User Association in Massive MIMO HetNets

Yi Xu, *Student Member, IEEE*, and Shiwen Mao, *Senior Member, IEEE*

***Abstract*—Massive multiple-input–multiple-output (MIMO) and small cell are both recognized as key technologies for the future fifth-generation wireless systems. In this paper, we investigate the problem of user association in a heterogeneous network (HetNet) with massive MIMO and small cells, where the macro base station (BS) is equipped with a massive MIMO, and the picocell BSs are equipped with regular MIMOs. We first develop centralized user association algorithms with proven optimality, considering various objectives such as rate maximization, proportional fairness, and joint user association and resource allocation. We then develop a repeated game model, which leads to distributed user association algorithms with proven convergence to the Nash equilibrium. We demonstrate the efficacy of these optimal schemes by comparing with several greedy algorithms through simulations.**

***Index Terms*—Game theory, heterogeneous networks (HetNets), massive multiple input multiple output (MIMO), small cells, unimodularity, user association.**

I. Introduction

EXISTING and future wireless networks are facing the grand challenge of a 1000-time increase in mobile data in the near future [1]. To boost wireless capacity, two technologies have gained most attention from both industry and academia. The first one is massive multiple input multiple output (MIMO; a.k.a., large-scale MIMO, full-dimension MIMO, or hyper MIMO) [2], [3]. The idea is to equip a base station (BS) with hundreds, thousands, or even tens of thousands of antennas, hereby providing an unprecedented level of degrees of freedom (DoFs) for mobile users. The massive MIMO concept has been successfully demonstrated in recent studies [4], [5]. The second technology is small cell, which is reshaping the future of cellular networks [7]. A great benefit of deploying small cells is that the distance of the user–BS link can be greatly shortened, leading to reduced transmit power, higher data rate, enhanced coverage, and better spatial reuse of spectrum. Both massive MIMO and small cells are recognized as key technologies of the future fifth-generation wireless systems [6].

In this paper, we consider a heterogeneous network (HetNet) with massive MIMO and small cells. Specifically, we consider a HetNet where the macrocell BS (MBS) is equipped with a massive number of antennas and the picocell BSs (PBSs) are equipped with a regular amount of antennas. To fully harvest the benefits promised by massive MIMO and small cells in an integrated HetNet system, it is critical to investigate the user association problem, i.e., how to assign active users to the BSs such that the system-wide capacity can be maximized and users' experience can be enhanced.

There are already several recent studies moving forward in this direction. In [8]–[11], the authors considered the problem of user association in massive MIMO systems operated in the frequency-division duplexing mode. These papers focused on a macrocell without small cells. In [12], user association in time-division duplexing (TDD) massive MIMO system was addressed, where fractional user association was allowed. Bayat *et al.* in [13] modeled the problem of user association in a femtocell HetNet as a dynamic matching game and derived the optimal user association. In [14], near-optimal user association schemes were proposed for HetNet with WiFi and conventional cellular networks. However, massive MIMO was not considered in these studies. In [15], the authors investigated the problem of user association with conventional MIMO BSs and proposed a simple bias-based selection criterion to approximate more complex selection rules. Björnson *et al.* [16] considered improving energy efficiency without sacrificing the quality of service of users in a massive MIMO and small cell HetNet.

Motivated by these interesting studies, we consider the user association problem in a TDD massive MIMO HetNet in this paper. We take into consideration the practical constraints, such as the limited load capacity at each BS, without allowing fractional user association. The main goal is to maximize the system capacity while enhancing user experience. More specifically, this paper contains two parts: centralized user association and distributed user association. For *centralized user association*, we investigate the problems of rate maximization, rate maximization with proportional fairness, and joint resource allocation and user association. We prove the unimodularity of our formulated problem and develop optimal user association algorithms to the problems of rate maximization and rate maximization with proportional fairness. We then propose a series of primal decomposition and dual decomposition algorithms to solve the problem of joint resource allocation and user association and prove the optimality of the proposed scheme. For *distributed user association*, we model the behavior and interaction between the service provider (who owns the BSs) and users as repeated games. We consider two types of operations: 1) the service provider sets the price and the users decide which BS to connect to and 2) the users bid for connection opportunities. We prove that, in both cases, the proposed algorithms converge to the respective Nash equilibrium (NE).

In the remainder of this paper, Section II introduces the system model and preliminaries. Optimal centralized and distributed user association schemes are presented in Sections III

Manuscript received December 20, 2014; revised May 16, 2015 and June 29, 2015; accepted August 30, 2015. Date of publication September 22, 2015; date of current version March 10, 2017. This work was supported in part by the National Science Foundation under Grant CNS-1247955 and Grant CNS-1320664 and in part by the Wireless Engineering Research and Education Center at Auburn University.

The authors are with the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849 USA (e-mail: YZX0010@tigermail.auburn.edu; smao@ieee.org).

Digital Object Identifier 10.1109/JSYST.2015.2475702

and IV, respectively. Section V presents the simulation study, and Section VI concludes this paper. Throughout this paper, a boldface upper (lower) case symbol denotes a matrix (vector), a normal symbol denotes a scalar, $(\cdot)^H$ denotes the Hermitian of a matrix, and $(\cdot)^T$ denotes the transpose of a matrix.

II. System Model and Preliminaries

The system considered in this paper consists of K users and J BSs inside a macro cell. There is a single MBS with a massive MIMO and $(J-1)$ PBSs, each with a regular amount of antennas. Following [19], the channel model we consider is $h_{j,k,n} = g_{j,k,n} l_{j,k}$, where $h_{j,k,n}$ is the channel of antenna n at BS j to user k, $g_{j,k,n}$ represents the small-scale fading coefficient between antenna n of BS j and user k, and $l_{j,k}$ stands for the large-scale fading coefficient between BS j and user k. To obtain the channel vector $\mathbf{h}_{j,k}$ between BS j and user k, we could concatenate all the channel coefficients from all the antennas of BS j as $\mathbf{h}_{j,k} = [h_{j,k,1}, h_{j,k,2}, \ldots, h_{j,k,n}]$. For the channel coefficient matrix of signals transmitted from BS j, we could obtain it by concatenating channel vector $\mathbf{h}_{j,k}$ as $\mathbf{H}_j = [\mathbf{h}_{j,1}, \mathbf{h}_{j,2}, \ldots, \mathbf{h}_{j,k}]$.

Let $\mathbf{y}_j$ denote the signals received by the users connecting to BS j, $\mathbf{W}_j$ the precoding matrix of BS j, and $\mathbf{d}_j$ the data sent from BS j. We have

$$\mathbf{y}_j = \mathbf{H}_j \mathbf{W}_j \mathbf{d}_j + \mathbf{n}_j \tag{1}$$

where $\mathbf{n}_j$ is the zero-mean circulant symmetric complex Gaussian noise vector.

Each active user has the options to connect to either the MBS or a PBS. For a user k, define user association index variable x_{k_j} as

$$x_{k_j} = \begin{cases} 1, & \text{if user } k \text{ is connected to BS } j \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Let the achievable rate of user k connecting to BS j be R_{k_j}. We define $\eta_{k_j} = x_{k_j} R_{k_j}$ and the overall data rate of user k to be η_k, where

$$\eta_k = \sum_j \eta_{k_j} = \sum_j x_{k_j} R_{k_j}. \tag{3}$$

For users connecting to a massive MIMO BS j (i.e., the MBS), their achievable rate can be approximated with the following deterministic rate [12]:

$$R_{k_j} = \log\left(1 + \frac{M_j - L_j + 1}{L_j} \frac{P_j l_{j,k}}{1 + \sum_{j' \neq j} P_{j'} l_{j',k}}\right) \tag{4}$$

where M_j is the number of antennas at the BS, L_j is the prefixed load parameter of the BS indicating how many users it could serve, and P_j is the transmit power from the MBS. Note that there is no small-scale fading factor in (4). This approximation has been proven to be accurate in [12].

As shown in [17], there are various sources of interferences in a HetNet. However, among the PBSs, intercell interference coordination could be effectively performed according to [18]. We hereby ignore the intercell interference and only consider the intracell interference. The achievable rate of a user k connecting to PBS j can be then represented as follows:

$$\widetilde{R}_{k_j} = \log\left(1 + \frac{P_j \left|\mathbf{h}_{j,k}^H \mathbf{w}_{j,k}\right|^2}{1 + \sum_{k' \neq k} P_j \left|\mathbf{h}_{j,k}^H \mathbf{w}_{j,k'}\right|^2}\right) \tag{5}$$

where $\mathbf{w}_{j,k}$ is the kth column of BS j's precoding matrix $\mathbf{W}_j$, and $|\cdot|$ represents the absolute value. There are many precoding designs for conventional MIMO BSs, such as matched filter (MF) precoding, zero-forcing (ZF) precoding, and regularized ZF precoding [19]. Without loss of generality, we adopt MF precoding with $\mathbf{W}_j = (1/\sqrt{\varphi})\mathbf{H}_j^H$, where φ is a power normalization factor. The signal received by all the users connecting to PBS j can be rewritten as follows:

$$\mathbf{y}_j = \begin{pmatrix} h_{j,1}^H h_{j,1} d_1 + h_{j,1}^H h_{j,2} d_2 + \cdots + h_{j,1}^H h_{j,k} d_k \\ h_{j,2}^H h_{j,1} d_1 + h_{j,2}^H h_{j,2} d_2 + \cdots + h_{j,2}^H h_{j,k} d_k \\ \cdots \\ h_{j,k}^H h_{j,1} d_1 + h_{j,k}^H h_{j,2} d_2 + \cdots + h_{j,k}^H h_{j,k} d_k \end{pmatrix}. \tag{6}$$

Thus, the achievable rate for user k with regard to PBS j can be obtained as follows:

$$\eta_{k_j} = \log\left(1 + \frac{P_j \left|x_{k_j} h_{j,k}^H h_{j,k}\right|^2}{1 + \sum_{k' \neq k} P_j \left|x_{k'_j} h_{j,k}^H h_{j,k'}\right|^2}\right). \tag{7}$$

III. Centralized User Association

Here, we consider the problem of centralized user association. We assume that the BSs have all the channel state information (CSI) via uplink training. We adopt the following two utility functions for each user k with achievable rate η_k:

$$\mathcal{U}(\eta_k) = \eta_k \qquad \mathcal{U}(\eta_k) = \begin{cases} \log(\eta_k), & \text{if } \eta_k > 0 \\ 0, & \text{if } \eta_k = 0. \end{cases} \tag{8}$$

Maximizing the sum of the first utility yields the maximization of the sum rate. Maximizing the sum of the second utility yields the maximization of the geometric mean rate (i.e., achieving proportional fairness). Our goal is to maximize the system utility by configuring the user–BS association.

