MIN-MAX LATENCY OPTIMIZATION FOR MULTIUSER COMPUTATION OFFLOADING IN FOG-RADIO ACCESS NETWORKS

Qiang Li, Jin Lei and Jingran Lin

School of Information and Communication Engineering
University of Electronic Science and Technology of China, Chengdu, P. R. China, 611731
E-mail: {lq, jingranlin}@uestc.edu.cn

ABSTRACT

This paper considers mobile computation offloading in fog-radio access networks (F-RAN), where multiple mobile users offload their computation tasks to the F-RAN through a number of fog nodes [a.k.a. enhanced remote radio heads (RRHs)]. In addition to communication capability, the fog nodes are also equipped with computational resources to provide computing services for users. Each user chooses one fog node to offload its task, while each fog node may simultaneously serve multiple users. Depending on computational burden at the fog nodes, the tasks may be completed at the fog nodes or further offloaded to the cloud via fronthaul links with limited capacities. To complete all the tasks as fast as possible, a joint optimization of radio and computational resources of F-RAN is proposed to minimize the maximum latency of all users. This problem is formulated as a mixed integer nonlinear program (MINP). We first show that the MINP can be reformulated as a continuous optimization problem with a difference-of-convex (DC) objective. Then, an inexact DC algorithm is proposed to handle the min-max problem with stationary convergence guarantee. Simulation results show that the proposed algorithm outperforms the minimum distance-based and the random-based offloading strategies.

***Index Terms*—** Fog-radio access networks, computation offloading, DC programming

1. INTRODUCTION

The fifth generation wireless communication systems are expected to provide ubiquitous connections for massive heterogenous devices with high speed and low latency. The current cloud-computing-based network infrastructure is facing challenges to meet these requirements, because massive heterogenous requests with different data sizes and latency requirements need to be forwarded to and processed at the central baseband processing units (BBUs), and this could cause heavy burden on the fronthaul and incur intolerable latency for some delay-critical missions. For example, in some interactive applications, e.g., virtual reality, industrial automation and vehicle-to-vehicle communications, the roundtrip delay may be required below a few tens of millisecond [1]. To meet the critical latency requirement and alleviate the pressure on the fronthaul, a fog-computing-based radio access network (F-RAN) has recently been considered as a promising solution [2]. The concept of F-RAN is developed from the fog computing, which was originally proposed by Cisco [3]. By shifting certain amount of computing, storage and networking functions from the cloud to the edge of networks, F-RAN is able to provide more prompt responses to users' requests with less fronthaul bandwidth occupation.

Evolving from cloud-computing-based RAN (C-RAN) to F-RAN, the wireless access point (AP) is endowed with more capabilities and functions. In this work, we focus on the computational aspect of F-RAN, and investigate how the enhanced APs (also called fog nodes in the rest of the paper) near the mobile users can help improve the latency performance in computation offloading applications. Specifically, we consider multiple low-performance mobile user equipments (UEs), each of which has a computing-intensive task to be offloaded to F-RAN. Each UE can access F-RAN through one of the fog nodes, and the tasks can be performed at the fog nodes or the cloud (BBU), depending on the computational loads and the fronthaul capacities at the fog nodes. Due to limited communication and computation resources of F-RAN, UEs are competing with each other. To guarantee fairness, we adopt a min-max latency criterion to optimize the communication and computation resources (including the UE-Fog association, transmit beamforming at UEs, computation tasks distribution between fogs and the cloud, computation resources and the fronthaul capacity allocations), so that the worst latency of all UEs induced by transmission and computation is as small as possible. This min-max latency optimization problem is formulated as a mixed integer nonlinear program (MINP). We show that the MINP can be reformulated as a continuous optimization problem. By employing the difference-of-convex (DC) programming [4] and the weighted MMSE (WMMSE) method [5], we develop an iterative algorithm to compute a solution for the min-max problem with stationary convergence guarantee.

