Chapter 11
Energy-efficient methods for cloud radio access networks

Kien-Giang Nguyen[1,2], Quang-Doanh Vu[2], Le-Nam Tran[3] and Markku Juntti[2]

Cloud radio access network (C-RAN) is an evolutionary radio network architecture in which a cloud-computing-based baseband (BB) signal-processing unit is shared among distributed low-cost wireless access points. This architecture offers a number of significant improvements over the traditional RANs, including better network scalability, spectral, and energy efficiency. As such C-RAN has been identified as one of the enabling technologies for the next-generation mobile networks. This chapter focuses on examining the energy-efficient transmission strategies of the C-RAN for cellular systems. In particular, we present optimization algorithms for the problem of transmit beamforming designs maximizing the network energy efficiency. In general, the energy efficiency maximization in C-RANs inherits the difficulty of resource allocation optimizations in interference-limited networks, i.e., it is an intractable nonconvex optimization problem. We first introduce a globally optimal method based on monotonic optimization (MO) to illustrate the optimal energy efficiency performance of the considered system. While the global optimization method requires extremely high computational effort and, thus, is not suitable for practical implementation, efficient optimization techniques achieving near-optimal performance are desirable in practice. To fulfill this gap, we present three low-complexity approaches based on the state-of-the-art local optimization framework, namely, successive convex approximation (SCA).

11.1 Introduction

Recent years witness the rapid evolution of wireless technologies toward the fifth-generation (5G) mobile networks to adapt the ever-growing demand of mobile data volumes [1,2]. From the perspective of RAN, novel network architecture, namely, *C-RAN*, has been emerging as a powerful candidate to be implemented in 5G and

[1] Nokia, Oulu, Finland
[2] Centre for Wireless Communications, University of Oulu, Oulu, Finland
[3] School of Electrical and Electronic Engineering, University College Dublin, Dublin, Ireland

beyond [3–5]. In fact, C-RAN has been named by 5G-PPP initiative as a key technology for the 5G network architecture [6]. Moreover, the first C-RAN solutions for commercial 5G have already been developed and released by several companies, such as Nokia (with Nokia AirScale C-RAN) [7] or Ericsson [8]. Unlike the conventional RAN where both BB signal-processing and radio frequency (RF) functionalities are installed at base stations (BSs), BB units (BBUs) in the C-RAN are reintegrated at a central cloud-computing-based platform, referred to as BBU pool. The BSs only need to carry the radio interfaces that can be replaced by low-cost low-power access points called remote radio heads (RRHs). The BBU pools are connected to RRHs by high-speed fronthaul links that are used to exchange the BB and control signals. The data is conveyed through the fronthaul links mainly following either data-sharing or compression-based strategies. For the former, users' messages are directly forwarded from the BBU to a set of cooperative RRHs, where they are precoded/beamformed before being transmitted [9,10]. Alternatively, the users' messages are centrally precoded/beamformed at the BBU in the latter strategy; the processed signals are then compressed before being forwarded to RRHs [11,12].

The innovative architecture of the C-RAN brings a number of advantages over the conventional distributed RAN architecture. The centralized signal-processing mechanism at the cloud across the connected RRHs can provide an efficient allocation of radio and computing resources. This feature also leverages the advanced interference management techniques, such as multicell cooperative transmission, to improve the system capacity. In addition to spectrum efficiency, the C-RAN also provides substantial gain in the energy efficiency of mobile networks. In particular, RRHs with simplified architecture require much reduced amount of power consumption compared to the traditional BSs. Moreover, centralized resource management allows BBU pool controlling the connected RRHs for power-saving purposes, i.e., some RRHs can be switched off if needed to save power consumption, while others remain active to guarantee required users' quality-of-service (QoS). However, one of the critical bottlenecks for the deployment of the C-RAN in practice is the requirement of high-speed fronthaul links. Regardless of the recent development in fronthauling technologies, the capacity of fronthaul links is physically limited [13].

With the mentioned advantages, the C-RAN is currently identified as one of the key enablers to the deployment of essential technologies in 5G to deal with the increasing capacity demand, such as small-cell and/or heterogeneous dense networks [14]. In fact the use of low-power RRHs reduces the size of the cells. This allows to deploy more cells in the same covering geographic area of the traditional BSs. In addition, intercell interference, which constitutes the main limiting factor to the dense deployment, can be efficiently managed at the cloud by means of the centralized signal processing. Nevertheless, such scenarios raise a concern over the huge amount of power consumption required to operate a large number of electronic circuit elements involved. As a result, developing energy-efficient transmission strategies for the C-RAN is deemed important. This chapter presents optimization algorithms for the energy efficiency maximization problems in the C-RANs.

This chapter is organized as follows. In Section 11.2, we provide mathematical preliminaries of optimization techniques that lay the foundations for the development of solutions for energy efficiency optimization in C-RANs. In Section 11.3, the system model of a C-RAN and the formulation of the energy efficiency maximization problem are described, followed by the application of the introduced optimization approaches in Section 11.2 to the design problem.

11.2 Energy efficiency optimization: mathematical preliminaries

A general energy efficiency maximization problem for wireless networks with data rate as the desired QoS metric can be expressed as

$$\underset{\mathbf{x}}{\text{maximize}} \quad \frac{f(\mathbf{x})}{g(\mathbf{x})} \tag{11.1a}$$

$$\text{subject to} \quad h_i(\mathbf{x}) \leq 0, \; i = 1, \ldots, m, \tag{11.1b}$$

where $f(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ represents the data rate of the system, $g(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ is the corresponding total power consumption, and $h_i(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ $(i = 1, \ldots, m)$ defines the design constrains, i.e., it can be a function of transmit power of a transmitter or an antenna, user-specific QoS, fronthaul capacity, etc. Function $h_i(\mathbf{x})$ can be either convex or nonconvex. In either case, (11.1) is a *nonconvex program* due to the fractional structure of the objective [15]. In the special case where $f(\mathbf{x})$, $g(\mathbf{x})$, and $h_i(\mathbf{x})$ $(i = 1, \ldots, m)$ are convex, (11.1) is quasi-concave program of which globally optimum solutions can be derived by linearly (or even superlinearly) convergent optimization methods such as Dinkelbach's algorithms or Charnes–Cooper's transformations [15]. In wireless communications, this likely corresponds to the noise-limited scenarios [16]. However, this is not the case of the C-RANs where multiple RRHs serve multiple users using the same sources of time and frequency, creating interference-limited channels. To be specific, the rate function in the C-RAN is nonconvex with respect to power or beamforming variables. In such scenarios, globally optimal solutions of nonconvex program (11.1) can be found by the non-polynomial time global optimization frameworks such as MO [16]. Nevertheless, such methods are suitable for benchmarking purposes only as it often requires prohibitively high computational complexity. In practice, efficient optimization approaches that tend to yield near-optimal solutions using lower computational resources are more desirable. In the following, we discuss global and local optimization frameworks that can be used to solve (11.1).

11.2.1 Global optimization method: monotonic optimization

Numerous nonconvex optimization problems in wireless communications fall to the class of MO [17]. These can be solved globally by a powerful algorithm in the framework of MO, namely, branch-reduce-and-bound (BRnB). In this section, we provide some backgrounds of the MO and describe the BRnB method. To proceed, some basic concepts of MO are first introduced [18].

Definition 11.1. *For any two vectors* $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$*, writing* $\mathbf{y} \geq \mathbf{x}$ *(*$\mathbf{y} > \mathbf{x}$*, resp.) means* $y_i \geq x_i$ *(*$y_i > x_i$*, resp.) for every* $i = 1, 2, \ldots, n$*. If* $\mathbf{a} \leq \mathbf{b}$*, then the box* $[\mathbf{a}, \mathbf{b}]$ *(*$(\mathbf{a}, \mathbf{b}]$*, resp.) is the set of all* $\mathbf{x} \in \mathbb{R}^n$ *satisfying* $\mathbf{a} \leq \mathbf{x} \leq \mathbf{b}$ *(*$\mathbf{a} < \mathbf{x} \leq \mathbf{b}$*, resp.).*

Definition 11.2. (Increasing function)*A function* $f : \mathbb{R}^n_+ \rightarrow \mathbb{R}$ *is said to be increasing on* $\mathbb{R}^n_+$ *if* $f(\mathbf{x}') \geq f(\mathbf{x})$ *when* $\mathbf{x}' \geq \mathbf{x}$.

Definition 11.3. (Normal set) *A set* $\mathscr{S} \subset [\mathbf{a}, \mathbf{b}]$ *is said to be normal in* $[\mathbf{a}, \mathbf{b}]$ *(or briefly, normal) if* $\mathbf{x} \in \mathscr{S} \Rightarrow [\mathbf{a}, \mathbf{x}] \subset \mathscr{S}$*. For a closed normal set* $\mathscr{S}$ *in* $[\mathbf{a}, \mathbf{b}]$*, a point* $\bar{\mathbf{x}}$ *is called an upper boundary point if the cone* $\boldsymbol{K}_{\bar{\mathbf{x}}} \triangleq \{\mathbf{x} \mid \mathbf{x} > \bar{\mathbf{x}}\}$ *contains no point of* $\mathbf{x} \in \mathscr{S}$*.*

The standard form of an MO problem is generally written as [18]

$$\underset{\mathbf{x}}{\text{maximize}} \quad f(\mathbf{x}) \tag{11.2a}$$

$$\text{subject to} \quad h_i(\mathbf{x}) - \tilde{h}_i(\mathbf{x}) \leq 0, \ i = 1, \ldots, m \tag{11.2b}$$

$$\mathbf{x} \subseteq B \triangleq [\mathbf{a}; \mathbf{b}], \tag{11.2c}$$

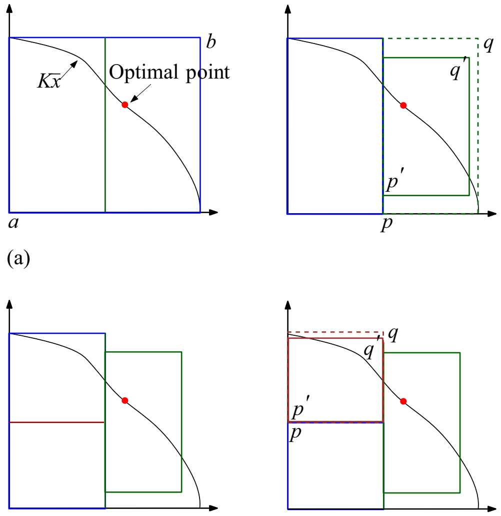
where $f(\mathbf{x})$, $h_i(\mathbf{x})$, and $\tilde{h}_i(\mathbf{x})$ $(i = 1, \ldots, m)$ are increasing functions of variable $\mathbf{x}$, and B is the box containing the feasible set of (11.2) i.e., $\{\mathbf{x} | h_i(\mathbf{x}) - \tilde{h}_i(\mathbf{x}) \leq 0, \ i = 1, \ldots, m\} \subseteq B$. Vectors **a** and **b** are the lower and upper vertices of B, respectively. We consider here a mixed-variable vector **x** of size N containing binary and continuous variables, i.e., $x_j \in \{0, 1\}$ for $j = 1, \ldots, s$ and $x_j \in \mathbb{R}$ for $j = s + 1, \ldots N$. The MO problem can be solved by the BRnB method introduced in [18,19]. The following presents the main steps of the BRnB.

Branch–reduce-and-bound algorithm

BRnB algorithm is an iterative procedure executing three basic operations at each iteration: *branching*, *reduction*, and *bounding*. Starting from initial box $[\mathbf{a}; \mathbf{b}]$, the algorithm (i) iteratively divides it into smaller and smaller ones; (ii) removes boxes that do not contain an optimal solution; and (iii) searches the remaining boxes for an improved solution until an error tolerance is met. An example of the BRnB procedure is illustrated in Figure 11.1.

It is worth mentioning that the BRnB algorithm was first developed for continuous MO problems [19]. As a result, when applying to problems containing discrete variables/feasible set as in (11.2), the algorithm may not return exact solutions of the discrete variables, but approximated ones [18]. To avoid such drawback, a modification of the BRnB was later developed for the discrete optimization programs in which an adjustment step is additionally applied to the discrete variables to map them into the corresponding discrete set. The details of BRnB method solving (11.2) are provided next.

For the ease of exposition, we introduce the following denotations that are used throughout this section: $\mathbf{e}_j$ is a unit vector such that $[\mathbf{e}_j] = 1$ and $[\mathbf{e}_i] = 0$ if $i \neq j$; n counts the iteration index; $\mathscr{B}_n$ denotes the set of candidate boxes at the nth iteration and $V = [\mathbf{p}; \mathbf{q}] \in \mathscr{B}_n$ denotes an arbitrary box in $\mathscr{B}_n$;$f_{\text{UpB}}(V)$ denotes the upper bound of V, i.e.,$f_{\text{UpB}}(V) \geq \max\{f(\mathbf{x}) \,|\, h_i(\mathbf{x}) - \tilde{h}_i(\mathbf{x}) \leq 0, i = 1, \ldots, m, \ \mathbf{x} \in [\mathbf{p}; \mathbf{q}]\}$; the *current*



(a)

(b)

Figure 11.1 Example of the BRnB procedure for branching and reduction operations over two iterations: (a) Iteration 1 and (b) Iteration 2

best objective (CBO) stands for the known point in the feasible set that offers the best objective value at the current iteration. Note that at the very first iteration, we have $\mathscr{B}_1 \equiv B$, $[\mathbf{p};\mathbf{q}] \equiv [\mathbf{a};\mathbf{b}]$, $f_{\text{UpB}}(V) = f(\mathbf{b})$.

Branching

At iteration n, one box in $\mathscr{B}_n$ is picked to be branched into two new boxes. In particular, the candidate box, denoted as V_{c}, is bisected along the longest edge, which is determined as $l = \arg\max_{1 \le j \le N}(q_j - p_j)$. This results in two new smaller boxes of equal size given by

$$V^1 = [\mathbf{p};\mathbf{q}'] \;\text{ where } q'_j = \begin{cases} q_j & \text{if } j \neq l, \\ 0 & \text{if } j = l \le s, \\ q_j - (q_j - p_j)/2 & \text{if } j = l > s, \end{cases} \tag{11.3}$$

$$V^2 = [\mathbf{p}';\mathbf{q}] \;\text{ where } p'_j = \begin{cases} p_j & \text{if } j \neq l, \\ 1 & \text{if } j = l \le s, \\ p_j + (q_j - p_j)/2 & \text{if } j = l > s. \end{cases} \tag{11.4}$$

We note that V_{c} in general should be selected such that it has the largest upper bound among boxes in $\mathscr{B}_n$, i.e., $V_{\text{c}} = \arg\max_{V \in \mathscr{B}_n} f_{\text{UpB}}(V)$ [19]. This is to ensure the monotonic decrease (increase, respectively) of the upper bound (lower bound, respectively) of the resulting boxes.

