Distributed Optimization in Fog Radio Access Networks — Channel Estimation and Multi-user Detection

Qi He[†§], Qi Zhang[*], Tony Q. S. Quek[§], Zhi Chen[†] and Shaoqian Li[†]
[†] National Key Laboratory on Communications, University of Electronic Science and Technology of China
[*] The Jiangsu Key Laboratory of Wireless Communications, Nanjing University of Posts and Telecommunications
[§] Information Systems Technology and Design Pillar, Singapore University of Technology and Design
heqi.tech@hotmail.com; zhangqiqi_1212@126.com; tonyquek@sutd.edu.sg; chenzhi@uestc.edu.cn; lsq@uestc.edu.cn

***Abstract*—In this paper, we consider the channel estimation and multi-user detection problems in fog radio access networks (F-RANs). Based on block coordinate descent algorithm, we propose two methods to solve a mixed $\ell_{2,1}$-regularization functional which exploits both the sparsity of user activities and the spatial sparsity of user signals in F-RAN. Both of our methods split the computation and corresponding data into multiple units of a cluster and solve the problem in a distributed manner. Hence they can be deployed flexibly at the distributed logical edges as well as the cloud baseband unit pool in F-RAN. The differences between the two methods are that the first one operates in a serial manner and is guaranteed to converge, while the second one works in parallel and under empirical guidance. Deployment details are also provided. Numerical results demonstrate the effectiveness of the proposed methods.**

***Index Terms*—Fog radio access network (F-RAN), compressed sensing, channel estimation, multi-user detection**

I. Introduction

Fog computing is a term for an alternative to cloud computing that deploys substantial amounts of computation and storage capabilities at the edge of network [1], [2]. Different from the cloud radio access network (C-RAN) which centralizes all the collaboration radio signal processing (CRSP) at cloud baseband unit (BBU) pool, fog radio access network (F-RAN) pushes part of CRSP functions to the distributed logical extremes of a network, such as remote radio heads (RRHs). The F-RAN architecture alleviates the burden on fronthaul links and BBU pool, and supports flexible CRSP deployment which has potential to make services more realtime and the network more adaptive to dynamic traffic. Compared to C-RAN, part of RRHs in F-RAN are upgraded to fog access points (F-APs) by adding storage and computing power, as shown in Fig. 1. RRHs are connected to adjacent F-APs and also connected to BBU pool directly or indirectly by fronthaul links. Each F-AP is connected to other F-APs through high-bandwidth and low-latency links.

The identification of user activities is vital for resource allocation in F-RAN and the acquisition of CSI is critical for optimal precoder design, energy-efficient resource allocation, and interference management [3]–[6]. The estimation of user activities and CSI which lead to the channel estimation (CE) and multi-user detection (MUD) problems have been investigated extensively in traditional cellular network [7], [8]. Conventionally, orthogonal identification pilots are assigned to different users within the same cell to eliminate the intra-cell interference, such that the pilot length needs to scale with the number of users multiply by the number of antennas per user. However, in F-RAN, since the CE and MUD problems are no longer restricted to local BS processing, the mutually orthogonal pilots will scale with the number of users in the whole network. Therefore, more efficient processing scheme is required for F-RAN.

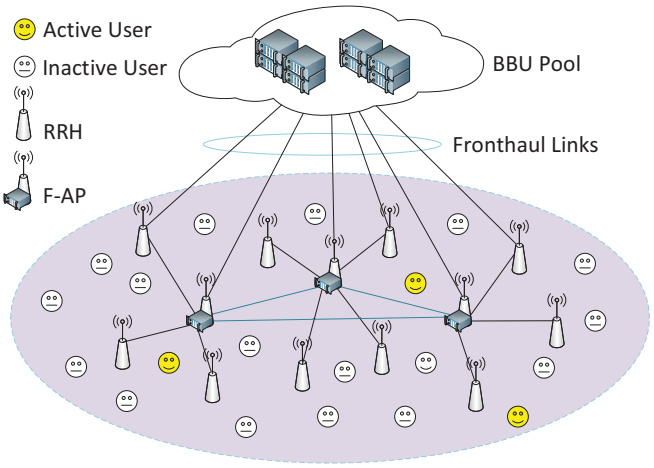

Fig. 1: Illustration of the F-RAN system.

Since most of users are silent in each time slot, in many recent works, MUD and CE are formulated as a sparse signal recovery problem that can be solved by compressed sensing (CS) technology [9]–[12]. Therefore, rigorously orthogonal characteristic between pilots belonging to different users is no longer required, and the pilot length can be reduced considerably. The CE and MUD problems in the similar C-RAN system are discussed in [13]–[15], where the received pilot data in RRHs are transmitted through fronthaul links to the BBU pool and jointly processed to meet the demand of the whole network. In F-RAN, however, the massive computing resources existing at network edges provide another option that we can solve MUD and CE problems at distributed F-APs rather than the BBU pool. Solving MUD and CE problems by F-APs and local links

This work was supported in part by the National Natural Science Foundation of China under Grant 61631004.

can alleviate the burden on fronthaul links and BBU pool, which is one of the main bottlenecks in C-RAN [14]. Just like the general CRSP in F-RAN, one flexible deployment strategy for MUD and CE is that the local solving of MUD and CE are triggered with assistance of F-APs and local links when traffic load on fronthaul links or computation load on BBU pool become high, while the centralized MUD and CE are triggered with assistance of BBU pool and fronthaul links when not executed efficiently by F-APs and local links.

Since architectures of C-RAN and F-RAN are similar, the problem formulation of MUD and CE in C-RAN and F-RAN is almost the same, and hence the methods proposed in [13]–[15] for C-RAN are still applicable for the centralized process scenario at BBU pool in F-RAN. Except a variant of methods proposed in [13], all the methods in [13]–[15] operate in a centralized fashion, and therefore can not apply to the distributed computation setting at the edge of F-RAN. The parallel method in [13] relies on the exact prior knowledge of path loss between each RRH and each user, which can be computationally intensive to obtain, especially when the F-RAN is large [16].

In this paper, we propose two new methods to solve MUD and CE problems in F-RAN which do not require any prior information of channel parameters. Both methods can apply to the distributed computation setting composed of F-APs and local links, as well as the centralized computation setting at BBU pool. Note that, in both methods, user pilots are stored uniquely and distributedly at F-APs, and hence they work well with dynamic changes of users in the network. Specifically, when a new user enters the F-RAN, it is assigned a randomly-generated pilot which is non-orthogonal with existing pilots. This pilot can be generated and then stored locally in one single F-AP and the storages in other F-APs do not need to be updated. On the other hand, when one user leaves the F-RAN, all the needed operation is deleting its pilot at the F-AP where it is stored.

The main contributions of this paper are as follows:

- A novel method based on block coordinate descent (BCD) algorithm is proposed to solve MUD and CE problems in F-RAN, which operates in a serial manner and is guaranteed to converge. A practical assumption is made which dramatically simplifies the computation and expression. As a baseline work, a BCD method is also proposed to solve the standard sparse group lasso criterion.
- To accelerate the optimization procedure, extended from BCD, we propose another method named hybrid BCD (HBCD) which works in a parallel fashion. Compared to BCD, HBCD has lower complexity but works under empirical guidance. Note that we introduce two levels of parallelism in HBCD: one is across distributed F-APs, and the other is among cores within each F-AP. Deployment details for HBCD are also provided.
- Simulations are conducted to verify the effectiveness of the proposed methods. It is shown that, even without any prior information on channels, BCD provides the same performance as state-of-the-art method, while HBCD works a little worse but has rather low computational complexity.

