PBUS: Efficient User Selection for Block Diagonalization in Dense Wireless Networks

Mengyao Ge and *Douglas M. Blough*
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332-0250
{mengyao, doug.blough}@ece.gatech.edu

***Abstract*—To mitigate the co-channel interference across adjacent access points (APs) and break the performance bottleneck in dense wireless networks, multiple APs are expected to share the channel state information and cooperatively process the user signals. While the inter-user interference can be eliminated using Block Diagonalization (BD) precoding technique, the number of simultaneous users is limited by the total number of transmit antennas. In this paper, we propose a novel pairing-and-binary-tree-based user selection algorithm (PBUS) to address the user selection issue for multi-user MIMO in dense environments. With the fitness metric evaluated for each pair of users, a binary tree is constructed to store multiple candidate user groups. The best user group is then selected from these candidates. PBUS can achieve both good sum-rate performance and low computational complexity, and also has the flexibility to trade off sum-rate performance and computational complexity.**

I. Introduction

The increasing use of advanced wireless devices is driving the demand for higher wireless data rates and is causing significant stress to existing wireless networks. The concepts of distributed MIMO and coordinated multipoint have attracted significant research interest because of their ability to dramatically increase wireless throughput [1]. These approaches are particularly suitable for the dense environment within an enterprise network, where multiple access points (APs) are closely deployed and the number of users can be quite large [2]. APs can cooperate to control the lower-layer parameters and to optimize the network performance, by connecting via low-latency links to a shared controller. The feasibility of synchronizing multiple cooperative APs has been demonstrated in existing work [3]. To reap the benefits of these approaches, advanced MIMO techniques need to be investigated, which can perform a combination of spatial multiplexing and interference suppression. In this work, we target dense enterprise wireless networks where there are multiple access points and a large number of users within a small geographical area. These dense network scenarios are among the most challenging for satisfying user demand.

Block Diagonalization is a low-complexity linear MIMO precoding technique, which eliminates inter-user interference by designing the precoder of each user to lie in the null space of the remaining users' channel matrices [4] [5]. However, the number of simultaneous users that can be handled with BD precoding is limited by the number of transmit antennas. When the number of users is larger than can be supported by the transmit antennas, the APs should determine a subset of users to optimize a desired utility function. This process is called *user selection*.

Since a brute-force search over all possible user combinations is prohibitive due to the high computational complexity with a large user population, greedy user selection algorithms have been investigated for BD precoding in [6]–[9]. These greedy approaches incrementally select one user in each iteration [6]–[9]. A capacity-based algorithm of this type, referred as the c-algorithm, is proposed in [6]. While the c-algorithm's aggregate performance is good, it requires numerous singular value decomposition (SVD) operations and a water-filling power allocation process in each iteration, and these calculations are quite time consuming. Alternative algorithms introduced in [6], [7] utilize the c-algorithm in the finalization step to refine the user selection, but this still generates a very high computational overhead, especially for networks with a large number of users. Low-complexity algorithms proposed in [8], [9] reduce the computational cost but achieve lower aggregate performance.

In this paper, we propose a **P**airing-and-**B**inary-tree-based **U**ser **S**election (PBUS) algorithm for a multi-user MIMO network with BD precoding. The PBUS algorithm has three phases: 1) a pairwise fitness evaluation to determine the fitness of different pairs of users, 2) a binary tree-based grouping to generate a varying number of user groups as candidates for selection, and 3) final user selection and its sum-rate maximization through the optimal power allocation with per-AP power constraint. The PBUS algorithm has several advantages as compared to existing approaches. First, it does not need full channel state information at the transmitter. In fact, it can work with the explicit channel feedback mechanism used in 802.11ac. Second, the number of candidate user groups can be easily adjusted through a parameter of the algorithm. This permits a trade-off between computational time and aggregate performance, i.e. as more candidate groups are considered, the aggregate performance is increased but the computation time is also increased, and vice versa as the number of candidate groups is decreased. This trade-off combined with the lower complexity operations performed by our algorithm provide significantly enhanced aggregate performance and running time, as compared to existing approaches. For example, with about

978-1-5090-5019-2/17/$31.00 ©2017 IEEE

60 users, we can achieve the same aggregate performance as the algorithm of [6] with about 1/5 the running time. Alternatively, with the same running time as the algorithm of [8], we can get about 15% higher sum rate performance. It is also noteworthy that the running time of our proposed PBUS varies in a narrow range when the number of candidate group changes. A final advantage of the PBUS algorithm is that it can efficiently update the user selection when most of the channels remain the same and only a few users experience channel changes. This reduces the running time of the algorithm even further under this condition.

II. System Model and Background

We consider a MIMO network with M APs cooperatively serving K users denoted by $\mathcal{K} = \{1, 2, \ldots, K\}$. For convenience, we assume each of the M APs is equipped with N_t antennas and each of the K users has N_r antennas. The matrix of complex channel gains between the cooperative APs and the antennas of the k^{th} user is denoted by $\boldsymbol{H}_k \in \mathbb{C}^{N_r \times MN_t}$. The data vector $\boldsymbol{x} = \left[\boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_K^T\right]^T$ with $\boldsymbol{x}_k \in \mathbb{C}^{N_r}$ is jointly precoded by the M APs using the precoding matrix $\boldsymbol{B} = [\boldsymbol{B}_1, \ldots, \boldsymbol{B}_K]$. $\boldsymbol{B}_k \in \mathbb{C}^{MN_t \times N_r}$ is the partition of $\boldsymbol{B}$ applied at the cooperative APs to precode the signals of the k^{th} user. $\boldsymbol{x}_k \in \mathbb{C}^{N_r}$ is the transmit signal vector for receiver k. It is assumed that the k^{th} user has N_r parallel data streams, although some of the streams can have a rate of zero.