There are some related works worth mentioning. The works [6] and [7] consider a joint optimization of radio and computational resources for energy minimization with latency constraints in single cell and multicell networks, respectively, where all the computation is done at the cloud and the UE-BS association is prefixed. In [8,9], a cooperative computation model is considered, but their focus is more on choosing appropriate number of fog nodes for each task, given the communication resource constraints. In [10, 11], Chen *et'al* studied the energy-plus-delay minimization for computation offloading with multiple UEs, one computing AP (or fog node) and a cloud server. Since there is only one computing AP, no UE-AP association optimization is needed, and moreover transmit beamforming is not included in their model. Recently, the work [12] deals with a similar problem as [10, 11] under the setting of one UE and multiple APs without considering further computation offloading from the AP to the cloud.

This work was supported in part by the National Natural Science Foundation of China under Grant 61531009, 61671120, and in part by the Fundamental Research Funds for the Central Universities under Grants ZYGX2016J011 and ZYGX2016J007

2. SYSTEM MODEL AND PROBLEM STATEMENT

Consider an F-RAN, consisting of K multi-antenna mobile users, L fog nodes and a cloud server. Each user has a computation task, however, due to limited computational capability at the users, all the tasks have to be offloaded to the F-RAN via the fog nodes. Suppose that user k's task T_k is described by a two-tuple of (D_k, B_k) integers, where D_k denotes the number of CPU cycles needed for completing T_k, and B_k represents the number of bits needed for encoding T_k. In other words, to offload the task to F-RAN, user k has to send the B_k bits to the fog nodes through wireless links. For simplicity, we assume that each user gets access to F-RAN through one fog node, while each fog node may simultaneously provide access for multiple users. The association between the fog nodes and the users is not prefixed and needs to be jointly optimized with other radio and computational resources. To highlight this, we introduce a binary variable $\alpha_{k,\ell} \in \{0,\ 1\}$ to reflect the association relationship. In particular,

$$\alpha_{k,\ell} = \begin{cases} 1, & \text{if user } k \text{ is served by fog node } \ell, \\ 0, & \text{otherwise,} \end{cases}$$

and $\sum_{\ell=1}^{L} \alpha_{k,\ell} = 1,\ \forall\, k \in \mathcal{K} \triangleq \{1, \ldots, K\}$.

Now, the offloading process can be described in the following two stages:

Stage 1: Wireless Transmissions from Users to Fog Nodes. For ease of exposition, let us assume that user k is associated with some fog node $\ell \in \mathcal{L} \triangleq \{1, \ldots, L\}$, i.e., $\alpha_{k,\ell} = 1$ and $\alpha_{k,\ell'} = 0, \forall \ell' \neq \ell$. Let $\boldsymbol{v}_k \in \mathbb{C}^{N_k}$ be the transmit beamformer of user k with N_k being the number of antennas at user k. Then, the received signal at the fog node ℓ is given by[1]

$$\boldsymbol{y}_\ell(t) = \boldsymbol{H}_{k,\ell}^H \boldsymbol{v}_k s_k(t) + \textstyle\sum_{j \neq k} \boldsymbol{H}_{j,\ell}^H \boldsymbol{v}_j s_j(t) + \boldsymbol{n}_\ell(t),$$

where $\boldsymbol{H}_{j,\ell} \in \mathbb{C}^{N_j \times M_\ell}$ is the channel between user j and fog node ℓ with M_ℓ being the number of antennas at fog node ℓ; $s_j(t) \in \mathbb{C}$ is the encoded signal of task T_j, and $\boldsymbol{n}_\ell(t) \sim \mathcal{CN}(\boldsymbol{0}, \sigma_\ell^2 \boldsymbol{I})$ is additive white Gaussian noise. The communication rate between user k and fog ℓ is given by

$$R_{k,\ell} = W \log\Big(1 + \boldsymbol{v}_k^H \boldsymbol{H}_{k,\ell}\big(\sigma_\ell^2 \boldsymbol{I} + \sum_{j \neq k} \boldsymbol{H}_{j,\ell}^H \boldsymbol{v}_j \boldsymbol{v}_j^H \boldsymbol{H}_{j,\ell}\big)^{-1} \boldsymbol{H}_{k,\ell}^H \boldsymbol{v}_k\Big), \tag{1}$$

where W (Hz) is the bandwidth of the wireless transmission. The corresponding wireless transmission delay may be calculated as

$$\tau_{k,\ell}^T = \frac{B_k}{R_{k,\ell}}. \tag{2}$$

Stage 2: Computing at Fog Nodes or Cloud. After reception, the fog node ℓ needs to determine whether user k's task should be processed by itself or further offloaded to the cloud based on its current computation load and the complexity of T_k. There are two cases:

1. Computing at the fog node. In such a case, let $f_{k,\ell}^F$ (in cycles/second) be the number of CPU cycles allocated to execute T_k in every second. Then, the time delay induced by computation is given by

$$\tau_{k,\ell}^F = \frac{D_k}{f_{k,\ell}^F}. \tag{3}$$

2. Computing at the cloud. In such a case, the processing delay consists of two parts. One is the transmission delay from the fog node to the cloud, and the other is the computing time at the cloud. We consider that fog ℓ is connected with the cloud via fronthaul with limited capacity $C_{\ell,\max}$ (bits/s). Let $C_{k,\ell}(\leq C_{\ell,\max})$ be the fronthaul capacity allocated by fog ℓ to further offload T_k to the cloud. Then, the total processing delay at the cloud may be expressed as

$$\tau_{k,\ell}^C = \frac{B_k}{C_{k,\ell}} + \frac{D_k}{f_k^C}, \tag{4}$$

where f_k^C (in cycles/second) is the CPU cycles allocated by the cloud to execute T_k in every second.

To differentiate the above two cases, we introduce a binary variable $\beta_k \in \{0,\ 1\}$ to indicate where the computation is performed. In particular,

$$\beta_k = \begin{cases} 0, & \text{if fog performs computation,} \\ 1, & \text{if the cloud performs computation.} \end{cases}$$

Based on the above offloading model, our goal is to optimize the communication and computational resources, including the UE-Fog association $\alpha_{k,\ell}$, the task distribution β_k, the beamforming $\boldsymbol{v}_k$, the fronthaul link capacity allocation $C_{k,\ell}$ and the CPU cycles $f_{k,\ell}^F$ and f_k^C, so that the worst transmission-plus-computation latency is minimized:[2]

$$\begin{aligned}
\min_{\substack{\{\boldsymbol{v}_k, f_k^C, \beta_k\}_k, \\ \{f_{k,\ell}^F, C_{k,\ell}, \alpha_{k,\ell}\}_{k,\ell}}} \quad & \max_{k \in \mathcal{K}} \sum_{\ell=1}^{L} \alpha_{k,\ell} \left(\tau_{k,\ell}^T + (1-\beta_k)\tau_{k,\ell}^F + \beta_k \tau_{k,\ell}^C\right) \\
\text{s.t.} \quad & f_{k,\ell}^F \leq \alpha_{k,\ell}(1-\beta_k) F_{\ell,\max},\ \forall\, k, \ell, && \text{(5a)} \\
& \sum_{k=1}^{K} f_{k,\ell}^F \leq F_{\ell,\max}, \quad f_{k,\ell}^F \geq 0,\ \forall\, k, \ell, && \text{(5b)} \\
& f_k^C \leq \beta_k F_{C,\max},\ \forall\, k, && \text{(5c)} \\
& \sum_{k=1}^{K} f_k^C \leq F_{C,\max}, \quad f_k^C \geq 0,\ \forall\, k, && \text{(5d)} \\
& C_{k,\ell} \leq \alpha_{k,\ell} \beta_k C_{\ell,\max},\ \forall k,\ \ell, && \text{(5e)} \\
& \sum_{k=1}^{K} C_{k,\ell} \leq C_{\ell,\max}, \quad C_{k,\ell} \geq 0,\ \forall\, k,\ \ell, && \text{(5f)} \\
& \|\boldsymbol{v}_k\|^2 \leq P_k,\ \forall\, k, && \text{(5g)} \\
& \alpha_{k,\ell} \in \{0,\ 1\}, \quad \sum_{\ell=1}^{L} \alpha_{k,\ell} = 1,\ \forall\, k, && \text{(5h)} \\
& \beta_k \in \{0,\ 1\},\ \forall\, k, && \text{(5i)}
\end{aligned}$$

where $F_{\ell,\max}$ and $F_{C,\max}$ are the maximum CPU cycles per second of fog ℓ and the cloud, respectively. The constraints (5a)-(5b) correspond to the computation resource allocation at fog ℓ. In particular, (5a) implies that fog ℓ will allocate CPU cycles for user k only if $\alpha_{k,\ell} = 1$ and $\beta_k = 0$, i.e., user k is associated with fog ℓ, and meanwhile the task T_k is processed at fog ℓ. Similarly, the constraints (5c)-(5d) correspond to the computational resource allocation at the

