

Power Minimization Based Joint Task Scheduling and Resource Allocation in Downlink C-RAN

Wenchao Xia, Jun Zhang, Tony Q. S. Quek, Shi Jin, and Hongbo Zhu

Abstract

In this paper, we consider the network power minimization problem in a downlink cloud radio access network (C-RAN), taking into account the power consumed at the baseband unit (BBU) for computation and the power consumed at the remote radio heads and fronthaul links for transmission. The power minimization problem for transmission is a fast time-scale issue whereas the power minimization problem for computation is a slow time-scale issue. Therefore, the joint network power minimization problem is a mixed time-scale problem. To tackle the time-scale challenge, we introduce large system analysis to turn the original fast time-scale problem into a slow time-scale one that only depends on the statistical channel information. In addition, we propose a bound improving branch-and-bound algorithm and a combinational algorithm to find the optimal and suboptimal solutions to the power minimization problem for computation, respectively, and propose an iterative coordinate descent algorithm to find the solutions to the power minimization problem for transmission. Finally, a distributed algorithm based on hierarchical decomposition is proposed to solve the joint network power minimization problem. In summary, this work provides a framework to investigate how execution efficiency and computing capability at BBU as well as delay constraint of tasks can affect the network power minimization problem in C-RANs.

W. Xia, J. Zhang, and H. Zhu are with the Jiangsu Key Laboratory of Wireless Communications, Nanjing University of Posts and Telecommunications, Nanjing 210003, P. R. China, E-mail addresses: { 2015010203,zhangjun,hbz}@njupt.edu.cn.

T. Q. S. Quek is with the Information Systems Technology and Design Pillar, Singapore University of Technology and Design, Singapore 487372, E-mail address: tonyquek@sutd.edu.sg.

S. Jin is with the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, P. R. China, E-mail address: jinshi@seu.edu.cn.

Parts of this work were accepted in IEEE Wireless Commun. Network Conf. (WCNC) [1], Barcelona, Spain, Apr. 2018.

Index Terms

Cloud radio access network, large system analysis, energy efficiency, power minimization, computation resource, task scheduling.

I. INTRODUCTION

During the last decade, the evolution of information and communication technology is causing energy consumption levels to reach a distressing rate, due to the dramatic increase in the quantity of subscribers and the number of devices [2]. The massive connectivity also leads to tremendous carbon dioxide emissions into the environment. To reduce energy consumption, many new technologies and network architectures are proposed for 5G green communications [3]. Cloud radio access network (C-RAN) is a new system architecture where computational resource is aggregated into a central baseband unit (BBU) pool to implement the baseband processing of the conventional base stations. The radio functions including amplification, A/D and D/A conversion, and frequency conversion are performed at remote radio heads (RRHs) [4]. In C-RANs, conventional base stations are replaced with low-cost RRHs and these RRHs are deployed close to user equipment terminals (UEs), so the transmission power is significantly reduced. Furthermore, virtualization technique can take full advantage of aggregated computational resources to improve hardware unitization and centralized signal processing can achieve cooperation gain [3, 5].

However, with the aforementioned advantages, new challenges also arise in C-RANs. With the dense deployment of RRHs, C-RANs consume considerable power. Hence turning the idle RRHs into sleep mode and designing energy efficient beamforming matrix are important issues [6, 7]. In addition, the increased traffic causes a heavy burden on fronthaul in terms of capacity demand and power consumption [8, 9]. Finally, the power consumption of baseband processing is also considerable, which is determined by the allocation of computational resource. Overall, all the three challenges have a great effect on the network power consumption in C-RANs.

The network power minimization problem has been extensively studied in [6–11]. Reference [9] jointly optimized downlink beamforming and admission control to minimize the network power. Reference [8] compared two transmission schemes, i.e., the data-sharing scheme and compression scheme. Reference [6] proposed a joint downlink and uplink UE-RRH association and beamforming design to reduce energy consumption. Precoding design and RRH selection

were optimized jointly in [7, 11]. However, the aforementioned papers only considered the first and second challenges, taking into account the power consumption for transmission, i.e., power consumptions of the RRHs and fronthaul links. Dealing with the third challenge in C-RANs is still an open issue. The computational resource aggregated in the BBU pool is provided by many physical servers. Each UE's task is first scheduled on one of these servers and then executed by a virtual machine (VM) created by the server. Therefore, task scheduling and computational resource allocation are the key to the third challenge. There exist some works on computational resource allocation [12–18]. References [12, 14] used a queueing model to represent UEs' data processing and transmitting behavior. Reference [15] modelled the power consumption for computation as an increasing function of UEs' rates. Reference [13] investigated a mobile cloud computing system with computational resource allocation. One thing these works have in common is that they all considered delay constraint. With the popularity of the online video and mobile game, as well as the development of the Internet of things, traffic delay is considered as a key metric to measure the quality-of-service (QoS). However, none of these works take into account task scheduling and computational resource allocation simultaneously. Besides, these works do not consider the time-scale challenge except reference [16], in which the sample averaging was adopted to approximate the time averaging of the power consumption of transmission.

Motivated by these facts, we aim to minimize the network power consumption under delay constraint where the aforementioned three challenges are considered simultaneously in this paper. We consider a downlink C-RAN composed of many RRHs which are connected to a BBU pool via fronthaul. In the BBU pool, there is a data center with a set of physical servers. Each UE has one task which is first scheduled on a certain server and a VM is created by the server to execute this task. Then, the output data is transmitted using RRHs via fronthaul to the UEs. Due to limited fronthaul capacity, the precoded signals are first compressed and then the corresponding compression descriptions are forwarded through the fronthaul. We formulate a joint network power minimization problem of task scheduling and resource allocation, which includes not only computational resource allocation but also power allocation for transmission. Note that the power minimization problem for transmission is a fast time-scale issue because it depends on small-scale fading which varies in the order of milliseconds. However, the power consumption problem for computation is a slow time-scale issue since the task scheduling and computation

resource allocation are usually executed much slower than milliseconds [16]. Therefore, the joint network power minimization problem is a mixed time-scale issue. The main contributions of this work are summarized as follows:

- We first formulate two power minimization problems for computation and transmission, respectively. The power minimization problem for computation is a slow time-scale issue and also a mixed-integer nonlinear programming, where the task scheduling and computation resource allocation are optimized jointly. However, the power minimization problem for transmission is a fast time-scale issue and also a nonconvex problem where power allocation and compression noise are optimized jointly. Then, a joint and mixed time-scale network power minimization problem combining the above two problems is also formulated.
- We translate the fast/mixed time-scale problem into a slow time-scale one. Different from reference [16], where the sample averaging was used to approximate the time averaging of the power consumption of transmission, we introduce the large system analysis to convert our problem into one that only depends on statistical channel information (i.e., large-scale fading) instead of small-scale fading. Therefore, the power minimization problem for transmission, as well as the joint network power minimization problem, is turned into a slow time-scale one.
- For the power minimization problem for computation, we propose a bound improving branch and bound (BnB) algorithm to determine the optimal solutions. To reduce the computational complexity and time, we also propose a suboptimal combinational algorithm. For the power minimization problem for transmission, an iterative coordinate descent algorithm is proposed to determine solutions. Finally, a distributed algorithm based on hierarchical decomposition is proposed to solve the joint network power minimization problem.

The remainder of this paper is organized as follows. Section II introduces the system model and formulates three power minimization problems. Section III proposes two algorithms, i.e., the BnB algorithm and combinational algorithm, to solve the power minimization problem for computation. Section IV proposes an iterative coordinate descent algorithm to solve the power minimization problem for transmission. Based on the analysis in Sections III and IV, a distributed algorithm based on hierarchical decomposition is proposed to solve the joint network power minimization problem in Section V. Numerical results are presented in Section VI. Finally,

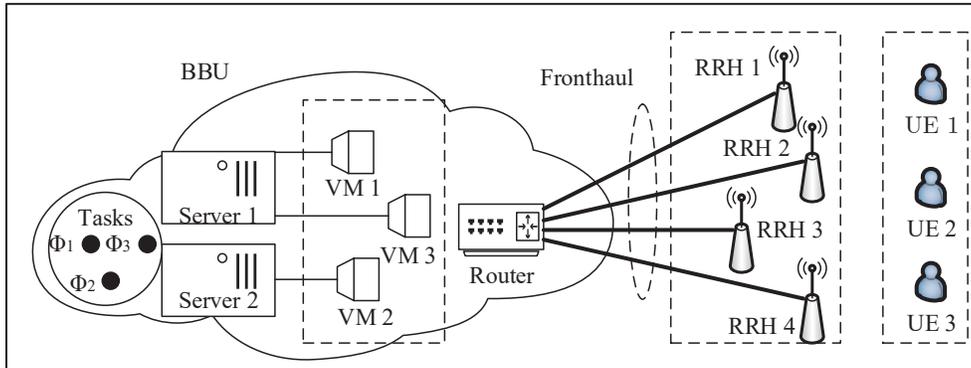


Fig. 1. A typical structure of downlink C-RAN with a data center.

conclusion is drawn in Section VII.

Notations: The notations are given as follows. Matrices and vectors are denoted by bold capital and lowercase symbols. $(\mathbf{A})^T$, $(\mathbf{A})^\dagger$, and $\text{tr}(\mathbf{A})$ stand for transpose, conjugate transpose, and trace of \mathbf{A} , respectively. $\mathbf{A} \succeq \mathbf{0}$ indicates that \mathbf{A} is a Hermitian positive semidefinite matrix. The notations $E(\bullet)$ and $\|\bullet\|_0$ are expectation and l_0 norm operators, respectively. Finally, $\mathbf{a} \sim \mathcal{CN}(\mathbf{0}, \Sigma)$ is a complex Gaussian vector with zero-mean and covariance matrix Σ .

II. CLOUD RADIO ACCESS NETWORK

A. System Model

Consider a downlink C-RAN where L RRHs, each with N antennas, serve K single-antenna UEs, as shown in Fig. 1. The sets of the RRHs and UEs are denoted as $\mathcal{N}_R \triangleq \{1, 2, \dots, L\}$ and $\mathcal{N}_U \triangleq \{1, 2, \dots, K\}$, respectively. In the BBU pool, there is a data center consisting of a set of servers $\mathcal{N}_S \triangleq \{1, 2, \dots, S\}$. The UEs' tasks are first processed at the data center before the output data is transmitted via the RRHs. It is assumed that the RRHs are connected to the BBU pool through high-speed but limited-capacity fronthaul links. In particular, the compress-and-forward scheme is adopted such that the signals for the UEs are first compressed and then the compression descriptions are forwarded to all the RRHs.

In the following, we consider that each UE has one delay-sensitive and computation-intensive task to be executed at the data center. Similar to references [13, 19], the task Φ_k of UE k is

modelled as

$$\Phi_k = \langle D_k, \tau_k, L_k \rangle, \quad (1)$$

where $\langle \cdot, \cdot, \cdot \rangle$ is a triplet, D_k is the amount of output data after accomplishing task Φ_k , τ_k denotes the total time constraint on task execution and data transmission, and L_k represents the load of task Φ_k . Here, we define the load as the execution time when it is executed on a VM with unit computation capability [18].

The tasks are scheduled on different servers for execution at the data center. We use binary variables $x_{s,k} \in \{0, 1\}$ to present the placement plan of tasks, where $x_{s,k} = 1$ indicates task Φ_k is placed on server $s \in \mathcal{N}_S$ and $x_{s,k} = 0$ otherwise. After the task Φ_k is placed on server s with computing capacity λ_s , a VM with computing capability $A_{s,k}$ is created by server s to complete task Φ_k . Due to the diversity of servers, different servers have different executing efficiencies and we define $\varsigma_{s,k}$ as the efficiency of executing task Φ_k on server s . Note that a task can be scheduled on one and only one server during a task execution period so we have the constraint $\sum_{s \in \mathcal{N}_S} x_{s,k} = 1, \forall k \in \mathcal{N}_U$. Then the corresponding execution time of task Φ_k is given as

$$T_k^{(EX)} = \frac{L_k}{\sum_{s \in \mathcal{N}_S} \varsigma_{s,k} x_{s,k} A_{s,k}}, \quad (2)$$

where $A_{s,k}$ should meet the computing capacity constraint of server s as follows:

$$\sum_{k \in \mathcal{N}_U} x_{s,k} A_{s,k} \leq \lambda_s, \forall s \in \mathcal{N}_S. \quad (3)$$

Once one task is finished, its resulting data is encoded and forwarded to the corresponding UE. We first define the channel matrix between all UEs and RRH l as $\mathbf{H}_l = [\mathbf{h}_{l,1}, \dots, \mathbf{h}_{l,K}] \in \mathbb{C}^{N \times K}$ with $\mathbf{h}_{l,k} = \sqrt{d_{l,k}} \tilde{\mathbf{h}}_{l,k}$, where $d_{l,k}$ is the large-scale fading factor caused by path loss and shadow fading between UE k and RRH l , and $\tilde{\mathbf{h}}_{l,k} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_N)$ is the small-scale fading factor. We assume that the UEs are static or moving slowly such that in a task execution period the large-scale fading is invariant.