A. Sum Rate Maximization

We first investigate the problem of maximizing the system sum rate, i.e., with utility function $\mathcal{U}(\eta_k) = \eta_k$. The problem can be formulated as follows:

$$\begin{aligned} \mathbf{P1-1}: \max_{\{x_{k_j}\}} \quad & \sum_{k=1}^{K} \eta_k \\ \text{s.t.} \quad & \sum_k x_{k_j} \leq L_j \leq M_j, \quad j = 1, 2, \ldots, J \\ & \sum_j x_{k_j} \leq 1, \quad k = 1, 2, \ldots, K \\ & \text{Constraints } (2), (3), (4), (7). \end{aligned} \tag{9}$$

Note that the first constraint requires the number of users connecting to a BS to be no more than its prefixed load L_j, which should, in turn, be no more than the number of antennas it has. This is because, theoretically, BS j can provide at most M_j DoFs. Assuming that the L_j are already chosen to satisfy $L_j \leq M_j$, we drop this constraint in the remainder of this paper. The second constraint simply enforces that each user can connect to at most one BS at a time.

A key observation is that, due to the binary x_{k_j}, if $x_{k_j} = 0$, $\eta_{k_j} = 0$; if $x_{k_j} = 1$, η_{k_j} is a logarithm function. Thus, we can pull x_{k_j} out of the $\log()$ function and rewrite (7) as

$$\eta_{k_j} = x_{k_j} \log\left(1 + \frac{P_j \left|h_{j,k}^H h_{j,k}\right|^2}{1 + \sum_{k' \neq k} P_j \left|x_{k'_j} h_{j,k}^H h_{j,k'}\right|^2}\right). \quad (10)$$

The $\widetilde{R}_{k_j}$ term in (5) can be redefined as

$$\widetilde{R}_{k_j} = \log\left(1 + \frac{P_j \left|h_{j,k}^H h_{j,k}\right|^2}{1 + \sum_{k' \neq k} P_j \left|x_{k'_j} h_{j,k}^H h_{j,k'}\right|^2}\right). \quad (11)$$

In (11), $\widetilde{R}_{k_j}$ also depends on other users' choices $x_{k'_j}$, for all $k \neq k'$, as well. To make the problem tractable, we adopt the worst case approximation by assuming that the users within the coverage of BS j (denoted by $\mathcal{G}_j$) all connect to BS j with perfect channels. This way, (11) can be approximated as

$$\widetilde{R}_{k_j} = \log\left(1 + \frac{P_j \left|h_{j,k}^H h_{j,k}\right|^2}{1 + (|\mathcal{G}_j| - 1) P_j}\right). \quad (12)$$

Note here that $|Z|$ stands for the absolute value. It also represents the cardinality if Z is a set.

Define auxiliary variables c_{k_j} as follows:

$$c_{k_j} = \begin{cases} R_{k_j} \text{ in (4)}, & \text{if BS } j \text{ is the MBS} \\ \widetilde{R}_{k_j} \text{ in (12)}, & \text{if BS } j \text{ is a PBS.} \end{cases} \quad (13)$$

The sum rate maximization problem can be reformulated as

$$\begin{aligned} \mathbf{P1-2}: \max_{\{x_{k_j}\}} \quad & \sum_{k=1}^{K} \sum_{j=1}^{J} x_{k_j} c_{k_j} \\ \text{s.t.} \quad & \sum_{k} x_{k_j} \leq L_j, \quad j = 1, 2, \ldots, J \\ & \sum_{j} x_{k_j} \leq 1, \quad k = 1, 2, \ldots, K \\ & \text{Constraints } (2), (13). \end{aligned} \quad (14)$$

Since the variables x_{k_j} are binary, problem $\mathbf{P1-2}$ falls into the category of *multiple knapsack problems*, which is one of the 21 NP-complete problems of Karp [20]. Although a greedy algorithm could be developed to compute suboptimal solutions, we show that problem $\mathbf{P1-2}$ can actually be optimally solved by taking advantage of its special structure.

Let $\mathbf{X}$ be a matrix with entries x_{k_j}, $k = 1, 2, \ldots, K$, $j = 1, 2, \ldots, J$. We first convert $\mathbf{X}$ to a vector $\mathbf{x}$ by concatenating the rows of $\mathbf{X}$ and taking a transpose as $\mathbf{x} = [x_{1_1} x_{2_1}, \ldots, x_{K_1}, \ldots, x_{1_J}, \ldots, x_{K_J}]^T$ and simplify the notation as $\mathbf{x} = [x_1 x_2, \ldots, x_{K_J}]^T$. We then apply the same conversion to the matrix comprising c_{k_j} and obtain vector $\mathbf{c}$. Problem $\mathbf{P1-2}$ can be rewritten as

$$\begin{aligned} \mathbf{P1-3}: \max_{\mathbf{x}} \quad & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \quad & \sum_{k=1}^{K} x_{(j-1)K+k} \leq L_j, \quad j = 1, 2, \ldots, J \\ & \sum_{j=1}^{J} x_{k+(j-1)K} \leq 1, \quad k = 1, 2, \ldots, K \\ & \text{Constraints } (2), (13). \end{aligned} \quad (15)$$

Ignore constraints (2) and (13). Define $\mathbf{A}$ as the constraint matrix of problem $\mathbf{P1-3}$, with entries being the coefficients of the first and second constraints.

Definition 1: A matrix $\mathbf{A}$ is called totally unimodular if the determinant of every square submatrix of $\mathbf{A}$ is either 0, $+1$, or -1 [21].

Lemma 1: The constraint matrix $\mathbf{A}$ of problem $\mathbf{P1-3}$ is totally unimodular.

Proof: The constraint matrix in problem $\mathbf{P1-3}$ is of the following form:

$$\mathbf{A} = \begin{pmatrix} 1\,1\cdots1 & 0\,0\cdots0 & 0\,0\cdots0 \\ 0\,0\cdots0 & 1\,1\cdots1 & 0\,0\cdots0 \\ \vdots & \vdots & \vdots \\ 0\,0\cdots0 & 0\,0\cdots0 & 1\,1\cdots1 \\ 1\,0\cdots0 & 1\,0\cdots0 & 1\,0\cdots0 \\ 0\,1\cdots0 & 0\,1\cdots0 & 0\,1\cdots0 \\ \ddots & \ddots & \ddots \\ 0\,0\cdots1 & 0\,0\cdots1 & 0\,0\cdots1 \end{pmatrix}. \quad (16)$$

We can divide $\mathbf{A}$ into blocks as follows:

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_1 & \mathbf{A}_2 & \cdots & \mathbf{A}_J \\ \mathbf{B}_1 & \mathbf{B}_2 & \cdots & \mathbf{B}_J \end{pmatrix} \quad (17)$$

where each $\mathbf{A}_j, j \in [1, J]$ is a submatrix of $\mathbf{A}$ of size $J \times K$, and each $\mathbf{B}_j, j \in [1, J]$ is an identity matrix of size $K \times K$.

Let $\mathbf{S}_n$ denote an arbitrary square submatrix of matrix $\mathbf{A}$ of size n. For any submatrix of $\mathbf{A}$ of size $n = 1$, it is trivial to see that the determinant of this submatrix is either 0 or $+1$. Thus, we only need to consider the case where the size of the square submatrix is greater than or equal to 2.

Case 1) $\mathbf{S}_n$ is taken entirely from one of the submatrices $\mathbf{A}_j$ or $\mathbf{B}_j$, $j \in [1, J]$. We can see from the structure that at least one row of $\mathbf{A}_j$ is all zero. Thus, if the square submatrix is entirely taken from $\mathbf{A}_j$, the determinant of the submatrix is zero. Since matrix $\mathbf{B}_j$, for all j, is simply an identity matrix, it is straightforward that the determinant of any square submatrix of $\mathbf{B}_j$ is either 0 or $+1$.

Case 2) $\mathbf{S}_n$ is not entirely taken from any one of the submatrices $\mathbf{A}_j$ or $\mathbf{B}_j$, $j \in [1, J]$. In this case, the square submatrix must be taken from $2n$ $(n = 1, \ldots, J)$ submatrices of the submatrix set $(\mathbf{A}_j \cup \mathbf{B}_j, j \in 1, \ldots, J)$. We next proceed with our proof by applying induction method.

For the base case $n = 1$, the square submatrix to be examined is of size 2. Since the entries can be only 0 or $+1$, the determinant can be only $0, +1$, or -1. Now, assuming that any square submatrix of size $(n-1)$ has determinant $0, +1$, or -1, we need to check if the same conclusion holds for any square submatrix of size n.

We first notice that each column of $\mathbf{A}$ has exactly two $+1$ s. Moreover, exactly one of them is in $\mathbf{A}_j$, and the other is in $\mathbf{B}_j$. Let $q^* = \arg\min_q \sum_i \mathbf{S}_{n_{i,q}}$, where $\mathbf{S}_{n_{i,q}}$ is the (i, q)th entry of $\mathbf{S}_n$. That is, column q^* has the minimum number of 1 s among all the columns of $\mathbf{S}_n$.

