High Throughput and Fair Scheduling for Multi-AP Multiuser MIMO in Dense Wireless Networks

Mengyao Ge and Douglas M. Blough

***Abstract*— This paper considers the fair scheduling problem for dense wireless networks with access point cooperation and multiple-input-multiple-output (MIMO) links. The problem is to maximize the aggregate throughput subject to a fairness constraint that is general enough to capture many different fairness objectives. We formally specify a non-convex optimization problem that captures all aspects of the problem setting, and we propose two algorithms to approximate its solution. The first algorithm jointly optimizes the selection of user sets, MIMO precoders, and assignment of user sets to time slots. The second algorithm separately optimizes first user sets and MIMO precoders and second assignment of user sets to time slots. The first algorithm guarantees perfect fairness and produces a local optimum or a saddle point for aggregate throughput at a fairly high computational cost. The second algorithm also guarantees perfect fairness and produces optimal aggregate throughput for a given set of (possibly non-optimal) user sets while having lower computational complexity. The second algorithm also has a parameter that allows throughput and fairness to be traded off for situations where maximizing throughput is critical and approximate fairness is acceptable. Analyses are complemented by simulation results, which show that: 1) the first algorithm produces significantly higher aggregate throughput than known approaches with a running time that is practical for scenarios with up to 50 users and 2) the second algorithm produces aggregate throughput that is very close to existing heuristics while having significantly lower running time.**

***Index Terms*— Scheduling, multiuser MIMO, AP cooperation, dense wireless networks, fairness.**

I. Introduction

OVER the last decade, wireless data traffic has experienced rapid growth driven by the increasing number of wireless devices and bandwidth-hungry applications. Improvements in wireless local access network (WLAN) technology and the dense deployment of access points (APs) are expected to help accommodate traffic demands. However, dense deployments with many nearby APs sharing the limited unlicensed spectrum lead to high co-channel interference, which can limit the overall performance improvement. Traditional techniques, such as assigning orthogonal channels to different APs and assigning non-overlapping time slots to different users for 802.11-based WLANs, at best equally divide the limited bandwidth among users. A potential way to break the bottleneck of performance in dense wireless networks is AP cooperation combined with advanced multiuser multiple-input-multiple-output (MIMO) processing techniques [1].

Coordinated multipoint transmission, which is a form of distributed MIMO, has attracted significant research interest because of its potential to increase wireless network throughput [2]. This approach is well suited to dense enterprise networks with clusters of closely deployed APs [3]. A common scenario is that APs share a network gateway with one Internet connection. In this scenario, multiple APs can cooperate to control lower-layer parameters and optimize performance and fairness. However, to reap the full benefits of this approach, advanced multiuser MIMO techniques, which can perform a combination of spatial multiplexing and interference suppression, need to be investigated.

In this paper, we mainly focus on enterprise environments, where most users are expected to be stationary for significant periods of time with intermittent shorter periods of mobility. Since there are many users sharing the limited resources of the wireless network, a key problem is MIMO link scheduling, i.e. determining how to activate MIMO links for a given scheduling period to meet organizational requirements. In general, throughput and fairness are two fundamental objectives in wireless networks that cannot be maximized simultaneously. This motivates the investigation of tradeoffs between the two objectives, where a common approach is to maximize performance subject to fairness constraints. We adopt the widely-used notion of time-based fairness [4]–[6], which avoids the performance anomaly associated with rate-based fairness in multi-rate wireless networks [7]. The basic idea is to allocate equal time to each user and the bandwidth of each user is then dependent on the number of users and its own data rate [5].

The specific problem we consider herein is scheduling users to achieve high aggregate performance while maintaining fairness and operating across a small group of APs that are assigned to the same carrier frequency and employing multi-user MIMO. Our contributions are as follows:

1) we provide a novel mathematical formulation of the maximum sum rate multi-slot scheduling problem with fairness constraints in the multi-AP MIMO setting,
2) although the formulated optimization problem is too complex to solve directly, we develop a series of transformations that lead to the first approximation algorithm for this type of problem that jointly optimizes selection of user sets, MIMO precoders and assignment of user sets to time slots,

Manuscript received August 21, 2017; revised February 1, 2018, June 11, 2018, and August 5, 2018; accepted August 6, 2018; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor C. F. Chiasserini. Date of publication September 13, 2018; date of current version October 15, 2018. This work was supported by the National Science Foundation under Award CNS-1319455 and Award CNS-1513884. *(Corresponding author: Mengyao Ge.)*

The authors are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0765 USA (e-mail: mengyaoge@gmail.com; doug.blough@ece.gatech.edu).

Digital Object Identifier 10.1109/TNET.2018.2867582

3) we also develop a novel and more efficient two-stage heuristic algorithm that separately optimizes selection of user sets with MIMO precoders and assignment of those sets to time slots,
4) we demonstrate that, for a given (but possibly non-optimal) set of user combinations, our two-stage heuristic produces a near-optimal schedule in terms of sum rate performance while achieving the fairness constraint, and
5) we provide detailed simulation results, which show that:
   - our proposed multi-slot scheduling strategies exhibit significant performance gains compared to slot-by-slot scheduling strategies,
   - our joint optimization algorithm produces significantly higher sum rate than existing approaches, handles at least 50 users across 2–6 APs, and achieves very close to perfect fairness, and
   - our two-stage heuristic algorithm has significantly lower running time than existing heuristic algorithms while achieving nearly the same sum rate and near-perfect fairness.

The rest of the paper is organized as follows. The problem setting is described in Section II. Related work is discussed in Section III. The detailed system model and problem formulation are presented in Section IV. We describe the joint scheduling algorithm in Section V and propose a two-stage heuristic algorithm in Section VI. Numerical results are presented in Section VII and Section VIII concludes.

II. Problem Setting

Our problem formulation and solutions are targeted at dense enterprise wireless networks of the kind commonly deployed in universities and other large institutions. In this setting, there can be hundreds of users and tens of APs within a fairly small area. For example, in the classroom wing of the Klaus Building at Georgia Tech, there are 4–8 APs in each large classroom. Although APs in the same classroom are mostly operating on different channels, adjoining classrooms typically have active APs operating on the same channel, so that it is common for 3 or more APs on the same channel to be within interference range of each other. These APs are often servicing several hundred users simultaneously. In the remainder of the building, there are offices, labs, conference rooms, and many open spaces where student groups meet to collaborate on projects. In these areas, it would not be uncommon to have 100 or more users within interference range of each other and within range of multiple APs operating on the same channel.

One key aspect of this problem setting that differentiates it from much of the prior work on multi-user MIMO, which has targeted cellular networks, is that the vast majority of users are stationary for minutes at a time. While there might be a few users walking through a hallway and accessing the enterprise wireless network, most are sitting in classrooms, offices, labs, conference rooms, or tables working. The highly stationary nature of the problem setting provides opportunities to better optimize the aggregate network performance and achieve fairness by explicitly scheduling users across multiple time slots over scheduling periods on the order of tens of seconds. This provides an opportunity to change both the problem formulation and the solutions, as compared to prior work that targeted highly dynamic settings and that, by necessity, focused on time slot by time slot scheduling and optimization.

Our quasi-stationary assumption for office, classroom, and lab environments is confirmed by measurement studies of these environments reported in [8] and [9]. Among the different scenarios considered in both of these measurement studies were the case where transmitter and receiver are stationary and there are moving scatterers. Both papers found that, with a modest amount of environmental mobility, channels were very similar to a fully stationary scenario with no environmental mobility. For the studied environmental mobility scenario, [9] reported channel coherence values of above 0.95 for intervals of seconds and stated "... channels are seemingly indefinitely stable other than brief periods where the channel is altered."

While it is also necessary to handle the small percentage of mobile users that exist in this problem setting, it is possible to detect users with rapidly varying channels and partition them into separate scheduling slots [10]. Herein, we assume that stationary and mobile users are partitioned into different scheduling slots and we focus on optimizing performance and achieving fairness for the stationary users. Since the percentage of mobile users is quite low in the targeted environments, they have only a small impact on the overall results.

III. Related Work

Prior work has considered various scheduling problems with fairness constraints in multi-user MIMO settings. However, the prior work almost exclusively considered a slot-by-slot scheduling paradigm.[1] Our work differs in that it computes an explicit multi-slot schedule of user transmissions over a scheduling period on the order of tens of seconds. This approach is enabled by the problem setting considered herein (see discussion in previous section) where the focus is on the large number of stationary users that exist in many enterprise wireless settings. In contrast, prior work was primarily targeted at cellular networks where a high percentage of users are mobile. Due to the highly dynamic environment in cellular networks and the need for accurate channel state information (CSI) for MIMO processing, slot-by-slot scheduling that uses current CSI is necessary in that context. The ability to schedule transmissions over multiple slots in our more stationary setting provides greater opportunities for optimizing performance and fairness. To our knowledge, ours is the first work that provides a rigorous mathematical formulation of the maximum sum rate problem with fairness constraints in the multi-user MIMO *multi-slot scheduling* context.

Slot-by-slot scheduling approaches select a user set that optimizes a utility function at each slot. A common choice of utility function is the weighted sum rate (WSR) of selected users and the corresponding single-slot scheduling problem is referred to as the maximum weighted sum rate (MWSR) problem, which has been researched extensively, e.g. [11]–[13]. When the number of users is greater than can be supported in one slot meaning that scheduling is required, the

[1]This is sometimes referred to as "one-shot scheduling".

MWSR problem has been addressed in one of two ways: (1) by jointly performing user selection and utility maximization, e.g. [14]–[16] or (2) by decoupling user selection and sum rate maximization, e.g. [17]–[20]. These approaches are sometimes referred to as the direct and indirect approaches, respectively [21]. An alternative way to solve these problems is through stochastic optimization methods, e.g. [22]–[25].

Fairness can be accommodated in the single-shot scheduling problem through the choice of utility function. The classic example of this is setting the utility function to be the sum of the log rates of each user, which provides proportional fairness, e.g. [26]–[28]. Another common fairness metric is max-min fairness, which can also be specified through utility functions, e.g. [24], [28]. Another utility function chooses users according to the ranking of their current channel quality among the channel qualities they have experienced over a specified length of time [29]. This provides some minimal fairness since users will definitely be scheduled when their channel qualities are best among their recent history.

Specific works that consider fairness in scheduling with multiple APs and multiple antennas per AP include [28] and [30]–[35]. In [28] and [30], Huh *et al.* consider a general utility function and illustrate how it can be used to model either proportional fairness or max-min fairness. In [32] and [33], proportional fairness is assumed, while in [31] and [34], variations of max-min fairness are considered. In [35], max-min SINR is achieved in each slot and users are scheduled into slots so as to achieve an overall performance objective. With the exception of [35], which is discussed in more detail below, all of these works adopt the slot-by-slot scheduling approach.

Fairness cannot be achieved in one slot when the number of users exceeds what a slot can support. Therefore, in slot-by-slot scheduling approaches, fairness is an emergent property that arises over a sequence of scheduling operations. These single-slot utility-based fairness approaches implement a *scheduling policy* at each time slot, whereas the multi-slot scheduling approach taken herein uses a *scheduling algorithm* to produce an explicit transmission schedule over an entire scheduling period. By necessity, single-slot approaches merge performance and fairness criteria into a single utility function. In the multi-slot approach, fairness can be achieved by scheduling users in different slots and so it is possible to separate performance and fairness criteria. Our optimization problem, presented in Section IV, does this by having as an objective the maximization of sum rate and specifying fairness criteria as separate constraints on the solution space.