[1] For simplicity, single data stream is assumed for each user, and the generalization to multiple data streams is straightforward.

[2] We consider a situation where the computing outputs contain very few bits, and thus can be delivered to users with negligible time.

cloud. The constraints (5e)-(5f) are introduced to account for the finite capacity of fronthaul, and (5g) limits the peak transmit powers at the users.

Problem (5) is a mixed integer nonlinear program (MINP), which is generally hard to solve. In the next section, we show that problem (5) can be reformulated as a discrete-variable-free form, and continuous optimization algorithms can be employed to handle it.

3. A TRACTABLE APPROACH TO PROBLEM (5)

Theorem 1 *The MINP problem* (5) *is equivalent to the following continuous optimization problem:*

$$\min_{\substack{\{\boldsymbol{v}_k, f_k^C\}_k,\\ \{f_{k,\ell}^F, C_{k,\ell}, \theta_{k,\ell}^F, \theta_{k,\ell}^C\}_{k,\ell}}} \max_{k\in\mathcal{K}} \sum_{\ell=1}^{L} (\theta_{k,\ell}^F(\tau_{k,\ell}^T + \tau_{k,\ell}^F) + \theta_{k,\ell}^C(\tau_{k,\ell}^T + \tau_{k,\ell}^C)) \tag{6a}$$

$$\text{s.t. } \theta_{k,\ell}^F \geq 0, \quad \theta_{k,\ell}^C \geq 0, \, \forall\, k, \ell, \tag{6b}$$

$$\sum_{\ell=1}^{L} \theta_{k,\ell}^F + \theta_{k,\ell}^C = 1, \, \forall\, k, \tag{6c}$$

$$\text{(5b), (5d), (5f) and (5g) satisfied.} \tag{6d}$$

Proof. Due to the page limit, we just give a sketched proof here. We first show that problem (6) is a relaxation of problem (5), i.e., every feasible solution of problem (5) is feasible for problem (6). Next, we show that for any optimal solution of problem (6), we can convert it into an optimal solution of problem (5) with the same optimal value, thereby establishing the equivalence of the two problems. ■

In view of Theorem 1, we consider solving problem (6). Let us denote $\boldsymbol{\theta}_k = [\theta_{k,1}^F, \ldots, \theta_{k,L}^F, \theta_{k,1}^C, \ldots, \theta_{k,L}^C]^T \in \mathbb{R}^{2L}$ and $\boldsymbol{\tau}_k = [\tau_{k,1}^T + \tau_{k,1}^F, \ldots, \tau_{k,L}^T + \tau_{k,L}^F, \tau_{k,1}^T + \tau_{k,1}^C, \ldots, \tau_{k,L}^T + \tau_{k,L}^C]^T \in \mathbb{R}^{2L}$. Problem (6) is rewritten as

$$\min_{\substack{\{\boldsymbol{v}_k, f_k^C, \boldsymbol{\tau}_k, \boldsymbol{\theta}_k\}_k,\\ \{R_{k,\ell}, f_{k,\ell}^F, C_{k,\ell}\}_{k,\ell}}} \max_{k\in\mathcal{K}} \boldsymbol{\theta}_k^T \boldsymbol{\tau}_k \tag{7a}$$

$$\text{s.t. } R_{k,\ell} \leq \phi_{k,\ell}(\boldsymbol{V}), \, \forall\, k, \ell, \tag{7b}$$