The received signal of the k^{th} user is given by

$$\boldsymbol{y}_k = \boldsymbol{H}_k \boldsymbol{B}_k \boldsymbol{x}_k + \sum_{j=1, j\neq k}^{K} \boldsymbol{H}_k \boldsymbol{B}_j \boldsymbol{x}_j + \boldsymbol{n}_k \ , \tag{1}$$

where $\boldsymbol{n}_k$ is the additive white Gaussian noise vector for the k^{th} user with variance $\boldsymbol{E}(\boldsymbol{n}_k \boldsymbol{n}_k^\dagger) = \sigma_k^2 \boldsymbol{I}$. $(\cdot)^\dagger$ is the conjugate transpose of $(\cdot)$.

A. Channel state information (CSI) feedback

In IEEE 802.11ac, a compressed beamforming report is generated for each user based on the singular value decomposition (SVD) of the channel [10]. For the k^{th} user, we have

$$\boldsymbol{H}_k = \boldsymbol{U}_k \boldsymbol{S}_k \boldsymbol{V}_k^\dagger \ . \tag{2}$$

where $\boldsymbol{S}_k \in \mathbb{C}^{N_r \times N_r}$ is the diagonal matrix containing the singular values in a decreasing order. $\boldsymbol{U}_k \in \mathbb{C}^{N_r \times N_r}$ and $\boldsymbol{V}_k \in \mathbb{C}^{MN_t \times N_r}$ are the left and right singular matrix, respectively.

The explicit feedback in 802.11ac requires the right singular matrix to be decomposed, quantized and then fed back to the AP for transmit beamforming [10]. Based on 802.11ac, the right singular matrix $\boldsymbol{V}_k$ can be decomposed as

$$\boldsymbol{V}_k = \left\{ \prod_{i=1}^{N_r} \left(\boldsymbol{D}_k^i \prod_{j=i+1}^{N_t} \boldsymbol{G}_k^{j,i} \right) \right\} \tilde{\boldsymbol{I}} \boldsymbol{\Phi}_k^\dagger \ , \tag{3}$$

where $\tilde{\boldsymbol{I}}$ is a matrix containing the first N_r columns of an $MN_t \times MN_t$ unitary matrix and the post-multiplication matrix $\boldsymbol{\Phi}_k$ is a diagonal matrix and its element has unit amplitude and same phase as the last row of $\boldsymbol{V}_k$. $\boldsymbol{D}_k^i$ is the diagonal rotation matrix to remove the imaginary parts from the i^{th} column, which can be represented by the angles $\phi_{i,i}, \phi_{i+1,i}, \ldots, \phi_{N_t-1,i}$. $\boldsymbol{G}_k^{j,i}$ is the Givens rotation matrix to nullify the $(j,i)^{\text{th}}$ entry determined by the angle $\psi_{j,i}$. The angles ϕ's and ψ's are then quantized and fed back to the APs, while the diagonal matrix $\boldsymbol{\Phi}_k$ can be absorbed into $\boldsymbol{S}_k$. Besides, in 802.11ac, signal-to-noise ratio (SNR) information feedback is substituted with $\boldsymbol{S}_k$ since the singular value is proportional to the SNR of each stream.

B. Block diagonalization (BD)

In this work, BD precoders are utilized at the AP side, which is a suboptimal solution to fully eliminate the inter-user interference. The key idea of BD is to precode the k^{th} user's signal using $\boldsymbol{B}_k$ such that $\boldsymbol{H}_j \boldsymbol{B}_k = 0$ for $j \neq k$. In other words, the precoder $\boldsymbol{B}_k$ should be in the null space of other concurrent users' channel matrices. Due to the rank constraint of BD precoder, the maximum number of users can be served simultaneously is bounded by $\lceil MN_t/N_r \rceil$ [6]. When the number of users is larger than the maximum supportive number, a user select procedure is required to determine a subset of users that maximizes the sum-rate performance.

Let $\mathcal{G} = \{\pi_1, \ldots, \pi_{K_0}\}$ be a subset of users with $K_0 \leq \lceil MN_t/N_r \rceil$. The BD precoders of the selected users are derived as follows. Let

$$\tilde{\boldsymbol{V}}_{\pi_k} = \left[\boldsymbol{V}_{\pi_1}, \ldots, \boldsymbol{V}_{\pi_{k-1}}, \boldsymbol{V}_{\pi_{k+1}}, \ldots, \boldsymbol{V}_{\pi_{K_0}}\right] \ ,$$

and the precoder $\boldsymbol{B}_{\pi_k}$ lies in the null space of $\tilde{\boldsymbol{V}}_{\pi_k}^\dagger$. To obtain the null space of $\tilde{\boldsymbol{V}}_{\pi_k}^\dagger$, we use QR decomposition,

$$\tilde{\boldsymbol{V}}_{\pi_k} = \left[\boldsymbol{W}_{\pi_k}^{(1)} \ \boldsymbol{W}_{\pi_k}^{(0)}\right] \begin{bmatrix} \boldsymbol{R}_{\pi_k} \\ \boldsymbol{0}_{(MN_t-n)\times n} \end{bmatrix} \ , \tag{4}$$

where $n = (K_0 - 1)N_r$ and $\boldsymbol{R}_{\pi_k} \in \mathbb{C}^{n \times n}$ is an upper triangular matrix, $\boldsymbol{W}_{\pi_k}^{(1)} \in \mathbb{C}^{MN_t \times n}$ forms an orthonormal basis for the column space of $\tilde{\boldsymbol{V}}_{\pi_k}$ and $\boldsymbol{W}_{\pi_k}^{(0)} \in \mathbb{C}^{MN_t \times (MN_t - n)}$ forms the null space of $\tilde{\boldsymbol{V}}_{\pi_k}^\dagger$. Thus, the columns of $\boldsymbol{B}_{\pi_k}$ can be chosen as the linear combination of those in $\boldsymbol{W}_{\pi_k}^{(0)}$. For example, we can simply choose $\boldsymbol{B}_{\pi_k} = \boldsymbol{W}_{\pi_k}^{(0)}$.

