A Novel Energy-Efficient Resource Allocation Approach in Limited Fronthaul Virtualized C-RANs

Conference Paper · April 2018

DOI: 10.1109/VTCSpring.2018.8417648

4 authors, including:

Phuong Luong
École de Technologie Supérieure

François Gagnon
École de Technologie Supérieure

Le-Nam Tran
University College Dublin

A Novel Energy-Efficient Resource Allocation Approach in Limited Fronthaul Virtualized C-RANs

Phuong Luong*, Charles Despins*, François Gagnon*, and Le-Nam Tran†
*École de Technologie Supérieure, email: thi-thu-phuong.luong.1@ens.etsmtl.ca,
{francois.gagnon,charles.despins}@etsmtl.ca †University College Dublin, email: nam.tran@ucd.ie

***Abstract*—We consider the downlink virtualized cloud-radio access networks (C-RANs) with limited capacity fronthaul. A novel virtual computing resource allocation (VCRA) which can dynamically split the users workload into smaller fragments to be served by virtual machines is presented. Under the proposed scheme, we aim at maximizing the network energy efficiency (EE) by a joint design of virtual computing resources, transmit beamforming, remote radio head (RRH) selection, and RRH-user association, considering rate dependent fronthaul power consumption model. The formulated problem is generally combinatorial and NP-hard. For an appealing solution approach, we resort to the difference of convex algorithm (DCA) to solve the continuous relaxed problem. In particular, Lipschitz continuity is derived for non-convex parts to arrive at a sequence of convex quadratic programs, which can be solved efficiently by modern convex solvers. Finally, a post-processing procedure is proposed to obtain a high-performance feasible solution from the continuous relaxation. Numerical results show that the proposed algorithms converge rapidly and the proposed scheme significantly improves the network EE compared to the existing schemes.**

I. Introduction

Fifth generation (5G) wireless network is expected to meet a set of challenging requirements (i.e., the high data rate, low latency) which are driven from the massive number of connected devices along with enormous applications. From a network architecture viewpoint, cloud-radio access networks (C-RANs) have received growing attention as a powerful candidate to implement 5G by significantly enhancing both system spectral and EE, and satisfying other quality-of-service (QoS) requirements. In C-RANs, each RRH is typically simplified with only radio frequency (RF) functions to handle the transmission/reception of radio signals to/from the users while sophisticated baseband signal processing is migrated to the virtualized BBU pool, which comprises several physical servers (PSs). At the BBU side, virtualization technique is embraced to enables each PS to dynamically split the dedicated computing resources into various virtual machines (VMs) sizes that is well adapted to varying traffic and therefore improve the efficiency of overall radio and computing resource utilization.

Recently, designing greener resource allocation for C-RANs has become one of the most attracting research trend [1], [2]. For example, distributed method in [3], [4] was proposed to optimize the EE in wireless backhaul Hetnet. In a green virtualized C-RANs, an energy-efficient design must be able to adapt computing and radio resources to elastic traffic [5] while consuming the least power. Thus, a joint design of radio resource management and network virtualization for C-RANs is necessary. [5] proposed a virtual base station (VBS) cluster associated with a subset of RRHs which can adapt to traffic dynamics. Likewise, [6] leveraged the VBS concept and optimized VBS formation using an mixed-integer linear program. Throughput maximization in C-RAN was studied in [7], taking into account the constraint of computing resource capacity of VBS pool. A joint design of VM computation capacity, RRH selection, and beamforming to minimize the total power consumption in C-RANs was proposed in [8].

In this paper we study an energy-efficient design of a virtualized C-RAN with limited-capacity fronthaul. Specifically, we consider a joint optimization of transmit beamforming, virtual computing resource allocation, RRH selection, and the RRH-user association which maximizes the network EE. Compared to the literature [5], [8], we first propose a novel virtual computing resource allocation (VCRA) scheme which can splits the users' workload into smaller pieces that can be served by different VMs in parallel. The distinguishing features of the proposed VCRA scheme are as follows: (i) data traffic of a user can be processed by heterogeneous VMs; (ii) VM's computing capacity is dynamically allocated according to the traffic condition; (iii) the assignment of VMs to PSs is done in such a way that the number of unused PSs is maximal. Further, unlike [8], we consider a more accurate power consumption model which includes the rate dependent fronthaul power consumption. We formulate the problem as a mixed-integer non-convex program, for which is generally difficult to solve. To this end, we propose a novel method based on DCA using the concept of Lipschitz continuity [9] to approximate the original problem into a series of convex quadratic programs, which can be optimally solved by dedicated convex solvers.

The rest of the paper is organized as follows. Section II introduces the system model. In Section III, we propose a low-complexity algorithm to find a high-quality feasible solution. Section IV presents numerical results and insight discussions under different simulation setups. Finally, the conclusion of the paper is given in Section V.

II. System Model and Problem Formulation

A. Transmission Model

We consider the downlink of a C-RAN consisting of $\mathcal{I} = \{1, \ldots, I\}$ set of RRHs and $\mathcal{K} = \{1, \ldots, K\}$ set of single antenna users. The ith RRH is equipped with M_i antennas,$\forall i \in \mathcal{I}$. All the RRHs are connected to BBU pool via the fronthaul links, e.g., high-speed optical ones, where the ith link has a maximum capacity C_i^{FH}. Each user is served by a specific group of RRHs and one RRH can serve more than one users simultaneously. Let s_k be the signal with unit power, i.e., $\mathbb{E}\{s_k s_k^*\} = 1$, intended for the kth user and $\mathbf{w}_{i,k} \in \mathbb{C}^{M_i \times 1}$ be the beamforming vector from the ith RRH to the kth user. The vector of channel coefficients from the ith RRH to the kth user is represented by $\mathbf{h}_{i,k} \in \mathbb{C}^{M_i \times 1}$. We denote the set of beamforming vectors intended for the kth user as $\mathbf{w}_k \triangleq [\mathbf{w}_{1,k}^T, \mathbf{w}_{2,k}^T, \ldots, \mathbf{w}_{I,k}^T]^T \in \mathbb{C}^{M \times 1}$, and channels from all RRHs to the kth user as $\mathbf{h}_k \triangleq [\mathbf{h}_{1,k}^T, \mathbf{h}_{2,k}^T, \ldots, \mathbf{h}_{I,k}^T]^T \in \mathbb{C}^{M \times 1}$, where $M = \sum_{i \in \mathcal{I}} M_i$. The received signal at the kth user is given by

$$y_k = \mathbf{h}_k^H \mathbf{w}_k s_k + \sum\nolimits_{j \in \mathcal{K} \setminus k} \mathbf{h}_k^H \mathbf{w}_j s_j + z_k \tag{1}$$

where $z_k \sim \mathcal{CN}(0, \sigma_0^2)$ is the additive white Gaussian noise (AWGN) and σ_0^2 is the noise power. We normalize $\sigma_0^2 = 1$ for the sake of notational simplicity. Note that in (1), we have assumed that the kth user is connected to all the RRHs, but the ith RRH serves the kth user only if $\left\|\mathbf{w}_{i,k}\right\|_2^2 > 0$. By treating interference as noise, the achievable rate in b/s/Hz for a given set of channel realizations at the kth user is given by $R_k(\mathbf{w}) = \log_2(1 + \Gamma_k(\mathbf{w}))$ where

$$\Gamma_k(\mathbf{w}) = \frac{|\mathbf{h}_k^H \mathbf{w}_k|^2}{\sum_{j \in \mathcal{K} \setminus k} |\mathbf{h}_k^H \mathbf{w}_j|^2 + \sigma_0^2} \tag{2}$$

where $\mathbf{w} \triangleq [\mathbf{w}_1^T, \mathbf{w}_2^T, \ldots, \mathbf{w}_k^T]^T \in \mathbb{C}^{(KM) \times 1}$.