Notations: We use uppercase (lowercase) boldface letters to denote matrices (column vectors). iff denotes "if and only if". The operators $(\cdot)^T$, $(\cdot)^H$, $\|\cdot\|_F$ stand for transpose, conjugate transpose and Frobenius norm, respectively. The operator $(a)_+$ denotes $max\{a, 0\}$. $\mathbf{I}_N$ denotes an $N \times N$ identity matrix. $\mathbf{1}_{K\times G}$ stands for a $K \times G$ matrix where each element equals to 1. $\otimes$ denotes the Kronecker product. vec($\mathbf{A}$) denotes the vectorization of $\mathbf{A}$ formed by stacking its columns into a single column vector. $\mathbf{A}_{[i,j]}$ stands for the $\{i, j\}$th element in $\mathbf{A}$, and $\mathbf{A}_{i,j}$ denotes the $\{i, j\}$th submatrix in $\mathbf{A}$.

II. System Model and Problem Formulation

We call F-APs and RRHs together as access points (APs) in this paper. Consider an uplink F-RAN system with G APs and K users. APs consist of C F-APs and $(G - C)$ RRHs. There are N antennas in each user device, and M antennas in each AP. The kth user is assigned with a pilot matrix $\mathbf{P}_k \in \mathbb{C}^{N\times L}$, where L denotes the length of training pilots. As only a small part of users are active, we denote the set of active users by $\mathcal{A} \subsetneqq \{1, ..., K\}$, and define indicator function $\Delta_k = 1$ if the kth user is active, otherwise $\Delta_k = 0$.

It is assumed that transmission and reception of training pilots are synchronized, which can be achieved by broadcasting periodical beacon signals in network, receiving GPS signals or other technologies. Then the received training pilot data in the gth AP can be described as

$$\mathbf{R}_g = \sum_{k\in\mathcal{A}} \mathbf{H}_{g,k}\mathbf{P}_k + \bar{\mathbf{N}}_g = \sum_{k=1}^{K} \mathbf{H}_{g,k}\Delta_k\mathbf{P}_k + \bar{\mathbf{N}}_g \quad (1)$$

where $\mathbf{H}_{g,k} \in \mathbb{C}^{M\times N}$ denotes the quasi-static channel from the kth user to the gth AP, and $\bar{\mathbf{N}}_g \in \mathbb{C}^{M\times L}$ denotes additive noise in the gth AP.

By concatenating received pilot data in all the G APs, the total received data in system can be described as

$$\mathbf{R} = \mathbf{H}\mathbf{\Lambda}\mathbf{P} + \bar{\mathbf{N}} \quad (2)$$

where $\mathbf{R} = [\mathbf{R}_1^T, \ ... \ , \mathbf{R}_G^T]^T$, $\mathbf{\Lambda} = \text{diag}[\Delta_1, \ ... \ , \Delta_K] \otimes \mathbf{I}_N$, $\mathbf{P} = [\mathbf{P}_1^T, \ ... \ , \mathbf{P}_K^T]^T$, $\bar{\mathbf{N}} = [\bar{\mathbf{N}}_1^T, \ ... \ , \bar{\mathbf{N}}_G^T]^T$, and

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{1,1} & \cdots & \mathbf{H}_{1,K} \\ \cdots & \cdots & \cdots \\ \mathbf{H}_{G,1} & \cdots & \mathbf{H}_{G,K} \end{bmatrix} \quad (3)$$

With the knowledge of training pilot matrix $\mathbf{P}$ as well as received data $\mathbf{R}$, the MUD problem is to detect $\mathbf{\Lambda}$ in (2), and the CE problem is to estimate matrices $[\mathbf{H}_{1,k}^T; ...; \mathbf{H}_{G,k}^T]^T$, $\forall k \in \mathcal{A}$. To accomplish the two objectives, we transform (2) into an under-determined measurement system as below, and aim to estimate $\mathbf{\Lambda}\mathbf{H}^H$ as a whole:

$$\mathbf{R}^H = \mathbf{P}^H\mathbf{\Lambda}\mathbf{H}^H + \bar{\mathbf{N}}^H \quad (4)$$

where $\mathbf{P}^H \in \mathbb{C}^{L\times KN}$ is a matrix with the assumption that the pilot length L is less than the number of users K times the antenna number per user N.

To regularize the expression, the system model (4) is rewritten as below,

$$\mathbf{B} = \mathbf{A}\mathbf{X} + \mathbf{N} \tag{5}$$

where $\mathbf{B} = \mathbf{R}^H \in \mathbb{C}^{L\times GM}$, $\mathbf{A} = \mathbf{P}^H \in \mathbb{C}^{L\times KN}$, $\mathbf{X} = \mathbf{\Lambda}\mathbf{H}^H \in \mathbb{C}^{KN\times GM}$, and $\mathbf{N} = \bar{\mathbf{N}}^H$.

We use $\mathbf{X}_i \in \mathbb{C}^{N\times GM}$ to denote the ith row submatrix $[\Delta_i\mathbf{H}_{1,i}^H, \ \dots\ , \Delta_i\mathbf{H}_{G,i}^H]$ in $\mathbf{\Lambda}\mathbf{H}^H$, which is also referred to as the ith "row chunk" in $\mathbf{X}$; and use $\mathbf{X}_{i,j} \in \mathbb{C}^{N\times M}$ to denote the $\{i,j\}$th element chunk $\Delta_i\mathbf{H}_{j,i}^H$ in $\mathbf{\Lambda}\mathbf{H}^H$, which is also referred to as the $\{i,j\}$th "element chunk" in $\mathbf{X}$. The structure of $\mathbf{X}$ is shown in Fig. 2. We further define $\mathbf{A}_i = \mathbf{P}_i^H \in \mathbb{C}^{L\times N}$ in (4), then $\mathbf{A} = [\mathbf{A}_1, \ \dots\ , \mathbf{A}_K]$.

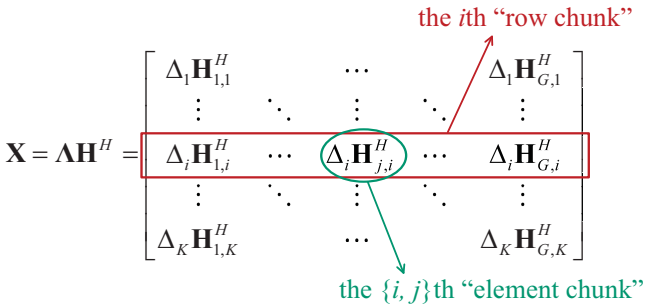

Fig. 2: The row chunks and element chunks in $\mathbf{X}$

The sparsity of the ith row chunk $\mathbf{X}_i$ is determined by Δ_i, that is whether the ith user is active. Therefore most of row chunks in $\mathbf{X}$ equal to zero, and $\mathbf{X}$ has row-chunk sparsity structure. In a F-RAN, numerous APs are distributed on large areas, and far away APs have negligible effect on a specific active user. This results in that if the ith row chunk $\mathbf{X}_i$ is nonzero, $\mathbf{X}_{i,j} = \mathbf{H}_{j,i}^H$ approximates to zero for most of $j \in \{1, ..., M\}$. Hence the non-zero row chunk $\mathbf{X}_i$ has element-chunk sparsity structure.

