OPTIMIZING MULTIUSER MIMO FOR ACCESS POINT COOPERATION IN DENSE WIRELESS NETWORKS

A Dissertation
Presented to
The Academic Faculty

By

Mengyao Ge

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology

May 2018

OPTIMIZING MULTIUSER MIMO FOR ACCESS POINT COOPERATION IN DENSE WIRELESS NETWORKS

Approved by:

Dr. Douglas M. Blough, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Mary Ann Weitnauer
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. John R. Barry
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Raghupathy Sivakumar
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Ellen W. Zegura
School of Computer Science
Georgia Institute of Technology

Date Approved: February 20, 2018

To my family, for your endless love and support.

ACKNOWLEDGEMENTS

Over the past four years of my doctoral study, I have received support and help from a great number of individuals in different forms. First and foremost, I would like to express my deepest gratitude to my advisor, Dr. Douglas M. Blough, for his continuous encouragement and support throughout my entire doctoral study. His excellent guidance and intellectual insights help me steadily advance towards the successful completion of the Ph.D. program, as well as this thesis. It has been a honor and privilege to be his student and work with him.

I would also like to thank all the academic members of the School of Electrical and Computer Engineering at the Georgia Institute of Technology for their help throughout the Ph.D. program. Special thanks go to Dr. John Barry and Dr. Mary Ann Weitnauer, who provided me valuable and constructive inputs on my work, and also kindly serve in my Ph.D. Defense Reading Committee. I also thank Dr. Raghupathy Sivakumar and Dr. Ellen Zegura who kindly serve in my Ph.D. Defense Committee. All of their valuable comments and advices have greatly improved my research and the quality of this dissertation.

I would also like to thank my friends and colleagues at Georgia Tech, Jenny Zhang, Qiang Hu, Lei Zhang, Huiye Liu and Shuai Nie, for their everlasting friendship and constant support, and for every unforgettable moments we have together at Atlanta. To Dr. Qiongjie Lin, Yubin Jian, , and many others at Georgia Tech ECE, thank you for the many enlightening discussions and insights.

Finally, I would like to express my gratitude to my family, particularly to my parents and husband, for their endless love and support. I could never successfully complete my Ph.D. program without your trust, sacrifice and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

As the usage of wireless devices continues to grow rapidly in popularity, wireless networks that were once designed to support a few laptops must now host a much wider range of equipments, including smart phones, tablets, and wearable devices, that often run bandwidth-hungry applications. Improvements in wireless local access network (WLAN) technology are expected to help accommodate the huge traffic demands. In particular, advanced multicell Multiple-Input Multiple-Output (MIMO) techniques, involving the cooperation of APs and multiuser MIMO processing techniques, can be used to satisfy the increasing demands from users in high-density environments.

The objective of this thesis is to address the fundamental problems for multiuser MIMO with AP cooperation in dense wireless network settings. First, for a very common multiuser MIMO linear precoding technique, block diagonalization, a novel pairing-and-binary-tree based user selection algorithm is proposed. The algorithm achieves high sum-rate performance with low complexity and good scalability, and has the ability to balance the aggregate performance and computational cost. Second, without the zero-forcing constraint on the multiuser MIMO transmission, a general weighted sum rate maximization problem is formulated for coordinated APs. A scalable algorithm that performs a combined optimization procedure is proposed to determine the user selection and MIMO weights. Third, we study the fair and high-throughput scheduling problem with multiuser MIMO transmission by formally specifying an optimization problem that captures all aspects of the problem settings, including the MIMO weights, practical power constraint, fairness and user selection. Two algorithms are proposed to solve the problem using either alternating optimization or a two-stage procedure. Fourth, with the coexistence of both stationary and mobile users, different scheduling strategies are suggested for different user types. An overall scheduling framework using mobility hints is proposed to reduce the protocol overhead and sustain good performance for all users. The approach exhibits noteworthy performance gain, espe-

cially for scenarios with limited mobility. Moreover, we also discuss how to integrate the proposed solutions for cooperative APs with 802.11 protocols. The provided theoretical analysis and simulation results in this thesis lay out the foundation for the realization of the clustered WLAN networks with AP cooperation.

CHAPTER 1
INTRODUCTION

Over the last decade, wireless data traffic has experienced a dramatic growth driven by the ever-increasing number of wireless devices and applications. Although advanced wireless local access network (WLAN) techniques and the dense deployment of access points (APs) are expected to increase the network capacity to accommodate huge traffic demands, the chaotic unplanned IEEE 802.11 WLAN deployment with many nearby APs sharing the limited spectrum leads to a high-level of co-channel interference, especially in dense deployments, which significantly hinders the overall performance improvement. To break the performance bottleneck within the unlicensed band, coordination of communications across multiple APs together with advanced MIMO processing techniques is anticipated to provide considerable performance improvements [1, 2, 3, 4, 5, 6].

1.1 Motivation and Research Objectives

The wireless access performance issues in unplanned WLANs span numerous deployment scenarios, as shown in Figure 1.1. For instance, large-scale enterprise wireless networks are becoming overwhelmed by high traffic demands in dense areas, which covers most office-type environments. However, simply deploying more APs might even reduce the per client performance, due to the exacerbated interference and protocol inefficiencies. Similar dilemma can occur in auditoriums-style rooms, such as lecture halls and large conference rooms, where there are generally many users sharing the limited bandwidth during peak hours. Even home users are beginning to experience deficient wireless performance, as they sign up for ultra-high-speed services for the wireless devices. With the development of the smart-home devices, sharing an AP among 5-10 wireless devices is present in many homes today. With the rapidly growing demands for data services over wireless networks,

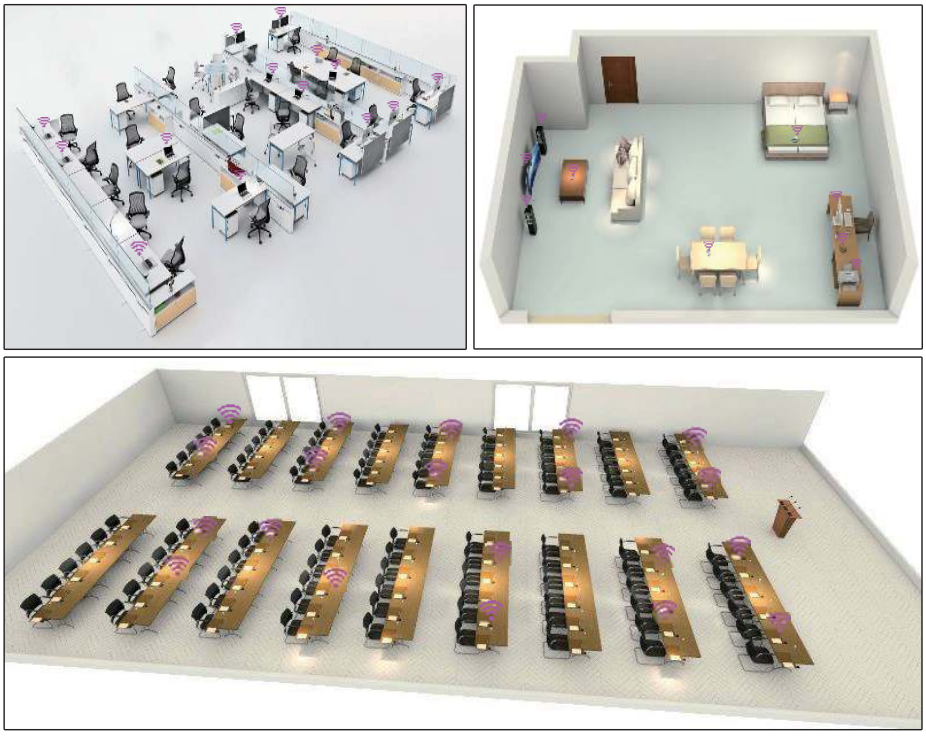

Figure 1.1: Dense wireless network deployment scenarios.

service providers are increasingly faced with the challenge of how to improve the spectral efficiency.

The multiple-input-multiple-output (MIMO) technique has drawn great attention as a method to boost the spectral efficiency without the need of additional bandwidth [7, 8]. MIMO systems exploit spatial degrees of freedom to support simultaneous multiple data streams, by equipping the both ends of a link with multiple antennas. Recent research on MIMO communication has shift the paradigm from point-to-point MIMO to multiuser contexts [9, 10, 11]. However, the spatial multiplexing gain bringing in by multiuser MIMO does not scale well to the multicell WLANs with distributed transmitters due to limited number of orthogonal channels and the inevitable inter-cell interference. A potential approach to break the wireless performance bottleneck in multicell WLANs is the development of advanced multicell MIMO techniques involving the cooperation of APs and coordination of communications across multiple devices. In our target scenarios, such as

most of the enterprise networks, the adjacent APs generally share one network gateway with one Internet connection. Therefore, inside an enterprise network, multiple APs can be clustered and cooperate to control the lower-layer parameters and to optimize the overall performance and fairness. The solutions can also be applied to other dense wireless deployments such as the wireless network in large office/apartment buildings and other commercial spaces. The clustered structure enables concurrent transmission from multiple distributed APs, and thus eliminates inter-cell interference. Combining this approach with the orthogonal channel deployment, the wireless performance are expected to exhibit good scalability for small-to-meduim size multicell WLANs.

On the way to pave for the realization of the clustered WLANs, there still exist a number of critical research challenges to be addressed even in the most ideal small-to-medium-scale environments. One of the main challenges is the physical-layer solutions to the targeted multi-AP co-channel environment. With the cooperation across multiple distributed APs, more users can be supported simultaneously by employing multiuser MIMO communication techniques. The key to improve the performance is to achieve a combination of spatial multiplexing and interference suppression. However, the practical constraints in AP-coordinated environments, such as channel estimation overhead, various channel variability, channel state information (CSI) accuracy and per-node power constraint, poses difficulties on solving the problem. With many users sharing the resources within a cluster, another key problem to solve is how to effectively use those resources to obtain high aggregate performance while also achieving fairness among the many competing users. Different from scheduling with single user transmission, scheduling with multiuser MIMO links depends on both the user combinations and their MIMO weights. Developing a schedule that integrates with the proposed physical-layer solution is a challenging task. Moreover, the fairness objectives and the aforementioned practical constraints will further complicate the design of the scheduling scheme.

Our focus is on optimizing the performance of dense wireless networks with a cluster

of cooperative APs when employing multiuser MIMO communication. The objective of this thesis is to address the challenges therein and establish the foundations of the clustered WLANs with AP cooperation.

1.2 Contributions

The primary contributions of this thesis are:

- The first contribution (Chapter 3) is that we propose an efficient and scalable user selection algorithm for the frequently-used block diagonalization precoding technique. 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.

- The second contribution (Chapter 4) is that we propose a combined optimization procedure that performs both user selection and MIMO weight calculation and scales well as the number of users increases. User selection eliminates some undesirable users, while MIMO weight calculation determines the precoders and combiners for all active nodes. A new performance metric, which takes into account available power, channel quality and orthogonality, and user weights, is used to perform an initial phase of user selection. A WSR maximization algorithm is then executed to optimize MIMO weights of selected users and further refine the user selection.

- The third contribution (Chapter 5) is that we address the fair scheduling problem with MIMO links to maximize the aggregate throughput subject to a fairness constraint that is general enough to capture many different fairness objectives. We formally specify a nonconvex optimization problem that captures all aspects of the problem setting and we propose two algorithms to approximate its solution. The first algo-

rithm jointly optimizes selection of user sets, MIMO precoders, and assignment of user sets to time slots, so that it guarantees perfect fairness and produces at least a local optimum for aggregate throughput. The second algorithm separately optimizes firstly user sets and MIMO precoders and secondly assignment of user sets to time slots. It has lower computational complexity and allows throughput and fairness to be traded off easily for situations where maximizing throughput is critical and approximate fairness is acceptable.

- The fourth contribution (Chapter 6) is that we propose a mobility-aware scheduling approach to accommodate both static and mobile users. It aims to alleviate the protocol overhead and provide satisfactory performance of both stationary and mobile users. The algorithm differentiates static and mobile users, separates them into different time slots and adaptively determines their transmission time fractions to balance fairness. Moreover, different scheduling strategies are applied for the two groups according to the user profile.

1.3 Organization of the Thesis

The rest of the thesis is organized as follows. In Chapter 2, we provide a brief introduction of basic MIMO concepts and techniques, and discuss the idea of clustered WLANs for dense wireless networks. Then, in Chapter 3, a novel pairing-and-binary-tree-based user selection algorithm to address the user selection issue for multi-user MIMO in dense environments is proposed for block diagonalization precoding technique. Moreover, in Chapter 4 the problem of weighted sum rate maximization with cooperative APs is addressed via a combined optimization procedure that scales well as the number of users increases. In addition, two scheduling algorithms are proposed to achieve both high throughput and target fairness in Chapter 5. In Chapter 6, a mobility-aware scheduling algorithm is developed to alleviate the protocol overhead and sustain high performance of both stationary and mobile users. Finally, in Chapter 7, our conclusions and suggestions for future work are provided.

CHAPTER 2

BACKGROUND AND NETWORK MODEL

Multiple-input multiple-output (MIMO) communication technologies provide promising improvements of wireless link performance, and has been integrated into the core of several wireless standards, such as LTE-A systems [12, 13] and wireless LANs [14, 15]. With multiple transmit and receive antennas, a MIMO system takes advantage of the spatial diversity from spatially separated antennas in a rich multipath scattering environment, promises substantial increase in channel capacity [16, 17].