Several prior works have considered the fairness issue in WLANs specifically, which is a similar network context to ours, either with multi-user MIMO with a single AP [36], [37], or with multiple APs but a single user per AP [4], [38], [39]. The works of [36] and [37] primarily consider the problem of user selection to maximize sum rate but [36] enforces a minimal fairness constraint by alternating users selected as the first user for a transmission slot while [37] states that their selection metric can be adapted to incorporate fairness but does not evaluate that aspect in detail. Both [38] and [39] consider how to associate users to APs to achieve fairness objectives but do not consider the scheduling aspect. Finally, [4] considers scheduling across multiple APs while accounting for interference but only with a single user per AP. None of the above-cited works consider a scenario with both multi-user MIMO and multiple APs, as we address in this paper.

The only prior works of which we are aware that consider the *multi-slot scheduling* problem with multiple APs and multiple antennas per AP are [6], [35], and [40]. In [35], Dartmann *et al.* present an optimization formulation over an entire multi-slot schedule. However, the formulation is restricted to MISO links and only allows interference suppression between APs, rather than full cooperation. In [6], the APs only coordinate to perform interference suppression while in [40], full cooperation among APs (joint data processing) is considered. The algorithms of [6] and [40] both generate a number of candidate single-time-slot solutions and then optimize the schedule using only those pre-determined candidates. Thus, they do not formulate a full optimization problem that considers both user selection within each time slot and scheduling of users across multiple time slots as we do herein.

IV. System Model and Problem Description

As discussed in detail in Section II, we consider a scenario in which single-hop wireless networks are densely deployed over a region, where the areas served by different access points (APs) can overlap, and most users are stationary for significant periods of time with intermittent shorter periods of mobility. The durations of stationary periods are expected to be on the same order as the scheduling period, which is tens of seconds or less for the scenario considered herein. We focus primarily on optimizing downlink transmissions since in typical indoor environments 80% or more of the traffic is on the downlink.

A. AP Cooperation

To address the interference problem in overlapping single-hop wireless networks, we consider the use of advanced MIMO techniques involving AP cooperation and coordination of communications across cells, which are envisioned to be widely used in next-generation wireless technologies. The complexity of coordination, backhaul limitations, and computational limits for scheduling will impose a relatively small upper bound on the number of APs that cooperate. Hence, we assume a small number of APs that are near each other and operate on the same channel are grouped into a cluster, as shown in Fig.1.[2] In the case of a large enterprise wireless network, the APs can be grouped into multiple clusters, where the APs within one cluster cooperate with each other. In this paper, we consider only the operation within one cluster.

We assume that there is a single entity for each cluster, which has access to CSI and the data signals intended for all users and that computes the overall schedule and the precoding and combining weights for all APs and users active within each slot. This entity could be one designated AP in the

[2]Our techniques can be applied independently across as many orthogonal channels as are available in a given wireless deployment.

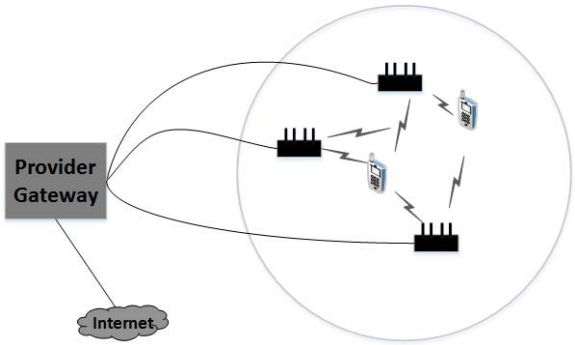

Fig. 1. An example of the clustered overlapping APs.

cluster or a network controller connected to all APs within a cluster.

B. Physical-Layer Model

Assume there are M cooperative access points (APs) in one cluster, where the m^{th} AP is equipped with $N_{tx,m}$ antennas. We assume that there are K users with $N_{rx,k}$ antennas for the k^{th} user. The user set is denoted by $\mathcal{K} = \{1, \ldots, K\}$. Let $N_{tx} = \sum_{m=1}^{M} N_{t,m}$ and $N_{rx} = \sum_{k=1}^{K} N_{r,k}$ be the total numbers of antennas at the AP and receiver side, respectively. The matrix of complex channel gains between the cooperative APs and the antennas of the k^{th} user is denoted by $\boldsymbol{H}_k \in \mathbb{C}^{N_{rx,k} \times N_{tx}}$. We assume that one scheduling period contains T time slots, each of which has the same duaration. The data vector $\boldsymbol{x}(t) = \left[\boldsymbol{x}_1(t)^T, \ldots, \boldsymbol{x}_K(t)^T\right]^T$ is jointly precoded by the M APs using the linear precoding matrix $\boldsymbol{V}(t) = [\boldsymbol{V}_1(t), \ldots, \boldsymbol{V}_K(t)]$ for time slot t. $\boldsymbol{x}_k(t) \in \mathbb{C}^{N_{rx,k}}$ is the transmit signal vector for receiver k, and $\boldsymbol{x}_k(t)$ is assumed to be independently encoded Gaussian codebook symbols with $\mathbb{E}[\boldsymbol{x}_k(t)\boldsymbol{x}_k(t)^\dagger] = \boldsymbol{I}$, where $(\cdot)^\dagger$ is the conjugate transpose of $(\cdot)$. It is assumed that the k^{th} user has $N_{rx,k}$ data streams, although some of the streams can have a rate of zero. $\boldsymbol{V}_k(t) \in \mathbb{C}^{N_t \times N_{rx,k}}$ is the partition of $\boldsymbol{V}(t)$ applied at the APs to precode the signals of user k.

The received vector at user k for time slot t is given by

$$\boldsymbol{y}_k(t) = \boldsymbol{H}_k \boldsymbol{V}_k(t)\boldsymbol{x}_k(t) + \sum_{l=1, l\neq k}^{K} \boldsymbol{H}_k \boldsymbol{V}_l(t)\boldsymbol{x}_l(t) + \boldsymbol{n}_k, \quad (1)$$

where $\boldsymbol{n}_k$ is the vector of Gaussian noise at the k^{th} user with covariance matrix $\boldsymbol{R}_{n_k}$. The corresponding covariance matrix of the received interference plus noise is given by

$$\boldsymbol{R}_{\bar{k}}(t) = \sum_{l=1, l\neq k}^{K} \boldsymbol{H}_k \boldsymbol{V}_l(t)\boldsymbol{V}_l(t)^\dagger \boldsymbol{H}_k^\dagger + \boldsymbol{R}_{n_k}. \quad (2)$$

Assume the received signal is equalized using the linear combiner $\boldsymbol{U}_k(t) \in \mathbb{C}^{N_{rx,k} \times N_{rx,k}}$. The received signal of the k^{th} receiver is given by $\hat{\boldsymbol{x}}_k(t) = \boldsymbol{U}_k(t)^\dagger \boldsymbol{y}_k(t)$.

As shown in [41], with linear precoder applied at the transmitter side, the maximum achievable rate for the k^{th} user over time slot t is then given by

$$R_k(t) = \log_2 \left| \boldsymbol{I} + \boldsymbol{R}_{\bar{k}}(t)^{-1} \boldsymbol{H}_k \boldsymbol{V}_k(t) \boldsymbol{V}_k(t)^\dagger \boldsymbol{H}_k^\dagger \right|. \quad (3)$$

Moreover, the post-processing data rate for the k^{th} user after the linear combining $\boldsymbol{U}_k$ is given by

$$\hat{R}_k(t) = \log_2 \left| \boldsymbol{I} + \frac{\boldsymbol{U}_k(t)^\dagger \boldsymbol{H}_k \boldsymbol{V}_k(t) \boldsymbol{V}_k(t)^\dagger \boldsymbol{H}_k^\dagger \boldsymbol{U}_k(t)}{\boldsymbol{U}_k(t)^\dagger \boldsymbol{R}_{\bar{k}}(t) \boldsymbol{U}_k(t)} \right|. \quad (4)$$

Herein, we assume the linear MMSE (LMMSE) receiver is used. In [12], it is shown that the LMMSE receiver achieves the optimal rate, i.e. it has $\hat{R}_k(t) = R_k(t)$.

The MSE covariance matrix of the k^{th} user is

$$\boldsymbol{E}_k = \mathbb{E}\left[(\hat{\boldsymbol{x}}(t) - \boldsymbol{x}_k(t))(\hat{\boldsymbol{x}}(t) - \boldsymbol{x}_k(t))^\dagger\right]. \quad (5)$$

C. Scheduling Problem Formulation

We aim to develop a fair and high-throughput schedule over T time slots, where the channels are assumed to be stationary during one scheduling period. Let $\boldsymbol{b} = \{b_1, \ldots, b_K\}$, where the k^{th} element of $\boldsymbol{b}$ stands for the target bandwidth fraction of the k^{th} user and $\sum_{k=1}^{K} b_k = 1$. Different fairness objectives can be achieved through different choices of $\boldsymbol{b}$. The scheduling problem is formulated to maximize the throughput for one scheduling period, while guaranteeing the fairness objective among users. Formally, the problem can be stated as:

$$\begin{aligned} \max_{\substack{\{\boldsymbol{V}_k(t)\}_{k\in\mathcal{K}} \\ t\in\mathcal{T}}} \quad & \sum_{t=1}^{T}\sum_{k=1}^{K} R_k(t) \\ s.t.\ \ & \mathrm{Tr}(\Gamma_m \sum_{k=1}^{K} \boldsymbol{V}_k(t)\boldsymbol{V}_k(t)^\dagger) \le P_m, \quad \forall m,\ \forall t \\ & \sum_{t=1}^{T} R_k(t) = b_k \sum_{t=1}^{T}\sum_{k=1}^{K} R_k(t), \quad \forall k \in \mathcal{K} \end{aligned} \quad (6)$$

where P_m is the maximum transmit power of the m^{th} AP and $\mathrm{Tr}(\cdot)$ denotes the trace of a matrix $(\cdot)$. Note that this formulation considers the problem of maximizing sum rate across an entire schedule of T time slots, i.e. it is significantly different than the classic maximum sum rate problem, which focuses only optimization of one single time slot.

In the above formulation, a diagonal matrix $\boldsymbol{\Gamma}_m \in \mathbb{R}^{N_t \times N_t}$ is introduced for each AP, in order to select the partition of precoding matrix $\boldsymbol{V}$ applied at the m^{th} AP. Thus, $\boldsymbol{\Gamma}_m$ has ones on the diagonal elements corresponding to the antennas of the m^{th} AP, and zeros in other positions. The fairness constraints require that the achieved throughput of each user should be proportional to its target bandwidth fraction. For example, rate-based fairness can be achieved by assigning $b_k = 1/K$, $\forall k$.

The formulated problem is non-convex w.r.t. $V_k(t)$, due to the non-convexity of the function $R_k(t)$. It can be proved that the formulated problem has at least one feasible solution when $T \geq K$, which can be found by activating one user for each time slot and setting the users' data rates so that they meet their target bandwidth fractions with respect to the sum rate over all users. The solution to problem (6) will force some users to have $R_k(t) = 0$ by allocating zero power to these users in a certain time slot, if it is necessary to maximize the throughput. Thus, we do not explicitly label which users are active in each time slot but this is implicit in the optimized rates that are produced by our algorithms.

To our knowledge, this is the first complete mathematical formulation of an optimized multi-slot scheduling problem with fairness constraints for multi-AP MIMO networks.

