IEEE Access
The journal for rapid open access publishing

Received July 13, 2014, accepted July 29, 2014, date of publication August 29, 2014, date of current version September 10, 2014..

Digital Object Identifier 10.1109/ACCESS.2014.2353297

User Grouping for Massive MIMO in FDD Systems: New Design Methods and Analysis

YI XU[1], (Student Member, IEEE), GUOSEN YUE[2], (Senior Member, IEEE), AND SHIWEN MAO[1], (Senior Member, IEEE)

[1]Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849-5201, USA
[2]Broadcom Corporation, Matawan, NJ 07747, USA

Corresponding author: S. Mao (smao@ieee.org)

This work was supported in part by the U.S. National Science Foundation under Grant CNS-1247955 and Grant CNS-1320664 and in part by the Broadband Wireless Access and Applications Center, Auburn University, Auburn, AL, USA.

ABSTRACT The massive multiple-input multiple-output (MIMO) system has drawn increasing attention recently as it is expected to boost the system throughput and result in lower costs. Previous studies mainly focus on time division duplexing (TDD) systems, which are more amenable to practical implementations due to channel reciprocity. However, there are many frequency division duplexing (FDD) systems deployed worldwide. Consequently, it is of great importance to investigate the design and performance of FDD massive MIMO systems. To reduce the overhead of channel estimation in FDD systems, a two-stage precoding scheme was recently proposed to decompose the precoding procedure into intergroup precoding and intragroup precoding. The problem of user grouping and scheduling thus arises. In this paper, we first propose three novel similarity measures for user grouping based on weighted likelihood, subspace projection, and Fubini–Study, respectively, as well as two novel clustering methods, including hierarchical and K-medoids clustering. We then propose a dynamic user scheduling scheme to further enhance the system throughput once the user groups are formed. The load balancing problem is considered when few users are active and solved with an effective algorithm. The efficacy of the proposed schemes are validated with theoretical analysis and simulations.

INDEX TERMS Massive multiple-input multiple-output (MIMO), frequency division duplexing (FDD), precoding, user grouping, load balancing.

I. INTRODUCTION

Last decades have witnessed ever-increasing demand for higher data rates in wireless networks. To cater for this demand, many advanced physical layer techniques have been developed, e.g., multiple input multiple output (MIMO) with orthogonal frequency division multiplexing (OFDM). However, with linear throughput improvement but the exponential growth on the data traffic, the gap between the demand and supply has been increasingly widened. To solve the problem, the next technology we could resort to is massive MIMO (a.k.a. large-scale MIMO, full-dimension MIMO, or hyper MIMO), which significantly increases the system capacity by employing a large number of antennas at the base station. As an emerging and promising technology, large-scale MIMO also enjoys many advantages such as low-power, robust transmissions, simplified transceiver design, and simplified multiple-access layer [1], [2], in addition to enhanced capacity. Recently, lab demo systems have demonstrated the benefits of massive MIMO [3], [4].

In general, the more transmit antennas, the more degrees of freedom a massive MIMO system can provide, resulting in higher reliability or larger throughput. It is expected that massive MIMO will tremendously boost up the system throughput by simultaneously serving many users. However, due to the difficulties of acquiring channel state information at the transmitter side (CSIT), it is challenging to simultaneously support a large number of users [2]. Most of the existing works on massive MIMO systems consider the time-division-duplexing (TDD) mode [5]–[7], within which by exploiting channel reciprocity, the downlink channel can be estimated through uplink training. Unfortunately, there is no such privilege in frequency-division-duplexing (FDD) systems, where pilot based channel estimation and uplink channel feedback are required. Such mechanisms usually consume considerable spectrum and power resources.

According to [8], there are much more FDD ($\geq$ 300) than TDD ($\leq$ 40) LTE licenses worldwide. It is therefore of great importance to investigate the massive MIMO design for FDD systems. To reduce pilot resources and the channel state information (CSI) feedback in FDD systems, a two-stage precoding scheme has been proposed in [9] recently. Firstly, the users in service are divided into groups, while each group of users have similar second-order channel statistics (i.e., transmit correlation). The same pre-beamforming, or the first-stage precoding, is then used for each group of users semi-statically. Next, with reduced dimensions on the effective channel, simplified channel feedback can be realized and the second-stage dynamic precoding can be applied. The performance of such system design is largely dependent on user grouping. In [10], a K-means clustering scheme, based on chordal distance as the clustering metric, is introduced for user grouping. In this paper, instead of chordal distance, we propose three similarity measures as grouping metric, namely, *weighted likelihood similarity measure*, *subspace projection based similarity measure*, and *Fubini-Study based similarity measure*. We also propose two clustering methods, i.e., *hierarchical clustering* and *K-medoids clustering*, for user grouping with the proposed metrics. Through theoretical analysis and simulations, we validate the proposed approach and find that the combination of weighted likelihood similarity measure and hierarchical clustering achieves the largest capacity among all the schemes examined in this paper.

Once user groups are formed, another important issue is user scheduling, i.e., selecting users for transmission based on instantaneous channel conditions. In this paper, we propose a dynamic user scheduling method and derive a lower bound for its achievable performance. If there are only a few active users, some groups may barely have users while some other groups are overloaded. Therefore, we also consider the load balancing problem and develop an effective solution algorithm. While some preliminary results can be found in [11], we substantially extend our study with new contributions including new methods, analysis, discussions, and results in this paper.

The remainder of this paper is organized as follows. Related works are discussed in Section II. In Section III, we present the system model and preliminaries. We address the user grouping and user scheduling problems in Sections IV and V, respectively. Joint user grouping and group load balancing is examined in Section VI. Our simulation study is presented in Section VII and Section VIII concludes this paper.

II. RELATED WORKS

As aforementioned, most of the existing works on massive MIMO focus on TDD systems. Although TDD has the advantage of exploiting the channel reciprocity, pilot contamination remains the biggest problem for TDD systems [1], [2], [5]. For FDD systems, the system bottleneck lies in the cost of acquiring CSIT. Broadly speaking there are two types of transmission modes: open-loop and closed-loop, representing the system without and with feedback, respectively. Our paper falls into the latter category.

Assuming that the base station and the users share a common set of training signals, both open-loop and closed-loop training frameworks are proposed in [12]. In the open-loop mode, the base station transmits training signals in a round-robin manner, so that the receivers could estimate the current channels using spatial or temporal correlations and previous channel estimations. In the closed-loop mode, users select the best training signal based on previously received signals and return the index of these training signals to the base station. In the next phase, the base station sends the training signals according to the feedback in previous phases. In [13], the feedback rate has been taken into consideration. Since for a fixed feedback rate per antenna, channel quantization grows exponentially with the number of transmit antennas, a noncoherent trellis-coded quantization is proposed with complexity growing linearly with the number of antennas.

Pilot pattern design for channel estimation is considered in [14]. Presuming wireless channel to be a stationary Gauss-Markov random process, pilot pattern is then designed based on Kalman filtering, spatial and temporal channel correlations. It is shown that the proposed scheme has low complexity but better performance, especially for the one-ring channel model.

A codebook design method is presented in [15] with limited or extremely low feedback, which could be considered as an open-loop approach. The compressive sensing technique is proposed in [16] to reduce the training and feedback overhead for CSIT acquisition. Due to the hidden joint sparsity structure of massive MIMO systems, a distributed compressive CSIT estimation scheme is proposed. The advantage is that compressed measurements are taken locally at users, while CSIT recovery is jointly performed at the base station. The proposed scheme has been shown to outperform five other algorithms in terms of normalized mean absolute error for CSIT recovery and have close performance to a so-called genie-aided scheme.

Similar to [9] and [10], Chen and Lau in [17] decompose the overall precoder into an outer precoder and an inner precoder, where the outer precoder suppresses the inter-cell or inter-cluster interference and the inner precoder is used for intra-cluster multiplexing. The contribution of [17] is to reduce the complexity of computing the outer precoder from $\mathcal{O}(M^3)$ to $\mathcal{O}(M^2)$, and it is an online algorithm that is suitable for time-varying channels.

It can be seen that these prior papers have not considered the user grouping and scheduling problems in massive MIMO systems. Based on the framework of [10], our recent work in [11] proposes an improved K-means clustering scheme and a dynamic user selection scheme. Another problem considered in [11] is the load balancing problem, which is also addressed in [18] for TDD systems. In summary, the main contribution of this paper on massive MIMO in FDD systems over [10], [11] lies in three aspects: new user grouping schemes with new grouping metrics, new user scheduling schemes, and an effective load balancing design.

III. SYSTEM MODEL AND PRELIMINARIES

We consider a downlink system with M antennas at the base station (BS) and a single antenna at each user terminal (UT). The transmit antennas can have different geometries, e.g., being placed along one axis to form a uniform linear array (ULA), along a circle to form a uniform circular array (UCA), or in two or three dimensions. Denote y_k as the received signal at user k, $k = 1, 2, \ldots, K$. The signals received by all UTs $\mathbf{y}$ can be written as

$$\mathbf{y} = \mathbf{H}^H\mathbf{V}\mathbf{d} + \mathbf{z}, \tag{1}$$

where $(\cdot)^H$ denots the Hermitian of a matrix; $\mathbf{H}$, of dimension $M \times K$, is the actual channel between the BS and the users; $\mathbf{V}$ is the precoding matrix of dimension $M \times S$; $\mathbf{d}$ is the data vector of dimension $S \times 1$; and $\mathbf{z}$ is the zero mean circulant symmetric complex Gaussian noise vector. Throughout this paper, we use a bold upper (lower) case symbol to denote a matrix (vector), and a normal symbol to denote a scalar.