In this chapter, we will first review the basic concepts of MIMO systems, ranging from point-to-point MIMO to multiuser MIMO. Then, we will focus on the dense wireless network and discuss how it benefits from the advanced MIMO techniques. The network model and optimization problems therein will be discussed.

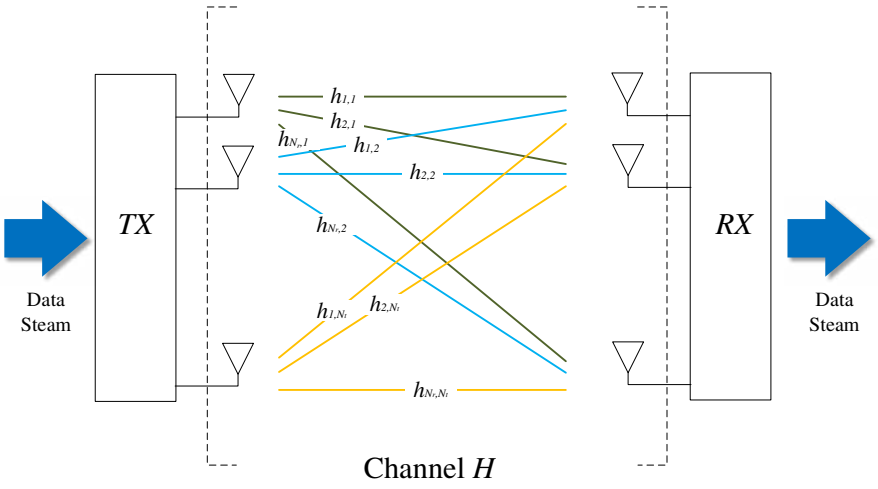

Figure 2.1: System block diagram for a point-to-point MIMO link.

2.1 Point-to-point MIMO

MIMO techniques were first investigated in point-to-point transmission, which transmit multiple streams between a transmitting-receiving nodes pair. In such a traditional single-user view of MIMO systems

2.1.1 Fundamentals of MIMO systems

Figure 2.1 illustrates a basic point-to-point MIMO system. In general, the impulse response of a time-variant MIMO channel is represented by [18]

$$\boldsymbol{H}(t;\tau) = \begin{bmatrix} h_{11}(t;\tau) & h_{12}(t;\tau) & \cdots & h_{1N_t}(t;\tau) \\ h_{21}(t;\tau) & h_{22}(t;\tau) & \cdots & h_{2N_t}(t;\tau) \\ \vdots & \vdots & \ddots & \vdots \\ h_{N_r1}(t;\tau) & h_{Nr2}(t;\tau) & \cdots & h_{N_rN_t}(t;\tau) \end{bmatrix} \tag{2.1}$$

where t and τ are the time and the propagation delay and $h_{i,j}(t;\tau)$ denotes the impulse response between the j^{th} transmit antenna and the i^{th} receive antenna.

Let $s_j(t)$ be the signal or symbol transmitted from the j^{th} transmit antenna and vector $\boldsymbol{s}(t) = \{s_1(t), \ldots, s_{N_t}(t)\}^T$ with covariance matrix $\boldsymbol{\Sigma} = \mathbb{E}[\boldsymbol{s}(t)\boldsymbol{s}(t)^\dagger]$. Denote the received signal at the i^{th} receive antenna by $y_i(t)$ and the vector of received signal by $\boldsymbol{y}(t) = \{y_1(t), \ldots, y_{N_r}(t)\}$, which is given by [19]

$$\boldsymbol{y}(t) = \int \boldsymbol{H}(t;\tau)\boldsymbol{s}(t-\tau)d\tau + \boldsymbol{n}(t) \ ,$$

where $\boldsymbol{n}(t)$ is an $N_r \times 1$ additive white Gaussian noise vector.

If the channel is time-invariant, the dependence of the MIMO channel on time vanishes and we can drop the t in $\boldsymbol{H}(t;\tau)$ (i.e., $\boldsymbol{H}(t;\tau) = \boldsymbol{H}(\tau)$). The frequency-domain

representation $\boldsymbol{H}(f) \in \mathbb{C}^{N_r \times N_t}$ is then given by

$$\boldsymbol{H}(f) = \int \boldsymbol{H}(\tau) e^{-j2\pi f \tau} d\tau \ .$$

For the rest of the dissertation, we address the channel, without loss of generality, in its frequency domain. The frequency response channel matrix $\boldsymbol{H}(f)$ can be further simplified as $\boldsymbol{H}$ for a flat-fading or narrowband channel. Thus, the MIMO channel matrix, denoted by $\boldsymbol{H} \in \mathbb{C}^{N_r \times N_t}$, is given by

$$\boldsymbol{H} = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1N_t} \\ h_{21} & h_{22} & \cdots & h_{2N_t} \\ \vdots & \vdots & \ddots & \vdots \\ h_{N_r1} & h_{Nr2} & \cdots & h_{N_rN_t} \end{bmatrix} \tag{2.2}$$

where the entry $h_{i,j}$ represents the frequency response (i.e., attenuation and phase shift) from the j^{th} transmit antenna to the i^{th} receive antenna.

2.1.2 MIMO channel modeling

Various forms of MIMO channels have been investigated during the past decade. In general, most MIMO channel models can be categorized into physical model or analytical model [20]. The physical models build the MIMO channel through the physical parameters, such as the angle of arrival (AoA), angle of departure (AoD) and time of arrival, etc. It describes the characteristics of the MIMO propagation channel as well as the surrounding scattering environments via deterministic and/or stochastic parameterization. For example, deterministic models using either ray-tracing techniques or/and stored measurements to characterize the physical propagation parameters [21]. In the stochastic channel models, the impulse response physical parameters applied to specific or random transmitter, receiver and scatter geometries are modeled in a stochastic manner, such as the extensions

of Saleh-Valenzuela model [22, 23] and WINNER channel model [24].

In contrast to the physical models, analytical channel models characterize the MIMO channel in a analytical way by absorbing the individual impulse responses into a MIMO channel matrix, such as the well-kown i.i.d. model, Kronecker model [25, 26] and Weichselberger model [27]. In particular, a simple but very common MIMO channel model is the i.i.d. Rayleigh fading model, where the entries of the channel matrix $\boldsymbol{H}$ are independent, identically distributed and circular symmetric complex Gaussian. The physical basis of the i.i.d. Rayleigh fading model relies on richly scattered environment with a significant number of multipaths with equal energy spread. The antenna elements should be either critically or sparsely spaced to ensure the independence of the entries, i.e., at least half-wavelength spacing [28]. We can simply model the i.i.d. Rayleigh fading model as [7]

$$h_{i,j} = \sqrt{\gamma/2}(X_1 + iX_2),$$

where X_1 and X_2 are the i.i.d. Gaussian random variables with a zero mean and unit variance and γ is the SNR of the MIMO link, i.e., $\mathbb{E}[|h_{i,j}|^2] = \gamma$. We will use Rayleigh fading model quite often in our simulation of the proposed algorithms. The basic method is to assume a quasi-static flat-fading Rayleigh channel model where the channel is assumed to be stationary for the duration of a burst, but random between bursts.

2.1.3 Channel capacity

MIMO technology has been shown to improve the capacity of the communication link without the need to increase the transmission power. The large spectral efficiency associated with MIMO channels is based on the premise that a rich scattering environment provides independent transmission paths from each transmit antenna to each receive antenna. With a total transmit power constraint P_t, the point-to-point MIMO channel capacity is given

by [8, 29]

$$C(\boldsymbol{H}, \boldsymbol{\Sigma}) = \max_{Tr(\boldsymbol{\Sigma}) \leq P_t} \log_2 \left| \boldsymbol{I} + \boldsymbol{H}\boldsymbol{\Sigma}\boldsymbol{H}^{\dagger}/\sigma^2 \right| \ .$$

If there is no CSI available at the transmitter side, the power is equally split among N_t transmit antennas, and the instantaneous channel capacity is given by

$$C_{SISO}^{noCSIT} = \log_2 \left| \boldsymbol{I} + \frac{P_t}{N_t \sigma^2} \boldsymbol{H}\boldsymbol{H}^{\dagger} \right| \ .$$

The point-to-point MIMO channel capacity can be maximized if the perfect CSI is available at the transmitter side. The optimal capacity is obtained by decomposing the MIMO channel into several parallel SISO channels without interfering with each other and sharing the total transmit power, using singular value decomposition (SVD). Let $r = \mathrm{rank}(\boldsymbol{H})$ and the compact SVD of channel matrix $\boldsymbol{H}$ is given by

$$\boldsymbol{H} = \boldsymbol{A}\boldsymbol{\Lambda}\boldsymbol{B}^{\dagger} \ ,$$

where $\boldsymbol{\Lambda} \in \mathbb{C}^{d \times d}$ is a diagonal matrix containing the r singular values. The $\boldsymbol{A} \in \mathbb{C}^{N_r \times d}$ and $\boldsymbol{B} \in \mathbb{C}^{N_t \times d}$ are the left and right singular matrix corresponding to the r singular values. The power is then allocated in an optimal way, which is known as waterfilling. Equivalently, the optimal capacity can be written as

$$C_{SISO}^{CSIT} = \sum_{i=1}^{r} \log_2(1 + p_i \lambda_i^2/\sigma^2) \ ,$$

where λ_i is the i^{th} singular value of $\boldsymbol{H}$. The waterfilling solution for power allocation is given by

$$p_i = (\mu - \sigma^2/\lambda_i^2)_+$$

where $(x)_+ = \max(x, 0)$. μ can be obtained through a bisection search process, which satisfies $\sum_{i=1}^{r} p_i = P_t$. The covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{A}\mathrm{diag}[p_1, \ldots, p_r]\boldsymbol{A}^{\dagger}$.

2.2 Multiuser MIMO

The attractive spatial multiplexing gain promised by point-to-point MIMO, also known as SU-MIMO, requires a rich multipath propagation environment and sophisticated receivers with multiple antennas. Unlike in the single-user setting, with multiuser MIMO techniques,

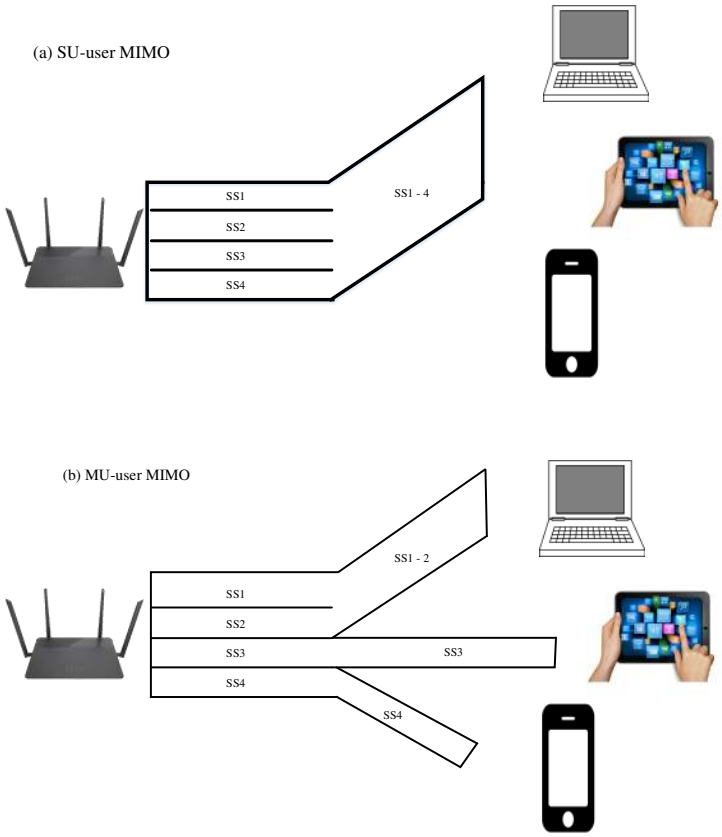

Figure 2.2: SU-MIMO v.s. multiuser MIMO.

the spatial multiplexing of data streams intended for different users can be transmitted simultaneously while users are equipped with single antenna receivers, as shown in Figure 2.2. Therefore, multiuser MIMO communications enable the capacity gains of MIMO while maintaining a low cost for user terminals.

The most substantial cost for a multiuser MIMO system is CSI required at the transmitter side in order to properly serve the spatially multiplexed users. CSI at transmitter side, while not essential in SU-MIMO communication channels, is of critical importance to most

downlink multiuser MIMO precoding techniques. Another challenge involved in multiuser MIMO design lies in the complexity of the scheduling procedure associated with the selection of a combination of users that will be served simultaneously. Optimal scheduling involves exhaustive search whose complexity is exponential in the group size and depends on the choice of precoding, decoding, and CSI feedback technique.