Let $\zeta_{q^*} = \min_q \sum_i \mathbf{S}_{n_{i,q}}$. ζ_{q^*} can only be 0, 1, or 2. If $\zeta_{q^*} = 0$, then all the entries of the q^*th column of $\mathbf{S}_n$ are 0, which results in $\det(\mathbf{S}_n) = 0$, where $\det$ is short for determinant.

If $\zeta_{q^*} = 1$, then we could calculate $\det(\mathbf{S}_n)$ by expanding the q^*th column and obtain $\det(\mathbf{S}_n) = \det(\mathbf{S}_{(n-1)})$. Since $\det(\mathbf{S}_{(n-1)})$ is 0, 1, or -1 by our induction hypothesis, we conclude that $\det(\mathbf{S}_n)$ is 0, 1, or -1.

If $\zeta_{q^*} = 2$, we could first negate all the entries taken from $\mathbf{B}_j$ and then add all the rows in $\mathbf{B}_j$ to any nonzero row in $\mathbf{A}_j$. After this procedure, if that nonzero row in $\mathbf{A}_j$ is still nonzero, add that row to any other nonzero row in $\mathbf{A}_j$. Repeat this process until we get a zero row in $\mathbf{A}_j$. This process always yields an all-zero row because we have equal number of $+1$ s in $\mathbf{A}_j$ and $\mathbf{B}_j$. Since any basic row operation does not change the determinant and we finally get an all-zero row, we have $\det(\mathbf{S}_n) = 0$. That completes our induction. ■

Fact 1: For a linear programming problem, if its constraint matrix satisfies total unimodularity, then it has all integral vertex solutions [21].

Fact 2: For a linear programming problem, if it has feasible optimal solutions, then at least one of them occurs at a vertex of the polyhedron defined by its constraints [22].

Given the facts and Lemma 1, we have the following theorem. The proof is straightforward and omitted.

Theorem 1: The optimal solution of problem $\mathbf{P1}$ can be obtained by solving a relaxed problem where the binary variables x_{k_j} are allowed to take real values between [0,1].

With Theorem 1, we can obtain the optimal solution of $\mathbf{P1}$ by solving the relaxed problem, termed $\mathbf{NP1}$, using an LP solver [21] with an average complexity of $\mathcal{O}(J^2K^2)$.

B. Proportional Fairness

Here, we take proportional fairness among user achievable rates into consideration. The goal is to maximize $\sum_k \mathcal{U}(\eta_k)$. Since if $\eta_k = 0$ we define $\mathcal{U}(\eta_k) = 0$, and negative utility will be excluded from the maximization, we actually maximize $\sum_k \max\{\log(\sum_{j=1}^J x_{k_j} c_{k_j}), 0\}$. The problem can be formulated as follows:

$$\begin{aligned}
\mathbf{P2-1}: \max_{\{x_{k_j}\}} \quad & \sum_{k=1}^{K} \max\left\{\log\left(\sum_{j=1}^{J} x_{k_j} c_{k_j}\right), 0\right\} \\
\text{s.t.} \quad & \sum_{k=1}^{K} x_{k_j} \le L_j, \quad j = 1, 2, \ldots, J \\
& \sum_{j=1}^{J} x_{k_j} \le 1, \quad k = 1, 2, \ldots, K \\
& \text{Constraints (2) (13).}
\end{aligned} \tag{18}$$

Problem $\mathbf{P2-1}$ is a nonlinear integer programming problem, which is generally NP-hard. To get a better understanding of the problem, we examine its equivalent problem as follows:

$$\begin{aligned}
\mathbf{P2-2}: \max_{x_{k_j}} \prod_{k=1}^{K} \left(\sum_{j=1}^{J} x_{k_j} c_{k_j}\right) \\
\text{s.t. same constraints as in (18).}
\end{aligned} \tag{19}$$

Problem $\mathbf{P2-2}$ is a geometric programming problem with binary variables. The objective function is a polynomial function with J^K terms. Conventionally, to solve geometric programming problems, we need to introduce new variables such as $y = \log(x)$ so that geometric programming can be solved via convex programming. However, here, x_{k_j} are binary. Since $\log(0) = -\infty$, we could not apply these techniques. Another heuristic scheme is to first sort these J^K coefficients and then find the L_j largest coefficients for each BS. However, even sorting these J^K coefficients could be computationally prohibitive even for a small system, which requires $\mathcal{O}(J^K \log(J^K))$ operations.

A key observation about the logarithm function is that $\log(\sum_i \tau_i) \le \sum_i \log(\tau_i)$, for all $\tau_i \ge 2$. Therefore, in practice,[1] the optimal value of problem $\mathbf{P2-1}$ is upper bounded by that of the following problem:

$$\begin{aligned}
\mathbf{NP2}: \max_{x_{k_j}} \sum_{k=1}^{K} \sum_{j=1}^{J} x_{k_j} \log(c_{k_j}) \\
\text{s.t. same constraints as in (18).}
\end{aligned} \tag{20}$$

We have the following results for the transformed problems.

Lemma 2: Problems $\mathbf{P2-1}$ and $\mathbf{NP2}$ are equivalent.

Proof: Recall that, if $\eta_k = 0$, we define $\mathcal{U}(\eta_k) = 0$. The second constraint $\sum_{j=1}^J x_{k_j} \le 1$ enforces that each user connect to at most one BS. Consequently, $\max\{\log(\sum_{j=1}^J x_{k_j} c_{k_j}), 0\} = \sum_j x_{k_j} \log(c_{k_j})$. Furthermore, we have $\sum_k \sum_j x_{k_j} \log(c_{k_j}) = \sum_k \max\{\log(\sum_{j=1}^J x_{k_j} c_{k_j}), 0\}$. Therefore, we conclude that problems $\mathbf{P2-1}$ and $\mathbf{NP2}$ are equivalent. ■

[1] Recall that c_{k_j} is the achievable rate of user k connecting to BS j. $c_{k_j} \ge 2$ is generally satisfied in current wireless systems with a sufficiently large bandwidth and sufficiently high transmission power.

Comparing problem $\mathbf{NP2}$ with $\mathbf{P1-2}$, we find that they are equivalent as well. Thus, we can obtain the optimal value of $\mathbf{P2-1}$ by applying the same technique used to solve problem $\mathbf{P1-2}$.

For comparison purposes, we propose two suboptimal greedy algorithms, i.e., Algorithm 1 and Algorithm 2, as benchmarks. They can be directly used for comparison with problem $\mathbf{P1-1}$. Note that there are many other algorithms that can be used to solve this problem, such as genetic algorithm [23] and swarm optimization [24]. However, these techniques have not been applied in the context of mass MIMO systems. We chose simple greedy algorithms to compare their performance with the optimal solutions. To compare with problem $\mathbf{P2-1}$, in Algorithm 1, we need to change steps 7 and 8 as "**while** $\exists j, L_j \neq 0 \;\&\; \max_{k,j} \log(c_{k,j}) > 0$ **do**" and "Find $(k^*, j^*) = \arg\max_{k,j}\{\log(c_{k,j})\}$," respectively. In Algorithm 2, we need to change steps 8 and 9 as "**while** $L_j \neq 0 \;\&\; \max_k \log(c_{k,j}) > 0$" and "Find $(k^*, j) = \arg\max_k \log(c_{k,j})$," respectively.

The complexity of both greedy algorithms is $\mathcal{O}(KJ)$.

Algorithm 1: Greedy Algorithm 1 for User Association

1 Initialize $\mathcal{K} = \{1, 2, \cdots, K\}$, $L_j, \forall j \in \mathcal{J}$ and x_{k_j} to be an all-zero matrix ;
2 **for** $k = 1$ *to* K **do**
3   **for** $j = 1$ *to* J **do**
4   Compute c_{k_j} as in (13) ;
5   **end**
6 **end**
7 **while** $\exists j,\ L_j \neq 0$ **do**
8   Find $(k^*, j^*) = \arg\max_{k,j}\{c_{k,j}\}$;
9   **if** $L_{j^*} \neq 0$ **then**
10     $x_{k^*_{j^*}} = 1$;
11     $L_{j^*} = L_{j^*} - 1$;
12     $\mathcal{K} = \mathcal{K} \backslash k^*$;
13   **end**
14 **end**

Algorithm 2: Greedy Algorithm 2 for User Association

1 Initialize $\mathcal{K} = \{1, 2, \cdots, K\}$, $L_j, \forall j \in \mathcal{J}$ and x_{k_j} to be an all-zero matrix ;
2 **for** $k = 1$ *to* K **do**
3   **for** $j = 1$ *to* J **do**
4   Compute c_{k_j} as in (13) ;
5   **end**
6 **end**
7 **for** $j = 1$ *to* J **do**
8   **while** $L_j \neq 0$ **do**
9   Find $(k^*, j) = \arg\max_k\{c_{k,j}\}$;
10     $x_{k^*_j} = 1$;
11     $L_j = L_j - 1$;
12     $\mathcal{K} = \mathcal{K} \backslash k^*$;
13   **end**
14 **end**

C. Joint Resource Allocation and User Association

Here, we take resource allocation into account. Consider a massive MIMO orthogonal frequency-division multiple access (OFDMA) HetNet. In OFDMA systems, such as long-term evolution, the time–frequency resource is divided into resource blocks (RBs). A typical RB consists of 12 subcarriers (180 kHz) in the frequency domain and seven OFDMA symbols in the time domain (0.5 ms). Thus, the system may have up to several hundreds of RBs. We normalize it to be a unit number. A user k connecting to a BS j gets a portion β_{k_j} of the overall resource. The goal is to maximize the system utility considering both resource allocation and user association.