We note that data for the kth user is routed from the BBU pool to the ith RRH via the ith fronthaul link only if $\left\|\mathbf{w}_{i,k}\right\|_2^2 > 0$. Let binary variables $a_{i,k} \in \{0, 1\}, \forall i \in \mathcal{I}, \forall k \in \mathcal{K}$ represent the association status between the ith RRH and the kth user, i.e., $a_{i,k} = 1$ implies that the kth user is served by the ith RRH and $a_{i,k} = 0$, otherwise. Then, the per-fronthaul capacity constraints can be written as

$$\sum\nolimits_{k \in \mathcal{K}} a_{i,k} R_k(\mathbf{w}) \le C_i^{\mathrm{FH}}, \forall i \in \mathcal{I}. \tag{C1}$$

B. Proposed VCRA Scheme

We consider a BBU pool consisting a set of $\mathcal{S} = \{1, \ldots, S\}$ PSs and each PS is capable of creating multiple VMs to process the incoming user packets in parallel. In contrast to [8], VCRA scheme considers a VM assignment in which one VM can only process the packets to at most one user but one user's packets can be served by several VMs with different computing capacities. To model this assignment, we introduce the binary variables $c_{s,k} \in \{0, 1\}$, $\forall k \in \mathcal{K}, \forall s \in \mathcal{S}$, where $c_{s,k} = 1$ means the packets of the kth user are processed by a VM in the sth PS and $c_{s,k} = 0$, otherwise. In addition, let binary variable $d_s \in \{0, 1\}$ and $\mathbf{d} = \{d_s, \forall s \in \mathcal{S}\}$ denote the operation mode of the sth PS, where $d_s = 0$ means the sth PS is turned OFF and $d_s = 1$ otherwise.

The packet arrival of the kth UE is assumed to follow a Poisson process with arrival rate Λ_k. For simplicity, we assume each packet has identical length. As illustrated in Fig. 1, packets of the kth user first arrive at the dispatcher and are subsequently split into smaller fragments that are then routed to VMs in different PSs for parallel processing. It is worth mentioning that each small fragment from the kth user's packets assigned to the VM in the sth PS also follows a Poisson process with arrival rate $\lambda_{s,k}$, where we have

$$\sum\nolimits_{s \in \mathcal{S}} \lambda_{s,k} = \Lambda_k, \forall k \in \mathcal{K}; \lambda_{s,k} \le c_{s,k} \Lambda_k, \tag{C2}$$

We assume that the baseband processing of each VM on each user packets can be described as a $M/M/1$ processing queue, where the service time at the VM of the sth PS follows an exponential distribution with mean $1/\mu_{s,k}$, and $\mu_{s,k}$ represents the computing capacity that the VM of sth PS can process the kth user's packets. Note that since each PS has a maximum computing capacity $C_s^{\mathrm{PS}}, \forall s \in \mathcal{S}$, we have the following constraints

$$\sum\nolimits_{k \in \mathcal{K}} \mu_{s,k} \le d_s C_s^{\mathrm{PS}}; \mu_{s,k} \le c_{s,k} C_s^{\mathrm{PS}}; c_{s,k} \le d_s \tag{C3}$$

Since the packets for the kth user can be processed by multiple VMs in different PSs, thus the effective response time τ_k to process all packets of the kth user in the BBU pool should be larger than the worst average response time among its serving VMs, which is expressed as $\tau_k = \max_{\forall s \in \mathcal{S}}\{\frac{c_{s,k}}{\mu_{s,k} - \lambda_{s,k}}\}$, $\forall k \in \mathcal{K}$, where $\frac{1}{\mu_{s,k} - \lambda_{s,k}}$ is the average response time to process each packet for the kth user at the VM of the sth PS and $\mu_{s,k} > \lambda_{s,k}$ is to guarantee the queue stability. We can equivalently rewrite the effective response time constraint as

$$\tau_k \ge \frac{c_{s,k}}{\mu_{s,k} - \lambda_{s,k}}; \mu_{s,k} > \lambda_{s,k}, \forall k \in \mathcal{K}, s \in \mathcal{S} \tag{C4}$$

After being processed by the VMs, the outcome packets from the processing queue are aggregated at a virtual switching node and transported via the corresponding fronthaul links to the RRHs and eventually transmitted to the users. For simplicity, we neglect the transportation delay. By Burke's Theorem, the arrival process of transmission queue for the kth user is still Poisson with rate Λ_k. The data transmission to the kth user from its serving RRHs can be modeled as a $M/M/1$ transmission queue service time $1/R_k(\mathbf{w})$ (cf. Fig. 1b). Therefore, total response time by a delay value D_k to ensure a low-latency transmission for each user, which is expressed as

$$\tau_k + \frac{1}{R_k(\mathbf{w}) - \Lambda_k} \le D_k, \forall k \in \mathcal{K} \tag{C5}$$

It is noteworthy that virtual computing constraints are coupled with the physical constraints via QoS delay constraint in (C5), motivating the cross-layer joint design considered in this paper.