At the BBU, maximum-ratio transmission is adopted at the signal vector $\mathbf{s} = [s_1, \dots, s_K]^T \in \mathbb{C}^{K \times 1}$, where $s_k \sim \mathcal{CN}(0, 1)$ is the signal for UE k . The perfect channel state information is assumed to be available at the BBU, then the precoded signals for RRH l is given by

$$\hat{\mathbf{x}}_{R_l} = \mathbf{V}_l \mathbf{s}, \quad (4)$$

where $\mathbf{V}_l = \xi_l \mathbf{H}_l \sqrt{\mathbf{P}_l}$ is the precoding matrix, $\mathbf{P}_l \in \mathbb{C}^{K \times K}$ is a diagonal matrix whose elements are adjustable such that power allocation is implemented to improve system performance, and ξ_l is the power scale factor which is given as $\xi_l^2 = \frac{1}{\mathbb{E}[\text{tr}(\mathbf{H}_l \mathbf{H}_l^\dagger)]}$.

Due to the limited capacities of fronthaul links, the precoded signal $\hat{\mathbf{x}}_{R_l}$ are first independently compressed and transmitted to the RRHs via fronthaul links. Here, we adopt point-to-point (P2P) compression for simplicity¹. The quantized signal is expressed as

$$\mathbf{x}_{R_l} = \hat{\mathbf{x}}_{R_l} + \mathbf{q}_l, \quad (5)$$

where $\mathbf{q}_l \sim \mathcal{CN}(0, \mathbf{\Psi}_l)$ is the quantization noise independent of signal $\hat{\mathbf{x}}_{R_l}$ with $\mathbf{\Psi}_l \triangleq \mathbb{E}(\mathbf{q}_l \mathbf{q}_l^\dagger)$. Note that the process of signal compression is independent so that the quantization noise signals \mathbf{q}_l and $\mathbf{q}_{l'}$, are uncorrelated, i.e., $\mathbb{E}(\mathbf{q}_l \mathbf{q}_{l'}^\dagger) = \mathbf{0}, l' \neq l$. According to reference [20], the signal \mathbf{x}_{R_l} can be recovered from $\hat{\mathbf{x}}_{R_l}$ at RRH l if the condition

$$R_{F_l} = \mathbb{E} \left(\log_2 \frac{|\mathbf{V}_l \mathbf{V}_l^\dagger + \mathbf{\Psi}_l|}{|\mathbf{\Psi}_l|} \right) \leq C_l, l \in \mathcal{N}_R, \quad (6)$$

is satisfied, where C_l is the fronthaul capacity for RRH l . Furthermore, the transmission power at RRH l should meet the power constraint given as follows:

$$P_{R_l}^{(TR)} = \mathbb{E} \left[\text{tr} \left(\mathbf{V}_l \mathbf{V}_l^\dagger + \mathbf{\Psi}_l \right) \right] \leq P_{R_l}^{(MAX)}, \forall l \in \mathcal{N}_R, \quad (7)$$

where $P_{R_l}^{(MAX)}$ is the transmission power budget.

The received signal at UE k is given by

$$y_{U_k} = \mathbf{h}_k^\dagger \mathbf{x}_R + z_{U_k}, \quad (8)$$

where $\mathbf{h}_k = [\mathbf{h}_{1,k}^T, \dots, \mathbf{h}_{L,k}^T]^T \in \mathbb{C}^{LN \times 1}$, $\mathbf{x}_R = [\mathbf{x}_{R_1}^T, \dots, \mathbf{x}_{R_L}^T]^T \in \mathbb{C}^{LN \times 1}$, and $z_{U_k} \sim \mathcal{CN}(0, \sigma^2)$ is the independent received noise with zero mean and variance σ^2 . Although the UEs do not know the exact effective channels, we assume that the average effective channels can be learned at the

¹There are two common compress-and-forward schemes, i.e., P2P compression and Wyner-Ziv (WZ) coding. WZ coding can achieve higher performance and make better use of limited fronthaul capacity than P2P compression scheme. However, such benefits come with a cost in terms of computational complexity. Besides, finding an optimal decompression order is a hard problem. In this work, for simplicity, we only consider P2P compression scheme. However, this work can be extended to the case of WZ coding scheme applied at fronthaul with a fixed decompression order.

UEs. Therefore, the achievable rate of UE k using a standard bound based on the worst-case uncorrelated additive noise [21, 22] is computed as

$$R_{U_k} = B \log_2 \left(1 + \frac{\text{Sig}_k}{\text{Int}_k} \right), \quad (9)$$

where B is the system bandwidth, $\text{Sig}_k = |\mathbf{E}(\mathbf{h}_k^\dagger \mathbf{v}_k)|^2$, $\mathbf{v}_k = [\mathbf{v}_{1,k}^T, \dots, \mathbf{v}_{L,k}^T]^T \in \mathbb{C}^{LN \times 1}$, and

$$\text{Int}_k = \text{var}(\mathbf{h}_k^\dagger \mathbf{v}_k) + \sum_{i \in \mathcal{N}_U \setminus \{k\}} \mathbf{E}|\mathbf{h}_k^\dagger \mathbf{v}_i|^2 + \mathbf{E}|\mathbf{h}_k^\dagger \mathbf{\Psi} \mathbf{h}_k| + \sigma^2, \quad (10)$$

where $\text{var}(x) = \mathbf{E}\{[x - \mathbf{E}(x)][x - \mathbf{E}(x)]^\dagger\}$ and $\mathbf{\Psi} = \text{diag}(\{\mathbf{\Psi}_l\}_{l=1}^L)$. Then, the transmission time² of output data of task Φ_k is given as $T_k^{(TR)} = \frac{D_k}{R_{U_k}}$.

B. Power Consumption Model

In the following, we are interested in the network power which includes the powers consumed at the RRHs, the fronthaul links, and the servers.

1) *RRH Power Consumption*: The power consumption at the RRHs consists of both circuit power consumption and transmitting power consumption, and we adopt a linear power consumption model given by [7, 23]

$$P_{R_l} = \begin{cases} \frac{1}{v_l} P_{R_l}^{(TR)} + P_{R_l}^{(Active)}, & \text{if } P_{R_l}^{(TR)} > 0, \\ P_{R_l}^{(Sleep)}, & \text{if } P_{R_l}^{(TR)} = 0, \end{cases} \quad (11)$$

where v_l is the efficiency of the power amplifier and $P_{R_l}^{(Active)}$ denotes the circuit power consumption to support RRH l to transmit signals. If there is no transmission at RRH l , it can be turned into sleep mode with lower power consumption $P_{R_l}^{(Sleep)}$. Generally, $P_{R_l}^{(Sleep)} < P_{R_l}^{(Active)}$, thus turning a RRH into sleep mode can save power. We define $P_{\Delta R_l} = P_{R_l}^{(Active)} - P_{R_l}^{(Sleep)}$ and P_{R_l} can be rewritten as

$$P_{R_l} = \frac{1}{v_l} P_{R_l}^{(TR)} + P_{R_l}^{(Sleep)} + \mathbb{I}_{\{P_{R_l}^{(TR)} > 0\}} P_{\Delta R_l}. \quad (12)$$

²In this paper, we assume that the transmission time at fronthaul links is constant and negligible so that it can be ignored.

2) *Fronthaul Power Consumption*: The power consumption model of fronthaul links depends on specific fronthaul technologies. Similar to reference [8], we use a general model to compute the power consumption of each fronthaul channel as

$$P_{F_l} = \eta_l R_{F_l}, \quad (13)$$

where $\eta_l = \frac{P_{F_l}^{(MAX)}}{C_l}$ and $P_{F_l}^{(MAX)}$ is the power consumed by the fronthaul link for RRH l when working at full capacity. This model has been used for microwave backhaul links in [24] and also can be generalized to other backhaul technologies, such as passive optical network, fiber-based Ethernet, etc., as mentioned in [25].

3) *Server Power Consumption*: The total power consumption of a server s is given by [26]

$$P_{S_s} = P_{S_s}^{(Static)} + \sum_{k \in \mathcal{N}_U} P_{VM_{s,k}}, \quad (14)$$

where $P_{S_s}^{(Static)}$ is constant no matter whether VMs are running or not and $P_{VM_{s,k}}$ is the power consumed by a VM. It is observed that the total power consumption of a VM is directly related to the system component utilization [18, 26, 27]. More utilization of the system components leads to more power consumption [26]. In the linear weighted model, the total power consumption $P_{VM_{s,k}}$ of VM k created by server s can be further decomposed into four components related to CPU, disk, IO devices, and memory as follows [18]:

$$P_{VM_{s,k}} = P_{VM_{s,k}}^{(CPU)} + P_{VM_{s,k}}^{(DISK)} + P_{VM_{s,k}}^{(IO)} + P_{VM_{s,k}}^{(MEMORY)}. \quad (15)$$

Because there exists a direct relation between the execution time of tasks on VMs and CPU utilization, we use the CPU power consumption to approximate the VM power consumption with a weight $\chi_{s,k}$ [18, 27], which can be expressed as

$$P_{VM_{s,k}} = x_{s,k} \chi_{s,k} A_{s,k}. \quad (16)$$

C. Problem Formulation

Since we are interested to minimize the network power consumption while meeting the delay constraint, we first formulate two power minimization problems for computation and transmission, respectively. Then, a joint network power minimization problem is also established.

1) *Power Minimization Problem for Computation:* It is assumed that the time limitation for finishing task Φ_k on a certain VM is $\tau_k^{(EX)}$ ($\tau_k^{(EX)} < \tau_k$). Based on the above analysis, the power minimization problem for computation where task scheduling and computational resource allocation are executed jointly is formulated as

$$\mathcal{P}_0 : \min_{\mathbf{x}, \mathbf{A}} \sum_{s \in \mathcal{N}_S} P_{S_s} \quad (17a)$$

$$\text{s.t. } T_k^{(EX)} \leq \tau_k^{(EX)}, \forall k \in \mathcal{N}_U, \quad (17b)$$

$$\sum_{k \in \mathcal{N}_U} x_{s,k} A_{s,k} \leq \lambda_s, \forall s \in \mathcal{N}_S, \quad (17c)$$

$$\sum_{s \in \mathcal{N}_S} x_{s,k} = 1, \forall k \in \mathcal{N}_U, \quad (17d)$$

$$A_{s,k} \geq 0, \forall k \in \mathcal{N}_U, \forall s \in \mathcal{N}_S, \quad (17e)$$

$$x_{s,k} \in \{0, 1\}, \forall k \in \mathcal{N}_U, \forall s \in \mathcal{N}_S. \quad (17f)$$

where \mathbf{x} is a collect of $x_{s,k}$'s, indicating the placement plan of tasks and \mathbf{A} is a collect of $A_{s,k}$'s denoting the resource allocation plan of the servers.

2) *Power Minimization Problem for Transmission:* Similarly, we first assume that the time constraint for transmitting the output signals of the tasks is $\tau_k^{(TR)}$ ($\tau_k^{(TR)} < \tau_k$). Then, we formulate the power minimization problem for transmission as

$$\mathcal{P}_1 : \min_{\mathbf{P}, \mathbf{\Psi}} \sum_{l \in \mathcal{N}_R} P_{R_l} + P_{F_l} \quad (18a)$$

$$\text{s.t. } T_k^{(TR)} \leq \tau_k^{(TR)}, \forall l \in \mathcal{N}_R, \quad (18b)$$

$$R_{F_l} \leq C_l, \forall l \in \mathcal{N}_R, \quad (18c)$$

$$P_{R_l}^{(TR)} \leq P_{R_l}^{(MAX)}, \forall l \in \mathcal{N}_R, \quad (18d)$$

where \mathbf{P} is a collect of $p_{l,k}$'s and $\mathbf{\Psi}$ is a collect of Ψ_l 's. Note that \mathbf{P} describes the power allocation scheme and $\mathbf{\Psi}$ indicates the quantization levels of the all RRHs.