III. PBUS User Selection

The aggregate performance of the MU-MIMO system is largely dependent on the selection of simultaneous users. Unlike the conventional algorithms [6]–[9], our proposed algorithm generates a small set of good candidate groups based on an efficient binary-tree-based procedure. The best user group is then selected out of these candidates. More importantly, our algorithm can adjust the number of candidate groups to balance between the computational cost and throughput performance, according to the computation effort and channel dynamics. In addition, when only some users experience channel changes, our proposed algorithm reuses good combinations and only needs to re-evaluate the combinations including the users whose channels changed. In this section, we elaborate the details of the proposed pairing-and-binary-tree-based user selection algorithm (PBUS).

A. PBUS overview

PBUS inherits the low-complexity characteristic from conventional greedy algorithms, and it can also achieve complexity reduction for update when the network is partially changed. In addition, unlike the conventional greedy algorithms that sequentially build a single candidate group, our algorithm expands multiple promising candidate groups in parallel and then selects the best group from these candidates. It therefore has a lower probability than the greedy algorithms of missing a high-performing user group.

At a high level, PBUS works as follows:

1) **Pairing**: First of all, the pairwise evaluation mechanism is carried out to evaluate the fitness of each pair of users. For example, there are 6 pairs if 4 users exist in the network, including $\{U_1, U_2\}$, $\{U_1, U_3\}$, $\{U_1, U_4\}$, $\{U_2, U_3\}$, $\{U_2, U_4\}$ and $\{U_3, U_4\}$. Due to the zero-interference constraint of BD precoding, the multi-user diversity gain is mainly from the fact that, for a sufficient number of users, we can find a user group whose channels are nearly orthogonal to each other. Therefore, the fitness metric is designed to reflect the orthogonality between the channels of two users.
2) **Grouping**: Then, a binary tree is created to store 2^{L-1} candidate user groups and the level of the tree is K_0, where K_0 is the number of maximum supportable users. The user with the highest interference-free data rate is selected as the root of the tree at level 1. As shown in Fig. 1, when the level of the tree is smaller than L, two users will be selected as the children of each node at that level based on the grouping preference. When the level of the tree reaches L, one user will be added as the child of each node in the subsequent levels. The grouping preference is calculated based on the pre-calculated pairing fitness metrics. As a result, there are 2^{L-1} leaves in the binary tree and each candidate user group is formed by collecting the users along the path from one leaf to the root. The parameter L can be tuned within the range of $[1, K_0]$ to control the total number of candidate user groups.
3) **Refining**: Finally, the best user group is selected out of the 2^{L-1} candidate user groups generated in Step 2) based on the estimated sum rate. The maximum sum rate of the finally selected user set is determined via optimal power allocation.
4) **Reduction in update**: To reduce the time complexity in update, PBUS caches the pre-generated pairwise fitness metrics and candidate user groups stored in the binary tree. It can reuse the pre-calculated information for static users while performing efficient update to accommodate users whose channels have changed.

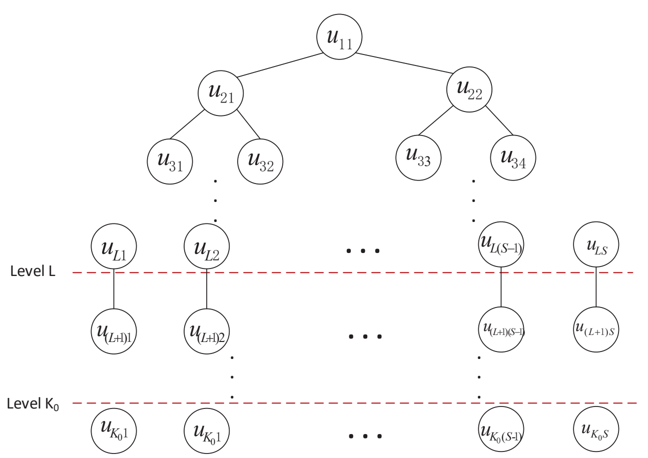

Fig. 1. Binary tree-based user grouping with $S = 2^{L-1}$ and $1 \le L \le K_0$.

The details of each step in PBUS are elaborated next.

B. Pairwise evaluation mechanism

In the first step, we propose a novel pairwise evaluation mechanism to determine the fitness of each pair of users. The fitness metric is proposed as

$$FM_{k,j} = \left[\gamma \left(\frac{\boldsymbol{S}_k \boldsymbol{V}_k \boldsymbol{N}_j}{\sigma_k ||\boldsymbol{N}_j||_F} \right), \gamma \left(\frac{\boldsymbol{S}_j \boldsymbol{V}_j \boldsymbol{N}_k}{\sigma_j ||\boldsymbol{N}_k||_F} \right) \right]^T , \quad (5)$$

where $\boldsymbol{N}_k = \boldsymbol{I} - \boldsymbol{V}_k \boldsymbol{V}_k^\dagger$ forms the null space of $\boldsymbol{V}_k^\dagger$. We define $\gamma(\boldsymbol{X}) = \log(1 + p\|\boldsymbol{X}\|_F^2)$ with $p = \sum_{m=1}^{M} P_m / K_0 / N_r$. When grouping the k^{th} user and the j^{th} user together, the channel matrix of one user should be projected into the null space of the other's channel matrix, in order to satisfy the zero-interference constraint. For a pair of users k and j, the first entry evaluates the total power gain from the eigenmodes of a null-space projected channel matrix of user k, and similar metric for user j is given in the second entry of the fitness metric $FM_{k,j}$. The fitness metric implicitly reflects channel orthogonality between the two users, and how much mutual interference each user generates in the other's subspace. Based on the symmetry of the pairwise fitness metric, we have $FM_{j,k} = \text{flip}(FM_{k,j})$, where $\text{flip}([a, b]^T) = [b, a]^T$.

C. Binary tree-based user grouping

In the grouping step, we expect to inherit the low-complexity property of conventional greedy algorithms while reducing the possibility of dropping good combinations during the iterations. Therefore, we introduce the binary tree-based user grouping method.