C. Power Consumption Model

We introduce a binary variable $b_i = \{0, 1\}, \forall i \in \mathcal{I}$ to represent the operation mode of each ith RRH, where $b_i = 0$ indicates that the ith RRH is in sleep mode and $b_i = 1$ otherwise. Thus, the entire network power consumption in the considered C-RAN system is formulated as

$$\begin{aligned} P(\mathbf{w}, \boldsymbol{\mu}, \mathbf{a}, \mathbf{b}, \mathbf{d}) = &\sum\nolimits_{i \in \mathcal{I}} \overbrace{\rho_i \sum\nolimits_{k \in \mathcal{K}} a_{i,k} R_k(\mathbf{w})}^{P_i^{\mathrm{FH}}(\mathbf{w},\mathbf{a})} \\ &+ \sum\nolimits_{i \in \mathcal{I}} \left(\overbrace{\frac{1}{\eta_i} \sum\nolimits_{k \in \mathcal{K}} \left\|\mathbf{w}_{i,k}\right\|_2^2 + b_i P_i^{\mathrm{ra}} + (1 - b_i) P_i^{\mathrm{ri}}}^{P_i^{\mathrm{RRH}}(\mathbf{w},\mathbf{b})} \right) \\ &+ \phi \sum\nolimits_{s \in \mathcal{S}} \left(\overbrace{d_s P_s^{PS} + \sum\nolimits_{\forall k \in \mathcal{K}} \kappa_s \mu_{s,k}^{\alpha_s}}^{P_s^{\mathrm{BBU}}(\boldsymbol{\mu},\mathbf{d})} \right) \end{aligned} \tag{3}$$

where $\eta_i \in [0, 1]$ is the power amplifier efficiency, ρ_i is the scaling factor of the ith fronthaul, $\phi > 0$ is a parameter to

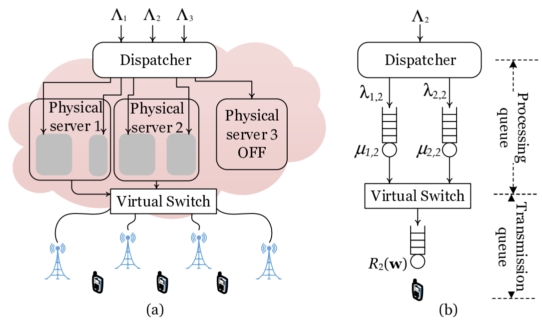


Fig. 1. (a) Limited fronthaul C-RANs with VCRA scheme, (b) Queuing model, i.e., for UE 2

strike a balance between the power consumption of RRHs, fronthaul and BBU pool. P_i^{ra} and P_i^{ri} are the powers to keep each ith RRH active and idle, respectively, P_s^{PS} and $\kappa_s \mu_{s,k}^{\alpha_s}$ are the powers spent by the sth PS and the associated VMs for processing the kth user's traffic, respectively, where $\kappa_s > 0, \alpha_s > 1$, $\mathbf{b} = [b_1, \ldots, b_I]^T$, $\mathbf{a} = \left[\mathbf{a}_1^T, \ldots, \mathbf{a}_I^T\right]^T$ and $\boldsymbol{\mu} = \{\mu_{s,k}, \forall s \in \mathcal{S}, \forall k \in \mathcal{K}\}$. Additionally, $P_i^{\text{RRH}}(\mathbf{w}, \mathbf{b})$, $P_i^{\text{FH}}(\mathbf{w}, \mathbf{a})$ and $P_s^{\text{BBU}}(\boldsymbol{\mu}, \mathbf{d})$ denote the power consumption at ith RRH, ith fronthaul and sth PS, respectively.

D. Problem Formulation

We aim at jointly optimizing the virtual computing resource allocation with beamforming, RRH selection and RRH-user association to maximize the network EE. To guarantee the stability of the transmission queue and the QoS requirement for each user k, we impose the following constraint

$$R_k(\mathbf{w}) \geq \max\left\{R_k^{\min}, \Lambda_k\right\} \tag{C6}$$

Moreover, the total transmit power at each RRH is limited by a power budget $P^{\max}$, which is expressed as

$$\sum\nolimits_{k \in \mathcal{K}} \left\|\mathbf{w}_{i,k}\right\|_2^2 \leq b_i P^{\max}; \;\; \left\|\mathbf{w}_{i,k}\right\|_2^2 \leq a_{i,k} P^{\max}; \;\; a_{i,k} \leq b_i \tag{C7}$$

The above constraint implies that when the ith RRH is in sleep mode, e.g., $b_i = 0$, no power will be transmitted from it. Similarly, we also guarantee that the transmit power $\left\|\mathbf{w}_{i,k}\right\|_2^2$ from the ith RRH to the kth user is zero if $a_{i,k} = 0$. Also, when $b_i = 0$, $a_{i,k} = 0$ for all $k \in \mathcal{K}$ and $\sum_{k \in \mathcal{K}} \left\|\mathbf{w}_{i,k}\right\|_2^2 = 0$. Now the considered problem is formulated as

$$(\mathcal{P}_0): \underset{\substack{\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d}\\ \boldsymbol{\lambda},\boldsymbol{\tau},\mathbf{w},\boldsymbol{\mu}}}{\text{maximize}} \;\; \frac{\sum_{k \in \mathcal{K}} R_k(\mathbf{w})}{P\left(\mathbf{w}, \boldsymbol{\mu}, \mathbf{a}, \mathbf{b}, \mathbf{d}\right)} \tag{4a}$$

$$\text{s.t.} \;\; \text{(C1); (C2); (C3); (C4); (C5); (C6); (C7)} \tag{4b}$$

where $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, $\mathbf{d}$ are implicitly understood to be binary.

III. Low-complexity Method

We will present a low-complexity algorithm in this section by resorting the DC programming to first solve the continuous relaxation of $(\mathcal{P}_0)$, denoted as $(\mathcal{P}_0^r)$ where we relax all the binary variables $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, $\mathbf{d}$ into continuous ones within $[0, 1]$.

A. DC Decomposition

We recall that the non-convexity of $(\mathcal{P}_0^r)$ is due to that of function $R_k(\mathbf{w})$ and also the term $a_{i,k} R_k(\mathbf{w})$. Based on the concept of DC programming, we will express each of the non-convex function $R_k(\mathbf{w})$ as a difference of two convex ones as

$$R_k(\mathbf{w}) = \underbrace{R_k(\mathbf{w}) + \xi_k \left\|\mathbf{w}\right\|_2^2}_{f_k(\mathbf{w})} - \xi_k \left\|\mathbf{w}\right\|_2^2 \tag{5}$$

for any $\xi_k > 0$. Intuitively if ξ_k is sufficiently large, the quadratic term $\xi_k \left\|\mathbf{w}\right\|_2^2$ will dominate $R_k(\mathbf{w})$ and thus $f_k(\mathbf{w})$ becomes convex eventually. The issue is that finding a proper value for ξ_k to make (5) a DC expression is very challenging. In this regard, we propose the following lemma.

Lemma 1. *For $\xi_k > \bar{\xi}_k$, where $\bar{\xi}_k$ is given in (23) in Appendix A, $f_k(\mathbf{w})$ is strongly convex .*

Proof: The proof and the derivation of $\bar{\xi}_k$ in Lemma 1 are involved and all the detailed algebra is presented in Appendix A. The idea is to show that $R_k(\mathbf{w})$ is $\bar{\xi}_k$-smooth, i.e.,

$$\left\|\nabla R_k(\mathbf{x}) - \nabla R_k(\mathbf{y})\right\|_2 \leq \bar{\xi}_k \left\|\mathbf{x} - \mathbf{y}\right\|_2 \tag{6}$$

where $\nabla f(\mathbf{x})$ is the gradient of $f(\mathbf{x})$ w.r.t. $\mathbf{x}$. Equivalently, $\nabla R_k(\mathbf{x})$ is Lipschitz continuous with a constant $\bar{\xi}_k$. □

By the same way, we propose the following DC decomposition of the term $a_{i,k} R_k(\mathbf{w}) = \gamma_k \left(\left\|\mathbf{w}\right\|_2^2 + a_{i,k}^2\right) - \underbrace{\left(\gamma_k \left(\left\|\mathbf{w}\right\|_2^2 + a_{i,k}^2\right) - a_{i,k} R_k(\mathbf{w})\right)}_{u_k(\mathbf{w}, a_{i,k})}$ and the following lemma is in order.