To exploit the two type of sparsities in $\mathbf{X}$ simultaneously, we propose a weighted $\ell_{2,1}$-regularization minimization functional as follows:

$$\min_{\mathbf{X}} \ \alpha_1 \sum_{i=1}^{K} \boldsymbol{w}_i \left\|\mathbf{X}_i\right\|_F + \alpha_2 \sum_{i=1}^{K}\sum_{j=1}^{G} \mathbf{W}_{i,j} \left\|\mathbf{X}_{i,j}\right\|_F + \frac{1}{2}\left\|\mathbf{A}\mathbf{X} - \mathbf{B}\right\|_F^2 \tag{6}$$

where $\boldsymbol{w} \in \mathbb{R}_+^{K\times 1}$ is a weight vector in which $\boldsymbol{w}_i$ is the weight scaler of the ith row-chunk $\mathbf{X}_i$, and $\mathbf{W} \in \mathbb{R}_+^{K\times G}$ is a weight matrix in which $\mathbf{W}_{i,j}$ is the weight scaler of the $\{i,j\}$th element-chunk $\mathbf{X}_{i,j}$.

The first term in (6) exploits the row chunk sparsity, and the second term takes into account the element chunk sparsity. The re-weighting strategy can provide democratic penalization on chunks and further enhance performance [17]. Different from our previous work in [15] where an element-level re-weighting strategy is used, we adopt a chunk-level re-weighting strategy in this paper in order to improve efficiency and fit the proposed methods.

The choices of α_1 and α_2 have critical effect on the performance of (6). A theorem on the rigorous upper bounds of α_1 and α_2 is shown as below, which is a direct inference of Theorem 1 in [15].

Theorem 1. *The solution of the functional (6) is* $\hat{\mathbf{X}} = \mathbf{0}$ *if* $\alpha_1 \geq \alpha_1^* = \max_i \left\{ \left\|(\mathbf{A}_i^H\mathbf{B})\right\|_F / \boldsymbol{w}_i \right\}$ *or* $\alpha_2 \geq \alpha_2^* = \max_{i,j} \left\{ \left\|(\mathbf{A}_i^H\mathbf{B}_j)\right\|_F / \mathbf{W}_{i,j} \right\}$.

In addition, we consider a special case where each RRH and user is equipped with only single antenna, i.e., $M = N = 1$. Then, the element chunk $\mathbf{X}_{i,j}$ in $\mathbf{X}$ reduces to a complex number. By defining $\mathbf{b} := \mathrm{vec}(\mathbf{B}^T)$, $\mathbf{x} := \mathrm{vec}(\mathbf{X}^T)$, $\mathbf{n} := \mathrm{vec}(\mathbf{N}^T)$ and $\bar{\mathbf{A}} := \mathbf{A} \otimes \mathbf{I}_G$, we obtain the simplified system model from (5) as follows:

$$\mathbf{b} = \bar{\mathbf{A}}\mathbf{x} + \mathbf{n} = \sum_{k=1}^{K} \bar{\mathbf{A}}^{(k)}\mathbf{x}^{(k)} + \mathbf{n}, \tag{7}$$

where $\mathbf{x}^{(k)} \in \mathbb{C}^{G\times 1}$ denotes the transposition of the kth row in $\mathbf{X}$, and $\bar{\mathbf{A}}^{(k)} \in \mathbb{C}^{LG\times G}$ denotes the column submatrix of $\bar{\mathbf{A}}$ corresponding to $\mathbf{x}^{(k)}$. We also address $\mathbf{x}^{(k)}$ as the kth group in $\mathbf{x}$. It can be observed that the vector $\mathbf{x}$ to be estimated has two levels of sparsity: group-wise sparsity and element-wise sparsity. By further removing the effect of weight vector $\boldsymbol{w}$ and matrix $\mathbf{W}$, the proposed functional in (6) is reduced to

$$\min_{\mathbf{x}} \ \alpha_1 \sum_{k=1}^{K} \left\|\mathbf{x}^{(k)}\right\|_F + \alpha_2 \left\|\mathbf{x}\right\|_1 + \frac{1}{2}\left\|\bar{\mathbf{A}}\mathbf{x} - \mathbf{b}\right\|_F^2, \tag{8}$$

where $\|\mathbf{x}\|_1$ denotes the sum of magnitudes of each complex element in $\mathbf{x}$.

The model in (7) belongs to single measurement vector (SMV) model which aims to recover an unknown sparse signal vector from a single measurement vector, while our targeted problem in (5) lies in the context of multiple measurement vectors (MMV) scenario. If we eliminate the first term in the simplified functional (8) by setting $\alpha_1 = 0$, only the sparse relationship between each active user and each effective RRH is exploited, and (8) reduces to the standard least-absolute shrinkage and selection operator (lasso) problem [18]. If we eliminate the second term in (8) by setting $\alpha_2 = 0$, only the sparsity of user activities is exploited, and (8) reduces to the group lasso problem [19]. The functional in (8) exploits element sparsity and group sparsity together, and is known as sparse group lasso criterion [20].

III. Block Coordinate Descent Method

In this section, a new method is proposed based on block coordinate descent (BCD) algorithm for the distributed memory and computation setting in F-RAN. The BCD algorithm is one class of iterative algorithms where coordinates are

partitioned into blocks, and the objective is optimized cyclically over each block-coordinate hyperplane while remaining unchanged at all the other block-coordinate hyperplanes [21]. In BCD, each optimization subproblem is a low-dimensional minimization problem that can be solved much easier than the full problem.

As a baseline work, we first consider the standard S-MV sparse group lasso criterion in (8) and extend it to solve the proposed functional later [20], [22]. Considering the simplified SMV system model in (7), inspired by [23] which handles the general group lasso problem, we assume that $\bar{\mathbf{A}}^{(k)}$ is an orthogonal matrix for each user k, that is $\bar{\mathbf{A}}^{(k)H}\bar{\mathbf{A}}^{(k)} = \mathbf{I}_G,\ \forall k = 1, ..., K$. It can be achieved by normalizing the pilot of each user antenna, i.e. each column in $\mathbf{A}$, as unit complex vector. Aiming at solving the functional in (8), we propose a BCD method as shown in Appendix B, which is based on a chunk-wise shrinkage operation defined in Appendix A. In each iteration, the estimation of $\mathbf{x}^{(k)}$ is obtained by (19) sequentially for $k = 1, ..., K$. The solution of (8) can be obtained by iterating until convergence.

Next, we consider the targeted functional in (6). It is practical to consider that the pilot length L is larger than the user-antenna number N, and we have the following assumption.

Assumption 1. *Within each user device in F-RAN, the pilot sequences transmitted by antennas are mutually orthogonal, that is* $\mathbf{A}_i^H\mathbf{A}_i = \mathbf{P}_i\mathbf{P}_i^H = \mathbf{I}_N,\ \forall i = 1,\ \dots\ , K$ *in (5).*

With Assumption 1, the BCD method to solve the functional in (6) is depicted in Appendix C. The convergence of BCD method is stated in the following, which is a direct extension of the results in [24].