V. Scheduling With Joint Optimization

In this section, we propose a scheduling algorithm to solve the formulated problem, which jointly determines the active user subset for each time slot and their MIMO weights. In order to make the problem tractable, we propose several transformations to Problem (6) that facilitate its solution.

The fairness constraints dictate that $\sum_{t=1}^{T} R_k(t) = b_k \sum_{k=1}^{K}\sum_{t=1}^{T} R_K(t),\ \forall k$. Since optimizing with inequality constraints is easier than with equality constraints such as these, we relax the problem in the following manner. We introduce an auxiliary variable c, which satisfies $c \le \sum_{k=1}^{K}\sum_{t=1}^{T} R_k(t)$. Then, the equality constraints can be converted into a set of inequality constraints, i.e., $b_k c \le \sum_{t=1}^{T} R_k(t),\ \forall k$. Thus, the optimization problem (6) can be reformulated as,

$$\begin{aligned} \max_{c,\{\boldsymbol{V}_k(t)\}_{\substack{k\in\mathcal{K}\\ t\in\mathcal{T}}}} \quad & c \\ s.t.\ \mathrm{Tr}(\Gamma_m \sum_{k=1}^{K} \boldsymbol{V}_k(t)\boldsymbol{V}_k(t)^\dagger) \le P_m, \quad & \forall m,\ \forall t \\ \sum_{t=1}^{T} R_k(t) \ge b_k c, \quad & \forall k \in \mathcal{K}, \end{aligned} \tag{7}$$

Since we have $c \le \sum_{k=1}^{K}\sum_{t=1}^{T} R_k(t)$, a solution of maximizing c in Problem (7) is a solution for maximizing $\sum_{k=1}^{K}\sum_{t=1}^{T} R_k(t)$ in Problem (6). Note also that the constraint $c \le \sum_{k=1}^{K}\sum_{t=1}^{T} R_k(t)$ is implicitly satisfied when the K constaints $\sum_{t=1}^{T} R_k(t) \ge b_k c,\ \forall k \in \mathcal{K}$ are met, since we have $\sum_{k=1}^{K} b_k = 1$.

Lemma 1: If we have a locally optimal solution $\boldsymbol{X} = (\{\boldsymbol{V}_k(1)\}_{k\in\mathcal{K}}, \cdots, \{\boldsymbol{V}_k(T)\}_{k\in\mathcal{K}})$ *to problem (7), it is also locally optimal for problem (6).*

Proof: Since Problem (7) and Problem (6) share the same power constraint, let $L(c, \boldsymbol{X}, \boldsymbol{\lambda})$ be the Lagrangian of problem (7) and $\hat{L}(\boldsymbol{X}, \boldsymbol{\lambda})$ be the Lagrangian of problem (6) without considering the power constraint,

$$\begin{aligned} &L(c, \{\boldsymbol{V}_k(t)\}_{\substack{k\in\mathcal{K}\\ t\in\mathcal{T}}}, \boldsymbol{\lambda}) \\ &= \left(\sum_{k=1}^{K} b_k\lambda_k - 1\right) c - \sum_{k=1}^{K}\lambda_k \sum_{t=1}^{T} R_k(t) \\ &\hat{L}(\{\boldsymbol{V}_k(t)\}_{\substack{k\in\mathcal{K}\\ t\in\mathcal{T}}}, \boldsymbol{\lambda}) \\ &= -\sum_{k=1}^{K}\sum_{t=1}^{T} R_k(t) + \sum_{k=1}^{K}\lambda_k (b_k \sum_{k=1}^{K}\sum_{t=1}^{T} R_k(t) - \sum_{t=1}^{T} R_k(t)) \end{aligned}$$

where $\boldsymbol{\lambda} = \{\lambda_1, \ldots, \lambda_K\}$ is the vector of Lagrange multipliers. If $\boldsymbol{X} = (\{\boldsymbol{V}_k(1)\}_{k\in\mathcal{K}}, \cdots, \{\boldsymbol{V}_k(T)\}_{k\in\mathcal{K}})$ is an optimal point of problem (7), i.e., $\nabla_c f = 0$, $\nabla_{\boldsymbol{X}} f = \boldsymbol{0}$ and $\nabla_{\boldsymbol{\lambda}} f = \boldsymbol{0}$, we will have $\nabla_{\boldsymbol{X}} g = \boldsymbol{0}$ and $\nabla_{\boldsymbol{\lambda}} g = \boldsymbol{0}$. With the fact that $\boldsymbol{X}$ achieves a local maximum of c and $c \le \sum_{k=1}^{K}\sum_{t=1}^{T} R_k(t)$, $\boldsymbol{X}$ is also an optimal point for problem (6). □

The solution to problem (7) involves $K \times T$ variables, i.e., the precoder $\boldsymbol{V}_k(t)$ for each user in each time slot.

TABLE I

Alternating Optimization of Multiuser Scheduling

1. Initialize $I = 0$ and $\boldsymbol{V}_k(t)\ \forall k \in \mathcal{K}, \forall t \in \mathcal{T}$ such that $\mathrm{Tr}(\Gamma_m \sum_{k=1}^{K} \boldsymbol{V}_k(t)\boldsymbol{V}_k(t)^\dagger) \le P_m, \forall m, \forall t \in \mathcal{T}$
2. ***while*** $I \le I_{max}$
3. $\quad I \leftarrow I + 1$
4. $\quad$ ***for*** t from 1 to T
5. $\quad\quad$ update $\boldsymbol{V}_k(t)$ and c for $k \in \mathcal{K}$ by solving problem (8)
6. $\quad$ ***Qiut if*** $|\sum_{t=1}^{T}\sum_{k=1}^{K} R_k^I(t) - \sum_{t=1}^{T}\sum_{k=1}^{K} R_k^{I-1}(t)| \le \varepsilon$

Since we assume a dense network setting, the number of users K can be quite large. T will also have to be fairly large in order to have enough time slots to accommodate all of the users and meet the fairness constraints.[3] The number of variables will therefore make the solution of Problem (7) quite complex. To reduce the number of variables, Problem (7) can be further decomposed into T subproblems and solved by alternating optimization. For a given time slot t, the t^{th} subproblem can be formulated as follows:

$$\begin{aligned} \max_{c,\{\boldsymbol{V}_k(t)\}_{k\in\mathcal{K}}} \quad & c \\ s.t.\ \mathrm{Tr}(\Gamma_m \sum_{k=1}^{K} \boldsymbol{V}_k(t)\boldsymbol{V}_k(t)^\dagger) \le P_m, \quad & \forall m \\ R_k(t) \ge b_k c - \sum_{s=1, s\neq t}^{T} R_k(s), \quad & \forall k \in \mathcal{K} \end{aligned} \tag{8}$$

The T subproblems can be solved iteratively to find a suboptimal solution to the problem (7). The iterative algorithm is summarized in Table I. In each iteration, it solves the problem (8) for each time slot sequentially. Since the problem (7) is non-convex, a global optimum is not guaranteed. However, since alternating optimization provides monotonously non-decreasing solutions c to Problem (7) and variable c is upper bounded, the alternating optimization solution will converge to a suboptimal solution of Problem (7).

The convergence of the alternating algorithm is proved as follows. Let

$$\boldsymbol{X}^{(i,t)} = \Big\{\boldsymbol{V}_k(1)^{(i)}, \ldots, \boldsymbol{V}_k(t)^{(i)}, \boldsymbol{V}_k(t+1)^{(i-1)}, \ldots, \boldsymbol{V}_k(T)^{(i-1)}\Big\}_{k\in\mathcal{K}}$$

be the optimized precoders after solving the t^{th} subproblem during the i^{th} iteration, which corresponds to an optimal value $c(\boldsymbol{X}_{i,t})$. The solution $\boldsymbol{X}_{i,t}$ and $c(\boldsymbol{X}_{i,t})$ are taken as the initial values for solving the $(t+1)^{\text{th}}$ problem during the i^{th} iteration. The maximization process of the $(t+1)^{\text{th}}$ problem leads to $c(\boldsymbol{X}_{i,t+1}) \ge c(\boldsymbol{X}_{i,t})$. Thus, we have $c(\boldsymbol{X}_{i+1,t}) \ge c(\boldsymbol{X}_{i,t})$. Since c is upper bounded, we will have $\boldsymbol{X}_{r,t+1} = \boldsymbol{X}_{r,t}$ for sufficient number of iterations.

Moreover, it is evident that each subproblem is a DCP (difference of convex functions programming) problem. It has been proved that strong duality holds for any DCP problem [42]. In other words, the optimal value for DCP is the same as the optimal value for its dual. The dual problem

[3] We consider up to 50 users and up to 100 time slots in our simulations.

of each subproblem is convex and can be solved using subgradient method, which will be elaborated later. The optimal point $\boldsymbol{V}(t)^*$ for the t^{th} subproblem with zero duality gap indicates the KKT optimality condition that

$$\frac{\partial L(c,\{\boldsymbol{V}_k(t)\}_{k\in\mathcal{K}},\boldsymbol{\lambda})}{\partial \boldsymbol{V}_k(t)}|_{\boldsymbol{V}_k(t)^*}=\mathbf{0},\quad \forall k.$$

The optimal points for the T subproblems form a stationary point for problem (7) such that

$$\frac{\partial L(c,\{\boldsymbol{V}_k(t)\}_{\substack{k\in\mathcal{K}\\ t\in\mathcal{T}}},\boldsymbol{\lambda})}{\partial \boldsymbol{V}_k(t)}|_{\boldsymbol{V}_k(t)^*}=\mathbf{0},\quad \forall k,t.$$

In other words, the optimal points for subproblems are chosen to break the bigger gradient of the Lagrangian of problem (7) into smaller pieces. However, a stationary point can be either a local maximum or a saddle point of problem (7). A saddle point behaves as a local maximum if looking only along the direction of certain grouped coordinates. The potential of convergence to a saddle point is the "price" for decomposing a joint optimization into a sequence of simpler ones.

Since the strong duality holds for each subproblem, we utilize the Lagrangian dual method to solve the subproblems. The dual function of (8) is given by

$$g(\boldsymbol{\lambda})=\min_{\text{Tr}(\Gamma_m\boldsymbol{V}(t)\boldsymbol{V}(t)^\dagger)\le P_m} L(c,\{\boldsymbol{V}_k(t)\}_{k\in\mathcal{K}},\boldsymbol{\lambda}), \quad (9)$$

where the Lagrangian of (8) is

$$L(c,\{\boldsymbol{V}_k(t)\}_{k\in\mathcal{K}},\boldsymbol{\lambda})=\left(\sum_{k=1}^K b_k\lambda_k-1\right)c - \sum_{k=1}^K\lambda_k R_k(t)-\sum_{k=1}^K\lambda_k\sum_{s=1,s\neq t}^T R_k(s),$$

and $\boldsymbol{\lambda}=\{\lambda_1,\ldots,\lambda_K\}$ with $\lambda_k\ge 0,\forall k\in\mathcal{K}$ are the Lagrange multipliers. Since a linear function is bounded below only when it is identically zero, it is straightforward to prove that $g(\boldsymbol{\lambda})=-\infty$ except when $\sum_{k=1}^K b_k\lambda_k-1=0$.