Lemma 2. *For $\gamma_k > \bar{\gamma}_k$ where $\bar{\gamma}_k$ is given in (29) in Appendix A, $u_k(\mathbf{w}, a_{i,k})$ is strongly convex.*

Proof: The proof of Lemma 2 follows the same steps as those for that of Lemma 1. □

Based on the above DC decomposition, we are now in a position to describe the proposed algorithm to solve $(\mathcal{P}_0^r)$ efficiently. First $(\mathcal{P}_0^r)$ can be equivalently rewritten as

$$\underset{\substack{\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d}\\ \boldsymbol{\lambda},\boldsymbol{\tau},\mathbf{w},\boldsymbol{\mu}}}{\max} \;\; \frac{\sum_{k \in \mathcal{K}} (f_k(\mathbf{w}) - \xi_k \left\|\mathbf{w}\right\|_2^2)}{\tilde{P}\left(\mathbf{w}, \boldsymbol{\mu}, \mathbf{a}, \mathbf{b}, \mathbf{d}\right)} \tag{7a}$$

$$\text{s.t.} \;\; f_k(\mathbf{w}) - \xi_k \left\|\mathbf{w}\right\|_2^2 \geq \max\left\{R_k^{\min}, \Lambda_k\right\} \tag{7b}$$

$$\sum_{k \in \mathcal{K}} \left(\gamma_k \left(\left\|\mathbf{w}\right\|_2^2 + a_{i,k}^2\right) - u_k(\mathbf{w}, a_{i,k})\right) \leq C_i^{\text{FH}} \tag{7c}$$

$$c_{s,k} + \frac{(\tau_k - \mu_{s,k} + \lambda_{s,k})^2}{4} \leq \frac{(\tau_k + \mu_{s,k} - \lambda_{s,k})^2}{4} \tag{7d}$$

$$f_k(\mathbf{w}) - \xi_k \left\|\mathbf{w}\right\|_2^2 - \Lambda_k \geq \frac{1}{D_k - \tau_k} \tag{7e}$$

$$a_{i,k}, b_i, c_{s,k}, d_s \in [0, 1] \tag{7f}$$

$$\text{(C2); (C3); (C7)} \tag{7g}$$

where $\tilde{P}\left(\mathbf{w}, \boldsymbol{\mu}, \mathbf{a}, \mathbf{b}, \mathbf{d}\right) = \sum_{i \in \mathcal{I}} P_i^{\text{RRH}}(\mathbf{w}, b_i) + \phi \sum_s P_s^{\text{BBU}}(\mathbf{d}, \boldsymbol{\mu}) + \sum_{i \in \mathcal{I}} \rho_i \sum_{k \in \mathcal{K}} \left(\gamma_k \left(\left\|\mathbf{w}\right\|_2^2 + a_{i,k}^2\right) - u_k(\mathbf{w}, a_{i,k})\right)$. Note that we have equivalently rewritten constraints (C4) and (C5) as (7d) and (7e), respectively. The purpose of these reformulations is to express (7) as a DC program that is amenable to application of DCA, which is presented next subsection.

B. DCA-based Algorithm

In this section we apply DCA-based method to solve (7). In particular, we can approximate (7b) as

$$F_k(\mathbf{w}; \mathbf{w}^{(n)}) - \xi_k \left\|\mathbf{w}\right\|_2^2 \geq \max\left\{R_k^{\min}, \Lambda_k\right\} \tag{8}$$

where $F_k(\mathbf{w}; \mathbf{w}^{(n)})$ is given in (9) shown at the top of the next page. Note that $F_k(\mathbf{w}; \mathbf{w}^{(n)})$ is a linearization of $f_k(\mathbf{w})$ around $\mathbf{w}^{(n)}$ and its derivation is in fact a by-product of Appendix A. Similarly, constraint (7e) can be approximated as

$$F_k(\mathbf{w}; \mathbf{w}^{(n)}) - \xi_k \left\|\mathbf{w}\right\|_2^2 - \Lambda_k \geq \frac{1}{D_k - \tau_k} \tag{10}$$

$$F_k(\mathbf{w};\mathbf{w}^{(n)}) = f_k(\mathbf{w}^{(n)}) + \frac{\sum_{j\in\mathcal{K}}\left(2\,\mathrm{Re}\big(\mathbf{w}_j^{(n)H}\mathbf{H}_k\mathbf{w}_j\big) - 2\mathbf{w}_j^{(n)H}\mathbf{H}_k\mathbf{w}_j^{(n)}\right)}{\sum_{j\in\mathcal{K}}|\mathbf{h}_k^H\mathbf{w}_j^{(n)}|^2+\sigma_0^2} - \frac{\sum_{j\in\mathcal{K}\setminus k}\left(2\,\mathrm{Re}\big(\mathbf{w}_j^{(n)H}\mathbf{H}_k\mathbf{w}_j\big) - 2\mathbf{w}_j^{(n)H}\mathbf{H}_k\mathbf{w}_j^{(n)}\right)}{\sum_{j\in\mathcal{K}\setminus k}|\mathbf{h}_k^H\mathbf{w}_j^{(n)}|^2+\sigma_0^2} + 2\xi_k\,\mathrm{Re}\big(\mathbf{w}^{(n)H}\mathbf{w}\big) - 2\xi_k\left\|\mathbf{w}^{(n)}\right\|_2^2 \quad (9)$$

which is a convex constraint since $1/(D_k-\tau_k)$ is convex and $F_k(\mathbf{w};\mathbf{w}^{(n)}) - \xi_k\|\mathbf{w}\|_2^2 - \Lambda_k$ is concave w.r.t all feasible variables $\mathbf{w},\tau_k$. In the same way, we can also derive the lower bound concave approximation of $u_k(\mathbf{w},a_{i,k})$ by $\tilde{F}_k(\mathbf{w},a_{i,k};\mathbf{w}^{(n)},a_{i,k}^{(n)})$ shown in (12) and approximate the left side of constraint (7c) by its concave upper bound as

$$\sum_{k\in\mathcal{K}}\left(\gamma_k\left(\|\mathbf{w}\|_2^2+a_{i,k}^2\right) - \tilde{F}_k(\mathbf{w},a_{i,k};\mathbf{w}^{(n)},a_{i,k}^{(n)})\right) \le C_i^{\mathrm{FH}} \quad (11)$$