Proposition 1. *For any* $\alpha_1, \alpha_2 > 0$*, and fixed* $\boldsymbol{w}$, $\mathbf{W}$*, the estimation of the* i*th row chunk* $\mathbf{X}_i$ *is obtained by (28) sequentially for* $i = 1, ..., K$ *in each iteration. Then, the iterations of* $\mathbf{X}$ *are guaranteed to converge to the global minimizer of (6).*

The BCD method loops for several times, in each of which the solution of (6) is obtained with fixed weight vector $\boldsymbol{w}$ and matrix $\mathbf{W}$. As an empirical law, each element in $\boldsymbol{w}$ and $\mathbf{W}$ are set inversely to the previous estimation of respective chunk norms, shown in (9). In summary, the BCD method is described in Table I.

$$\boldsymbol{w}_i = \frac{1}{\|\mathbf{X}_i\|_F + \epsilon}, \quad \mathbf{W}_{i,j} = \frac{1}{\|\mathbf{X}_{i,j}\|_F + \epsilon} \tag{9}$$

For deployment of BCD method in F-RAN, we consider a setup that consists of C distributed F-APs with a ring connecting topology. An example with three F-APs is shown in Fig. 3 with definition $\mathbf{R} := \mathbf{B} - \mathbf{A}\hat{\mathbf{X}}$, where $\hat{\mathbf{X}}$ denotes the current estimation of $\mathbf{X}$. To update the estimation of $\mathbf{X}_i$ as described at **Steps 5** in Table I, with the received $\mathbf{R}$, the variable $\mathbf{R}_{-i}$ in (28) is obtained by $\mathbf{R}_{-i} = \mathbf{R} + \mathbf{A}_i\hat{\mathbf{X}}_i$ where $\hat{\mathbf{X}}_i$ denotes the current estimation of $\mathbf{X}_i$. It can be seen that, except $\mathbf{R}$, the updating of $\hat{\mathbf{X}}_i$ needs $\mathbf{A}_i, \mathbf{W}_i, \boldsymbol{w}_i$ and the current estimation of $\mathbf{X}_i$, all of which can be stored locally.

TABLE I: BCD Method

Algorithm Block Coordinate Descent Method

1: Initialize weight vector $\boldsymbol{w}$ and weight matrix $\mathbf{W}$.
2: Initialize parameters α_1, α_2, ϵ and the max loop number $MaxCount$. Set the loop counter $count \leftarrow 1$.
3: **while** $count \leq MaxCount$ **do**
4: **while** not convergent and stopping criterion not met **do**
5: Get the estimation of row chunk $\mathbf{X}_i$ by (28) sequentially for $i = 1,\ \dots\ , K$.
6: **end while**
7: Update $\boldsymbol{w}$ and $\mathbf{W}$ with (9).
8: $count \leftarrow count + 1$.
9: **end while**

In the deployment, the K row-chunk coordinates are sequentially partitioned into C blocks, $\mathcal{P}_1$, ... , $\mathcal{P}_C$, with cardinality $\sum_c |\mathcal{P}_c| = K$. The computation task of the cth block $\mathcal{P}_c$ is assigned to the cth F-AP. Within the cth F-AP, row chunks are computed sequentially, and the finally updated $\mathbf{R}$ is passed to the $\{c+1\}$th F-AP. We further define the column submatrix in pilot matrix $\mathbf{A}$ corresponding to $\mathcal{P}_c$ as $\mathbf{A}_{\mathcal{P}c}$. $\mathbf{W}_{\mathcal{P}c}, \boldsymbol{w}_{\mathcal{P}c}$ and $\mathbf{X}_{\mathcal{P}c}$ are defined similarly. Then $\mathbf{A}_{\mathcal{P}c}, \mathbf{W}_{\mathcal{P}c}, \boldsymbol{w}_{\mathcal{P}c}$ and the estimation result $\hat{\mathbf{X}}_{\mathcal{P}c}$ are only stored on the cth F-AP. Data and computation in BCD method are thus distributed to C F-APs. Other deployment details of BCD are omitted in this paper. Besides, it can be seen that the computational complexity of BCD per iterate is approximately $\mathcal{O}\big(KGMN\big)$.

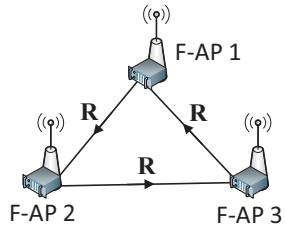

Fig. 3: Example of BCD method deployment

IV. Hybrid Block Coordinate Descent Method

The BCD method solves our problems in a serial fashion, and it is intuitive to extend it to a parallel method in order to further accelerate the solving procedure. Randomized coordinate descent methods, where multiple coordinates are updated parallelly in each iteration, were proposed in several recent works [25], [26] to utilize the multi-core and shared-memory setup in one single computer. While in [27], hybrid coordinate descent method was proposed to fit the modern multiple distributed computer scenario.

With the distributed computing setting at the edge of F-RAN, we propose a hybrid block coordinate descent (H-BCD) method to solve the functional (6) in parallel. The

term "hybrid" refers to parallelism at two levels: (i) across distributed F-APs and (ii) among independent computation cores within each F-AP. Assume there are $C+1$ F-APs and each of the first C F-APs is equipped with E cores. The K row-chunk coordinates are separated regularly or randomly into C blocks, and the cth block of coordinates is labelled as $\mathbf{X}_{\mathcal{P}_c}$. The calculation of $\hat{\mathbf{X}}_{\mathcal{P}_c}$ is deployed to the cth F-AP, where $\mathbf{A}_{\mathcal{P}_c}, \mathbf{W}_{\mathcal{P}_c}, \boldsymbol{w}_{\mathcal{P}_c}$ are stored locally. For dynamic changes of users in the network, the pilot matrix of a newly arrived user can be generated and stored in any F-AP, and hence we can assume that $|\mathcal{P}_i| \approx |\mathcal{P}_j|, \forall i \neq j$. The HBCD method is described in Table II, and an example of HBCD deployment with five F-APs is shown in Fig. 4.

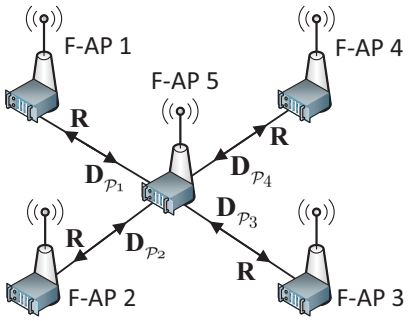

Fig. 4: Example of HBCD method deployment

We now comment on Table II. **Steps 4-8** are executed in each of the first C F-APs parallelly. In the cth F-AP, a coordinate set $\mathcal{S}_c$ of cadinality τ is picked from $\mathcal{P}_c$ uniformly at random and independently of other F-APs. Then, similarly to BCD method, $\mathbf{R}_{-i}$ for $i \in \mathcal{S}_c$ is obtained using equation $\mathbf{R}_{-i} = \mathbf{R} + \mathbf{A}_i\hat{\mathbf{X}}_i$, where $\mathbf{R}$ is received from the $\{C+1\}$th F-AP. With data $\mathbf{R}_{-i}$, the estimation of $\mathbf{X}_i$ for $i \in \mathcal{S}_c$ and the intermediate matrix $\mathbf{D}_i$ in **Steps 6** are obtained independently and parallelly on E cores. The communication data $\mathbf{D}_{\mathcal{P}_c} \in \mathbb{C}^{L\times GM}$ is given by $\mathbf{D}_{\mathcal{P}_c} = \sum_{i\in\mathcal{P}_c}\mathbf{D}_i = \mathbf{A}_{\mathcal{P}_c}\hat{\mathbf{X}}_{\mathcal{P}_c}$. The convergence is determined locally based on the comparison between the new and old estimations of $\mathbf{X}_{\mathcal{P}_c}$, and then the result as well as $\mathbf{D}_{\mathcal{P}_c}$ is passed to the $\{C+1\}$th F-AP. In **Step 10**, the $\{C+1\}$th F-AP confirms convergence if converged in all the first C F-APs or other stopping criterion met. Otherwise, in **Step 18**, $\mathbf{R}$ is calculated in the $\{C+1\}$th F-AP and passed to all the first C F-APs.