With the two-stage precoding approach in [9], precoding is conducted as a multiplication of two precoding matrices, i.e., $\mathbf{V} = \mathbf{B}\mathbf{P}$. The first part $\mathbf{B}$ of dimension $M \times b$ is the pre-beamforming matrix, which is designed based on the second order channel statistics, or in particular, the transmit spatial correlation. The same pre-beamforming matrix is semi-statically applied to the users with the same or similar transmit correlation, which forms a user group. Therefore, the pre-beamforming matrix is designed to suppress the interferences across different groups. We can see that the effective transmit size after pre-beamforming is b, which is determined by the dominant eigenmodes of the average transmit correlation of user groups. The second part $\mathbf{P}$ of dimension $b \times S$, is designed to suppress the interferences within each group with dynamical channel conditions. To compute $\mathbf{P}$, we apply the conventional zero-forcing beamforming (ZFBF) or regularized zero-forcing beamforming (RZFBF). Note that we have $S \leq b$ as the second-stage precoding is supposed to suppress the interference within the group. Denote $\tilde{\mathbf{H}} = \mathbf{B}^H\mathbf{H}$ as the effective channel after pre-beamforming. The received signal in (1) can be rewritten as

$$\mathbf{y} = \mathbf{H}^H\mathbf{B}\mathbf{P}\mathbf{d} + \mathbf{z} = \tilde{\mathbf{H}}^H\mathbf{P}\mathbf{d} + \mathbf{z}. \tag{2}$$

We adopt the one-ring channel model in [9] and [11]. Let θ be the azimuth angle of the user location, s the distance between the BS and the user, r the radius of the scattering ring, and Δ the angle spread, which can be approximated as $\Delta \approx \arctan(r/s)$. Then the (m, p)-th entry of the channel covariance matrix $\mathbf{R}$ of the transmitter is given by

$$[\mathbf{R}]_{m,p} = \frac{1}{2\Delta}\int_{-\Delta}^{\Delta} e^{j\mathbf{k}^T(\alpha+\theta)(\mathbf{u}_m-\mathbf{u}_p)}d\alpha, \tag{3}$$

where $\mathbf{k}(\alpha) = -\frac{2\pi}{\lambda}(\cos(\alpha), \sin(\alpha))^T$ is the vector for a planar wave impinging with Angle of Arrival (AoA) α, λ is the carrier wavelength, $\mathbf{u}_m$ and $\mathbf{u}_p$ are the position vectors of antennas m and p, respectively, and $(\cdot)^T$ denotes the transpose operation. It can be verified that $\mathbf{R}$ is a normal matrix. With eigen-decomposition, we have

$$\mathbf{R} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^H, \tag{4}$$

where $\mathbf{U}$ is a unitary matrix comprising eigenvectors of $\mathbf{R}$ and $\mathbf{\Lambda}$ is a diagonal matrix with eigenvalues of $\mathbf{R}$ as the diagonal entries. Furthermore, the actual channel is generated using the following model

$$\mathbf{h} = \mathbf{U}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{w}, \tag{5}$$

where $\mathbf{w}$ is a vector of complex random variables and $\mathbf{w} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I})$, i.e., $\mathbf{w}$ is circularly-symmetric Gaussian.

Let G be the number of groups, $\mathcal{S}_g$ the user set in group g, and $|\mathcal{S}_g|$ the size of group g. We then have $\mathbf{H}_g = [\mathbf{h}_{g1}, \mathbf{h}_{g2}, \ldots, \mathbf{h}_{g|\mathcal{S}_g|}]$, $\mathbf{H} = [\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_G]$, $\mathbf{B} = [\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_G]$, and $\tilde{\mathbf{H}}_g = \mathbf{B}_g^H\mathbf{H}_g$, where h_{g_i} is the channel of user i in group g and $\mathbf{B_g}$ is the precoding matrix for group g. The signal vector received by the g-th group of users is then given by

$$\mathbf{y}_g = \tilde{\mathbf{H}}_g^H\mathbf{P}_g\mathbf{d}_g + \sum_{g' \neq g}\mathbf{H}_g^H\mathbf{B}_{g'}\mathbf{P}_{g'}\mathbf{d}_{g'} + \mathbf{z}_g,$$
$$g = 1, 2, \ldots, G. \tag{6}$$

The details of designing $\mathbf{B}_g$ and $\mathbf{P}_g$ are omitted here. Interested readers are referred to [11] and references therein.

IV. USER GROUPING IN MASSIVE MIMO SYSTEM

In order to suppress the inter-group interference, the pre-beamforming matrix $\mathbf{B}_g$ for group g shall be carefully designed based on all the group centers $\mathbf{R}_g$, $g = 1, 2, \ldots, G$. Note that the group center can be obtained by averaging the subspace of all the group members or by simply assigning one of the group members to be the group center. User grouping also has impacts on user scheduling, since for each pre-beamforming group, only the users within the group can be scheduled. Therefore it is important to design an effective user grouping method for enhanced system capacity.

The idea of user grouping is illustrated in Fig. 1. The big triangle in the middle represents the massive MIMO base station. Other markers except the red-cross represent users. Users from different groups are differentiated by different markers and colors. The red cross is the virtual group center. The dashed lines indicate the connections between users and group centers.

For user grouping, we first need to obtain the similarities (or distances) among the users and groups, and then group users based on a certain metric. Each user grouping scheme consists of two parts, the similarity measure and clustering method. In this section, we first review the K-means clustering method and the chordal distance as similarity measure presented in [10] and [11]. Then we propose new similarity measures as the grouping metric and new clustering methods.

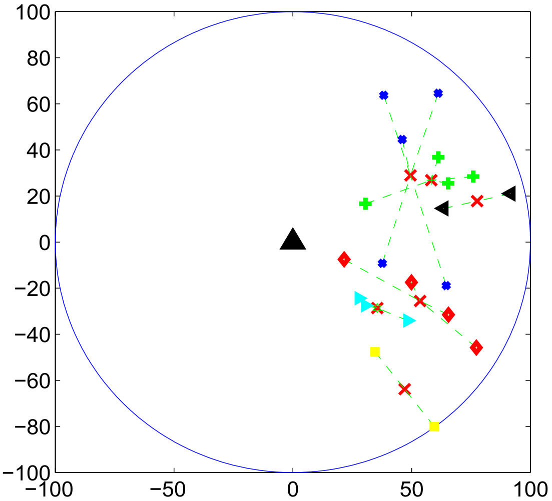

FIGURE 1. User grouping scenario.

Most of the clustering schemes in the literature only handle the matrix dataset, i.e., the entire dataset is a matrix. However, for our case here, each data entry is a matrix. The entire dataset is comprised of a large number of matrices. Thus, one of our contributions is to form efficient low-complexity grouping methods for datasets with many matrices. Also note that different clustering methods and similarity measures can be combined in various ways. It is useful to evaluate the various combinations to find the best one.

A. K-MEANS USER GROUPING AND CHORDAL DISTANCE

In [10], a K-means clustering algorithm for user grouping is presented. The similarity measure of the K-means clustering algorithm is the chordal distance defined as

$$d_c(\mathbf{U}_k, \mathbf{V}_g) = \left\|\mathbf{U}_k\mathbf{U}_k^H - \mathbf{V}_g\mathbf{V}_g^H\right\|_F^2, \qquad (7)$$

where $\mathbf{U}_k$ is the matrix of the eigenvectors of $\mathbf{R}_k$, i.e., $\mathbf{R}_k = \mathbf{U}_k\mathbf{\Lambda}_k\mathbf{U}_k^H$, and $\mathbf{V}_g$ is the matrix of the eigenvectors of the group center $\mathbf{R}_g$. User grouping is then achieved with an iterative process. In each iteration, each user is assigned to the group with the minimum chordal distance. Then the group center is updated using unitary matrix of users currently associated with the group as

$$\mathbf{V}_g = \Upsilon\Big\{\frac{1}{|\mathcal{S}_g|}\sum_{k\in\mathcal{S}_g}\mathbf{U}_k\mathbf{U}_k^H\Big\}. \qquad (8)$$

Note that $\Upsilon(\cdot)$ denotes the unitary matrix after eigen decomposition.

B. WEIGHTED LIKELIHOOD SIMILARITY MEASURE

Instead of chordal distance, we first propose a weighted likelihood function as the similarity measure between a user and a group, which is defined as

$$\mathcal{L}(\mathbf{R}_k, \mathbf{V}_g) \triangleq \left\|(\mathbf{U}_k\mathbf{\Lambda}_k^{\frac{1}{2}})^H\mathbf{V}_g\right\|_F^2. \qquad (9)$$

The proposed likelihood metric uses the projection of the eigenspaces of the users to that of the group centers, so that users can be readily separated into different groups. For instance, if user k is very close to group center g, or $\mathbf{U}_k \approx \mathbf{V}_g$, then $\mathbf{U}_k^H\mathbf{V}_g$ would result in a large value due to the property of unitary matrix. If $\mathbf{U}_k$ is much different from $\mathbf{V}_g$, then $\mathbf{U}_k^H\mathbf{V}_g$ would produce a very small value due to the orthogonality of unitary matrices. The weighted likelihood also takes into account the weights of different eigenmodes so that the user's group is mainly determined by the dominant eigenmodes.

Given $\mathcal{L}(\mathbf{R}_k, \mathbf{V}_g)$ for each user k and group g, we assign each user to the group with the maximum likelihood. The group center $\mathbf{V}_g$ and the total likelihood $\mathcal{L}_{tot}$ are updated as

$$\mathbf{V}_g = \Upsilon\Big\{\frac{1}{|\mathcal{S}_g|}\sum_{k\in\mathcal{S}_g}\mathbf{R}_k\Big\}, \quad \mathcal{L}_{tot} = \sum_{g=1}^{G}\sum_{k\in\mathcal{S}_g}\mathcal{L}(\mathbf{R}_k, \mathbf{V}_g). \qquad (10)$$

Note that the weights of eigenmodes is considered in the proposed algorithm when updating the group center and the total likelihood.