2.2.1 System model

In the downlink illustrated in Figure 2.3, assume there are N_t transmit antennas and K users each equipped with $N_{r,k}$ receive antennas. Let $\boldsymbol{s}_k \in \mathbb{C}^{N_t \times 1}$ be the signal vector transmitted

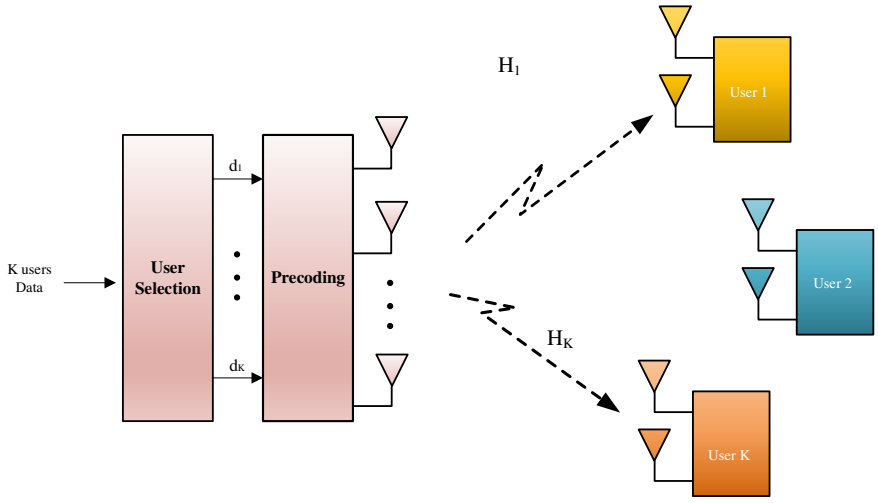

Figure 2.3: System block diagram for a multiuser MIMO system.

for the k^{th} user from the N_t transmit antennas and $\boldsymbol{H}_k \in \mathbb{C}^{N_t \times N_{r,k}}$ be the channel matrix between N_t transmit antennas and $N_{r,k}$ receive antennas of the k^{th} user. The user selection module is required to select a subset of users that can be supported simultaneously, while the precoding calculation module determines the precoding matrices for the active users. These two modules can be implemented either jointly or separately. The received signal at

the k^{th} receiver can be written as

$$\boldsymbol{y}_k = \boldsymbol{H}_k \boldsymbol{s}_k + \underbrace{\sum_{j \neq k, j=1}^{K} \boldsymbol{H}_k \boldsymbol{s}_j}_{\text{inter-user interference}} + \boldsymbol{n}_k \ .$$

where $\boldsymbol{n}_k \in \mathbb{C}$ is the vector white Gaussian noise. The covariance matrix of transmit signal $\boldsymbol{s}_k$ is $\boldsymbol{\Sigma}_k = \mathbb{E}[\boldsymbol{s}_k \boldsymbol{s}_k^{\dagger}]$. With a transmit power constraint P_t, it implies $\sum_{k=1}^{K} \text{Tr}(\boldsymbol{\Sigma}_k) \leq P_t$.

2.2.2 Capacity region

An achievable region for the MIMO broadcasting channel (BC) was first derived in [30] and extended to a more general multiuser MIMO case in [31]. As shown in [32] that the capacity region of a MIMO BC is equal to the dirty paper coding (DPC) rate region. If the user are encoded by the order of $(\pi(1), \pi(2), \ldots, \pi(K))$, then the DPC rate of user $\pi(i)$ can be computed as

$$\mathcal{C}_{\pi(i)}^{DPC} = \log \left| \frac{\boldsymbol{I} + \boldsymbol{H}_{\pi(i)} \left(\sum_{j \geq i} \boldsymbol{\Sigma}_{\boldsymbol{\pi(j)}} \right) \boldsymbol{H}_{\pi(i)}^{\dagger}}{\boldsymbol{I} + \boldsymbol{H}_{\pi(i)} \left(\sum_{j > i} \boldsymbol{\Sigma}_{\boldsymbol{\pi(j)}} \right) \boldsymbol{H}_{\pi(i)}^{\dagger}} \right|, k = 1, \ldots, K \tag{2.3}$$

The DPC rate region, which is the same as the BC capacity region, with a given permutation $(\pi(1), \pi(2), \ldots, \pi(K))$ is given by

$$\mathcal{C}^{DPC} = \bigcup_{\sum_{k=1}^{K} \boldsymbol{\Sigma}_k \leq P_t} \mathcal{C}_{\pi(i)}^{DPC}(\boldsymbol{\Sigma}_{\pi_i})$$

where the expression should in turn be optimized over each possible user ordering. Based on the duality of the MAC and DPC capacity region, the BC capacity region can be calculated through the union of regions of the dual MAC with uplink power allocation meeting the sum power constraint [33].

However, the DPC is difficult to implement in practice due to the complicated encoding and decoding schemes coupled with user ordering. An alternative and more practical technique for multiuser MIMO transmission is known as linear precoding.

2.2.3 Multiuser transmission via linear processing

Linear processing techniques are of interest because of their simplicity. The multiple users are assigned with different precoding matrices at the transmitter side. The precoders are designed jointly based on CSI of all the users to achieve a certain design objective. Typical design criteria include, interference minimization, error probability, sum-rate, signal-to-interference-plus-noise (SINR), etc.

Consider the vector of data signal for the k^{th} user given by $\boldsymbol{x}_k \in \mathbb{C}^{d_k \times 1}$ with d_k data streams. With linear precoding, the data signal of the k^{th} user is mapped to the N_t transmit antenna using a linear precoding matrix $\boldsymbol{V}_k \in \mathbb{C}^{N_t \times d_k}$, that is, $\boldsymbol{s}_k = \boldsymbol{V}_k \boldsymbol{x}_k$. The achievable rate of the k^{th} user is given by

$$R_k = \log_2 \left| \boldsymbol{I} + \tilde{\boldsymbol{R}}_k^{-1} \boldsymbol{H}_k \boldsymbol{V}_k \boldsymbol{V}_k^{\dagger} \boldsymbol{H}_k^{\dagger} \right| ,$$

where

$$\tilde{\boldsymbol{R}}_k = \sum_{i=1, i \neq k}^{K} \boldsymbol{H}_k \boldsymbol{V}_i \boldsymbol{V}_i^{\dagger} \boldsymbol{H}_k^{\dagger} + \sigma_k^2 \boldsymbol{I}$$

is the interference-plus-noise covariance matrix at the k^{th} user's receiver. The received signal $\boldsymbol{y}_k$ is then equalized by the linear combiner $\boldsymbol{U}_k \in \mathbb{C}^{N_{r,k} \times d_k}$ so that the estimated signal at the k^{th} user's receiver is given by $\hat{\boldsymbol{x}}_k = \boldsymbol{U}_k^{\dagger} \boldsymbol{y}_k$.

Zero-forcing (ZF)

One of the simplest linear precoding technique is known as zero-forcing (ZF), which can eliminate the inter-user interference [34]. The ZF precoder was first derived for multiuser MIMO transmission with single antenna at each receiver. For this special case,

the channel vector between the N_t transmit antenna and single receive antenna is denoted by $\boldsymbol{h}_k \in \mathbb{C}^{1\times N_t}$ for the k^{th} user. Let the precoder for the k^{th} user be $\boldsymbol{v}_k \in \mathbb{C}^{N_t\times 1}$ and $\boldsymbol{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_K\}$. The ZF precoder is given by

$$\boldsymbol{V} = \boldsymbol{H}^{\dagger}(\boldsymbol{H}\boldsymbol{H}^{\dagger})^{-1}\boldsymbol{D}$$

where $\boldsymbol{H} = \left[\boldsymbol{h}_1^T, \boldsymbol{h}_2^T, \ldots, \boldsymbol{h}_K^T\right]^T$ and $\boldsymbol{D}$ is a diagonal matrix which adjusts the transmit power so that $Tr(\boldsymbol{V}\boldsymbol{V}^{\dagger}) \leq P_t$. By using ZF precoding, each stream can only be heard by its intended receiver, and the interference are nulled out at other unintended receivers. Due to the property of channel inversion, at most N_t users can be served simultaneously using ZF precoding scheme.

The idea of ZF method can be extended to the multiuser MIMO systems for terminals with multiple receive antennas, which is called block diagonalization (BD) [35]. With BD, each user's precoding matrix is restricted to lie in the null space of other cocurrent users' channels. Therefore, the inter-user interference can be eliminated as $\boldsymbol{H}_i\boldsymbol{V}_k = 0$, for $i \neq k$. Let $\tilde{\boldsymbol{H}}_k = \left[\boldsymbol{H}_1^T, \ldots, \boldsymbol{H}_{k-1}^T, \boldsymbol{H}_{k+1}^T, \ldots, \boldsymbol{H}_K^T\right]^T$. The precoder matrix of the k^{th} user should lie in the null space of $\tilde{\boldsymbol{H}}_k$. The rank condition $rank(\boldsymbol{H}_k\boldsymbol{V}_k) \geq 1$ should be satisfied. For example, with the assumption that each element in $\boldsymbol{H}_k$ is generated by an i.i.d. complex Gaussian distribution and user utilized all its receiver antennas, the maximum number of simultaneous users is $\lceil N_t/N_r \rceil$ for $N_r, k = N_r, \forall k$.

For ZF methods, finding optimal concurrent user group, however, is computationally prohibitive, especially for a large user population. Different suboptimal user selection algorithms are proposed for ZF methods [34, 36, 37, 38, 39], achieving certain tradeoffs between the aggregate performance and complexity.

Minimum mean-square-error (MMSE)

Different from ZF methods, which completely nullify the interference and could cause an elevated noise level, the minimum mean-square-error (MMSE) criterion balances the effects of noise enhancement and interference suppression [40]. For multiuser MIMO, the MMSE criterion minimizes the expected sum of the norms between each $\tilde{\boldsymbol{x}}_k$ and $\boldsymbol{x}_k$, yielding the problem for designing the precoders $\boldsymbol{V}_k$'s and combiners $\boldsymbol{U}_k$'s

$$\begin{aligned} \min_{\{\boldsymbol{V}_k,\boldsymbol{U}_k\}_{k=1}^K} \quad & \sum_{k=1}^{K} Tr\left(\mathbb{E}\left[(\tilde{\boldsymbol{x}}_k-\boldsymbol{x}_k)(\tilde{\boldsymbol{x}}_k-\boldsymbol{x}_k)^\dagger\right]\right) \\ s.t. \quad & \sum_{k=1}^{K} Tr(\boldsymbol{V}_k\boldsymbol{V}_k^\dagger) \le P_t \ . \end{aligned} \tag{2.4}$$

Due to the inherent interdependence between the precoders and combiners, the closed-form solution for the global optimal $\boldsymbol{V}_k$'s and $\boldsymbol{U}_k$'s are unknown. The alternating MMSE solution can be obtained via Karush-Kuhn-Tucker (KKT) conditions. With given $\boldsymbol{V}_k$'s, the MMSE combiner is given by

$$\boldsymbol{U}_k^{MMSE} = \left(\tilde{\boldsymbol{R}}_k + \boldsymbol{H}_k\boldsymbol{V}_k\boldsymbol{V}_k^\dagger\boldsymbol{H}_k^\dagger\right)^{-1}\boldsymbol{H}_k\boldsymbol{V}_k.$$

With given $\boldsymbol{U}_k$'s, the MMSE precoder is given by

$$\boldsymbol{V}_k^{MMSE} = \left(\mu\boldsymbol{I} + \sum_{k=1}^{K}\boldsymbol{H}_k^\dagger\boldsymbol{U}_k\boldsymbol{U}_k^\dagger\boldsymbol{H}_k\right)^{-1}\boldsymbol{H}_k^\dagger\boldsymbol{U}_k.$$

where μ is the Lagrangian multiplier chosen to meet the total power constraint on the precoder. It can be optimized via a simple bisection search process.

SINR Maximization

For sum-rate maximization, a desirable metric would directly account for the postprocessing SINR at each receiver. The SINR of the i^{th} stream for the k^{th} user is

$$\gamma_{k,i} = \frac{\boldsymbol{u}_{k,i}^{\dagger}\boldsymbol{H}_k\boldsymbol{v}_{k,i}\boldsymbol{v}_{k,i}^{\dagger}\boldsymbol{H}_k^{\dagger}\boldsymbol{u}_{k,i}}{\boldsymbol{u}_{k,i}^{\dagger}\sigma_k^2\boldsymbol{u}_{k,i} + \sum_{l=1}^{K}\boldsymbol{u}_{k,i}^{\dagger}\boldsymbol{H}_k\boldsymbol{V}_l\boldsymbol{V}_l^{\dagger}\boldsymbol{H}_k^{\dagger}\boldsymbol{u}_{k,i} - \boldsymbol{u}_{k,i}^{\dagger}\boldsymbol{H}_k\boldsymbol{v}_{k,i}\boldsymbol{v}_{k,i}^{\dagger}\boldsymbol{H}_k^{\dagger}\boldsymbol{u}_{k,i}},$$

where $\boldsymbol{u}_{k,i}$ and $\boldsymbol{v}_{k,i}$ are the i^{th} column of $\boldsymbol{U}_k$ and $\boldsymbol{V}_k$.

However, the total SINR for multiple terminals with multiple antennas is not strictly defined in the literature. Different global SINR metrics are constructed and used for multiuser MIMO optimization. For example, in [41], the MIMO precoder and combiner are jointly optimized to maximize the minimum SINR. In [42], the SINR metric is defined as the sum signal power across all receivers divided by the sum interference power, incorporating the interstream interference. While most prior work with max-SINR criterion has focused on computing the precoder and combiner, a priori specification of the active receivers, as well as the active streams for each receiver is required.

2.3 Dense Wireless Network with Single-hop MIMO

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. While the performance of individual wireless devices can be improved due to the adoption of advanced physical layer and signal processing techniques, most wireless networks in the unlicensed band are experiencing difficulties to achieved the anticipated overall performance. A major challenge is the high-level co-channel interference caused by the proliferation of both devices and access points (APs) in a limited-spectrum environment, which has led to poor per-user bandwidth.

Traditional techniques, such as assigning orthogonal channels to different APs and as-

signing non-overlapping time slots to different users for 802.11-based WLANs, at best equally divide the limited bandwidth among users. Moreover, since there are typically 3 orthogonal channels available in the unlicensed band, the practice of spatial reuse can only promise the performance increase that scale almost linearly to a factor of 3. The co-channel interference introduced by any additional AP will limit the performance increase, or even reduce the overall performance, as shown in Figure 2.4.