The procedure of grouping users is summarized in Table I. The parameter L is predetermined within the range of $[1, K_0]$, and is used to control the total number of generated user groups. The user with the highest interference-free data rate is picked as the root of the tree denoted by u_{11}. The root node is at the level of 1 as an initial user group $\mathcal{G}_{1,1} = \{u_{11}\}$. We then introduce the grouping preference metric of adding user k into the previously selected user group $\mathcal{G}_s$ as

$$GM(k, \mathcal{G}_s) = \sum_{i \in \mathcal{G}_s} \begin{bmatrix} 1/|\mathcal{G}_s| \\ 1 \end{bmatrix}^T FM_{k,i} , \quad (6)$$

where $|\mathcal{G}_s|$ represents the total number of users in set $\mathcal{G}_s$. The grouping preference metric $GM(\cdot)$ coarsely evaluate the

TABLE I
BINARY TREE-BASED USER GROUPING PROCEDURE

input:	K_0, $1 \le L \le K_0$, $\mathcal{K}=\{1,\dots,K\}$,
	$FM_{k,j}$ for $k,j=1,\dots,K,k\neq j$
output:	$\mathcal{T}$, $\mathcal{G}_{K_0,j}$ for $j=1,\dots,2^{L-1}$
1:	Find user u_{11} with highest interference-free data rate
2:	$\mathcal{T}_1 \leftarrow \{u_{11}\}$, $\mathcal{G}_{1,1} \leftarrow \{u_{11}\}$
3:	**for** i from 2 to L **do**
4:	**for** j from 1 to 2^{i-2} **do**
5:	$u^*_{2j-1,i} = \underset{k\in\mathcal{K}\setminus\mathcal{G}_{i-1,j}}{\operatorname{argmax}} GM(\mathcal{G}_{i-1,j},k)$
6:	$u^*_{2j,i} = \underset{k\in\mathcal{K}\setminus\{\mathcal{G}_{i-1,j}\cup u^*_{2j-1,i}\}}{\operatorname{argmax}} GM(\mathcal{G}_{i-1,j},k)$
7:	$\mathcal{G}_{i,2j-1} = \mathcal{G}_{i-1,j} \cup u^*_{2j-1,i}$
8:	$\mathcal{G}_{i,2j} = \mathcal{G}_{i-1,j} \cup u^*_{2j,i}$
9:	**endfor**
10:	$\mathcal{T}_i \leftarrow \{u^*_{j,i} \mid j=1,\dots,2^{i-1}\}$
11:	**endfor**
12:	**for** i from $L+1$ to K_0 **do**
13:	**for** j from 1 to 2^{L-1} **do**
14:	$u^*_{j,i} = \operatorname{argmax}_{k\in\mathcal{K}\setminus\mathcal{G}_{i-1,j}} GM(\mathcal{G}_{i-1,j},k)$
15:	$\mathcal{G}_{i,j} = \mathcal{G}_{i-1,j} \cup u^*_{j,i}$
16:	**endfor**
17:	$\mathcal{T}_i \leftarrow \{u^*_{j,i} \mid j=1,\dots,2^{L-1}\}$
18:	**endfor**

sum-rate performance by considering pairwise channel orthogonality. According to the grouping preference metric, two best users with the highest $GM(k,\mathcal{G}_{1,1})$ are selected at level 2 as the children of the root user, which produces the intermediate user groups $\mathcal{G}_{2,1}=\{u_{21},u_{11}\}$ and $\mathcal{G}_{2,2}=\{u_{22},u_{11}\}$. Here, for the j^{th} node at the i^{th} level, we define the intermediate user group $\mathcal{G}_{i,j}$ as the set containing the selected users along the path from the j^{th} node at the i^{th} level to the root.

Repeat the process for each node at the following levels by finding two children for each node based on the grouping preference metric until the tree grows to level L. Specifically, if $i<L$, for each intermediate user group $\mathcal{G}_{i,j}$, it will generate two intermediate user groups $\mathcal{G}_{i+1,2j-1}$ and $\mathcal{G}_{i+1,2j}$ at level $i+1$ by choosing $u_{(i+1)(2j-1)}$ and $u_{(i+1)(2j)}$ as the children of node $u_{i,j}$, as shown in line 3-11 in Table I. When the level of the tree reaches L, conventional incremental selection will be performed for each intermediate user set $\mathcal{G}_{L,j}$ until the maximum number of simultaneously supportable users is reached in each user group. In other words, only the best user will be selected as the child of each node at the levels higher than L until the level of the tree reaches K_0, as shown in line 12-18 in Table I. Finally, the constructed binary tree will store at most 2^{L-1} distinct candidate user groups.

The adjustable parameter L in the grouping stage determines the number of generated candidate user groups. When $L=1$, the proposed grouping approach degenerates into the conventional incremental user selection procedure, which simply selects one user at each time and produces one user group. When $L=K_0$, the tree structure in Fig. 1 becomes a full binary tree, producing 2^{K_0-1} possible user groups with more computational cost. Since more candidate user groups have higher possibility to find a better combination with higher sum rate, the parameter L controls fundamental tradeoffs between the aggregate performance and complexity. For example, large L is preferred when the network is static or slowly time-varying, which allows more computation time to obtain higher throughput. However, when the network is highly dynamic, smaller L is a better choice to speed up the user selection procedure. Although we do not report the detailed results herein, we also evaluated the performance of different non-binary-tree-based grouping schemes and found that they exhibit similar performance to the binary-tree-based scheme, as long as the numbers of finally generated candidate groups are the same.