Algorithm 1 Improved K-Means Clustering Algorithm With Weighted Likelihood Similarity Measure

```
Set $n = 0$, $\mathcal{L}_{tot}^{(0)} = 1$ ;
Randomly choose $G$ different indices (denoted as $\pi(g), \forall g$)
from the set $\{1, 2, \cdots, K\}$ and set $\mathbf{V}_g^{(n)} = \mathbf{U}_{\pi(g)}, \forall g$;
$n = 1$, $\mathcal{L}_{tot}^{(n)} = 0$;
while $\left|\mathcal{L}_{tot}^{(n)} - \mathcal{L}_{tot}^{(n-1)}\right| > \epsilon\mathcal{L}_{tot}^{(n-1)}$ do
    Let $\mathcal{S}_g^{(n)} = \emptyset, g = 1, 2, \cdots, G$ ;
    for $k = 1, 2, \cdots, K$ do
        for $g = 1, 2, \cdots, G$ do
            Compute $\mathcal{L}(\mathbf{R}_k, \mathbf{V}_g^{(n-1)}) = \left\|(\mathbf{U}_k\mathbf{\Lambda}_k^{\frac{1}{2}})^H\mathbf{V}_g^{(n-1)}\right\|_F^2$ ;
        end
        Find $g_k^* = \arg\max_{g'} \mathcal{L}(\mathbf{R}_k, \mathbf{V}_{g'}^{(n-1)})$ and let
        $\mathcal{S}_{g_k^*}^{(n)} = \mathcal{S}_{g_k^*}^{(n)} \cup \{k\}$ ;
    end
    for $g = 1, 2, \cdots, G$ do
        $\mathbf{V}_g^{(n)} = \Upsilon\left\{\frac{1}{|\mathcal{S}_g^{(n)}|}\sum_{k\in\mathcal{S}_g^{(n)}}\mathbf{R}_k\right\}$ ;
    end
    Compute $\mathcal{L}_{tot}^{(n)} = \sum_{g=1}^{G}\sum_{k\in\mathcal{S}_g^{(n)}}\mathcal{L}(\mathbf{R}_k, \mathbf{V}_g^{(n)})$ ;
    $n = n + 1$ ;
end
Assign $\mathbf{V}_g = \mathbf{V}_g^{(n)}$ and $\mathcal{S}_g = \mathcal{S}_g^{(n)}$.
```

With the weighted likelihood similarity measure, we now propose an improved K-means clustering algorithm, which is described in Algorithm 1. Note that in Algorithm 1, $\mathbf{U}_{\pi(g)}$ is the unitary matrix of the user with index $\pi(g)$ and ϵ is a small number to control the termination of the iterative algorithm.

C. SUBSPACE PROJECTION BASED SIMILARITY MEASURE

We next propose another similarity measure, which is based on subspace projection, given by

$$\mathcal{P}(\mathbf{U}_k, \mathbf{V}_g) = \left\|\mathbf{V}_g\mathbf{V}_g^H\mathbf{U}_k - \mathbf{U}_k\right\|_F^2. \tag{11}$$

We can see from the above definition that we measure the similarity between user k and group g by firstly projecting user k to group g and then calculating the distance between user k and its projection on group g. If user k is the group center or in close proximity to the group center, $\mathcal{P}(\mathbf{U}_k, \mathbf{V}_g)$ would be close to zero.

D. FUBINI STUDY BASED SIMILARITY MEASURE

The third similarity measure we propose is the Fubini-Study (FS) based similarity measure, which is given by

$$\mathcal{F}_S(\mathbf{U}_k, \mathbf{V}_g) = \arccos\left|\det\left(\mathbf{U}_k^H\mathbf{V}_g\right)\right|. \tag{12}$$

We can see that if user k is close to the group center g, then $\mathcal{F}(\mathbf{U}_k, \mathbf{V}_g)$ would be close to 0. Otherwise, $\mathcal{F}(\mathbf{U}_k, \mathbf{V}_g)$ would be larger if the user is farther from the group center. The FS distance can then be another choice of the similarity measure for the user grouping.

E. HIERARCHICAL USER GROUPING

In addition to the new similarity measures, we also propose new user grouping schemes. The first new user grouping scheme that we propose employs the agglomerative hierarchical clustering method. Different from the K-means method, which essentially looks at all possible combinations of users and groups, the agglomerative hierarchical clustering method starts with each individual user forming a user group. It then proceeds by a series of successive mergers based on certain criteria. Eventually, all users can form one single group. We can terminate the scheme when the desired number of groups is reached.

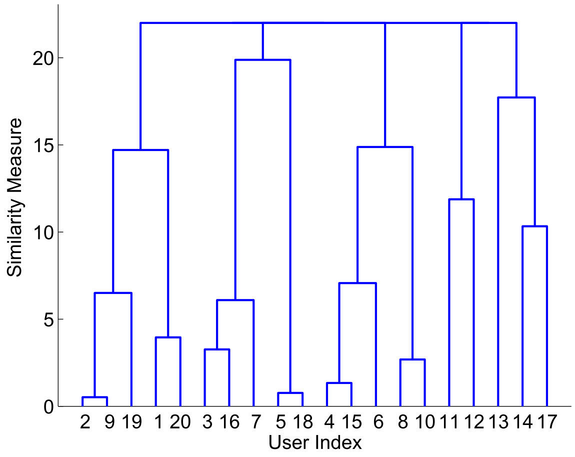

FIGURE 2. Hierarchical clustering illustration.

An example of the agglomerative hierarchical clustering method is illustrated in Fig. 2. Initially, there are 20 users and thus 20 groups. The distance between any two users (or two initial groups) is calculated. At the first iteration, we find that the distance between users 2 and 9 is the smallest. So users 2 and 9 are merged into one group as shown in Fig. 2. At the second iteration, the distance between users 4 and 15 is found to be the smallest. So users 4 and 15 are merged as a group. We iterate such group merging process until the desired number of groups is reached.

One may notice in the example above that at one intermediate step, the group comprised of users 1 and 20 is found to be close to the group comprised of users 2, 9 and 19. So one important issue in hierarchical clustering is how to define the similarity measure or distance between existing *groups* and newly defined groups (called linkage methods). Typical linkage methods include: single linkage, complete linkage, average linkage, ward linkage, median linkage, and weighted average linkage, which are explained as follows.

Since we only merge two groups at each step, the linkage methods can be defined in an inductive manner. Suppose we have merged groups v_i and v_j to get a new group $(v_iv_j) \triangleq v_i \bigcup v_j$. Now we need to define the distance between the remaining groups and the new group (v_iv_j). Let group v_q be one of the remaining groups. The distance between (v_iv_j) and v_q given by the single linkage method is

$$d_{(v_iv_j),v_q} \triangleq \min\left\{d_{v_i,v_q},\ \ d_{v_j,v_q}\right\}. \tag{13}$$

That means the distance between group (v_iv_j) and v_q is the minimum of the two distances d_{v_i,v_q} and d_{v_j,v_q}, where distance d_{v_i,v_q} and d_{v_j,v_q} have been previously calculated in the same manner.

Complete linkage defines the distance between (v_iv_j) and v_q as the maximum of the two distances d_{v_i,v_q} and d_{v_j,v_q}, given by

$$d_{(v_iv_j),v_q} \triangleq \max\left\{d_{v_i,v_q},\ \ d_{v_j,v_q}\right\}. \tag{14}$$

Average linkage defines the distance between (v_iv_j) and v_q as the average of all the pair-wise distances, given by

$$d_{v_iv_j,v_q} = \frac{|v_i|d_{v_i,v_q} + |v_j|d_{v_j,v_q}}{|(v_iv_j)|}. \tag{15}$$

Ward linkage defines the distance between (v_iv_j) and v_q as

$$d_{(v_iv_j),v_q} \triangleq \frac{(|v_i| + |v_q|)d_{v_i,v_q} + (|v_j| + |v_q|)d_{v_j,v_q} - |v_q|d_{v_i,v_j}}{|v_i| + |v_j| + |v_q|}. \tag{16}$$

Median linkage defines the distance between (v_iv_j) and v_q as

$$d_{(v_iv_j),v_q} \triangleq \frac{1}{2}d_{v_i,v_q} + \frac{1}{2}d_{v_j,v_q} - \frac{1}{4}d_{v_i,v_j}. \tag{17}$$

Finally, weighted average linkage defines the distance between (v_iv_j) and v_q as

$$d_{(v_iv_j),v_q} \triangleq \frac{1}{2}d_{v_i,v_q} + \frac{1}{2}d_{v_j,v_q}. \tag{18}$$

Algorithm 2 Hierarchical Clustering Algorithm

```
For given $G$ and the user set $\mathcal{U} = \{1, 2, \cdots, K\}$, start with each user forming a group, i.e., $v_q = \{q\}, q = 1, 2, \cdots, K$ ;
for $k = 1, 2, \cdots, K$ do
    for $k' = 1, 2, \cdots, K$ do
        Calculate pair-wise similarity between users (or groups) using (7) or (9) ;
    end
end
while The number of groups is greater than G do
    Search for and merge the groups with the maximal similarity ;
    Calculate the pair-wise distance between user (or group) and updated group using one of the linkage methods (13)–(18) ;
end
```

Given the linkage definitions, we propose our hierarchical clustering algorithm as presented in Algorithm 2. As discussed, the algorithm keeps on merging the closest groups, until the desired number of groups is reached.