Considering the logarithm rate utility and defining $\Phi_j = \{k \mid x_{k_j} = 1\}$, the problem is formulated as follows:

$$\begin{aligned}
\mathbf{P3-1}: \ \max_{\{x_{k_j}, \beta_{k_j}\}} \ & \sum_{k=1}^{K} \log\left(\sum_{j=1}^{J} x_{k_j} c_{k_j} \beta_{k_j}\right) \\
\text{s.t.} \ & \sum_{k \in \Phi_j} \beta_{k_j} \leq 1, \quad j = 1, 2, \ldots, J \\
& \sum_{j=1}^{J} x_{k_j} \leq 1, \quad k = 1, 2, \ldots, K \\
& \sum_{k \in \Phi_j} \beta_{k_j} \leq 1, \quad j = 1, 2, \ldots, J \\
& \text{Constraints } (2), (13).
\end{aligned} \tag{21}$$

To solve problem $\mathbf{P3-1}$, we need to do the following: select users for each BS to serve and allocate resources to the associated users at each BS. We next propose a series of primal decomposition and dual decomposition to optimally solve the problem.

Due to binary variables x_{k_j} and real variables β_{k_j}, problem $\mathbf{P3-1}$ is a mixed-integer nonlinear programming problem, which is generally NP-hard. However, next, we propose an algorithm to obtain its optimal solution.

Note that what we actually maximize is $\sum_{k=1}^{K} \max\{\log(\sum_{j=1}^{J} x_{k_j} c_{k_j} \beta_{k_j}), 0\}$. For easier notation, the objective function has been simplified. Since x_{k_j} take binary values and $\sum_{j=1}^{J} x_{k_j} \leq 1$, we have $\sum_{k=1}^{K} \log(\sum_{j=1}^{J} x_{k_j} c_{k_j} \beta_{k_j}) = \sum_{k=1}^{K} \sum_{j=1}^{J} x_{k_j} \log(c_{k_j} \beta_{k_j})$. Thus, problem $\mathbf{P3-1}$ can be reformulated as

$$\begin{aligned}
\mathbf{P3-2}: \ \max_{\{x_{k_j}, \beta_{k_j}\}} \ & \sum_{k=1}^{K} \sum_{j=1}^{J} x_{k_j} \log\left(c_{k_j} \beta_{k_j}\right) \\
& \text{s.t. same constraints as problem } \mathbf{P3-1}.
\end{aligned} \tag{22}$$

The choices of β_{k_j} rely on the values of x_{k_j}. Given these coupled variables, we first apply the primal decomposition method [25] to decompose problem $\mathbf{P3-2}$ into the following two levels of problems. Fixing variables x_{k_j}, we have the *lower level problem* as

$$\begin{aligned}
\max_{\{\beta_{k_j}\}} \ & \sum_{k=1}^{K} \sum_{j=1}^{J} x_{k_j} \log\left(c_{k_j} \beta_{k_j}\right) \\
\text{s.t.} \ & \sum_{k \in \Phi_j} \beta_{k_j} \leq 1, \quad j = 1, 2, \ldots, J.
\end{aligned} \tag{23}$$

When β_{k_j} are fixed, the *higher level problem* (or, the *master problem*) is given by

$$\begin{aligned}
\max_{\{x_{k_j}\}} \ & \sum_{k=1}^{K} \sum_{j=1}^{J} x_{k_j} \log(c_{k_j} \beta_{k_j}) \\
& \text{s.t. same constraints as problem } \mathbf{P2-1}.
\end{aligned} \tag{24}$$

Since there are no couplings among the subproblems, the lower level problem (23) can be further decomposed into L subproblems as follows:

$$\max_{\{\beta_{k_j}\}} \sum_{k=1}^{K} x_{k_j} \log(c_{k_j}\beta_{k_j})$$
$$\text{s.t.} \sum_{k\in\Phi_j} \beta_{k_j} \le 1, \quad j = 1, 2, \ldots, J. \tag{25}$$

Defining Lagrange multiplier λ, the Lagrangian of problem (25) is defined as

$$\mathcal{L} = \sum_{k=1}^{K} x_{k_j} \log(c_{k_j}\beta_{k_j}) + \lambda \left(1 - \sum_{k=1}^{K} \beta_{k_j}\right). \tag{26}$$

Applying Karush–Kuhn–Tucker conditions [26], the optimal solution can be obtained as follows:

$$\beta_{k_j} = \frac{x_{k_j}}{\sum_{k=1}^{K} x_{k_j}}. \tag{27}$$

Substituting (27) into the master problem, the objective function becomes

$$\sum_{k=1}^{K}\sum_{j=1}^{J} x_{k_j} \log\left(\frac{c_{k_j}}{\sum_{k=1}^{K} x_{k_j}}\right). \tag{28}$$

Note that we have dropped one x_{k_j} term in (28), since, due to the definition in (2), we have $(x_{k_j})^2 = x_{k_j}$. Since $\sum_{k=1}^{K} x_{k_j}$ is in the denominator, problem (28) has coupled objectives. The main idea of addressing the coupled objective is to introduce auxiliary variables and additional equality constraints so that the coupling in the objective function is transferred to coupling in the constraint [25]. We thus introduce a new variable, which is defined as

$$\Xi_j = \sum_{k=1}^{K} x_{k_j}. \tag{29}$$

To solve the aforementioned problem, we relax x_{k_j} to a real number in [0, 1]. However, we will show later that, even if we have relaxed the variables, we could still find the optimal solution to the original problem. The relaxed problem to be solved is

$$\max_{\{x_{k_j}\}} \sum_{k=1}^{K}\sum_{j=1}^{J} x_{k_j} \log\left(\frac{c_{k_j}}{\Xi_j}\right)$$
$$\text{s.t.} \quad \Xi_j \le L_j, \quad j = 1, 2, \ldots, J$$
$$\sum_{j=1}^{J} x_{k_j} \le 1, \quad k = 1, 2, \ldots, K$$
$$0 \le x_{k_j} \le 1, \text{ for all } k, j$$
$$\text{Constraints (13), (29).} \tag{30}$$

Problem (30) is a convex optimization problem. Defining Lagrange multipliers for the equality constraints (29), problem (30) can be solved with the dual decomposition method. Alternatively, we propose Algorithm 3 to obtain the optimal solution of problem (30) [10], [11], [27]. In Algorithm 3, $\delta^{(t)}$ is the step size at the tth iteration given by

$$\delta^{(t)} = \frac{\vartheta}{t + \gamma} \tag{31}$$

where ϑ and γ are positive numbers.

Theorem 2: Algorithm 3 optimally solves problem (30).

Proof: Let $\mathbf{x}_k^{(t)}$ denote the solution produced by Algorithm 3 at step t. Let $\partial\mathcal{U}(\mathbf{x}_k^{(t)})$ be the subgradient of the objective function in problem (30) at step t. It can be easily verified that the updated direction in step 16 of Algorithm 3 is the subgradient direction. Since Ξ_j is upper bounded by L_j and K and $\sum_{k=1}^{K} x_{k_j}$ is upper bounded by K, $\partial\mathcal{U}(\mathbf{x}_k^{(t)})$ is also bounded.

Denote $\mathcal{U}_a$ as the final result produced by Algorithm 3 and $\mathcal{U}^*$ as the optimal solution of problem (30). We prove the theorem by contradiction. Assume that $\mathcal{U}_a$ is not optimal. Then there must exist an $\epsilon > 0$ such that $\mathcal{U}_a + 2\epsilon < \mathcal{U}^*$. Then there must be a solution $\hat{\mathbf{x}}_k$ so that

$$\mathcal{U}_a + 2\epsilon < \mathcal{U}(\hat{\mathbf{x}}_k). \tag{32}$$

Let t_0 be sufficiently large so that, for any $t > t_0$, we have

$$\mathcal{U}\left(\mathbf{x}_k^{(t)}\right) \le \mathcal{U}_a + \epsilon. \tag{33}$$

Combining (32) and (33), we have $\mathcal{U}(\mathbf{x}_k^{(t)}) + \epsilon < \mathcal{U}(\hat{\mathbf{x}}_k)$.

Let κ be a positive number that satisfies $\kappa \le \inf\{\|\partial\mathcal{U}(\mathbf{x}_k^{(t)})\|\}$, for all t. It follows that

$$\begin{aligned}
\left\|\mathbf{x}_k^{(t+1)} - \hat{\mathbf{x}}_k\right\|^2 &= \left\|\mathbf{x}_k^{(t)} - \delta^{(t)}\partial\mathcal{U}^{(t)} - \hat{\mathbf{x}}_k\right\|^2 \\
&= \left\|\mathbf{x}_k^{(t)} - \hat{\mathbf{x}}_k\right\|^2 + \left(\delta^{(t)}\right)^2 \left\|\partial\mathcal{U}^{(t)}\right\|^2 \\
&\quad - 2\delta^{(t)} \left(\partial\mathcal{U}^{(t)}\right)^H \left(\mathbf{x}_k^{(t)} - \hat{\mathbf{x}}_k\right) \\
&\ge \left\|\mathbf{x}_k^{(t)} - \hat{\mathbf{x}}_k\right\|^2 + \left(\delta^{(t)}\right)^2 \left\|\partial\mathcal{U}^{(t)}\right\|^2 \\
&\quad - 2\delta^{(t)} \left(\mathcal{U}\left(\mathbf{x}_k^{(t)}\right) - \mathcal{U}(\hat{\mathbf{x}}_k)\right) \\
&\ge \left\|\mathbf{x}_k^{(t)} - \hat{\mathbf{x}}_k\right\|^2 + \left(\delta^{(t)}\right)^2 \kappa^2 + 2\delta^{(t)}\epsilon \\
&\ge \left\|\mathbf{x}_k^{(t)} - \hat{\mathbf{x}}_k\right\|^2 + 2\delta^{(t)}\epsilon \ge \cdots \\
&\ge \left\|\mathbf{x}_k^{(t_0)} - \hat{\mathbf{x}}_k\right\|^2 + 2\epsilon \sum_{j=t_0}^{t} \delta^{(j)}.
\end{aligned} \tag{34}$$