The dual problem is then given by

$$\max_{\boldsymbol{\lambda}}\ \left\{\min_{\text{Tr}(\Gamma_m\boldsymbol{V}(t)\boldsymbol{V}(t)^\dagger)\le P_m} -\sum_{k=1}^K\lambda_k R_k(t)\right\}$$
$$s.t.\ \sum_{k=1}^K b_k\lambda_k=1,\quad \lambda_k\ge 0,\ \forall k\in\mathcal{K}. \quad (10)$$

The dual problem can be solved iteratively: the precoders ($\boldsymbol{V}_k(t)$'s) are updated by solving a minimization problem in each iteration and the Lagrange multipliers (λ_k's) can be updated via the subgradient-based method. To solve for the $\boldsymbol{V}_k(t)$'s with given Lagrange multipliers, the minimization problem in (10) can be rewritten as a WSRM problem under per-AP power constraint,

$$\max_{\text{Tr}(\Gamma_m\boldsymbol{V}(t)\boldsymbol{V}(t)^\dagger)\le P_m}\sum_{k=1}^K\lambda_k R_k(t). \quad (11)$$

Due to the non-convexity of the objective function and the interdependence of the precoders of simultaneous users in (11), it is difficult to find the solution to (11) based on the Karush-Kuhn-Tucker (KKT) conditions. Therefore, a more tractable approach is developed by solving an equivalent weighted sum MSE minimization problem with the same power constraint, which is formulated as,

$$\min_{\{\boldsymbol{V}_k(t)\}_k\in\mathcal{K}}\ \sum_{k=1}^K\text{Tr}\left(\boldsymbol{W}_k(t)\boldsymbol{E}_k(t)\right)$$
$$s.t.\ \text{Tr}(\Gamma_m\sum_{k=1}^K\boldsymbol{V}_k(t)\boldsymbol{V}_k(t)^\dagger)\le P_m,\quad \forall m. \quad (12)$$

Following a similar approach to that of [13], it can be shown that the gradient of the Lagrangian of (11) and (12) are equal if the MMSE receiver is given by

$$\boldsymbol{U}_k(t)=\left(\boldsymbol{H}_k\boldsymbol{V}_k(t)\boldsymbol{V}_k(t)^\dagger\boldsymbol{H}_k^\dagger+\boldsymbol{R}_{\bar{k}}(t)\right)^{-1}\boldsymbol{H}_k\boldsymbol{V}_k(t) \quad (13)$$

and the MSE weights satisfy

$$\boldsymbol{W}_k(t)=\lambda_k\left(\boldsymbol{I}+\boldsymbol{V}_k(t)^\dagger\boldsymbol{H}_k^\dagger\boldsymbol{R}_{\bar{k}}^{-1}(t)\boldsymbol{H}_k\boldsymbol{V}_k(t)\right). \quad (14)$$

This implies that Problems (10) and (12) share the same optimal solution $\{\boldsymbol{V}_k(t)\}_{k\in\mathcal{K}}$ under Conditions (13) and (14).

From this analysis, Problem (8) can be solved iteratively by solving a set of equivalent weighted sum mean square error (WSMSE) minimization problems, as summarized in Table II. The algorithm alternatively updates the precoders, MSE weights and MMSE combiners for each user, which involves a weighted sum MSE minimization problem in each iteration. As analyzed in [13], the algorithm converges to a local optimum. The key to finding the solution lies in solving the weighted sum MSE minimization problem (12) in each iteration.

TABLE II
ITERATIVE ALGORITHM FOR SOLVING WSRM

1.	Initialization: $n=0$, $\boldsymbol{V}_k(t)=\boldsymbol{V}_k(t)^0$ for $k\in\mathcal{K}$ such that $\text{Tr}(\Gamma_m\sum_{k=1}^K\boldsymbol{V}_k(t)\boldsymbol{V}_k(t)^\dagger)\le P_m,\forall m$;
2.	While $n<I_{max}$
3.	$n\leftarrow n+1$;
4.	Compute $\boldsymbol{U}_k(t)$ using (13) for $\forall k$ for given $\boldsymbol{V}_k$'s;
5.	Compute $\boldsymbol{W}_k(t)$ using (14) for all $\forall k$ for given $\boldsymbol{V}_k$'s and $\boldsymbol{U}_k$'s;
6.	Update $\boldsymbol{V}_k(t)$'s by solving problem (12);
7.	Quit if $\left\|\sum_{k=1}^K\lambda_k R_k^{(n)}(t)-\sum_{k=1}^K\lambda_k R_k^{(n-1)}(t)\right\|\le\varepsilon_1$.

The weighted sum MSE minimization problem can be solved by extending the algorithms in [13] and [43] to the formulated problem with per-AP power constraint. However, those algorithms cannot decouple the precoder of the k^{th} user and its combiner, which means power allocated to a link can only be gradually reduced, thereby requiring many iterations to fully deactivate a link.

In order to determine the active user set for each time slot and their corresponding precoders and stream allocation efficiently, we propose a different approach to address the weighted sum MSE minimization problem based on its duality. First, the Lagrangian dual function of the weighted sum

MSE minimization problem is given by

$$q(\boldsymbol{\mu}(t)) = \min_{\{\boldsymbol{V}_k(t)\}_{k\in\mathcal{K}}} \left\{ \sum_{k\in\mathcal{K}} Tr(\boldsymbol{W}_k(t)\boldsymbol{E}_k(t)) + \sum_{m=1}^{M} \mu_m(t) \left(\sum_{k=1}^{K} Tr(\boldsymbol{\Gamma}_m \boldsymbol{V}_k(t)\boldsymbol{V}_k(t)^\dagger) - P_m \right) \right\},$$

where $\mu_m(t) \geq 0$ for $m = 1, \ldots, M$ are the Lagrange multipliers. Optimizing $\{\boldsymbol{V}_k(t)\}_{k\in\mathcal{K}}$ with given $\boldsymbol{\mu}(t)$ yields

$$\boldsymbol{H}_k^\dagger \boldsymbol{U}_k(t)\boldsymbol{W}_k(t) = \sum_{m=1}^{M} \mu_m(t)\boldsymbol{\Gamma}_m(t)\boldsymbol{V}_k(t) + \sum_{l\in\mathcal{K}} \boldsymbol{H}_l^\dagger \boldsymbol{U}_l(t)\boldsymbol{W}_l(t)\boldsymbol{U}_l(t)^\dagger \boldsymbol{H}_l \boldsymbol{V}_k(t). \tag{15}$$

The above equation can only provide the precoder $\boldsymbol{V}_k$ as a function of its combiner $\boldsymbol{U}_k$, which will result in the same solution as that in [13] and [43]. To decouple the precoder and combiner of user k, we recall the equation for MMSE combiner and rewrite it into

$$\left(\boldsymbol{H}_k \boldsymbol{V}_k(t)\boldsymbol{V}_k(t)^\dagger \boldsymbol{H}_k^\dagger + \boldsymbol{R}_{\bar{k}}(t) \right) \boldsymbol{U}_k(t) = \boldsymbol{H}_k \boldsymbol{V}_k(t) \tag{16}$$

Then we can solve the precoder using (15) and (16). We first perform the following singular value decomposition (SVD),

$$\boldsymbol{R}_{\bar{k}}(t)^{-1/2} \boldsymbol{H}_k \boldsymbol{\Pi}(t)_{\bar{k}}^{-1/2} = \boldsymbol{F}_k \boldsymbol{D}_k \boldsymbol{G}_k^\dagger, \tag{17}$$

where we have

$$\boldsymbol{\Pi}_{\bar{k}}(t) = \sum_{l\in\mathcal{K}, l\neq k} \boldsymbol{H}_l^\dagger \boldsymbol{U}_l(t)\boldsymbol{W}_l(t)\boldsymbol{U}_l(t)^\dagger \boldsymbol{H}_l + \sum_{m=1}^{M} \mu_m(t)\boldsymbol{\Gamma}_m,$$

and $\boldsymbol{D}_k \in \mathbb{R}^{N_{r,k}\times N_{r,k}}$ is a diagonal matrix containing the singular values of the left-hand side of (17) ordered in decreasing order; $\boldsymbol{F}_k \in \mathbb{C}^{N_{rx,k}\times N_{rx,k}}$ and $\boldsymbol{G}_k \in \mathbb{C}^{N_t \times N_{rx,k}}$ are the corresponding left and right singular vectors of the left-hand side of (17). Based on [44] and [45], the precoder has the following structure with a given $\boldsymbol{\mu}$,

$$\boldsymbol{V}_k(t) = \boldsymbol{\Pi}_{\bar{k}}(t)^{-1/2} \boldsymbol{G}_k \boldsymbol{\Psi}_k, \tag{18}$$

where $\boldsymbol{\Psi}_k$ is an $N_{rx,k} \times N_{rx,k}$ diagonal matrix, given by

$$\boldsymbol{\Psi}_k = \left(\boldsymbol{W}_k(t)^{1/2} \boldsymbol{D}_k^{-1} - \boldsymbol{D}_k^{-2} \right)_+^{1/2}, \tag{19}$$

and $(\cdot)_+$ is the matrix $(\cdot)$ with the negative elements replaced with zeros. Here, the $(\cdot)_+$ operation in component $\boldsymbol{\Psi}_k$ can potentially turn off some streams by allocating zero power.

To find the optimal Lagrange multipliers $\boldsymbol{\mu}(t)$ that minimize the Lagrangian dual function, the additive update method can be used to optimize $\boldsymbol{\mu}(t)$ iteratively. It involves two steps in each iteration: i) solving the precoder $\boldsymbol{V}_k(t)$ for each user with fixed Lagrange multipliers using (17)–(19), and ii) updating the Lagrange multipliers using the subgradient-based method.

TABLE III
PRE-USER SELECTION PSEUDOCODE

1.	Let $\mathcal{U}_r = \{1, \ldots, K\}$ and $\mathcal{U}_s = \emptyset$
	$k^* = \text{argmax}_{k\in\mathcal{U}_r} w_k \log \left\| \boldsymbol{I} + P_k \bar{\boldsymbol{H}}_k \bar{\boldsymbol{H}}_k^H / N_r \right\|$
2.	Update $\mathcal{U}_s = \{k^*\}$ and $\mathcal{U}_r = \mathcal{U}_r - \{k^*\}$
3.	*While* $(\|U_s\| < K_0)$
4.	Calculate priority metric $f(\boldsymbol{H}_k, \boldsymbol{H}_{sel})$ for $\forall k \in \mathcal{U}_r$ using (20).
5.	Quit if $\max_{k\in\mathcal{U}_r} f(\boldsymbol{H}_k, \boldsymbol{H}_{sel}) < 0$
6.	$k^* = \text{argmax}_{k\in\mathcal{U}_r} f(\boldsymbol{H}_k, \boldsymbol{H}_{sel})$
7.	$\mathcal{U}_s = \mathcal{U}_s \cup k^*$ and $\mathcal{U}_r = \mathcal{U}_r - \{k^*\}$

VI. SCHEDULING VIA A TWO-STAGE APPROACH

In this section, we propose a lower-complexity heuristic to approximate the solution given by the algorithm of Section V. Since the channels in our target scenarios are assumed to be fixed during T time slots, we can optimize performance within each slot while achieving fairness across the entire T slots. The basic idea is to decompose the scheduling problem into two stages. First, the scheduler generates a set of diverse and high-performance communication sets by solving a set of WSRM problems, after collecting the CSI from all APs. Next, the scheduler computes a communication schedule that specifies the number of time slots allocated for each communication set in order to achieve a given fairness objective.

A. Communication Sets Generation

Here, we present an efficient iterative approach to generate diverse and high-performance communication sets. One communication set is generated per iteration through a 3-step procedure. First, pre-user selection is performed to find a "good" subset of users that can potentially maximize performance. Second, a WSRM problem is solved to calculate the MIMO weights of the selected users, which eliminates additional users and determines the stream allocation. Third, in preparation for the next iteration, the user weights are updated according to the previously generated communication sets and the target bandwidth fraction of each user. After a specified number of communication sets are generated in this iterative manner, a final group of communication sets is added to ensure that there is a solution that satisfies the fairness constraints.