Next we present a complexity analysis for K-means clustering and hierarchical clustering methods. Note that the framework of K-means is essentially similar to Algorithm 1. Denote the complexity of computing similarities for all user-group (or group-group) pairs as $C_{s-kmean}$ for K-means clustering and C_{s-hier} for hierarchical clustering; searching for the maximal similarity pair and pairing them up as $C_{m-kmean}$ for K-means clustering and C_{m-hier} for hierarchical clustering; and updating group center as $C_{u-kmean}$ for K-means clustering and C_{u-hier} for hierarchical clustering, respectively. We have the following proposition for K-means clustering. The proof is omitted for brevity.

Proposition 1: The complexity of K-means clustering is $\mathcal{O}(G^K C_{s-kmean})$. More specifically, it is $\mathcal{O}(G^K KG \times (2M^3 + M^2))$ for K-means clustering with chordal distance and $\mathcal{O}(G^K KG \times \left[(r^*)^3 + (Mr^*)^2\right])$ for K-means clustering with weighted likelihood similarity measure, where r^* is the effective rank of $\mathbf{R}_k$, i.e., the number of columns in $\mathbf{U}_k$.

For Algorithm 2, we have the following proposition. The proof is omitted for brevity.

Proposition 2: The complexity of hierarchical clustering is $\mathcal{O}(C_{s-hier})$. More specifically, it is $\mathcal{O}(\frac{K(K-1)}{2}(2M^3+M^2))$ for hierarchical clustering with chordal distance and $\mathcal{O}(\frac{K(K-1)}{2}[(r^*)^3 + (Mr^*)^2])$ for hierarchical clustering with weighted likelihood similarity measure.

We can see that the algorithm complexities depend on the number of users K, the number of antennas M, the effective rank r^*, and the choice of number of groups G. If K is relatively small and G is relatively large, hierarchical clustering is more computationally efficient. However, if K is much large and G is small, K-means clustering may be more computationally efficient. Fig. 3 presents the complexity comparison for an example case. In this simulation, we let $M = 100$, $G = 6$, and $r^* = 11$.

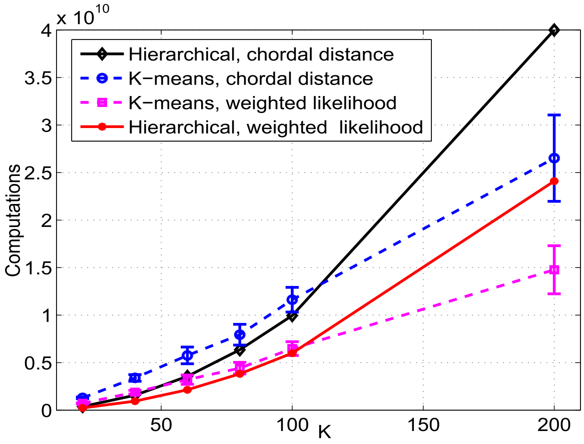

FIGURE 3. Complexity comparison between K-means clustering with similarity measure (7) and Hierarchical-clustering with similarity measure (9).

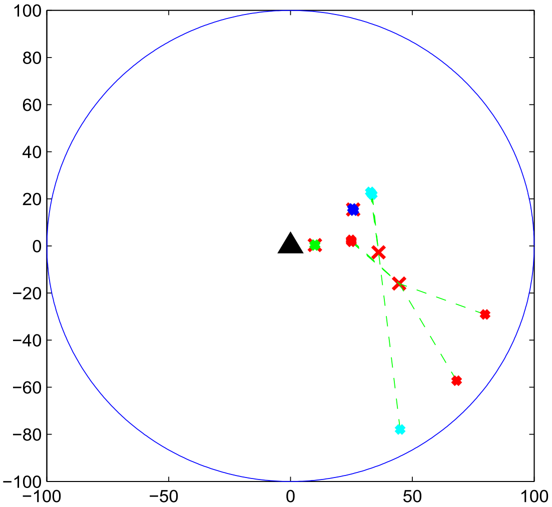

FIGURE 4. User grouping result with K-means clustering.

There are two advantages of hierarchical clustering compared with K-means clustering. First, hierarchical clustering does not rely on the initial choices of group center. For example, given the users' distributions, K-means clustering may end up with user groups shown in Fig. 4. We can see that there are several crossing lines for different groups, which suggests possibly inappropriate user grouping. On the contrary, Fig. 5 shows the grouping results obtained by hierarchical clustering, which is clearly a better grouping configuration. This advantage is especially true when the number of users is small. Second, according to Propositions 1 and 2, hierarchical clustering is generally more computationally efficient when the number of users is less than or equal to 100.

F. K-MEDOIDS USER GROUPING

The second user grouping scheme we propose is the K-medoids clustering method. K-medoids clustering is similar to K-means clustering. However, the main difference lies in the approach of updating group center. While K-means uses the average of the group members (or called centroids), K-medoids tries every group member (medoids) as the group

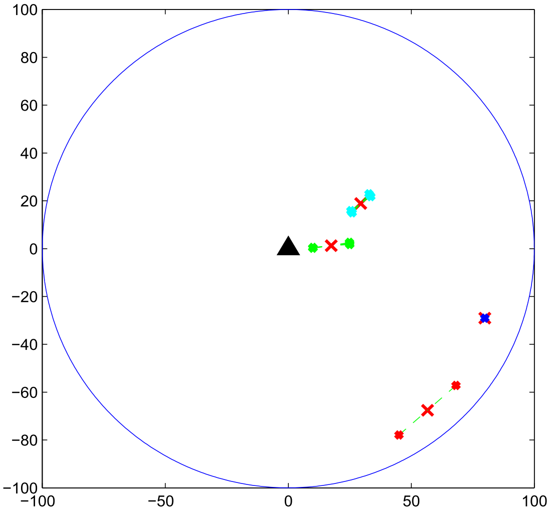

FIGURE 5. User grouping result with hierarchical clustering.

Algorithm 3 K-Medoids Clustering Algorithm

```
Set $n = 0$, $\mathcal{L}_{tot}^{(0)} = 1$ ;
Randomly choose $G$ different indices (denoted as $\pi(g)$, $\forall g$) from the set $\{1, 2, \cdots, K\}$ and set $\mathbf{V}_g^{(n)} = \mathbf{U}_{\pi(g)}, \forall g$ ;
$n = 1$, $\mathcal{L}_{tot}^{(n)} = 0$ ;
while $\left|\mathcal{L}_{tot}^{(n)} - \mathcal{L}_{tot}^{(n-1)}\right| > \epsilon \mathcal{L}_{tot}^{(n-1)}$ do
    Let $\mathcal{S}_g^{(n)} = \emptyset$, $g = 1, 2, \cdots, G$;
    for $k = 1, 2, \cdots, K$ do
        for $g = 1, 2, \cdots, G$ do
            Compute similarity measure using (7) or (9) ;
        end
        Find $g_k^* = \arg\max_{g'} \mathcal{L}(\mathbf{R}_k, \mathbf{V}_{g'}^{(n-1)})$, or $g_k^* = \arg\min_{g'} d_c(\mathbf{U}_k, \mathbf{V}_{g'}^{(n-1)})$ and let $\mathcal{S}_{g_k^*}^{(n)} = \mathcal{S}_{g_k^*}^{(n)} \cup \{k\}$ ;
    end
    for $g = 1, 2, \cdots, G$ do
        for $k \in \mathcal{S}_g^{(n)}$ do
            $WGRSS(g)_k^{(n)} = \sum_{k \in \mathcal{S}_g^{(n)}} \mathcal{L}(\mathbf{R}_k, \mathbf{R}_{k'})$ or $WGRSS(g)_k^{(n)} = \sum_{k \in \mathcal{S}_g^{(n)}} d_c(\mathbf{U}_k, \mathbf{U}_{k'})$ ;
        end
        Find $k^* = \arg\max_k WGRSS(g)_k^{(n)}$ for likelihood or $k^* = \arg\min_k WGRSS(g)_k^{(n)}$ for distance, and let $\mathbf{V}_g^{(n)} = \mathbf{U}_{k^*}$ ;
    end
    $\mathcal{L}_{tot}^{(n)} = \sum_{g=1}^{G} WGRSS(g)_{k^*}^{(n)}$ ;
    $n \leftarrow n + 1$ ;
end
Assign $\mathbf{V}_g = \mathbf{V}_g^{(n)}$ and $\mathcal{S}_g = \mathcal{S}_g^{(n)}$ ;
```

center and uses the one with the least within group residue sum of squares (WGRSS) for distance (or the one with the largest WGRSS for likelihood). The user grouping algorithm based on the K-medoids clustering method is presented in Algorithm 3. Due to the exhaustive search of group centers, the computational complexity of K-medoids is lower bounded by the complexity of K-means, and is hence comparably higher.

V. USER SCHEDULING IN MASSIVE MIMO SYSTEMS

After forming the user groups, we can obtain the pre-beamforming matrix $\mathbf{B}_g$ for each group g. At a particular time slot, based on the instantaneous channel conditions of the users, we dynamically schedule a subset of users in each group for the transmissions in this time slot.

In [10], a MAX and an ALL user scheduling algorithm are presented. The MAX user scheduling is only based on the feedback of beam index with the max SINR, while the ALL user scheduling is based on the user's feedback of all beamforming SINRs, i.e., SINR for every beam selection. Different from this approach, we propose a dynamic user scheduling algorithm that schedules users in a greedy manner. In particular, at each step, the proposed algorithm only schedules the user that can achieve the largest gain in the system throughput. The proposed algorithm is presented in Algorithm 4.