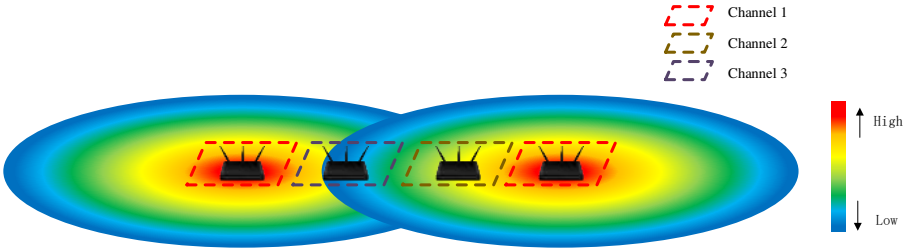

Figure 2.4: An example of wireless LAN deployment.

The concept of network MIMO, which is a form of cooperative MIMO or distributed MIMO, has attracted significant research interest because of its ability to dramatically increase the throughput of wireless networks [43, 44, 45, 46]. This approach is particularly suited for dense enterprise networks with clusters of closely deployed APs [47]. A common scenario is that APs share a network gateway with one Internet connection. In this scenario, multiple APs can be clustered and cooperate to control the lower-layer parameters and to optimize the overall performance and fairness, by connecting via low-latency links to a shared controller. The feasibility of synchronizing multiple cooperative APs has been demonstrated in existing work [47, 48]. To reap the 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 thesis, we target dense enterprise wireless networks where there are multiple APs and a large number of users within a small geographical area. These dense network scenarios are among the most challenging for satisfying user demands.

2.3.1 Access point cooperation

In general, two levels of downlink cooperation have been discussed in the literature, primarily for cellular networks [43, 49]. One possibility is that the cooperative transmitters obtain CSI of both direct and interfering links. This information allows the APs to coordinate their signaling strategies, such as precoder design and power allocation, to effectively suppress interference across different users. We refer to this approach as *interference coordination* (IC). If the cooperative transmitters are tied together via high-speed links, as the case in most enterprise wireless network deployments, they can share not only the CSI, but also the data signals intended for the users, which enables a more powerful form of cooperation. In this case, multiple APs can jointly craft their downlink signals to cooperatively serve the users and the interference can be used to enhance performance, not degrade it. We refer to this approach as *cooperative processing* or *full cooperation*. With the dramatic improvement of the wired speeds to the last hop, the low-latency backhaul connection facilitates the realization of the full cooperation among a small number of APs, which promises better overall performance.

Considering the practical constraints such as complexity of coordination, backhaul limitations, and computational overhead, the practical way to realize network MIMO in dense environments is to group a small number of nearby APs into a cluster as shown in Figure 5.1. Thus, we divide a large enterprise wireless network into clusters, where the APs within the same cluster can cooperate with each other. [1] This clustered structure can be extended to large enterprise wireless networks by forming multiple clusters, where the APs within the same cluster can cooperate with each other. Determining AP clusters is beyond the scope of this dissertation. Actually, many environment provides a natural way of clustering APs or any reasonable clustering algorithm can provide the type structure we envision.

Throughout this dissertation, we consider a scenario in which single-hop wireless net-

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

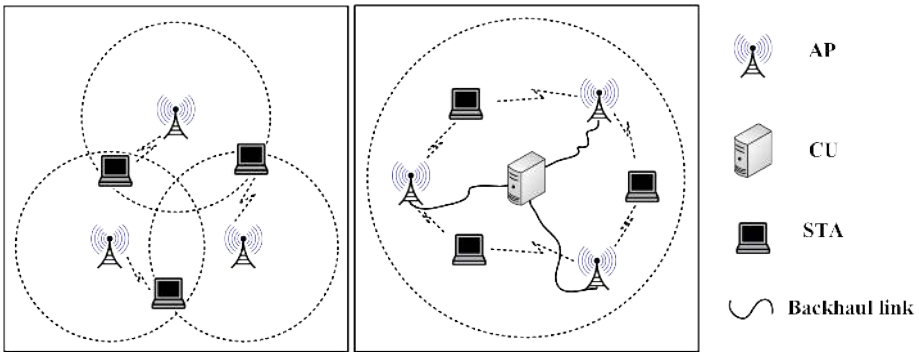

Figure 2.5: Traditional multi-cell WLAN (left) and clustered WLAN (right).

works are densely deployed over a region, where the areas served by different APs can overlap. These APs can form a number of clusters, and APs within each cluster serve the users via *cooperative processing*. 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 a network controller connected to all APs within a cluster. We assume predetermined AP clusters and user association and are particularly interested in optimizing the performance within a single cluster.

2.3.2 Linear precoder and combiner design

We consider a MIMO network with M cooperative access points (APs), where the m^{th} AP is equipped with $N_{t,m}$ antennas. We assume that there are K users with $N_{r,k}$ antennas for the k^{th} user. Let $N_t = \sum_{m=1}^{M} N_{t,m}$ and $N_r = \sum_{k=1}^{K} N_{r,k}$ 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 $\boldsymbol{H}_k \in \mathbb{C}^{N_{r,k} \times N_t}$. The data vector $\boldsymbol{x} = \left[\boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_K^T\right]^T$ is jointly precoded by the M APs using the linear precoding matrix $\boldsymbol{V} = [\boldsymbol{V}_1, \ldots, \boldsymbol{V}_K]$. $\boldsymbol{x}_k \in \mathbb{C}^{N_{r,k} \times 1}$ is the transmit signal vector for receiver k, and $\boldsymbol{x}_k$ is assumed to be independently encoded Gaussian codebook symbols with $\mathbb{E}[\boldsymbol{x}_k \boldsymbol{x}_k^\dagger] = \boldsymbol{I}$, where $(\cdot)^\dagger$ is the conjugate transpose of $(\cdot)$. It is assumed that the k^{th} user has $N_{r,k}$ parallel data streams, although some of the streams can have a rate of

zero. $\boldsymbol{V}_k = \left[\boldsymbol{V}_{k,1}^T, \ldots, \boldsymbol{V}_{k,M}^T\right]^T \in \mathbb{C}^{N_t \times N_{r,k}}$, where $\boldsymbol{V}_{k,m}$ is the partition of $\boldsymbol{V}_k$ applied at the m^{th} AP to precode the signals of user k.

The received vector at user k is given by

$$\boldsymbol{y}_k = \boldsymbol{H}_k \boldsymbol{V}_k \boldsymbol{x}_k + \sum_{l=1, l\neq k}^{K} \boldsymbol{H}_k \boldsymbol{V}_l \boldsymbol{x}_l + \boldsymbol{n}_k \ , \tag{2.5}$$

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

$$\boldsymbol{R}_{\bar{k}} = \sum_{l=1, l\neq k}^{K} \boldsymbol{H}_k \boldsymbol{V}_l \boldsymbol{V}_l^{\dagger} \boldsymbol{H}_k^{\dagger} + \boldsymbol{R}_{n_k} \ . \tag{2.6}$$

The instantaneous data rate in bits/s/Hz of the k^{th} receiver before receive filtering is given by

$$R_k = \log_2 \left| \boldsymbol{I} + \boldsymbol{R}_{\bar{k}}^{-1} (\boldsymbol{H}_k \boldsymbol{V}_k \boldsymbol{V}_k^{\dagger} \boldsymbol{H}_k^{\dagger}) \right| \ . \tag{2.7}$$

Assume the received vector $\boldsymbol{y}_k$ is equalized using the linear combiner $\boldsymbol{U}_k \in \mathbb{C}^{N_{r,k} \times N_{r,k}}$. The received signal of the k^{th} receiver is given by $\hat{\boldsymbol{x}}_k = \boldsymbol{U}_k^{\dagger} \boldsymbol{y}_k$, which results in the MSE covariance matrix of the k^{th} user as

$$\boldsymbol{E}_k = \mathbb{E}\left[(\boldsymbol{U}_k^{\dagger} \boldsymbol{y}_k - \boldsymbol{x}_k)(\boldsymbol{U}_k^{\dagger} \boldsymbol{y}_k - \boldsymbol{x}_k)^{\dagger} \right] \tag{2.8}$$

and its postprocessing data rate as

$$\hat{R}_k = \log \left| \boldsymbol{I} + (\boldsymbol{U}_k^{\dagger} \boldsymbol{R}_{\bar{k}}^{-1} \boldsymbol{U}_k)(\boldsymbol{U}_k^{\dagger} \boldsymbol{H}_k \boldsymbol{V}_k \boldsymbol{V}_k^{\dagger} \boldsymbol{H}_k^{\dagger} \boldsymbol{U}_k) \right| \ . \tag{2.9}$$

We aim to design the linear precoder and combiner that performs a combination of interference suppression and spatial multiplexing with the measured CSI. We particularly focus on developing low-complexity and scalable linear processing strategies, which can work well for the dense environment settings with large user populations. Practical constraints

will be taken into account, such as the per-node power constraint, overhead for channel estimation and various levels of CSI accuracy. To facilitate the analysis in the following chapters, we will introduce the per-node power constraint in this section. Assume the maximum transmit power of the m^{th} AP is P_m. Since the transmit antennas are from distributed APs, the precoder $\boldsymbol{V}$ needs to satisfy a set of per-AP power constraints expressed as,

$$Tr(\boldsymbol{\Gamma}_m \boldsymbol{V}\boldsymbol{V}^\dagger) \leq P_m, m = 1, \ldots, M \ , \tag{2.10}$$

or equivalently,

$$\sum_{k=1}^{K} Tr(\boldsymbol{\Gamma}_m \boldsymbol{V}_k \boldsymbol{V}_k^\dagger) \leq P_m, m = 1, \ldots, M \ , \tag{2.11}$$

where a diagonal matrix $\boldsymbol{\Gamma}_m \in \mathbb{R}^{N_t \times N_t}$ is introduced for the m^{th} AP, in order to select the partition of precoding matrix $\boldsymbol{V}$ applied at AP m. Thus, $\boldsymbol{\Gamma}_m$ contains ones on the diagonal elements corresponding to the antennas of AP m and zeros elsewhere, i.e.,

$$\boldsymbol{\Gamma}_m = \text{diag}\left\{ \underbrace{0, \ldots, 0}_{\sum_{m=1}^{m-1} N_{t,m}}, \underbrace{1, \ldots, 1}_{N_{t,m}}, \underbrace{0, \ldots, 0}_{\sum_{m=m+1}^{M} N_{t,m}} \right\} \ .$$

Note that in the special case of $M = 1$, the per-AP constraint reduces to the conventional sum power constraint. Later, in Chapter 3 and Chapter 4, different linear precoding algorithms will be discussed under the per-AP power constraint.

2.3.3 MIMO link scheduling

The physical-layer solution using linear precoder and combiner performs both spatial multiplexing and interference suppression, which is primarily focuses on optimizing the performance at a given moment of time. However, in our target dense environments, there are many users sharing the resources within a cluster. Scheduling to achieve both high-throughput and fairness can be a challenging problem. The reason is that the per-user

performance with multiuser MIMO transmission differs from one user group to another, and this difference is highly coupled with the physical-layer solution. Determining the active user group for a time slot requires the knowledge of the performance of these multiuser MIMO links, which requires the intensive computation of the precoder and combiner. Besides, the fairness consideration will further complicate the design of the scheduling algorithm.

In this dissertation, we aim to build a fair and high-throughput schedule for a cluster of M cooperative APs and K users. The network central controller collects the physical layer information from distributed APs and generate a explicit communication schedule. The objective for the scheduling algorithm is to schedule a highly-optimized set of communications for each time slot, which achieves high aggregate performance and satisfies certain fairness criteria. We will investigate different scheduling algorithms that can work with the physical-layer algorithm in Chapter 5. Moreover, scheduling strategies that can accommodate both static and mobile users will be discussed in Chapter 6.

2.3.4 Integration with 802.11 protocol

To realize the clustered WLANs with AP cooperation we envisioned, we need to revisit and tailor the 802.11 protocols for distributed APs. In particular, the 802.11ac, a faster and more scalable version of 802.11n, is a significant landmark for WLANs, as it pushes towards higher rate limits by enabling downlink multiuser MIMO transmission. In this section, we will discuss some modifications to 802.11 protocols for the realization of downlink multiuser MIMO communication with cooperative APs.

Client association

In conventional WLANs, APs advertise their presence by broadcasting beacon frames. Prior to association, clients gather information about the APs by scanning the channels one by one either through passive scanning or active scanning. When a client use passive

scanning, it moves into each channel and listens the beacons on the channel. With active scanning, the client station sends out probe request frames on each channel. APs respond to these requests with probe response frames, which are similar to beacon frames. If a client station receives beacons or probe responses from multiple APs, the default 802.11 rule use received signal strength indicator (RSSI) as the association metric, namely, connecting the client to the AP with the strongest RSS.

In our target network with AP cooperation, a client is actually associated to a cluster, instead of a specific AP as in conventional WLANs. This can be achieved with a modification of the conventional Client-to-AP association method. Upon determining the associated clients for each AP, the central controller of the corresponding cluster gathers the information from the cooperative APs and assign the cluster-based IDs to the associated clients, which are then shared by the cooperative APs. Alternatively, the cluster identity can be included in the beacon or probe response frame, i.e., each cluster has its own unique cluster ID. When a client receives beacons or probe responses from multiple APs, it evaluates the effective RSS from each cluster if these APs have different cluster IDs. The effective RSS can be obtained by maximal ratio combining of the beacons or probe response signals from APs with the same cluster ID. The cluster-based ids are given to each client upon association.

