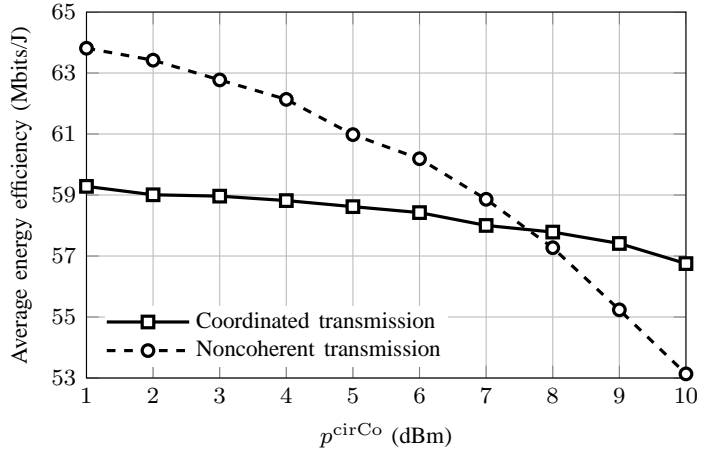


Figure 1.5: Average energy efficiency performance of noncoherent transmission scheme and coordinated transmission scheme with different  $p^{\text{cirCo}}$ ;  $p_u^{\text{cirRx}} = 10$  dBm for all  $u$ ,  $p_0^{\text{cirTx}} = 30$  dBm, and  $p_b^{\text{cirTx}} = 15$  dBm for all  $b \in \mathcal{B}$ .

enough, coordinated transmission outperforms noncoherent transmission.

## 1.4. Conclusions

Energy efficiency optimization for wireless communications usually involves dealing with fractional programs due to the EE definition. We have introduced the optimization tools for overcome various classes of fractional programs including concave fractional programs, max-min fractional programs, and generalized nonconvex fractional programs. While the efficient optimal solutions for the first two classes are available, that for the last is still an open problem. We have also introduced ADMM as an efficient tool for developing distributed implementations.

We have presented the applications of the provided optimization tools in optimizing energy efficiency for dense networks including spectrum sharing networks and dense small-cell networks. In each scenario, centralized and dis-

tributed solutions have been provided.

The results in Figure 1.5 suggests an approach to further improving the energy efficiency performance for the networks that is appropriately controlling the operation modes of the nodes. The approach involves dealing with Boolean (or discrete) variables. The interested reader is referred to [23, Section IV] and [24, Section IV] for examples.

# Bibliography

- [1] Amir Beck, Aharon Ben-Tal, and Luba Tetruashvili. A sequential parametric convex approximation method with applications to nonconvex truss topology design problems. *Journal of Global Optimization*, 47(1):29–51, 2010.
- [2] Emil Bjornson, Marios Kountouris, and Merouane Debbah. Massive MIMO and small cells: Improving energy efficiency by optimal soft-cell coordination. In *20th International Conference on Telecommunications (ICT)*, pages 1–5, May 2013.
- [3] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge, 1st edition, 2004.
- [4] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.
- [5] D.E. Charilas and A.D. Panagopoulous. Multiaccess radio network environments. *IEEE Vehicular Technology Magazine*, 5(4):40–49, 2010.
- [6] A. Charnes and W. W. Cooper. Programming with linear fractional functionals. *Naval Research Logistics Quarterly*, 9(3-4):181–186, 1962.
- [7] Yonghoon Choi, Hoon Kim, Sang wook Han, and Youngnam Han. Joint resource allocation for parallel multi-radio access in heterogeneous wireless networks. *IEEE Transactions on Wireless Communications*, 9(11):3324–3329, Nov. 2010.
- [8] J. P. Crouzeix, J. A. Ferland, and S. Schaible. An algorithm for generalized fractional programs. *Journal of Optimization Theory and Applications*, 47(1):35–49, Sep 1985.
- [9] Jean-Pierre Crouzeix and Jacques A. Ferland. Algorithms for generalized fractional programming. *Mathematical Programming*, 52(1):191–207, May 1991.
- [10] T. Dinh Quoc and M. Diehl. Sequential convex programming methods for solving non-linear optimization problems with DC constraints. *ArXiv e-prints*, July 2011.