Algorithm 4 Greedy Algorithm for Dynamic User Selection and Beamforming With Determined User Grouping

```
User groups $\{\mathcal{S}_g\}$ are given ;
Initially set $\mathcal{U} = \{1, 2, \cdots, K\}$, $\mathcal{C} = 0$, and $\mathcal{K}_g = \emptyset, \forall g$ ;
while Termination conditions ($\sum_g |\mathcal{K}_g| = \sum_g b_g$, $\kappa(k^*, g_{k^*}) = 0$, or $\mathcal{U} = \emptyset$) are not satisfied do
    for $k \in \mathcal{U}$ do
        if $|\mathcal{K}_{g_k}| < S_g$ then
            Set $\mathcal{K}'_g = \mathcal{K}_g \cup \{k\}$ if $k \in \mathcal{S}_g$, and $\mathcal{K}'_{g'} = \mathcal{K}_{g'}, \forall g' \neq g$ ;
            Perform ZFBF or RZFBF based on $\{\mathcal{K}'_g\}$ and $\{\mathbf{B}_g\}$;
            Compute the gain $\kappa(k, g) = \max\left\{0, \mathcal{C}(\{\mathcal{K}'_g\}, \{\mathbf{B}_g\}) - \mathcal{C}(\{\mathcal{K}_g\}, \{\mathbf{B}_g\})\right\}$;
        end
    end
    Obtain $(k^*, g_{k^*}) = \arg\max_{k \in \mathcal{U}} \kappa(k, g)$ ;
    if $(k^*, g_{k^*}) \neq \emptyset$ then
        $\mathcal{U} \leftarrow \mathcal{U} \backslash k^*$ ;
        $\mathcal{K}_{g_{k^*}} \leftarrow \mathcal{K}_{g_{k^*}} \cup \{k^*\}$ ;
    end
end
```

Given the user grouping and scheduling, we can calculate the instantaneous SINR, γ_{g_k}, for user k in group g as

$$\gamma_{g_k} = \frac{\frac{p}{\sum_g S_g} \zeta_g^2 \left|\mathbf{h}_{g_k}^H \mathbf{B}_g \mathbf{P}_g(:, g_k)\right|^2}{1 + I_{in}(g, k) + I_{it}(g, k)}, \tag{19}$$

where I_{in} and I_{it} denote the inner group and inter group interferences, respectively, computed as

$$I_{in}(g, k) = \frac{p}{\sum_g S_g} \zeta_g^2 \sum_{j \neq k} \left|\mathbf{h}_{g_k}^H \mathbf{B}_g \mathbf{P}_g(:, g_j)\right|^2$$

$$I_{it}(g, k) = \frac{p}{\sum_g S_g} \sum_{g' \neq g} \zeta_{g'}^2 \sum_j \left|\mathbf{h}_{g_k}^H \mathbf{B}_{g'} \mathbf{P}_{g'}(:, g_j)\right|^2,$$

$\mathbf{P}_g(:, g_k)$ denotes the submatrix containing all the rows and the g_k-th column of $\mathbf{P}_g$, and ζ_g^2 is the scaling factor for satisfying certain power constraint, which can be obtained as $\zeta_g^2 = \frac{S_g}{\mathrm{tr}(\mathbf{P}_g^H \mathbf{B}_g^H \mathbf{B}_g \mathbf{P}_g)}$. The rate for scheduled user g_k is $\eta_{g_k} = \log_2(1 + \gamma_{g_k})$ and the overall system throughput $\mathcal{C}$ is obtained as

$$\mathcal{C} = \sum_{g=1}^{G} \sum_{k \in \mathcal{K}_g} \eta_{g_k},$$

where $\mathcal{K}_g$ is the scheduled user set in group g. Obviously, $\mathcal{C}$ is a function of $\{\mathcal{K}_g\}$ and precoding for all co-scheduled users, denoted as $\mathcal{C}(\{\mathcal{K}_g\}, \{\mathbf{B}_g\}, \{\mathbf{P}_{g_k}\})$.

In the following part of this section, we present a lower bound of the proposed greedy algorithm for dynamic user selection.

Lemma 1: In Algorithm 4, the first user scheduled achieves the largest rate increase.

Proof: This is resulted from Step 11 of Algorithm 4 and the fact that the first user scheduled has the largest rate among all users without any interference. For each user scheduled in the subsequent iterations, the resulting user rate is always smaller than the user rate evaluated in the first iteration due to the intra- and inter-group interference from the users already scheduled and power splitting among scheduled users. Therefore the rate increase in all other iterations is smaller that that of the first iteration. ■

Denote the achievable rate of the first scheduled user as $\mathcal{Z}_1$, the system sum rate of Algorithm 4 as x, and the system sum rate of the optimal user scheduling as $\mathcal{X}$. We have the following lemma.

Lemma 2: $\mathcal{Z}_1 \leq x \leq |\mathcal{U}|\mathcal{Z}_1$.

Proof: $\mathcal{Z}_1 \leq x$ is trivial, since Algorithm 4 would schedule at least one user. Since there are $|\mathcal{U}|$ users, from Lemma 1 we know that the achievable rates of them are all upper bounded by $\mathcal{Z}_1$, $x \leq |\mathcal{U}|\mathcal{Z}_1$ thus holds. ■

Using similar arguments as the proof of Lemma 1, we can show that the following lemma holds.

Lemma 3: $\mathcal{X} \leq |\mathcal{U}|\mathcal{Z}_1$.

From Lemma 2 and Lemma 3, we have that $\frac{\mathcal{X}}{|\mathcal{U}|} \leq \mathcal{Z}_1 \leq x \leq |\mathcal{U}|\mathcal{Z}_1$. So we have the following theorem.

Theorem 1: The greedy algorithm for dynamic user selection can achieve an objective value that is at least $\frac{1}{|\mathcal{U}|}$ of the optimal user selection solution.

Lemma 3 and Theorem 1 not only give the lower and upper bounds for the greedy algorithm, but also for the optimal user scheduling scheme. Fig. 6 illustrates this bound of the optimal scheme. For obvious reason, we let $G = 1$ here. We can see that when the number of users is not large, our greedy user scheduling algorithm approaches the upper bound of the optimal user scheduling. Note that as the number of users increases, the bound becomes looser.

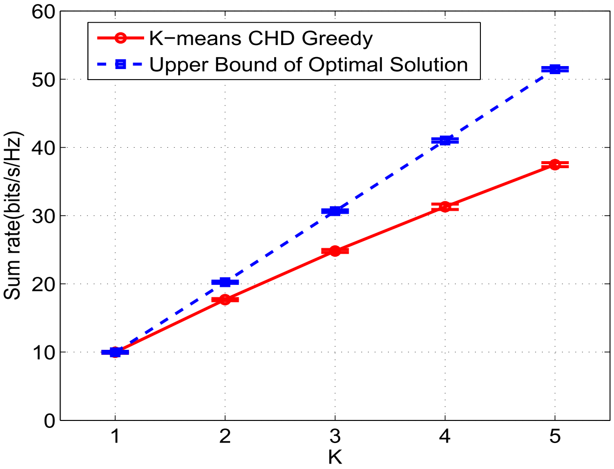

FIGURE 6. Greedy algorithm versus the upper bound.

VI. USER GROUPING WITH JOINT GROUP LOAD BALANCING AND PRECODING DESIGN

In real life applications, many users may gather at one geographic location (e.g., in a skyscraper). If we design the precoder exactly as discussed, these users will form a big group. It would be desirable to offload some of the users to other groups, to achieve fairness among hte users. This is because with more members in a group, each member's chance of getting scheduled for transmission will be smaller. We develop a user grouping method considering group load balancing and user proportional fairness in this section. User grouping with proportional fairness can be formulated as the following optimization problem.

$$\max_{\{x_{kg}\}} \quad \mathcal{J} = \sum_{k=1}^{K} \sum_{g=1}^{G} x_{kg} \log\left(\frac{\overline{\eta}_{gk}}{\sum_i x_{ig}}\right) \tag{20}$$

$$s.t. \quad \sum_g x_{kg} = 1, \quad \forall k \in \{1, 2, \cdots, K\} \tag{21}$$

$$x_{kg} \in \{0, 1\}, \quad \forall k, g, \tag{22}$$

where $\mathcal{J}$ denotes the utility to optimize, $\overline{\eta}_{gk}$ is the average user throughput, i.e., $\overline{\eta}_{gk} = \log_2(1 + \overline{\gamma}_{gk})$, $\overline{\gamma}_{gk}$ is the average SINR when user k is assigned to group g, and x_{kg} is the assignment indicator defined as

$$x_{kg} = \begin{cases} 1, & \text{if user } k \text{ is in group } g \\ 0, & \text{otherwise,} \end{cases} \quad \forall k, g. \tag{23}$$

Given constraint (23), we can see that the optimization problem (20) is combinatorial in nature. If we apply exhaustive search for this problem, the complexity is exponential. To make the problem tractable, we relax the variable x_{kg} to be a real number in the range of $[0, 1]$. The relaxed problem has the same objective function (20) and constraint (refeq:sumxkg), but with the following new constraint, which replaces constraint (22):

$$0 \leq x_{kg} \leq 1, \quad \forall k, \ g. \tag{24}$$

Lemma 4: The relaxed problem with constraint (24) is a convex optimization problem.

Proof: The objective function of problem (24) can be represented as $\sum_k \sum_g x_{kg} \log(\overline{\eta}_{gk}) - \sum_k \sum_g x_{kg} \log(\sum_i x_{ig})$.

The first term is affine. The second term is basically two concatenated sums of $x\log(x+a)$, where $0 \le a \le (K-1)$. The second derivative of $x\log(x+a)$ is $\frac{1}{x+a} + \frac{a}{(x+a)^2}$, which is positive for $0 \le a \le (K-1)$. So $x\log(x+a)$ is a convex function and $-\sum_k\sum_g x_{kg}\log(\sum_i x_{ig})$ is concave due to negative sums. Therefore $\sum_k\sum_g x_{kg}\log(\overline{\eta}_{gk}) - \sum_k\sum_g x_{kg}\log(\sum_i x_{ig})$ is concave. Since the constraints are linear, the problem is convex. ■

Given Proposition 4, we could apply an effective convex optimization technique to solve the relaxed problem. However, an important issue is how to obtain the average SINR $\overline{\gamma}_{gk}$. The challenge is, without user grouping and scheduling information, we cannot calculate the exact SINR for each user. Moreover, over different time slots, different users will be scheduled based on the user grouping result and the instantaneous channel states. Thus, in order to solve the problem, we need to find a way to approximate average SINR for each user in each group. We propose to approximate the average SINR based on following assumptions.