CSI feedback mechanism

Since the multiuser MIMO performance is very sensitive to the interference, downlink multiuser MIMO works well with the explicit beamforming feedback in 802.11ac. Here, to enable the configuration of distributed APs, we consider a modification of the explicit feedback mechanism specified in 802.11ac. In 802.11ac, before a multiuser MIMO transmission, an AP initiates channel sounding by transmitting a VHT null data packet (NDP) announcement, which specifies the set of users that are going to be polled for CSI feedback. After the NDP announcement, the AP transmits an NDP, which is used by the receivers for

channel estimation.

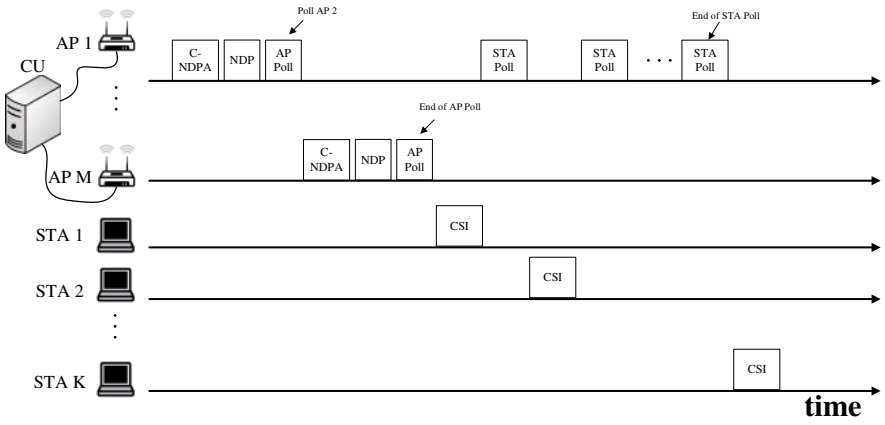

Figure 2.6: CSI feedback mechanism for AP cooperation

With AP cooperation, each receiver needs to estimate the composed channel from all APs. This can be done by modifying the single AP mechanism from 802.11ac, as shown in Figure 2.6. The CU synchronizes the cooperative APs within the same cluster. The APs transmit the cooperative NDP announcement (C-NDPA) and NDP sequentially in a pre-determined order to enable the receivers to measure the wireless channels. The C-NDPA frame identifies the AP cooperation mode, transmitter addresses and intended recipients' address. The C-NDPA can be reduce to the conventional NDPA in 802.11 if there is only one AP to transmit. Upon transmission of the NDP for channel estimation, an AP will send the AP-poll to notify the next AP for C-NDPA and NDP transmission. Each client estimates the channel between itself and each AP, i.e. the channel matrix between the m^{th} AP and k^{th} client denoted by $\boldsymbol{H}_{k,m} \in \mathbb{C}^{N_r \times N_{t,m}}$. After receiving the last NDP, the k^{th} client concatenates its channel matrix from M cooperative APs as

$$\boldsymbol{H}_k = [\boldsymbol{H}_{k,1}, \boldsymbol{H}_{k,2}, \ldots, \boldsymbol{H}_{k,M}].$$

For CSI feedback, a master AP is assigned to poll the receives one by one by sending a STA-poll, e.g., AP 1 is selected as the master AP in Figure 2.6. The first user will send

back its compressed CSI report immediately after receiving the end of AP-poll, while other users will send their CSI after receiving the corresponding STA-poll.

The CSI feedback is always sent at the lowest modulation rate for reliability. To alleviate the feedback overhead, we use a compressed beamforming report as specified in 802.11ac for each polled user. It uses a quantized representation of the estimated channel, based on the SVD of the channel. Let $\boldsymbol{H}_k$ be the channel matrix of k^{th} user, which can be represented via compact SVD $\boldsymbol{H}_k = \boldsymbol{A}_k \boldsymbol{S}_k \boldsymbol{B}_k^{\dagger}$, where $\boldsymbol{S}_k \in \mathbb{C}^{N_r \times N_r}$ is the diagonal matrix containing the singular values in a decreasing order. $\boldsymbol{A}_k \in \mathbb{C}^{N_r \times N_r}$ and $\boldsymbol{B}_k \in \mathbb{C}^{N_t \times N_r}$ 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. $\boldsymbol{B}_k$ is a semi-unitary matrix with $\boldsymbol{B}_k^{\dagger}\boldsymbol{B}_k = \boldsymbol{I}$ and $\boldsymbol{B}_k^{\dagger}$ forms an orthonormal row basis of the channel matrix $\boldsymbol{H}_{k,n}$. Based on 802.11ac, the right singular matrix $\boldsymbol{B}_k$ can be decomposed using the Givens decomposition:

$$\boldsymbol{B}_k = \left\{ \prod_{i=1}^{N_r} \left(\boldsymbol{D}_k^i \prod_{j=i+1}^{N_t} \boldsymbol{G}_k^{i,j} \right) \right\} \tilde{\boldsymbol{I}} \boldsymbol{\Phi}_k^{\dagger} \,, \tag{2.12}$$

where $\tilde{\boldsymbol{I}}$ is a matrix containing the first N_r columns of an $N_t \times N_t$ unitary matrix. $\boldsymbol{D}_k^i = \mathrm{diag}\left(\mathbf{1}_{i-1}, e^{j\phi_{i,i}}, \ldots, e^{j\phi_{N_t-1,i}}, 1\right)$ and $\boldsymbol{G}_k^{i,j}$ is the Givens rotation matrix

$$\boldsymbol{G}_k^{j,i} = \begin{bmatrix} \boldsymbol{I}_{i-1} & & & & \\ & \cos\psi_{i,j} & & \sin\psi_{i,j} & \\ & & \boldsymbol{I}_{j-i-1} & & \\ & -\sin\psi_{i,j} & & \cos\psi_{i,j} & \\ & & & & \boldsymbol{I}_{N_t-j} \end{bmatrix}$$

where $\boldsymbol{\Phi}_k$ is a diagonal matrix, which can be absorbed into $\boldsymbol{S}_k$. For example, 802.11ac will

quantize and feedback the angles $\phi_{i,j}$ and $\psi_{i,j}$ using a uniform quantizer [50] as follows,

$$\psi = \frac{k\pi}{2^{b_\psi}+1} + \frac{\pi}{2^{b_\psi}+2}, k = 0, 1, \ldots, 2^{b_\psi} - 1 \ ,$$

$$\phi = \frac{k\pi}{2^{b_\phi}+1} + \frac{\pi}{2^{b_\phi}+2}, k = 0, 1, \ldots, 2^{b_\phi} - 1 \ ,$$

where b_ψ and b_ϕ are the number of bits used to quantize ψ and ϕ. Two different feedback types are specified for multiuser MIMO transmission. Type I uses 5 bits for ψ and 7 bits for ϕ, while type II uses 7 bits for ψ and 9 bits for ϕ. The SNR value represented by the singular value is quantized using 8 bits for each data stream. The details can be found in [50]. For example, there are 13 pairs of angles for a 8×2 matrix $\boldsymbol{B}_k$, which requires 156 bits using type I codebook and 208 bits using type II codebook. Obviously, higher quantization bits leads to better approximation of $\boldsymbol{B}_k$ at the price of larger feedback overhead.

Multiuser MIMO transmission

Multiuser MIMO technique in 802.11ac is also referred to as spatial diversity multiple access (SDMA). The reported CSI allows the AP to calculate the multiuser MIMO user group and the corresponding precoding matrices. However, this calculation is not specified in the standard. Later in this dissertation, we will investigate different algorithms to determine the user group and MIMO weights specification. To initiate the multiuser MIMO transmission, it clearly requires knowing which clients are served simultaneously, which is calculated by the central controller after collecting the up-to-date CSI. When the precoders and combiners are jointly calculated by the central controller, the head AP in the cluster needs to broadcast the IDs of the active clients in the cluster and their combining matrices, before transmitting the data packets with the aid of other cooperative APs. If the combiner is determined at the receiver of each active client based on its precoder and channel matrix, the head AP can broadcast the precoders to the client for combiner calculation.

With a certain user group for multiuser MIMO transmission, the APs need to deter-

mine the data rate that best matches the channel quality, which is still an open problem for 802.11ac. Many wireless protocols, including 802.11, supports multiple bit-rates, achieved by different modulation and coding schemes (MCSs). For instance, IEEE 802.11ac supports 10 different modulation and coding scheme options, which are shown in Table I. The corresponding bit-rates can be found in [50] with up to 8 spatial streams.

Table 2.1: Modulation and code rate in IEEE 802.11ac

MCS Index	Modulation	Code Rate
0	BPSK	1/2
1	QPSK	1/2
2	QPSK	3/4
3	16-QAM	2/3
4	16-QAM	3/4
5	64-QAM	2/3
6	64-QAM	3/4
7	64-QAM	5/6
8	256-QAM	3/4
9	256-QAM	5/6

Similar to 802.11ac, we assume multiple streams use the same MCS for a certain client. Obviously, the MCS selection procedure is required to determines the highest possible data rates for clients. Various rate selection algorithms are proposed in existing work, which have been shown to work well for the legacy 802.11a/b/g networks and can be adjusted for MIMO settings [51, 52, 53, 54]. For example, the Minstrel RA algorithm is a general purpose strategy proposed for 802.11 networks, as part of the MadWifi driver, that rapidly became widely accepted [54]. To facilitate the analysis in this dissertation, we simply follow the rationale of Minstrel by tracking the previous statistics periodically and maintaining the packet error rate for each bit-rate. The objective of rate selection is to determine the bit-rate that maximizes the MAC layer throughput which is calculated as bit-rate$*$(1-PER), where PER is the packet error rate at a given bit-rate. For a client with multiple data streams, we obtain the packet reception rate (i.e., 1-PER) by multiplying the probabilities that each stream of the packet is decoded successfully and independently.

Multiple access techniques

Carrier sense multiple access with collision avoidance (CSMA/CA) is a random multiple access scheme used by the 802.11 standard. Prior to transmitting, a node first listens to the shared channel. If another node was heard or detected, the node will wait for a random period of time before listening again. Otherwise, the channel is identified as idle, and the node acquires the channel and transmits its packets. The random waiting time consists of a fixed duration of waiting time and a random contention window (CW), which is used to resolve the potential contention among nodes that trying to access the channel.

Although this distributed MAC protocol has the advantages from the perspective of complexity, scalability and robustness, high contention level can significantly lower its efficiency, which is a common phenomenon in dense environments [55]. In fact, centralized MAC protocols are more suitable for the high-density wireless networks, which is more controllable and enables the cooperation among distributed nodes. The communication schedule proposed in this dissertation can be carried out using a time division multiple access (TDMA) MAC. With TDMA, transmission time is divided into a number of time slots, which generally have the same duration. It is a reservation-based and contention-free multiple access scheme, which assigns the communication links into different time slots [56]. We can aggregate as many packets as can fit within a time slot with fixed duration and have each receiver simultaneously acknowledge these packets within one time slot.

When operating in infrastructure mode, the 802.11 standard also defines a point coordination function (PCF), which resides in an AP to coordinate the communication and can operate on top of distributed contention function (which employs CSMA/CA). The basic functionality of PCF is to let the AP acquire the channel for a fixed period of time and poll the transmission of its clients without contention. Therefore, the PCF allows the implementation of scheduled communications by polling the communication of each multiuser MIMO group. Alternatively, the target communication schedule determined by the central entity might also be implemented in a purely distributed fashion as described in [57],

which achieves the "scheduled WiFi" using the distributed contention mechanism in the 802.11 distributed contention function. The approach in [57] was originally proposed for single-input-single-output transmission and more work is needed to incorporate MIMO techniques into the approach.

2.4 Chapter Summary

In this chapter, we first reviewed the basic concepts and principles for MIMO communications. To be specific, we elaborated the MIMO physical layer model and analyzed the capacity region for both point-to-point MIMO and multiuser MIMO. We have also introduced the linear processing for multiuser MIMO transmission to achieve a combination of interference suppression and spatial multiplexing. With the aid of advanced MIMO technique, we then further proposed an AP cooperation approach to alleviate the high-interference problem encountered in dense wireless networks. We explored the open problems that need to be solved for optimizing the performance in coordinated-AP environments. Finally, we discussed details of integrating the proposed schemes for AP cooperation with the 802.11 protocols.

CHAPTER 3

EFFICIENT USER SELECTION FOR BLOCK DIAGONALIZATION

3.1 Introduction

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 [35, 58]. 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 [59, 60, 61, 39]. These greedy approaches incrementally select one user in each iteration [59, 60, 61, 39]. A capacity-based algorithm of this type, referred as the c-algorithm, is proposed in [59]. 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 [59, 60] 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 [61, 39] reduce the computational cost but achieve lower aggregate performance.

In this chapter, 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 can work well with the explicit CSI feedback mechanism as discussed in Section 2.3.4. 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 60 users, we can achieve the same aggregate performance as the algorithm of [59] with about 1/5 the running time. Alternatively, with the same running time as the algorithm of [61], 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.