1) Pre-User Selection: The objective of the user selection procedure is to select $K_0 < K$ users, that can potentially contribute to high-WSR performance. With a targeted K_0, it is costly to enumerate and evaluate all $\binom{K}{K_0}$ possible user groups. In this section, we propose an incremental selection algorithm to determine a high-performance user group.

In dense wireless networks, the inter-user interference is generally substantial. To improve the WSR performance, important factors should be taken into account for user selection procedure: (1) mutual orthogonality of selected users' channels, (2) the channel quality of selected users, (3) the user weights $w_k's$ and (4) the available power. Our proposed efficient user selection algorithm that incorporates all of these factors is shown in Table III.

The algorithm starts by selecting the user with highest interference-free weighted data rate. Let $\boldsymbol{Q}_k$ be the row basis of $\bar{\boldsymbol{H}}_k$, where $\bar{\boldsymbol{H}}_k = \boldsymbol{R}_{n_k}^{-1/2} \boldsymbol{H}_k$. The selected user set is denoted by $\mathcal{U}_s$ and the remaining user set is denoted by $\mathcal{U}_r$.

The number of users in $\mathcal{U}_s$ is given by $|\mathcal{U}_s|$. The priority metric is defined as follows:

$$f(\boldsymbol{H}_k, \boldsymbol{H}_{sel}) = w_k \log_2(1 + \frac{P_t/N_r}{|\mathcal{U}_s|+1}||\boldsymbol{H}_{e,k}||_F^2) + \sum_{i\in\mathcal{U}_s} w_i \log_2(1 + \frac{P_t/N_r}{|\mathcal{U}_s|+1}||\bar{\boldsymbol{H}}_i\boldsymbol{H}_{e,k}^{\perp}||_F^2) - \sum_{i\in\mathcal{U}_s} w_i \log_2(1 + \frac{P_t}{N_r|U_s|}||\bar{\boldsymbol{H}}_i||_F^2), \quad (20)$$

where $\boldsymbol{H}_{sel} = \left[\bar{\boldsymbol{H}}_i\right]_{i\in\mathcal{U}_s}$, $\boldsymbol{H}_{e,k} = \bar{\boldsymbol{H}}_k \times null(\boldsymbol{H}_{sel})$ and $\boldsymbol{H}_{e,k}^{\perp} = \boldsymbol{I} - \boldsymbol{Q}_k\boldsymbol{Q}_k^{\dagger}$. $||\cdot||_F$ denotes the Frobenius norm. $P_t = \sum_{m=1}^{M} P_m$ is the total transmit power of the cooperative APs. The first term in (20) evaluates the WSR contribution of user k when its precoder lies in the null space of the selected users' channel matrices. In the second term, the channels of previously selected users are projected to the null space of user k's equivalent channel. The selection priority metric implicitly reflects how much WSRM performance gain is contributed by user k. Then, the user with highest priority metric will be selected in each round. However, the maximum value of the priority metric could be less than 0, indicating that adding a new user may even hurt the overall performance. In this case, the user selection will terminate before K_0 users are selected.

Note that the parameter K_0 in the user selection algorithm can be tuned to achieve different tradeoffs between the aggregate performance and computational complexity. Smaller K_0 will eliminate more users at this stage and the achievable WSR will degrade as the price of lower computational complexity for MIMO weights computation. With larger K_0, fewer users will be excluded by the user selection procedure and the loss of WSR performance will be smaller.

2) Solving Weighted Sum Rate Maximization Problem: After the pre-user selection, the selected K_0 users in $\mathcal{U}_s$ serve as the input to a WSRM problem. The general form of a WSRM problem with per-AP power constraint is given as follows:

$$\max_{\{\boldsymbol{V}_k\}_{k\in\mathcal{U}_s}} \sum_{k=1}^{K_0} w_k R_k \quad s.t. \sum_{k=1}^{K_0} Tr(\boldsymbol{\Gamma}_m \boldsymbol{V}_k \boldsymbol{V}_k^{\dagger}) \le P_m, \quad \forall m, \quad (21)$$

where $w_k \ge 0$ is the weight assigned to the k^{th} user's rate.

The formulated WSRM problem in (21) for the selected K_0 users is similar to problem (11). Therefore, it can be solved via the proposed algorithm in Table II. It calculates the precoders and combiners of each selected users, which can determine the number of streams allocated each user by allocating zero power to deactivate some streams. In other words, the users with no active streams are further eliminated from the current communication set.

3) Adjusting the Link Weights: To ensure a good representation of a large number of users, multiple communication sets are generated by solving a set of WSRM problems with adjusted user weights. Let $\boldsymbol{r}_k$ be a $1 \times N$ vector that contains the data rates of the k^{th} user, i.e. $r_{k,n}$ denotes the data rate of the k^{th} user in the n^{th} communication set. When generating the $(i+1)^{\text{th}}$ communication set after the first i sets have already been generated, the basic idea is to assign larger weights to users that are more below their desired bandwidth proportions when considering the first i sets. A user k that is at or above its desired bandwidth proportion is assigned weight $w_k = 0$ and is therefore excluded from the current round of communication set calculation. This approach yields satisfying results in balancing high-performance communication sets and incorporating user diversity into the chosen high-performing sets. Mathematically, there are various ways to achieve the aforementioned weight adjustment idea. A general form is

$$w_k \ge w_j \quad \text{if } 0 \le \frac{u_k}{b_k} \le \frac{u_j}{b_j} \le 1$$
$$w_k = 0 \quad \text{if } \frac{u_k}{b_k} \ge 1 \quad (22)$$

where u_k is the bandwidth proportion of the kth user from previously computed N' communication sets, given by

$$u_k = \sum_{n=1}^{N'} r_{k,n} / \sum_{k=1}^{K}\sum_{n=1}^{N'} r_{k,n}.$$

In order to maximize the throughput over one scheduling period, we aim to maximize the sum rate performance of each time slot with different active user subsets. Therefore, in this paper, we update the user weights as follows:

$$w_k = \max(1 - u_k/b_k, 0). \quad (23)$$

Thus, the users that have achieved their target bandwidth fractions are excluded from the current round of calculation.

4) Single-User MIMO Communication Sets: After the first three steps are iterated a specified number of times, a final post-processing step is performed. In this step, we compute communication sets with a single active user per set. In this case, the active user achieves its interference-free data rate and is jointly served by the cooperative APs. The interference-free data rate of the single user is given by

$$\rho_k = \max_{\left\{\text{Tr}(\boldsymbol{\Gamma}_m \boldsymbol{V}_k \boldsymbol{V}_k^{\dagger}) \le P_m\right\}} \log_2 \left| \boldsymbol{I} + \boldsymbol{R}_{n_k}^{-1} \boldsymbol{H}_k \boldsymbol{V}_k \boldsymbol{V}_k^{\dagger} \boldsymbol{H}_k^{\dagger} \right|.$$

Let $\boldsymbol{\Sigma}_k = \boldsymbol{V}_k \boldsymbol{V}_k^{\dagger}$ be the transmit covariance matrix. Since ρ_k is convex over $\boldsymbol{\Sigma}_k$, its optimal solution can be obtained through standard techniques.

B. Multiuser MIMO Link Scheduling

After generating N_{tot} communication sets as discussed in Section VI-A, our focus is on developing a schedule to achieve both high aggregate performance and the target fairness. Let $\boldsymbol{r}_n = \{r_{1,n}, r_{2,n}, \ldots, r_{K,n}\}$ be the data rates of the n^{th} communication set, where $r_{k,n}$ is the data rate of the k^{th} user in the n^{th} communication set. If $r_{k,n} = 0$, it indicates that the k^{th} user is inactive in the n^{th} communication set. Recall the original formulation of the scheduling problem in (6). The per-AP power constaints are met during the communication sets generation stage. With the calculated data rates of each communication set, the problem reduces to the assignment of communication sets for T time slots that maximize the throughput while meeting the fairness constraints.

In this section, we reformulate the scheduling problem based on a set of N_{tot} candidate communication sets. Let $x_i \geq 0$ be the number of time slots to schedule the i^{th} communication set $\mathcal{G}_i$. The data rate of user j over one schedule period can be represented as

$$R_j = \sum_{i=1}^{N_{tot}} r_{j,i} x_i / T \tag{24}$$

We wish to find a vector $\boldsymbol{x} \in \mathbb{Z}^{N_{tot} \times 1}$ that satisfies the fairness constraints:

$$C_1 : R_1 : R_2 : \cdots : R_K = b_1 : b_2 : \cdots : b_K,$$

For a given schedule period with T time slots, we have the following constraint:

$$C_2 : \sum_{i=1}^{N_{tot}} x_i \leq T$$

The scheduling problem is then formulated to maximize the aggregate throughput under the fairness constraint,

$$\begin{aligned} \max_{\boldsymbol{x}} \ & \sum_{j=1}^{K} R_j \\ s.t. \ & x_i \geq 0, \quad x_i \in \mathbb{Z}, \ i = 1, \ldots, N_{tot} \\ & C_1 - C_2 \end{aligned} \tag{25}$$

Note that the fairness constraint C_1 contains $K-1$ equality constraints (i.e., $R_j/b_j = R_1/b_1$, for $j = 2, \ldots, K$). However, the perfect fairness imposed by C_1 lacks the flexibility to accommodate different scenarios. Therefore, we introduce the notion of ϵ-approximate fairness and relax the C_1 into a set of inequality constraints

$$C_3 : d(1-\epsilon)b_j \leq R_j \leq d(1+\epsilon)b_j, \quad j = 1, \ldots, K. \tag{26}$$

where $d = \sum_{j=1}^{K} R_j$. By replacing C_1 in (25) with C_3, a new scheduling problem with a variable fairness objective is formulated. Note that ϵ is the fairness factor, which controls the achieved fairness among users. For example, if $\epsilon = 0$, C_3 becomes the same as C_1, which leads to perfect fairness. When ϵ becomes sufficiently large, the scheduling problem corresponds to a throughput maximization problem with no fairness constraint. Since the formulated problem is a mixed integer linear programming (LP) problem, it can be solved by the commercial solvers, such as Gurobi. The basic strategy used in the solver is to use a linear-programming based branch-and-bound algorithm. Specifically, the relaxed LP problem is solved using standard approaches by removing all of the integrality restrictions, such as interior-point method. Then, a branch-and-bound algorithm is performed to search systematically for the optimal integer solution. Modern mixed integer LP solvers might also implement advanced mixed-integer programming preprocessing techniques before the branch-and-bound algorithm to limit the size of the branch-and-bound tree.

We did some tests with Gurobi and found that including a branch and bound search after solving the relaxed LP problem roughly doubled the overall computation time for the scheduling algorithm. To improve computation time, we execute the following procedure to approximate the optimal integer solution. First, the $res_i = x_i^* - \lfloor x_i^* \rfloor$ are sorted in descending order. Then, the solution x_i^*'s of the first I user sets with the higher res_i value are rounded to $\lceil x_i^* \rceil$, while the remaining x_i^*'s are rounded to $\lfloor x_i^* \rfloor$, where $I = T - \sum_{i=1}^{N_{tot}} \lfloor x_i^* \rfloor$. The rounded solution determines the number of time slots assigned to each communication set. The main disadvantage of rounding is that the fairness constraint C_3 might not be exactly satisfied in the final solution. However, it will be demonstrated in Section VII that the deviation from the targeted fairness is quite small and the schedule produced after rounding has *higher* aggregate throughput than the solution obtained by the branch-and-bound procedure.