(i) Conjugate precoding [19], [20] for the target user;
(ii) There are no intra-group co-scheduled users;
(iii) Identity precoding for inter-group co-scheduled users.

We can obtain the SINR approximation as

$$\overline{\gamma}_{g_k} = \frac{\frac{p}{\sum_g b_g}\left|tr(\mathbf{B}_g^H\mathbf{R}_{g_k}\mathbf{B}_g)\right|}{1 + \frac{p}{\sum_g b_g}\sum_{g'\neq g}\left|tr(\mathbf{B}_{g'}^H\mathbf{R}_{g_k}\mathbf{B}_{g'})\right|}. \tag{25}$$

Due to the dynamic nature of the user scheduling and the objective of user group assignment itself, it is difficult to obtain the average SINR presuming multiuser MIMO scheduling. However, as in [21], when we consider the load balancing problem, it is reasonable to consider the single user resource allocation with user average SNR for the targeted cell. Therefore in our case, when we compute the average SINR for a user, we assume that in an instantaneous time slot, only the user of interest is scheduled in its group. Moreover, we treat other groups as the virtual neighboring cells and consider the identity precoding matrix for the interfering groups, which is a fairly good approximation for the interference. With these assumptions, we assume the best resource allocation for each user with average interference, which we think is appropriate for studying the user load balancing among groups. Otherwise it would be very difficult to approximate the average intra-group and inter-group interferences.

After obtaining the SINR approximation, similar to [21], the procedure to solve the relaxed user grouping optimization problem with load balancing is presented in Algorithm 6.

Theorem 2: The solution to the relaxed problem is also feasible and optimal to the original problem (20).

Proof: Since we relax the variables from binary to fractional, the solutions to the relaxed problem with constraint (24) actually upper bounds the solution to the original problem. However, we can see from Algorithm 6 that the solutions to the relaxed problem are integers other than fractions. So the optimal solution to the relaxed problem is also feasible to the original problem (20). Since the solutions to problem (20) cannot achieve higher utility than the solutions to the relaxed problem, the solutions to the relaxed problem is also optimal to the original problem. ■

Algorithm 5 User Grouping With Joint Group Load Balancing and Precoding Design Algorithm

```
Perform $K$-means Clustering Algorithm or Algorithm 1 to
obtain user group ID $x_{ij}$ ;
while $\mathcal{J}^{*(n-1)} - \mathcal{J}^{*(n-2)} > \epsilon\mathcal{J}^{*(n-2)}$ do
    for $g \in G$ do
        Find $\mathbf{V}_g^{*(n)}$ using (8) or the proposed weighted
        likelihood (10) ;
    end
    for $g \in G$ do
        Find $\mathbf{B}_g$ using approximate BD approach ;
    end
    for $k = 1, 2, \cdots, K$ do
        for $g = 1, 2, \cdots, G$ do
            Find $\gamma_{g_k}$ using (25) ;
        end
    end
    Optimize (20) using Algorithm 6 ;
    Update $x_{ij}$ and $\mathcal{J}^{*(n)}$ ;
end
```

Algorithm 6 Optimization Algorithm for (20)

```
$n = 0$, $\mu^{(1)} = 0$;
while the optimization has not converged do
    $n \leftarrow n + 1$;
    for $k = 1, 2, \cdots, K$ do
        for $g = 1, 2, \cdots, G$ do
            Compute $\overline{\gamma}_{g_k}$ and $\overline{\eta}_{g_k}$ ;
        end
        Assign user $k$ to group $g^*$ where
        $g^* = \arg\max_g\left(\log(\overline{\eta}_{g_k}) - \mu_g^{(n)}\right)$, and let $x_{kg^*}^{(n)} = 1$,
        $x_{kg}^{(n)} = 0$ for $g \neq g^*$ ;
    end
    for $g = 1, 2, \cdots, G$ do
        Each group chooses a step size $\delta^{(n)}$ and computes
        $K_g^{(n+1)} = \min\{K, e^{(\mu_g^{(n)}-1)}\}$,
        $\mu_g^{(n+1)} = \mu_g^{(n)} - \delta^{(n)}(K_g^{(n)} - \sum_k x_{kg}^{(n)})$ ;
    end
end
```

VII. SIMULATION STUDY

More numerical simulations are performed to evaluate the proposed schemes. The system configuration is provided in Table 1. In particular, we consider a 120° sector. For each user drop, the azimuth angle θ_k, angle spread Δ_k and distance s_k for user k are uniformly generated within the intervals $[\theta_{\min}, \theta_{\max}]$, $[\Delta_{\min}, \Delta_{\max}]$ and $[s_{\min}, s_{\max}]$, respectively. We average over 100 user drops for the entire simulation. In each user drop, we evaluate the performance with 200 channel realizations. We fix the number of groups as $G = 6$. For the antenna configuration, we consider the ULA case

TABLE 1. System configuration in the simulations.

Parameter	*Value*	*Parameter*	*Value*
θ_{min}	$-60°$	M	100
θ_{max}	$60°$	D	0.5
Δ_{min}	$5°$	G	6
Δ_{max}	$15°$	r_g^*	11
s_{min}	20 (m)	ϵ	10^{-3}
s_{max}	100 (m)	p	10, 20 dB

and place 100 antennas along the y-axis with 0.5λ spacing. According to (3), the (m, p)-th entry of the covariance matrix is given by

$$[\mathbf{R}]_{m,p} = \frac{1}{2\Delta} \int_{-\Delta}^{\Delta} e^{-j2\pi D(m-p)\sin(\alpha+\theta)} d\alpha. \tag{26}$$

Throughout the simulations, to find the first and second stage precoding matrices, we adopt the approximate BD approach and the regularized ZF precoding approach, respectively.

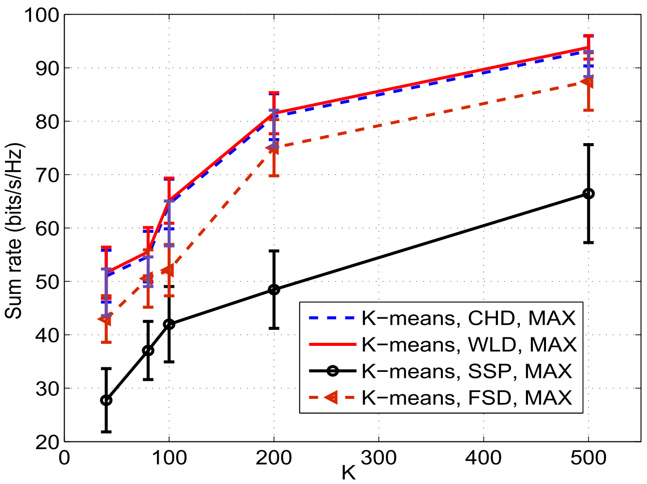

FIGURE 7. Similarity measure comparison.

Fig. 7 presents a comparison of the proposed similarity measures. For a fair comparison, we use the same clustering method K-means and user scheduling method MAX. Note that CHD stands for chordal distance defined in (7); WLD stands for weighted likelihood defined in (9); SSP stands for subspace projection defined in (11); FSD represents Fubini Study distance defined in (12). We find that WLD has a slightly higher throughput than CHD, which verifies the effectiveness of our proposed scheme. However, the sum rates of FSD and SSP are lower than that of CHD. Therefore we will not consider these similarity measures in the following simulations.

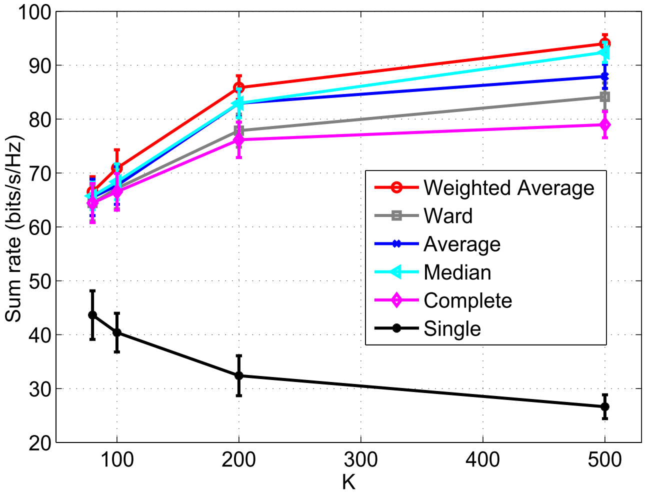

FIGURE 8. Comparison of linkage methods for hierarchical clustering.

Fig. 8 provides a comparison of the several linkage methods for hierarchical clustering. For a fair comparison, we use agglomerative hierarchical clustering, weighted likelihood similarity measure, and MAX user scheduling for all the linkage methods. We find that as the number of users increases, the sum rate of single linkage drops gradually. This is because the dimension of each cluster grows when there are more users. Using the distance between the nearest points of two clusters to represent the distance between two clusters becomes inaccurate. We can also observe that weighted average linkage achieves the highest throughput. Carefully looking into the definition of weighted average linkage, we can see that weighted average linkage puts higher weights on the members who join the group late, which are less similar to other group members. By giving higher weights to members who join the group late, better performance can be achieved in our scheme. We thus use weighted average linkage method for hierarchical clustering hereafter.