Note that the first inequality is due to the property of subgradient. Thus, we finally have $\|\mathbf{x}_k^{(t+1)} - \hat{\mathbf{x}}_k\|^2 \ge \|\mathbf{x}_k^{(t_0)} - \hat{\mathbf{x}}_k\|^2 + 2\epsilon \sum_{j=t_0}^{t} \delta^{(j)}$, which cannot hold for sufficiently large t. Thus, Algorithm 3 optimally solves problem (30). ■

Algorithm 3: Two Layer Dual Decomposition Algorithm for Optimization Problem (30)

$t = 0$, $\lambda^{(1)} = 0$;
while *not converged* **do**
  $t \leftarrow t + 1$;
  for $k = 1, \cdots, K$ **do**
    for $j = 1, \cdots, J$ **do**
      Compute c_{k_j} as in (13) ;
    end
    Find $j^* = \arg\max_j \left\{ \log(c_{k_j} - \lambda_j^{(t)}) \right\}$;
    Let $x_{k_j}^{(t)} = 0$ for $j \neq j^*$;
    if $\log(c_{k_j} - \lambda_j^{(t)}) \geq 0$ **then**
      $x_{k_j^*}^{(t)} = 1$;
    else
      $x_{k_j^*}^{(t)} = 0$;
    end
  end
  for $j = 1, \cdots, J$ **do**
    Each BS chooses a step size $\delta^{(t)}$ and computes $\Xi_j^{(t+1)} = \min\{L_j, e^{(\lambda_j^{(t)} - 1)}\}$ and $\lambda_j^{(t+1)} = \lambda_j^{(t)} - \delta^{(t)}(\Xi_j^{(t)} - \sum_{k=1}^{K} x_{k_j}^{(t)})$;
  end
end

Theorem 3: The optimal solution to problem (30) is also feasible and optimal to problem (28).

Proof: From problem (28)–(30), the following change has been applied:

$$x_{k_j} \in \mathcal{B} \to x_{k_j} \in \mathcal{F} \tag{35}$$

where $\mathcal{B} = \{0, 1\}$ is set containing only 0 and 1, and $\mathcal{F} = [0, 1]$ is the set containing all the fractions from 0 to 1, inclusive of 0 and 1. That is, we relax the variables from binary to real.

On one hand, since the set $\mathcal{F}$ contains much more numbers than the set $\mathcal{B}$, the optimal value to problem (30) provides an upper bound to that of problem (28). Denote the optimal utility in (28) as U_1 and the optimal utility in (30) as U_2, we have

$$U_2 \geq U_1. \tag{36}$$

On the other hand, it can be observed from Algorithm 3 that

$$x_{k_j^*}^{(t)} = 1 \quad x_{k_j^*}^{(t)} = 0. \tag{37}$$

That means that the solutions to problem (30) are 0 s and 1 s, which fall into $\mathcal{B}$. Therefore, we have

$$U_1 \geq U_2. \tag{38}$$

With (37) and (38), we conclude that $U_1 = U_2$. Thus, although we transform problem (28) to problem (30), the optimal solution is not affected by the transformation. ■

To sum up, the optimal solution to problem (28) can be solved with Algorithm 3. For comparison purposes, we also propose two greedy algorithms as benchmarks, which are presented in Algorithm 4 and Algorithm 5. The main idea of the greedy algorithms is to first identify the most desirable user–BS pair and then to allocate all the resource to that user. This is repeated until convergence is reached.

The complexity of both greedy algorithms is $\mathcal{O}(JK)$. For each iteration of Algorithm 3, the complexity is $\mathcal{O}(JK)$. Thus, the overall complexity of Algorithm 3 is higher than that of Algorithm 4 and Algorithm 5. However, we will show later that the complexity cost pays off. The optimal solution achieves a great performance gain over the greedy algorithms.

Algorithm 4: Greedy Algorithm 4 for Joint Resource Allocation and User Association

Initialize $\mathcal{K} = \{1, 2, \cdots, K\}$ and $\mathcal{J} = \{1, 2, \cdots, J\}$ x_{k_j} to be an all-zero matrix ;
for $k = 1$ *to* K **do**
  for $j = 1$ *to* J **do**
    Compute c_{k_j} as in (13) ;
  end
end
while $\max_{k,j} \log(c_{k,j}) > 0$ **do**
  Find $(k^*, j^*) = \arg\max_{k,j} \log(c_{k,j})$;
  $x_{k_{j^*}^*} = 1$;
  $\mathcal{K} = \mathcal{K} \backslash k^*$;
  $\mathcal{J} = \mathcal{J} \backslash J^*$;
end

Algorithm 5: Greedy Algorithm 5 for Joint Resource Allocation and User Association

Initialize $\mathcal{K} = \{1, 2, \cdots, K\}$, $\mathcal{J} = \{1, 2, \cdots, J\}$ and x_{k_j} to be an all-zero matrix ;
for $k = 1$ *to* K **do**
  for $j = 1$ *to* J **do**
    Compute c_{k_j} as in (13) ;
  end
end
for $j = 1$ *to* J **do**
  if $\max_k \log(c_{k,j}) > 0$ **then**
    Find $(k^*, j) = \arg\max_k c_{k,j}$;
    $x_{k_j^*} = 1$;
    $\mathcal{K} = \mathcal{K} \backslash k^*$;
  end
end

IV. Distributed User Association

In the previous section, we assume a central controller that has global information and assigns users to the BSs. Here, we consider distributed user association. We still assume that the BSs have all the CSI via uplink training. We further assume that all the BSs belong to the same service provider. Each user makes its own decision based on the broadcast and local information. Throughout this section, we do not allow fractional connection. We omit constraint (2) in the problem formulation, which is, however, enforced when solving the problem.

We model the behavior and interactions among the service provider and users using repeated game theory. The first key problem is to determine whether the game will converge. The second key problem is to analyze whether both sides are satisfactory about the outcome of the game, i.e., existence of the NE.

A. Service Provider Sets the Price

The players of the repeated game include the service provider and the users. During each round of the game, the service provider determines the price of the connection service. The users decide whether to connect, and if to connect, to which BS. The strategy of the service provide is to set the price p_{k_j} of each BS j for each user k, whereas the strategy of each user k is to set x_{k_j} to either 0 or 1 for $j \in \mathcal{J}$.

The utility of the service provider is defined as $\mathcal{U}_B = \sum_{k=1}^{K} \sum_{j=1}^{J} x_{k_j} p_{k_j}$. Since each BS is constrained by its maximum load capacity L_j, the service provider aims to solve the following problem:

$$\begin{aligned} \max_{\{p_{k_j}\}} \ & \mathcal{U}_B = \sum_{k=1}^{K} \sum_{j=1}^{J} x_{k_j} p_{k_j} \\ \text{s.t.} \ & \sum_{k} x_{k_j} \leq L_j, \quad j = 1, 2, \ldots, J. \end{aligned} \tag{39}$$

The utility of each user is the data rate achieved minus its payment. Thus, each user aims to solve the following problem:

$$\begin{aligned} \max_{\{x_{k_j}\}} \ & \mathcal{U}_k = \max \left\{ \omega_k \log \left(\sum_{j=1}^{J} x_{k_j} c_{k_j} \right) - \sum_{j=1}^{J} x_{k_j} p_{j_k}, 0 \right\} \\ \text{s.t.} \ & \sum_{j} x_{k_j} \leq 1 \end{aligned} \tag{40}$$

where the logarithmic function represents the satisfaction level of a user k toward its achievable rate, and ω_k is a weight used to tradeoff rate satisfaction and monetary payment. We assume that the weight ω_k of each user is drawn from a finite set $\mathcal{W}$ with $|\mathcal{W}|$ elements. This assumption is true in real-world practice. For instance, \$30 for a wireless service with a 60-Mb/s data rate is considered to be cheap, \$45 is considered to be reasonable, \$60 would be acceptable, \$80 would be expensive for most people, \$100 would be too expensive, and \$150 or above would not be an option for most people. Thus, the weight of the users has generally finite choices of values based on common sense and is typically in a range $= (0, W_M)$, where W_M is the maximum possible value for ω_k.

The repeated game is played as follows. Initially, the service provider sets a price for each BS for each user and broadcasts the prices to the users. Knowing the prices, the users will feedback the service provider of their choices based on their own calculations. Then, the service provider updates the prices and broadcasts them to the users. Users again inform the service provider of their choices, and so forth. The process is repeated until both the service provider and users are all satisfied with the price.

Given the players and their strategies and utilities, we have the following definition for the NE of the user association game.

Definition 2: A strategy set $\{p_{k_j}^*, x_{k_j}^*\}$, for all k, j, is an NE of the repeated game if $\mathcal{U}_B(p_{k_j}^*, x_{k_j}^*) \geq \mathcal{U}_B(p_{k_j}, x_{k_j}^*)$, for all p_{k_j}, and $\mathcal{U}_k(p_{k_j}^*, x_{k_j}^*) \geq \mathcal{U}_k(p_{k_j}^*, x_{k_j})$, for all k, x_{k_j}.

Due to the constraint that each user can only connect to one BS, $\omega_k \log(\sum_{j=1}^{J} x_{k_j} c_{k_j}) = \sum_{j=1}^{J} x_{k_j} \omega_k \log(c_{k_j})$. Therefore, the objective function of problem (40) becomes

$$\mathcal{U}_k = \max \left\{ \sum_{j=1}^{J} x_{k_j} \omega_k \log(c_{k_j}) - \sum_{j=1}^{J} x_{k_j} p_{j_k}, 0 \right\}. \tag{41}$$

For the reformulated problem (41), the constraint $\sum_j x_{k_j} \leq 1$ indicates that a user may choose not to connect to any of the BSs. On the other hand, if we restrict $\sum_j x_{k_j} = 1$, then, even if the service provider sets the prices to infinity, each user will still connect to a BS, which is clearly unreasonable.