D. Refining user selection

After the grouping procedure, there are at most 2^{L-1} candidate user groups, each of which includes the users selected along one user at the K_0^{th} level to the root, as shown in Fig. 1. The achievable sum rate of the j^{th} user group $\mathcal{G}_j$ can be estimated as follows by assuming equal power allocation,

$$\tilde{R}(\mathcal{G}_j)=\sum_{i\in\mathcal{G}_j}\log\left|\boldsymbol{I}+\boldsymbol{S}_i^2\boldsymbol{V}_i\tilde{\boldsymbol{B}}_{i,j}\tilde{\boldsymbol{B}}_{i,j}^{\dagger}\boldsymbol{V}_i^{\dagger}/\sigma_i^2\right|, \qquad (7)$$

where the subscript K_0 for $\mathcal{G}_{K_0,j}$ is omitted and $\tilde{\boldsymbol{B}}_{i,j}$ is the BD precoder for the i^{th} user in the group $\mathcal{G}_j$, which satisfies the per-AP power constraint. For example, with equal power allocation, we have $\tilde{\boldsymbol{B}}_{i,j}=\boldsymbol{P}_{i,j}\boldsymbol{B}_{i,j}$, where $\boldsymbol{P}_{i,j}=\operatorname{Diag}(\underbrace{\alpha_1,\dots,\alpha_1}_{N_t},\underbrace{\alpha_2,\dots,\alpha_2}_{N_t},\dots,\underbrace{\alpha_M,\dots,\alpha_M}_{N_t})$, such that $\operatorname{Tr}(\boldsymbol{\Gamma}_m\boldsymbol{P}_{i,j}\boldsymbol{B}_{i,j}\boldsymbol{B}_{i,j}^{\dagger}\boldsymbol{P}_{i,j}^{\dagger})=P_m/K_0$. The diagonal matrix $\boldsymbol{\Gamma}_m\in\mathbb{R}^{MN_t\times MN_t}$ is introduced for each AP to select the partition of $\tilde{\boldsymbol{B}}_{i,j}$ applied at the m^{th} AP and P_m is the maximum transmit power of the m^{th} AP. Thus, $\boldsymbol{\Gamma}_m$ contains ones on the diagonal elements corresponding to the antennas of the m^{th} AP and zeros elsewhere. $B_{i,j}$ can be obtained based on Sec.II-B.

Thus the best user group with highest estimated data rate is selected, i.e., $\mathcal{G}^*=\operatorname{argmax}_{\mathcal{G}_l}\tilde{R}(\mathcal{G}_l)$. The achievable sum-rate of the finally selected user group can be maximized via optimal power allocation, which will be discussed in Sec. IV.

E. Fast update with limited mobility

Since the sum-rate performance of a user group changes with the variation in channels, the user group needs to be updated accordingly. It is, however, very inefficient to completely re-perform the selection algorithm when only a few users experience channel variations. Therefore, the proposed PBUS algorithm can reuse the partial information calculated in the pairwise evaluation step to reduce the complexity in the first stage, i.e., the previously calculated pairwise fitness metric does not need to be updated if the channels of the two users are unchanged.

Besides, a small parameter L can be used for the grouping stage to accommodate users with channel changes, since we can reuse these previously generated user groups without mobile users as much as possible. This can speed up the grouping procedure with limited mobility while guaranteeing the aggregate performance.

F. Achieving fairness

For the targeted dense environment, there are typically a large number of users, only some of which are selected by our algorithm for a given communication round. This raises the question of overall fairness, i.e. how do we guarantee that users not selected in a particular round will eventually be served by the network? Although we do not evaluate it herein, our PBUS scheme can easily work with a scheduling algorithm such as the one in [11] to accommodate various fairness criteria. The algorithm of [11] operates by initially choosing a set of candidate high-performing user groups that cover all users. This set is then input into a scheduling algorithm that assigns the different groups to slots in an overall transmission schedule to achieve maximum performance while meeting specified fairness criteria. To generate candidate user groups, we can perform the PBUS algorithm multiple times starting with different root users, in order to find multiple high-performance candidate user groups. These candidate groups can then be fed into the scheduling algorithm, to meet the performance-maximizing fairness objective.

IV. Sum Rate Maximization with Per-AP Power Constraint

For the finally selected user group $\mathcal{G} = \{\pi_1, \ldots, \pi_{K_0}\}$, we denote BD precoder for the user π_k as $\boldsymbol{B}_{\pi_k}$. Let $\bar{\boldsymbol{V}}_{\pi_k} = \boldsymbol{S}_{\pi_k}\boldsymbol{V}_{\pi_k}\boldsymbol{B}_{\pi_k}$. The achievable sum rate with optimal power allocation is given by,

$$R_{BD}(\mathcal{G}) = \max_{\substack{\mathcal{G}\subset\mathcal{K}, \boldsymbol{Q}_{\pi_k}\succeq \boldsymbol{0} \\ \left\{\sum\limits_{\pi_k\in\mathcal{G}} \mathrm{Tr}(\boldsymbol{\Gamma}_m \boldsymbol{B}_{\pi_k}\boldsymbol{Q}_{\pi_k}\boldsymbol{B}_{\pi_k}^{\dagger}) \leq P_m\right\}}} \sum_{\pi_k\in\mathcal{G}} \log\left|\boldsymbol{I} + \bar{\boldsymbol{V}}_{\pi_k}\boldsymbol{Q}_{\pi_k}\bar{\boldsymbol{V}}_{\pi_k}^{\dagger}/\sigma_{\pi_k}^2\right| , \tag{8}$$

where $\boldsymbol{E}(\boldsymbol{x}_{i,j}\boldsymbol{x}_{i,j}^{\dagger}) = \boldsymbol{Q}_{i,j}$ is its transmit covariance matrix. The problem (8) reduces to a conventional sum-rate maximization problem with sum power constraint when $\boldsymbol{\Gamma}_m$ becomes an identity matrix with $M = 1$.

The optimal solution $\boldsymbol{Q}_{\pi_k}$'s to the right-hand side of (8) for user set $\mathcal{G}$ can be solved via Lagrange duality method. The Lagrange function of the right-hand side of (8) is given by

$$\begin{aligned} L(\{\boldsymbol{Q}_{\pi_k}\}_{\pi_k\in\mathcal{G}}, \boldsymbol{\mu}) = &- \sum_{\pi_k\in\mathcal{G}} \log\left|\boldsymbol{I} + \frac{\bar{\boldsymbol{V}}_{\pi_k}\boldsymbol{Q}_{\pi_k}\bar{\boldsymbol{V}}_{\pi_k}^{\dagger}}{\sigma_{\pi_k}^2}\right| \\ &+ \sum_{m=1}^{M} \mu_m\Big(\sum_{\pi_k\in\mathcal{G}} \mathrm{Tr}(\boldsymbol{\Gamma}_m\boldsymbol{B}_{\pi_k}\boldsymbol{Q}_{\pi_k}\boldsymbol{B}_{\pi_k}^{\dagger}) - P_m\Big) , \end{aligned} \tag{9}$$

where $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_M]$ with $\mu_m \geq 0$ is the Lagrange multipliers and the dual problem is given by

$$\max_{\boldsymbol{\mu}\geq 0} \quad q(\boldsymbol{\mu}) = \min_{\{\boldsymbol{Q}_{\pi_k}\succeq\boldsymbol{0}\}_{\pi_k\in\mathcal{G}}} L(\{\boldsymbol{Q}_{\pi_k}\}_{\pi_k\in\mathcal{G}}, \boldsymbol{\mu}) \tag{10}$$