VII. Simulation Results

In this section, we report on simulation experiments to evaluate the performance of our proposed scheduling algorithms from Section V and Section VI under time-based fairness (TF) criteria, which we denote by **Joint_TF** and **TwoStage_TF** in this section. The optimal solution to the LP relaxation problem is referred to as **TwoStage_RelaxedTF**, which serves as an upper bound of the **TwoStage_TF** solution with a given set of communication sets. For comparison, we also consider the following algorithms:

- **TwoStage_NUS_TF**: This algorithm is developed in our preliminary research [40] and is similar to the proposed **TwoStage_TF**. However, **TwoStage_NUS_TF** works without pre-user selection during communication set generation. **TwoStage_NUS_TF** also uses the notion of ϵ-approximate fairness.
- **IC_TF**: This algorithm solves the MIMO link scheduling problem with IC across multiple APs [6]. However, the data for a single user is transmitted solely by one AP. **IC_TF** is designed to achieve time fairness among users and, like **TwoStage_TF**, it uses a two-stage approach that first generates a set of communication sets and then chooses a schedule using the generated sets. In our simulations, the AP-user association for **IC_TF** is determined by the SNR at the user device, i.e., a user is served by the AP that provides the highest SNR.
- **TDMA**: This is a basic time-fair TDMA scheduling algorithm, where the links are scheduled sequentially in a round robin manner. Since in each time slot, there is only one user scheduled and served by all APs, it can achieve the interference-free data rates using the SVD MIMO weights. Moreover, TDMA allocates the bandwidth with perfect fairness in a time-based sense.
- **MaxRateMinFair**: This algorithm uses the generated communication sets of **TwoStage_TF** and optimizes the scheduler to maximize the throughput but with only minimal fairness. Minimal fairness is defined as having at least one time slot allocated to each user.

A. Simulation Setup

Settings for the simulation experiments are as follows. There are M APs and K users uniformly distributed in a circular region with a radius of 50 meters. We set each AP to have 4 antenna elements and each user to have 2 antenna elements. To compute the SNR and SINR values, we use a quasi-static Rayleigh flat-fading channel model with a path-loss

exponent of 3 and the noise power of -85 dBm. The transmit power of each AP is 23 dBm. The number of time slots within one scheduling period is denoted by T. Unless otherwise specified, we consider the downlink transmission with 3 cooperative APs, fairness factor $\epsilon = 0.05$ for **TwoStage_TF** and **TwoStage_NUS_TF**, the number of communication sets generated for **TwoStage_TF** and **TwoStage_NUS_TF** is $N = 1.5K$, and $T = 100$. All presented results are averaged over 500 random deployments. To evaluate fairness, we use the fairness index proposed in [4],

$$FI(\boldsymbol{u}, \boldsymbol{b}) = \exp\left(-\sum_{k=1}^{K} \left|\ln(u_k/b_k)\right| / K\right), \tag{27}$$

where u_k is the fraction of bandwidth allocated to the k^{th} user. The fairness index given by (27) takes values in $[0, 1]$, with 1 representing perfect fairness among users.

We focus primarily on aggregate throughput and fairness in our evaluations since those are the parameters included in our optimization problem formulation. Since latency is not included in our formulation, we do not include it in our evaluations. However, we would like to note that latency should not be equated with the length of the schedule, T, which can be quite large in our approach, e.g. 50 or 100 time slots. A longer schedule length simply improves our ability to meet a given fairness constraint. With longer schedule lengths, it is very likely that each user will be scheduled multiple times within each scheduling period so that the latency will be significantly lower than T time slots.

B. Fairness Constraints

Different choices of parameter $\boldsymbol{b}$ achieve different fairness objectives. In our evaluations, we use the notion of time-based fairness, which is particularly well-suited for multi-rate wireless networks. In [4], the idea of time-based fairness is extended to take interference into account. Specifically, each user is allocated an equal number of interference-free time slots, where its bandwidth then depends on the number of users and its own channel quality. Different from the standard notion of time-based fairness in wireless networks, this fairness notion eliminates interference-induced distortions on data rates introduced by the scheduling algorithm. The target bandwidth fraction is defined as $b_k = \rho_k / \sum_{k=1}^{K} \rho_k$, $\forall k$, where ρ_k is the interference-free data rate as discussed in Section VI-A.

C. Multi-Slot Scheduling Versus Slot-by-Slot Scheduling

In Figure 2, we compare the sum-rate performance of the slot-by-slot and multi-slot scheduling strategies, where max-min fairness is considered. For slot-by-slot scheduling, the utility function is chosen as maximizing the minimum user rate for each time slot. The max-min fairness can be achieved by setting $b_k = 1/K$ in our proposed multi-slot scheduling schemes. To guarantee the feasible solution for the slot-by-slot scheduling scheme, we consider the special case that the number of transmit antennas is the same as the number of receive antennas. As the number of users increases, the number of transmit antennas for each of 3 APs

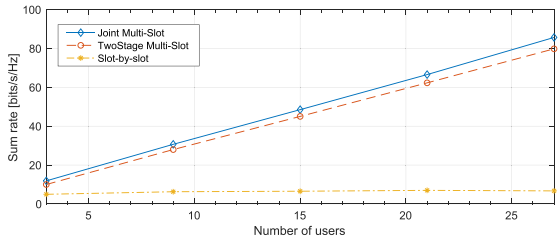

Fig. 2. Sum-rate vs. number of users.

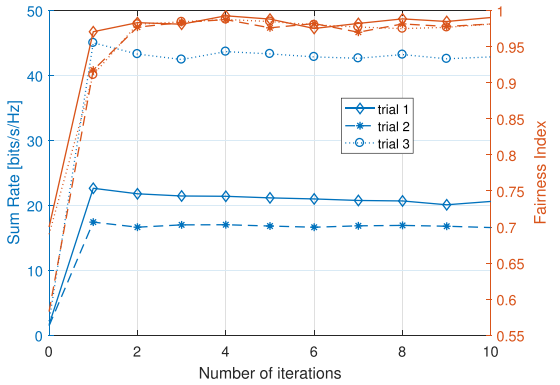

Fig. 3. Sum-rate and fairness vs. number of iterations for alternating optimization method.

are increasing accordingly. As shown in Figure 2, the sum-rate performance of the slot-by-slot is flat as the number of user increases, since it is always limited by the user with poorest channel quality. Moreover, simultaneous transmission of all users might even lower the sum-rate because of the strong inter-user interference. On the contrary, our proposed multi-slot scheduling scheme can exploit different multiuser combinations and combine them into a high-performance schedule over multiple time slots. With more users, it enables better opportunities to achieve higher sum-rate.

*D. Convergence Properties of **Joint_TF***

We first investigate the convergence properties of the alternating optimization method **Joint_TF**. As the algorithm iterates, it tries to improve the sum rate while approximating the desired fairness requirement.

To demonstrate the convergence of the algorithm, both sum rate and fairness are plotted as a function of the number of iterations with $T = 50$ in Figure 3. Three random trials of experiments are performed for $K = 10$. For all cases, the algorithm converges extremely quickly, reaching close to the final sum rate value after only 1 or 2 iterations. The small fluctuations within a narrow range after 2 iterations find the best operating point between sum-rate maximization and desired fairness.

E. Performance With Downlink Traffic

In this section, we focus on the downlink transmission and evaluate the sum-rate and fairness performance of the proposed algorithms.

1) Sum-Rate and Fairness Versus Number of Users: Figure 4 shows the achieved sum-rate and fairness of different algorithms as a function of the number of users. Overall,

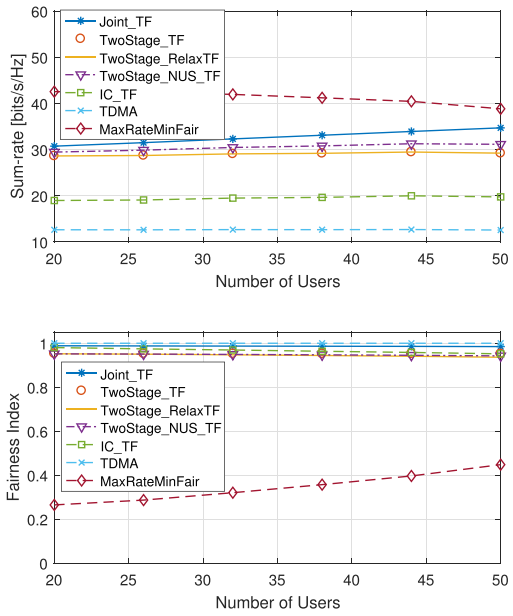

Fig. 4. Sum-rate and fairness vs. number of users.

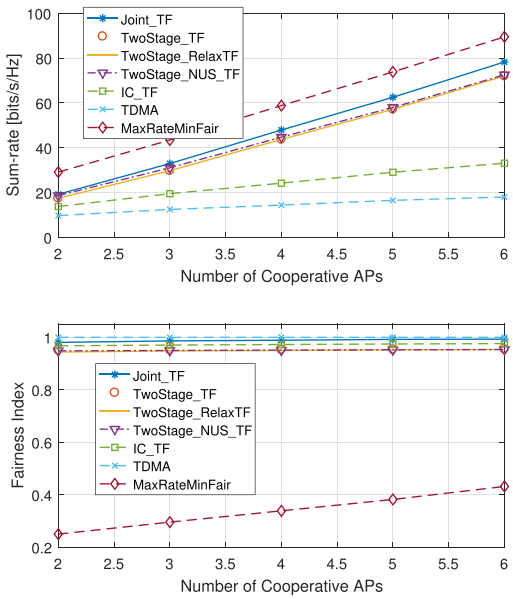

Fig. 5. Sum-rate and fairness vs. number of APs.

Joint_TF performs the best as it achieves very close to perfect fairness for all numbers of users and its performance improves steadily as the number of users increases. For 50 users, the sum rate of **Joint_TF** is within 10% of the greedy algorithm, which achieves a fairness value of only around 0.45 at that point. **TwoStage_TF** also achieves good fairness for all numbers of users. However, its sum rate performance gap compared to **Joint_TF** increases with the number of users, because the heuristic algorithm cannot fully explore the good user combinations when the number of users is large. Note, however, that the upper bound **TwoStage_RelaxedTF** is well approximated by **TwoStage_TF**, indicating that our proposed heuristic algorithm achieves a near-optimal solution for the chosen communication sets. Moreover, the sum-rate loss due to the pre-user selection can be estimated by comparing **TwoStage_TF** with **TwoStage_NUS_TF**. Although there is about a 5% sum-rate loss, we will see in Section VII-E that the pre-user selection in **TwoStage_TF** greatly improves the algorithm efficiency. Finally, we can see the advantage of full AP cooperation compared to only interference coordination in the significant sum-rate gap between **IC_TF** and the algorithms proposed herein.

2) Sum-Rate and Fairness Versus Number of APs: The sum-rate and fairness achieved by different algorithms is illustrated as a function of the number of cooperative APs in Figure 5, where the number of users is fixed to 30. Note that all algorithms make use of more APs to improve sum-rate, albeit to varying degrees. The algorithms that perform joint data transmission to all users have a sum rate that increases linearly with the number of APs. **IC_TF** performs interference coordination among APs but does not do joint data processing and its sum rate increases at a much lower rate. This shows very clearly the potential performance advantages associated with joint data transmission. For example, with 6 cooperative APs, the **Joint_TF** and **TwoStage_TF** achieve more than 2 times the sum-rate of **IC_TF**. Here, the joint optimization of user selection and scheduling done by **Joint_TF** consistently produces about 10% higher sum-rate than when separating those concerns, e.g. with **TwoStage_TF**. While one might think that TDMA performance would not increase with the number of APs since it schedules only one user per time slot, it does experience some rate increase due to increased total transmit power with more APs. The fairness values of the different algorithms are fairly similar to those seen as a function of the number of users.