Alternatively, we apply the first order approximation to the right side of constraint (7d) to obtain

$$c_{s,k}+\frac{(\tau_k-\mu_{s,k}+\lambda_{s,k})^2}{4} \le \frac{\tau_k^{(n)}+\mu_{s,k}^{(n)}-\lambda_{s,k}^{(n)}}{2}(\tau_k+\mu_{s,k}-\lambda_{s,k}) - \frac{(\tau_k^{(n)}+\mu_{s,k}^{(n)}-\lambda_{s,k}^{(n)})^2}{4} \quad (13)$$

By applying the above approximations, we can formulate the approximation of (7) at iteration $n+1$ as

$$\max_{\substack{\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d}\\ \boldsymbol{\lambda},\boldsymbol{\tau},\mathbf{w},\boldsymbol{\mu}}} \frac{\sum_{k\in\mathcal{K}}\left(F_k(\mathbf{w};\mathbf{w}^{(n)}) - \xi_k\|\mathbf{w}\|_2^2\right)}{\hat{P}\left(\mathbf{w},\boldsymbol{\mu},\mathbf{a},\mathbf{b},\mathbf{d};\mathbf{w}^{(n)},a_{i,k}^{(n)}\right)} \quad (14a)$$

$$\text{s.t.}(C2);(C3);(C7);(7f);(8);(10);(11);(13) \quad (14b)$$

where

$$\hat{P}\left(\mathbf{w},\boldsymbol{\mu},\mathbf{a},\mathbf{b},\mathbf{d};\mathbf{w}^{(n)},a_{i,k}^{(n)}\right) = \sum_{i\in\mathcal{I}}P_i^{\mathrm{RRH}}(\mathbf{w},b_i) + \phi\sum_s P_s^{\mathrm{BBU}}(\mathbf{d},\boldsymbol{\mu}) + \sum_{i\in\mathcal{I}}\rho_i\sum_{k\in\mathcal{K}}\gamma_k\left(\|\mathbf{w}\|_2^2+a_{i,k}^2\right) - \sum_{i\in\mathcal{I}}\rho_i\sum_{k\in\mathcal{K}}\tilde{F}_k(\mathbf{w},a_{i,k};\mathbf{w}^{(n)},a_{i,k}^{(n)}) \quad (15)$$

Note that the fractional objective (14a) can be easily transformed into a linear subtractive form using Dinkelback approach. This subsequently makes (14a) concave and can be solved by the following DCA-based algorithm, which is outlined in Algorithm 1.

Algorithm 1 DCA-based Algorithm for solving (7).

1: Set $n := 0$ and initialize starting points of $\mathbf{w}^{(n)},\boldsymbol{a}^{(n)},\boldsymbol{\tau}^{(n)},\boldsymbol{\mu}^{(n)},\boldsymbol{\lambda}^{(n)}$;
2: **repeat**
3: Solve the approximated problem (14) at $\mathbf{w}^{(n)},\boldsymbol{a}^{(n)},\boldsymbol{\tau}^{(n)},\boldsymbol{\mu}^{(n)},\boldsymbol{\lambda}^{(n)}$ to achieve the optimal solution $\mathbf{a}^\star,\mathbf{b}^\star,\mathbf{c}^\star,\mathbf{d}^\star,\boldsymbol{\lambda}^\star,\boldsymbol{\tau}^\star,\mathbf{w}^\star,\boldsymbol{\mu}^\star$;
4: Set $n := n+1$;
5: Update $\mathbf{w}^{(n)} = \mathbf{w}^\star, \mathbf{a}^{(n)} = \mathbf{a}^\star, \boldsymbol{\tau}^{(n)} = \boldsymbol{\tau}^\star, \boldsymbol{\mu}^{(n)} = \boldsymbol{\mu}^\star, \boldsymbol{\lambda}^{(n)} = \boldsymbol{\lambda}^\star$
6: **until** Convergence of the objective (14a);

C. An Accelerated Version of Algorithm 1

As being shown, the sufficient conditions of ξ_k and γ_k to ensure the DC forms of functions (and thus the convergence of Algorithm 1) are that $\xi_k \ge \bar{\xi}_k$ and $\gamma_k \ge \bar{\gamma}_k$. However, smaller values of ξ_k and γ_k may significantly increase the convergence rate of Algorithm 1 in practice since they can lead to tighter approximation in each iteration of Algorithm 1. This is easily seen from (5) where $f_k(\mathbf{w})$ is close to $R_k(\mathbf{w})$ for small ξ_k. Thus, we set ξ_k and γ_k to a small value in each iteration of Algorithm 1. If monotonic increase of the objective is not achieved, we then set ξ_k and γ_k to $\bar{\xi}_k$ and $\bar{\gamma}_k$, respectively.

D. Post-Processing Algorithm

A post-processing step is proposed to map the relaxed variables from solving (7) to the binary values, which is required due to the continuous relaxation. The process starts by assuming that all the RRHs and PSs are OFF and there is no association between RRHs, VMs and users. Let us denote $\mathbf{a}^\star,\mathbf{b}^\star,\mathbf{c}^\star,\mathbf{d}^\star$ as the solution achieved by solving (7). The RRH-UE association and VM assignment are then gradually updated by fixing untreated relaxed variables to be 1. Intuitively, the connection between the ith RRH and the kth user is more likely if the channel link condition is good and the power consumed to transmit fronthaul data is smaller than the others. Similarly, the kth user is preferred to be processed by VM in the sth PS if the power expended for switching on the sth PS is the smaller than the others and the total signal processing power consumed in the sth PS is larger. Consequently, solving the continuous relaxation would possibly yield higher $a_{i,k}^\star$ for the connection between the ith RRH and the kth UE and higher $c_{s,k}^\star$ for assigning VM in sth PS to the kth user. According to constraints in (C3) and (C7), the variables b_i and d_s, $\forall i,s$ are fixed with respect to its associated variables $\mathbf{a}_i = \{a_{i,k},\forall k\in\mathcal{K}\}$ and $\mathbf{c}_s = \{c_{s,k},\forall k\in\mathcal{K}\}$, respectively. For example, we need to set $b_i = 1$ or $d_s = 1$ if we fix any $a_{i,k} = 1$ or $c_{s,k} = 1$. The process stops if the problem (7) starts to be infeasible or it is feasible but the objective value is smaller than that obtained from the previous iteration. The detail of post processing algorithm is similar to Algorithm 3 in [10]. We skip its presentation here due to space limitation.

IV. Numerical Results

We employ the parameters in Table I in our simulations. We consider a network consisting of I=6 RRHs which are uniformly located and $K = 4$ users are randomly scattered across the considered network coverage. The path-loss is modeled as $(d_{ik}/d_0)^{-3}$ where d_{ik} is the distance between the ith RRH and the kth user and $d_0 = 100$ m is the reference distance. We set $\kappa_s = 10^{-26}$, $\mu_{s,k}$ is in cycle/s and $\mu_{s,k}$ b/s $= (8/1900)\times\mu_{s,k}$ cycle/s is used to compute the processing response time for users [8]. We set $\xi_k = \gamma_k = 0.1$ in accelerated algorithm. Algorithm 1 is terminated when the increase in the objective between two consecutive iterations is less than 10^{-5}.