Since the right-hand side of (8) is convex and satisfies the Slater's condition, the duality gap between the optimal objective of (8) and that of the dual problem (10) is zero. The Lagrange multipliers in the dual problem can be solved iteratively, where in each iteration the optimal $\boldsymbol{Q}_{\pi_k}$'s are solved with a given set of $\boldsymbol{\mu}$, and the Lagrange multipliers can be updated using subgradient-based method.

To solve the optimal $\boldsymbol{Q}_{\pi_k}$'s for a fixed set of μ_m's, the problem (10) can be further decomposed into K_0 independent subproblems,

$$\min_{\boldsymbol{Q}_{\pi_k}\succeq\boldsymbol{0}} \quad -\log\left|\boldsymbol{I} + \frac{\bar{\boldsymbol{V}}_{\pi_k}\tilde{\boldsymbol{A}}_{\pi_k}^{-1/2}\tilde{\boldsymbol{Q}}_{\pi_k}\tilde{\boldsymbol{A}}_{\pi_k}^{-1/2}\bar{\boldsymbol{V}}_{\pi_k}^{\dagger}}{\sigma_{\pi_k}^2}\right| + \mathrm{Tr}(\tilde{\boldsymbol{Q}}_{\pi_k})$$

where

$$\tilde{\boldsymbol{A}}_{\pi_k} = \boldsymbol{B}_{\pi_k}^{\dagger}\Big(\sum_{m=1}^{M}\mu_m\boldsymbol{\Gamma}_m\Big)\boldsymbol{B}_{\pi_k}, \ \tilde{\boldsymbol{Q}}_{\pi_k} = \tilde{\boldsymbol{A}}_{\pi_k}^{1/2}\boldsymbol{Q}_{\pi_k}\tilde{\boldsymbol{A}}_{\pi_k}^{1/2} .$$

The optimal solution $\tilde{\boldsymbol{Q}}_{\pi_k}$ can be obtained via SVD of $\bar{\boldsymbol{V}}_{\pi_k}\tilde{\boldsymbol{A}}_{\pi_k}^{-1/2}$ as follows,

$$\bar{\boldsymbol{V}}_{\pi_k}\tilde{\boldsymbol{A}}_{\pi_k}^{-1/2} = \tilde{\boldsymbol{F}}_{\pi_k}\boldsymbol{\Theta}_{\pi_k}\tilde{\boldsymbol{G}}_{\pi_k}^{\dagger} , \tag{11}$$

where $\boldsymbol{\Theta}_{\pi_k} = \mathrm{diag}(\theta_{\pi_k,1}, \ldots, \theta_{\pi_k,N_r})$ containing the singular values of $\bar{\boldsymbol{V}}_{\pi_k}\tilde{\boldsymbol{A}}_{\pi_k}^{-1/2}$ ordered in decreasing order. The optimal $\tilde{\boldsymbol{Q}}_{\pi_k}$ is given by the water-filling solution,

$$\tilde{\boldsymbol{Q}}_{\pi_k}^{*} = \tilde{\boldsymbol{G}}_{\pi_k}\Lambda_{\pi_k}\tilde{\boldsymbol{G}}_{\pi_k}^{\dagger} ,$$

where $\boldsymbol{\Lambda}_{\pi_k} = \mathrm{diag}(\lambda_{\pi_k,1}, \ldots, \lambda_{\pi_k,N_r})$ and $\lambda_{\pi_k,i} = \max\left(1 - 1/\theta_{\pi_k,i}^2, 0\right)$. Then, we have

$$\boldsymbol{Q}_{\pi_k}^{*} = \tilde{\boldsymbol{A}}_{\pi_k}^{-1/2}\tilde{\boldsymbol{Q}}_{\pi_k}^{*}\tilde{\boldsymbol{A}}_{\pi_k}^{-1/2} . \tag{12}$$

Although the user grouping stage selects K_0 users, which attempts to serve as many users as possible, the optimal power allocation algorithm may allocate zero power to some users if it is necessary to maximize the sum rate. In this case, the users with zero-power are actually harmful to the sum-rate performance, because these redundant users reduce the size of the null space for other users. Thus, as a final step, we remove the users with zero power from the user group, and the sum-rate is updated accordingly based on (9)-(12).

V. Simulation Results

In this section, simulation experiments are conducted to evaluate the performance of our proposed PBUS scheme. We consider that 4 APs are located in a line with an interval of 30 meters. There are $K/4$ users uniformly distributed around each AP within a radius of Y meters. We set each AP to have 4 antenna elements and each user to have 2 antenna elements. The quasi-static Rayleigh flat-fading channel model with a path-loss exponent of 3 and the noise power of -85 dBm is assumed. The transmit power of each AP is set to 23 dBm.

The sum-rate performance and computational complexity of the proposed PBUS algorithm are evaluated and compared with those of the following algorithms:

- Iterative power allocation for DPC with sum power constraint [12], which provides the upper bound of the sum-rate performance.
- Capacity-based user selection for BD (c-algorithm) [6], which is claimed to achieve the sum-rate close to that of the exhaustive search method.

- Frobenius norm-based user selection for BD (n-algorithm) proposed in [6].
- Upperbound-based user selection algorithm (u-algorithm) proposed in [8].