Reduction

The newly created boxes may contain some portions of no interest, i.e., those have no feasible solution, or if they do, their feasible points give objective values smaller than the CBO. Thus, efficiently cutting such regions out of the box shall limit the feasible space, thereby accelerating the convergence. The reduction step is developed for this purpose. In particular, given a box $V = [\mathbf{p}; \mathbf{q}]$, we use reduction cut to find a smaller box, denoted as $\mathrm{r}(V) \triangleq [\mathbf{p}'; \mathbf{q}'] \subset V$, such that an optimal solution (if it exists in V) must be contained in $\mathrm{r}(V)$. In other words, we eliminate the portions $[\mathbf{p}; \mathbf{p}')$ and $(\mathbf{q}'; \mathbf{q}]$ that are checked to not contain an optimal solution. To do so, we replace $\mathbf{p}$ by $\mathbf{p}' \geq \mathbf{p}$ where $\mathbf{p}' = \mathbf{q} - \sum_{j=1}^{N} \rho_j^{(1)}(q_j - p_j)\mathbf{e}_j$ and

$$\begin{aligned}\rho_j^{(1)} &= \sup\{\rho^{(1)} | 0 \leq \rho^{(1)} \leq 1,\ h_i(\mathbf{p}) - \tilde{h}_i(\mathbf{q} - \rho^{(1)}(q_j - p_j)\mathbf{e}_j) \leq 0, \\ &\qquad f(\mathbf{q} - \rho^{(1)}(q_j - p_j)\mathbf{e}_j) \geq \mathrm{CBO}\},\ \text{for } j = 1, \ldots, N.\end{aligned} \tag{11.5}$$

To find $\mathbf{q}'$, we replace $\mathbf{q}$ by $\mathbf{q}' \leq \mathbf{q}$ where $\mathbf{q}' = \mathbf{p}' + \sum_{j=1}^{N} \rho_j^{(2)}(q_j - p_j')\mathbf{e}_j$ and

$$\rho_j^{(2)} = \sup\{\rho^{(2)} | 0 \leq \rho^{(2)} \leq 1,\ h(\mathbf{p}' + \rho^{(2)}(q_j - p_j')\mathbf{e}_j) - \tilde{h}_i(\mathbf{q}) \leq 0\}. \tag{11.6}$$

The values of $\rho_j^{(1)}$ and $\rho_j^{(2)}$ can be found easily using the bisection method.

Remark that for $j = 1, \ldots, s$ (i.e., *binary variables*), we have $q_j - p_j = 1$, and thus, p_j' can be quickly obtained by

$$p_j' = \begin{cases} 0 & \text{if } h_i(\mathbf{p}) - \tilde{h}_i(\mathbf{q} - \mathbf{e}_j) \leq 0 \\ 1 & \text{otherwise.} \end{cases}$$

If $p_j' = 0$, then $q_j - p_j' = 1$, and thus q_j' is computed as

$$q_j' = \begin{cases} 1 & \text{if } h(\mathbf{p}' + \mathbf{e}_j) - \tilde{h}_i(\mathbf{q}) \leq 0 \\ 0 & \text{otherwise.} \end{cases}$$

Bounding

Bounding is the most crucial operation to ensure the convergence of the branch-and-bound-type methods in general and the BRnB method in particular. In this step, we update the upper and lower bounds of the boxes after reduction (i.e., $\mathrm{r}(V) \triangleq [\mathbf{p}'; \mathbf{q}']$). Due to the monotonicity, i.e., $\mathbf{p}' \leq \mathbf{x} \leq \mathbf{q}'$ for any $\mathbf{x} \in \mathrm{r}(V)$, the upper and lower bounds of box $\mathrm{r}(V)$ can be simply found as $f_{\mathrm{LoB}}(V) = f(\mathbf{p}')$ and $f_{\mathrm{UpB}}(V) = f(\mathbf{q}')$, respectively. The convergence is declared when $f_{\mathrm{UpB}}(V) - f_{\mathrm{LoB}}(V) \leq \varepsilon$ for small threshold $\varepsilon > 0$.

It is worth mentioning that if the newly updated upper bound of a box is smaller than the CBO, we can remove that box to save the computational complexity. Thus, together with bounding, efficiently updating the CBO is beneficial for improving the algorithm's efficiency.

Algorithm 11.1: The SCA procedure solving (11.7)

```
**Initialization:** Set $n := 0$, choose an initial feasible point $\mathbf{x}^{(n)}$
**repeat** $\{n := n + 1\}$
    Solve (11.9) and obtain optimal value $\mathbf{x}^*$
    Update $\mathbf{x}^{(n)} := \mathbf{x}^*$
**until** Convergence
**Output:** $\mathbf{x}^{(n)}$
```

11.2.2 Local optimization method: successive convex approximation

For a general nonconvex program, optimal solutions could be found applying the branch-and-bound-type methods but with the price of extremely high computational complexity. Instead, using local optimization approaches to arrive at near-optimal solutions is more appealing in practice [20]. Among the local optimization approaches, the SCA method is increasingly applied for nonconvex problems in wireless communications and also other fields.

SCA is an iterative method that aims at locating a Karush–Kuhn–Tucker (KKT) solution to a nonconvex program. The central idea is to iteratively approximate the nonconvex parts by the proper convex ones [21,22]. The general procedure of the SCA is outlined in Algorithm 11.1, and the details are discussed next. Let us consider a general optimization program given by

$$\underset{\mathbf{x}}{\text{minimize}}\ f_0(\mathbf{x}) \quad \text{subject to} \quad \{f_i(\mathbf{x}) \leq 0,\ i = 1, ..., m\}, \tag{11.7}$$

where $f_i(\mathbf{x})$ $(i = 0, \dots, l)$ are all continuously differentiable functions over $\mathbb{R}^N$. The feasible set is denoted as $\mathcal{S}(\mathbf{x}) \triangleq \{\mathbf{x} \in \mathbb{R}^N |\ f_i(\mathbf{x}) \leq 0,\ i = 1, \dots, m\}$, which is assumed to be a compact set. In $\mathcal{S}(\mathbf{x})$, we also assume that $f_i(\mathbf{x})$ $(i = 0, \dots, l)$ are convex functions and $f_i(\mathbf{x})$ $(i = l + 1, \dots, m)$ are nonconvex ones. Clearly the problem is nonconvex due to the last $l - m$ constraints. The central idea of the SCA is to iteratively approximate $f_i(\mathbf{x})$ $(i = l + 1, \dots, m)$ by its convex upper bounds [21,22]. More specifically, given a feasible point $\mathbf{x}' \in \mathcal{S}(\mathbf{x}), f_i(\mathbf{x})$ $(i = l + 1, \dots, m)$ is replaced by a convex function $\tilde{f}_i(\mathbf{x}, g(\mathbf{x}'))$ satisfying the following properties:

$$\begin{aligned} &\textbf{(a)}\, f_i(\mathbf{x}) \leq \tilde{f}_i(\mathbf{x}; g_i(\mathbf{x}')) \\ &\textbf{(b)}\, f_i(\mathbf{x}') = \tilde{f}_i(\mathbf{x}'; g_i(\mathbf{x}')) \\ &\textbf{(c)}\ \nabla_{\mathbf{x}} f_i(\mathbf{x}') = \nabla_{\mathbf{x}} \tilde{f}_i(\mathbf{x}', g_i(\mathbf{x}')), \end{aligned} \tag{11.8}$$

for all $\tilde{\mathcal{S}}(\mathbf{x}; \mathbf{x}') \triangleq \{\mathbf{x} \in \mathbb{R}^N | f_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, l; \quad \tilde{f}_i(\mathbf{x}, g(\mathbf{x}')) \neq 0, \quad i = l + 1, \dots, m\}$, where $g_i(\mathbf{x}')$: $\mathbb{R}^N \to \mathbb{R}^K$ is a parameter vector. We remark that $\tilde{\mathcal{S}}(\mathbf{x}; \mathbf{x}')$

needs to satisfy Slater's constraint qualification [21]. The replacement leads to the following convex subproblem:

$$\underset{\mathbf{x}}{\text{minimize}}\ f_0(\mathbf{x}) \quad \text{subject to} \quad \{\mathbf{x} \in \tilde{\mathscr{S}}(\mathbf{x};\mathbf{x}')\}. \tag{11.9}$$

Since the optimal solution of (11.9), denoted as $\mathbf{x}^*$, belongs to $\tilde{\mathscr{S}}(\mathbf{x};\mathbf{x}')$ due to the conditions (a) and (b), it is relevant to use $\mathbf{x}^*$ to form subproblem (11.9) for the next iteration. To be specific, let $\mathbf{x}^{(n)}$ denote the optimal solution at iteration n, then the feasible set of the subproblem at iteration $n+1$ is $\tilde{\mathscr{S}}(\mathbf{x};\mathbf{x}^{(n)})$ (see Step 4 in Algorithm 11.1). The process is iteratively carried out until the convergence is established.

The update rule and the conditions (a) and (b) guarantee the convergence of Algorithm 11.1. In particular, recall that $\tilde{\mathscr{S}}(\mathbf{x};\mathbf{x}^{(n)})$ is the feasible set of the subproblem at iteration $n+1$. Thus, as $\mathbf{x}^{(n+1)}$ is the optimal solution obtained by solving (11.9) with $\tilde{\mathscr{S}}(\mathbf{x};\mathbf{x}^{(n)})$, we immediately have $f_0(\mathbf{x}^{(n+1)}) \leq f_0(\mathbf{x}^{(n)})$ since $\mathbf{x}^{(n)}$ is a point in $\tilde{\mathscr{S}}(\mathbf{x};\mathbf{x}^{(n)})$ due to (11.8)(a) and (b). This implies the monotonic decrease of the sequence of objective values. As a result, the convergence is ensured given that the feasible set $\mathscr{S}(\mathbf{x})$ is bounded. On the other hand, the convergence points satisfy the KKT optimality conditions due to the properties (b) and (c) as shown in [21,22].

It is worth noting that if $f_i(\mathbf{x})$ $(i = l+1,\ldots,m)$ are not differentiable but subdifferentiable at $\mathbf{x}'$, the algorithm can still produce a monotonically decreasing sequence of objective because (11.8)(a) and (b) remain satisfied in this case. However, the approximation function here may not hold (11.8)(c), and thus achieved solutions at the convergence may not satisfy the KKT optimality conditions. Another remark is that the objective $f_0(\mathbf{x})$ might be nonconvex in some applications. In such problems, we can apply the SCA principle for $f_0(\mathbf{x})$, i.e., $f_0(\mathbf{x})$ is replaced by $\tilde{f}_0(\mathbf{x}; g_0(\mathbf{x}'))$ satisfying the conditions in (11.8) in the approximate problem (11.9). For more convergence results of the SCA, we refer the interested readers to references [21–23].

Obviously, it is important to find the convex approximations that hold the conditions in (11.8). We provide next some SCA-applicable functions and their approximated formulations that hold conditions in (11.8). Those are useful to derive SCA-based algorithms for the energy efficiency maximization problems.

Useful approximate formulations

- Fractional-linear function: $\phi(x,y) \triangleq \frac{x}{y}$ where $(x,y) \in \mathbb{R}^2_{++}$. Its convex approximation is given by

$$\phi^{\text{frac}}(x,y;\lambda) \triangleq 0.5\left(\lambda x^2 + \frac{1}{\lambda y^2}\right), \text{ where } \lambda = \frac{1}{x'y'}. \tag{11.10}$$

- Quadratic-over-linear function: $\phi(\mathbf{x},y;\mathbf{H}) \triangleq \frac{\mathbf{x}^{\mathrm{H}}\mathbf{H}\mathbf{x}}{y}$ where $\mathbf{x} \in \mathbb{C}^n$, $y \in \mathbb{R}_{++}$, and $\mathbf{H} \succeq 0$. The approximation of $\phi(\mathbf{x},y;\mathbf{H})$ can be achieved by means of the first-order Taylor series, i.e.,

$$\begin{aligned}\phi^{\text{qol}}(\mathbf{x},y;\mathbf{x}',y';\mathbf{H}) &\triangleq \phi(\mathbf{x}',y';\mathbf{H}) + \left\langle [\nabla_{\mathbf{x}}\phi(\mathbf{x}',y';\mathbf{H}), \nabla_y\phi(\mathbf{x}',y';\mathbf{H})]^{\mathrm{T}}, [\mathbf{x}-\mathbf{x}', y-y']^{\mathrm{T}} \right\rangle \\ &= \frac{(\mathbf{x}')^{\mathrm{H}}\mathbf{H}\mathbf{x}'}{(y')^2}y - \frac{2\Re((\mathbf{x}')^{\mathrm{H}}\mathbf{H}\mathbf{x})}{y'}.\end{aligned} \tag{11.11}$$

- Power function: $\phi(x;p) \triangleq x^p$ where $x \in \mathbb{R}_{++}$. An approximation of $\phi(x;p)$ is

$$\phi^{\text{po}}(x;x';p) \triangleq (p-1)(x')^p - p(x')^{p-1}x. \qquad (11.12)$$

11.3 Cloud radio access networks: system model and energy efficiency optimization formulation

We focus on examining the problem of beamformer design maximizing the network energy efficiency of a downlink C-RAN taking into account the impact of capacity-finite fronthaul links. To address the fronthaul limitation, the idea is to reduce the amount of BB signals exchanged through the fronthaul links. This is done by selecting a properly small subset of users associated with an RRH, giving rise to the RRH-user association problem that is jointly designed with the transmit beamforming. In addition, to exploit the potential energy efficiency gain of the C-RAN, RRH selection scheme is used in which some RRHs are properly switched into a sleep mode for power-saving purposes. Noticeably, RRH-user association and RRH selection are commonly adopted in the resource allocation for the C-RAN to cope with the fronthaul constraints and/or to improve the energy efficiency.

The energy-efficient designs for the C-RAN employing RRH-user association and RRH selection have been addressed in some recent works [24,25]. Therein, however, energy efficiency is improved by minimizing the total power consumption [24,26]. Since the data rate is not jointly optimized in the objective, energy efficiency performance achieved by such methods may be far from the optimal one. Very recently, approaches addressing the energy efficiency objective in C-RAN have been developed in [10,12,27], which are summarized herein.

We present a global optimization method to the joint design problem based on the BRnB. For practical applications, we provide three suboptimal solutions based on the SCA that aim at achieving near-optimal solutions but with remarkably reduced complexity compared to the globally optimal method.

11.3.1 System model

We consider the downlink of a C-RAN in which R RRHs, each equipped with M antennas,[*] cooperatively serve K single-antenna users following joint transmission scheme [28]. In particular, data information for each user is transmitted from multiple RRHs. In this section, we focus on the data-sharing-based fronthauling strategy. Energy-efficient transmission designs for compression-based CRANs can be referred to [12,27]. For notational convenience, we denote by $\mathscr{R} \triangleq \{1,\ldots,R\}$ the set of RRHs, and $\mathscr{K} \triangleq \{1,\ldots,K\}$ the set of users. The transmit signals are processed at a common BBU pool, which are then forwarded to the RRHs via finite-capacity fronthaul links. A simplified model of the considered system is illustrated in Figure 11.2. The BBU pool is assumed to perfectly know the channel state information associated with all the users in the network. We denote by $\mathbf{h}_{i,k} \in \mathbb{C}^{1\times M}$ the channel vector between RRH

[*] The assumption on the number of equipped antennas for all RRHs is merely for notational simplicity.