3.2 System Model

5.1 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 $\boldsymbol{x}_k \in \mathbb{C}^{N_r \times 1}$ is jointly precoded by the M APs

using the precoding matrix $\boldsymbol{V} = [\boldsymbol{V}_1, \ldots, \boldsymbol{V}_K]$. $\boldsymbol{V}_k \in \mathbb{C}^{MN_t \times N_r}$ is the partition of $\boldsymbol{V}$ 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

$$\boldsymbol{y}_k = \boldsymbol{H}_k \boldsymbol{V}_k \boldsymbol{x}_k + \sum_{j=1, j\neq k}^{K} \boldsymbol{H}_k \boldsymbol{V}_j \boldsymbol{x}_j + \boldsymbol{n}_k \ , \tag{3.1}$$

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 $(\cdot)$. With the explicit CSI feedback mechanism discussed in Section 2.3.4, each user will sent the quantized right singular matrix and singular values of $\boldsymbol{H}_k$ to the AP side, which are denoted by $\boldsymbol{B}_k \in \mathbb{C}^{MN_t \times N_r}$ and $\boldsymbol{S}_k = \text{diag}\{s_1, \ldots, s_{N_r}\}$, respectively. Let $\tilde{\boldsymbol{B}}_k$ and $\tilde{\boldsymbol{S}}_k$ be the quantized $\boldsymbol{B}_k$ and $\boldsymbol{S}_k$ available at the AP side.

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 $\boldsymbol{V}_k$ such that $\boldsymbol{H}_j \boldsymbol{V}_k = 0$ for $j \neq k$. In other words, the precoder $\boldsymbol{V}_k$ 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 $\lceil MN_t/N_r \rceil$ [59]. 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

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

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

QR decomposition,

$$\bar{\boldsymbol{B}}_{\pi_k} = \left[\boldsymbol{W}_{\pi_k}^{(1)}\ \boldsymbol{W}_{\pi_k}^{(0)}\right] \begin{bmatrix} \boldsymbol{R}_{\pi_k} \\ \boldsymbol{0}_{(MN_t-n)\times n} \end{bmatrix}, \tag{3.2}$$

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 $\bar{\boldsymbol{B}}_{\pi_k}$ and $\boldsymbol{W}_{\pi_k}^{(0)} \in \mathbb{C}^{MN_t\times(MN_t-n)}$ forms the null space of $\bar{\boldsymbol{B}}_{\pi_k}^{\dagger}$. Thus, the columns of $\boldsymbol{V}_{\pi_k}$ can be chosen as the linear combination of those in $\boldsymbol{W}_{\pi_k}^{(0)}$. For example, we can simply choose $\boldsymbol{V}_{\pi_k} = \boldsymbol{W}_{\pi_k}^{(0)}$.

3.3 PBUS User Selection

The aggregate performance of the MU-MIMO system is largely dependent on the selection of simultaneous users. Unlike the conventional algorithms [59, 60, 61, 39], 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).

3.3.1 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 (i.e., U_1, U_2, U_3 and U_4) 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

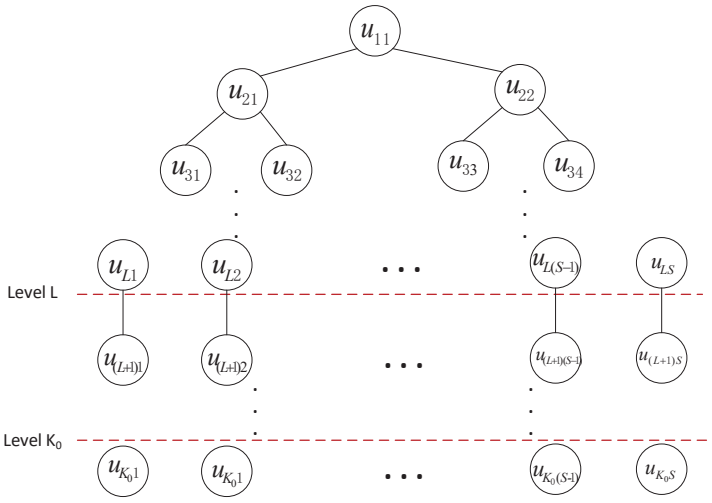

Figure 3.1: Binary tree-based user grouping with $S = 2^{L-1}$ and $1 \leq L \leq K_0$.

user with the highest interference-free data rate is selected as the root of the tree at level 1. As shown in Figure 3.1, 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.

The details of each step in PBUS are elaborated next.

3.3.2 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

$$FM_{k,j} = \left[\gamma\left(\frac{\tilde{\boldsymbol{S}}_k \tilde{\boldsymbol{B}}_k \boldsymbol{N}_j}{\sigma_k ||\boldsymbol{N}_j||_F} \right), \gamma\left(\frac{\tilde{\boldsymbol{S}}_j \tilde{\boldsymbol{B}}_j \boldsymbol{N}_k}{\sigma_j ||\boldsymbol{N}_k||_F} \right) \right]^T , \tag{3.3}$$

where $\boldsymbol{N}_k = \boldsymbol{I} - \tilde{\boldsymbol{B}}_k \tilde{\boldsymbol{B}}_k^{\dagger}$ forms the null space of $\tilde{\boldsymbol{B}}_k^{\dagger}$. 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 fit-

ness 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 $\text{flip}([a,b]^T) = [b,a]^T$.

3.3.3 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 3.1. 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

$$GM(k, \mathcal{G}_s) = \sum_{i \in \mathcal{G}_s} \begin{bmatrix} 1/|\mathcal{G}_s| \\ 1 \end{bmatrix}^T FM_{k,i} \ , \tag{3.4}$$

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

Table 3.1: Binary Tree-based User Grouping Procedure

input:	K_0, $1 \leq L \leq K_0$, $\mathcal{K} = \{1, \ldots, K\}$,
	$FM_{k,j}$ for $k, j = 1, \ldots, K, k \neq j$
output:	$\mathcal{T}$, $\mathcal{G}_{K_0,j}$ for $j = 1, \ldots, 2^{L-1}$
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:	$\mathcal{T}_i \leftarrow \{u^*_{j,i} \vert j = 1, \ldots, 2^{i-1}\}$
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:	$\mathcal{T}_i \leftarrow \{u^*_{j,i} \vert j = 1, \ldots, 2^{L-1}\}$
18:	**endfor**

$i < L$, 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 3.1. 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 3.1. 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 Figure 3.1 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.

3.3.4 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 Figure 3.1. The achievable sum rate of the j^{th} user group $\mathcal{G}_j$ can be estimated as follows by assuming equal power allocation,

$$\tilde{R}(\mathcal{G}_j) = \sum_{i \in \mathcal{G}_j} \log \left| \boldsymbol{I} + \tilde{\boldsymbol{S}}_i^2 \tilde{\boldsymbol{B}}_i \tilde{\boldsymbol{V}}_{i,j} \tilde{\boldsymbol{V}}_{i,j}^{\dagger} \tilde{\boldsymbol{B}}_i^{\dagger} / \sigma_i^2 \right| , \tag{3.5}$$

where the subscript K_0 for $\mathcal{G}_{K_0,j}$ is omitted and $\tilde{\boldsymbol{V}}_{i,j}$ 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 $\tilde{\boldsymbol{V}}_{i,j} = \boldsymbol{V}_{i,j}\boldsymbol{P}_{i,j}$, where $\boldsymbol{P}_{i,j} = \text{Diag}(\underbrace{\alpha_1, \ldots, \alpha_1}_{N_t}, \underbrace{\alpha_2, \ldots, \alpha_2}_{N_t}, \ldots, \underbrace{\alpha_M, \ldots, \alpha_M}_{N_t})$, such that $\text{Tr}(\boldsymbol{\Gamma}_m \boldsymbol{V}_{i,j} \boldsymbol{P}_{i,j} \boldsymbol{P}_{i,j}^{\dagger} \boldsymbol{V}_{i,j}^{\dagger}) = P_m / K_0$. The diagonal matrix $\boldsymbol{\Gamma}_m \in \mathbb{R}^{MN_t \times MN_t}$ is introduced for each AP to select the partition of $\tilde{\boldsymbol{V}}_{i,j}$ 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. $\boldsymbol{V}_{i,j}$ can be obtained based on Section 5.1.

Thus the best user group with highest estimated data rate is selected, that is,

$$\mathcal{G}^* = \text{argmax}_{\mathcal{G}_l} \tilde{R}(\mathcal{G}_l) \ .$$

The achievable sum-rate of the finally selected user group can be maximized via optimal power allocation, which will be discussed in Section 3.4.

3.3.5 Fast update

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.

3.3.6 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 [62] to accommodate various fairness criteria. The algorithm of [62] 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.

3.4 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 $\boldsymbol{V}_{\pi_k}$. Let $\bar{\boldsymbol{V}}_{\pi_k} = \tilde{\boldsymbol{S}}_{\pi_k}\tilde{\boldsymbol{B}}_{\pi_k}\boldsymbol{V}_{\pi_k}$. The achievable sum rate with optimal power allocation is given by,

$$R_{BD}(\mathcal{G}) = \max_{\substack{\mathcal{G}\subset\mathcal{K}, \boldsymbol{Q}_{\pi_k}\succeq\boldsymbol{0} \\ \sum_{\pi_k\in\mathcal{G}} \mathrm{Tr}(\boldsymbol{\Gamma}_m\boldsymbol{V}_{\pi_k}\boldsymbol{Q}_{\pi_k}\boldsymbol{V}_{\pi_k}^\dagger)\le P_m}} \sum_{\pi_k\in\mathcal{G}} \log\left|\boldsymbol{I} + \bar{\boldsymbol{V}}_{\pi_k}\boldsymbol{Q}_{\pi_k}\bar{\boldsymbol{V}}_{\pi_k}^\dagger/\sigma_{\pi_k}^2\right| , \tag{3.6}$$

where $\boldsymbol{E}(\boldsymbol{x}_{i,j}\boldsymbol{x}_{i,j}^\dagger) = \boldsymbol{Q}_{i,j}$ is its transmit covariance matrix. The problem (3.6) 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 (3.6) for user set $\mathcal{G}$ can be solved via Lagrange duality method. The Lagrange function of the right-hand side of (3.6) is given by

$$\begin{aligned} L(\{\boldsymbol{Q}_{\pi_k}\}_{\pi_k\in\mathcal{G}}, \boldsymbol{\mu}) = &- \sum_{\pi_k\in\mathcal{G}} \log\left|\boldsymbol{I} + \frac{\bar{\boldsymbol{V}}_{\pi_k}\boldsymbol{Q}_{\pi_k}\bar{\boldsymbol{V}}_{\pi_k}^\dagger}{\sigma_{\pi_k}^2}\right| \\ &+ \sum_{m=1}^{M} \mu_m\Big(\sum_{\pi_k\in\mathcal{G}} \mathrm{Tr}(\boldsymbol{\Gamma}_m\boldsymbol{V}_{\pi_k}\boldsymbol{Q}_{\pi_k}\boldsymbol{V}_{\pi_k}^\dagger) - P_m\Big) , \end{aligned} \tag{3.7}$$

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

given by

$$\max_{\boldsymbol{\mu}\geq 0} \quad q(\boldsymbol{\mu}) = \min_{\{\boldsymbol{Q}_{\pi_k}\succeq \mathbf{0}\}_{\pi_k\in\mathcal{G}}} L(\{\boldsymbol{Q}_{\pi_k}\}_{\pi_k\in\mathcal{G}}, \boldsymbol{\mu}) \tag{3.8}$$

Since the right-hand side of (3.6) is convex and satisfies the Slater's condition, the duality gap between the optimal objective of (3.6) and that of the dual problem (5.8) 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 (5.8) can be further decomposed into K_0 independent subproblems,

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

where

$$\begin{aligned}\tilde{\boldsymbol{A}}_{\pi_k} &= \boldsymbol{V}_{\pi_k}^{\dagger}(\sum_{m=1}^{M}\mu_m\boldsymbol{\Gamma}_m)\boldsymbol{V}_{\pi_k},\\ \tilde{\boldsymbol{Q}}_{\pi_k} &= \tilde{\boldsymbol{A}}_{\pi_k}^{1/2}\boldsymbol{Q}_{\pi_k}\tilde{\boldsymbol{A}}_{\pi_k}^{1/2}\ .\end{aligned}$$

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,

$$\bar{\boldsymbol{V}}_{\pi_k}\tilde{\boldsymbol{A}}_{\pi_k}^{-1/2} = \tilde{\boldsymbol{F}}_{\pi_k}\boldsymbol{\Theta}_{\pi_k}\tilde{\boldsymbol{G}}_{\pi_k}^{\dagger}\ , \tag{3.9}$$

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

$$\boldsymbol{Q}_{\pi_k}^{*} = \tilde{\boldsymbol{A}}_{\pi_k}^{-1/2}\tilde{\boldsymbol{Q}}_{\pi_k}^{*}\tilde{\boldsymbol{A}}_{\pi_k}^{-1/2}\ . \tag{3.10}$$

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 (3.7)-(4.15).

3.5 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, as shown in Figure 3.2. We set each AP to have 4 antenna elements and each

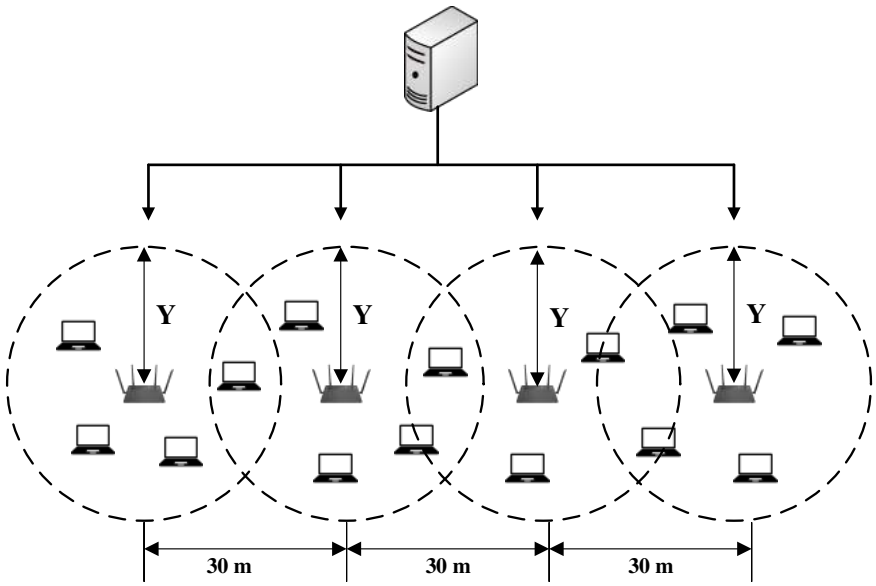

Figure 3.2: Network topology with 4 cooperative APs.