In Fig. 2(a), Algorithm 1 needs a much smaller number of iterations to converge, compared to the DB-based WMMSE algorithm in [11]. As expected, the accelerated version of

$$\tilde{F}_k(\mathbf{w}, a_{i,k}; \mathbf{w}^{(n)}, a_{i,k}^{(n)}) = \tilde{f}_k(\mathbf{w}^{(n)}, a_{i,k}^{(n)}) + \left[\log(\sum_{j\in\mathcal{K}} |\mathbf{h}_k^H \mathbf{w}_j^{(n)}|^2 + \sigma_0^2) - \log(\sum_{j\in\mathcal{K}\setminus k} |\mathbf{h}_k^H \mathbf{w}_j^{(n)}|^2 + \sigma_0^2)\right](a_{i,k} - a_{i,k}^{(n)})$$
$$+ a_{i,k}^{(n)} \left[\frac{\sum_{j\in\mathcal{K}} \left(2\,\mathrm{Re}\big(\mathbf{w}_j^{(n)H}\mathbf{H}_k\mathbf{w}_j\big) - 2\mathbf{w}_j^{(n)H}\mathbf{H}_k\mathbf{w}_j^{(n)}\right)}{\sum_{j\in\mathcal{K}} |\mathbf{h}_k^H\mathbf{w}_j^{(n)}|^2 + \sigma_0^2} - \frac{\sum_{j\in\mathcal{K}\setminus k} \left(2\,\mathrm{Re}\big(\mathbf{w}_j^{(n)H}\mathbf{H}_k\mathbf{w}_j\big) - 2\mathbf{w}_j^{(n)H}\mathbf{H}_k\mathbf{w}_j^{(n)}\right)}{\sum_{j\in\mathcal{K}\setminus k} |\mathbf{h}_k^H\mathbf{w}_j^{(n)}|^2 + \sigma_0^2}\right]$$
$$+ 2\gamma_k\,\mathrm{Re}\big(\mathbf{w}^{(n)H}\mathbf{w} + a_{i,k}^{(n)}a_{i,k}\big) - 2\gamma_k\left(\left\|\mathbf{w}^{(n)}\right\|_2^2 + a_{i,k}^{(n)2}\right) \quad (12)$$

TABLE I
SIMULATION PARAMETERS

Notation	Value	Notation	Value
P_s	17 dBW	M_i	2
η_i	0.35	$P^{\max}$	10 dBW
P_i^{ra}	12.5 dBW	P_i^{ri}	2.5 dBW
ϕ, ρ_i, $\forall i$	1	$C_i^{\text{FH}} = C^{\text{FH}}, \forall i$	15 b/s/Hz
$C_i^{\text{PS}} = C^{\text{PS}}, \forall s$	2.5×10^3 cycle/s	$D_k = D, \forall k$	0.5 s
S	4	α_s	3

Algorithm 1 achieves an improved convergence rate due to the reason explained in III-C. Moreover, the convergence of the post-processing is also presented. It is clear that the objective value achieved by post-processing algorithm at convergence is very close to the optimal value which can be achieved by applying the branch and bound method in [10]. Furthermore, the DB-based WMMSE algorithm converges to a smaller objective compared to that achieved by our proposed algorithms. This is explained that the DB-based WMMSE algorithm simply assign a fixed rate obtained from the previous iteration to overcome the non-convex fronthaul constraint while we directly tackle it by proposing the novel DC transformations and approximations. This demonstrates the superiority of our proposed algorithms.

In Fig. 2(b), we compare the EE performance of our proposed algorithms with both DB-based WMMSE and SCA-based algorithms in [11] and [7], respectively, with respect to C_i^{FH}. We set $C_i^{\text{FH}} = C^{\text{FH}}, \forall i \in \mathcal{I}$. We first observe that when C^{FH} increases, EE increases accordingly. This is because larger fronthaul capacities allows that higher data rate is transported and this subsequently requires a smaller number of activated RRHs to serve the demanding users. However, EE curves saturate at the high fronthaul capacity regime because there always exists the multi-user interference. Furthermore, our proposed algorithms outperforms the DB-based WMMSE and SCA-based algorithm in terms of achieved EE.

Next we compare the EE performance of our proposed VCRA scheme and the rate-dependent fronthaul power consumption (RDFP) model to other existing schemes. Proposed VCRA without the RDFP model is considered as scheme A. Instead of using the RDFP model, a linear fronthaul power consumption model is set as $P_i^{\text{FH}} = \sum_{k\in\mathcal{K}} a_{i,k} P_{i,k}^{\text{FH}}$ where $P_{i,k}^{\text{FH}}$ is a fixed power consumption, i.e., $P_{i,k}^{\text{FH}} = 2$ Watts, $\forall k \in \mathcal{K}, \forall i \in \mathcal{I}$. Scheme B considers non segmentation and thus the entire workload of one user is served by only one VM (c.f., [8]). In scheme C, PS switching ON/OFF is not considered (c.f., [8]). Fig. 2(c) plots the EE performance of the above listed schemes as a function of the workload arrival rates. Here, we set $\Lambda_k = \Lambda$, $\forall k \in \mathcal{K}$. From Fig. 2(c), it is obviously seen that the EE decreases when Λ increases. In addition, our proposed scheme outperforms scheme A, which verifies the benefit of considering the RDFH model in the formulated C-RAN optimization problem. Moreover, there is a remarkably large performance gap between our proposed scheme and schemes B and C. This again validates the advantages of our proposed VCRA scheme over the others in comparison.

V. CONCLUSION

We have considered the joint design of VCRA and radio resource allocation in a limited fronthaul C-RAN under a RDFP model for maximizing the network EE. We have developed a novel low-complexity algorithm inspired by the DCA method combined with the Lipschitz continuity concept to approximate the non-convex problem into a sequence of convex quadratic ones. A post-processing routine was executed to find a high-performance solution which is feasible to the original problem. Numerical results have showed that our proposed algorithms converge rapidly and outperform the other existing methods.