$$\tau_{k,\ell}^T \geq \frac{B_k}{R_{k,\ell}}, \;\; \tau_{k,\ell}^F \geq \frac{D_k}{f_{k,\ell}^F}, \;\; \tau_{k,\ell}^C \geq \frac{B_k}{C_{k,\ell}} + \frac{D_k}{f_k^C}, \quad \forall\, k, \ell, \tag{7c}$$

$$\text{(5b), (5d), (5f), (5g), (6b) and (6c) satisfied.} \tag{7d}$$

where $\boldsymbol{V} \triangleq \{\boldsymbol{v}_k\}_k$ and $\phi_{k,\ell}(\boldsymbol{V}) \triangleq W \log\Big(1 + \boldsymbol{v}_k^H \boldsymbol{H}_{k,\ell}\big(\sigma_\ell^2 \boldsymbol{I} + \sum_{j\neq k} \boldsymbol{H}_{j,\ell}^H \boldsymbol{v}_j \boldsymbol{v}_j^H \boldsymbol{H}_{j,\ell}\big)^{-1} \boldsymbol{H}_{k,\ell}^H \boldsymbol{v}_k\Big)$. Notice that in (7b) and (7c) we have changed the equalities in (1)-(4) as inequalities. This does not incur any loss of optimality because the inequalities in (7b) and (7c) must be active at the optimal solution; otherwise we can further decrease $\tau_{k,\ell}^X, X \in \{T, F, C\}$ and increase $R_{k,\ell}$ to get a lower objective value.

The constraints (7c) and (7d) are convex, but the objective (7a) and the constraint (7b) are still nonconvex. Since the objective can be written as the following DC form

$$\boldsymbol{\theta}_k^T \boldsymbol{\tau}_k = \frac{\|\boldsymbol{\theta}_k + \boldsymbol{\tau}_k\|^2 - (\|\boldsymbol{\theta}_k\|^2 + \|\boldsymbol{\tau}_k\|^2)}{2}, \tag{8}$$

DC programming can be employed to handle problem (7). Specifically, let $\mathcal{X} \triangleq \{\boldsymbol{v}_k, f_k^C, \boldsymbol{\tau}_k, \boldsymbol{\theta}_k, R_{k,\ell}, f_{k,\ell}^F, C_{k,\ell}\}$ be a collection of optimization variables, and $(\boldsymbol{\theta}_k^{(0)}, \boldsymbol{\tau}_k^{(0)})$ be some starting point. The DC programming repeatedly performs the following updates

$$\begin{aligned} &(\boldsymbol{\theta}_k^{(t+1)}, \boldsymbol{\tau}_k^{(t+1)}) \\ &= \arg\min_{\mathcal{X}} \max_{k\in\mathcal{K}} \frac{\|\boldsymbol{\theta}_k + \boldsymbol{\tau}_k\|^2}{2} - (\boldsymbol{\theta}_k^{(t)})^T \boldsymbol{\theta}_k - (\boldsymbol{\tau}_k^{(t)})^T \boldsymbol{\tau}_k \\ &\qquad \text{s.t.} \quad \text{(7b)} - \text{(7d) satisfied,} \end{aligned} \tag{9}$$

for $t = 0, 1, 2, \ldots$ until some stopping criterion is satisfied.

According to the classical DC convergence result [4], we have that every limit point of the iterates generated by (9) is a stationary point of problem (7). However, this convergence result holds under the premise that each DC subproblem is *optimally* solved. As for the considered problem (9), it may be hard to do so due to the nonconvex constraints (7b). To circumvent this difficulty, we apply the WMMSE method [5] to find an approximate solution for (9). Specifically, define $\boldsymbol{u}_{k,\ell} \in \mathbb{C}^M$ as the receive beamformer employed at node ℓ to receive user k's signal. Then, the rate function $\phi_{k,\ell}(\boldsymbol{V})$ can be represented as the following WMMSE form:

$$\phi_{k,\ell}(\boldsymbol{V}) = \max_{\boldsymbol{u}_{k,\ell}, w_{k,\ell} \geq 0} f_{k,\ell}(\boldsymbol{u}_{k,\ell}, w_{k,\ell}, \boldsymbol{V}) \tag{10}$$