The HBCD method is inherently synchronous, and the computational complexity of HBCD is $\mathcal{O}\big(LGMN\tau/E\big)$ per iterate, which is only $\mathcal{O}\big(L\tau/(KE)\big)$ fold of the BCD method for each iterate. It can be seen from (28) that HBCD adopts the step size $\beta = 1$ as recommended in [26]. Note that subsets of $\{1, ..., K\}$ with cardinality $C\tau$ are not chosen with equal probability, and hence the analysis in [26] does not apply. The HBCD method adopts the same memory-distributed setting as in [27], which provides theoretical analysis on the condition that $\beta \geq 2\beta^*$ where $\beta^* \approx 1 + C\tau$. However, the experiments in [27] admit that poor performance is achieved when $\beta \geq 2\beta^*$ and massive speedups can be obtained by choosing step size β even hundreds of times smaller than β^*. Therefore, we fix the step size β equal to one, and set parameter τ to a small value according to the empirical results in [26], [27]. Furthermore, based on experiments of [27] and ourselves, we provide Remark 1 on the choice of τ which has great influence on the performance.

TABLE II: HBCD Method

Algorithm Hybrid Block Coordinate Descent Method

1: Initialize α_1, α_2, ϵ, $\mathbf{W}_{\mathcal{P}_c}$ and $\boldsymbol{w}_{\mathcal{P}_c}$ locally and identically in each of the C F-APs. Set $count \leftarrow 1$ and $MaxCount$ in the $\{C+1\}$th F-AP.
2: **loop**
3: **for each** F-AP $c \in \{1, ..., C\}$ **in parallel do**
4: Pick a random set of row chunks $\mathcal{S}_c \subseteq \mathcal{P}_c$ and $|\mathcal{S}_c| = \tau$.
5: **for each** row chunk $i \in \mathcal{S}_c$ **do**
6: Obtain the estimation of $\mathbf{X}_i$ by (28), and $\mathbf{D}_i := \mathbf{A}_i\mathbf{X}_i$.
7: **end for**
8: Obtain $\mathbf{D}_{\mathcal{P}_c} := \sum_{i\in\mathcal{P}_c}\mathbf{D}_i$.
9: **end for**
10: **if** convergent or stopping criterion met **then**
11: **if** $count \leq MaxCount$ **then**
12: $count \leftarrow count + 1$.
13: Update $\boldsymbol{w}_{\mathcal{P}_c}$ and $\mathbf{W}_{\mathcal{P}_c}$ with (9) locally in each F-AP $c \in \{1, ..., C\}$.
14: **else**
15: The estimation of $\mathbf{X}$ is obtained. HBCD ends.
16: **end if**
17: **else**
18: Obtain $\mathbf{R} := \mathbf{B} - \sum_{c=1}^{C}\mathbf{D}_{\mathcal{P}_c}$.
19: **end if**
20: **end loop**

Remark 1. *Large value of τ may lead to divergency in HBCD. However, empirically, larger τ in safe range not only accelerates the converging speed but also achieves higher estimation accuracy. Hence, HBCD would benefit from a line-search procedure for the selection of τ.*

V. Numerical Results

Consider a F-RAN with $G = 60$ APs and $K = 600$ users. The APs are distributed at grid points and users are uniformly and randomly distributed in a rectangular region. Rayleigh fading channel is assumed between APs and users. There are five F-APs among APs, and each of the first $C = 4$ F-APs is equipped with $E = 10$ cores. The number of antennas in each AP and user are $M = 3$ and $N = 2$, respectively. The number of active users is 60, and SNR is set at 20 dB. According to Assumption 1, we adopt orthogonalized Gaussian random pilot matrix $\mathbf{A}_i$ for each user i, which satisfies $\mathbf{A}_i^H\mathbf{A}_i = \mathbf{I}_N$. Parameters α_1 and α_2 in functional (6) are empirically set equal to a small percentage of α_1^* and α_2^* in Theorem 1, respectively, that is 1%-5%. The number of re-weighting

times $MaxCount$ is set to 2, and ϵ in (9) is 10^{-8}. The parameter τ in HBCD is set to 10.

For comparison, the method "modified Bayesian compressive sensing with clustering" ("ClusterBCS" for short) in [13] is added into the simulation. Based on the prior information of path loss between APs and users, ClusterBCS partitions APs into clusters and operate parallelly in each cluster. Here we also include another case named "ClusterBCS2" in which ClusterBCS is provided with inaccurate path-loss values. Assuming the precise path loss parameter between the kth user and the gth AP is $\gamma_{g,k}$, we provide ClusterBCS2 with $\gamma_{g,k} \cdot (1 + 0.5n)$ where n obeys the standard normal distribution. APs are partitioned into 6 clusters in ClusterBCS and ClusterBCS2. All the results are averaged over 200 runs.