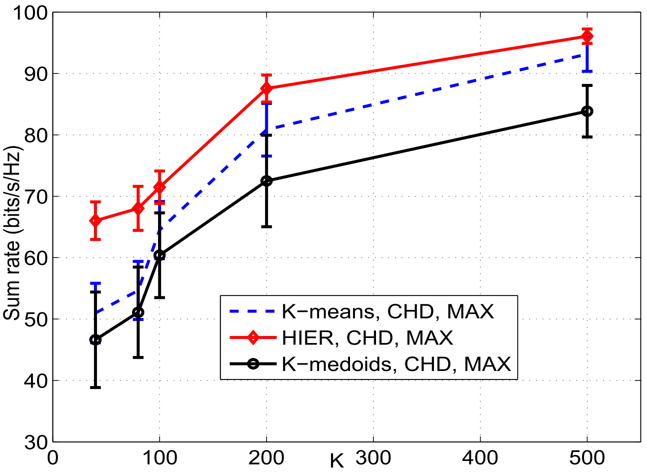

FIGURE 9. Clustering method comparison.

Fig. 9 presents a comparison of the proposed clustering methods. For a fair comparison, we use the same similarity measure CHD defined in (7) and the MAX user scheduling scheme. It can be observed that hierarchical clustering has the highest throughput, while K-medoids clustering has the lowest throughput for the entire range of user numbers.

Due to the relatively higher computational complexity and inferior performance, we do not consider K-medoids clustering in the following simulations. However, the efficacy of agglomerative hierarchical clustering has been demonstrated in Fig. 9. Moreover, hierarchical clustering has the advantage of lower computational complexity, which has been illustrated in Fig. 3.

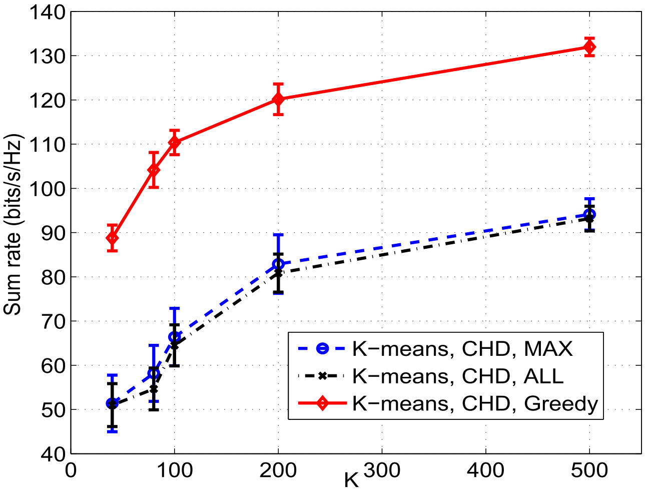

FIGURE 10. Scheduling methods comparison.

Fig. 10 is a comparison of the user scheduling schemes. For a fair comparison, we use the same K-means clustering and CHD similarity measure. It can be seen that our proposed greedy algorithm achieves the highest throughput. Although the proposed greedy algorithm is suboptimal, it greatly enhances the system throughput.

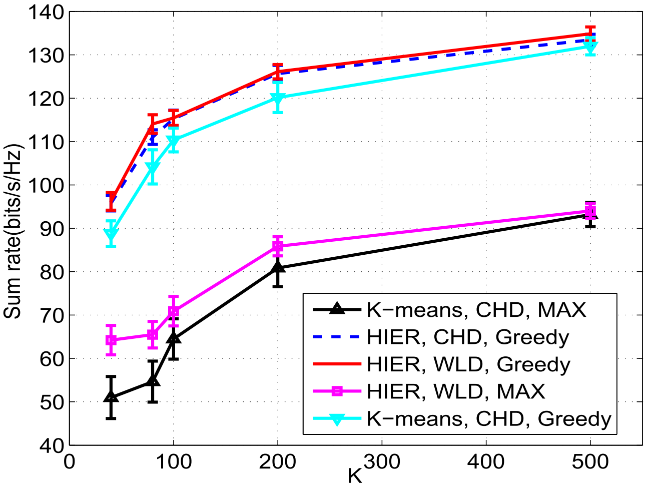

FIGURE 11. System sum rate versus the number of users when $M = 100$.

Fig. 11 presents the sum rate comparison of our proposed schemes with the scheme proposed in [10]. We can see that all the proposed schemes outperform the scheme in [10]. In particular, hierarchical clustering greedy user selection with weighted likelihood has the highest system throughput. Hierarchical clustering greedy user selection with chordal distance has slightly lower throughput than the highest one. Hierarchical clustering MAX user scheduling with weighted likelihood and K-means clustering greedy user selection with chordal distance both have higher throughputs than the scheme in [10]. We find that greedy user scheduling has a greater impact than the user grouping methods on the system throughput. This is because no matter how the groups are formed, greedy user scheduling has the direct impact on the throughput and could always select the users who benefit the throughput performance most. It is also interesting to see that, as the number of users increases, the gap between the K-means and hierarchical clustering curves narrows. This is because with a large number of users, different grouping schemes tend to produce similar user grouping results.

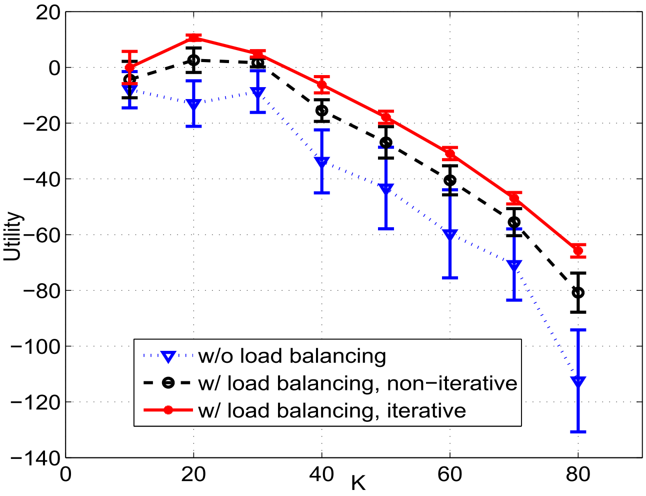

FIGURE 12. Total utility versus the number of users, for user grouping with joint group load balancing and precoding design when $M = 100$.

For user grouping with load balancing, we set $p = 20$ dB. Fig. 12 plots the resulted utility metric of problem (20) solved by Algorithm 5. Note that the negative values of utility are resulted from the log function of achievable rate over the number of group members. We can see that the proposed scheme outperforms the scheme without considering load balancing even with one iteration. The proposed iterative load balancing scheme could achieve even higher total utility.

In addition to the total utility, we are also interested in the number of users in each group. Note that the average number of users in each group is not very helpful, since in each simulation run, the number of users in each group is random. Averaging over these random numbers is approximately K/G for every group. So we just look at one particular simulation, which is depicted in Fig. 13. The total number of user is $K = 40$ in the simulation. We can see that the number of users is $\{14, 3, 7, 10, 2, 4\}$ for groups $1 - 6$ without considering load balancing. The number of users is $\{11, 6, 7, 5, 4, 7\}$ for the non-iterative load balancing scheme and $\{8, 9, 7, 5, 5, 6\}$ for the load balancing scheme iteratively executed. So the

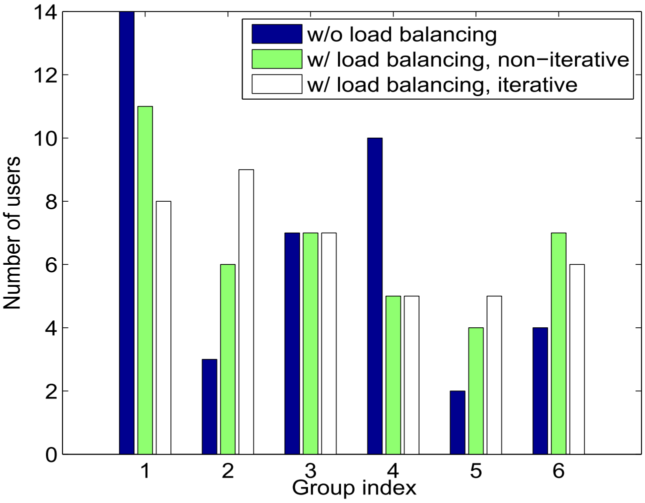

FIGURE 13. Group sizes for user grouping with joint group load balancing and precoding design when $M = 100$. From left to right of each stack, the bars are for the scheme without load balancing, the scheme with load balancing non-iterative and the scheme with load balancing iterative, respectively.

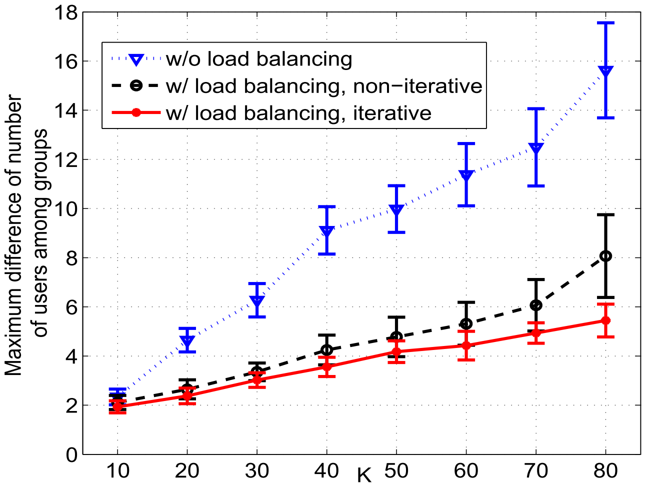

FIGURE 14. Maximum difference of number of users among groups versus the number of users, for user grouping with joint group load balancing and precoding design when $M = 100$ and $G = 6$.

difference between the most loaded group and the least loaded group is 7 for the proposed non-iterative scheme, only 4 for the proposed iterative scheme, but 12 for the scheme without considering load balancing.