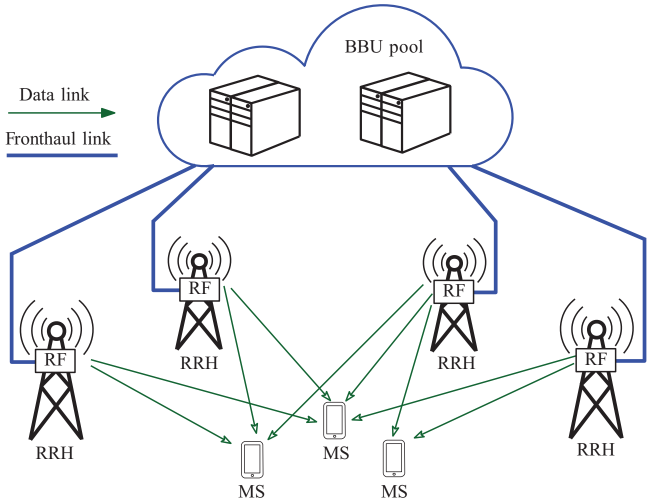


Figure 11.2 A simplified system model of the downlink of a multiuser C-RAN

i and user k. We also assume that linear beamforming scheme is adopted to form the RF transmit signals at the RRHs. Let d_k denote the unit-energy data symbol intended for user k, i.e., $\mathbb{E}[|d_k|^2] = 1$, and $\mathbf{w}_{i,k} \in \mathbb{C}^{M\times 1}$ be the transmit beamformer for d_k at RRH i. The received signal at user k can be written as

$$\begin{aligned} y_k &= \underbrace{\Big(\sum_{i\in\mathcal{R}} \mathbf{h}_{i,k}\mathbf{w}_{i,k}\Big)d_k}_{\text{desired signal}} + \underbrace{\sum_{j\in\mathcal{K}\setminus k}\Big(\sum_{i\in\mathcal{R}} \mathbf{h}_{i,k}\mathbf{w}_{i,j}\Big)d_j}_{\text{interference}} + n_k \\ &= \mathbf{h}_k\mathbf{w}_k d_k + \sum_{j\in\mathcal{K}\setminus k} \mathbf{h}_k\mathbf{w}_{i,j} d_j + n_k, \end{aligned} \tag{11.13}$$

where $n_k \sim \mathcal{CN}(0,\sigma_k^2)$ is the additive white Gaussian noise at user k. In (11.13), we have denoted $\mathbf{h}_k \triangleq [\mathbf{h}_{1,k}, \mathbf{h}_{2,k}, \ldots, \mathbf{h}_{R,k}] \in \mathbb{C}^{1\times MR}$ and $\mathbf{w}_k \triangleq [\mathbf{w}_{1,k}^{\mathrm{T}}, \mathbf{w}_{2,k}^{\mathrm{T}}, \ldots, \mathbf{w}_{R,k}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{C}^{MR\times 1}$ that are the aggregate vectors of all channels and beamformers from all RRHs to user k, for notational simplicity. Assuming that each user decodes its own data symbol while treating interference symbol as noise. Then the signal-to-interference-plus-noise can be written as

$$\gamma_k(\mathbf{w}) \triangleq \frac{|\mathbf{h}_k\mathbf{w}_k|^2}{\sum_{j\in\mathcal{K}\setminus k}|\mathbf{h}_k\mathbf{w}_j|^2 + \sigma_k^2}. \tag{11.14}$$

Let r_k be the achievable data rate transmitted to user k. To guarantee reliable communication, the rate must be below the instantaneous mutual information between channel input and output or the constraint

$$r_k \leq W\log(1+\gamma_k(\mathbf{w})),\ \forall k \in \mathcal{K},$$

where W is the transmission bandwidth, must hold.

11.3.2 Power constraints

The power used for transmission at RRH i should not exceed its available power budget, denoted as P_i. The power constraint for RRH i can be expressed as

$$\sum_{k \in \mathcal{K}} \|\mathbf{w}_{i,k}\|_2^2 \le P_i, \ \forall i \in \mathcal{R}.$$

In addition, the power amplifier (PA) of each antenna chain has the maximum RF output power. Let $[\mathbf{w}_{i,k}]_m$ denote the beamforming coefficient in vector $\mathbf{w}_{i,k}$ corresponding to antenna m, and $\tilde{\mathbf{w}}_{i,m} \triangleq [[\mathbf{w}_{i,1}]_m; [\mathbf{w}_{i,2}]_m; \ldots; [\mathbf{w}_{i,K}]_m] \in \mathbb{C}^{K \times 1}$ stack all $[\mathbf{w}_{i,k}]_m$. The RF power radiated at antenna m should satisfy:

$$\sum_{k \in \mathcal{K}} |[\mathbf{w}_{i,k}]_m|^2 = \|\tilde{\mathbf{w}}_{i,m}\|_2^2 \le P_{i,m}^{\text{ant}},$$

where $P_{i,m}^{\text{ant}}$ is the maximum RF output power at the antenna m of RRH i. Without loss of generality, we assume that $P_i = \bar{P}, \forall i$; and $P_{i,m}^{\text{ant}} = P^{\text{ant}}, \forall i, m$.

11.3.3 Fronthaul constraint

We assume that the maximum capacity of a fronthaul link is $\bar{C}$. For feasible transmission, the total data rate of the wireless physical layer of RRH i should not be larger than $\bar{C}$. It is noted that the fronthaul rate in each link is proportional to the number of users served by the corresponding RRH. As a result, under the fronthaul-constrained scenario, each RRH might not need to serve all users in the network, but a smaller group of selected users to which it can potentially provide the best network performance [9,25]. In such cases, the RRH-user association method is needed.

For the formulation purpose, let us define the following indicator function:

$$\mathbb{I}(x) = \begin{cases} 1 & \text{if } x \neq 0 \\ 0 & \text{if } x = 0 \end{cases}. \tag{11.15}$$

Now we can present by the indicator function on the beamforming vector $\mathbf{w}_{i,k}$, whether RRH i sends information to user k as

$$\mathbb{I}(\|\mathbf{w}_{i,k}\|_2^2) = \begin{cases} 1 & \text{if } \|\mathbf{w}_{i,k}\|_2^2 > 0 \\ 0 & \text{if } \|\mathbf{w}_{i,k}\|_2^2 = 0 \end{cases}.$$

Accordingly, the aggregate data rate, which can be reliably transmitted by the wireless interface of RRH i, is calculated as $\sum_{k \in \mathcal{K}} \mathbb{I}(\|\mathbf{w}_{i,k}\|_2^2) r_k$. Thus, the fronthaul constraint can be expressed as

$$\sum_{k \in \mathcal{K}} \mathbb{I}(\|\mathbf{w}_{i,k}\|_2^2) r_k \le \bar{C}, \ \forall i \in \mathcal{R}.$$

In practice, the values of $\bar{C}$ range from tens of Mbits/s to tens of Gbits/s, depending on fronthaul transport solutions (e.g., passive optical networks, fiber, microwave, Ethernet cables, and wireless communications) and deployments/network requirements (e.g., indoor, outdoor, and urban or rural environments) [14,29,30].

11.3.4 Power consumption

The power consumption is the crucial element to quantify the energy efficiency measure. In this section, we model the network power consumption following those in [31–34] which includes three main components: power for operating electronic circuits, power for signal processing and fronthauling, and power dissipated on PAs.

11.3.4.1 Circuit power consumption

A significant amount of power is required for operating the RF chain of RRHs (e.g., analog-to-digital or digital-to-analog converters, filters, and mixers) as well as the elements of network infrastructure. Assuming that RRHs and the associated network units are configured to be switchable between sleeping and active modes, each of them consumes a dedicated power P^{act} and P^{sl}, respectively, with $P^{\mathrm{act}} \gg P^{\mathrm{sl}}$. For power-saving purposes, an RRH should be put into a sleep mode when it does not transmit to any user, and otherwise it is active when having data to transmit. To represent the operation modes of the RRH, we can use the indicator function introduced in (11.15). By noting that $\sum_{k\in\mathcal{K}} \|\mathbf{w}_{i,k}\|_2^2$ is the total radiated power at RRH i, the operation modes of the RRH can be interpreted as $\mathbb{I}(\sum_{k\in\mathcal{K}} \|\mathbf{w}_{i,k}\|_2^2)$. The circuit power consumption can be then modeled as [32,33]

$$P_{\mathrm{cir}}(\mathbf{w}) \triangleq \sum_{i\in\mathcal{R}} \mathbb{I}\left(\sum_{k\in\mathcal{K}} \|\mathbf{w}_{i,k}\|_2^2\right) P^{\mathrm{act}} + \sum_{i\in\mathcal{R}} \left(1 - \mathbb{I}\left(\sum_{k\in\mathcal{K}} \|\mathbf{w}_{i,k}\|_2^2\right)\right) P^{\mathrm{sl}} + P_0, \quad (11.16)$$

where P_0 includes the fixed power consumption to keep the network operating, e.g., power supply for the network infrastructures [32].

11.3.4.2 Signal processing and fronthauling power

The data needs to be encoded/decoded and modulated/demodulated. In general, a higher data rate generally requires a larger codebook, and the larger number of bits incurs higher power for encoding and decoding on the BB circuit boards. Moreover, the fronthaul network transports the data signals, and, thus, also consumes power. In fact, the power consumption for fronthauling varies with the data rate. For example, in wireless fronthaul networks, higher fronthaul rate requires higher power consumption. From this standpoint, the amount of power consumed for signal processing and fronthauling is rate dependent. Assuming that this amount of power is linearly scaled with the total fronthaul rate [31], the signal processing and fronthauling power can be written as

$$P_{\text{SP-FH}}(\mathbf{w}, \mathbf{r}) \triangleq \delta \sum_{i\in\mathcal{R}} \sum_{k\in\mathcal{K}} \mathbb{I}(\|\mathbf{w}_{i,k}\|_2^2) r_k, \quad (11.17)$$

where δ is a constant coefficient with unit W/(Gbits/s) and $\mathbf{r} \triangleq [r_1, \ldots, r_K]^{\mathrm{T}}$.

11.3.4.3 Dissipated power on PA

The amount of power dissipated on the PAs strongly depends on the PA's efficiency which is defined as the ratio between the total RF output power and the direct current input power. Conventionally, PA's efficiency is assumed to be a constant over the operating range. However, this assumption may hold true for only some transceiver types, in which advanced PA architectures can be implemented to make the PA's transfer function operating in its linear region (e.g., macro and micro BSs in LTE systems). For smaller size and/or low-cost transmitter such as RRHs, such techniques are expensive to employ [32]. Therefore, for this scenario, PA's efficiency is highly dependent on the output power region. To account for this, we consider the nonlinear power consumption model of the PA, in which the PA's efficiency is a function of its radiated power [34,35], i.e.:

$$\varepsilon_{i,m}(\mathbf{w}) \triangleq \frac{\sqrt{P^{\text{ant}}}}{\varepsilon_{\max}} \sqrt{\sum_{k\in\mathcal{K}} |[\mathbf{w}_{i,k}]_i|^2} = \varepsilon \|\tilde{\mathbf{w}}_{i,m}\|_2, \tag{11.18}$$

where $\varepsilon \triangleq \frac{\sqrt{P^{\text{ant}}}}{\varepsilon_{\max}}$ and P^{ant} and $\varepsilon_{\max} \in [0, 1]$ are the maximum power of the PA and the maximum PA's efficiency, respectively. Following the definition of the PA's efficiency, the power consumed at the PA m of RRH i for radiating the signals outward the antenna is

$$P_{\text{PA},(i,m)}(\mathbf{w}) \triangleq \frac{\|\tilde{\mathbf{w}}_{i,m}\|_2^2}{\varepsilon_{i,m}(\mathbf{w})} = \frac{1}{\varepsilon}\|\tilde{\mathbf{w}}_{i,m}\|_2, \tag{11.19}$$

which is clearly a function of the beamforming vector. From (11.19), the total power dissipated on all the PAs is calculated as

$$P_{\text{PA}}(\mathbf{w}) \triangleq \frac{1}{\varepsilon}\sum_{i\in\mathcal{R}}\sum_{m=1}^{M} \|\tilde{\mathbf{w}}_{i,m}\|_2. \tag{11.20}$$

11.3.4.4 Total power consumption

To summarize, the total consumed power in the considered system, denoted as $P(\mathbf{w}, \mathbf{r})$, can be expressed as

$$\begin{aligned} P(\mathbf{w},\mathbf{r}) &\triangleq P_{\text{cir}}(\mathbf{w}) + P_{\text{SP-FH}}(\mathbf{w},\mathbf{r}) + P_{\text{PA}}(\mathbf{w}) \\ &= \frac{1}{\varepsilon}\sum_{i\in\mathcal{R}}\sum_{m=1}^{M} \|\tilde{\mathbf{w}}_{i,m}\|_2 + (P^{\text{act}} - P^{\text{sl}})\sum_{i\in\mathcal{R}} \mathbb{I}\left(\sum_{k\in\mathcal{K}} \|\mathbf{w}_{i,k}\|_2^2\right) \\ &\quad + \delta \sum_{i\in\mathcal{R}}\sum_{k\in\mathcal{K}} \mathbb{I}\left(\sum_{k\in\mathcal{K}} \|\mathbf{w}_{i,k}\|_2^2\right) r_k + \underbrace{RP^{\text{sl}} + P_0}_{\triangleq P_{\text{c}}}. \end{aligned} \tag{11.21}$$

11.3.5 Problem formulation

We focus on maximizing the overall network energy efficiency of the C-RAN while satisfying the user-specific QoS constraints, fronthaul constraints, and power constraints. In particular, the joint design problem of beamforming, data rate, and RRH-user association and RRH selection design is expressed as

$$\underset{\mathbf{w},\mathbf{r}}{\text{maximize}} \quad \frac{\sum_{k\in\mathcal{K}} r_k}{P(\mathbf{w},\mathbf{r})} \tag{11.22a}$$

$$\text{subject to} \quad r_k \le \log(1+\gamma_k(\mathbf{w})),\ \forall k \in \mathcal{K} \tag{11.22b}$$

$$r_k \ge r_0,\ \forall k \in \mathcal{K} \tag{11.22c}$$

$$\sum_{k\in\mathcal{K}} \mathbb{I}(\|\mathbf{w}_{i,k}\|_2) r_k \le \bar{C},\ \forall i \in \mathcal{R} \tag{11.22d}$$

$$\sum_{k\in\mathcal{K}} \|\mathbf{w}_{i,k}\|_2^2 \le \bar{P},\ \forall i \in \mathcal{R} \tag{11.22e}$$

$$\|\tilde{\mathbf{w}}_{i,m}\|_2^2 \le P^{\text{ant}},\ \forall i \in \mathcal{R}, m = 1,\ldots,M. \tag{11.22f}$$

where r_0 is the predefined data rate threshold of a user. Problem (11.22) is a general nonsmooth nonconvex optimization problem due to the nonconvex objective (11.22a) and the constraints (11.22b) and (11.22d). In fact, by means of the RRH-user association, the joint beamforming optimization in (11.22) can be viewed as finding group sparse solutions of $\mathbf{w}$. Hence we can apply some sort of sparse optimization algorithms to solve (11.22) [25,36].