APPENDIX A
PROOF OF LEMMA 1

In this section, we show that $\bar{\xi}_k$ in Lemma 1 is a Lipschitz constant of $\nabla R_k(\mathbf{w})$, which is then used to prove Lemma 1. Since there is a mapping rule from a complex-valued vector into a real domain and also due to the space limitation, we will treat $\mathbf{w}$ as a real-valued vector in the following. For ease of mathematical presentation, let us rewrite $R_k(\mathbf{w})$ as

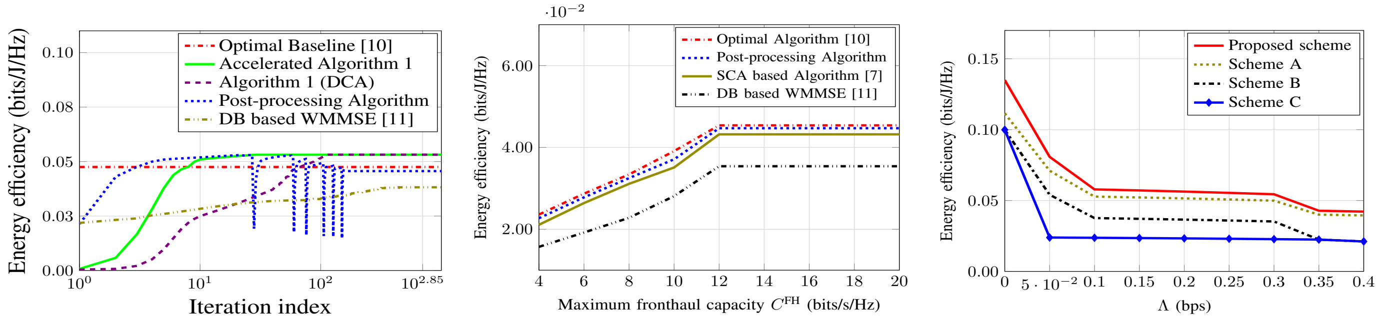
$$R_k(\mathbf{w}) = \log\left(\sigma_0^2 + \mathbf{w}^H\mathbf{H}_k\mathbf{w}\right) - \log\left(\sigma_0^2 + \mathbf{w}^H\mathbf{G}_k\mathbf{w}\right) \quad (16)$$

where $\mathbf{H}_k = \text{blkdiag}(\tilde{\mathbf{H}}_k^{[1]}, \ldots, \tilde{\mathbf{H}}_k^{[K]})$, and $\mathbf{G}_k = \text{blkdiag}(\tilde{\mathbf{H}}_k^{[1]}, \ldots, \mathbf{0}^{[k]} \ldots, \tilde{\mathbf{H}}_k^{[K]})$, (the kth diagonal element $\tilde{\mathbf{H}}_k^{[k]}$is written as the superscript, i.e., $[k]$ and $\tilde{\mathbf{H}}_k = \mathbf{h}_k\mathbf{h}_k^H$). Then the gradient of $R_k(\mathbf{w})$ is given by

$$\begin{aligned}\nabla R_k(\mathbf{w}) &= \frac{2\mathbf{w}^H\mathbf{H}_k}{\mathbf{w}^H\mathbf{H}_k\mathbf{w} + \sigma_0^2} - \frac{2\mathbf{w}^H\mathbf{G}_k}{\mathbf{w}^H\mathbf{G}_k\mathbf{w} + \sigma_0^2} \\ &= h_1(\mathbf{w})\,h_2(\mathbf{w}) - g_1(\mathbf{w})\,g_2(\mathbf{w}) \end{aligned} \quad (17)$$

where $h_1(\mathbf{w}) = 2\mathbf{w}^H\mathbf{H}_k$, $h_2(\mathbf{w}) = \frac{1}{\mathbf{w}^H\mathbf{H}_k\mathbf{w} + \sigma_0^2}$, $g_1(\mathbf{w}) = 2\mathbf{w}^H\mathbf{G}_k$, and $g_2(\mathbf{w}) = \frac{1}{\mathbf{w}^H\mathbf{G}_k\mathbf{w} + \sigma_0^2}$. Next we will find a Lipschitz constant of each term in (17). From the definition of $h_1(\mathbf{w})$, the following inequality holds

$$\begin{aligned}\|h_1(\mathbf{w}) - h_1(\bar{\mathbf{w}})\|_2 &= \left\|\left(\mathbf{w}^H - \bar{\mathbf{w}}^H\right)\mathbf{H}_k\right\|_2 \\ &\le \|\mathbf{H}_k\|_F\,\|(\mathbf{w} - \bar{\mathbf{w}})\|_2 \end{aligned} \quad (18)$$



(a) (b) (c)

Fig. 2. Comparison of: (a)-(b) Convergence behavior and achieved EE at different algorithms; (c) Achieved EE at different schemes.

In words, a Lipschitz constant of $h_1(\mathbf{w})$ is $\|\mathbf{H}_k\|_F$. Next we have

$$\begin{aligned} \|h_2(\mathbf{w}) - h_2(\bar{\mathbf{w}})\|_2 &= \left\| \frac{\bar{\mathbf{w}}^H \mathbf{H}_k \bar{\mathbf{w}} - \mathbf{w}^H \mathbf{H}_k \mathbf{w}}{(\mathbf{w}^H \mathbf{H}_k \mathbf{w} + \sigma_0^2)(\bar{\mathbf{w}}^H \mathbf{H}_k \bar{\mathbf{w}} + \sigma_0^2)} \right\|_2 \\ &\le \left\| \bar{\mathbf{w}}^H \mathbf{H}_k \bar{\mathbf{w}} - \bar{\mathbf{w}}^H \mathbf{H}_k \mathbf{w} + \bar{\mathbf{w}}^H \mathbf{H}_k \mathbf{w} - \mathbf{w}^H \mathbf{H}_k \mathbf{w} \right\|_2 \\ &\le \|\bar{\mathbf{w}}\|_2 \|\mathbf{H}_k\|_F \|\mathbf{w} - \bar{\mathbf{w}}\|_2 + \|\mathbf{w}\|_2 \|\mathbf{H}_k\|_F \|\mathbf{w} - \bar{\mathbf{w}}\|_2 \\ &\le 2P \|\mathbf{H}_k\|_F \|\mathbf{w} - \bar{\mathbf{w}}\|_2 \end{aligned} \tag{19a}$$

where $P = \sqrt{IP^{\max}}$. Note that the last inequality occurs due to the sum power constraint. Using (18) and (19) we can find a Lipschitz constant of the product $h_1(\mathbf{w}) h_2(\mathbf{w})$ as

$$\begin{aligned} &\|h_1(\mathbf{w}) h_2(\mathbf{w}) - h_1(\bar{\mathbf{w}}) h_2(\bar{\mathbf{w}})\|_2 \\ &\le \|h_2(\mathbf{w})\|_2 \|h_1(\mathbf{w}) - h_1(\bar{\mathbf{w}})\|_2 + \|h_1(\bar{\mathbf{w}})\|_2 \|h_2(\mathbf{w}) - h_2(\bar{\mathbf{w}})\|_2 \\ &\le \left(\|\mathbf{H}_k\|_F + \left(2P \|\mathbf{H}_k\|_F\right)^2 \right) \|(\mathbf{w} - \bar{\mathbf{w}})\|_2 \end{aligned} \tag{20a}$$