Fig. 14 depicts the maximum difference of number of users among all groups. The number of groups G is set to be 6. We can see that when $K = 10$, the maximum differences are 2.33, 2.11, and 1.93 for the scheme without load balancing, with load balancing but non-iterative, with load balancing and iterations, respectively. When $K = 40$, the numbers become 9.11, 4.24, and 3.56. When $K = 80$, the maximum differences are 15.62, 8.07, and 5.44. Therefore, the proposed scheme strikes a much better balance as the users are more evenly distributed among all the groups.

VIII. CONCLUSIONS

In this paper, we have studied the user grouping and scheduling problems based on a two-stage precoding framework for FDD massive MIMO systems. We have proposed weighted likelihood similarity measure, subspace projection based similarity measure, Fubini Study based similarity measure, hierarchical clustering, and K-medoids clustering for user grouping. We have also proposed a dynamic user scheduling scheme and a user grouping algorithm to achieve load balancing and user fairness for FDD massive MIMO systems. The efficacy of the proposed schemes has been validated with analysis and simulations.

REFERENCES

[1] E. Larsson, F. Tufvesson, O. Edfors, and T. Marzetta, “Massive MIMO for next generation wireless systems,” *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 185–195, Feb. 2014.

[2] F. Rusek *et al.*, “Scaling up MIMO: Opportunities and challenges with very large arrays,” *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 40–60, Jan. 2013.

[3] C. Shepard *et al.*, “Argos: Practical many-antenna base stations,” in *Proc. 18th Annu. Int. Conf. MobiCom*, Istanbul, Turkey, Aug. 2012, pp. 53–64.

[4] C. Shepard, H. Yu, and L. Zhong, “ArgosV2: A flexible many-antenna research platform,” in *Proc. 19th Ann. Int. Conf. MobiCom*, Miami, FL, USA, Sep. 2013, pp. 163–165.

[5] J. Jose, A. Ashikhmin, T. L. Marzetta, and S. Vishwanath, “Pilot contamination and precoding in multi-cell TDD systems,” *IEEE Trans. Wireless Commun.*, vol. 10, no. 8, pp. 2640–2651, Aug. 2011.

[6] F. Fernandes, A. Ashikhmin, and T. L. Marzetta, “Inter-cell interference in noncooperative TDD large scale antenna systems,” *IEEE J. Sel. Areas Commun.*, vol. 31, no. 2, pp. 192–201, Feb. 2013.

[7] J. Hoydis, K. Hosseini, S. ten Brink, and M. Debbah, “Making smart use of excess antennas: Massive MIMO, small cells, and TDD,” *Bell Labs Tech. J.*, vol. 18, no. 2, pp. 5–21, Sep. 2013.

[8] Wikipedia. *List of LTE Networks*. [Online]. Available: http://en.wikipedia.org/wiki/List_of_LTE_networks, accessed Sept. 3, 2014.

[9] A. Adhikary, J. Nam, J.-Y. Ahn, and G. Caire, “Joint spatial division and multiplexing—The large-scale array regime,” *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6441–6463, Oct. 2013.

[10] A. Adhikary and G. Caire. “Joint spatial division and multiplexing: Opportunistic beamforming and user grouping,” *IEEE J. Sel. Topics Signal Process.*, vol. PP, no. 99, p. 1, Mar. 2014, doi: 10.1109/JSTSP.2014.2313808.

[11] Y. Xu, G. Yue, N. Prasad, S. Rangarajan, and S. Mao, “User grouping and scheduling for large scale MIMO systems with two-stage precoding,” in *Proc. IEEE ICC*, Sydney, Australia, Jun. 2014, pp. 5208–5213.

[12] J. Choi, D. Love, and P. Bidigare, “Downlink training techniques for FDD massive MIMO systems: Open-loop and closed-loop training with memory,” *IEEE J. Sel. Topics Signal Process.*, to be published.

[13] J. Choi, Z. Chance, D. J. Love, and U. Madhow, “Noncoherent trellis coded quantization: A practical limited feedback technique for massive MIMO systems,” *IEEE Trans. Commun.*, vol. 61, no. 12, pp. 5016–5029, Dec. 2013.

[14] S. Noh, M. Zoltowski, Y. Sung, and D. Love, “Pilot beam pattern design for channel estimation in massive MIMO Systems,” *IEEE J. Sel. Topics Signal Process.*, vol. PP, no. 99, p. 1, May 2014, doi: 10.1109/JSTSP.2014.2327572.

[15] H. Noh, Y. Kim, J. Lee, and C. Lee, “Codebook design of generalized space shift keying for FDD massive MIMO systems in spatially correlated channels,” *IEEE Trans. Veh. Technol.*, vol. PP, no. 99, p. 1, May 2014, doi: 10.1109/TVT.2014.2324822.

[16] X. Rao and V. K. N. Lau, “Distributed compressive CSIT estimation and feedback for FDD multi-user massive MIMO systems,” *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3261–3271, Jun. 2014.

[17] J. Chen and V. Lau, “Two-tier precoding for FDD multi-cell massive MIMO time-varying interference networks,” *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1230–1238, Jun. 2014, doi: 10.1109/JSAC.2014.2328391.

[18] D. Bethanabhotla, O. Y. Bursalioglu, H. C. Papadopoulos, and G. Caire, ''User association and load balancing for cellular massive MIMO,'' in *Proc. IEEE Inf. Theory Appl. Workshop*, San Diego, CA, USA, Feb. 2014, pp. 1–10.

[19] T. L. Marzetta, ''Noncooperative cellular wireless with unlimited numbers of base station antennas,'' *IEEE Trans. Wireless Commun.*, vol. 9, no. 11, pp. 3590–3600, Nov. 2010.

[20] C. Shepard, N. Anand, and L. Zhong, ''Practical performance of MU-MIMO precoding in many-antenna base stations,'' in *Proc. Workshop Cellular Netw., Oper., Challenges, Future Design (CellNet)*, Taipei, Taiwan, Jun. 2013, pp. 13–18.

[21] Q. Ye, B. Rong, Y. Chen, M. Al-Shalash, C. Caramanis, and J. G. Andrews, ''User association for load balancing in heterogeneous cellular networkss,'' *IEEE Trans. Wireless Commun.*, vol. 12, no. 6, pp. 2706–2716, Jun. 2013.

YI XU (S'11) received the M.S. degree in electronic engineering from Tsinghua University, Beijing, China, in 2010, and the B.S. degree in electronic information engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2007. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL, USA. His research interests include optimization, game theory, MIMO, OFDM, IDMA, and cognitive radio networks.

GUOSEN YUE (S'03–M'04–SM'09) received the B.S. degree in physics and the M.S. degree in electrical engineering from Nanjing University, Nanjing, China, in 1994 and 1997, respectively, and the Ph.D. degree in electrical engineering from Texas A&M University, College Station, TX, USA, in 2004. He was a Senior Research Staff with the Department of Mobile Communications and Networking Research, NEC Laboratories America, Princeton, NJ, USA, where he conducted research on broadband wireless systems and mobile networks. His research interests are in the general areas of wireless communications and signal processing. In 2013, he joined Broadcom Corporation, Matawan, NJ, USA, as a System Design Scientist. He serves as an Associate Editor of the IEEE Transactions on Wireless Communications. He has served as an Associate Editor of *Research Letters in Communications*, the Guest Editor of *EURASIP Journal of Wireless Communication and Networking* special issue on interference management, Elsevier's *Physical Communication* special issue on signal processing and coding. He served as the Symposium Co-Chair of the 2010 IEEE International Conference on Communications, the Track Co-Chair of the 2008 IEEE International Conference on Computer Communication and Networks, and a Steering Committee Member of the 2009 IEEE Radio and Wireless Symposium.

SHIWEN MAO (S'99–M'04–SM'09) received the Ph.D. degree in electrical and computer engineering from Polytechnic University, Brooklyn, NY, USA, in 2004. He is a McWane Associate Professor with the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL, USA.

His research interests include wireless networks and multimedia communications, with current focus on cognitive radio, small cells, 60-GHz millimeter-wave networks, free-space optical networks, and smart grid. He is on the Editorial Board of the IEEE Transactions on Wireless Communications, the IEEE Internet of Things Journal, the IEEE Communications Surveys and Tutorials, Elsevier's *Ad Hoc Networks* journal, and Wiley's *International Journal on Communication Systems*. He serves as the Vice Chair of the Letters of Multimedia Communications Technical Committee (MMTC) and the IEEE Communications Society for 2014–2016, and served as the Director of MMTC E-Letter from 2012 to 2014. He serves as the Technical Program Vice Chair of Information Systems (EDAS) of the 2015 IEEE Conference on Computer Communications, Symposium Co-Chair for many conferences, including the IEEE International Conference on Communications (ICC), the IEEE Global Communications Conference, the IEEE International Conference on Computer Communication and Networks, the IEEE International Conference on Industrial Technology and Southeastern Symposium on System Theory, among others, a Steering Committee Member of the IEEE International Conference on Multimedia and Expo (2014–2016) and AdhocNets, and in various roles in the Organizing Committees of many conferences.

Dr. Mao is a Distinguished Lecturer of the IEEE Vehicular Technology Society–Class 2014. He was a recipient of the 2013 IEEE ComSoc MMTC Outstanding Leadership Award and the NSF CAREER Award in 2010. He was also a co-recipient of the IEEE ICC 2013 Best Paper Award and the 2004 IEEE Communications Society's Leonard G. Abraham Prize in the field of communications systems.