Given the utility function (41) and the constraint in (40), the optimal solution for each user can be derived as

$$j^* = \arg\max_{j \in \mathcal{J}} \left[\omega_k \log(c_{k_j}) - p_{j_k} \right] \tag{42}$$

$$x_{k_j} = \begin{cases} 1, & \text{if } j = j^* \text{ and } \omega_k \log\left(c_{k_j^*}\right) \geq p_{j_k^*} \\ 0, & \text{otherwise.} \end{cases} \tag{43}$$

Such users' decision can be interpreted this way. A user will choose the best connection based on its own evaluation. If its evaluation of the connection is greater than or equal to the price, it will connect to this BS. Otherwise, the user will not connect to the BS. Thus, we readily have the following result.

Lemma 3: The highest profit the service provider can obtain from a user k toward BS j is the user's evaluation.

The service provider aims to solve problem (39) by tuning variables p_{k_j}, $k = 1, 2, \ldots, K$, $j = 1, 2, \ldots, J$. However, the constraint $\sum_k x_{k_j} \leq L_j$ is implicitly coupled with all p_{k_j}, since, according to the user's choice, $j^* = \arg\max_{j \in \mathcal{J}} \left[\omega_k \log(c_{k_j}) - p_{j_k}\right]$. The service provider problem is actually with the following form:

$$\begin{aligned} \max_{\{p_{k_j}\}} \ & \mathcal{U}_B = \sum_{k=1}^{K} \sum_{j=1}^{J} x_{k_j} p_{k_j} \\ \text{s.t.} \ & \sum_{k} x_{k_{j(p_{k_j})}} \leq L_j, \quad j = 1, 2, \ldots, J. \end{aligned} \tag{44}$$

Since problem (44) has coupling constraints, one may try to introduce Lagrange multipliers to the constraint and solve the resulting problem using dual decomposition. However, since p_{k_j} is implicitly contained in the constraint, the gradient and subgradient are difficult to find. Next, we propose Algorithm 6 for the service provider and then prove that the algorithm achieves optimal utility for the service provider and the users. Note that, in Algorithm 6, $|F_k|$ is the feedback size of user k.

Theorem 4: If the service provider adopts Algorithm 6, the game converges, and the NE can be achieved.

Proof: We first notice that the service provider has priority over the users. The users always make decisions based upon the service provider's price settings. Basically, the service provider controls when the repeated game terminates.

In Algorithm 6, the service provider tests out the weight of each user using binary search with $\mathcal{O}(\log_2(|\mathcal{W}|))$ steps. Once the service provider obtains ω_k, $k = 1, 2, \ldots, K$, it then estimates the users' price evaluation matrix $\mathbf{V}$ as follows:

$$v_{k_j} = c_{k_j} \omega_k \tag{45}$$

where v_{k_j} is the entry of matrix $\mathbf{V}$ at row j and column k. Following Lemma 3, the service provider can obtain its optimal price strategy by first selecting users for each BS and solving the following problem:

$$
\begin{aligned}
\max_{x_{k_j}} \quad & \sum_{k=1}^{K}\sum_{j=1}^{J} x_{k_j} v_{k_j} \\
\text{s.t.} \quad & \sum_{k} x_{k_j} \le L_j, \quad j = 1, 2, \ldots, J \\
& \sum_{j} x_{k_j} \le 1, \quad k = 1, 2, \ldots, K \\
& \text{Constraints } (2), (45).
\end{aligned} \tag{46}
$$

The optimal solution $x^*_{k_j}$ to the preceding problem can be solved in a similar way as solving problem $\mathbf{P1-2}$. Then, the optimal prices for the service provider can be obtained as follows:

$$
p^*_{k_j} = \begin{cases} v_{k_j}, & \text{if } x^*_{k_j} = 1 \\ v_{k_j} + \epsilon, & \text{otherwise} \end{cases} \tag{47}
$$

where ϵ is an arbitrary positive number.

Algorithm 6: Algorithm for Service Provider

Initialize ω_{MAX}, ω_{MIN}, $t = 0$;
for $k = 1$ *to* K **do**
  for $j = 1$ *to* J **do**
    Compute c_{k_j} as in (13) ;
  end
end
for $k = 1$ *to* K **do**
  $\omega^u_k(t) = \omega_{MAX}$;
  $\omega^l_k(t) = \omega_{MIN}$;
end
while *not converged* **do**
  for $k = 1$ *to* K **do**
    $\hat{\omega}_k(t) = \frac{1}{2}(\omega^u_k(t) + \omega^l_k(t))$;
    for $j = 1$ *to* J **do**
      $p_{k_j}(t) = \max\left\{\hat{\omega}_k(t)\log(c_{k_j}), 0\right\}$;
    end
  end
  $t \leftarrow t + 1$;
  for $k = 1$ *to* K **do**
    if $|F_k| > 1$ **then**
      $\omega^u_k(t) = \omega^u_k(t-1)$;
      $\omega^l_k(t) = \omega^l_k(t-1)$;
    else if $|F_k| = 1$ **then**
      $\omega^u_k(t) = \omega^u_k(t-1)$;
      $\omega^l_k(t) = \hat{\omega}_k(t)$;
    else
      $\omega^u_k(t) = \hat{\omega}_k(t)$;
      $\omega^l_k(t) = \omega^l_k(t-1)$;
    end
  end
end
for $k = 1$ *to* K **do**
  for $j = 1$ *to* J **do**
    Calculate v_{k_j} as in (45) using $\hat{\omega}_k$;
  end
end
Solve (46) and find optimal price as in (47) ;

Therefore, by adopting Algorithm 6, the optimal utility (highest) can be reached for the service provider. Meanwhile, we could see that all the users' utility must be 0 due to the optimal price setting (i.e., each user's rate satisfaction matches its monetary payment). That means that all the users achieve the optimal utility given the price setting as well. Therefore, the game converges to the NE. ■

Note that it is possible that the optimal utility of the service provider will be lower than the maximum utility during the game, because the load capacity constraint may be violated due to the distributed operation. The complexity of each iteration of Algorithm 6 is $\mathcal{O}(KJ)$.

B. User-Bidding-Based Approach

We next consider a bidding approach to the problem. Before service starts, users bid to the service provider according to their predicted satisfaction toward each BS. In addition, service provider determines whether to accept a user's bid and feedback the decisions to users. Then, the users make another round of bids according to its predicted satisfaction and the service provider's decision history. The service provider again decides whether to accept a user's bid and feedback the decision, and so forth.

Assuming date-intensive users that strive for as high data rate as possible, each user solves the following problem:

$$
\begin{aligned}
\max_{\{p_{k_j}\}} \quad & \mathcal{U}_k = \max\left\{\sum_{j=1}^{J} x_{k_j}\omega_k \log(c_{k_j}), 0\right\} \\
\text{s.t.} \quad & \sum_{j} x_{k_j(p_{k_j})} \le 1.
\end{aligned} \tag{48}
$$

On the other hand, the service provider aims to maximize its utility, i.e., the total payment made by all the users, i.e.,

$$
\begin{aligned}
\max_{\{x_{k_j}\}} \quad & \mathcal{U}_B = \sum_{k=1}^{K}\sum_{j=1}^{J} x_{k_j} p_{k_j} \\
\text{s.t.} \quad & \sum_{k} x_{k_j} \le L_j, \quad j = 1, 2, \ldots, J.
\end{aligned} \tag{49}
$$

Note that the decision variables in these two problems are different from those in problems (39) and (40), respectively.

We assume the general case that $K \ge \sum_{j=1}^{J} L_j$ (i.e., not all the users can be served). In order to achieve the greatest level of satisfaction, each user makes the highest possible payment. Thus, the optimal solution for each user is

$$
p_{k_j} = \max\left\{\sum_{j=1}^{J} x_{k_j}\omega_k \log(c_{k_j}), 0\right\}. \tag{50}
$$

The optimal strategy for the service provider is summarized in Algorithm 7.

Algorithm 7: Algorithm for the Service Provider with the Bidding Approach

while *not converged* **do**
  for $j = 1$ *to* J **do**
    if *BS j is bidden by $\le L_j$ users* **then**
      Keep all the users in BS j's waiting list ;
    else
      Keep the top L_j users with the highest bids and reject the other users ;
    end
  end
end

During the first stage of the game, each user offers a price to its most desirable BS. Algorithm 7 is used to check if each BS j receives more than L_j bids. The service provider only puts L_j top users on BS j's waiting list based on the offered prices and rejects all other users. If BS j receives no more than L_j bids, all these users will be put on BS j's waiting list.

At the second stage, if a user is in a BS's waiting list, it will keep on bidding the same BS with the same price to guarantee the highest utility. However, if a user gets rejected in the previous round, as being selfish, it will exclude the BSs that have rejected it and offers a price to its most desirable BS among the remaining ones. For the service provider, it adopts the same strategy. If the number of bids received for a BS outnumbers the load capacity of that BS, the service provider only keeps the L_j most desirable users on the waiting list and rejects the others. It keeps all users on the waiting list if the number of offers received is less than a BS's load capacity. These two stages repeat until convergence is achieved. The complexity of each iteration of Algorithm 7 is $\mathcal{O}(J)$.

Lemma 4: The sequence of bids made by a user is nonincreasing in the user's preference list.

Proof: Before a user makes an offer, it computes the satisfaction of all the BSs to obtain a preference list. Since a user aims to maximize its utility, it first proposes to the BS with the highest satisfaction. If it is rejected by the BS, it will propose to the BS with the second highest satisfaction, and so forth. Note that, even if a user may be on the waiting list of a BS, it may be removed from that waiting list at a later stage. If that happens, this user will start bidding to other BS. A user will repeat this procedure until it is finally in a BS's serving list or rejected by all BSs. This concludes the proof. ■

Lemma 5: The sequence of bids a BS put on the waiting list is nondecreasing in its preference list.