3) *Joint Network Power Minimization Problem*: Finally, the joint network power minimization problem for computation and transmission is formulated as

$$\mathcal{P}_2 : \min_{\mathbf{x}, \mathbf{A}, \mathbf{P}, \Psi} \sum_{s \in \mathcal{N}_S} P_{S_s} + \omega \sum_{l \in \mathcal{N}_R} (P_{R_l} + P_{F_l}) \quad (19a)$$

$$\text{s.t. } T_k^{(EX)} + T_k^{(TR)} \leq \tau_k, \forall k \in \mathcal{N}_U, \quad (19b)$$

$$(17c) - (17f), (18c), \text{ and } (18d), \quad (19c)$$

where ω is a factor to balance the power consumption of computation and transmission.

We observe that \mathcal{P}_0 is a slow time-scale problem but the joint optimization of power allocation and quantization noise in \mathcal{P}_1 is a fast time-scale problem since it depends on small-scale fading. Consequently, \mathcal{P}_2 is a mixed time-scale issue that needs further attention [16]. To solve this challenge caused by the time-scale issue, authors in [16] used ensemble averaging over fast time-scale samples so that the final problem became a slow time-scale problem. Instead, we introduce large system analysis to transform \mathcal{P}_1 and \mathcal{P}_2 into slow time-scale problems depending only on large-scale fading [28, 29]. Furthermore, we assume that the UEs are static or moving slowly such that the large-scale fading remains invariant within a task execution period.

III. POWER MINIMIZATION PROBLEM FOR COMPUTATION

For \mathcal{P}_0 to be solvable, it is assumed that task Φ_k can be further divided into S sub-task $\phi_{s,k}$'s, each with load $l_{s,k}$, and placed on S servers, respectively [18, 30]. This assumption can be interpreted as a relaxation of the binary variable $x_{s,k}$ to a real variable, i.e., $x_{s,k} \in [0, 1]$, then the variable $x_{s,k}$ is absorbed in the new defined variable $l_{s,k} = x_{s,k} L_k$. The total load of sub-tasks should satisfy the constraint $\sum_{s \in \mathcal{N}_S} l_{s,k} = L_k$. Then, a VM with computation capability $a_{s,k}$ is created by server s for sub-task $\phi_{s,k}$ and the associated execution time is $t_{s,k}^{(EX)} = \frac{l_{s,k}}{c_{s,k} a_{s,k}}$, where $a_{s,k}$ satisfies the constraint $\sum_{k \in \mathcal{N}_U} a_{s,k} \leq \lambda_s$. Accordingly, the power consumption of sub-task

$\phi_{s,k}$ is given as $p_{VM_{s,k}} = \chi_{s,k} a_{s,k}$ and the relaxed version of \mathcal{P}_0 can be written as

$$\mathcal{P}_{0-1} : \min_{\mathbf{a}, \mathbf{l}} \sum_{s \in \mathcal{N}_S} \sum_{k \in \mathcal{N}_U} p_{VM_{s,k}} \quad (20a)$$

$$l_{s,k} - \varsigma_{s,k} a_{s,k} \tau_k^{(EX)} \leq 0, k \in \mathcal{N}_U, s \in \mathcal{N}_S, \quad (20b)$$

$$\sum_{k \in \mathcal{N}_U} a_{s,k} \leq \lambda_s, s \in \mathcal{N}_S, \quad (20c)$$

$$\sum_{s \in \mathcal{N}_S} l_{s,k} = L_k, k \in \mathcal{N}_U, \quad (20d)$$

$$a_{s,k} \geq 0, l_{s,k} \geq 0, k \in \mathcal{N}_U, s \in \mathcal{N}_S, \quad (20e)$$

where \mathbf{a} is a collect of $a_{s,k}$'s and \mathbf{l} is a collect of $l_{s,k}$'s. Note that $P_{S_s}^{(Static)}$ is constant and thus omitted. The objective function and constraints (20b)-(20e) are linear so \mathcal{P}_{0-1} can be solved easily. However, the solution determined from \mathcal{P}_{0-1} is generally not the optimal solution to \mathcal{P}_0 . In what follows, we introduce the BnB algorithm to find the optimal solution to \mathcal{P}_0 based on the solution to \mathcal{P}_{0-1} .

A. Branch and Bound Algorithm

We define a set $\mathcal{S} = \{(s, k) | \forall s \in \mathcal{N}_S, \forall k \in \mathcal{N}_U\}$ that contains all the task-server pairs and introduce another two task-server pair sets $\mathcal{S}_0 = \{(s, k) | x_{s,k} = 0, \forall s \in \mathcal{N}_S, \forall k \in \mathcal{N}_U\}$ and $\mathcal{S}_1 = \{(s, k) | x_{s,k} = 1, \forall s \in \mathcal{N}_S, \forall k \in \mathcal{N}_U\}$. With the defined sets, we formulate an equivalent problem of \mathcal{P}_0 as follows:

$$\mathcal{P}_{0-2} : \min_{\mathbf{x}, \mathbf{A}} \sum_{s \in \mathcal{N}_S} \sum_{k \in \mathcal{N}_U} P_{VM_{s,k}} \quad (21a)$$

$$\text{s.t. (17b) - (17e),} \quad (21b)$$

$$x_{s,k} = 1, \forall (s, k) \in \mathcal{S}_1, \quad (21c)$$

$$x_{s,k} = 0, \forall (s, k) \in \mathcal{S}_0, \quad (21d)$$

$$x_{s,k} \in \{0, 1\}, (s, k) \in \mathcal{S} \setminus (\mathcal{S}_0 \cup \mathcal{S}_1). \quad (21e)$$

Similarly, an equivalent problem of \mathcal{P}_{0-1} is formulated as

$$\mathcal{P}_{0-3} : \min_{\mathbf{a}, \mathbf{l}} \sum_{s \in \mathcal{N}_S} \sum_{k \in \mathcal{N}_U} p_{VM_{s,k}} \quad (22a)$$

$$\text{s.t. (20b) – (20e),} \quad (22b)$$

$$l_{s,k} = L_k, \forall (s, k) \in \mathcal{S}_1, \quad (22c)$$

$$l_{s,k} = 0, \forall (s, k) \in \mathcal{S}_0, \quad (22d)$$

$$0 \leq l_{s,k} \leq L_k, (s, k) \in \mathcal{S} \setminus (\mathcal{S}_0 \cup \mathcal{S}_1). \quad (22e)$$

For notational convenience, we use the related parameter tuples $(z, \mathcal{S}_0, \mathcal{S}_1)$ and $(z, \mathcal{S}_0, \mathcal{S}_1)'$ to denote \mathcal{P}_{0-2} and \mathcal{P}_{0-3} , respectively, where z is the optimal value of the objective function in \mathcal{P}_{0-3} . The BnB algorithm for \mathcal{P}_0 is provided in **Algorithm 1**. At the beginning, we define F as the set of branch problems and z^* as the best-known objective value. The main process of the BnB algorithm consists of two important steps as follows:

1) **Branching:** In each iteration process, we choose the problem that achieves the minimum lower bound, denoted as $(\hat{z}, \hat{\mathcal{S}}_0, \hat{\mathcal{S}}_1)$, to branch. Then, the task-server pair with the highest priority (s^*, k^*) is chosen to be divided into two smaller branch problems: one is with $x_{s^*, k^*} = 0$ and the other is with $x_{s^*, k^*} = 1$. Accordingly, the relaxed problems of the two branches are given as: one is with $l_{s^*, k^*} = 0$ and the other is with $l_{s^*, k^*} = L_{k^*}$. Evidently, the priority function plays an important role in reducing the complexity and we define the priority function as $f_p(s, k) = \frac{\chi_{s,k} L_k}{\varsigma_{s,k}}$.

2) **Bounding and Pruning:** According to the selected branch, we compute the lower bounds of sub-problems $(z^{(B1,n)}, \mathcal{S}_0^{(B1,n)}, \mathcal{S}_1^{(B1,n)})'$ and $(z^{(B2,n)}, \mathcal{S}_0^{(B2,n)}, \mathcal{S}_1^{(B2,n)})'$, respectively. The two branch problems are stored in F for further branching when their lower bounds are less than the current best-known value z^* . If a new feasible solution is found which is lower than the current best-known value z^* , the current best-known solution is updated. Besides, the stored branches in F having an lower bound larger than the value of the new best-known feasible solution can be deleted.

B. Suboptimal Task Scheduling Algorithm

Although the BnB algorithm can find the global optimal solution, the convergence rate can be slow, especially for large number task-server pairs. Therefore, we introduce a suboptimal but

Algorithm 1 BnB algorithm for task scheduling.

- 1: **Initialization:** $z^* = +\infty$, $\mathcal{S}_0^{(0)} = \mathcal{S}_1^{(0)} = \emptyset$, and $F = \{(z^{(0)}, \mathcal{S}_0^{(0)}, \mathcal{S}_1^{(0)})\}$, and $n = 0$.
 - 2: **while** $F \neq \emptyset$ **do**
 - 3: Find the problem $(\hat{z}, \hat{\mathcal{S}}_0, \hat{\mathcal{S}}_1)$ according to $\hat{z} = \min_{(z, \mathcal{S}_0, \mathcal{S}_1) \in F} z$ from F and update $F = F \setminus \{(\hat{z}, \hat{\mathcal{S}}_0, \hat{\mathcal{S}}_1)\}$.
 - 4: Select the task-server pair with the highest priority, i.e., $(s^*, k^*) = \arg \max_{(s, k) \in \mathcal{S} \setminus (\hat{\mathcal{S}}_0 \cup \hat{\mathcal{S}}_1)} f_p(s, k)$, and set $n = n + 1$.
 - 5: Update $\mathcal{S}_0^{(B_1, n)} = \hat{\mathcal{S}}_0 \cup \{(s^*, k^*)\}$, $\mathcal{S}_1^{(B_1, n)} = \hat{\mathcal{S}}_1$, $\mathcal{S}_0^{(B_2, n)} = \hat{\mathcal{S}}_0$, and $\mathcal{S}_1^{(B_2, n)} = \hat{\mathcal{S}}_1 \cup \{(s^*, k^*)\}$;
 - 6: Solve problems $(z^{(B_i, n)}, \mathcal{S}_0^{(B_i, n)}, \mathcal{S}_1^{(B_i, n)})'$, $i = 1, 2$. If there is no feasible solution, set $z^{(B_i, n)} = +\infty$.
 - 7: **if** $z^{(B_i, n)} < z^*$, $i = 1, 2$, **then**
 - 8: **if** $\mathcal{S} == \mathcal{S}_0^{(B_i, n)} \cup \mathcal{S}_1^{(B_i, n)}$, **then**
 - 9: Set $z^* = z^{(B_i, n)}$, $\mathcal{S}_0^* = \mathcal{S}_0^{(B_i, n)}$, and $\mathcal{S}_1^* = \mathcal{S}_1^{(B_i, n)}$.
 - 10: **else**
 - 11: Update $F = F \cup \{(z^{(B_i, n)}, \mathcal{S}_0^{(B_i, n)}, \mathcal{S}_1^{(B_i, n)})\}$.
 - 12: **end if**
 - 13: **end if**
 - 14: Check and prune existing branches. If branch problem $(z^{(j)}, \mathcal{S}_0^{(j)}, \mathcal{S}_1^{(j)})$ in F meets the constraint $z^{(j)} > z^*$, $j = 1, 2, \dots, |F|$, then it can be pruned, i.e., $F = F \setminus \{(z^{(j)}, \mathcal{S}_0^{(j)}, \mathcal{S}_1^{(j)})\}$.
 - 15: **end while**
 - 16: Return z^* , \mathcal{S}_0^* , and \mathcal{S}_1^* .
-

fast task scheduling algorithm which is referred to as heuristic task scheduling algorithm, as shown in **Algorithm 2**.