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 [63], which provides the upper bound of the sum-rate performance.
- Capacity-based user selection for BD (c-algorithm) [59], 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 [59].
- Upperbound-based user selection algorithm (u-algorithm) proposed in [61].

The greedy algorithms in [59] and [61] 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 [60] 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 [60] is also very similar to the n-algorithm.

3.5.1 Sum-rate performance

In Figure 3.3, 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 [63] 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

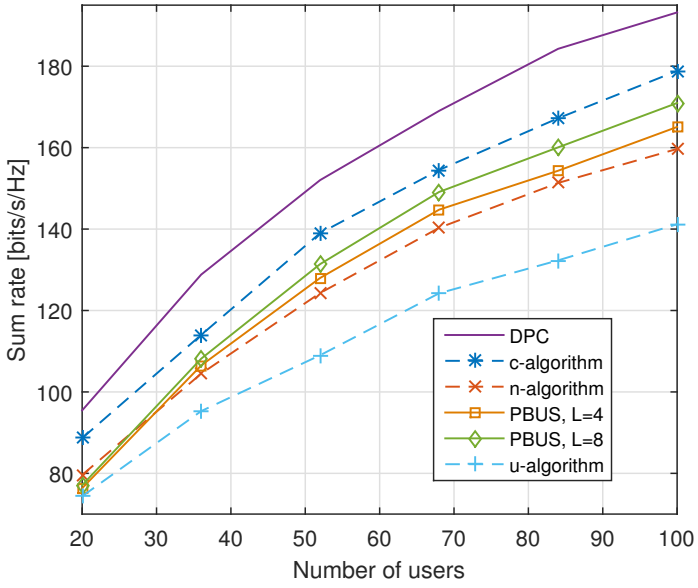

Figure 3.3: Sum-rate as a function of number of users at $Y = 30$.

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.

In Figure 3.4, 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 [59], 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.

3.5.2 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

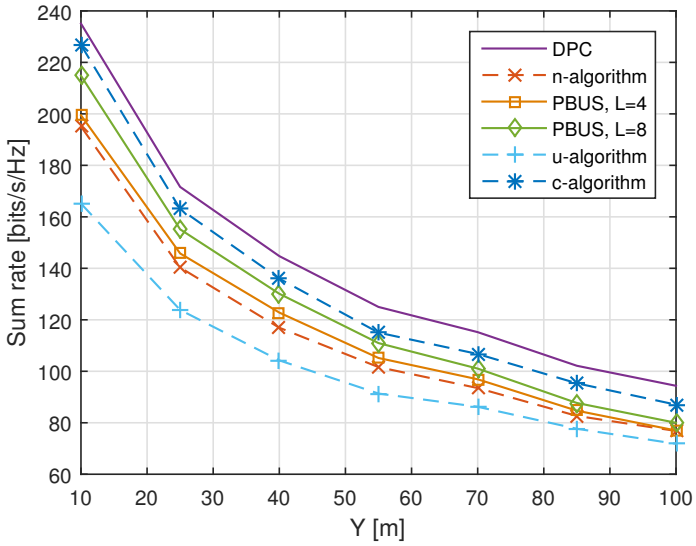

Figure 3.4: Sum-rate as a function of the radius Y with $K = 60$.

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 Figure 3.5, 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 Figure 3.5, 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

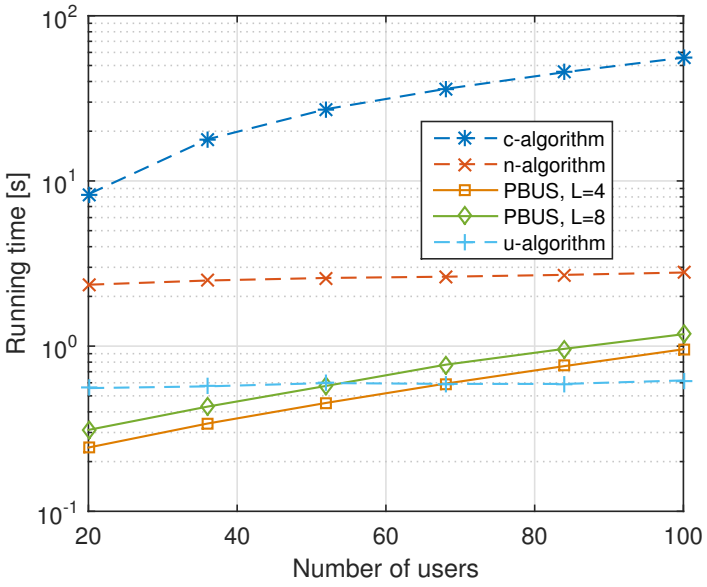

Figure 3.5: Running time as a function of number of users

in Figure 3.3. In addition, for a small to moderate number of users, our proposed algorithm can achieve higher sum-rate with even lower complexity as compared to the u-algorithm.

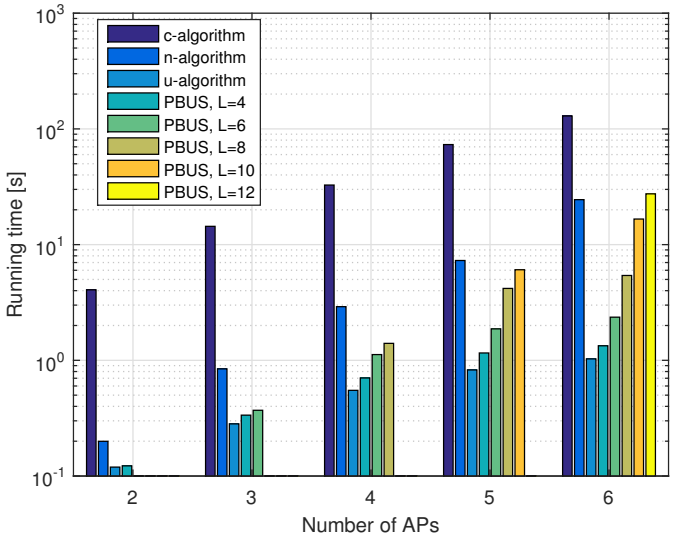

Figure 3.6: Running time as a function of the number of APs with 15 users for each AP

Moreover, we evaluate the running time of the proposed algorithm with different numbers of APs in Figure 3.6. The number of cooperative APs varies from 2 to 6 with 15 users around each AP. The computational complexity of c-algorithm increases from tens

of seconds to one hundred seconds as the number of APs grows from 2 to 6 as the cost of high sum-rate performance, while the proposed PBUS dramatically lowers the running time. Although the computational time of PBUS increases as more simultaneous users can be supported, the parameter L can be tuned to accommodate different requirements of computational efficiency without significantly sacrificing the sum-rate performance. For example, for 6 APs with 90 users, reducing the value of L from 12 to 8 can save 3/4 running time with only 7% loss of sum rate. With a small L, the proposed PBUS achieves similar sum-rate performance as n-algorithm, but consumes much less running time than n-algorithm. If the network is static, we can increase the value of L to obtain higher sum rate than n-algorithm. The computation time of PBUS is actually very similar to the u-algorithm for 20-30 users, while the u-algorithm is the fastest for large numbers of users. However, the sum-rate performance of the u-algorithm is substantially lower than that of the other algorithms.

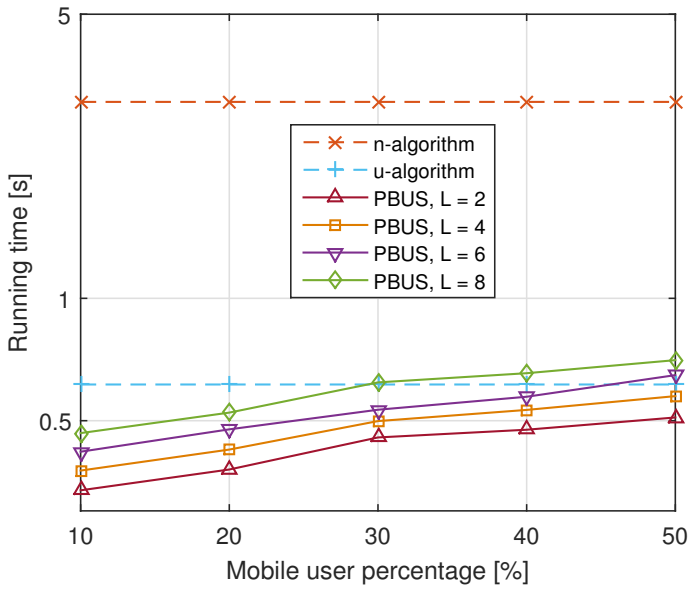

Figure 3.7: Update time as a function of mobile user percentage

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 Section 3.3.5, our proposed PBUS can reuse the calculated pairwise fitness metric and the constructed binary tree for static users as much as possible. In Figure 3.7, 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 Figure 3.5 even with a maximum value of $L = 8$, which is 25% lower than u-algorithm.

3.6 Chapter Summary

In this chapter, 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.

CHAPTER 4

COMBINED USER SELECTION AND MIMO WEIGHT CALCULATION

4.1 Introduction

In this chapter, we focus on optimizing the performance of a single cluster with cooperative APs under the dense wireless network settings. The performance of AP cooperation largely relies on the MIMO precoder and combiner design. Although the BD precoding with efficient user selection scheme has low complexity to achieve the multiuser MIMO communication, it does not fully exploit the potential performance gain promised by AP cooperation and therefore we consider more advanced processing techniques in this chapter. However, jointly optimizing the user selection and, precoding and combining weights is complicated by their inherent interdependence. In the limited mobility scenarios considered herein, most users are stationary for some period of time, which means channel conditions are only slowly time-varying. This allows more computationally expensive algorithms to be used in optimizing the signaling strategies of APs. The specific problem we consider is to maximize the downlink weighted sum rate (WSR) of a dense wireless network with cooperative processing and a per-AP power constraint. We also assume that the number of users is large, as the case of heavily-used dense wireless network deployments. As the number of users increases, the computation time for WSR maximization can quickly become impractical, even when channels are only slowly time-varying. Thus, we also develop a novel method that performs user selection as a pre-processing step to eliminate some users from consideration by the WSR maximization algorithm.

Different methods have been investigated in the literature for multicell networks, e.g. [64, 65, 59, 66, 67]. In [64], by exploiting the D.C. (difference of convex functions) structure of the sum-rate function, the convex-concave procedure is performed to find a local optimal

solution. This algorithm requires solving a max-det problem in each iteration, where the semi-definite programming algorithm is used. In [65], an enhanced block diagonalization (BD) precoding method is proposed to improve the sum rate performance. However, user selection is not considered and so the algorithm cannot handle a large number of users. In [68], the weighted sum rate (WSR) problem is address by the interference pricing based method. By maximizing the utility function of each user defined by the data rate and the interference cost, the algorithm reaches a stationary point of the WSR maximization problem. However, the algorithm allows one user to update its beamformer at one time, which may lead to excessive overhead of price exchange. In [59], two greedy user selection algorithms are proposed for BD precoding. However, the performance suffers due to the greedy user selection, which might not produce the best set of users to consider. In [66], two different approaches are considered to optimize the sum-rate performance, where dirty paper precoding (DPC) is assumed within each cell. When dropping the DPC constraint, the second approach becomes equivalent to the solution in [67]. Unfortunately, the computational complexity increases rapidly with the number of users for all methods proposed in [66, 67] making them unsuitable for the problem considered herein. To our knowledge, ours is the first scalable approach that considers a true sum rate maximization problem, i.e. it does not constrain solutions to use zero forcing, block diagonalization, or other approaches where performance is secondary to interference nullification.

To address the WSR maximization problem with cooperative APs, we propose a combined user selection and MIMO weights optimization approach, which determines the active user set and the MIMO precoders and combiners. Our approach has a very low computational complexity, even for a large number of users. A novel user selection algorithm that incorporates multiple decision factors is run as a pre-processing step to eliminate some users from consideration. Then, a modified WSR maximization algorithm optimizes the MIMO precoders and combiners. This modified WSR maximization algorithm can further eliminate users by allocating them zero power and it also determines the number of streams

for each active user. Simulation results demonstrate that, with 48 users and 3 APs, our approach increases aggregate performance by 25% compared to the best existing algorithm while running in 1/3 of the time.