Proof: Given the fact any BS has a finite load capacity and $K \geq \sum_{j=1}^{J} L_j$, all the BSs will have at least one user bidding to it at some stage of the game. Since a BS aims to maximize its utility, it puts all the users who make an offer on the waiting list. On the condition that there are too many users, it will reject the users who it will never serve. In the next round of game, the BS will often have more or at least the same amount of bids compared with its current waiting list. This means that the BS has more choices. The BS again only keeps the most profitable ones and reject or remove the others from the waiting list. Thus, the sequence of bids a BS put on the list is nondecreasing in its preference list. ■

Theorem 5: The repeated bidding game converges.

Proof: We prove this theorem using contradiction as follows. Based on Lemma 4 and Lemma 5, we prove this theorem by contradiction. Suppose that this repeated game does converge. Then there must be a stage of the game that 1) there is a user k and a BS j pair so that user k is connected to another BS j' or is not connected to any BS, 2) user k prefers BS j to BS j' or prefers to be not connected, and 3) BS j prefers user k to a user k' who is on its serving list.

Consider the case where user k is served by BS j'. Since the sequence of bids made by a BS is nondecreasing, it must be the case that user k has never bidden to BS j during the game. Otherwise, if user k has bidden to BS j, BS j would not have ended up with choosing k' over k. In this case, user k would never have bidden to BS j' either, since user k prefers j to j' and the bids (see Lemma 4). However, user k is now served by BS j', user k must have bidden to BS j', which contradicts that user k would never have bidden to BS j'.

The same reasoning holds for the case when user k is not connected to any BS. If BS j prefers k to k' on the serving list, BS j would never reject user k while keeping user k'.

Therefore, the game converges when every user is either on a waiting list or has been rejected by every BS, and the game will converge. ■

From the proof, we can actually see that the game terminates when the least popular BS becomes fully loaded.

Theorem 6: The outcome of the repeated bidding game is optimal for both the users and the service provider.

Proof: We prove this theorem using contradiction as follows. Suppose that the outcome of the game is not optimal for a user k, who is connected to BS j. Then there must be another BS j', which has higher ranking than BS j in the preference list of user k and has a serving list of users $\{j'_1, j'_2, \ldots, j'_{L_{j'}}\}$. Since BS j' serves these users, it means that BS j' prefers them to user k and BS j' is at the top of the preference lists of these users. If, at some stage, user k is in the waiting list of BS j (or it is inserted by force), the game must have not terminated.

Since user k is in the waiting list, then one of the final users $j'_1, j'_2, \ldots, j'_{L_{j'}}$ must be currently off the list, for example, user $j'_{L_{j'}}$. Then user $j'_{L_{j'}}$ will immediately bid for BS j', since BS j' is at the top of its preference list among all the remaining BSs. In addition, BS j' will remove user k from its waiting list, since user k has a lowest ranking in the preference list of BS j'. Thus, when the repeated game terminates, the outcomes are optimal for each user. It is obvious that the outcome is also optimal for the service provider as well. ■

From Theorem 5 and Theorem 6, we conclude that the game converges to the NE when the game terminates.

V. Simulation Study

We validate the proposed schemes with MATLAB simulations. The same random seeds were used for different algorithms for a fair comparison. For user location, we randomly generated their locations 1000 times. For each user generation, we randomly generate channel coefficients 2000 times. Thus, the result presented is the average of 2 000 000 simulations.

Throughout the simulations, we assume $l_{j,k} = 1/(1 + (d_{j,k}/40)^{3.5})$ for the path loss between a user and the massive MIMO BS and $l_{j,k} = 1/(1 + (d_{j,k}/40)^{4})$ for the path loss between a user and a small cell BS [12]. We assume that the power of small-scale fading follows a uniform distribution in [0.8,1]. We fix the location of the massive MIMO BS at the center of the cell. The other BSs are randomly placed across in the cell. Users are randomly placed in the area. The other parameter settings are listed in Table I. The error bars in the plots are 95% confidence intervals.

Table II presents a comparison of rate maximization with the optimal solution and the two proposed greedy algorithms. Table III shows a comparison of rate maximization considering proportional fairness with the optimal solution and the two

TABLE I
SYSTEM CONFIGURATION

Parameter	Value	Parameter	Value
$M_{massive}$	100	M	4
$L_{massive}$	10	L	4
$P_{massive}$	40 dBm	P	40 dBm
Area	1000×1000 m^2	J	11

TABLE II
RATE MAXIMIZATION OF CENTRALIZED CONTROL

K	50	100	150	200	250
Optimal Rate Maximation	382.9	483.4	543.1	572.3	594.0
Greedy Algorithm 1	363.5	480.0	540.4	571.1	592.6
Greedy Algorithm 2	191.0	279.6	340.6	371.5	392.5

TABLE III
LOG RATE UTILITY OF CENTRALIZED CONTROL

K	50	100	150	200	250
Optimal Log Rate Max.	127.9	154.3	167.3	172.6	175.9
Greedy Algorithm 1	115.0	153.3	166.9	172.4	175.7
Greedy Algorithm 2	67.5	97.7	122.0	133.5	140.0

TABLE IV
JOINT RESOURCE ALLOCATION AND USER ASSOCIATION

K	50	100	150	200	250
Optimal Joint Resource Allocation and User Association	42.3	50.8	55.8	56.1	62.0
Greedy Algorithm 4	35.3	37.9	39.3	39.9	40.3
Greedy Algorithm 5	35.1	37.9	39.3	39.8	40.2

proposed greedy algorithms. We can see from both tables that the optimal solution achieves the highest network utility. We also notice that, as the number of users increases, the gaps between the optimal utility and the greedy solutions become narrower. This is because, as there are more users, the user diversity effect becomes stronger. Thus, the greedy algorithms and the optimal user association algorithm tend to produce similar solutions. Comparing greedy Algorithm 1 and Algorithm 2, we find that, in each step, Greedy Algorithm 1 always chooses the globally best user–BS pair, whereas Greedy Algorithm 2 chooses the best user for each BS sequentially. This is why Greedy Algorithm 1 always achieves better performance than Greedy Algorithm 2.

Table IV presents a comparison of the optimal joint resource allocation and user association algorithm and the two proposed greedy algorithms. We find that the optimal scheme achieves the highest utility. Moreover, the gap between the optimal scheme and the greedy schemes is quite large. We also consider the equality constraint problem as a benchmark for the comparison. For a fair comparison, we set the sum capacity of this system equal to the number of users. Thus, there are a total of $K = 50$ active users in the system. The optimal solution of problem $\mathbf{P3-2}$ achieves a network utility of -59.8462, whereas the optimal solution of problem (21) has a network utility of 29.5433. We also found that if we connect every user, some edge users will be harmful for the network utility.

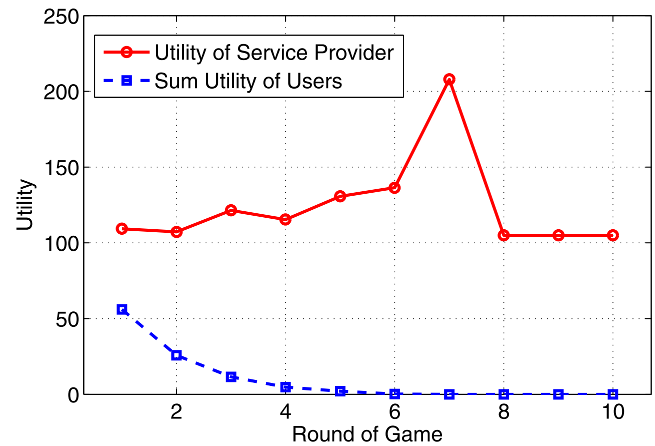


Fig. 1. Convergence of the repeated game when the service provider sets the price and $K = 100$.

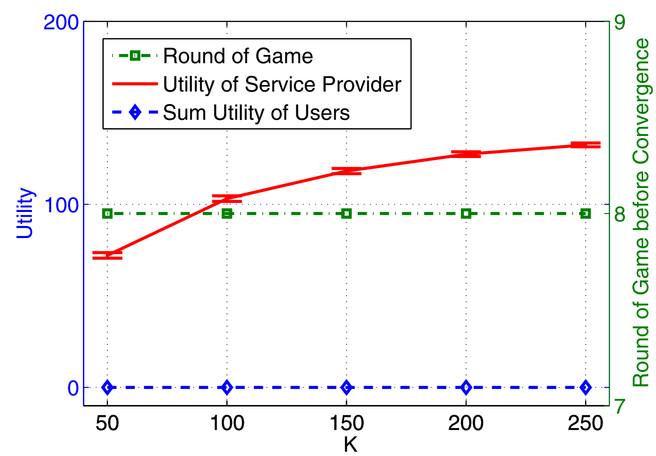


Fig. 2. Utility of the service provider, utility of the users, and the number of rounds for convergence for systems with various numbers of users.

Fig. 1 shows the utility of the service provider and all users when the service provider sets the price (as in Section IV-A). It can be seen that the repeated game converges after eight rounds. Furthermore, the utility of all users is monotonically decreasing. That is because once a user's evaluation is known to the service provider, the service provider will set prices for the highest profit, which results in 0 utility for that user. As discussed, the utility for the service provider is not monotonically increasing, since, during the game, the load capacity constraint may be violated. Since it is not the final result of the game, the capacity constraint could be violated for the time being. Since the loading capacity is violated, the utility of the service provider could be very high. However, that actually could not be achieved after the game converges. Thus, the utility after the game converges could be lower than that during the repetition process of the game. Fig. 2 plots the utilities of the service provider and users versus the number of users. We can see that, as the number of user increases, the utility of the service provider also increases. This is mainly due to the effect of multiuser diversity. We can also observe that the game terminates after about eight rounds no matter how many users are active.