In **Algorithm 2**, the unscheduled task Φ_{k^*} with the highest load is first considered and server s^* which has the highest execution efficiency for this task has a priority. When the available resource in server s^* is sufficient to support task Φ_{k^*} , then server s^* allocates as little computing resource as possible to task Φ_{k^*} , i.e., $A_{s^*, k^*} = \frac{L_{k^*}}{\tau_{k^*} \varsigma_{s^*, k^*}}$. Otherwise, task Φ_{k^*} continues to search the potential server. Note that different from the BnB algorithm, the heuristic task scheduling algorithm cannot always find solutions to \mathcal{P}_0 . However, in the case with high execution efficiency or abundant

computation resource, **Algorithm 2** can achieve satisfying performance with lower computational complexity and time. Therefore, we propose a combinational algorithm where **Algorithm 2** is first adopted to find the suboptimal solutions. If no solution is found via **Algorithm 2**, we continue to resort to **Algorithm 1**. We refer to such an algorithm as combinational task scheduling algorithm, as shown in **Algorithm 3**.

Algorithm 2 Heuristic task scheduling algorithm.

- 1: **Initialize** $\mathcal{N}_{S'} = \mathcal{N}_S$. Find task $\Phi_{k^*} = \arg \max_{k \in \mathcal{N}_U} L_k$ and update $\mathcal{N}_U \triangleq \mathcal{N}_U \setminus \{k^*\}$.
 - 2: **if** $\mathcal{N}_{S'}$ is not empty, **then**
 - 3: Find server $s^* = \arg \min_{s \in \mathcal{N}_{S'}} \frac{\chi_{s,k}}{\varsigma_{s,k^*}}$ for task Φ_{k^*} and update $\mathcal{N}_{S'} = \mathcal{N}_{S'} \setminus \{s^*\}$.
 - 4: **if** $\frac{L_{k^*}}{\tau_{k^*} \varsigma_{s^*,k^*}} \leq \lambda_{s^*}$, **then**
 - 5: Update $A_{s^*,k^*} = \frac{L_{k^*}}{\tau_{k^*} \varsigma_{s^*,k^*}}$ and $\lambda_{s^*} = \lambda_{s^*} - A_{s^*,k^*}$. Then, go to step 1.
 - 6: **else**
 - 7: Go to step 2.
 - 8: **end if**
 - 9: **else**
 - 10: Heuristic task scheduling fail.
 - 11: **end if**
-

Algorithm 3 Combinational task scheduling algorithm.

- 1: **Algorithm 2** is adopted.
 - 2: **if** no available solution is found via **Algorithm 2**, **then**
 - 3: **Algorithm 1** is adopted to find the optimal solution.
 - 4: **else**
 - 5: Return the solution found by **Algorithm 2**.
 - 6: **end if**
-

IV. POWER MINIMIZATION PROBLEM FOR TRANSMISSION

In this section, we first introduce approximate results with large system analysis and then find the solution to \mathcal{P}_1 based on these approximations. According to large system analysis, we can take care of the small-scale fading using the following lemma.

Lemma 1. *Given that $\tilde{\mathbf{h}}_{l,k}$'s are i.i.d. complex Gaussian variables with independent real and imaginary parts. According to the law of large numbers and the large-dimensional random matrix theory, as $N \rightarrow \infty$, then we have the following results:*

1) $P_{R_l}^{(TR)} - \bar{P}_{R_l}^{(TR)} \rightarrow 0$, where

$$\bar{P}_{R_l}^{(TR)} = \bar{\xi}_l^2 N \sum_{k \in \mathcal{N}_U} p_{l,k} d_{l,k} + \text{tr}(\Psi_l), \quad (23)$$

with $\bar{\xi}_l^2 = \frac{1}{N \sum_{k \in \mathcal{N}_U} d_{l,k}}$.

2) $R_{U_k} - \bar{R}_{U_k} \rightarrow 0$, where

$$\bar{R}_{U_k} = \log_2 \left[1 + \frac{\overline{\text{Sig}}_k}{\overline{\text{Int}}_k} \right], \quad (24)$$

where $\overline{\text{Sig}}_k = (\bar{\mathbf{d}}_k^T \sqrt{\mathbf{p}_k})^2$, $\overline{\text{Int}}_k = \frac{1}{N} \sum_{i \in \mathcal{N}_U \setminus \{k\}} (\bar{\mathbf{d}}_i \circ \bar{\mathbf{d}}_k)^T \mathbf{p}_i + \frac{1}{N^2} \sum_{l \in \mathcal{N}_R} d_{l,k} \text{tr}(\Psi_l) + \frac{1}{N^2} \sigma^2$, $\bar{\mathbf{d}}_k = [\xi_1 d_{1,k}, \dots, \xi_L d_{L,k}]^T$, $\mathbf{p}_k = [p_{1,k}, \dots, p_{L,k}]^T$, and $\bar{\mathbf{d}}_i \circ \bar{\mathbf{d}}_k$ is the Hadamard product whose l -th element is $\xi_l^2 d_{l,i} d_{l,k}$.

3) $R_{F_l} - \bar{R}_{F_l} \rightarrow 0$, where

$$\bar{R}_{F_l} = \frac{1}{\log 2} (\Delta_l - \log |\Psi_l|), \quad (25)$$

where $\Delta_l = \log |\Lambda_l| + \sum_{k \in \mathcal{N}_U} \left(\frac{1}{1+e_{l,k}} - \log \frac{1}{1+e_{l,k}} \right) - K$, $e_{l,k} = \bar{\xi}_l^2 p_{l,k} d_{l,k} \text{tr} \Lambda_l^{-1}$, and

$$\Lambda_l = \sum_{k \in \mathcal{N}_U} \frac{\bar{\xi}_l^2 p_{l,k} d_{l,k}}{1 + e_{l,k}} \mathbf{I}_N + \Psi_l. \quad (26)$$

Proof: See the Appendix.

The approximate results in (23), (24), and (25) are obtained with the assumption that $N \rightarrow \infty$. Note that these results can achieve satisfying accuracy even when N is not too large.

Based on the approximate results, we formulate an alternative to \mathcal{P}_1 as:

$$\mathcal{P}_{1-1} : \min_{\mathbf{P}, \Psi} \sum_{l \in \mathcal{N}_R} \bar{P}_{R_l} + \bar{P}_{F_l} \quad (27a)$$

$$\text{s.t. } (2^{\tau_k \frac{D_k}{(TR)_B}} - 1) \overline{\text{Int}}_k \leq \overline{\text{Sig}}_k, \forall k \in \mathcal{N}_U, \quad (27b)$$

$$\bar{R}_{F_l} \leq C_l, \forall l \in \mathcal{N}_R, \quad (27c)$$

$$\bar{P}_{R_l}^{(TR)} \leq P_{R_l}^{(MAX)}, \forall l \in \mathcal{N}_R, \quad (27d)$$

where $\bar{P}_{F_l} = \eta_l \bar{R}_{F_l}$ and $\bar{P}_{R_l} = \left(\frac{1}{v_l} + \rho_l P_{\Delta R_l} \right) \bar{P}_{R_l}^{(TR)} + P_{R_l}^{(Sleep)}$. According to reference [31], the l_0 -norm can be approximated with convex relaxation l_1 -norm as $\|\bar{P}_{R_l}^{(TR)}\|_0 \approx \rho_l \bar{P}_{R_l}^{(TR)}$, where

$\rho_l = \frac{c_1}{\bar{P}_{R_l}^{(TR)} + c_2}$ is iteratively updated, c_1 is a constant, and c_2 is a small constant to guarantee numerical satiability. However, \mathcal{P}_{1-1} is still non-convex with respect to $p_{l,k}$ and Ψ_l because of \bar{R}_{F_l} and $\bar{\text{Sig}}_k$. To achieve a stationary point of \mathcal{P}_{1-1} , we first introduce the following lemma.

Lemma 2 ([32, 33]). *For any two $N \times N$ positive definite Hermitian matrices Λ and Γ , then*

$$\log |\Lambda| \leq -\log |\Gamma| + \text{tr}(\Gamma\Lambda) - N, \quad (28)$$

with the equality if and only if $\Gamma = \Lambda^{-1}$. When $N = 1$, the inequality (28) is simplified as

$$\log(\varphi) \leq -\log(\gamma) + \varphi\gamma - 1, \quad (29)$$

with the equality if and only if $\gamma = \varphi^{-1}$.

Applying (28) to the denominator of \bar{R}_{F_l} , then we have

$$\tilde{\bar{R}}_{F_l} = -\log_2 |\Gamma_l| + \text{tr}(\Gamma_l \Lambda_l) - N - \log_2 |\Psi_l| + \sum_{k \in \mathcal{N}_U} \left(\frac{1}{1 + e_{l,k}} - \log \frac{1}{1 + e_{l,k}} \right) - K, \quad (30)$$

which is equivalent to \bar{R}_{F_l} when $\Gamma_l = \Lambda_l^{-1}$.

Next, we change optimization variable $p_{l,k}$ to $\mathbf{W} = \mathbf{w}\mathbf{w}^T \in \mathbb{C}^{KL \times KL}$ where $\mathbf{w} = [\mathbf{w}_1^T, \dots, \mathbf{w}_K^T]^T$ and $\mathbf{w}_k = \sqrt{\mathbf{p}_k}$. Then, $\bar{P}_{R_l}^{(TR)}$ can be rewritten as

$$\bar{P}_{R_l}^{(TR)} = N \text{tr}(\mathbf{A}_l \mathbf{T} \mathbf{W}) + \text{tr}(\mathbf{B}_l \Psi),$$

where $\mathbf{A}_l = \text{diag}([\mathbf{a}_l^T, \dots, \mathbf{a}_l^T]^T) \in \mathbb{C}^{KL \times KL}$, $\mathbf{a}_l \in \mathbb{C}^{L \times 1}$ represents a vector whose l -th element is 1 and 0 elsewhere, $\mathbf{T} = \text{diag}(\{\mathbf{t}_k\}_{k=1}^K) \in \mathbb{C}^{KL \times KL}$ is a diagonal matrix, $\mathbf{t}_k \in \mathbb{C}^{L \times 1}$ denotes a vector whose l -th element is $\bar{\xi}_l^2 d_{l,k}$, and $\mathbf{B}_l \in \mathbb{C}^{NL \times NL}$ is a diagonal matrix whose main diagonal elements from $((l-1)N+1)$ -th to (lN) -th are 1's and 0 elsewhere. Λ_l can be rewritten as

$$\Lambda_l = \text{tr}(\mathbf{A}_l \mathbf{E} \mathbf{A}_l \mathbf{W}) \mathbf{I}_N + \mathbf{J}_l \mathbf{B}_l \Psi \mathbf{J}_l^H,$$

where $\mathbf{E} = \text{diag}(\{\mathbf{e}_k\}_{k=1}^K) \in \mathbb{C}^{KL \times KL}$ is a diagonal matrix, \mathbf{e}_k is a vector whose l -th diagonal element is $\frac{\bar{\xi}_l^2 d_{l,k}}{1 + e_{l,k}}$, and $\mathbf{J} = [\mathbf{0}_1, \dots, \mathbf{0}_{l-1}, \mathbf{I}_N, \mathbf{0}_{l+1}, \dots, \mathbf{0}_L] \in \mathbb{C}^{N \times NL}$ with $\mathbf{0}_l \in \mathbb{C}^{N \times N}$ being a zero-matrix. Similarly, based on the new defined variable \mathbf{W} , $e_{l,k}$, $\bar{\text{Sig}}_k$, and $\bar{\text{Int}}_k$ can be rewritten as $e_{l,k} = \text{tr}(\mathbf{T} \mathbf{G}_{l,k} \mathbf{W}) \text{tr} \Lambda_l^{-1}$, $\bar{\text{Sig}}_k = \text{tr}(\mathbf{F}_k \bar{\mathbf{D}} \mathbf{F}_k \mathbf{W})$, and

$$\bar{\text{Int}}_k = \frac{1}{N} \sum_{i \in \mathcal{N}_U \setminus \{k\}} \text{tr}(\tilde{\mathbf{D}}_{ik} \mathbf{F}_i \mathbf{W}) + \frac{1}{N^2} \text{tr}(\mathbf{D}_k \Psi) + \frac{1}{N^2} \sigma^2,$$

respectively, where \mathbf{G}_{lk} is a diagonal matrix whose $(l + (k - 1)L)$ -th main diagonal element is 1 and 0 elsewhere, $\bar{\mathbf{D}} = \bar{\mathbf{d}}\bar{\mathbf{d}}^T \in \mathbb{C}^{KL \times KL}$ with $\bar{\mathbf{d}} = [\bar{\mathbf{d}}_1^T, \dots, \bar{\mathbf{d}}_K^T]^T$, $\mathbf{F}_k \in \mathbb{C}^{KL \times KL}$ is a matrix whose main diagonal elements from $((k - 1)L + 1)$ -th to (kL) -th are 1's and elsewhere 0, $\tilde{\mathbf{D}}_{ik} = \text{diag}([\mathbf{z}_1^T, \dots, \mathbf{z}_{i-1}^T, (\bar{\mathbf{d}}_i \circ \bar{\mathbf{d}}_k)^T, \mathbf{z}_{i+1}^T, \dots, \mathbf{z}_K^T]^T)$, and $\mathbf{D}_k = \text{diag}(\{\mathbf{d}_{lk}\}_{l=1}^L)$ with $\mathbf{d}_{lk} = [d_{l,k}, \dots, d_{l,k}]^T \in \mathbb{C}^{N \times 1}$.