Nevertheless, (11.22) can be alternatively cast as a mixed integer program to which a wider class of optimization tools are available. In particular, for beamforming designs in wireless communications, indicator function (11.15) can be simplified by the following constraints:

$$\|\mathbf{w}_{i,k}\|_2 \le x_{i,k}\sqrt{\bar{P}}, x_{i,k} \in \{0,1\},\ \forall i \in \mathcal{R}, k \in \mathcal{K}, \tag{11.23a}$$

$$s_i \ge x_{i,k}, \forall k \in \mathcal{K};\ s_i \le \sum_{k\in\mathcal{K}} x_{i,k}, s_i \in \{0,1\},\ \forall i \in \mathcal{R}, \tag{11.23b}$$

where $x_{i,k}$ can be seen as a preference variable representing the association between RRH i and user k, i.e., RRH i does not transmit to user k if $x_{i,k} = 0$, and it does if $x_{i,k} = 1$. Clearly we can see by (11.23a) that $\|\mathbf{w}_{i,k}\|_2 = 0$ if $x_{i,k} = 0$, and $\|\mathbf{w}_{i,k}\|_2 > 0$ otherwise. Similarly, s_i is the preference variable representing the operating state of RRH i, i.e., $s_i = 0$ if RRH i is in sleep mode, and $s_i = 1$ if RRH i is active. Constraint (11.23b) is to tighten the relationship between s_i and $x_{i,k}$, i.e., $s_i = 1$ if there exists (i,k) such that $x_{i,k} = 1$, and $s_i = 0$ otherwise. In other words, RRH i is active if it transmits information to at least one user and sleeps if having no data to transmit.

Based on the previous discussion, we can replace indicator $\mathbb{I}(\cdot)$ by binary variables $x_{i,k}$ and s_i and recast the considered problem as

$$\underset{\mathbf{w},\mathbf{x},\mathbf{s},\mathbf{r}}{\text{maximize}} \quad \frac{\sum_{k\in\mathscr{K}} r_k}{P(\mathbf{w},\mathbf{r},\mathbf{s},\mathbf{x})} \tag{11.24a}$$

$$\text{subject to} \quad r_k \le \log(1+\gamma_k(\mathbf{w})),\ \forall k \in \mathscr{K} \tag{11.24b}$$

$$r_k \ge r_0,\ \forall k \in \mathscr{K} \tag{11.24c}$$

$$\sum_{k\in\mathscr{K}} x_{i,k} r_k \le \bar{C},\ \forall i \in \mathscr{R} \tag{11.24d}$$

$$\|\mathbf{w}_{i,k}\|_2 \le x_{i,k}\sqrt{\bar{P}}, x_{i,k} \in \{0,1\},\ \forall k \in \mathscr{K}, \forall i \in \mathscr{R} \tag{11.24e}$$

$$\sum_{i\in\mathscr{R}} \|\mathbf{w}_{i,k}\|_2^2 \le \bar{P},\ \forall i \in \mathscr{R} \tag{11.24f}$$

$$\|\tilde{\mathbf{w}}_{i,m}\|_2^2 \le P^{\text{ant}},\ \forall i \in \mathscr{R} \tag{11.24g}$$

$$s_i \ge x_{i,k}, \forall k \in \mathscr{K};\ s_i \le \sum_{k\in\mathscr{K}} x_{i,k},\ s_i \in \{0,1\}, \tag{11.24h}$$

where $\mathbf{x} \triangleq [x_{1,1},\ldots,x_{i,k},\ldots,x_{R,K}]^{\mathrm{T}}$, $\mathbf{s} = [s_1,\ldots,s_R]^{\mathrm{T}}$, and

$$P(\mathbf{w},\mathbf{r},\mathbf{s},\mathbf{x}) \triangleq \frac{1}{\varepsilon}\sum_{i\in\mathscr{R}}\sum_{i=1}^{M} \|\tilde{\mathbf{w}}_{i,m}\|_2 + (P^{\text{act}} - P^{\text{sl}})\sum_{i\in\mathscr{R}} s_i + \delta\sum_{i\in\mathscr{R}}\sum_{k\in\mathscr{K}} x_{i,k} r_k + P_{\text{c}}.$$

The new formulation is still a nonconvex optimization problem. However, (11.24) allows the application of global optimization framework such as the ones in Section 11.2. In the following sections, we first introduce an optimal algorithm based on the BRnB procedure to solve (11.24). Then efficient algorithms achieving near-optimal performance are presented; two of which are derived based on(11.24) and the other is based on (11.22).

11.4 Energy-efficient methods for cloud radio access networks

11.4.1 Globally optimal solution via BRnB algorithm

The BRnB algorithm cannot be applied to (11.24) since the optimization problem is not in a standard monotonic form. In particular, we can quickly observe that the fractional objective in (11.24a) is not an increasing function of the involved variables. Thus, the idea is to apply the BRnB on an alternative problem, which admits a monotonic formulation while sharing the same optimal solution set with (11.24). To this end, we consider the following formulation:

$$\underset{\eta,\mathbf{w},\mathbf{x},\mathbf{s},\mathbf{r},\mathbf{t}}{\text{maximize}} \quad \eta \tag{11.25a}$$

$$\text{subject to} \quad \eta P(\mathbf{s},\mathbf{x},\mathbf{r},\mathbf{t}) - \sum_{k\in\mathscr{K}} r_k \le 0 \tag{11.25b}$$

$$\sum_{m=1}^{M} ||\tilde{\mathbf{w}}_{i,m}||_2 \le t_i, \ \forall i \in \mathscr{R} \tag{11.25c}$$

$$\text{(11.24b)–(11.24h)}, \tag{11.25d}$$

where η and $\mathbf{t} \triangleq [t_1,\ldots,t_R]^{\mathrm{T}}$ are newly introduced variables and $P_{\text{total}}(\mathbf{w},\mathbf{r},\mathbf{s},\mathbf{x})$ is redefined as $P(\mathbf{s},\mathbf{x},\mathbf{r},\mathbf{t}) \triangleq \frac{1}{\varepsilon}\sum_{i\in\mathscr{R}} t_i + (P^{\text{act}} - P^{\text{sl}})\sum_{i\in\mathscr{R}} s_i + \delta \sum_{i\in\mathscr{R}}\sum_{k\in\mathscr{K}} x_{i,k} r_k + P_{\text{c}}$. We can easily justify that (11.25b) and (11.25c) hold equality at the optimality that guarantees the equivalence between (11.24) and (11.25) in terms of optimal solution set. The application of the BRnB to (11.25) is inspired by the following key observations: (i) objective (11.25a) and constraints (11.24c), (11.24d), (11.24h), and (11.25b) are monotone with respect to η, $\mathbf{x}$, $\mathbf{s}$, $\mathbf{r}$, and $\mathbf{t}$; (ii) given feasible values of $(\mathbf{s},\mathbf{x},\mathbf{r},\mathbf{t})$, we can determine the corresponding beamforming vector $\mathbf{w}$ by the following lemma.

Lemma 11.1. *Let* $(\hat{\mathbf{x}},\hat{\mathbf{s}},\hat{\mathbf{r}},\hat{\mathbf{t}},\hat{\mathbf{w}})$ *be a feasible point of (11.25). Given the values of* $(\hat{\mathbf{x}},\hat{\mathbf{s}},\hat{\mathbf{r}},\hat{\mathbf{t}})$*, then* $\hat{\mathbf{w}}$ *can be computed as*

$$\hat{\mathbf{w}} = \text{find}\{\mathbf{w}|(11.24\text{b}),(11.24\text{e})\text{–}(11.24\text{g}),(11.25\text{c})\}, \tag{11.26}$$

in which we replace $(\mathbf{s},\mathbf{x},\mathbf{r},\mathbf{t})$ *by* $(\hat{\mathbf{s}},\hat{\mathbf{x}},\hat{\mathbf{r}},\hat{\mathbf{t}})$.

The proof of Lemma 11.1 can be found in [10]. Based on these observations, we can develop a BRnB procedure to find optimal solution $(\mathbf{x},\mathbf{s},\mathbf{r},\mathbf{t},\eta)$ of (11.25). The optimal beamforming $\mathbf{w}$ can be derived accordingly by means of Lemma 11.1. This is the central idea of the proposed algorithm as described next.

To apply the BRnB method introduced in Section 11.2.1, we first determine the monotonic problem with respect to $(\mathbf{s},\mathbf{x},\mathbf{r},\mathbf{t},\eta)$, i.e.,

$$\underset{\eta,\mathbf{x},\mathbf{s},\mathbf{r},\mathbf{t}}{\text{maximize}} \quad \eta \tag{11.27a}$$

$$\text{subject to} \quad (11.24\text{c}),(11.24\text{d}),(11.24\text{h}),(11.25\text{b}) \tag{11.27b}$$

$$[\mathbf{s},\mathbf{x},\mathbf{r},\mathbf{t},\eta] \in [\mathbf{a},\mathbf{b}], \tag{11.27c}$$

in which $\mathbf{a} \triangleq [\underline{\mathbf{s}},\underline{\mathbf{x}},\underline{\mathbf{r}},\underline{\mathbf{t}},\underline{\eta}]$ and $\mathbf{b} \triangleq [\bar{\mathbf{s}},\bar{\mathbf{x}},\bar{\mathbf{r}},\bar{\mathbf{t}},\overline{\eta}]$ define the lower and upper bounds of the initial box. Vertices in $\mathbf{a}$ and $\mathbf{b}$ are calculated as follows. Since $\mathbf{s}$ and $\mathbf{x}$ are binary variables, hence $\underline{s}_b = 0, \bar{s}_b = 1, \underline{x}_{b,k} = 0, \bar{x}_{b,k} = 1$. By (11.22c), we have $r_k \geq \underline{r}_k = r_0$, and by (11.22b)

$$r_k \leq \bar{r}_k = \min\left\{\bar{C}, \log\left(1 + |\mathbf{h}_k\mathbf{w}_k|^2/\sigma_k^2\right)\right\} \leq \min\left\{\bar{C}, \log\left(1 + R\bar{P}\|\mathbf{h}_k\|_2^2/\sigma_k^2\right)\right\},$$

as $|\mathbf{h}_k\mathbf{w}_k|^2 \leq \|\mathbf{h}_k\|_2^2\|\mathbf{w}_k\|_2^2$ by the Cauchy–Schwarz inequality, and $\|\mathbf{w}_k\|_2^2 \leq R\bar{P}$. We also have $t_i \geq \underline{t}_i = 0$ and $t_i \leq \bar{t}_i = M\sqrt{P^{\text{ant}}}$; and by (11.25b)

$$\underline{\eta} \triangleq \frac{\sum_{k\in\mathcal{K}} \underline{r}_k}{P(\bar{\mathbf{s}}, \bar{\mathbf{x}}, \bar{\mathbf{r}}, \bar{\mathbf{t}})} \quad \text{and} \quad \bar{\eta} \triangleq \frac{\sum_{k\in\mathcal{K}} \bar{r}_k}{P(\underline{\mathbf{s}}, \underline{\mathbf{x}}, \underline{\mathbf{r}}, \underline{\mathbf{t}})}. \tag{11.28}$$

Since feasible set of (11.27) is upper bounded by the power and fronthaul constraints, it satisfies the normal and finite properties required by the BRnB algorithm (see Definition 11.3), thus, (11.27) is now in standard form of MO problem and can be solved by the BRnB method.

Remark that our aim is to find optimal solutions of (11.25) through solving (11.27). Therefore, we need to ensure that optimal solutions obtained by the BRnB method are feasible to (11.25), thereby the desired optimal solutions can be found by Lemma 11.1. To this end, every newly created box needs to be checked if potentially containing feasible solutions to (11.25), otherwise that box should be eliminated. This can be done by solving the following feasibility problem for a box V:

$$\underset{\mathbf{w}}{\text{minimize}} \quad \sum_{i\in\mathcal{R}}\sum_{m=1}^{M} \|\tilde{\mathbf{w}}_{i,m}\|_2 \tag{11.29a}$$

$$\text{subject to} \quad \mathbf{h}_k\mathbf{w}_k \geq \sqrt{(e^{\underline{r}_k} - 1)\left(\sum_{j\neq k}^{K} |\mathbf{h}_k\mathbf{w}_j|^2 + \sigma_k^2\right)} \tag{11.29b}$$

$$\underline{s}_i\underline{t}_i \leq \sum_{m=1}^{M} ||\tilde{\mathbf{w}}_{i,m}||_2 \leq \bar{s}_i\bar{t}_i, \ i \in \mathcal{R} \tag{11.29c}$$

$$\|\tilde{\mathbf{w}}_{i,m}\|_2^2 \leq \bar{s}_i P^{\text{ant}}, \ \|\mathbf{w}_{i,k}\|_2 \leq \bar{x}_{i,k}\sqrt{\bar{P}}, i \in \mathcal{R} \tag{11.29d}$$

$$\sum_{k\in\mathcal{K}} \|\mathbf{w}_{i,k}\|_2^2 \leq \bar{s}_i\bar{P} \ \forall i \in \mathcal{R}. \tag{11.29e}$$

Specifically, (11.29) can be viewed as minimizing the power consumption subject to minimum users' rate requirement $\underline{\mathbf{r}}$ given the power budget $\bar{\mathbf{t}}$ and RRH-user association $\bar{\mathbf{x}}$ and $\bar{\mathbf{s}}$. Obviously, if there does not exist any beamformer set making $\underline{\mathbf{r}}$ achievable, then V is infeasible to (11.25) and needs to be immediately discarded.

To summarize, the globally optimal method solving (11.25) is described in Algorithm 11.2. Noticeably, we have made modifications to the standard BRnB procedure in the algorithm so as to accelerate the convergence rate of Algorithm 11.2. These

Algorithm 11.2: The BRnB algorithm solving (11.25)

```
**Initialization:** Compute $\mathbf{a}$, $\mathbf{b}$ and apply box reduction to box $[\mathbf{a};\mathbf{b}]$. Let $n := 1$,
$\mathscr{B}_1 = \mathrm{r}([\mathbf{a};\mathbf{b}])$ and $f_{\mathrm{UpB}}(B) = \underline{\eta}$
**repeat** $\{n := n + 1\}$
    **Branching:** select a box $V_{\mathrm{c}} = [\mathbf{p};\mathbf{q}] \subset \mathscr{B}_{n-1}$ and branch $V_{\mathrm{c}}$ into two smaller
    ones $V_{\mathrm{c}}^1$ and $V_{\mathrm{c}}^2$, then remove $V_{\mathrm{c}}$ from $\mathscr{B}_{n-1}$.
    **Reduction:** apply box reduction to each box $V_{\mathrm{c}}^l$ $(l = \{1,2\})$ and obtain reduced
    box $\mathrm{r}(V_{\mathrm{c}}^l)$.
    **Bounding:** for each box $\mathrm{r}(V_{\mathrm{c}}^l)$
    **if** solving (11.29) is feasible **then**
        Achieve $\mathbf{w}^*$, calculate $\mathbf{t}^*$ and extract $\mathbf{x}^*$.
        Update $\underline{\mathbf{t}} := \mathbf{t}^*$ and calculate $f_{\mathrm{UpB}}(\mathrm{r}(V_{\mathrm{c}}^l))$ by (11.31).
        Check $\mathbf{x}^*$ with (11.33), if true, compute feasible objective $\hat{\eta}$ as (11.32), and
        update new CBO, i.e., CBO $:= \max\{\hat{\eta}, \mathrm{CBO}\}$.
        Update $\mathscr{B}_n := \mathscr{B}_{n-1} \cup \{\mathrm{r}(V_{\mathrm{c}}^l) | f_{\mathrm{UpB}}(\mathrm{r}(V_{\mathrm{c}}^l)) \geq \mathrm{CBO}\}$.
    **end if**
**until** Convergence
**Output:** With $(\mathbf{x}^*, \mathbf{s}^*, \mathbf{r}^*, \mathbf{t}^*)$, recover $\mathbf{w}^*$ by (11.26) to achieve the globally optimal
solution of (11.24), i.e., $(\mathbf{w}^*, \mathbf{x}^*, \mathbf{s}^*, \mathbf{r}^*)$.
```

changes exploit some useful properties of the energy efficiency maximization in the C-RAN. The modifications (compared to the generic framework) made in Algorithm 11.2 are presented next.