3) Sum-Rate and Fairness Versus Number of Communication Sets: The performance of the proposed **TwoStage_TF** algorithm, as well as the other two-stage algorithms, depends on how many communication sets are generated during the first stage. As the number of communication sets increases, the scheduling algorithm can better exploit the potential performance of multiuser MIMO, albeit with increased running time to generate the sets. In Figure 6, the sum-rate is plotted as a function of the number of generated communication sets, which varies from $0.4K$ to $2K$, where $K = 30$. With a larger number of candidate communication sets, both **TwoStage_TF** and **TwoStage_NUS_TF** achieve sum rate performance close to that of the **Joint_TF**. For example, with $N = 2K$, **TwoStage_TF** achieves more than 92% of the sum rate of **Joint_TF** and **TwoStage_NUS_TF** achieves more than 95% of the joint algorithm's sum rate.

4) Sum-Rate and Fairness Versus Fairness Factor: We also present the results obtained with $K = 30$ for different choices of fairness factor ϵ. Figure 7 shows the sum rate and fairness index achieved by different algorithms, where the fairness factor ϵ is varied from 0.1 to 0.5. Since only the proposed **TwoStage_TF** and the similar algorithm **TwoStage_NUS_TF** allow different tradeoffs between the aggregate performance and fairness, the performance of other algorithms is not affected by the fairness factor. The difference

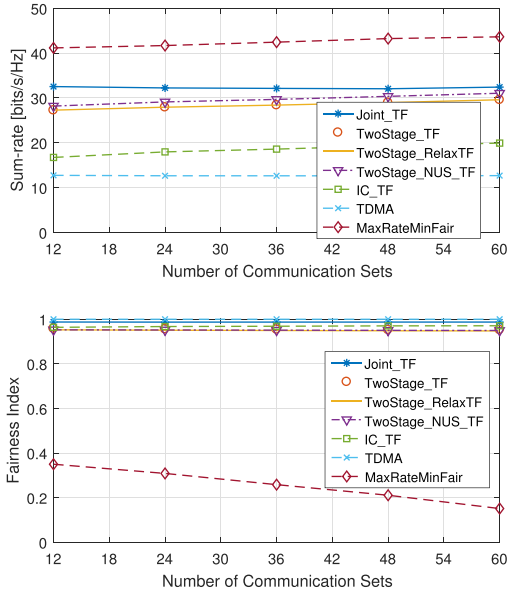

Fig. 6. Sum-rate and fairness vs. number of communication sets.

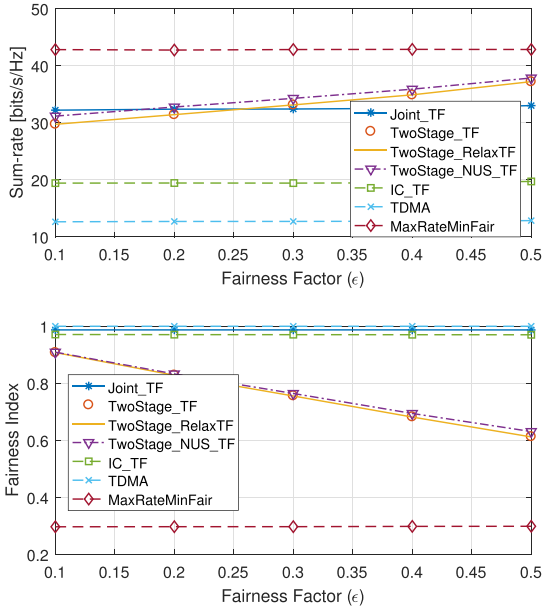

Fig. 7. Sum-rate and fairness vs. fairness factor.

between **TwoStage_TF** and **TwoStage_NUS_TF** caused by pre-user selection is quite small. Figure 7 also illustrates how the two-stage algorithms allow for a performance-fairness tradeoff. Based on Figure 7, this can be achieved by choosing the best operating point (ϵ) along the performance and fairness curves for either **TwoStage_NUS_TF** or **TwoStage_TF**. One use of this is to essentially solve the inverse optimization problem, namely to determine the best fairness possible for a given minimum performance threshold. This can be done by setting ϵ to the smallest value that achieves the required performance level, which can be found from the sum-rate curve of Figure 7. The achieved fairness index can then be determined from the fairness curve. Any other operating point in between the solutions to these two problems can also be determined from the plots.

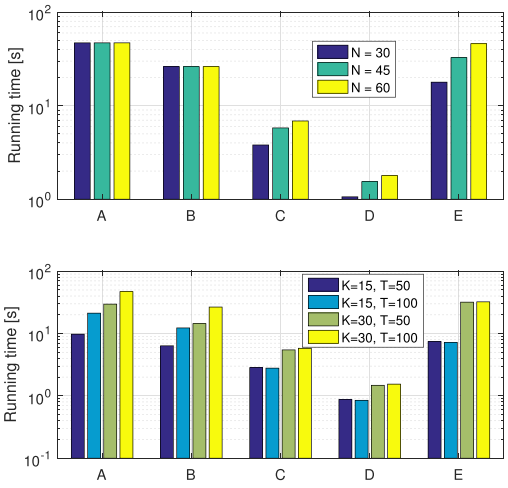

Fig. 8. Running time of different algorithms (A = **Joint_TF** with $T/T_w = 1$, B = **Joint_TF** with $T/T_w = 2$, C = **TwoStage** with $P = 1$, D = **TwoStage** with $P = 4$, E = **TwoStage_NUS_TF** with $P = 1$).

F. Running Time Evaluation

The computational complexity of the scheduler is an important issue since its computation time plus the use time of the schedule must fall within the stationary time of the network. In this subsection, we evaluate the execution times of the best performing scheduling algorithms evaluated in prior subsections. The algorithms were implemented in *Matlab* and run on an Intel i7-2700K 3.5 GHz CPU with 32 GB RAM.

Since the running time of the algorithms that use the two-stage approach is dominated by the first stage where multiple high-performing communication sets are generated, the **IC_TF** and **MaxRateMinFair** algorithms have nearly identical running time performance as our proposed **TwoStage_TF**. Thus, we use **TwoStage** to represent these three algorithms. Compared to other algorithms, **TDMA** exhibits much lower complexity since it simply calculates the SVD MIMO weights for each single-user transmission and does optimal power allocation. Its complexity grows linearly with the number of users, being about 0.2 s for 15 users and 0.4 s for 30 users, and is almost independent of other parameters. While **TDMA** is significantly faster than the other algorithms (see Figure 8), its throughput performance is not competitive with the others.

For **Joint_TF**, we also consider the impact of aggregating multiple time slots. Specifically, we evaluate the running time with unaggregated time slots and a version where each two consecutive time slots are combined into one slot. For **TwoStage** algorithms, we also consider the impact of parallel execution to speed up the communication set generation stage, which is the dominant factor in the running time.

Figure 8 shows the running times of the various algorithms for a few choices of parameters, such as number of communication sets (N), number of users (K) and schedule length (T), with a log scale on the y-axis. First, we evaluate the running time with different numbers of communication sets, i.e., $N = 30, 45, 60$, for $K = 30$. Although more communication sets provides higher sum-rate performance for two-stage approaches, including **TwoStage** and **TwoStage_NUS_TF**, as indicated in Figure 6, generating $N = 60$ communication

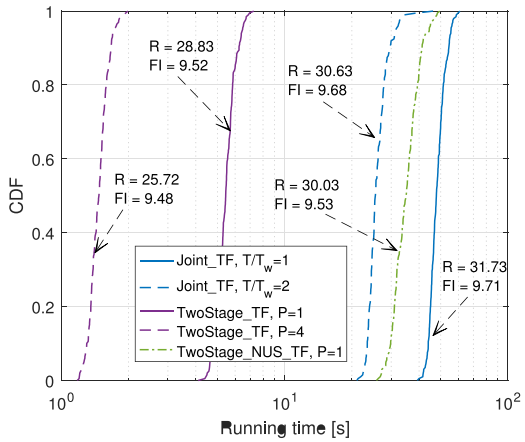

Fig. 9. CDF of the running time for $K = 30$, $T = 100$.

sets approximately doubles the running time compared to $N = 30$. From the figure, we see that the running time is reduced substantially when pre-user selection is employed. The **TwoStage**, which employs pre-user selection, has running times from about 2.5 to 6 seconds, which is more than 5 times faster than **TwoStage_NUS_TF**. For environments that need even shorter execution times, **TwoStage** can be sped up further with parallel calculation of communication sets. Using four processors, **TwoStage** only needs 1 to 2 seconds and is about 4 times faster than **TwoStage**. While the complexity of the joint algorithm is independent of N, it is significantly affected by K and T. From Figure 8 with different choices of K and T, we see that the running time for the unmodified **Joint_TF** ranges from 10 seconds to almost 50 seconds, which is clearly at the high end of what might be practical even in low-mobility environments. As is expected, aggregating pairs of time slots into a single slot (B bars in Figure 8) cuts these times in half, which brings the execution time down to more acceptable levels, particularly if the number of users is not too large. Interestingly, the two-stage approach without pre-user selection (**TwoStage_NUS_TF**) has an execution time that is on the same order of magnitude as **Joint_TF** (slightly less than the unmodified **Joint_TF** for most cases but slightly higher than the aggregated **Joint_TF**). We also note that our proposed two-stage method can work with other heuristic algorithms to generate candidate communication sets, which might be able to further lower the computational cost. Finding heuristic communication set generation techniques that have lower complexity without sacrificing too much performance is left as a topic for future research.

It is also important to understand the trade-offs between sum rate performance and running time for the different algorithms under consideration. Figure 9 shows the performance impact of time-slot aggregation for **Joint_TF** and parallel execution for **TwoStage_TF**. Sum rate for the time-aggregated **Joint_TF** is only about 4% lower than without aggregation, while execution time is halved. The impact of parallelization for **TwoStage_TF** is higher: sum rate is reduced by about 10% while running time is almost 4 times shorter, as compared to the sequential version.

The sum-rate loss caused by parallel processing in **TwoStage_TF** is caused by the randomized initialization of user weights for each parallel process. This loss can be partially compensated for by generating more communication sets over each parallel process to achieve a good tradeoff between the running time and sum-rate performance, as illustrated in Table IV. For example, assigning the workload of generating 50 communication sets to 2 parallel processes will introduce 4% sum-rate loss with about 1/2 the running time of the centralized processing. By increasing the workload of each process from 25 to 30, the sum-rate achieved by parallel processing with $P = 2$ reaches 99.7% of the sequential version while consuming less than 60% of the running time. Similar results can be observed for the case of $P = 4$.