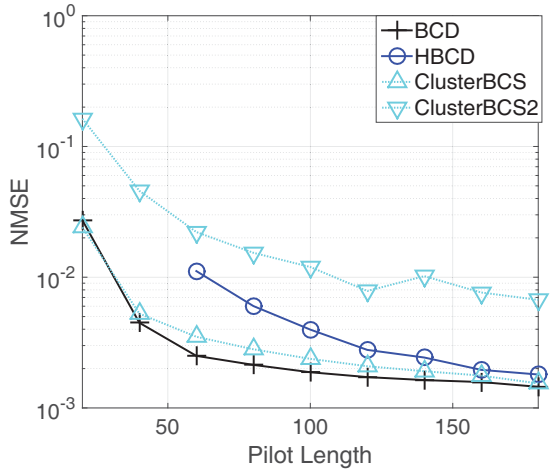

Fig. 5: Channel estimation results of respective methods

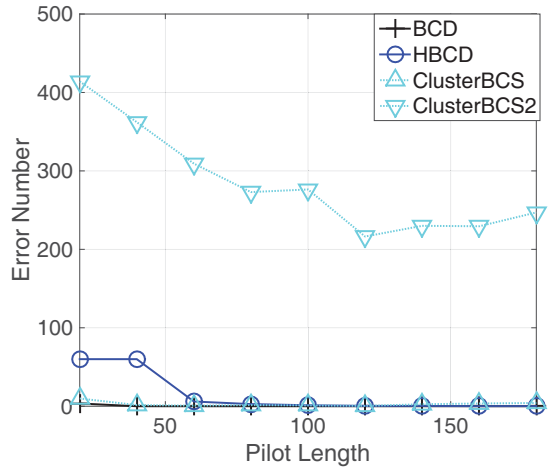

Fig. 6: Multi-user detection results of respective methods

Figure 5 shows the CE results of respective methods, where NMSE denotes the normalized mean squared error. It can be seen that BCD provides the highest estimation accuracy even without any prior information on channels. The HBCD needs longer pilot to achieve the same estimation accuracy as BCD. The ClusterBCS performs slightly worse than BCD, while ClusterBCS2 has the worst performance. In the HBCD curve, the points where pilot length equals to 20 and 40 do not exist because too short pilots cause a certain probability of diversity according to our simulations. The MUD results are shown in Fig. 6 where "Error Number" denotes the number of wrongly detected users in respective methods. It can be seen that BCD and ClusterBCS exhibit the best performances in MUD. After evaluating the performance of respective methods, we turn to the computational complexity which is shown in Fig. 7. It can be seen that HBCD has the lowest computational complexity and thus is more competitive when the problem size is larger.

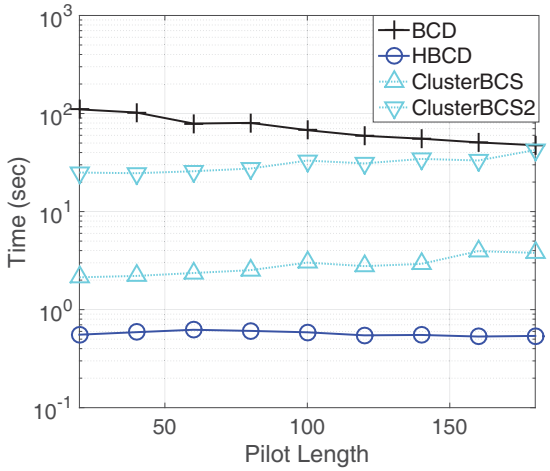

Fig. 7: Computation time of respective methods.

At last, we illustrate the influence of parameter τ on the performance of HBCD. By fixing the value of pilot length at $L = 70$, HBCD with different values of τ and BCD are executed for one time and the normalized square error (NSE) of respective methods versus iteration number are shown in Fig. 8. The "weight-updated point" denotes the point where the weight vector $\boldsymbol{w}$ and matrix $\mathbf{W}$ are updated by (9). As the number of re-weighting times $MaxCount$ is set to two, there is only one weight-updated point in each curve. The weight-updated points in HBCD-class methods are all at 600 because the maximum iteration number for each loop in Table II is set to 600. The BCD needs much less iteration number to converge than HBCD, in part because all the row chunks are updated in one iterate of BCD while only a small portion are updated in one iterate of HBCD. For HBCD, a value of τ larger than 10 leads to a certain probability of divergency according to our simulations. While a larger τ which is less than or equal to 10 achieves higher estimation accuracy and needs less iteration numbers to converge, which agrees with the discussion in Remark 1.

VI. Conclusion

To provide flexible solutions for CE and MUD problems in F-RAN, we propose two methods named BCD and HBCD,

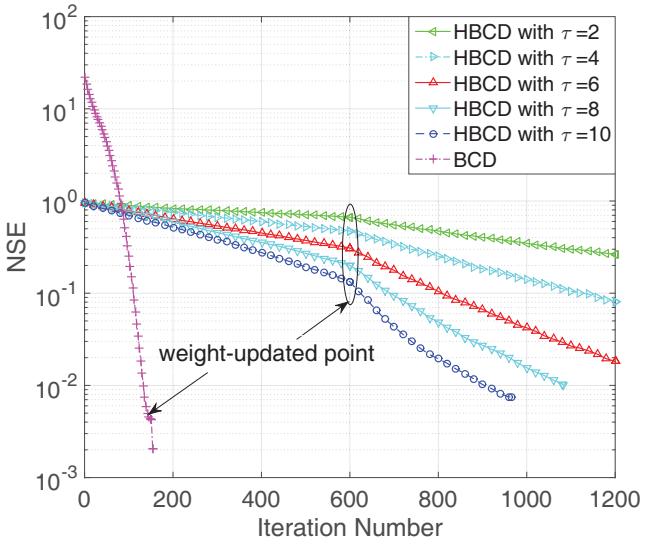

Fig. 8: Normalized square error of methods vs. iteration

respectively, both of which can operate with distributed computation and storage setting. Moreover, these two methods can both be deployed at the BBU pool as well as the edges of F-RAN composed of F-APs and local links. The BCD operates in a serial fashion and is guaranteed to converge while HBCD works in parallel and under empirical guidance. Deployment details are also provided. Simulation results show that BCD can provide the state-of-art performance even without any prior information on channels, while HBCD performs a little worse but owns the lowest computational complexity.

Appendix A
Definition of Chunk-wise Shrinkage Operation

To simplify expressions in Appendices B and C, we denote the chunk-wise shrinkage operation for complex matrix as $Shrink(\cdot)$, which is an extension of the basic one-dimensional soft thresholding method or shrinkage in, e.g., [28]. Assume that $\mathbf{Y}, \mathbf{X} \in \mathbb{C}^{KN \times GM}$ and $\mathbf{C} \in \mathbb{R}_+^{K \times G}$, and then $\mathbf{Y} = Shrink_{(N,M)}(\mathbf{X}, \mathbf{C})$ represents that (N, M) is the dimension of shrinkage operation in the objective matrix $\mathbf{X}$, and

$$\mathbf{Y}_{i,j} = \frac{\mathbf{X}_{i,j}}{\|\mathbf{X}_{i,j}\|_F}\left(\|\mathbf{X}_{i,j}\|_F - \mathbf{C}_{[i,j]}\right)_+,$$

where $\mathbf{C}_{[i,j]} \in \mathbb{R}_+$ stands for the $\{i,j\}$th element in $\mathbf{R}$, and $\mathbf{X}_{i,j}, \mathbf{Y}_{i,j} \in \mathbb{C}^{N \times M}$ denote the $\{i,j\}$th submatrix in $\mathbf{X}$ and $\mathbf{Y}$ respectively.