Improved branching dimension

Normally, one entry of $[\mathbf{s}, \mathbf{x}, \mathbf{r}, \mathbf{t}, \eta]$ is branched at each iteration, and, thus, the total number of iterations may increase exponentially with the problem size. Intuitively, we can lower the computational complexity by minimizing the branching dimensions while still guaranteeing convergence. For (11.25), it turns out that we can skip branching on η and $\mathbf{t}$. In particular, recall that lower and upper bounds of η can be determined via those of $[\mathbf{s}, \mathbf{x}, \mathbf{r}, \mathbf{t}]$ as (11.28). Thus, branching on η is not essential. On the other hand, consider problem (11.29) and denote by $\mathbf{w}^*$ the optimal solution if solving (11.29) is successful, and $\mathbf{t}^* \triangleq \{t_i^*\}_i$ with $t_i^* = \sum_{m=1}^{M} \|\tilde{\mathbf{w}}_{i,m}^*\|_2$. Obviously, $\mathbf{t}^*$ is the minimum power required to achieve $\underline{\mathbf{r}}$, and it holds $\underline{\mathbf{t}} \leq \mathbf{t}^*$. Also, $\mathbf{t}^*$ is unique solution because of the structure of the objective in (11.29) [37, Chapter 3]. At this point, we can replace $\underline{\mathbf{t}}$ by $\mathbf{t}^*$ to yield a tighter lower bound on $\mathbf{t}$. Thus, it is sufficient to only branch $(\mathbf{x}, \mathbf{s}, \mathbf{r})$ as the lower bound on $\mathbf{t}$ is always improved with $\underline{\mathbf{r}}$. The presented properties significantly accelerate the convergence of the BRnB.

Improved branching order

In principle, we can randomly select a variable to perform branching in each iteration of the BRnB algorithm, because this operation does not delete any feasible solution. For the considered problem, we can potentially reduce the computational complexity if we start the algorithm by first branching on $\mathbf{s}$ due to its dependency on other factors. Intuitively, the number of active RRHs provides the degrees of freedom that can make the desired data rate $\underline{\mathbf{r}}$ achievable. Moreover, we can immediately obtain $x_{i,k} = 0, \forall k \in \mathscr{K}$ whenever $s_i = 0$, implying that the effective dimension in V is reduced by K times. Therefore, by first keeping branching on $\mathbf{s}$ until $\underline{\mathbf{s}} = \bar{\mathbf{s}}$, we can quickly determine that the combinations of $\{s_i\}_i$ make (11.25) infeasible. This is done by solving (11.29) with given $\bar{\mathbf{s}}$ and target rate r_0 for all users. Moreover, since the length of $\mathbf{s}$ is much smaller than that of $\mathbf{x}$ in most of wireless communications applications, it is expected that branching on $\mathbf{s}$ may take a relatively small number of iterations and, thus, may create a relatively small number of new boxes.

Improved upper bound

Upper bounding as in (11.28) is not tight enough and thus is inefficient for the considered problem. To be specific, let us recall the lower bound of $P(\mathbf{s}, \mathbf{x}, \mathbf{r}, \mathbf{t})$, i.e.:

$$P(\underline{\mathbf{s}}, \underline{\mathbf{x}}, \underline{\mathbf{r}}, \underline{\mathbf{t}}) = \frac{1}{\varepsilon}\sum_{i\in\mathscr{R}} \underline{t}_i + (P^{\mathrm{act}} - P^{\mathrm{sl}})\sum_{i\in\mathscr{R}} \underline{s}_i + \delta \sum_{i\in\mathscr{R}} \sum_{k\in\mathscr{K}} \underline{x}_{i,k}\underline{r}_k + P_{\mathrm{c}}, \qquad (11.30)$$

and observe that $P(\underline{\mathbf{s}}, \underline{\mathbf{x}}, \underline{\mathbf{r}}, \underline{\mathbf{t}}) = \sum_{i\in\mathscr{R}} \underline{t}_i/\varepsilon + P_{\mathrm{c}}$ if $\underline{\mathbf{x}} = \mathbf{0}$ and $\underline{\mathbf{s}} = 0$. However, it holds that $\sum_{i\in\mathscr{R}} s_i \geq 1$, since at least one RRH is active for transmission, and $\sum_{i\in\mathscr{R}} \sum_{k\in\mathscr{K}} x_{i,k} r_k \geq \sum_{k\in\mathscr{K}} \underline{r}_k$ because the fronthaul networks must at least convey the minimum required data rate for all users. In other words, the second and third terms in (11.30) are always nonzero. Moreover, from the result of solving (11.29) feasible, we have $\underline{\mathbf{t}} \leq \mathbf{t}^*$. Thus, we can upper tighten the bound of $P(\underline{\mathbf{s}}, \underline{\mathbf{x}}, \underline{\mathbf{r}}, \underline{\mathbf{t}})$ as

$$\begin{aligned} P(\underline{\mathbf{s}}, \underline{\mathbf{x}}, \underline{\mathbf{r}}, \underline{\mathbf{t}}) \leq \underline{P}(\underline{\mathbf{s}}, \underline{\mathbf{x}}, \underline{\mathbf{r}}, \mathbf{t}^*) \triangleq\ & \sum_{i\in\mathscr{R}} \frac{1}{\varepsilon} t_i^* + (P^{\mathrm{act}} - P^{\mathrm{sl}}) \max\left\{1, \sum_{i\in\mathscr{R}} \underline{s}_i\right\} \\ & + \delta \max\left\{\sum_{k\in\mathscr{K}} \underline{r}_k, \sum_{i\in\mathscr{R}, k\in\mathscr{K}} \underline{x}_{i,k}\underline{r}_k\right\} + P_{\mathrm{c}}. \end{aligned}$$

One can see that replacing $P(\underline{\mathbf{s}}, \underline{\mathbf{x}}, \underline{\mathbf{r}}, \underline{\mathbf{t}})$ by $\underline{P}(\underline{\mathbf{s}}, \underline{\mathbf{x}}, \underline{\mathbf{r}}, \mathbf{t}^*)$ does not remove any feasible solution. As such a tighter upper bound on η over V can be recalculated as

$$\overline{\eta} = \frac{\sum_{k\in\mathscr{K}} \overline{r}_k}{\underline{P}(\underline{\mathbf{s}}, \underline{\mathbf{x}}, \underline{\mathbf{r}}, \mathbf{t}^*)}. \qquad (11.31)$$

Updating the best current objective

As mentioned earlier in Section 11.2.1, boxes, the upper bounds of which are smaller than the CBO, can be removed out of set $\mathscr{B}_n$ to reduce the complexity. Thus, efficiently updating the CBO is deemed important for improving the algorithm's efficiency. To do so, we need to find a feasible point within each box in order to calculate the feasible objective value. In particular, let $[\hat{\mathbf{s}}, \hat{\mathbf{x}}, \hat{\mathbf{r}}, \hat{\mathbf{t}}]$ be some feasible point in box V. The feasible objective is

$$\hat{\eta} = \frac{\sum_{k\in\mathscr{K}} \hat{r}_k}{P(\hat{\mathbf{s}}, \hat{\mathbf{x}}, \hat{\mathbf{r}}, \hat{\mathbf{t}})}. \tag{11.32}$$

If $\hat{\eta} >$ CBO, $\hat{\eta}$ is updated to be the new CBO. We now present a heuristic method to obtain a feasible point within box V. We first recall the feasibility problem (11.29) and note that $\underline{\mathbf{r}}$ is feasible if solving (11.29) is so. Thus, our idea is to set $\hat{\mathbf{r}} = \underline{\mathbf{r}}$ and determine the values of the other variables $(\hat{\mathbf{x}}, \hat{\mathbf{s}}, \hat{\mathbf{t}})$ corresponding to $\underline{\mathbf{r}}$. Particularly for $\hat{\mathbf{x}}$, we can simply compute $\hat{\mathbf{x}}$ by setting $\hat{x}_{i,k} = 0$ if $\|\mathbf{w}^*_{i,k}\|_2 = 0$ and vice versa $\hat{x}_{i,k} = 1$ if $\|\mathbf{w}^*_{i,k}\|_2 > 0$ where $\mathbf{w}^*$ is an optimal solution obtained by solving (11.29). The obtained solution is verified feasible if it stays in the feasible set of (11.27), i.e.,

$$\hat{\mathbf{x}} \in \left\{ \mathbf{x} \mid \sum_{i\in\mathscr{R}} x_{i,k} \geq 1, k \in \mathscr{K}, \sum_{k\in\mathscr{K}} x_{i,k}\underline{r}_k \leq \bar{C}, i \in \mathscr{R} \right\}. \tag{11.33}$$

The value of $\hat{\mathbf{s}}$ is found based on $\hat{\mathbf{x}}$ according to (11.24h), while the value of $\hat{\mathbf{t}}$ is $\hat{t}_i = \sum_{m=1}^{M} \|\tilde{\mathbf{w}}^*_{i,m}\|_2, \forall i \in \mathscr{R}$.

We can easily check that these modifications made in Algorithm 11.2 do not drop off any feasible solution. As a result, Algorithm 11.2 is still guaranteed to converge and its output is the globally optimal solutions of (11.24).

11.4.2 Suboptimal solutions via successive convex approximation

In general, a global optimization algorithm often takes enormous complexity to output a solution that is in fact not practically preferable. Instead, it often serves as the performance benchmark to evaluate the quality of other methods used in practice. Toward more efficient algorithms, we next present three iterative suboptimal approaches that aim at achieving near-optimal solutions but using much reduced computation. These approaches apply different techniques to deal the binary variables, before employing the SCA framework in Section 11.2.2 to find a stationary solution of the design problem. Noticeably, the first two methods are developed in accordance with formulation (11.24), while the third method is based on (11.22).

11.4.2.1 SCA-based mixed integer programming

In the first method, we keep the binary variables unchanged while using the SCA framework to convexify the nonconvex parts of problem (11.24) then solved using modern mixed integer convex solvers. We shall refer this algorithm as SCA-MI in the sequel.

To apply the SCA, recall that the nonconvexity must have a convex bound that satisfies the conditions in (11.8). However, this is not the case of (11.24). Thus, we first need to transform the problem into an equivalent form suitable for application of the SCA. To be simple, we shall consider the following transformation of (11.24) of which the nonconvex parts have similar formulations with those introduced in Section 11.2.2, i.e.:

$$\underset{\substack{t,\mathbf{w},\mathbf{s},\mathbf{x},\\ \mathbf{r},\mathbf{g},\mathbf{q},\mathbf{v}}}{\text{maximize}} \quad \frac{\sum_{k\in\mathscr{K}} r_k}{t+\delta\sum_{i\in\mathscr{R}}\sum_{k\in\mathscr{K}} v_i^2} \tag{11.34a}$$

$$\text{subject to} \quad t \geq \frac{1}{\varepsilon}\sum_{i\in\mathscr{R}}\sum_{m=1}^{M}\|\tilde{\mathbf{w}}_{i,m}\|_2 + \sum_{i\in\mathscr{R}}(P^{\text{act}}-P^{\text{sl}})s_i + P_{\text{c}} \tag{11.34b}$$

$$\log(1+g_k) \geq r_k \ \forall k \in \mathscr{K} \tag{11.34c}$$

$$\sum_{j\neq k}^{K} |\mathbf{h}_k\mathbf{w}_j|^2 + \sigma_k^2 \leq \frac{|\mathbf{h}_k\mathbf{w}_k|^2}{g_k}, \ \forall k \in \mathscr{K} \tag{11.34d}$$

$$x_{i,k} \leq \frac{v_{i,k}^2}{r_k}, \ \forall k \in \mathscr{K}, \forall i \in \mathscr{R} \tag{11.34e}$$

$$\sum_{k\in\mathscr{K}} v_{i,k}^2 \leq \bar{C}, \ \forall i \in \mathscr{R} \tag{11.34f}$$

$$(11.24\text{c}), (11.24\text{e})\text{–}(11.24\text{h}), \tag{11.34g}$$

where t, $\mathbf{g} \triangleq [g_1,\ldots,g_K]^{\text{T}}$, $\mathbf{q} \triangleq [q_1,\ldots,q_K]^{\text{T}}$, $\mathbf{v} \triangleq [v_1,\ldots,v_R]^{\text{T}}$ are newly introduced slack variables. Specifically, the objective (11.24a) is equivalently rewritten by (11.34a) and (11.34b) with t; (11.24b) is rewritten by (11.34c) and (11.34d) using $\mathbf{g}$; (11.24d) is replaced by (11.34e) and (11.34f) using $\mathbf{v}$. It is easily checked that (11.24) and (11.34) are equivalent in terms of the optimal set. We keep rewriting (11.34) as

$$\underset{\boldsymbol{\theta}\in\mathscr{S}}{\text{minimize}} \quad f(\boldsymbol{\theta}) \triangleq \frac{t}{\sum_{k\in\mathscr{K}} r_k} + \frac{\delta\sum_{i\in\mathscr{R}}\sum_{k\in\mathscr{K}} v_i^2}{\sum_{k\in\mathscr{K}} r_k} \tag{11.35a}$$

$$\text{subject to} \quad (11.34\text{d}), (11.34\text{e}), \tag{11.35b}$$

where $\boldsymbol{\theta} \triangleq \{t,\mathbf{w},\mathbf{s},\mathbf{x},\mathbf{r},\mathbf{g},\mathbf{q},\mathbf{v}\}$ denotes the involved optimization variables and

$$\mathscr{S} \triangleq \{\boldsymbol{\theta}|(11.24\text{c}), (11.24\text{e})\text{–}(11.24\text{h}), (11.34\text{b}), (11.34\text{c}), (11.34\text{f})\},$$

is the set of convex constraints of (11.34). The nonconvexity of (11.35a) and (11.35b) is obvious due to the first term of objective and the right side of (11.34d) and (11.34e). These are the SCA-applicable functions (cf. Section 11.2.2). The application of the SCA to (11.35a) and (11.35b) is now straightforward. In particular, the nonconvex constraints can be replaced by the following convex ones:

$$\sum_{j\neq k}^{K} |\mathbf{h}_k\mathbf{w}_j|^2 + \sigma_k^2 \leq \phi^{\text{qol}}\left(\mathbf{w}_k, g_k; \mathbf{w}_k^{(n)}, g_k^{(n)}; \mathbf{h}_k^{\text{H}}\mathbf{h}_k\right) \tag{11.36}$$

$$x_{i,k} \leq \phi^{\text{qol}}\left(v_{i,k}, r_k; v_{i,k}^{(n)}, r_k^{(n)}\right), \ \forall i \in \mathscr{R}, k \in \mathscr{K}, \tag{11.37}$$

in which the right side of (11.34d) and (11.34e) have been approximated using (11.11). For the objective, the nonconvexity can be approximated by the function in (11.10). As a result, we have the following approximate objective:

$$f(\boldsymbol{\theta};\boldsymbol{\theta}^{(n)}) \triangleq \phi^{\text{frac}}\left(t;\sum_{k\in\mathcal{K}} r_k;\lambda^{(n)}\right) + \frac{\delta\sum_{i\in\mathcal{R}}\sum_{k\in\mathcal{K}} v_{i,k}^2}{\sum_{k\in\mathcal{K}} r_k}, \tag{11.38}$$

where $\lambda^{(n)} = \frac{1}{t^{(n)}\sum_{k\in\mathcal{K}} r_k^{(n)}}$.