TABLE IV

SUM-RATE AND RUNNING TIME PERFORMANCE FOR PARALLEL PROCESSING WITH DIFFERENT N_{tot} AND P

P	N_{tot}/P	sum-rate [bits/s/Hz]	running time [s]
1	50	28.6330	6.1537
2	25	27.5748	2.9723
2	30	28.5412	3.5088
4	12	25.8629	1.5175
4	15	27.2899	1.8399
4	20	28.6054	3.5437

VIII. CONCLUSION

In this paper, we studied the MIMO link scheduling problem for a cluster of cooperative APs and a number of stationary users. We proposed two scheduling algorithms, one that jointly optimizes the MIMO weights and user selection for users over the multiple time slots of a complete schedule, and a second that operates in two phases: first, high-performance communication sets for single time slots are generated via an iterative weighted sum-rate maximization procedure, and next an integer programming problem is solved to produce a schedule that provides near-optimal performance for the chosen communication sets and given fairness constraint Results showed that the joint optimization method achieves near-perfect fairness with aggregate throughput close to a greedy algorithm that has only minimal fairness. The two-stage algorithm sacrifices performance for running time; its aggregate throughput is 10–15% lower than the alternating optimization algorithm but running time evaluations showed that it is 5-10 times faster.

REFERENCES

[1] D. Gesbert *et al.*, "Multi-cell MIMO cooperative networks: A new look at interference," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 9, pp. 1380–1408, Dec. 2010.

[2] R. Irmer *et al.*, "Coordinated multipoint: Concepts, performance, and field trial results," *IEEE Commun. Mag.*, vol. 49, no. 2, pp. 102–111, Feb. 2011.

[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] D. Blough, G. Resta, and P. Santi, "Interference-aware proportional fairness for multi-rate wireless networks," in *Proc. IEEE INFOCOM*, Apr./May 2014, pp. 2733–2741.

[5] G. Tan and J. Guttag, "Time-based fairness improves performance in multi-rate WLANs," in *Proc. USENIX Conf.*, 2004, p. 23.

[6] L. M. Cortés-Peña and D. M. Blough, "MIMO link scheduling for interference suppression in dense wireless networks," in *Proc. IEEE WCNC*, Mar. 2015, pp. 1225–1230.
[7] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Performance anomaly of 802.11 b," in *Proc. IEEE INFOCOM*, vol. 2, Mar./Apr. 2003, pp. 836–843.
[8] J. B. Andersen, J. O. Nielsen, G. F. Pedersen, G. Bauch, and G. Dietl, "Doppler spectrum from moving scatterers in a random environment," *IEEE Trans. Wireless Commun.*, vol. 8, no. 6, pp. 3270–3277, Jun. 2009.
[9] C. Shepard, J. Ding, R. Guerra, and L. Zhong, "Understanding real many-antenna MU-MIMO channels," in *Proc. Asilomar Conf. Signals, Syst. Comput.*, 2016, pp. 461–467.
[10] M. Ge and D. M. Blough, "Mobility-aware multi-user MIMO link scheduling for dense wireless networks," in *Proc. IEEE Int. Conf. Commun.*, May 2018, pp. 1–7.
[11] S. W. Peters and R. W. Heath, Jr., "Cooperative algorithms for MIMO interference channels," *IEEE Trans. Veh. Technol.*, vol. 60, no. 1, pp. 206–218, Jan. 2011.
[12] F. Negro, S. Shenoy, I. Ghauri, and D. Slock, "On the MIMO interference channel," in *Proc. Inf. Theory Appl. Workshop*, 2010, pp. 1–9.
[13] S. S. Christensen, R. Agarwal, E. De Carvalho, and J. M. Cioffi, "Weighted sum-rate maximization using weighted MMSE for MIMO-BC beamforming design," *IEEE Trans. Wireless Commun.*, vol. 7, no. 12, pp. 4792–4799, Dec. 2008.
[14] G. Dimić and N. D. Sidiropoulos, "On downlink beamforming with greedy user selection: Performance analysis and a simple new algorithm," *IEEE Trans. Signal Process.*, vol. 53, no. 10, pp. 3857–3868, Oct. 2005.
[15] Z. Shen, R. Chen, J. G. Andrews, R. W. Heath, Jr., 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, Sep. 2006.
[16] J. Nam, A. Adhikary, J.-Y. Ahn, and G. Caire, "Joint spatial division and multiplexing: Opportunistic beamforming, user grouping and simplified downlink scheduling," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 876–890, Oct. 2014.
[17] V. K. N. Lau, "Asymptotic analysis of SDMA systems with near-orthogonal user scheduling (NEOUS) under imperfect CSIT," *IEEE Trans. Commun.*, vol. 57, no. 3, pp. 747–753, Mar. 2009.
[18] T. Yoo and A. Goldsmith, "On the optimality of multiantenna broadcast scheduling using zero-forcing beamforming," *IEEE J. Sel. Areas Commun*, vol. 24, no. 3, pp. 528–541, Mar. 2006.
[19] J. Mao, J. Gao, Y. Liu, and G. Xie, "Simplified semi-orthogonal user selection for MU-MIMO systems with ZFBF," *IEEE Wireless Commun. Lett.*, vol. 1, no. 1, pp. 42–45, Feb. 2012.
[20] L.-N. Tran, M. Bengtsson, and B. Ottersten, "Iterative precoder design and user scheduling for block-diagonalized systems," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3726–3739, Jul. 2012.
[21] E. Castañeda, A. Silva, A. Gameiro, and M. Kountouris, "An overview on resource allocation techniques for multi-user MIMO systems," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 239–284, 1st Quart., 2017.
[22] L. Georgiadis, M. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," *Found. Trends Netw.*, vol. 1, no. 1, pp. 1–144, 2006.
[23] M. J. Neely, E. Modiano, and C.-P. Li, "Fairness and optimal stochastic control for heterogeneous networks," in *Proc. INFOCOM*, Mar. 2005, pp. 1723–1734.
[24] H. Shirani-Mehr, G. Caire, and M. J. Neely, "MIMO downlink scheduling with non-perfect channel state knowledge," *IEEE Trans. Commun.*, vol. 58, no. 7, pp. 2055–2066, Jul. 2010.
[25] R. Elliott and W. Krzymien, "Downlink scheduling via genetic algorithms for multiuser single-carrier and multicarrier MIMO systems with dirty paper coding," *IEEE Trans. Veh. Technol.*, vol. 58, no. 7, pp. 3247–3262, Sep. 2009.
[26] H. Bang, T. Ekman, and D. Gesbert, "Channel-predictive proportional fair scheduling," *IEEE Trans. Wireless Commun.*, vol. 7, no. 2, pp. 482–487, Feb. 2008.
[27] V. Lau, "Proportional fair space-time scheduling for wireless communications," *IEEE Trans. Commun.*, vol. 53, no. 8, pp. 1353–1360, Aug. 2005.
[28] H. Huh, G. Caire, S.-H. Moon, and I. Lee, "Multi-cell MIMO downlink with fairness criteria: The large system limit," in *Proc. Int. Symp. Inf. Theory*, 2010, pp. 2058–2062.
[29] M. Kountouris, R. de Francisco, D. Gesbert, D. Slock, and T. Salzer, "Scheduling for multiuser MIMO downlink channels with ranking-based feedback," *EURASIP J. Adv. Signal Process.*, vol. 2008, Dec. 2008, Art. no. 854120.
[30] H. Huh, S.-H. Moon, Y.-T. Kim, I. Lee, and G. Caire, "Multi-cell MIMO downlink with cell cooperation and fair scheduling: A large-system limit analysis," *IEEE Trans. Inf. Theory*, vol. 57, no. 12, pp. 7771–7786, Dec. 2011.
[31] S. Park, J. Choi, and D. Love, "Multicell cooperative scheduling for two-tier cellular networks," *IEEE Trans. Inf. Theory*, vol. 62, no. 2, pp. 536–551, Feb. 2014.
[32] J. Zhang, R. Chen, J. G. Andrews, A. Ghosh, and R. W. Heath, Jr., "Networked MIMO with clustered linear precoding," *IEEE Trans. Wireless Commun.*, vol. 8, no. 4, pp. 1910–1921, Apr. 2009.
[33] W. Yu, T. Kwon, and C. Shin, "Multicell coordination via joint scheduling, beamforming and power spectrum adaptation," *IEEE Trans. Wireless Commun.*, vol. 12, no. 7, pp. 1–14, Jul. 2013.
[34] E. Björnson, G. Zheng, M. Bengtsson, and B. Ottersten, "Robust monotonic optimization framework for multicell MISO systems," *IEEE Trans. Signal Process.*, vol. 60, no. 5, pp. 2508–2523, May 2012.
[35] G. Dartmann, X. Gong, and G. Ascheid, "Application of graph theory to the multicell beam scheduling problem," *IEEE Trans. Veh. Technol.*, vol. 62, no. 4, pp. 1435–1449, May 2013.
[36] X. Xie and X. Zhang, "Scalable user selection for MU-MIMO networks," in *Proc. IEEE INFOCOM*, Apr. 2014, pp. 808–816.
[37] N. Anand, J. Lee, S.-J. Lee, and E. W. Knightly, "Mode and user selection for multi-user MIMO WLANs without CSI," in *Proc. IEEE INFOCOM*, Apr./May 2015, pp. 451–459.
[38] L. Li, M. Pal, and Y. R. Yang, "Proportional fairness in multi-rate wireless LANs," in *Proc. IEEE INFOCOM*, Apr. 2008, pp. 1004–1012.
[39] W. Li *et al.*, "AP association for proportional fairness in multirate WLANs," *IEEE/ACM Trans. Netw.*, vol. 22, no. 1, pp. 191–202, Feb. 2014.
[40] M. Ge and D. M. Blough, "High-throughput and fair scheduling for access point cooperation in dense wireless networks," in *Proc. IEEE WCNC*, Mar. 2017, pp. 1–6.
[41] R. S. Blum, "MIMO capacity with interference," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 5, pp. 793–801, Jun. 2003.
[42] P. D. Tao and L. T. H. An, "Convex analysis approach to D. C. programming: Theory, algorithms and applications," *Acta Math. Vietnamica*, vol. 22, no. 1, pp. 289–355, 1997.
[43] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, "An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel," *IEEE Trans. Signal Process.*, vol. 59, no. 9, pp. 4331–4340, Sep. 2011.
[44] H. Sampath, P. Stoica, and A. Paulraj, "Generalized linear precoder and decoder design for MIMO channels using the weighted MMSE criterion," *IEEE Trans. Commun.*, vol. 49, no. 12, pp. 2198–2206, Dec. 2001.
[45] L. M. Cortés-Peña, J. R. Barry, and D. M. Blough, "Jointly optimizing stream allocation, beamforming and combining weights for the MIMO interference channel," *IEEE Trans. Wireless Commun.*, vol. 14, no. 4, pp. 2245–2256, Apr. 2015.

Mengyao Ge received the B.S. and M.S. degrees in electrical engineering from Nanjing University, China, in 2011 and 2014, respectively, and the Ph.D. degree in electrical engineering from the Georgia Institute of Technology in 2018. She is currently a Staff Engineer with MediaTek, Irvine, CA, USA. Her research interests are in the areas of wireless communications and networks.

Douglas M. Blough received the B.S. degree in electrical engineering and the M.S. and Ph.D. degrees in computer science from Johns Hopkins University in 1984, 1986, and 1988, respectively. He is currently a Professor of electrical and computer engineering with the Georgia Institute of Technology, where he also holds a joint appointment in the School of Computer Science. His research interests include wireless networks and dependable distributed systems. He has served on the Technical Program Committee for numerous conferences, including DSN, MobiHoc, MobiCom, MASS, Infocom, and ICDCS, among others. He is an Area Chair for Infocom 2019 and has an been Associate Editor for four different IEEE TRANSACTIONS, including the IEEE TRANSACTIONS ON MOBILE COMPUTING from 2008 to 2013.