The greedy algorithms in [6] and [8] were originally proposed for the downlink transmission with single transmitter, which are extended to the targeted scenario with multiple cooperative APs in the simulation. The product square row norms-based algorithm [7] has been shown to exhibit similar sum-rate performance to the n-algorithm so we do not include it in our comparison. In addition, both algorithms rely on the c-algorithm to refine the user selection, which dominates the computational overhead. Therefore, the computation time required by [7] is also very similar to the n-algorithm.

A. Sum-rate performance

In Fig. 2, the achieved sum rate is illustrated as a function of the total number of users in the network at radius $Y = 30$ meters. There are 4 cooperative APs and thus the number of simultaneous users is limited by $K_0 = 8$. The sum-rate performance of the proposed PBUS with different values of L (i.e., $L = 4$ and $L = 8$) is evaluated. The sum rate achieved by DPC with sum power constraint [12] is deemed as the upper bound. The c-algorithm performs closer to DPC as the number of users increases. For the proposed PBUS scheme, larger L contributes to higher sum-rate performance due to the lower possibility of dropping good user combinations. PBUS with $L = 8$ achieves 6% higher sum-rate than n-algorithm. As the number of users grows, the sum-rate performance of PBUS with $L = 8$ gets closer to that of the c-algorithm. The sum rate achieved by the u-algorithm is much lower than that of other algorithms, especially for a large user population.

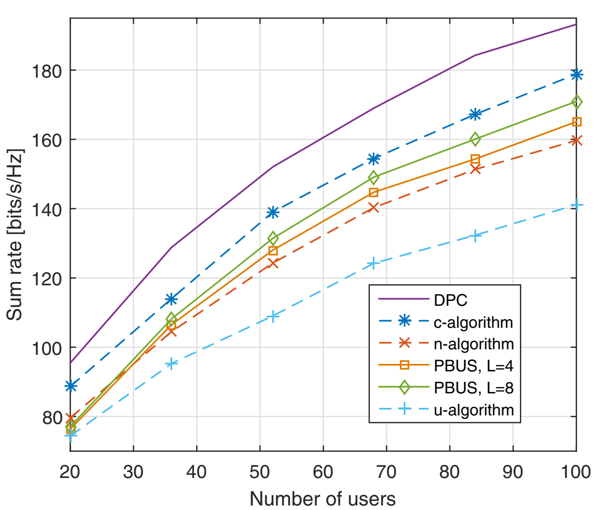

Fig. 2. Sum-rate as a function of number of users at $Y = 30$.

In Fig. 3, the achieved sum rate is shown as a function of the radius Y with 60 users. Smaller Y indicates higher average SNR at the receivers, which achieves higher sum rate. In the low average SNR region, different greedy algorithms perform similarly to each other. However, for higher average SNR region, our proposed algorithm with $L = 8$ performs closer to the upper bound, achieving about 10% higher sum rate than the n-algorithm. As discussed in [6], the Frobenius norm is a reliable metric to reflect the channel quality in the low SNR region, but it is not suitable for high SNR region. Finally, we note that the u-algorithm produces significantly lower sum rate than all other methods.

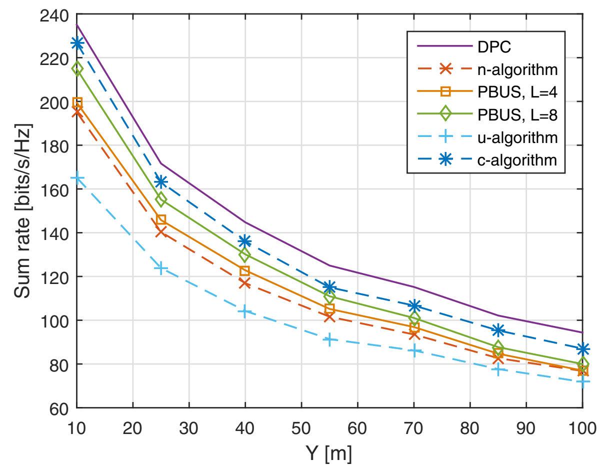

Fig. 3. Sum-rate as a function of the radius Y with $K = 60$.

B. Time complexity

We also evaluate the time complexity of our proposed algorithm and compare it to the c-algorithm, n-algorithm and u-algorithm. The algorithms are implemented in *MATLAB* and run on an i7-2700K Intel CPU rated at 3.5 GHz. The running time is counted by in-built *tic-toc* function in *Matlab*. In Fig. 4, the running time for single-round selection is plotted as a function of the number of users with 4 cooperative APs. In particular, there is no pre-calculated information available for our proposed PBUS. As shown in Fig. 4, c-algorithm consumes the highest running time among all algorithms. It requires tens of seconds for single-round selection even with a moderate number of users, which is too costly for practical systems. Although the running time of our proposed PBUS method increases with the size of the user population, it is still much lower than that of the n-algorithm even with up to 100 users and $L = 8$, which runs in about 1/3 of the time of n-algorithm. This is because n-algorithm uses the high-complexity c-algorithm in its finalization step, while our proposed algorithm simply performs the optimal power allocation to finalize the active user set. Moreover, the running time varies in a narrow range when L changes, although the number of generated candidate groups varies dramatically. For example, by increasing the value of L from 4 to 8, at most 120 additional candidate groups will be generated during the grouping stage, which only consumes 20% more running time. Although the u-algorithm runs the fastest, it lacks the ability to exploit the achievable sum rate as shown in Fig. 2. In addition, for a small to moderate number of users, our

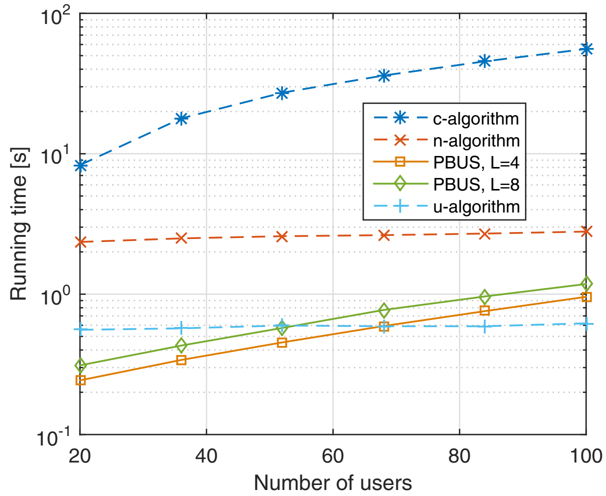

Fig. 4. Running time as a function of number of users

proposed algorithm can achieve higher sum-rate with even lower complexity as compared to the u-algorithm.