where $f_{k,\ell}(\cdot)$ is defined as

$$f_{k,\ell}(\boldsymbol{u}_{k,\ell}, w_{k,\ell}, \boldsymbol{V}) = W\left(-w_{k,\ell} e_{k,\ell}(\boldsymbol{u}_{k,\ell}, \boldsymbol{V}) + \log(w_{k,\ell}) + 1\right)$$

and $e_{k,\ell}(\boldsymbol{u}_{k,\ell}, \boldsymbol{V})$ is the MSE of estimating user k's signal at fog ℓ with receive beamformer $\boldsymbol{u}_{k,\ell}$, which takes the following form:

$$\begin{aligned} &e_{k,\ell}(\boldsymbol{u}_{k,\ell}, \boldsymbol{V}) \\ &= \|1 - \boldsymbol{u}_{k,\ell}^H \boldsymbol{H}_{k,\ell}^H \boldsymbol{v}_k\|^2 + \sum_{j\neq k} \|\boldsymbol{v}_j^H \boldsymbol{H}_{j,\ell} \boldsymbol{u}_{k,\ell}\|^2 + \sigma_\ell^2 \|\boldsymbol{u}_{k,\ell}\|^2. \end{aligned} \tag{11}$$

By substituting the WMMSE reformulation (10) into (9), the DC subproblem is equivalently written as

$$\begin{aligned} \min_{\mathcal{X}, \{\boldsymbol{u}_{k,\ell}, w_{k,\ell}\}_{k,\ell}} \; & \max_{k\in\mathcal{K}} \frac{\|\boldsymbol{\theta}_k + \boldsymbol{\tau}_k\|^2}{2} - (\boldsymbol{\theta}_k^{(t)})^T \boldsymbol{\theta}_k - (\boldsymbol{\tau}_k^{(t)})^T \boldsymbol{\tau}_k \\ & \text{s.t. } R_{k,\ell} \leq f_{k,\ell}(\boldsymbol{u}_{k,\ell}, w_{k,\ell}, \boldsymbol{V}), \, \forall\, k, \ell, \\ & \quad \text{(7c)} - \text{(7d) satisfied,} \end{aligned} \tag{12}$$

which can be efficiently handled by block coordinate descent (BCD) method. In particular, given $\boldsymbol{V}$ the optimal $\boldsymbol{u}_{k,\ell}$ and $w_{k,\ell}$ for (12) is given by [5]

$$\boldsymbol{u}_{k,\ell} = (\sigma_\ell^2 \boldsymbol{I} + \textstyle\sum_{j=1}^{K} \boldsymbol{H}_{j,\ell}^H \boldsymbol{v}_j \boldsymbol{v}_j^H \boldsymbol{H}_{j,\ell})^{-1} \boldsymbol{H}_{k,\ell}^H \boldsymbol{v}_k, \tag{13a}$$

$$w_{k,\ell} = e_{k,\ell}^{-1}(\boldsymbol{u}_{k,\ell}, \boldsymbol{V}). \tag{13b}$$

Given $(\boldsymbol{u}_{k,\ell}, w_{k,\ell})$, problem (12) is convex with respect to the remaining variables, and thus can be optimally solved, e.g., by `CVX` [13]. Theoretically speaking, the above BCD procedure needs to be performed sufficiently large number of rounds in order to obtain a good approximate solution for problem (9). However, this could incur high computational complexity for each DC update. To tradeoff the solution quality and the computational complexity, we propose a computationally-cheap inexact DC algorithm for problem (7); see Algorithm 1, where for the tth DC iteration, we perform only a small number $J^{(t)}$ rounds of BCD update to compute an approximate solution for problem (9). The parameter $J^{(t)}$ controls the solution quality for each DC iteration. Although Algorithm 1 performs DC iterations with approximate solutions, the following theorem reveals that the same convergence result as the exact DC (i.e., using the optimal solution of (9) to perform DC iterations) holds.

Theorem 2 *Every limit point generated by the inexact DC is a stationary point of problem* (7).

The idea of proving Theorem 2 is that the inexact DC update (even for the case of $J^{(t)} = 1, \forall\, t$) is sufficient to provide certain improvement for the objective (7a). By accumulating these improvements, the DC iterations will finally reside at a stationary point of problem (7). The detailed proof is omitted due to the page limit.