In summary, we solve the following approximate convex program of (11.35a) and (11.35b) at every iteration of the SCA procedure outlined in Algorithm 11.3

$$\underset{\boldsymbol{\theta}\in\mathcal{S}}{\text{minimize}} \quad f(\boldsymbol{\theta};\boldsymbol{\theta}^{(n)}) \quad \text{subject to} \quad \{(11.36),(11.37)\}. \tag{11.39}$$

The convergence of Algorithm 11.3 can be proved following the arguments in Section 11.2.2 and those in the related references. Problem (11.39) is a mixed integer convex program that can be solved effectively by off-the-shelf convex solvers such as MOSEK [38].

It is interesting to see that most of constraints in (11.39) are second-order-cone (SOC) representable except the logarithmic constraint in (11.34c) (cf. [39] for SOC-presentable formulations). Thus, the complexity of solving (11.39) would remarkably be reduced if (11.34c) can be approximated by SOC ones. For this purpose, we can use approaches presented in [40]. By applying such method, the resulting problem becomes SOC programming (SOCP) that allows more available off-the-shelf convex solvers, such as CPLEX, GUROBI, applicable to solve it with lower complexity.

The SCA-MI uses a significantly smaller number of iterations to find the solutions compared to the BRnB method. In addition, it can achieve exact binary solutions. However, since the SCA-MI method deals directly with binary variables in each iteration, the per-iteration complexity is still relatively high, and so is the total run time, as we shall see via numerical demonstrations. In the following, we present the second and third approaches that have further lower per-iteration complexity.

Algorithm 11.3: Proposed method for solving (11.34)

```
Initialization: Set $n := 0$, choose initial values for $\boldsymbol{\theta}^{(0)}$
repeat $\{n := n + 1\}$
    Solve (11.39) and achieve $\boldsymbol{\theta}^*$
    Update $\boldsymbol{\theta}^{(n)} := \boldsymbol{\theta}^*$
until Convergence
```

11.4.2.2 SCA-based regularization method

The second method also aims at using the SCA framework to solve (11.24), achieving exact binary solutions. Unlike the SCA-MI method, the binary variables here are equivalently represented by a constraint of continuous variables. A regularization technique is then applied to make the resulting problem to be SCA applicable. More specifically, we first use the well-known polynomial constraint of binary variables for $\mathbf{x}$ which is given by

$$x_{i,k} \in \{0,1\}, \forall i,k \Leftrightarrow \sum_{i\in\mathscr{R}} \sum_{k\in\mathscr{K}} x_{i,k}^2 - x_{i,k} \geq 0,\ x_{i,k} \in [0,1]. \tag{11.40}$$

We can adjust the previous representation by the fact that $x_{i,k}^2 - x_{i,k} < 0$ for $x_{b,k} \in (0,1)$. For $\mathbf{s}$, we simply relax it as $s_i \in [0,1]$ because s_i is automatically binary when $x_{b,k}$ is so, which is due to (11.24h). In the sequel of this subsection, $x_{i,k}$ and s_i are seen as continuous over $[0,1]$, $\forall i,k$. Now, to apply the SCA method, we can utilize the transformation presented in the previous subsection which leads to the following equivalent problem of (11.24):

$$\underset{\boldsymbol{\theta}\in\mathscr{S}}{\text{minimize}}\ f(\boldsymbol{\theta}) \quad \text{subject to } \{x_{i,k}, s_i \in [0,1],\ (11.40), (11.34\text{d}), (11.34\text{e})\}, \tag{11.41}$$

while two nonconvex constrains (11.34d) and (11.34e) can be handled by the same way as done in Section 11.4.2.1. Unfortunately, (11.40) is not suitable for the use of the SCA. This is because its convex approximation does not satisfy Slater's constraint qualification, which is the necessary condition for the SCA. To understand this, let us apply the SCA principle to (11.40) resulting in the following approximate constraint:

$$\phi^{\text{po}}(\mathbf{x};\mathbf{x}';2) \triangleq \sum_{i\in\mathscr{R}} \sum_{k\in\mathscr{K}} \left(2x_{i,k}x'_{i,k} - (x'_{i,k})^2\right) - \sum_{i\in\mathscr{R}} \sum_{k\in\mathscr{K}} x_{i,k} \geq 0, \tag{11.42}$$

where $x'_{i,k} \in [0,1]$ is a feasible point in (11.41). It is not difficult to check that the set $\{x_{i,k} \in (0,1) | \phi^{\text{po}}(\mathbf{x};\mathbf{x}';2) > 0\}$ is empty violating Slater's condition. To cope with this issue, we can apply the regularization technique [41,42]. In particular, auxiliary variable $q \geq 0$ is added to (11.41) to arrive at the following regularized problem:

$$\underset{\boldsymbol{\theta}\in\mathscr{S}}{\text{minimize}}\ \ f(\boldsymbol{\theta}) + \alpha q \tag{11.43a}$$

$$\text{subject to}\ \ \sum_{i\in\mathscr{R}} \sum_{k\in\mathscr{K}} x_{i,k}^2 - x_{i,k} \geq q,\ x_{i,k}, s_i \in [0,1] \tag{11.43b}$$

$$(11.34\text{d}), (11.34\text{e}). \tag{11.43c}$$

In (11.43), the last term in the objective stands for the cost that (11.43b) is violated when $x_{i,k} \neq \{0,1\}$, which is the objective to be minimized; $\alpha > 0$ is the penalty parameter. We can justify that the use of q allows (11.43b) to be satisfied for any

$x_{i,k} \in [0, 1]$, and (11.43) is identical with (11.41) when $q = 0$. At this point, the SCA algorithm can be applied to (11.43) similarly to Section 11.4.2.1, leading to the following convex approximation problem solved at each iteration:

$$\underset{\boldsymbol{\theta}\in\mathscr{S},q}{\text{minimize}} \quad f(\boldsymbol{\theta};\boldsymbol{\theta}^{(n)}) + \alpha q \tag{11.44a}$$

$$\text{subject to} \quad \sum_{i\in\mathscr{R}}\sum_{k\in\mathscr{K}}\Big(2x_{i,k}x_{i,k}^{(n)} - (x_{i,k}^{(n)})^2\Big) - \sum_{i\in\mathscr{R}}\sum_{k\in\mathscr{K}} x_{i,k} \geq q,\ x_{i,k}, s_i \in [0,1] \tag{11.44b}$$

$$(11.36), (11.37). \tag{11.44c}$$

We outline the procedure solving (11.43) in Algorithm 11.4. The convergence of the method can be justified following the discussion in Section 11.2.2. An important point in Algorithm 11.4 is that the value of penalty parameter α is increased at each iteration, i.e., Step 5. We note that a high value of α will encourage q to be zero which means that $x_{i,k}$ $\forall i, k$ are binary values. Thus, the idea is to start Algorithm 11.4 with a small value of α to prioritize minimizing the original objective and then increase α in subsequent iterations to force q to be zero.

Remark: For Algorithm 11.4, our expectation is that the obtained solutions at the convergence shall eventually be exact binary values. To ensure this, we can replace (11.24e) by

$$\|\mathbf{w}_{i,k}\|_2 \leq x_{i,k}^{p}\sqrt{\bar{P}}, \tag{11.45}$$

where p is a constant. This maneuver is inspired by two observations. First, for all value $p > 1$, (11.45) is equivalent to (11.24e) if $x_{b,k} \in \{0, 1\}$. Second, $x_{i,k}^{p} \geq x_{i,k}^{p+1}$ for $x_{i,k} \in [0, 1]$ with $p > 0$ which means that:

$$\|\mathbf{w}_{i,k}\|_2 \leq x_{i,k}^{p+1}\sqrt{\bar{P}} \leq x_{i,k}^{p}\sqrt{\bar{P}} \leq \cdots \leq x_{i,k}\sqrt{\bar{P}}. \tag{11.46}$$

Thus, this replacement is nothing but to tighten the feasible set of (11.41) such that a tighter continuous relaxation can be obtained with higher values of p. Remark that (11.45) for $p > 1$ is nonconvex constraints and its right side is the power function.

Algorithm 11.4: Proposed method for solving (11.43)

```
1: Initialization: Set $n := 0$, choose initial values for $\boldsymbol{\theta}^{(0)}$ and set $\alpha^{(0)}$ small
2: repeat $\{n := n + 1\}$
3:     Solve (11.44) and achieve $\boldsymbol{\theta}^*$
4:     Update $\boldsymbol{\theta}^{(n)} := \boldsymbol{\theta}^*$
5:     Update $\alpha^{(n)} := \min\{\alpha_{\max}; \alpha^{(n-1)} + \varepsilon\}$ for small $\varepsilon$
6: until Convergence
```

In light of the SCA principle, we can replace (11.45) by the following approximate constraint when solving (11.44) (Step 3 in Algorithm 11.4), i.e.,

$$\|\mathbf{w}_{i,k}\|_2 \le \phi^{\mathrm{po}}\left(x_{i,k}; x_{i,k}^{(n)}; p\right)\sqrt{\bar{P}}. \tag{11.47}$$

We note that (11.45) has not been used in the development of the global optimization algorithm and the SCA-MI method, since these methods do not work on the continuous relaxation of the binary variables.

11.4.2.3 SCA-based ℓ_0-approximation method

In the third method, we use the formulation (11.22) to derive solutions for the joint design problem. By viewing the joint design as finding sparse solutions of the beamformers, we can leverage the sparsity-based approach combining with SCA method to solve the considered problem. In particular, the nonsmooth function $\mathbb{I}(\cdot)$ is first approximated by a continuous one using ℓ_0/ℓ_2 norm method [25,43]. As such the SCA for the continuous optimization problem can be applied.

To proceed, we consider the simplified expression of (11.15)

$$\mathbb{I}\big(\|\mathbf{w}_{i,k}\|_2^2\big) \Leftrightarrow \mathbb{I}(u_{i,k}); \mathbb{I}\left(\sum_{k\in\mathcal{K}} \|\mathbf{w}_{i,k}\|_2^2\right) \Leftrightarrow \mathbb{I}\left(\sum_{k\in\mathcal{K}} u_{i,k}\right), \text{ where } \|\mathbf{w}_{i,k}\|_2 \le u_{i,k}, \tag{11.48}$$

and $u_{i,k}$ is the slack variable associated with the power of $\mathbf{w}_{i,k}$. We justify (11.48) by the fact that $u_{i,k} = 0$ if $\|\mathbf{w}_{i,k}\| = 0$ and otherwise. We note that $u_{i,k}$ is additionally introduced merely to make the presentation of the algorithm easier to follow, as one can directly apply the method presented next on $\mathbb{I}(\|\mathbf{w}_{i,k}\|_2^2)$ to find the solutions. Following ℓ_0/ℓ_2 norm method, $\mathbb{I}(u_{i,k})$ can be continuously approximated by some (concave) functions. For example, Table 11.1 lists continuous approximations of the indicator function that have often been used in sparse optimization problems in wireless communications. We can see that $\mathbb{I}(u_{i,k}) \approxeq \psi_\beta(u_{i,k})$ when approximation

Table 11.1 ℓ_0-Approximation function $\psi_\beta(y)$ and the corresponding subgradient $\partial\psi_\beta(y)$ and first-order approximations $\bar{\psi}_\beta(y;y^{(n)})$ [23]

	Function $\psi_\beta(y)$	**Subgradient $\partial\psi_\beta(y)$**	**Approximation $\bar{\psi}_\beta(y;y^{(n)})$**
Exp	$1-\exp(-\beta y)$	$\beta e^{-\beta y}$	$1-e^{-\beta y^{(n)}}(1-\beta(y-y^{(n)}))$
Capped-ℓ_1	$\min\{1, \beta y\}$	$\begin{cases} 0 & \text{if } y \ge \frac{1}{\beta} \\ \beta & \text{if otherwise} \end{cases}$	$\begin{cases} 1 & \text{if } y^{(n)} \ge \frac{1}{\beta} \\ \beta y & \text{otherwise} \end{cases}$

parameter β is sufficiently *large*. Replacing $\mathbb{I}(u_{i,k})$ by $\psi_\beta(u_{i,k})$, we arrive at the following continuously approximated problem (11.22):

$$\underset{\mathbf{w},\mathbf{r},\mathbf{u}}{\text{maximize}} \quad \frac{\sum_{k\in\mathcal{K}} r_k}{P(\mathbf{w},\mathbf{r},\mathbf{u})} \tag{11.49a}$$

$$\text{subject to} \quad \|\mathbf{w}_{i,k}\|_2 \le u_{i,k}, \ \sum_{k\in\mathcal{K}} u_{i,k}^2 \le \bar{P}, \ \forall b \in \mathcal{B} \tag{11.49b}$$

$$\sum_{k\in\mathcal{K}} \psi_\beta(u_{i,k}) r_k \le \bar{C} \tag{11.49c}$$

$$\text{(11.22b), (11.22c), (11.22f),} \tag{11.49d}$$

where $P(\mathbf{w},\mathbf{r},\mathbf{u}) = \sum_{i\in\mathcal{R}}(\frac{1}{\varepsilon}\sum_{i=1}^{M}\|\tilde{\mathbf{w}}_{i,m}\|_2 + (P^{\text{act}} - P^{\text{sl}})\psi_\beta(\sum_{k\in\mathcal{K}} u_{i,k}) + \delta\sum_{k\in\mathcal{K}}$ $\psi_\beta(u_{i,k})r_k) + P_{\text{c}}$. We note that (11.49) is still nonconvex but can be solved following the same manner with those in the previous subsections. In particular, we first recall the transformation:

$$\underset{t,\mathbf{w},\mathbf{r},\mathbf{g},\mathbf{u},\mathbf{v},\tilde{\mathbf{x}},\tilde{\mathbf{s}}}{\text{minimize}} \quad \frac{t}{\sum_{k\in\mathcal{K}} r_k} + \frac{p_{\text{SP}}\sum_{i\in\mathcal{R}}\sum_{k\in\mathcal{K}} v_i^2}{\sum_{k\in\mathcal{K}} r_k} \tag{11.50a}$$

$$\text{subject to} \quad \tilde{x}_{i,k} \ge \psi_\beta(u_{i,k}), \ \forall i \in \mathcal{R}, k \in \mathcal{K} \tag{11.50b}$$

$$\tilde{s}_i \ge \psi_\beta(\textstyle\sum_{k\in\mathcal{K}} u_{i,k}), \ \forall i \in \mathcal{R} \tag{11.50c}$$

$$t \ge \frac{1}{\varepsilon}\sum_{i\in\mathcal{R}}\sum_{k\in\mathcal{K}}\|\tilde{\mathbf{w}}_{i,m}\|_2 + \sum_{i\in\mathcal{R}}(P^{\text{act}} - P^{\text{sl}})\tilde{s}_i + P_{\text{c}} \tag{11.50d}$$

$$\tilde{x}_{i,k} \le \frac{v_{i,k}^2}{r_k}, \ \forall i \in \mathcal{R}, k \in \mathcal{K} \tag{11.50e}$$

$$\sum_{k\in\mathcal{K}} v_{i,k}^2 \le \bar{C}, \ \forall i \in \mathcal{R} \tag{11.50f}$$

$$\text{(11.22c), (11.22f), (11.34c), (11.34d), (11.49b).} \tag{11.50g}$$

In (11.50), $\tilde{\mathbf{x}} \triangleq [\tilde{x}_{1,1},\ldots,\tilde{x}_{i,k},\ldots,\tilde{x}_{R,K}]^{\text{T}}$, $\mathbf{s} = [\tilde{s}_1,\ldots,\tilde{s}_R]^{\text{T}}$ are additionally introduced to facilitate the transformation such that (11.50) has similar form with those nonconvex programs in Sections 11.4.2.1 and 11.4.2.2. Second, we apply the SCA to solve the continuous problem (11.50). Observing that the nonconvexity of (11.50) is due to the constraints (11.50b), (11.50c), (11.34d), (11.50e), and the objective. For the first two constraints, they can be convexified as

$$\tilde{x}_{i,k} \ge \bar{\psi}_\beta\left(\cancel{\textstyle\sum_{k\in\mathcal{K}}} u_{i,k}; \cancel{\textstyle\sum_{k\in\mathcal{K}}} u_{i,k}^{(n)}\right) \tag{11.51}$$

$$\tilde{s}_i \ge \bar{\psi}_\beta\left(\textstyle\sum_{k\in\mathcal{K}} u_{i,k}; \sum_{k\in\mathcal{K}} u_{i,k}^{(n)}\right), \tag{11.52}$$

respectively, where $\bar{\psi}_\beta(\cdot)$ is the first-order approximation provided in Table 11.1. The remaining parts of the nonconvexity are approximated as done in (11.36), (11.37), and (11.38), respectively. Finally, we arrive at the approximate convex program of problem (11.50), i.e.:

$$\text{minimize}_{\tilde{\boldsymbol{\theta}} \in \tilde{\mathscr{S}}} \quad f(\tilde{\boldsymbol{\theta}}; \tilde{\boldsymbol{\theta}}^{(n)}) \quad \text{subject to} \quad \{(11.36), (11.37), (11.51), (11.52)\}, \tag{11.53}$$

where $\tilde{\boldsymbol{\theta}} \triangleq \{t, \mathbf{w}, \mathbf{r}, \mathbf{v}, \mathbf{g}, \mathbf{u}, \tilde{\mathbf{x}}, \tilde{\mathbf{s}}\}$ includes involved optimization variables and

$$\tilde{\mathscr{S}} \triangleq \{\tilde{\boldsymbol{\theta}} | (11.22\text{c}), (11.22\text{f}), (11.50\text{d}), (11.34\text{c}), (11.50\text{f}), (11.49\text{b})\},$$

defines the convex parts of (11.50). We describe the third suboptimal method in Algorithm 11.5. Similarly to Algorithm 11.4, parameter β is also updated after each iteration. The idea is the same as β is viewed to provide the tightness of the binary approximation $\psi_\beta(\cdot)$.

Convergence of Algorithm 11.5 is guaranteed following the discussion in Section 11.2.2. We remark that if the Capped-ℓ_1 function is adopted in the algorithm, the achieved stationary point is not ensured to hold the first-order optimality of (11.50) since the Capped-ℓ_1 function is not smooth.

11.4.3 Complexity analysis of the presented optimization algorithms

We now estimate the computational complexity of the optimization algorithms presented in this section. In particular, Table 11.2 provides the worst case per-iteration complexity of Algorithms 11.2–11.5. Here, we remark that the complexity of

Algorithm 11.5: Proposed method for solving (11.50)

```
Initialization: Set $n := 0$, choose initial values for $\tilde{\boldsymbol{\theta}}^{(0)}$ and set $\beta^{(0)}$ small
repeat $\{n := n + 1\}$
    Solve (11.53) and achieve $\tilde{\boldsymbol{\theta}}^*$
    Update $\tilde{\boldsymbol{\theta}}^{(n)} := \tilde{\boldsymbol{\theta}}^*$
    Update $\beta^{(n)} := \min\{\beta_{\max}; \beta^{(n-1)} + \varepsilon\}$ for small $\varepsilon$
until Convergence and output $\tilde{\boldsymbol{\theta}}^*$
```

Table 11.2 Estimated worst case per-iteration complexity of the optimization algorithms presented in Section 11.2

	Solved optimization program	**Per-iteration complexity**
Algorithm 11.2	SOCP	$\mathscr{O}(\sqrt{R(K+M)}R^3K^3M^3)$
Algorithm 11.3	MI-SOCP	$\mathscr{O}(2^{RK+R}\sqrt{R(K+M)}R^3K^3M^3)$
Algorithm 11.4	SOCP	$\mathscr{O}(\sqrt{R(K+M)}R^3K^3M^3)$
Algorithm 11.5	SOCP	$\mathscr{O}(\sqrt{R(K+M)}R^3K^3M^3)$

Algorithm 11.2 is dominated by the step of solving the feasibility problem (11.29). For suboptimal methods, the approximated programs solved at each iteration are general convex programs but can be represented as SOCPs as discussed in [10,40]. Recall that an SOCP can be solved by path-following interior-point method in most of the available off-the-shelf convex solvers. Therefore, the complexity of the presented optimization procedures can be estimated following [44, Lecture 6.6]. It is worth noting that Algorithm 11.3 deals with the MI-SOCP (11.39) in each iteration. Indeed, MI-SOCP is a combinatorial optimization problem. In some powerful convex solvers such as MOSEK, the MI-SOCP can be solved by implementing conic branch-and-cut and conic outer approximation frameworks [38]. Thus, its complexity is scaled exponentially with the length of the involved Boolean variables and the size of the solved SOCP.

We can see that Algorithms 11.2, 11.4, and 11.5 have similar computational cost for each iteration. This is because they all deal with beamforming variables. This operation dominates the size of solved problems, i.e., the length of $\mathbf{w}$ is significantly larger than the size of the other optimization variables. However, the use of Algorithm 11.2 is prohibited in practice due to the extremely high number of iterations required as will be demonstrated in the numerical experiments. For Algorithm 11.3, its complexity is higher than that of the other two suboptimal methods, because it needs to handle the Boolean variables. Thus, Algorithm 11.3 is more suitable for problems of small and moderate sizes. On the other hand, Algorithms 11.4 and 11.5 can be applied for solving large-scale problems.

11.5 Numerical examples

We consider a network with $R = 3$ RRHs, each is equipped with $M = 2$ antennas, and $K = 4$ single-antenna users. The RRHs are placed at the coordinates $(-100, 0)$, $(100, 0)$, and $(0, 100\sqrt{3})$ m. Users are randomly distributed in the area covered by all RRHs. We use the path loss model for the small-cell BS, i.e., $30\log_{10}(D_{i,k}) + 38 + \mathcal{N}(0, 8)$ in dB where $D_{i,k}$ is the distance in meters between RRH i and user k. The operating bandwidth is 10 MHz and the noise power density is -143 dBW. We set $\bar{P} = MP^{\text{ant}}$, $\varepsilon_{\max} = 0.55$ [34] and $\delta = 0.5$ (W/Mbits/s). The minimum required data rate for each user is $r_0 = 1$ bit/s/Hz. For the penalty parameters, we take $\alpha_{\max} = 10^3$, $\beta_{\max} = 10^6$ and initialize $\alpha^{(0)} = 10^{-3}$ and $\beta^{(0)} = 0.1$. All the convex programs in this chapter are solved by MOSEK solver in MATLAB® environment [38].

11.5.1 Convergence results

The first set of experiments demonstrates the convergence behavior of the presented optimization algorithms. We first show the convergence of the globally optimal method, i.e., Algorithm 11.2, for one channel realization in Figure 11.3. In particular, the figure plots the upper bound returned when Algorithm 11.2 proceeds for a random channel realization. It is seen that the upper bound monotonically converges to the optimal value. In the figure, we also illustrate the convergence of the BRnB procedure without applying the presented modifications in Algorithm 11.2. As expected, the result clearly shows that the modifications provide remarkable improvement to the convergence speed of the BRnB.

The convergences of the SCA-based suboptimal methods are illustrated in Figure 11.4, where we plot the sequence of energy efficiency objectives obtained by Algorithms 11.3, 11.4, and 11.5 over iterations for two random channel realizations. For comparison purposes, the optimal objective value from the BRnB method for each case of considered channel is also provided. As can be seen, the three suboptimal approaches are able to converge to a very close optimal performance while requiring much smaller number of iterations compared to the globally optimal method. We also note that the SCA procedure in Algorithms 11.4 and 11.5 does not show monotonic convergence behavior as first few iterations. This is because the impact of the updating the parameters in Algorithms 11.4 and 11.5, which is specifically discussed in the

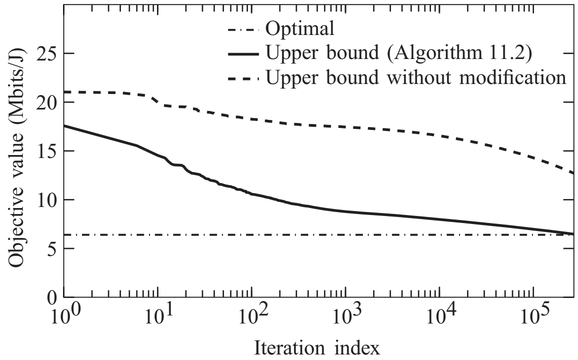


Figure 11.3 Convergence behavior of Algorithm 11.2 for one channel realization with $\bar{P} = 27$ dBm, $\bar{C} = 200$ Mbits/s

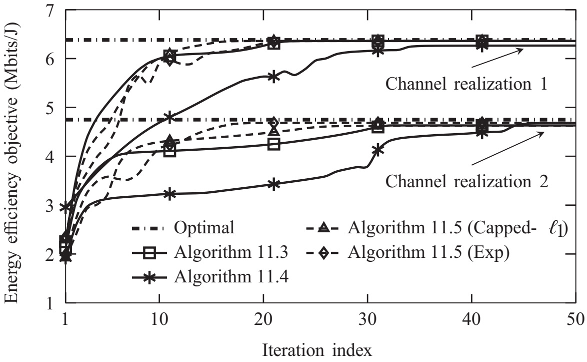


Figure 11.4 Convergence behavior of the SCA-based suboptimal algorithms for two random channel realizations with $\bar{P} = 27$ dBm, $\bar{C} = 200$ Mbits/s

context of the next figure. In fact, when the parameters are fixed after some number of iterations (i.e., $\alpha^{(n)} = \alpha_{\max}$ for Algorithm 11.4 and $\beta^{(n)} = \beta_{\max}$ for Algorithm 11.5), these approaches perform the monotonic convergence.

Figure 11.5 illustrates the convergence of Algorithms 11.4 and 11.5 when the parameters are fixed, i.e., $\alpha^{(n)} = \alpha_{\max}$ and $\beta^{(n)} = \beta_{\max}\ \forall n$, respectively. Note that the result is plotted using Channel realization 2 in Figure 11.4 for the ease of comparison. We can see that the considered methods converge quickly and monotonically (compared to the result in Figure 11.4), but the achieved performances are far from the optimal values. On the other hand, increasing the value of the parameters over the iterations uses more iterations. However, doing so provides a better performance.

To compare the suboptimal methods in terms of time complexity, we provide in Table 11.3 the average per-iteration and total run time of Algorithms 11.3, 11.4, and 11.5. As expected, the SCA-MI requires significantly higher per-iteration run time compared to the other methods and so does the total run time. This is because the SCA-MI in fact still deals with the combinatorial optimization problem and, thus, has high computational complexity, whereas the other two approaches only need to solve the continuous one. We can also see that Algorithms 11.4 and 11.5 have approximately

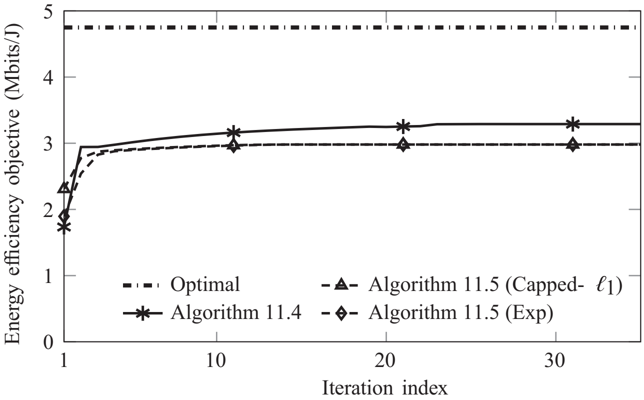


Figure 11.5 Convergence of Algorithms 11.4 and 11.5, when the parameters are fixed, for one random channel realization with $\bar{P} = 27$ dBm, $\bar{C} = 200$ Mbits/s

Table 11.3 Comparison on average per-iteration and total run time (s) of the SCA-based suboptimal methods with $\bar{P} = 27$ dBm, $\bar{C} = 200$ Mbits/s

Run time (s)	Algorithm 11.3	Algorithm 11.4	Algorithm 11.5
Per-iteration	0.9	0.03	0.02
Total	25	2	1.5

similar run time complexity. This result is understandable since they solve the problem with the similar size of involved variables.