Following the same algebraic manipulations, Lipschitz constants of $g_1(\mathbf{w})$ and $g_2(\mathbf{w})$ are $\|\mathbf{G}_k\|_F$ and $2P\|\mathbf{G}_k\|_F$, respectively. Thus, a Lipschitz constant of $g_1(\mathbf{w}) g_2(\mathbf{w})$ is simply given by

$$\begin{aligned} &\|g_1(\mathbf{w}) g_2(\mathbf{w}) - g_1(\bar{\mathbf{w}}) g_2(\bar{\mathbf{w}})\| \\ &\qquad \le \left(\|\mathbf{G}_k\|_F + 4P^2 \|\mathbf{G}_k\|_F^2 \right) \|(\mathbf{w} - \bar{\mathbf{w}})\|_2 \end{aligned} \tag{21}$$

Combining (20a) and (21) results in

$$\|\nabla R_k(\mathbf{w}) - \nabla R_k(\bar{\mathbf{w}})\|_2 \quad \le \quad \bar{\xi}_k \|(\mathbf{w} - \bar{\mathbf{w}})\|_2 \tag{22}$$

where

$$\bar{\xi}_k = \|\mathbf{H}_k\|_F + \left(2P\|\mathbf{H}_k\|_F\right)^2 + \|\mathbf{G}_k\|_F + \left(2P\|\mathbf{G}_k\|_F\right)^2 \tag{23}$$

In other words $R_k(\mathbf{w})$ is a $\bar{\xi}_k$-smooth function. We now prove that $f_k(\mathbf{w}) = R_k(\mathbf{w}) + \xi_k \|\mathbf{w}\|_2^2$ is strongly convex. Since $R_k(\mathbf{w})$ is $\bar{\xi}_k$-smooth, the following inequality holds

$$\left| R_k(\mathbf{w}) - R_k(\bar{\mathbf{w}}) - \nabla R_k(\bar{\mathbf{w}})^T (\mathbf{w} - \bar{\mathbf{w}}) \right| \le \frac{\bar{\xi}_k}{2} \|\mathbf{w} - \bar{\mathbf{w}}\|^2 \tag{24}$$

which implies

$$R_k(\mathbf{w}) \ge -\frac{\bar{\xi}_k}{2} \|\mathbf{w} - \bar{\mathbf{w}}\|^2 + R_k(\bar{\mathbf{w}}) + \nabla R_k(\bar{\mathbf{w}})^T (\mathbf{w} - \bar{\mathbf{w}}) \tag{25}$$

Due to the strong convexity of $\xi_k \|\mathbf{w}\|_2^2$ we have

$$\xi_k \|\mathbf{w}\|_2^2 \ge \xi_k \|\bar{\mathbf{w}}\|_2^2 + 2\xi_k \bar{\mathbf{w}}^T (\mathbf{w} - \bar{\mathbf{w}}) + \frac{\xi_k}{2} \|\mathbf{w} - \bar{\mathbf{w}}\|^2 \tag{26}$$

Combining (25) and (26) we obtain

$$\begin{aligned} R_k(\mathbf{w}) + \xi_k \|\mathbf{w}\|_2^2 &\ge \frac{\xi_k - \bar{\xi}_k}{2} \|\mathbf{w} - \bar{\mathbf{w}}\|^2 + R_k(\bar{\mathbf{w}}) + \\ &\xi_k \|\bar{\mathbf{w}}\|_2^2 + \mathrm{Re}\left\{ \left(\nabla R_k(\bar{\mathbf{w}}) + 2\xi_k \bar{\mathbf{w}}\right)^T (\mathbf{w} - \bar{\mathbf{w}}) \right\} \end{aligned} \tag{27}$$

which is equivalent to

$$f_k(\mathbf{w}) \ge \frac{\xi_k - \bar{\xi}_k}{2} \|\mathbf{w} - \bar{\mathbf{w}}\|^2 + f_k(\bar{\mathbf{w}}) + \nabla f_k(\bar{\mathbf{w}})^T (\mathbf{w} - \bar{\mathbf{w}}) \tag{28}$$

(28) implies that $f_k(\mathbf{w})$ is $(\xi_k - \bar{\xi}_k)$-strongly convex, $\forall \xi_k > \bar{\xi}_k$ which completes the proof. Similar to the proof of Lemma 1, the strong convexity of $u_k(\mathbf{w}, a_{i,k})$ is proved if we show that $\nabla\big(a_{i,k} R_k(\mathbf{w})\big)$ has a Lipschitz constant of γ_k where

$$\bar{\gamma}_k = \sqrt{2\bar{\xi}_k^2 + 8\left(\|\mathbf{H}_k\|_F^2 + \|\mathbf{G}_k\|_F^2\right) P^2}. \tag{29}$$

References

[1] P. Luong, C. Despins, F. Gagnon, and L.-N. Tran, "Designing green C-RAN with limited fronthaul via mixed-integer second order cone programming," in *Proc. IEEE Int. Conf. Communications (ICC'17)*, Paris, France, May 2017, pp. 1–6.

[2] P. Luong, F. Gagnon, C. Despins, and L.-N. Tran, "Joint virtual computing and radio resource allocation in limited fronthaul green C-RANs," *IEEE Trans. Wireless Commun.*, vol. pp, no. 99, pp. 1–16, Feb. 2018.

[3] T. M. Nguyen, A. Yadav, W. Ajib, and C. Assi, "Centralized and distributed energy efficiency designs in wireless backhaul hetnets," *IEEE Trans. Wireless Commun.*, vol. 16, no. 7, pp. 4711–4726, July 2017.

[4] ——, "Achieving energy-efficiency in two-tier wireless backhaul hetnets," in *Proc. IEEE Int. Conf. Communications (ICC'16)*, Kuala Lumpur, Malaysia, May 2016, pp. 1–6.

[5] D. Pompili, A. Hajisami, and T. X. Tran, "Elastic resource utilization framework for high capacity and energy efficiency in cloud RAN," *IEEE Commun. Mag.*, vol. 54, no. 1, pp. 26–32, Jan. 2016.

[6] X. Wang, S. Thota, M. Tornatore, H. S. Chung, H. H. Lee, S. Park, and B. Mukherjee, "Energy-efficient virtual base station formation in optical-access-enabled Cloud-RAN," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1130–1139, May 2016.

[7] T. X. Tran and D. Pompili, "Dynamic radio cooperation for User-Centric Cloud-RAN with computing resource sharing," *IEEE Trans. Wireless Commun.*, To appear.

[8] 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.

[9] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in optimization*, vol. 1, no. 3, Nov. 2014.

[10] P. Luong, F. Gagnon, C. Despins, and L.-N. Tran, "Optimal joint remote radio head selection and beamforming design for limited fronthaul C-RAN," *IEEE Trans. Signal Process.*, vol. 65, no. 21, pp. 5605–5620, Nov. 2017.

[11] M. Peng *et al.*, "Energy-efficient resource allocation optimization for multimedia heterogeneous cloud radio access networks," *IEEE Trans. Multimedia*, vol. 18, no. 5, pp. 879–892, May 2016.