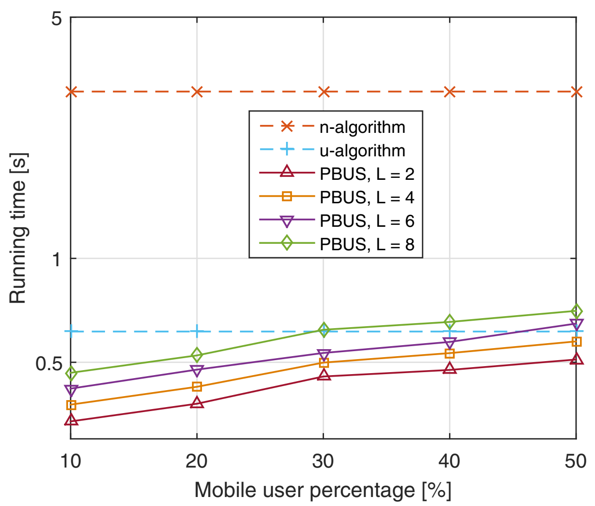

Fig. 5. Update time as a function of mobile user percentage

Finally, we investigate the update efficiency of the algorithms when only some users experience channel changes. We assume there are 4 cooperative APs and 100 users, and some of the users are mobile. For conventional algorithms, the update procedure is exactly the same as a single-round selection by completely re-computing the selection metric and constructing the selected user group, such as c-algorithm, n-algorithm and u-algorithm. As discussed in Sec. III-E, our proposed PBUS can reuse the calculated pairwise fitness metric and the constructed binary tree for static users as much as possible. In Fig. 5, the running time for update is plotted as a function of the mobile user percentage, which varies from 10% to 50%. The update time for n-algorithm and u-algorithm is unaffected as the mobile user percentage changes, because they lack the ability to reduce computational complexity even if only a few channels change. For our proposed PBUS, the update time can be reduced via the partial reuse of the pre-calculated information of static users, which is especially visible when the mobile user percentage is small. For example, with 10% mobile users, PBUS can update the user selection in 1/2 of the time as compared to Fig. 4 even with a maximum value of $L = 8$, which is 25% lower than u-algorithm.

Acknowledgment

This research was supported in part by the U.S. National Science Foundation through Awards CNS–1319455 and CNS–1513884.

VI. Conclusion

In this paper, a novel user selection scheme, referred to as PBUS, for block diagonalization (BD) in dense wireless networks with AP cooperation was presented. Different from conventional greedy algorithms, the proposed method can store multiple high-performance user groups in a binary tree based on the pairwise evaluation mechanism. It reduces the probability of missing good user groups while also having lower computational time compared to conventional methods. The proposed method is also shown to allow tradeoffs between sum-rate performance and computational complexity for a moderate to large number of users.

References

[1] D. Gesbert, S. Hanly, H. Huang, S. Shamai Shitz, O. Simeone, and W. Yu, "Multi-cell MIMO cooperative networks: A new look at interference," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 9, pp. 1380–1408, 2010.

[2] H. Rahul, S. S. Kumar, and D. Katabi, "Megamimo: Scaling wireless capacity with user demand," in *Proc. ACM SIGCOMM*, 2012.

[3] H. V. Balan, R. Rogalin, A. Michaloliakos, K. Psounis, and G. Caire, "AirSync: Enabling distributed multiuser MIMO with full spatial multiplexing," *IEEE/ACM Trans. Netw.*, vol. 21, no. 6, pp. 1681–1695, Dec. 2013.

[4] Q. H. Spencer, A. L. Swindlehurst, and M. Haardt, "Zero-forcing methods for downlink spatial multiplexing in multiuser MIMO channels," *IEEE Trans. Signal Process.*, vol. 52, no. 2, pp. 461–471, Feb 2004.

[5] L.-U. Choi and R. D. Murch, "A transmit preprocessing technique for multiuser MIMO systems using a decomposition approach," *IEEE Trans. Wireless Commun.*, vol. 3, no. 1, pp. 20–24, 2004.

[6] Z. Shen, R. Chen, J. G. Andrews, R. W. Heath, and B. L. Evans, "Low complexity user selection algorithms for multiuser MIMO systems with block diagonalization," *IEEE Trans. Signal Process.*, vol. 54, no. 9, pp. 3658–3663, Sept 2006.

[7] L.-N. Tran, M. Bengtsson, and B. Ottersten, "Iterative precoder design and user scheduling for block-diagonalized systems," *IEEE Trans. Signal Proc.*, vol. 60, no. 7, pp. 3726–3739, Jul. 2012.

[8] X. Zhang and J. Lee, "Low complexity MIMO scheduling with channel decomposition using capacity upperbound," *IEEE Trans. Commun.*, vol. 56, no. 6, pp. 871–876, June 2008.

[9] R. Kudo, Y. Takatori, K. Nishimori, A. Ohta, and S. Kubota, "User selection method for block diagonalization in multiuser MIMO systems," in *Proc. IEEE GLOBECOM*, Nov. 2007, pp. 3295–3300.

[10] H. Lou, M. Ghosh, P. Xia, and R. Olesen, "A comparison of implicit and explicit channel feedback methods for MU-MIMO WLAN systems," in *Proc. IEEE PIMRC*, Sept. 2013, pp. 419–424.

[11] M. Ge and D. Blough, "High-throughput and fair scheduling for access point cooperation in dense wireless networks," in *Proc. IEEE WCNC*, March 2017.

[12] N. Jindal, W. Rhee, S. Vishwanath, S. A. Jafar, and A. Goldsmith, "Sum power iterative water-filling for multi-antenna gaussian broadcast channels," *IEEE Trans. Info. Theory*, vol. 51, no. 4, pp. 1570–1580, 2005.