As a result, \mathcal{P}_{1-1} can be reformulated as a semidefinite programming as follows:

$$\mathcal{P}_{1-2} : \min_{\mathbf{W}, \Psi, \Gamma} \sum_{l \in \mathcal{N}_R} \bar{P}_{R_l} + \eta_l \tilde{\tilde{R}}_{F_l} \quad (31a)$$

$$\text{s.t. } \tilde{\tilde{R}}_{F_l} \leq C_l, \forall l \in \mathcal{N}_R, \quad (31b)$$

$$\mathbf{W} \succeq 0, \quad (31c)$$

$$\text{rank}(\mathbf{W}) = 1, \quad (31d)$$

$$(27b) \text{ and } (27d), \quad (31e)$$

where Γ is a set of Γ_l 's. The optimal value of Γ_l in \mathcal{P}_{1-2} , according to **Lemma 2**, is $\Gamma_l^* = \Lambda_l^{-1}, \forall l \in \mathcal{N}_R$. Relaxing the rank constraint $\text{rank}(\mathbf{W}) = 1$, \mathcal{P}_{1-2} is still non-convex over three variables \mathbf{W} , Ψ , and Γ . But it is convex with respect to any one of these variables and can converge to a stationary point by an iterative coordinate descent algorithm as shown in **Algorithm 4**.

In **Algorithm 4**, at t iteration, $\mathbf{W}^{(t)}$ and $\Psi^{(t)}$ are optimized simultaneously, whereas $\Gamma^{(t)}$ is updated directly as $\Gamma_l^{(t)} = (\Lambda_l^{(t)})^{-1}, \forall l \in \mathcal{N}_R$, according to (28). Such process is repeated until convergence. Note that **Algorithm 4** does not take the rank-one constraint into consideration. After the semidefinite relaxation (SDR) of \mathcal{P}_{1-2} is solved, the optimal solution \mathbf{W}^* should be converted into a feasible solution to \mathcal{P}_1 . Since the rank of \mathbf{W}^* may not equal to one, we can extract the feasible solution to \mathcal{P}_1 from \mathbf{W}^* with Gaussian randomization method [34]. **Algorithm 4** generates a non-increasing sequence of objective values, thus the convergence is guaranteed [32]. The main computational complexity of **Algorithm 4** lies in step 2, where the SDR of \mathcal{P}_{1-2} is solved. The computational complexity of the SDR of \mathcal{P}_{1-2} is $\mathcal{O}(D_{SDP}^{3.5} \log(1/\epsilon))$ with a custom-built interior-point algorithm [35], where $\epsilon > 0$ is the solution accuracy and $D_{SDP} = KL + NL$ is the dimension. Assuming that **Algorithm 4** converges in T_1 iterations, the total complexity of **Algorithm 4** is $\mathcal{O}(T_1 D_{SDP}^{3.5} \log(1/\epsilon_2))$ [8].

Algorithm 4 Iterative coordinate descent algorithm.

- 1: **Initialization:** $\mathbf{W}^{(0)}, \Psi_l^{(0)}, \Gamma_l^{(0)} = (\Lambda_l^{(0)})^{-1}$ and $t = 0$.
 - 2: Update $t = t + 1$ and find the optimal $\Psi_l^{(t)}$ and $\mathbf{W}^{(t)}$ with given $\Gamma_l^{(t-1)}$ via solving problem \mathcal{P}_{1-2} .
 - 3: Update $\Gamma_l^{(t)} = (\Lambda_l^{(t)})^{-1}$.
 - 4: Repeat steps 2 and 3 until convergence.
 - 5: Return $\Psi_l^{(t)}$ and $\mathbf{W}^{(t)}$ as the optimal solution Ψ_l^* and \mathbf{W}^* , respectively.
-

V. JOINT NETWORK POWER MINIMIZATION PROBLEM FOR COMPUTATION AND TRANSMISSION

In this section, we find the solution to the joint network power minimization problem \mathcal{P}_2 .

A. Problem Reformulation

We find that \mathcal{P}_2 has to confront with all the difficulties in \mathcal{P}_0 and \mathcal{P}_1 because \mathcal{P}_2 is combination of two problems coupled by the delay constraint. To avoid the nonconvexity, we first reformulate \mathcal{P}_2 as

$$\mathcal{P}_{2-1} : \min_{\mathbf{x}, \mathbf{A}, \mathbf{P}, \Psi} \sum_{s \in \mathcal{N}_S} \sum_{k \in \mathcal{N}_U} \chi_{s,k} A_{s,k} + \omega \sum_{l \in \mathcal{N}_R} (P_{R_l} + P_{F_l}) \quad (32a)$$

$$\text{s.t. } \frac{L_k}{\sum_{s \in \mathcal{N}_S} \varsigma_{s,k} A_{s,k}} + T_k^{(TR)} \leq \tau_k, \forall k \in \mathcal{N}_U, \quad (32b)$$

$$\sum_{k \in \mathcal{N}_U} A_{s,k} \leq \lambda_s, \forall s \in \mathcal{N}_S, \quad (32c)$$

$$A_{s,k} \leq x_{s,k} \lambda_s, \forall s \in \mathcal{N}_S, \forall k \in \mathcal{N}_U, \quad (32d)$$

$$(17c) - (17f), (18c), \text{ and } (18d), \quad (32e)$$

where (32d) indicates that server s does not allocate any resource to task Φ_k , if task Φ_k is not assigned on server s , i.e., $x_{s,k} = 0 \implies A_{s,k} = 0$. As mentioned above, \mathcal{P}_{2-1} is a mixed time-scale problem. Similar to \mathcal{P}_1 , we turn \mathcal{P}_{2-1} into a slow time-scale problem based on the asymptotic

results in Section IV and formulate the SDR of \mathcal{P}_{2-1} as follows:

$$\mathcal{P}_{2-2} : \min_{\mathbf{x}, \mathbf{A}, \mathbf{W}, \Psi, \Gamma, \varphi} \sum_{s \in \mathcal{N}_S} \sum_{k \in \mathcal{N}_U} \chi_{s,k} A_{s,k} + \omega \sum_{l \in \mathcal{N}_R} \bar{P}_{R_l} + \eta_l \tilde{R}_{F_l} \quad (33a)$$

$$\text{s.t.} \quad \frac{L_k}{\sum_{s \in \mathcal{N}_S} \varsigma_{s,k} A_{s,k}} + \frac{D_k}{\tilde{R}_{U_k}} \leq \tau_k, \forall l \in \mathcal{N}_R, \quad (33b)$$

$$x_{s,k} \in [0, 1], \quad (33c)$$

$$(17d), (17e), (27d), (31b), (31c), (32c), \text{ and } (32d), \quad (33d)$$

where φ is a set of φ_k 's and

$$\tilde{R}_{U_k} = \log_2(\overline{\text{Sig}}_k + \overline{\text{Int}}_k) + \log_2(\varphi_k) - \varphi_k \overline{\text{Int}}_k + 1. \quad (34)$$

In \mathcal{P}_{2-2} , $x_{s,k}$'s are relaxed as continuous variables within $[0, 1]$ and (28) and (29) are applied to \bar{R}_{F_l} and \bar{R}_{U_k} , respectively. Then, \mathcal{P}_{2-2} is convex with respect to either $\{\mathbf{x}, \mathbf{A}, \mathbf{W}, \Psi\}$ or $\{\Gamma, \varphi\}$. Thus, we find the solution to \mathcal{P}_{2-2} by alternately solving the following two problems:

$$\mathcal{P}_{2-3} : \min_{\mathbf{x}, \mathbf{A}, \mathbf{W}, \Psi} \sum_{s \in \mathcal{N}_S} \sum_{k \in \mathcal{N}_U} \chi_{s,k} A_{s,k} + \omega \sum_{l \in \mathcal{N}_R} \bar{P}_{R_l} + \eta_l \tilde{R}_{F_l} \quad (35a)$$

$$\text{s.t.} \quad (17d), (17e), (27d), (31b), (31c), (32c), (32d), (33b), \text{ and } (33c), \quad (35b)$$

and

$$\mathcal{P}_{2-4} : \min_{\Gamma, \varphi} \sum_{l \in \mathcal{N}_R} \bar{P}_{R_l} + \eta_l \tilde{R}_{F_l} \quad (36a)$$

$$\text{s.t.} \quad (27d), (31b), \text{ and } (33b), \quad (36b)$$

where the optimal solution to \mathcal{P}_{2-4} is given as

$$\Gamma_l^* = \Lambda_l^{-1} \text{ and } \varphi_k^* = \text{Int}_k^{-1}. \quad (37)$$

By applying the dual decomposition to \mathcal{P}_{2-3} [36–38], the Lagrangian function associated with problem \mathcal{P}_{2-3} is given by

$$\begin{aligned} & L(\mathbf{x}, \mathbf{A}, \mathbf{W}, \Psi, \boldsymbol{\mu}) \\ &= \sum_{s \in \mathcal{N}_S} \sum_{k \in \mathcal{N}_U} \chi_{s,k} A_{s,k} + \omega \sum_{l \in \mathcal{N}_R} (\bar{P}_{R_l} + \eta_l \tilde{R}_{F_l}) + \sum_{k \in \mathcal{N}_U} \mu_k \left(\frac{L_k}{\sum_{s \in \mathcal{N}_S} \varsigma_{s,k} A_{s,k}} + \frac{D_k}{\tilde{R}_{U_k}} - \tau_k \right), \end{aligned}$$

where $\boldsymbol{\mu} = [\mu_1, \dots, \mu_K]^T \in \mathbb{C}^{K \times 1}$ is composed of the Lagrangian multipliers. The corresponding Lagrangian dual function is given by

$$g(\boldsymbol{\mu}) = g_1(\boldsymbol{\mu}) + g_2(\boldsymbol{\mu}) - \sum_{k \in \mathcal{N}_U} \mu_k \tau_k, \quad (38)$$

where

$$\begin{cases} g_1(\boldsymbol{\mu}) = \inf_{\mathbf{x}, \mathbf{A}} \sum_{s \in \mathcal{N}_S} \sum_{k \in \mathcal{N}_U} \chi_{s,k} A_{s,k} + \sum_{k \in \mathcal{N}_U} \mu_k \frac{L_k}{\sum_{s \in \mathcal{N}_S} \chi_{s,k} A_{s,k}}, \\ \text{s.t. (17d), (17e), (32c), (32d), and (33c),} \end{cases} \quad (39)$$

and

$$\begin{cases} g_2(\boldsymbol{\mu}) = \inf_{\boldsymbol{\Psi}, \mathbf{W}} \sum_{l \in \mathcal{N}_R} (\omega \bar{P}_{R_l} + \omega \eta_l \tilde{R}_{F_l}) + \sum_{k \in \mathcal{N}_U} \mu_k \frac{D_k}{\bar{R}_{U_k}}, \\ \text{s.t. (27d), (31b), and (31c).} \end{cases} \quad (40)$$

Then, the master dual problem associated with \mathcal{P}_{2-3} is formulated as

$$\mathcal{P}_{2-5}: \max_{\boldsymbol{\mu}} g(\boldsymbol{\mu}). \quad (41)$$

Since \mathcal{P}_{2-3} is convex and satisfies the Slater's condition, the duality gap of \mathcal{P}_{2-3} and its dual problem \mathcal{P}_{2-5} is zero [39]. In the following, we propose a distributed algorithm based on hierarchical decomposition to find the optimal solution to \mathcal{P}_{2-2} .