11.5.2 Energy efficiency performance

The second set of experiments illustrates the energy efficiency performance of the C-RAN. For comparison purposes, we provide the achieved energy efficiency of the existing design that is the sum rate maximization for C-RAN in [9], dubbed as maxSR. In addition, the optimal energy efficiency of the design without applying RRH-user association and RRH selection (i.e., full cooperation scenario) is also plotted to demonstrate the significance of these selection schemes.[†]

Figure 11.6 shows the average energy efficiency versus the different setting of transmit power $\bar{P}$. Our first observation is that, in the average sense, the performances achieved by the suboptimal algorithms are relatively close to that of the optimal one in all cases of $\bar{P}$. We can also see that the energy efficiency performances achieved by the introduced methods first increase and then, after some point, decrease. The result can be explained as follows. In an energy efficiency maximization problem, when we increase the transmit power in the small value region of $\bar{P}$, the increase in the data rate often outweighs the increase of the power consumption. On the other hand, when $\bar{P}$ is large enough, the energy efficiency optimization mechanism tends not to use all available transmit power. For the linear model of PA's efficiency, the optimized transmit power and data rate would remain unchanged as $\bar{P}$ increases beyond a certain

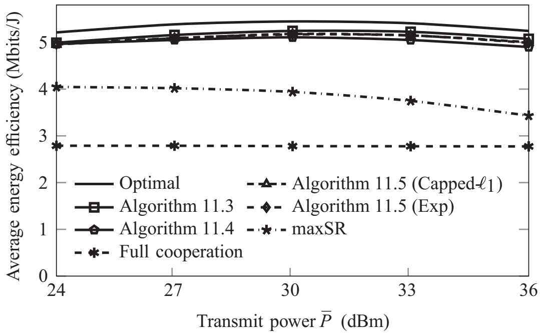


Figure 11.6 Average energy efficiency performances of the considered schemes versus the maximum transmit power with $\bar{C} = 200$ Mbits/s

[†] In the full cooperation scenario, we solve problem (11.24) but with variables **x** and **s** being fixed as $\mathbf{x} = \mathbf{1}$ and $\mathbf{s} = \mathbf{1}$. In this regard, the fronthaul constraints in (11.24d) are replaced by the following one:

$$\sum_{k \in \mathcal{K}} r_k \leq \bar{C}.$$

The performance of this scheme is found by applying the BRnB algorithm to the resulting problem.

value [45,46]. However, for the considered nonlinear model, a larger value of $\bar{P}$ reduces the PA's efficiency due to the nonlinear behavior of the PA as can been seen by (11.18). This leads to the significant increase in the power consumption on PA, and, thus, downgrades the energy efficiency. For the maxSR scheme in [9], the energy efficiency performance monotonically decreases versus $\bar{P}$. This is understandable, since the sum rate performance of the maxSR is constrained by the fronthaul capacity $\bar{C}$. Thus, increasing $\bar{P}$ does not result in the sum rate improvement when $\bar{C}$ is fixed but leads to the increase of the power consumption on the PAs, as discussed earlier. On the other hand, the performance of the full cooperation scheme is inferior to the others. To explain this, we recall that the optimized sum data rate for the full cooperation transmission scheme is less than or equal to $\bar{C}$. As a result, the achieved sum rate is small when $\bar{C}$ is so, and thus the optimized transmit power is small and far from the maximum RF output power of the PAs. This implies the low efficiency of the PAs resulting in high power dissipated on the RF chains. We shall see via the next experiment that energy efficiency of the full cooperation scheme increases significantly when $\bar{C}$ grows sufficiently large.

In Figure 11.7, we plot the average energy efficiency performances versus the maximum fronthaul capacity $\bar{C}$. Similarly to Figure 11.6, we can observe a small gap between the performance of the optimal and suboptimal schemes. Another important observation is that the energy efficiency increases as $\bar{C}$ increases for all considered schemes and becomes saturated when $\bar{C}$ is sufficiently large. The result can be understood as follows. Recall that the capacity-finite fronthaul links restrict the number of users that can be served by an RRH. In the small value region of $\bar{C}$, increasing $\bar{C}$ allows each RRH serving a larger set of users, which improves the cooperation among the RRHs, i.e., more users can be served by multiple RRHs. This increases the system cooperation gain and so does the system performance. When the fronthaul capacity is large enough such that the additional cooperation gain provides no improvement in the achieved performance, increasing $\bar{C}$ does not change the performance. For the full cooperation scheme, the energy efficiency increases proportionally with $\bar{C}$ as expected. However, in all cases of $\bar{C}$, its achieved performance is far from that of the

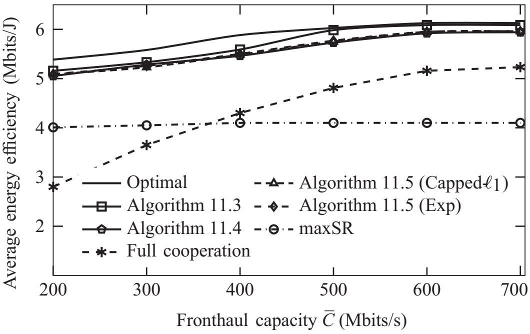


Figure 11.7 Average energy efficiency performances of the considered schemes versus the maximum fronthaul capacity with $\bar{P} = 27$ dBm

introduced methods. This result shows the benefit of the RRH-user association and RRH selection in the fronthaul-constrained C-RAN designs.

11.6 Conclusion

In the C-RANs, the energy-efficient transmission strategies necessarily involve the RRH-user association and RRH selection schemes to deal with the constraints of the limited fronthaul capacity and to boost the energy efficiency. As such, the energy efficiency optimization needs to deal with mixed integer nonconvex programs of which optimal solutions are generally difficult to find. We have introduced a globally optimal method via MO framework to access the optimal energy efficiency performances of the considered C-RAN design. We have also discussed the possible modifications on the generic global optimization framework to improve the optimal algorithm's efficiency, which are based on exploiting the specific properties of the C-RAN. The presented global optimization methods can serve as benchmarks for the energy efficiency optimization approaches in the C-RANs. For practically appealing methods, we have discussed three suboptimal approaches via the SCA procedure. The presented methods have been numerically shown to achieve very close to optimal performance with much reduced complexity.

The introduced optimization algorithms in this chapter provide the design guidelines for not only the energy efficiency maximization, but also other resource allocation problems in the C-RANs. Beyond the C-RAN architectures, many of the presented methods are readily applicable to other wireless communications design problems as well.

Some major challenges in the C-RAN need to be addressed in future research [6,7,47]. One of the main practical issues is the requirement on strict phase-synchronization among cooperative RRHs for coherent transmission. This poses a barrier for practical C-RAN designs, especially in ultradense C-RANs where a very large number of RRHs are co-deployed. A promising solution for this issue is to adopt noncoherent transmission to C-RAN [48]. Another challenge for practical implementation of C-RAN is to meet the requirement of latency. In particular, the limited capacity of the fronthaul links may cause a high latency as 5G RAN is expected to transport a huge amount of data. Therefore, the BB signal rate requirement on the fronthaul should be reduced. This could be done, e.g., by functional splitting such that parts of BB signal-processing functionalities can be done at the RRHs and only the really needed ones at the BBU [6,47].

References

[1] NGMN Alliance. 5G white paper. NGMN Alliance; 2015.

[2] Ericsson. Ericsson mobility report. Ericsson, EAB-18:014144 Uen; 2019.

[3] Mobile C. C-RAN: the road towards green RAN. White Paper. 2011;2.

[4] Rost P, Bernardos CJ, Domenico AD, *et al.* Cloud technologies for flexible 5G radio access networks. IEEE Commun Mag. 2014;52(5):68–76.

[5] Hossain E and Hasan M. 5G cellular: key enabling technologies and research challenges. IEEE Instrum Meas Mag. 2015;18(3):11–21.

[6] 5G PPP Architecture Working Group. View on 5G architecture. White Paper; 2016.

[7] Nokia. 5G immersive service opportunities with Edge Cloud and Cloud RAN. White Paper; 2019.

[8] Ericsson. Virtualized 5G RAN: why, when and how? Ericsson. 2020 [Online]. https://www.ericsson.com/en/blog/2020/2/virtualized-5g-ran-why-when-and-how.

[9] Dai B and Yu W. Sparse beamforming and user-centric clustering for downlink cloud radio access network. IEEE Access. 2014;2(6):1326–1339.

[10] Nguyen KG, Vu QD, Juntti M, and Tran LN. Energy efficiency maximization for C-RANs: discrete monotonic optimization, penalty, and ℓ_0-approximation methods. IEEE Trans Signal Process. 2018;66(17):4435–4449.

[11] Park SH, Simeone O, Sahin O, and Shitz SS. Fronthaul compression for cloud radio access networks: signal processing advances inspired by network information theory. IEEE Signal Process Mag. 2014;31(6):69–79.

[12] Nguyen K, Vu Q, Juntti M, and Tran L. Energy efficient precoding C-RAN downlink with compression at fronthaul. In: 2017 IEEE International Conference on Communications (ICC), Paris, 2017, pp. 1–6,

[13] Peng M, Wang C, Lau V, and Poor HV. Fronthaul-constrained cloud radio access networks: insights and challenges. IEEE Wireless Commun. 2015;22(2): 152–160.

[14] Peng M, Sun Y, Li X, and Wang C. Recent advances in cloud radio access networks: system architectures, key techniques, and open issues. IEEE Commun Surv Tutor. 2016 Third quarter;18(3):2282–2308.

[15] Schaible S and Shi J. Recent developments in fractional programming: single ratio and max-min case. Nonlinear Anal Convex Anal. 2004;493506:493–505.

[16] Zappone A and Jorswieck E. Energy efficiency in wireless networks via fractional programming theory. Found Trends Commun Inf Theory. 2015; 11(3–4):185–396.

[17] Zhang YJ, Qian L, and Huang J. Monotonic optimization in communication and networking systems. Found Trends Networking. 2013;7(1):1–75.

[18] Tuy H, Minoux M, and Hoai-Phuong NT. Discrete monotonic optimization with application to a discrete location problem. SIAM J Optim. 2006;17(1):78–97.

[19] Tuy H, Al-Khayyal F, and Thach PT. Monotonic optimization: branch and cut methods. In: Essays and Surveys in Global Optimization. Springer; 2005. pp. 39–78.

[20] Boyd S and Vandenberghe L. Convex optimization. Cambridge University Press; 2004.

[21] Marks BR and Wright GP. A general inner approximation algorithm for nonconvex mathematical programs. Oper Res. 1978;26(4):681–683.

[22] Beck A, Ben-Tal A, and Tetruashvili L. A sequential parametric convex approximation method with applications to nonconvex truss topology design problem. J Global Optim. 2010;47(1):29–51.

[23] Le Thi HA, Dinh TP, Le HM, and Vo XT. DC approximation approaches for sparse optimization. Eur J Oper Res. 2015;244(1):26–46.

[24] Dai B and Yu W. Energy efficiency of downlink transmission strategies for cloud radio access networks. IEEE J Sel Areas Commun. 2016;34(4): 1037–1050.

[25] Shi Y, Zhang J, and Letaief KB. Group sparse beamforming for green cloud-RAN. IEEE Trans Wireless Commun. 2014;13(5):2809–2823.

[26] Hajisami A, Tran TX, and Pompili D. Elastic-Net: Boosting Energy Efficiency and Resource Utilization in 5G C-RANs. In: 2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS). Orlando, FL, USA; 2017. pp. 466–470.

[27] Vu TT, Ngo DT, Dao MN, Durrani S, Nguyen DH, and Middleton RH. Energy efficiency maximization for downlink cloud radio access networks with data sharing and data compression. IEEE Trans Wireless Commun. 2018;17(8):4955–4970.

[28] Marsch P and Fettweis GP. Coordinated multi-point in mobile communications: from theory to practice. Cambridge University Press; 2011.

[29] Checko A, Christiansen HL, Yan Y, *et al.* Cloud RAN for mobile networks – a technology overview. IEEE Commun Surv Tutor. 2015 First quarter;17(1):405–426.

[30] Pfeiffer T. Next generation mobile fronthaul and midhaul architectures [Invited]. IEEE/OSA J Opt Commun Networking. 2015;7(11):B38–B45.

[31] Isheden C and Fettweis GP. Energy-Efficient Multi-Carrier Link Adaptation with Sum Rate-Dependent Circuit Power. In: 2010 IEEE GLOBECOM 2010; 2010. pp. 1–6.

[32] Auer G, Giannini V, Desset C, *et al.* How much energy is needed to run a wireless network? IEEE Wireless Commun. 2011;18(5):40–49.

[33] Dhaini AR, Ho PH, Shen G, and Shihada B. Energy efficiency in TDMA-based next-generation passive optical access networks. IEEE/ACM Trans Netw. 2014;22(3):850–863.

[34] Persson D, Eriksson T, and Larsson EG. Amplifier-aware multiple-input multiple-output power allocation. IEEE Commun Lett. 2013;17(6): 1112–1115.

[35] Mikami S, Takeuchi T, Kawaguchi H, Ohta C, and Yoshimoto M. An Efficiency Degradation Model of Power Amplifier and the Impact Against Transmission Power Control for Wireless Sensor Networks. In: 2007 IEEE Radio and Wireless Symposium; 2007. pp. 447–450.

[36] Candès EJ, Wakin MB, and Boyd SP. Enhancing sparsity by reweighted ℓ_1 minimization. J Fourier Anal Appl. 2008;14(5):877–905.

[37] Dattorro J. Convex optimization & Euclidean distance geometry. Lulu.com; 2010.

[38] ApS M. The MOSEK optimization toolbox for MATLAB manual. Version 9.0; 2019. Available from: http://docs.mosek.com/9.0/toolbox/index.html.

[39] Alizadeh F and Goldfarb D. Second-order cone programming. Math Program. 2001;95:3–51.

[40] Nguyen KG, Tervo O, Vu QD, Tran LN, and Juntti M. Energy-efficient transmission strategies for CoMP downlink—overview, extension, and numerical comparison. EURASIP J Wireless Commun Netw. 2018;2018(1):207.

[41] Mehanna O, Huang K, Gopalakrishnan B, Konar A, and Sidiropoulos ND. Feasible point pursuit and successive approximation of non-convex QCQPs. IEEE Signal Process Lett. 2015;22(7):804–808.

[42] Vu QD, Tran LN, Farrell R, and Hong EK. An efficiency maximization design for SWIPT. IEEE Signal Process Lett. 2015;22(12):2189–2193.

[43] Le Thi HA and Nguyen MC. DCA based algorithms for feature selection in multi-class support vector machine. Ann Oper Res. 2017;249(1–2):273–300.

[44] Ben-Tal A and Nemirovski A. Lectures on modern convex optimization. Philadelphia: MPS-SIAM Series on Optimization, SIAM; 2001.

[45] Nguyen D, Tran LN, Pirinen P, and Latva-aho M. Precoding for full duplex multiuser MIMO systems: spectral and energy efficiency maximization. IEEE Trans Signal Process. 2013;61(16):4038–4050.

[46] Nguyen KG, Vu QD, Juntti M, and Tran LN. Distributed solutions for energy efficiency fairness in multicell MISO downlink. IEEE Trans Wireless Commun. 2017;16(9):6232–6247.

[47] Westerberg E. 4G/5G RAN architecture: how a split can make the difference. Ericsson Technol Rev. 2016;93(6):1–15.

[48] Vu Q, Tran L, and Juntti M. Distributed Noncoherent Transmit Beamforming for Dense Small Cell Networks. In: ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2019. pp. 4599–4603.