Algorithm 1 An Inexact DC Algorithm for Solving Problem (7)

```
Initialize a feasible point $(\boldsymbol{\theta}^{(0)}, \boldsymbol{\tau}^{(0)}, \boldsymbol{V}^{(0)})$, a set of small positive integers $\{J^{(t)}\}_{t=0,1,\ldots}$ and set $t = 0$
repeat
    Set $\boldsymbol{V}^{(t_0)} = \boldsymbol{V}^{(t)}$;
    for $j = 0, 1, \ldots, J^{(t)}$ do
        Update $\boldsymbol{u}_{k,\ell}^{(t_j)}$ according to (13a) with $\boldsymbol{V} = \boldsymbol{V}^{(t_j)}$;
        Update $w_{k,\ell}$ according to (13b) with $(\boldsymbol{V}, \boldsymbol{u}_{k,\ell}) = (\boldsymbol{V}^{(t_j)}, \boldsymbol{u}_{k,\ell}^{(t_j)})$;
        Update $\boldsymbol{V}^{(t_{j+1})}$ by solving (12) with fixed $(\boldsymbol{u}_{k,\ell}, w_{k,\ell}) = (\boldsymbol{u}_{k,\ell}^{(t_j)}, w_{k,\ell}^{(t_j)})$;
    end for
    Set $\boldsymbol{V}^{(t+1)} = \boldsymbol{V}^{(t_{j+1})}$;
    $t \leftarrow t + 1$
until some stopping criterion is satisfied
Output $\mathcal{X}^{(t)}$.
```

4. SIMULATION RESULTS

In this section, we test the performance of Algorithm 1 by simulations. The following simulation settings are used: $N_j = 4$, $M_\ell = 8$, $\forall\, j \in \mathcal{K}, \ell \in \mathcal{L}$, $P_k = 10^3$, $\forall\, k \in \mathcal{K}$, $\sigma_\ell^2 = 1$, $\forall\, \ell \in \mathcal{L}$, $W = 20\,\text{MHz}$ and $J^{(t)} = 1$, $\forall\, t$. There are four fog nodes and ten users which are randomly distributed in a cell with radius 1 Km. The channels are randomly generated according to distance model; the channel coefficients between user k and fog ℓ are modeled as zero mean circularly symmetric complex Gaussian vector with $(2000/d_{k,\ell})^3 \beta_{k,\ell}$ as variance for both real and imaginary dimensions, where $10 \log 10(\beta_{k,\ell}) \sim \mathcal{CN}(0, 64)$ is a real Gaussian random variable modeling the shadowing effect.

Fig. 1 shows the convergence behavior of the proposed inexact DC algorithm, where we have set $F_{C,\max} = 2 \times 10^3$ (Giga cycles/second), $F_{1:4,\max} = [3,\ 4,\ 4,\ 5] \times 10^2$ (Giga cycles/second), $C_{1:4,\max} = [30,\ 35,\ 40,\ 50]$ (Mbps), $D_{1:10} = [2, 2, 2, 6, 6, 6, 6, 8, 8, 8] \times 10^2$ (Mega cycles) and $B_{1:10} = [20, 20, 20, 40, 40, 40, 40, 60, 60, 60]$ (Kilobits). From Fig. 1 we see that the maximum delay decreases monotonically and converges after 25 iterations. Fig. 2 shows the corresponding UE-Fog association and the task distribution after convergence of Fig. 1. The solid black line means that the computation is performed at the fog, whereas the blue broken line means that at the cloud. From the figure we see that the UE-Fog association is not solely determined by the distance, and most of the users offload tasks to the fourth fog node, which has the most powerful communication and computational capabilities.

Fig. 3 studies the relationship between the number of users and the maximum latency for different offloading strategies. Two UE-Fog association strategies are compared, namely the minimum distance based association and the random-based association, under which the fog nodes equally allocate their resources for their served users. The number of users increases from 2 to 11 according to the settings in Figs. 1-2. We see from Fig. 3 that the proposed inexact DC outperforms the other offloading strategies, due to the optimized UE-Fog association as well as more balanced task distributions between fog nodes and the cloud.