Appendix B
BCD Method for Functional in (7)

The functional (7) is convex in $\mathbf{x}$. As a direct consequence of Karush-Kuhn-Tucker (KKT) condition, for the kth group in the optimal solution $\hat{\mathbf{x}}$, $\hat{\mathbf{x}}^{(k)}$ should satisfy the subgradient equation as follows:

$$-\bar{\mathbf{A}}^{(k)H}(\mathbf{r}^{(-k)} - \bar{\mathbf{A}}^{(k)}\hat{\mathbf{x}}^{(k)}) + \alpha_1\mathbf{u} + \alpha_2\mathbf{v} = \mathbf{0}, \quad (10)$$

where $\mathbf{r}^{(-k)} = \mathbf{b} - \bar{\mathbf{A}}\hat{\mathbf{x}}^{(-k)}$, and $\hat{\mathbf{x}}^{(-k)} = (\hat{\mathbf{x}}^{(1)T}, ..., \hat{\mathbf{x}}^{(k-1)T}, \mathbf{0}, \hat{\mathbf{x}}^{(k+1)T}, ..., \hat{\mathbf{x}}^{(K)T})^T$. $\mathbf{u}$ and $\mathbf{v}$ are the subgradients of $\|\hat{\mathbf{x}}^{(k)}\|_F$ and $\|\hat{\mathbf{x}}^{(k)}\|_1$ with regard to $\hat{\mathbf{x}}^{(k)}$, respectively, and are given by

$$\mathbf{u} = \begin{cases} \frac{\hat{\mathbf{x}}^{(k)}}{\|\hat{\mathbf{x}}^{(k)}\|_F} & \text{iff } \hat{\mathbf{x}}^{(k)} \neq \mathbf{0}, \\ \in \{\mathbf{u} : \|\mathbf{u}\|_F \leq 1\} & \text{iff } \hat{\mathbf{x}}^{(k)} = \mathbf{0}, \end{cases} \quad (11)$$

$$\mathbf{v}_j = \begin{cases} \frac{\hat{\mathbf{x}}_j^{(k)}}{\|\hat{\mathbf{x}}_j^{(k)}\|_F} & \text{iff } \hat{\mathbf{x}}_j^{(k)} \neq \mathbf{0}, \\ \in \{\mathbf{v}_j : \|\hat{\mathbf{x}}_j^{(k)}\|_F \leq 1\} & \text{iff } \hat{\mathbf{x}}_j^{(k)} = \mathbf{0}, \end{cases} \quad (12)$$

where $\mathbf{v}_j$ is the jth element in $\mathbf{v}$, and $\hat{\mathbf{x}}_j^{(k)}$ is the jth element in $\hat{\mathbf{x}}^{(k)}$.

With the assumption $\bar{\mathbf{A}}^{(k)H}\bar{\mathbf{A}}^{(k)} = \mathbf{I}$, we get the following solution from (10) as follows:

$$-\bar{\mathbf{A}}^{(k)H}\mathbf{r}^{(-k)} + \hat{\mathbf{x}}^{(k)} + \alpha_1\mathbf{u} + \alpha_2\mathbf{v} = \mathbf{0}. \quad (13)$$

It can be seen that $\hat{\mathbf{x}}^{(k)} = \mathbf{0}$ if and only if

$$\left\|Shrink_{(1,1)}(\bar{\mathbf{A}}^{(k)T}\mathbf{r}^{(-k)}, \alpha_2)\right\|_F \leq \alpha_1. \quad (14)$$

If $\hat{\mathbf{x}}^{(k)} \neq \mathbf{0}$, then for a particular $\hat{\mathbf{x}}_j^{(k)}$,

$$-\bar{\mathbf{A}}_j^{(k)T}\mathbf{r}^{(-k)} + \hat{\mathbf{x}}_j^{(k)} + \alpha_1\frac{\hat{\mathbf{x}}_j^{(k)}}{\|\hat{\mathbf{x}}^{(k)}\|_F} + \alpha_2\mathbf{v}_j = \mathbf{0}. \quad (15)$$

And then $\hat{\mathbf{x}}_j^{(k)} = 0$ if and only if

$$\left\|\bar{\mathbf{A}}_j^{(k)T}\mathbf{r}^{(-k)}\right\|_F = \|\alpha_2\mathbf{v}_j\|_F \leq \alpha_2. \quad (16)$$

If non-zero, $\hat{\mathbf{x}}_j^{(k)}$ should satisfy

$$(1 + \frac{\alpha_1}{\|\hat{\mathbf{x}}^{(k)}\|_F})\hat{\mathbf{x}}_j^{(k)} = \bar{\mathbf{A}}_j^{(k)T}\mathbf{r}^{(-k)} - \alpha_2\frac{\hat{\mathbf{x}}_j^{(k)}}{\|\hat{\mathbf{x}}_j^{(k)}\|_F}. \quad (17)$$

By combining (16) and (17), we get

$$(1 + \frac{\alpha_1}{\|\hat{\mathbf{x}}^{(k)}\|_F})\hat{\mathbf{x}}^{(k)} = Shrink_{(1,1)}\left(\bar{\mathbf{A}}^{(k)T}\mathbf{r}^{(-k)}, \alpha_2\right). \quad (18)$$

We further combine (18) and (14), and get the estimation of $\mathbf{x}^{(k)}$ as follows:

$$\hat{\mathbf{x}}^{(k)} = \left(1 - \frac{\alpha_1}{\left\|Shrink_{(1,1)}\left(\bar{\mathbf{A}}^{(k)T}\mathbf{r}^{(-k)}, \alpha_2\right)\right\|_F}\right)_+ \cdot Shrink_{(1,1)}\left(\bar{\mathbf{A}}^{(k)T}\mathbf{r}^{(-k)}, \alpha_2\right). \quad (19)$$

Appendix C
BCD Method for Functional in (6)

The functional (6) is convex in $\mathbf{X}$. Combined with Assumption 1, the ith row-chunk block in the optimal solution $\hat{\mathbf{X}}$ should satisfy

$$-\mathbf{A}_i^T\mathbf{R}_{-i} + \hat{\mathbf{X}}_i + \alpha_1\boldsymbol{w}_i\mathbf{U} + \alpha_2\left(\text{diag}(\mathbf{W}_i) \otimes \mathbf{I}_N\right)\mathbf{V} = \mathbf{0}, \quad (20)$$

where $\mathbf{R}_{-i} = \mathbf{B} - \mathbf{A}\hat{\mathbf{X}}_{-i}$, and $\hat{\mathbf{X}}_{-i} = (\hat{\mathbf{X}}_1^T, ..., \hat{\mathbf{X}}_{i-1}^T, \mathbf{0}, \hat{\mathbf{X}}_{i+1}^T, ..., \hat{\mathbf{X}}_K^T)^T$. $\mathbf{U}$ and $\mathbf{V}$ are the

subgradients of $\left\|\hat{\mathbf{X}}_i\right\|_F$ and $\sum_{j=1}^{G}\left\|\hat{\mathbf{X}}_{i,j}\right\|_F$ with regard to $\hat{\mathbf{X}}_i$, respectively, and are given by

$$\mathbf{U}=\begin{cases}\frac{\hat{\mathbf{X}}_i}{\left\|\hat{\mathbf{X}}_i\right\|_F} & \text{iff } \hat{\mathbf{X}}_i\neq\mathbf{0}\\ \in\{\mathbf{U}:\|\mathbf{U}\|_F\leq 1\} & \text{iff } \hat{\mathbf{X}}_i=\mathbf{0}\end{cases} \tag{21}$$

$$\mathbf{V}_j=\begin{cases}\frac{\hat{\mathbf{X}}_{i,j}}{\left\|\hat{\mathbf{X}}_{i,j}\right\|_F} & \text{iff } \hat{\mathbf{X}}_{i,j}\neq\mathbf{0}\\ \in\{\mathbf{V}_j:\|\mathbf{V}_j\|_F\leq 1\} & \text{iff } \hat{\mathbf{X}}_{i,j}=\mathbf{0}\end{cases} \tag{22}$$

where $\mathbf{V}_j\in\mathbb{C}^{M\times N}$ is the jth element chunk in $\mathbf{V}$.