B. Distributed Algorithm Based on Hierarchical Decomposition

In **Algorithm 5**, the upper level primal decomposition is conducted, which introduces \mathcal{P}_{2-3} and \mathcal{P}_{2-4} . Based on \mathcal{P}_{2-3} , the lower level dual decomposition is conducted to formulate the dual problem \mathcal{P}_{2-5} . Therefore, the distributed algorithm should involve two level iterations: the outer iteration is for \mathcal{P}_{2-3} and \mathcal{P}_{2-4} to converge and the inner iteration is for \mathcal{P}_{2-5} to converge.

In the outer iteration, the optimal solution to \mathcal{P}_{2-4} is directly given as $\Gamma_l^{(t)} = (\Lambda_l^{(t)})^{-1}$ and $\varphi_k^{(t)} = (\text{Int}_k^{(t)})^{-1}$. However, to obtain the optimal solution to \mathcal{P}_{2-3} , it relies on the dual problem \mathcal{P}_{2-5} , whose optimal solution can be achieved via the inner iteration where $(\mathbf{x}^{(q)}, \mathbf{A}^{(q)})$ and $(\mathbf{W}^{(q)}, \boldsymbol{\Psi}^{(q)})$ are alternatingly updated. Specifically, at p -th inner iteration:

- 1) **Data center's algorithm** jointly optimizes task scheduling and computation resource allocation. $\mathbf{x}^{(p)}$ and $\mathbf{A}^{(p)}$ are updated by solving subproblem $g_1(\boldsymbol{\mu})$ with a BnB algorithm similar to **Algorithm 1** or a combinational algorithm similar to **Algorithm 3**.

- 2) **BBU pool's algorithm** jointly optimizes power allocation and compression noise. $\mathbf{W}^{(p)}$ and $\Psi^{(p)}$ are updated by solving subproblem $g_2(\boldsymbol{\mu})$ with an iterative coordinate descent algorithm similar to **Algorithm 4**.
- 3) On the other hand, the price factor is adjusted by **UEs' algorithm**. Since $g(\boldsymbol{\mu})$ is not differentiable over μ_k , a sub-gradient approach is adopted here to update the price factor μ_k at UE k , i.e.,

$$\mu_k^{(p+1)} = \left[\mu_k^{(p)} + \delta_\mu^{(p)} \left(\frac{L_k}{\sum_{s \in \mathcal{N}_S} \varsigma_{s,k} A_{s,k}^{(p)}} + \frac{D_k}{\bar{R}_{U_k}} - \tau_k \right) \right]^+, \forall k \in \mathcal{N}_U, \quad (42)$$

where $\delta_\mu^{(p)}$ is dynamically chosen stepsize sequence [36,40]. Similarly, after the SDR problem of \mathcal{P}_{2-2} is solved, we need to extract $p_{l,k}$'s from \mathbf{W}^* with Gaussian randomization method [34]. Note that there are three sub-algorithms named data center's algorithm, BBU pool's algorithm, and UEs' algorithm in **Algorithm 5** and they are executed in parallel in the data center, the BBU pool, and the UEs, respectively. Therefore, the complexity is significantly reduced compared to the direct optimization of \mathcal{P}_{2-2} .

VI. NUMERICAL RESULTS

In this section, we present the numerical results to show the performance of our proposed algorithms, where L RRHs and K UEs are distributed uniformly and independently in an area with a radius of 100 m. The outer interference combined with background noise is set as -150 dBm/Hz and the path loss function is given as $128.1 + 37.6 \log_{10}(d)$ where d in km. The system bandwidth is $B = 20$ MHz and the number of RRH antenna is $N = 5$. For simplicity, we assume that each RRH has the same parameters and is subject to the same constraints, i.e., $P_{R_l}^{(MAX)} = 1$ W, $C_l = 2$ bps/Hz, $\eta_l = 0.5$, $\nu_l = 0.25$, $p_{R_l}^{(Active)} = 6.8$ W, and $p_{R_l}^{(Sleep)} = 4.3$ W, $\forall l \in \mathcal{N}_R$. The task load L_k is assumed to be uniformly distributed in $[0.01, 0.1]$ and the output data is $D_k = 1.6$ Mbits, $\forall k \in \mathcal{N}_U$. The parameters of the servers are set as $P_{S_s}^{(Static)} = 2$ W, $\chi_{s,k} = 1, \forall s \in \mathcal{N}_S, \forall k \in \mathcal{N}_U$, and the computing capacity λ_s is uniformly distributed in $[\lambda_{lb}, \lambda_{ub}]$. Moreover, we assume the execution efficiency $\varsigma_{s,k}$ is distributed uniformly in $[\varsigma_{lb}, \varsigma_{ub}]$, $c_1 = 1$, $c_2 = 10^{-5}$, and $\omega = 1$.

Algorithm 5 Distributed algorithm based on hierarchical decomposition.

- 1: **Initialization:** $\underline{\mathbf{W}}^{(0)}, \underline{\Psi}^{(0)}, t = 0$.
 - 2: **while** Convergence of outer iteration (t) not achieved **do**
 - 3: $t = t + 1$ and $p = 0$.
 - 4: **while** Convergence of inner iteration (p) not achieved **do**
 - 5: $p = p + 1$;
 - 6: **Data center's algorithm:** Update $\mathbf{x}^{(p)}$ and $\mathbf{A}^{(p)}$ by solving subproblem $g_1(\boldsymbol{\mu})$.
 - 7: **BBU pool's algorithm:** Update $\mathbf{W}^{(p)}$ and $\Psi^{(p)}$ by solving subproblem $g_2(\boldsymbol{\mu})$.
 - 8: **UEs' algorithm:** Update $\boldsymbol{\mu}^{(p)}$ according to (42).
 - 9: **end while**
 - 10: Update $\{\underline{\mathbf{x}}^{(t)}, \underline{\mathbf{A}}^{(t)}, \underline{\mathbf{W}}^{(t)}, \underline{\Psi}^{(t)}\}$ as $\{\mathbf{x}^{(p)}, \mathbf{A}^{(p)}, \mathbf{W}^{(p)}, \Psi^{(p)}\}$ at the convergence of the inner iteration.
 - 11: Update $\underline{\Lambda}_l^{(t)} = \Lambda_l^{-1}(\underline{\mathbf{W}}^{(t)}, \underline{\Psi}^{(t)})$, $\forall l \in \mathcal{N}_R$, and $\underline{\varphi}_k^{(t)} = \text{Int}_k^{-1}(\underline{\mathbf{W}}^{(t)}, \underline{\Psi}^{(t)})$, $\forall k \in \mathcal{N}_U$.
 - 12: **end while**
 - 13: **Return** $\{\underline{\mathbf{x}}^{(t)}, \underline{\mathbf{A}}^{(t)}, \underline{\mathbf{W}}^{(t)}, \underline{\Psi}^{(t)}\}$ as the optimal solution at the convergence of the outer iteration.
-

A. Power Consumption for Computation

We first consider that each task has the same execution delay constraint, i.e., $\tau_k^{(EX)} = \tau^{(EX)}, \forall k \in \mathcal{N}_U$. Fig. 2 shows the sum of power consumed by all the VMs, i.e., $P_{VM} = \sum_{s \in \mathcal{N}_S} \sum_{k \in \mathcal{N}_U} P_{VM_{s,k}}$, versus the execution delay constraint $\tau^{(EX)}$ with $\{K = 6, S = 4, \lambda_{lb} = 1, \lambda_{ub} = 2\}$. It is observed that with the increase of $\tau^{(EX)}$, the consumed power decreases accordingly because the VMs have more time to finish the tasks with a lower power. To study the influence of the execution efficiency $\varsigma_{s,k}$ on the power consumption of the VMs, we set two different regimes $[0.1, 0.5]$ and $[0.6, 1]$ representing low and high execution efficiency cases, respectively. It is observed that higher execution efficiency leads to less power consumption.

Next, we compare the BnB algorithm (i.e., **Algorithm 1**) with the combinational algorithm (i.e., **Algorithm 3**) versus the execution efficiency $\varsigma_{s,k}$ in Fig. 3 with $\{K = 6, S = 4, \lambda_{lb} = 0.1, \lambda_{ub} = 1\}$. Since $\varsigma_{s,k}$'s are random variables distributed uniformly in $[\varsigma_{lb}, \varsigma_{ub}]$, we divide the value range into many segments $[\varsigma_{lb}, \varsigma_{lb} + 0.1]$ with a fixed length 0.1 for fairness and use the

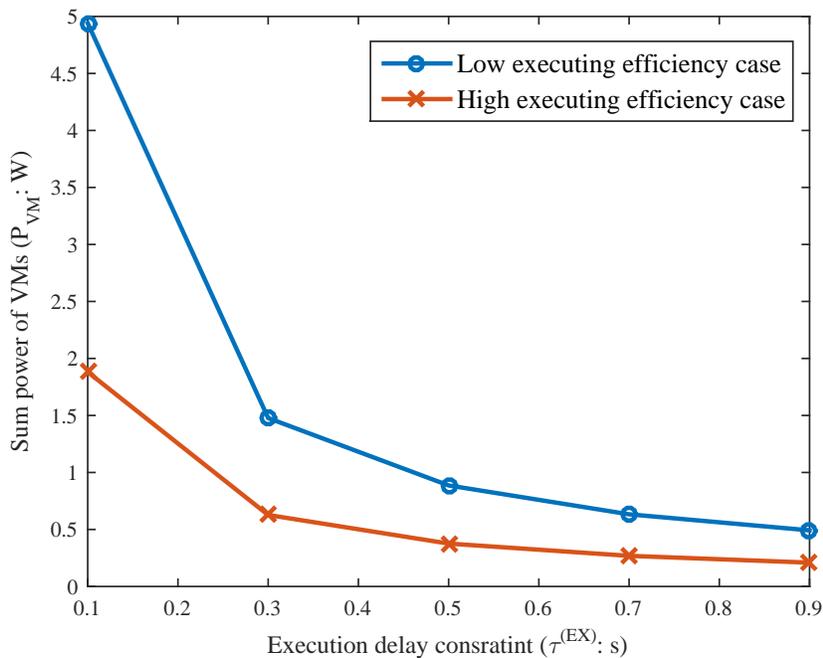
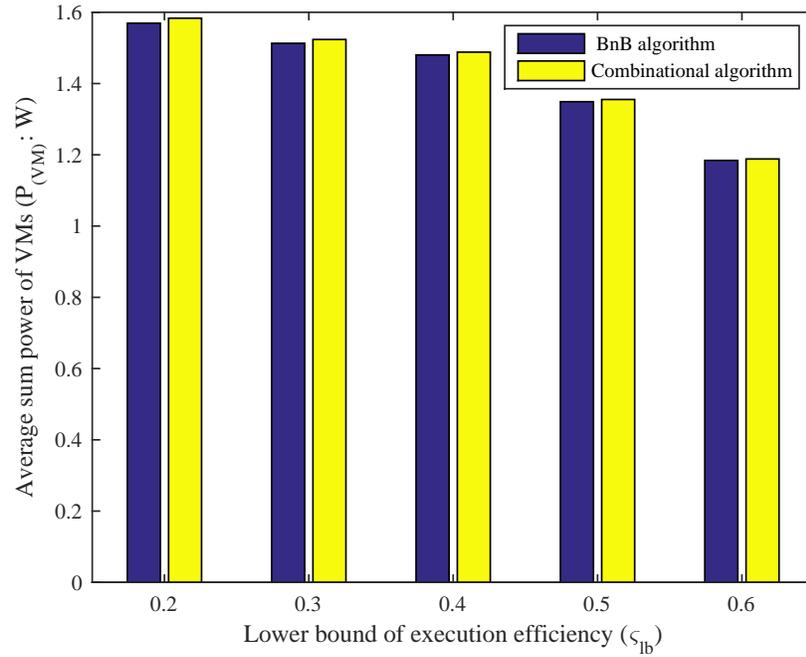


Fig. 2. Sum of power consumed by VMs versus execution delay constraint.