5. CONCLUSIONS

We have studied a multiuser computation offloading problem in fog-radio access networks. To guarantee the worst delay performance of all users, a joint communication and computational resource allocation problem is formulated as a min-max mixed integer nonlinear program. By leveraging a continuous reformulation, we develop an efficient iterative solution for the min-max problem with stationary convergence guarantee. Simulation results corroborate the effectiveness of our proposed method.

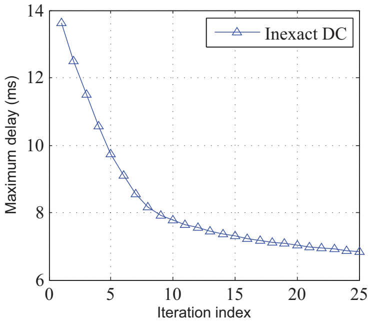

Fig. 1: Convergence behavior of inexact DC.

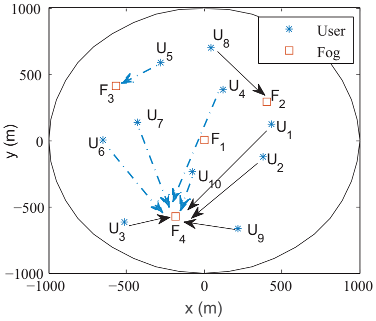

Fig. 2: UE-Fog association after convergence.

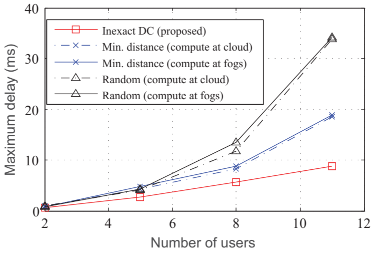

Fig. 3: Maximum delay vs. number of users.

6. REFERENCES

[1] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[2] M. Peng, S. Yan, K. Zhang, and C. Wang, "Fog-computing-based radio access networks: Issues and challenges," *IEEE Network*, vol. 30, no. 4, pp. 46–53, July 2016.

[3] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proc. ACM SIGCOMM Workshop on Mobile Cloud Computing*, 2012, pp. 13–16.

[4] B. K. Sriperumbudur and G. R. Lanckriet, "On the convergence of the concave-convex procedure," in *Proc. 22nd Adv. Neural Inf. Process. Syst.*, 2009, pp. 1759–1767.

[5] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, "An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel," *IEEE Trans. Signal Process.*, vol. 59, no. 9, pp. 4331–4340, Sep. 2011.

[6] M. Salmani and T. N. Davidson, "Multiple access computational offloading with computation constraints," in *IEEE Workshop on Sig. Proc. Adv. in Wireless Commun.*, July 2017, pp. 385–389.

[7] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 1, no. 2, pp. 89–103, Jun. 2015.

[8] T.-C. Chiu, W.-H. Chung, A.-C. Pang, Y.-J. Yu, and P.-H. Yen, "Ultra-low latency service provision in 5G fog-radio access networks," in *Proc. of IEEE PIMRC*, Sept. 2016.

[9] A.-C. Pang, W.-H. Chung, T.-C. Chiu, and J. Zhang, "Latency-driven cooperative task computing in multi-user fog-radio access networks," in *IEEE 37th International Conference on Distributed Computing Systems*, June 2017, pp. 615–624.

[10] M.-H. Chen, B. Liang, and M. Dong, "A semidefinite relaxation approach to mobile cloud offloading with computing access point," in *IEEE Workshop on Sig. Proc. Adv. in Wireless Commun.*, June 2015, pp. 186–190.

[11] M.-H. Chen, M. Dong, and B. Liang, "Joint offloading decision and resource allocation for mobile cloud with computing access point," in *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, Mar. 2016, pp. 3516–3520.

[12] T. Q. Dinh, J. Tang, Q. D. La, and Q. S. Quek, "Offloading in mobile edge computing: Task allocation and computational frequency scaling," *IEEE Trans. Commun.*, vol. 65, no. 8, pp. 4798–4810, Aug. 2017.

[13] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.