It can be seen that $\hat{\mathbf{X}}_i=\mathbf{0}$ if and only if

$$\left\|Shrink_{(N,M)}\left(\mathbf{A}_i^T\mathbf{R}_{-i},\alpha_2\mathbf{W}_i\right)\right\|_F\leq\alpha_1\boldsymbol{w}_i \tag{23}$$

If $\hat{\mathbf{X}}^{(i)}\neq\mathbf{0}$, then for the $\{i,j\}$th element chunk $\hat{\mathbf{X}}_{i,j}$, we have

$$-\mathbf{A}_i^T(\mathbf{R}_{-i})_j+\hat{\mathbf{X}}_{i,j}+\alpha_1\boldsymbol{w}_i\frac{\hat{\mathbf{X}}_{i,j}}{\left\|\hat{\mathbf{X}}_i\right\|_F}+\alpha_2\mathbf{W}_{i,j}\mathbf{V}_j=\mathbf{0}, \tag{24}$$

where $(\mathbf{R}_{-i})_j\in\mathbb{C}^{L\times M}$ is the jth element chunk in $\mathbf{R}_{-i}$. Then, we have $\hat{\mathbf{X}}_{i,j}=\mathbf{0}$ if and only if

$$\left\|\mathbf{A}_i^T(\mathbf{R}_{-i})_j\right\|_F\leq\alpha_2\mathbf{W}_{i,j}, \tag{25}$$

and $\hat{\mathbf{X}}_{i,j}\neq 0$ if and only if

$$\mathbf{A}_i^T(\mathbf{R}_{-i})_j-\alpha_2\mathbf{W}_{i,j}\frac{\mathbf{A}_i^T(\mathbf{R}_{-i})_j}{\left\|\mathbf{A}_i^T(\mathbf{R}_{-i})_j\right\|_F}=(\frac{\alpha_1\boldsymbol{w}_i}{\left\|\hat{\mathbf{X}}_i\right\|_F}+1)\hat{\mathbf{X}}_{i,j}. \tag{26}$$

By combining (25) and (26), we obtain

$$(\frac{\alpha_1\boldsymbol{w}_i}{\left\|\hat{\mathbf{X}}_i\right\|_F}+1)\hat{\mathbf{X}}_{i,j}=max\{1-\frac{\alpha_2\mathbf{W}_{i,j}}{\left\|\mathbf{A}_i^T(\mathbf{R}_{-i})_j\right\|_F},0\}\mathbf{A}_i^T(\mathbf{R}_{-i})_j. \tag{27}$$

We further combine (27) and (23), and get the estimation of $\mathbf{X}_i$ as follows:

$$\hat{\mathbf{X}}_i=\big(1-\frac{\alpha_1\boldsymbol{w}_i}{\left\|Shrink_{(N,M)}(\mathbf{A}_i^T\mathbf{R}_{-i},\alpha_2\mathbf{W}_i)\right\|_2}\big)_+\cdot Shrink_{(N,M)}(\mathbf{A}_i^T\mathbf{R}_{-i},\alpha_2\mathbf{W}_i). \tag{28}$$

REFERENCES

[1] 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, 2016.

[2] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proc. the first edition of the MCC workshop on Mobile cloud computing*. ACM, 2012, pp. 13–16.

[3] M. Peng, C. Wang, V. K. 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.

[4] 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.

[5] J. Li, J. Wu, M. Peng, and P. Zhang, "Queue-aware energy-efficient joint remote radio head activation and beamforming in cloud radio access networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 6, pp. 3880–3894, Jun. 2016.

[6] 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.

[7] M. Biguesh and A. B. Gershman, "Training-based MIMO channel estimation: a study of estimator tradeoffs and optimal training signals," *IEEE Trans. Signal Process.*, vol. 54, no. 3, pp. 884–893, 2006.

[8] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *IEEE Trans. Wireless Commun.*, vol. 9, no. 11, pp. 3590–3600, 2010.

[9] H. Zhu and G. B. Giannakis, "Exploiting sparse user activity in multiuser detection," *IEEE Trans. on Commun.*, vol. 59, no. 2, pp. 454–465, 2011.

[10] B. Shim and B. Song, "Multiuser detection via compressive sensing," *IEEE Commun. Letters*, vol. 16, no. 7, pp. 972–974, 2012.

[11] X. Li, A. Rueetschi, A. Scaglione, and Y. C. Eldar, "Compressive link acquisition in multiuser communications," *IEEE Trans. Signal Process.*, vol. 61, no. 12, pp. 3229–3245, 2013.

[12] Y. Jin, Y. H. Kim, and B. D. Rao, "Limits on support recovery of sparse signals via multiple-access communication techniques," *IEEE Trans. Inf. Theory*, vol. 57, no. 12, pp. 7877–7892, 2011.

[13] X. Xu, X. Rao, and V. K. Lau, "Active user detection and channel estimation in uplink CRAN systems," in *Proc. IEEE Int. Conf. Commun. (ICC)*, London, UK, June 2015, pp. 2727–2732.

[14] Z. Utkovski, O. Simeone, T. Dimitrova, and P. Popovski, "Random access in C-RAN for user activity detection with limited-capacity fronthaul," *IEEE Signal Process. Letters*, vol. 24, no. 1, pp. 17–21, 2017.

[15] Q. He, T. Q. S. Quek, Z. Chen, and S. Li, "Compressive channel estimation and multi-user detection in C-RAN," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Paris, France, May 2017.

[16] D. Har, H. H. Xia, and H. L. Bertoni, "Path-loss prediction model for microcells," *IEEE Trans. on Veh. Tech.*, vol. 48, no. 5, pp. 1453–1462, 1999.

[17] E. J. Candes, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *J. Fourier anal. Appl.*, vol. 14, no. 5-6, pp. 877–905, 2008.

[18] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[19] L. Meier, S. Van De Geer, and P. Bühlmann, "The group lasso for logistic regression," *J. Royal Statistical Society. Series B*, vol. 70, no. 1, pp. 53–71, 2008.

[20] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani, "A sparse-group lasso," *J. Comput. and Graphical Statistics*, vol. 22, no. 2, pp. 231–245, 2013.

[21] S. J. Wright, "Coordinate descent algorithms," *Math. Program.*, vol. 151, pp. 3–34, 2015.

[22] R. Foygel and M. Drton, "Exact block-wise optimization in group lasso and sparse group lasso for linear regression," *arXiv:1010.3320*, 2010.

[23] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. Royal Statistical Society. Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[24] P. Tseng, "Convergence of a block coordinate descent method for nondifferentiable minimization," *J. Optimization Theory Appl.*, vol. 109, no. 3, pp. 475–494, 2001.

[25] P. Richtárik and M. Takáč, "Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function," *Mathematical Programming*, vol. 144, no. 1-2, pp. 1–38, 2014.

[26] J. K. Bradley, A. Kyrola, D. Bickson, and C. Guestrin, "Parallel coordinate descent for L1-regularized loss minimization," *arXiv preprint arXiv:1105.5379*, 2011.

[27] P. Richtárik and M. Takác, "Distributed coordinate descent method for learning with big data," *J. Mach. Learn. Res.*, vol. 17, no. 75, pp. 1–25, 2016.

[28] N. Parikh, S. P. Boyd *et al.*, "Proximal algorithms," *Found. Trends Optimization*, vol. 1, no. 3, pp. 127–239, 2014.