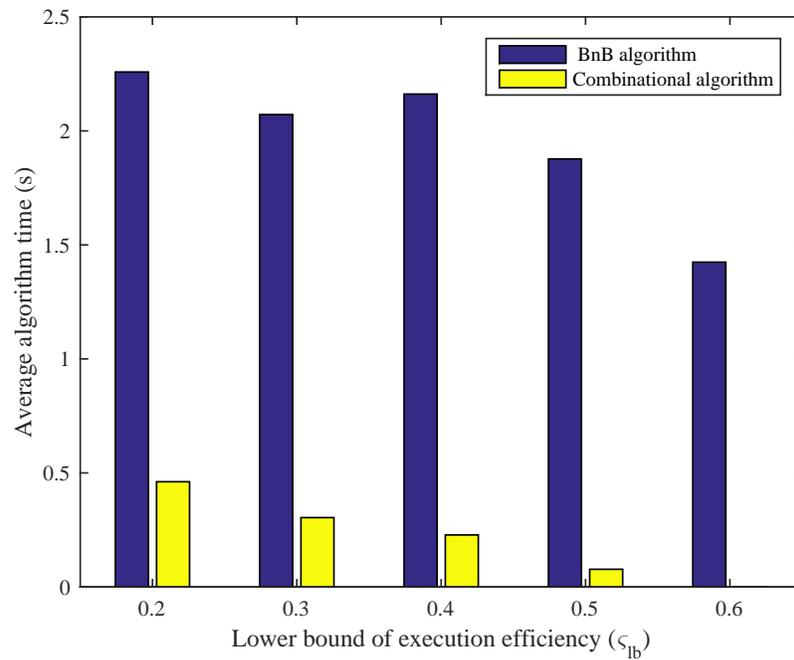
lower bound of the execution efficiency ς_{lb} as the x-axis. Fig. 3 is the average result of 200 independent realizations. It is found that the power consumption for computation decreases as the lower bound of the execution efficiency ς_{lb} increases because the demand for computation resource is reduced. We also observe that the solutions obtained by the combinational task scheduling algorithm are suboptimal and require a little more power consumption but with much less runtime, compared to the BnB algorithm. Therefore, in order to save time and reduce computation complexity, it is suggested to adopt the combinational task scheduling algorithm with a little performance loss. However, the optimal solution can be found via the BnB algorithm at the cost of computational time and complexity. Besides, Fig. 3(b) suggests that as the lower bound of the execution efficiency ς_{lb} increases, indicating that the overall execution efficiency of servers is improved, then the fraction of times where **Algorithms 2** fails decreases.

B. Power Consumption for Transmission

In the following, we first validate the accuracy of the approximate results derived in Section IV. We define $\epsilon_1 = \sum_{k \in \mathcal{N}_U} \frac{|\bar{R}_{U_k} - R_{U_k}|}{R_{U_k}}$, $\epsilon_2 = \sum_{l \in \mathcal{N}_R} \frac{|\bar{R}_{F_l} - R_{F_l}|}{R_{F_l}}$, and $\epsilon_3 = \sum_{l \in \mathcal{N}_R} \frac{|\bar{P}_{R_l} - P_{R_l}|}{P_{R_l}}$ as the



(a)



(b)

Fig. 3. Comparison of BnB algorithm and combinational algorithm: (a) average power consumption of the VMs and (b) average algorithm time.

inaccuracy levels of the sum rate of UEs $\bar{R}_U = \sum_{k \in \mathcal{N}_U} \bar{R}_{U_k}$, the sum rate of fronthaul links $\bar{R}_F = \sum_{l \in \mathcal{N}_R} \bar{R}_{F_l}$, and the sum power of RRHs $\bar{P}_R = \sum_{l \in \mathcal{N}_R} \bar{P}_{R_l}$, respectively. Fig. 4 shows that these approximate results are not only close to their original expressions but become more accurate as the number of RRH antennas N increases.

For notational simplicity, we define $P^{(TR)} = \sum_{l \in \mathcal{N}_R} (P_{R_l} + P_{F_l})$ as the total power consumption for transmission. It is also assumed that each UE has the same transmission delay, i.e., $\tau_k^{(TR)} = \tau^{(TR)}, \forall k \in \mathcal{N}_U$. To compare compression-based transmission scheme with the data-sharing transmission scheme, Fig. 5 plots $P^{(TR)}$ versus the transmission delay constraint $\tau^{(TR)}$. It can be observed that the power consumption of both transmission schemes decrease as the transmission delay constraint increases. In addition, Fig. 5 indicates that with strict transmission delay constraint, i.e., small values of $\tau^{(TR)}$, compression-based transmission scheme produces less power consumption. However, when the transmission delay constraint is loose, i.e., large values of $\tau^{(TR)}$, the data-sharing transmission scheme achieves a better performance. This is because the fronthaul rate of compression scheme relies on the signal-to-quantization-noise ratio whereas that of data-sharing scheme depends on the UEs' rates and the serving RRH numbers, since the data-sharing scheme delivers each UE's message to all the RRHs that serve this UE via fronthaul links. A smaller value of $\tau^{(TR)}$ suggests a higher data-rate demand, then more RRHs are required to serve the UEs. Therefore, a faster increase of the fronthaul rate occurs in the data-sharing scheme. However, a gradual increase of the fronthaul rate in the compression scheme as the data-rate demand rises.

C. Joint Network Power Minimization Problem for Computation and Transmission

Finally, we present the network power minimization with respect to the delay constraint τ_k under different transmission schemes (i.e., compression based scheme and data-sharing based scheme) and different executing efficiency cases (i.e., low executing efficiency case with $\{\varsigma_{lb} = 0.1, \varsigma_{ub} = 0.5\}$ and high executing efficiency case with $\{\varsigma_{lb} = 0.6, \varsigma_{ub} = 1\}$) in Fig. 6. For simplicity, we assume $\tau_k = \tau, \forall k \in \mathcal{N}_U$. The network power consumption decreases with the increase of the delay constraint because when the delay constraint increases, the QoS level decreases and less power is required to meet the QoS. Similarly, when the average executing efficiency is improved, less computational resource is required thus the network power consumption is also reduced. It is also observed that the network adopts the transmission scheme based

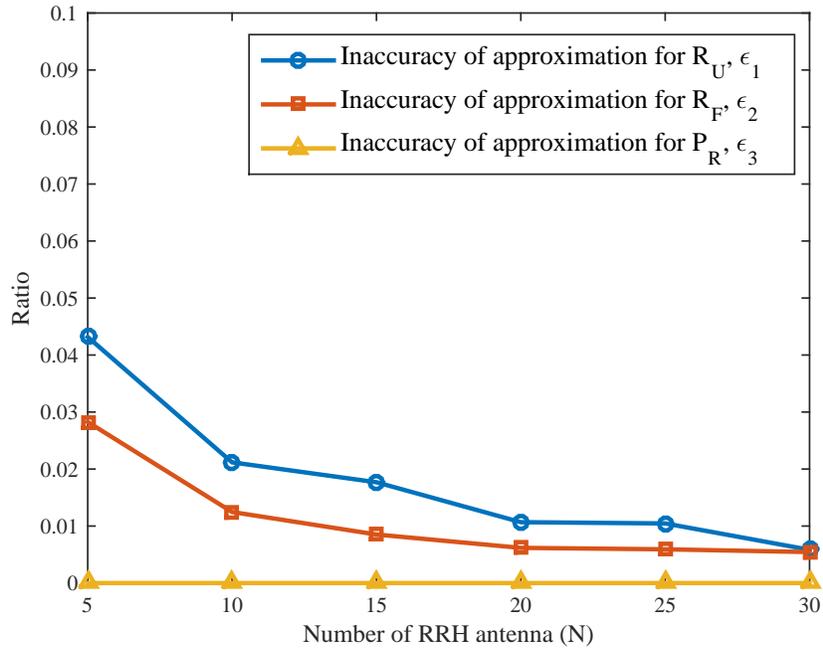


Fig. 4. Accuracy of approximative results.

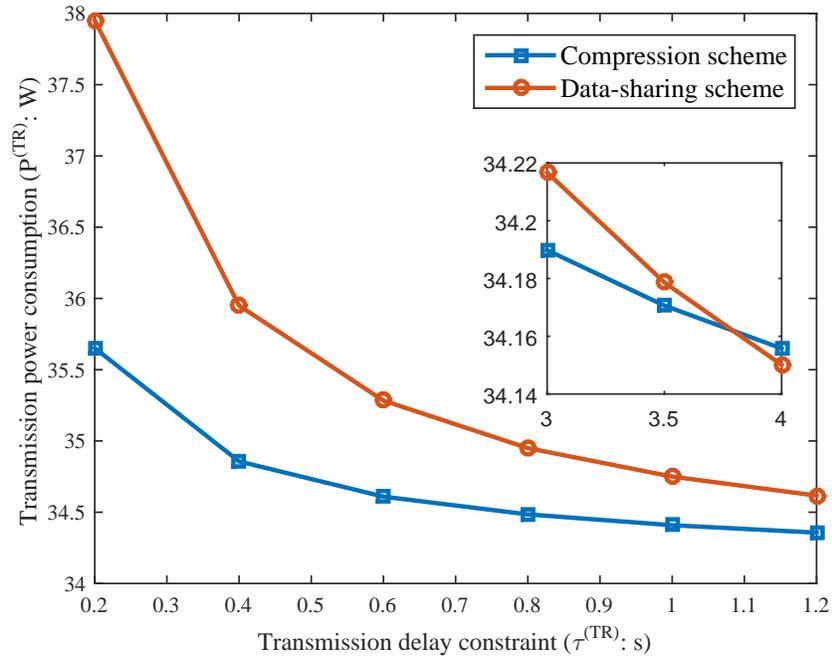


Fig. 5. Comparison of compression and data-sharing strategies.

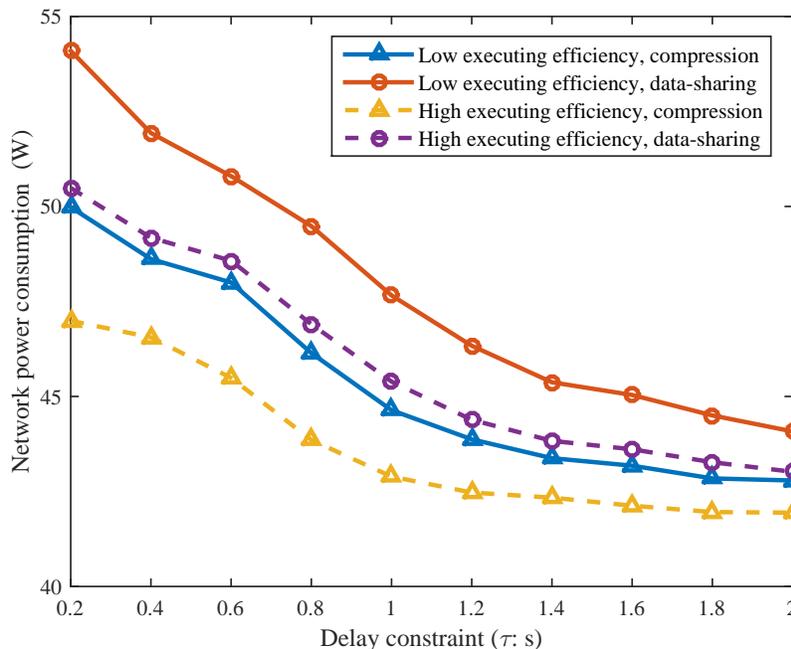


Fig. 6. Network power consumption versus different delay constraint τ values under different transmission schemes and different executing efficiency cases.

on compression shows a better performance than the data-sharing transmission scheme.