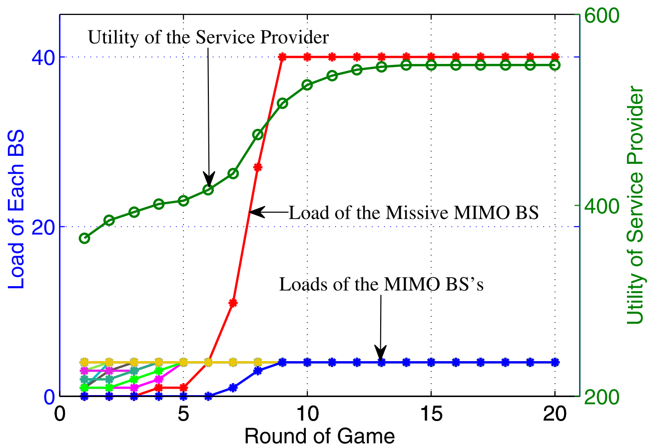


Fig. 3. Convergence of the repeated game with respect to BS load when users bid.

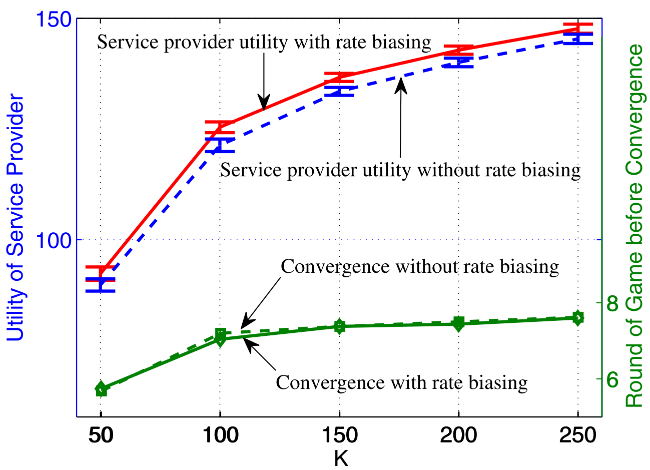


Fig. 4. Utility of the service provider with or without rate bias and convergence of the games under different numbers of users.

Fig. 3 depicts the process of the game when users bid for BSs (as in Section IV-B). Here, we deploy $J = 41$ BSs. The massive MIMO BS has $M = 400$ antennas. There are $K = 350$ users. The left-hand-side y-axis represents the load of the 41 BSs. The right-hand-side y-axis represents the utility of the service provider. We find the game converges in about ten rounds, and the utility of the service provider is monotonically increasing as the game continues.

To encourage offloading from the macro BS, we consider rate bias for the BSs in this experiment. Specifically, we multiple the rate of the massive MIMO BS with a factor of 0.5 to encourage connection to the PBSs. Fig. 4 shows the result when configuration shown in Table I is used. It can be observed that the utility with rate bias is higher than the utility without considering rate bias. This result demonstrates the efficacy of rate bias and offloading. It can be seen that both games terminate in less than eight rounds.

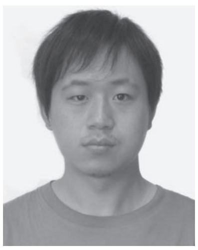
VI. Conclusion

In this paper, we have investigated the user association problem in a massive MIMO HetNet. By leveraging total unimodularity, we developed optimal algorithms for rate maximization and rate maximization with proportional fairness problems. We also developed optimal algorithms to the joint resource allocation and user association problem with primal decomposition and dual decomposition. Modeling the behavior and interaction of the service provider and users with repeated games, we developed effective distributed algorithms with proven convergence to the NE. Simulation results verify the efficacy of the proposed schemes.

References

[1] Qualcomm, "The 1000x data challenge," Qualcomm Technol., Inc., San Diego, CA, USA, Tech. Rep., 2014. [Online]. Available: https://www.qualcomm.com/invention/technologies/1000x
[2] E. G. Larsson, F. Tufvesson, O. Edfors, and T. L. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 185–195, Feb. 2014.
[3] 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.
[4] C. Shepard *et al.*, "Argos: Practical many-antenna base stations," in *Proc. ACM MobiCom*, Istanbul, Turkey, Aug. 2012, pp. 53–64.
[5] C. Shepard, H. Yu, and L. Zhong, "ArgosV2: A flexible many-antenna research platform," in *Proc. ACM MobiCom*, Miami, FL, USA, Sep. 2013, pp. 163–165.
[6] J. Andrews *et al.*, "What will 5G be?" *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014.
[7] J. G. Andrews, "Seven ways that HetNets are a cellular paradigm shift," *IEEE Commun. Mag.*, vol. 51, no. 3, pp. 136–144, Mar. 2013.
[8] 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.
[9] A. Adhikary and G. Caire, "Joint spatial division and multiplexing: Opportunistic beamforming and user grouping." [Online]. Available: http://arxiv.org/abs/1305.7252
[10] 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, NSW, Australia, Jun. 2014, pp. 5208–5213.
[11] Y. Xu, G. Yue, and S. Mao, "User grouping for Massive MIMO in FDD systems: New design methods and analysis," *IEEE Access*, vol. 2, no. 1, pp. 947–959, Sep. 2014.
[12] 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.
[13] S. Bayat, R. H. Y. Louie, Z. Han, B. Vucetic, and Y. Li, "Distributed user association and femtocell allocation in heterogeneous wireless networks," *IEEE Trans. Commun.*, vol. 62, no. 8, pp. 3027–3043, Aug. 2014.
[14] Y. Zhang, D. Bethanabhotla, T. Hao, and K. Psounis, "Near-optimal user-cell association schemes for real-world networks," in *Proc. Inf. Theory Appl. Workshop*, San Diego, CA, USA, Feb. 2015, pp. 1–10.
[15] A. Gupta, H. Dhillon, S. Vishwanath, and J. Andrews, "Downlink multi-antenna heterogeneous cellular network with load balancing," *arXiv:1310.6795v2*. [Online]. Available: http://arxiv.org/abs/1310.6795
[16] E. Björnson, M. Kountouris, and M. Debbah, "Massive MIMO and small cells: Improving energy efficiency by optimal soft-cell coordination," *arXiv:1304.0553v3*. [Online]. Available: http://arxiv.org/abs/1304.0553
[17] Y. Xu, S. Mao, and X. Su, "On adopting Interleave Division Multiple Access to two-tier femtocell networks: The uplink case," in *Proc. IEEE ICC*, Ottawa, ON, Canada, Jun. 2012, pp. 591–595.
[18] A. Damnjanovic *et al.*, "A survey on 3GPP heterogeneous networks," *IEEE Wireless Commun. Mag.*, vol. 18, no. 3, pp. 10–21, Jun. 2011.
[19] L. Lu, G. Li, A. Swindlehurst, A. Ashikhmin, and R. Zhang, "An overview of massive MIMO: Benefits and challenges," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 742–758, Oct. 2014.
[20] R. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*, R. Miller, J. Thatcher, and J. Bohlinger, Eds. New York, NY, USA: Springer-Verlag, 1972, pp. 85–103.
[21] A. Schrijver, *Theory of Linear and Integer Programming*. Hoboken, NJ, USA: Wiley, Jun. 1998.
[22] C. Berenstein and R. Gay, *Complex Variables: An Introduction*. New York, NY, USA: Springer-Verlag, 1997.
[23] Yandra and H. Tamura, "A new multiobjective genetic algorithm with heterogeneous population for solving flowshop scheduling problems," *Int. J. Comput. Integr. Manuf.*, vol. 20, no. 5, pp. 465–477, Jul. 2007.

[24] N. Cheung, X. Ding, and H. Shen, "OptiFel: A convergent heterogeneous particle swarm optimization algorithm for Takagi–Sugeno fuzzy modeling," *IEEE Trans. Fuzzy Syst.*, vol. 22, no. 4, pp. 919–933, Aug. 2014.
[25] D. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1439–1451, Aug. 2006.
[26] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[27] Q. Ye *et al.*, "User association for load balancing in heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 6, pp. 2706–2716, Jun. 2013.



Yi Xu (S'11) received the Ph.D. in electrical and computer engineering and the Master in probability and statistics from Auburn University, Auburn, AL, USA, in 2015 and 2014, respectively. He also received the M.S. degrees in electronic engineering from Tsinghua University, Beijing, China, in 2010 and the B.S. degree in electronic information engineering from University of Electronic Science and Technology of China, Chengdu, China, in 2007. His research interests include Optimization, Game Theory, MIMO, OFDM, IDMA, Cognitive Radio Networks and Machine Learning.

Shiwen Mao (S'99–M'04–SM'09) received Ph.D. in electrical and computer engineering from Polytechnic University, Brooklyn, NY, USA, in 2004. Currently, he is the Samuel Ginn Endowed Professor and Director of the Wireless Engineering Research and Education Center (WEREC) in the College of Engineering, Auburn University, Auburn, AL, USA. His research interests include wireless networks and multimedia communications, such as cognitive radio, small cells, mmWave networks, free space optical networks, and smart grid.

He is a Distinguished Lecturer of the IEEE Vehicular Technology Society in the Class of 2014. He is on the Editorial Board of IEEE Transactions on Multimedia, IEEE Internet of Things Journal, IEEE Communications Surveys and Tutorials, and IEEE Multimedia, among others. He serves as Area TPC Chair of IEEE INFOCOM 2016, Technical Program Vice Chair for Information Systems (EDAS) of IEEE INFOCOM 2015, symposium co-chairs for many conferences, including IEEE ICC, IEEE GLOBECOM, ICCCN, IEEE ICIT-SSST, among others, Steering Committee Voting Member for IEEE ICME and AdhocNets, and in various roles in the organizing committees of many conferences. He is the Vice Chair–Letters & Member Communications of IEEE ComSoc Multimedia Communications Technical Committee. He received the 2013 IEEE ComSoc MMTC Outstanding Leadership Award and the NSF CAREER Award in 2010. He is a co-recipient of the IEEE WCNC 2015 Best Paper Award, the IEEE ICC 2013 Best Paper Award, and the 2004 IEEE Communications Society Leonard G. Abraham Prize in the Field of Communications Systems.