- [11] Werner Dinkelbach. On nonlinear fractional programming. *Management Science*, 13(7):492–498, Oct. 1967.
- [12] EARTH. Most suitable efficiency metrics and utility functions. Technical report. URL <https://www.ict-earth.eu>.
- [13] Christian Isheden, Zhijiat Chong, E. Jorswieck, and Gerhard Fettweis. Framework for link-level energy efficiency optimization with informed transmitter. *IEEE Transactions on Wireless Communications*, 11(8):2946–2957, August 2012.
- [14] M. Ismail and Weihua Zhuang. Decentralized radio resource allocation for single-network and multi-homing services in cooperative heterogeneous wireless access medium. *IEEE Transactions on Wireless Communications*, 11(11):4085–4095, Nov. 2012.
- [15] M. Ismail and Weihua Zhuang. A distributed multi-service resource allocation algorithm in heterogeneous wireless access medium. *IEEE Journal on Selected Areas in Communications*, 30(2):425–432, Feb. 2012.
- [16] Barry R. Marks and Gordon P. Wright. A general inner approximation algorithm for nonconvex mathematical programs. *Operations Research*, 26(4):681–683, August 1978.
- [17] Jie Miao, Zheng Hu, Kun Yang, Canru Wang, and Hui Tian. Joint power and bandwidth allocation algorithm with QoS support in heterogeneous wireless networks. *IEEE Communications Letters*, 16(4):479–481, Apr. 2012.
- [18] Kien-Giang Nguyen, Le-Nam Tran, O. Tervo, Quang-Doanh Vu, and M. Juntti. Achieving energy efficiency fairness in multicell MISO downlink. *IEEE Communications Letters*, 19(8):1426–1429, Aug 2015.
- [19] D. Niyato and E. Hossain. Dynamics of network selection in heterogeneous wireless networks: An evolutionary game approach. *IEEE Transactions on Vehicular Technology*, 58(4):2008–2017, 2009.
- [20] S. Schaible and Köln. Parameter-free convex equivalent and dual programs of fractional programming problem. *Zeitschrift für Operations Research*, pages 187–196, Apr. 1973.
- [21] Siegfried Schaible and Jianming Shi. Recent developments in fractional programming: Single ratio and max-min case. *Nonlinear Analysis and Convex Analysis*, pages 493–505, 2004.

- [22] O. Tervo, A. Tölli, M. Juntti, and L. N. Tran. Energy-efficient beam coordination strategies with rate-dependent processing power. *IEEE Transactions on Signal Processing*, 65(22):6097–6112, Nov 2017.
- [23] Q. D. Vu, L. N. Tran, M. Juntti, and E. K. Hong. Energy-efficient bandwidth and power allocation for multi-homing networks. *IEEE Transactions on Signal Processing*, 63(7):1684–1699, April 2015.
- [24] Q. D. Vu, L. N. Tran, R. Farrell, and E. K. Hong. Energy-efficient zero-forcing precoding design for small-cell networks. *IEEE Transactions on Communications*, 64(2):790–804, Feb 2016.
- [25] Lin Xiao and Stephen Boyd. Fast linear iterations for distributed averaging. *Syst. Control Lett.*, 53:65–78, 2004.
- [26] Peng Xue, Peng Gong, Jae Hyun Park, Daeyoung Park, and Duk Kyung Kim. Radio resource management with proportional rate constraint in the heterogeneous networks. *IEEE Transactions on Wireless Communications*, 11(3):1066–1075, Mar. 2012.
- [27] Weihua Zhuang and M. Ismail. Cooperation in wireless communication networks. *IEEE Wireless Communications*, 19(2):10–20, 2012.