VII. CONCLUSION

In this paper, we considered the network power consumption including power consumptions for computation and transmission in a downlink C-RAN. The power minimization problem for computation was a slow time-scale problem, since the joint design of task scheduling and computing resource allocation was generally executed much slower than milliseconds. However, the power minimization problem for transmission was a fast time-scale problem because the joint optimization of power allocation and compression was based on small-scale fading. Therefore, the joint network power minimization problem was a mixed time-scale problem. To overcome the time-scale challenge, we introduced the approximate results of the original problems according to large system analysis. The approximate results were dependent on statistical channel information and independent on small-scale fading, thus the fast/mixed time-scale problem was turned into a slow time-scale one. We proposed a BnB algorithm and a combinational algorithm to find

the optimal and suboptimal solutions to the power minimization problem for computation, respectively, and introduced an iterative coordinate descent algorithm to find solutions to the power minimization problem for transmission. Then a distributed algorithm based on hierarchical decomposition was also proposed to solve the joint network power minimization problem. Simulation results showed that for the power minimization problem for computation, the combinational algorithm achieved the suboptimal solutions with much less computational complexity and time, compared to the BnB algorithm. In addition, as the delay constraint increased, suggesting the decrease of the QoS demand, the joint network power consumption was also reduced.

APPENDIX

PROOF OF LEMMA 1

Based on the law of large numbers, results (23) and (24) can be directly obtained with the following expressions [22, 41]:

$$\frac{1}{N} \tilde{\mathbf{h}}_{l,k} \tilde{\mathbf{h}}_{l,k}^\dagger \xrightarrow{N \rightarrow \infty} \frac{1}{N} \mathbf{I}_N \text{ and } \frac{1}{N} \tilde{\mathbf{h}}_{l,k} \tilde{\mathbf{h}}_{l,k'}^\dagger \xrightarrow{N \rightarrow \infty} \mathbf{0}, k' \neq k. \quad (43)$$

Then we focus on result (25). We first define a function $f(z) = \log_2 |\mathbf{H}_l \mathbf{P}_l \mathbf{H}_l^\dagger + z \mathbf{I}_N + \mathbf{\Psi}_l|$, which tends to the numerator of R_{F_l} as $z \rightarrow 0$. The derivative of $f(z)$ over z is

$$\frac{\partial f(z)}{\partial z} = \frac{1}{\log 2} \text{tr}(\mathbf{H}_l \mathbf{P}_l \mathbf{H}_l^\dagger + z \mathbf{I}_N + \mathbf{\Psi}_l)^{-1}. \quad (44)$$

Using the random matrix theory, we have

$$\text{tr}(\mathbf{H}_l \mathbf{P}_l \mathbf{H}_l^\dagger + z \mathbf{I}_N + \mathbf{\Psi}_l)^{-1} \asymp \text{tr} \left(\sum_{k \in \mathcal{N}_U} \frac{p_{l,k} d_{l,k}}{1 + e_{l,k}} \mathbf{I}_N + z \mathbf{I}_N + \mathbf{\Psi}_l \right)^{-1}. \quad (45)$$

Then according to [42, 43], we have result (25) with $z \rightarrow 0$. ■

REFERENCES

- [1] W. Xia, J. Zhang, T. Q. S. Quek, S. Jin, and H. Zhu, "Energy-efficient task scheduling and resource allocation in downlink C-RAN," in *Proc. IEEE Wireless Commun. Networking Conf. (WCNC)*, Barcelona, Spain, Apr. 2018.
- [2] P. Gandotra, R. K. Jha, and S. Jain, "Green communication in next generation cellular networks: A survey," *IEEE Access*, vol. 5, pp. 11 727–11 758, 2017.
- [3] T. Q. S. Quek, M. Peng, O. Simeone, and W. Yu, *Cloud Radio Access Networks: Principles, Technologies, and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2017.
- [4] J. Wu, Z. Zhang, Y. Hong, and Y. Wen, "Cloud radio access network (C-RAN): a primer," *IEEE Netw.*, vol. 29, no. 1, pp. 35–41, Jan. 2015.

- [5] M. Peng, C. Wang, V. Lau, and H. V. Poor, "Fronthaul-constrained cloud radio access networks: Insights and challenges," *IEEE Wireless Commun.*, vol. 22, no. 2, pp. 152–160, Apr. 2015.
- [6] S. Luo, R. Zhang, and T. J. Lim, "Downlink and uplink energy minimization through user association and beamforming in C-RAN," *IEEE Trans. Wireless Commun.*, vol. 14, no. 1, pp. 494–508, Jan. 2015.
- [7] Y. Shi, J. Zhang, and K. B. Letaief, "Group sparse beamforming for green cloud-RAN," *IEEE Trans. Wireless Commun.*, vol. 13, no. 5, pp. 2809–2823, May 2014.
- [8] B. Dai and W. Yu, "Energy efficiency of downlink transmission strategies for cloud radio access networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 4, pp. 1037–1050, Apr. 2016.
- [9] Z. Yu, K. Wang, L. Chen, and H. Ji, "Joint multiuser downlink beamforming and admission control for green cloud-RANs with limited fronthaul based on mixed integer semi-definite program," in *arXiv preprint arXiv:1610.04851*, Oct. 2016.
- [10] M. Peng, Y. Li, J. Jiang, J. Li, and C. Wang, "Heterogeneous cloud radio access networks: A new perspective for enhancing spectral and energy efficiencies," *IEEE Wireless Commun.*, vol. 21, no. 6, pp. 126–135, Dec. 2014.
- [11] C. Pan, H. Zhu, N. J. Gomes, and J. Wang, "Joint precoding and RRH selection for user-centric green MIMO C-RAN," *IEEE Trans. Wireless Commun.*, vol. 16, no. 5, pp. 2891–2906, May 2017.
- [12] K. Guo, M. Sheng, J. Tang, T. Q. S. Quek, and Z. Qiu, "Exploiting hybrid clustering and computation provisioning for green C-RAN," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 4063–4076, Dec. 2016.
- [13] K. Wang, K. Yang, and C. Magurawalage, "Joint energy minimization and resource allocation in C-RAN with mobile cloud," *IEEE Trans. Cloud Comput.*, 2018, in press.
- [14] J. Tang, W. P. Tay, and T. Q. S. Quek, "Cross-layer resource allocation with elastic service scaling in cloud radio access network," *IEEE Trans. Wireless Commun.*, vol. 14, no. 9, pp. 5068–5081, Sep. 2015.
- [15] Q. Liu, T. Han, and G. Wu, "Computing resource aware energy saving scheme for cloud radio access networks," in *Proc. IEEE Int. Conf. BDCloud-SocialCom-SustainCom*, Atlanta, GA, USA, 2016, pp. 541–547.
- [16] J. Tang, T. Q. S. Quek, C. Tsung-Hui, and S. Byonghyo, "Systematic resource allocation in cloud RAN with caching as a service under two time-scale," *IEEE J. Sel. Areas Commun.*, submitted.
- [17] J. Tang, W. P. Tay, T. Q. S. Quek, and B. Liang, "System cost minimization in cloud RAN with limited fronthaul capacity," *IEEE Trans. Wireless Commun.*, vol. 16, no. 5, pp. 3371–3384, May 2017.
- [18] L. Shi, Z. Zhang, and T. Robertazzi, "Energy-aware scheduling of embarrassingly parallel jobs and resource allocation in cloud," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 6, pp. 1607–1620, Jun. 2017.
- [19] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974–983, Apr. 2015.
- [20] A. El Gamal and Y.-H. Kim, *Network information theory*. Cambridge, U.K.: Cambridge university press, 2011.
- [21] B. Hassibi and B. M. Hochwald, "How much training is needed in multiple-antenna wireless links?" *IEEE Trans. Inf. Theory*, vol. 49, no. 4, pp. 951–963, Apr. 2003.
- [22] J. Hoydis, S. Ten Brink, and M. Debbah, "Massive MIMO in the UL/DL of cellular networks: How many antennas do we need?" *IEEE J. sel. Areas Commun.*, vol. 31, no. 2, pp. 160–171, Feb. 2013.
- [23] G. Auer, V. Giannini, C. Desset, I. Godor, P. Skillermark, M. Olsson, M. A. Imran, D. Sabella, M. J. Gonzalez, O. Blume, and A. Fehske, "How much energy is needed to run a wireless network?" *IEEE Wireless Commun.*, vol. 18, no. 5, pp. 40–49, Oct. 2011.
- [24] A. J. Fehske, P. Marsch, and G. P. Fettweis, "Bit per joule efficiency of cooperating base stations in cellular networks," in *Proc. IEEE Globecom Workshops*, Miami, FL, USA, Dec. 2010, pp. 1406–1411.

- [25] J. Wu, S. Rangan, and H. Zhang, *Green Communications: Theoretical Fundamentals, Algorithms, and Applications*. Boca Raton, FL, USA: CRC Press, 2016.
- [26] A. E. H. Bohra and V. Chaudhary, "Vmeter: Power modelling for virtualized clouds," in *Proc. IEEE Int. Symp. Parallel Distrib. Process. Workshops Phd Forum (IPDPSW)*, Atlanta, GA, USA, Apr. 2010, pp. 1–8.
- [27] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Generation Comput. Syst.*, vol. 28, no. 5, pp. 755–768, May 2012.
- [28] J. Zhang, C.-K. Wen, S. Jin, X. Gao, and K.-K. Wong, "Large system analysis of cooperative multi-cell downlink transmission via regularized channel inversion with imperfect CSIT," *IEEE Trans. Wireless Commun.*, vol. 12, no. 10, pp. 4801–4813, Oct. 2013.
- [29] W. Xia, J. Zhang, S. Jin, C. K. Wen, F. Gao, and H. Zhu, "Large system analysis of resource allocation in heterogeneous networks with wireless backhaul," *IEEE Trans. Commun.*, vol. 65, no. 11, pp. 5040–5053, Nov. 2017.
- [30] V. Bharadwaj, T. G. Robertazzi, and D. Ghose, *Scheduling Divisible Loads in Parallel and Distributed Systems*. Los Alamitos, CA, USA: IEEE Comput. Soc. Press, 1996.
- [31] E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *J. Fourier Anal. Appl.*, vol. 14, no. 5, pp. 877–905, Dec. 2008.
- [32] Q. Li, M. Hong, H.-T. Wai, Y.-F. Liu, W.-K. Ma, and Z.-Q. Luo, "Transmit solutions for MIMO wiretap channels using alternating optimization," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 9, pp. 1714–1727, Sep. 2013.
- [33] Y. Zhou and W. Yu, "Optimized backhaul compression for uplink cloud radio access network," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1295–1307, Jun. 2014.
- [34] Z.-Q. Luo, W.-K. Ma, A. M.-C. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 20–34, May 2010.
- [35] C. Helmberg, F. Rendl, R. J. Vanderbei, and H. Wolkowicz, "An interior-point method for semidefinite programming," *SIAM J. Optim.*, vol. 6, no. 2, pp. 342–361, 1996.
- [36] Q. Ye, B. Rong, Y. Chen, M. Al-Shalash, C. Caramanis, and J. G. Andrews, "User association for load balancing in heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 6, pp. 2706–2716, Jun. 2013.
- [37] K. Shen and W. Yu, "Distributed pricing-based user association for downlink heterogeneous cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1100–1113, Jun. 2014.
- [38] N. Wang, E. Hossain, and V. K. Bhargava, "Joint downlink cell association and bandwidth allocation for wireless backhauling in two-tier HetNets with large-scale antenna arrays," *IEEE Trans. Wireless Commun.*, vol. 15, no. 5, pp. 3251–3268, May 2016.
- [39] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
- [40] D. P. Bertsekas, *Convex optimization theory*. Belmont: Athena Scientific, 2009.
- [41] H. Q. Ngo, E. G. Larsson, and T. L. Marzetta, "Energy and spectral efficiency of very large multiuser MIMO systems," *IEEE Trans. Commun.*, vol. 61, no. 4, pp. 1436–1449, Apr. 2013.
- [42] C.-K. Wen, G. Pan, K.-K. Wong, M. Guo, and J. C. Chen, "A deterministic equivalent for the analysis of non-gaussian correlated MIMO multiple access channels," *IEEE Trans. Inf. Theory*, vol. 59, no. 1, pp. 329–352, Jan. 2013.
- [43] J. Zhang, C.-K. Wen, S. Jin, X. Gao, and K.-K. Wong, "On capacity of large-scale MIMO multiple access channels with distributed sets of correlated antennas," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 2, pp. 133–148, Feb. 2013.