4.2 System Model and Problem Description

We consider a MIMO network with M cooperative access points (APs), where the m^{th} AP is equipped with $N_{t,m}$ antennas. We assume that there are K users with $N_{r,k}$ antennas for the k^{th} user. Let $N_t = \sum_{m=1}^{M} N_{t,m}$ and $N_r = \sum_{k=1}^{K} N_{r,k}$ 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 $\boldsymbol{H}_k \in \mathbb{C}^{N_{r,k} \times N_t}$. The data vector $\boldsymbol{x} = \left[\boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_K^T\right]^T$ is jointly precoded by the M APs using the precoding matrix $\boldsymbol{V} = [\boldsymbol{V}_1, \ldots, \boldsymbol{V}_K]$. $\boldsymbol{x}_k \in \mathbb{C}^{N_{r,k}}$ is the transmit signal vector for receiver k, and $\boldsymbol{x}_k$ is assumed to be independently encoded Gaussian codebook symbols with $\mathbb{E}[\boldsymbol{x}_k \boldsymbol{x}_k^\dagger] = \boldsymbol{I}$, where $(\cdot)^\dagger$ is the conjugate transpose of $(\cdot)$. It is assumed that the k^{th} user has $N_{r,k}$ parallel data streams, although some of the streams can have a rate of zero. $\boldsymbol{V}_k \in \mathbb{C}^{N_t \times N_{r,k}}$ is the partition of $\boldsymbol{V}$ applied at the APs to precode the signals of user k. Assume the linear combiner $\boldsymbol{U}_k \in \mathbb{C}^{N_{r,k} \times N_{r,k}}$ is used at the k^{th} receiver.

The received vector of user k is given by

$$\begin{aligned}
\hat{\boldsymbol{x}}_k &= \boldsymbol{U}_k^\dagger \boldsymbol{H}_k \boldsymbol{V}_k \boldsymbol{x}_k + \boldsymbol{U}_k^\dagger \sum_{l=1, l\neq k}^{K} \boldsymbol{H}_k \boldsymbol{V}_l \boldsymbol{x}_l + \boldsymbol{U}_k^\dagger \boldsymbol{n}_k \\
&= \boldsymbol{U}_k^\dagger \boldsymbol{A}_k \boldsymbol{S}_k \boldsymbol{B}_k^\dagger \boldsymbol{V}_k \boldsymbol{x}_k + \boldsymbol{U}_k^\dagger \sum_{l=1, l\neq k}^{K} \boldsymbol{A}_k \boldsymbol{S}_k \boldsymbol{B}_k^\dagger \boldsymbol{V}_l \boldsymbol{x}_l + \boldsymbol{U}_k^\dagger \boldsymbol{n}_k \\
&= \tilde{\boldsymbol{U}}_k^\dagger \tilde{\boldsymbol{H}}_k \boldsymbol{V}_k \boldsymbol{x}_k + \tilde{\boldsymbol{U}}_k^\dagger \tilde{\boldsymbol{H}}_k \sum_{l=1, l\neq k}^{K} \boldsymbol{V}_l \boldsymbol{x}_l + \tilde{\boldsymbol{U}}_k^\dagger \boldsymbol{A}_k^\dagger \boldsymbol{n}_k \ ,
\end{aligned} \tag{4.1}$$

where $\boldsymbol{n}_k$ is the vector of Gaussian noise at the k^{th} user with covariance matrix $\sigma_k \boldsymbol{I} \in \mathbb{C}^{N_{r,k} \times 1}$. Recall the SVD of $\boldsymbol{H}_k$, which yields $\boldsymbol{H}_k = \boldsymbol{A}_k \boldsymbol{S}_k \boldsymbol{B}_k^\dagger$, where the quantized singular values in diagonal matrix $\boldsymbol{S}_k \in \mathbb{C}^{N_{r,k} \times N_{r,k}}$ and the right singular matrix $\boldsymbol{B}_k \in \mathbb{C}^{N_t \times N_{r,k}}$ are set back to the AP side based on the CSI feedback mechanism. We have $\tilde{\boldsymbol{U}}_k = \boldsymbol{A}_k^\dagger \boldsymbol{U}_k$

and $\tilde{\boldsymbol{H}}_k = \boldsymbol{S}_k \boldsymbol{B}_k^{\dagger}$. With limited CSI feedback, $\tilde{\boldsymbol{U}}_k$ can be deemed as a whole for optimizing the combiner. The quantized $\boldsymbol{S}_k$ and $\boldsymbol{B}_k$ are denoted by $\hat{\boldsymbol{S}}_k$ and $\hat{\boldsymbol{B}}_k$, which are available at the AP side. We have $\hat{\boldsymbol{H}}_k = \hat{\boldsymbol{S}}_k \hat{\boldsymbol{B}}_k^{\dagger}$.

The mean-square-error (MSE) covariance matrix of the k^{th} user evaluated at the AP side is given by

$$\begin{aligned} \boldsymbol{E}_k &= \mathbb{E}\left[(\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k)(\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k)^{\dagger}\right] \\ &= \tilde{\boldsymbol{U}}_k^{\dagger} \hat{\boldsymbol{H}}_k \boldsymbol{V}_k \boldsymbol{V}_k^{\dagger} \hat{\boldsymbol{H}}_k^{\dagger} \tilde{\boldsymbol{U}}_k + \tilde{\boldsymbol{U}}_k^{\dagger} \tilde{\boldsymbol{R}}_{\bar{k}} \tilde{\boldsymbol{U}}_k - \tilde{\boldsymbol{U}}_k^{\dagger} \hat{\boldsymbol{H}}_k \boldsymbol{V}_k - \boldsymbol{V}_k^{\dagger} \hat{\boldsymbol{H}}_k^{\dagger} \tilde{\boldsymbol{U}}_k + \boldsymbol{I} \end{aligned} \tag{4.2}$$

where $\tilde{\boldsymbol{R}}_{\bar{k}} = \sum_{l=1, l\neq k}^{K} \hat{\boldsymbol{H}}_k \boldsymbol{V}_l \boldsymbol{V}_l^{\dagger} \hat{\boldsymbol{H}}_k^{\dagger} + \sigma_k^2 \boldsymbol{I}$. The data rate of the k^{th} user can be rewritten as

$$\hat{R}_k = \log \left| \boldsymbol{I} + (\tilde{\boldsymbol{U}}_k^{\dagger} \tilde{\boldsymbol{R}}_{\bar{k}} \tilde{\boldsymbol{U}}_k)^{-1} (\tilde{\boldsymbol{U}}_k^{\dagger} \hat{\boldsymbol{H}}_k \boldsymbol{V}_k \boldsymbol{V}_k^{\dagger} \hat{\boldsymbol{H}}_k^{\dagger} \tilde{\boldsymbol{U}}_k) \right| . \tag{4.3}$$

Most systems simply sum throughput over all users with equal weighting, but this can result in favoring high-rate connections with good channel qualities at the expense of lower-rate clients, which may be undesirable, especially when quality of service (QoS) is considered as a performance metric.

Our goal is to maximize the weighted downlink sum-rate, which is useful for prioritizing different users and covers different practical applications. For instance, when identical weights are applied for all receivers, the problem becomes sum-rate maximization corresponding to a best effort service. Weighted sum rate maximization can also form the basis for higher-level scheduling algorithms that generate fair schedules with high throughput [69]. Since, in our problem setting, the transmitters are distinct APs that are at different physical locations, the transmit power of each AP should be bounded, which translates into a per-AP power constraint in the WSR maximization problem. This problem can be written

as

$$\begin{aligned} \max_{\substack{\{\boldsymbol{V}_k\}_{k\in\mathcal{U}} \\ \{\tilde{\boldsymbol{U}}_k\}_{k\in\mathcal{U}}}} \quad & \sum_{k\in\mathcal{U}} \omega_k \hat{R}_k \\ s.t. \quad & \sum_{k\in\mathcal{U}} Tr(\boldsymbol{\Gamma}_m \boldsymbol{V}_k \boldsymbol{V}_k^{\dagger}) \leq P_m, m = 1, \ldots, M \ , \end{aligned} \tag{4.4}$$

where the user set is denoted by $\mathcal{U} = \{1, \ldots, K\}$ and 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 AP m. Thus, $\boldsymbol{\Gamma}_m$ contains ones on the diagonal elements corresponding to the antennas of AP m and zeros elsewhere. ω_k is the weight for the k^{th} user and P_m is the maximum transmit power of AP m. Note that, instead of optimizing the combiner $\boldsymbol{U}_k$'s, we can optimize the $\tilde{\boldsymbol{U}}_k$'s with the limited CSI feedback. To facilitate the following analysis, the variable $\tilde{\boldsymbol{U}}_k$ is called composed combiner.

4.3 Combined User Selection and MIMO Weights Calculation

In our targeted high-density environment, the number of users is relatively large, meaning that only a subset of users can be served simultaneously in one time slot. The algorithm proposed in [70] for interfering MIMO channels, which can jointly optimize the user selection (i.e., allocating zero power to deactivate users) and MIMO weights, could be extended to solve the formulated WSR maximization problem with the added per-AP power constraint. However, the computational complexity of this modified algorithm increases rapidly as the number of users increases. Other algorithms address the problem by completely separating the user selection and precoder design [59, 60]. In those greedy incremental user selection algorithms, however, a previously selected user might become redundant when new users are added and this limits the performance of the final solution.

To overcome the problems in existing work, we propose a combined user selection and MIMO weights optimization algorithm. First, a fast pre-user selection procedure is performed to approximate a "good" subset of users by selecting K_0 users out of K users based on the performance metric, i.e. maximizing the potential WSR. Then, the MIMO

1. Let $\mathcal{U}_r = \{1, \ldots, K\}$ and $\mathcal{U}_s = \emptyset$
   $k^* = \text{argmax}_{k \in \mathcal{U}_r} w_k \log \left| \boldsymbol{I} + p \bar{\boldsymbol{H}}_k \bar{\boldsymbol{H}}_k^H \right|$
2. Update $\mathcal{U}_s = \{k^*\}$ and $\mathcal{U}_r = \mathcal{U}_r - \{k^*\}$
3. *If* $(|U_s| < K_0)$
4. Calculate the priority metric $f(\hat{\boldsymbol{H}}_k, \boldsymbol{H}_{sel})$ for $\forall k \in \mathcal{U}_r$ using (4.5).
5. $k^* = \text{argmax}_{k \in \mathcal{U}_r} \ f(\hat{\boldsymbol{H}}_k, \boldsymbol{H}_{sel})$
6. Quit if $f(\hat{\boldsymbol{H}}_{k^*}, \boldsymbol{H}_{sel}) < 0$; Else go to step 2
7. *Else* Quit

Figure 4.1: Pre-user selection pseudocode

precoders and combiners of the selected users are optimized, where the proposed algorithm can further refine the user selection and stream allocation by removing redundant users and deactivating streams with zero power, if necessary to improve the final WSR.

4.3.1 Pre-user selection

The objective of the user selection procedure is to select $K_0 < K$ users, that will potentially contribute to high-WSR performance. With a targeted K_0, it is costly to enumerate and evaluate $\binom{K}{K_0}$ 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 Figure 4.1.

The algorithm starts by selecting the user with highest interference-free weighted data rate. Equal power allocation is assumed during pre-user selection stage. With the CSI at transmitter side, let $\boldsymbol{Q}_k$ be the row basis of $\hat{\boldsymbol{H}}_k$. 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:

$$\begin{aligned} f(\hat{\boldsymbol{H}}_k, \boldsymbol{H}_{sel}) = \; & w_k \log_2(1 + \frac{p}{|\mathcal{U}_s|+1}||\boldsymbol{H}_{e,k}||_F^2) \\ & + \sum_{i \in \mathcal{U}_s} w_i \log_2(1 + \frac{p}{|\mathcal{U}_s|+1}||\bar{\boldsymbol{H}}_i \boldsymbol{H}_{e,k}^{\perp}||_F^2) \\ & - \sum_{i \in \mathcal{U}_s} w_i \log_2(1 + \frac{P_t}{N_r|U_s|}||\bar{\boldsymbol{H}}_i||_F^2), \end{aligned} \tag{4.5}$$

where $\boldsymbol{H}_{sel} = \left[\hat{\boldsymbol{H}}_i\right]_{i \in \mathcal{U}_s}$, $\boldsymbol{H}_{e,k} = \hat{\boldsymbol{H}}_k \times null(\boldsymbol{H}_{sel})$ and $\boldsymbol{H}_{e,k}^{\perp} = \boldsymbol{I} - \boldsymbol{Q}_k \boldsymbol{Q}_k^{\dagger}$. $||\cdot||_F$ denotes the Frobenius norm. In addition, $p = \sum_{m=1}^{M} P_m/N_r/\sigma^2$. The first term in (4.5) 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.

4.3.2 MIMO precoder and combiner calculation

Once the pre-user selection is complete, the precoders and combiners of selected users are determined by solving the WSR maximization problem for the remaining users. The targeted WSR maximization problem in (4.4) needs to be modified by replacing the user

set $\mathcal{U}$ with $\mathcal{U}_s$. However, the WSR maximization problem is a non-convex problem, which is difficult to solve based on the Karush-Kuhn-Tucker (KKT) conditions for the formulated problem.

Therefore, we consider a more tractable approach to solve the problem. Consider the weighted MSE minimization problem as follows,

$$\begin{aligned} \min_{\substack{\{\boldsymbol{V}_k\}_{k\in\mathcal{U}_s} \\ \{\tilde{\boldsymbol{U}}_k\}_{k\in\mathcal{U}_s}}} \quad & \sum_{k\in\mathcal{U}_s} Tr(\boldsymbol{W}_k\boldsymbol{E}_k) \\ s.t. \quad & \sum_{k\in\mathcal{U}_s} Tr(\boldsymbol{\Gamma}_m\boldsymbol{V}_k\boldsymbol{V}_k^\dagger) \leq P_m, m=1,\ldots,M. \end{aligned} \tag{4.6}$$

Based on [67], it can be proved that the gradient of WSR maximization with respect to $\boldsymbol{V}_k$ and the gradient of weighted sum MSE minimization with respect to $\boldsymbol{V}_k$ are identical if

$$\tilde{\boldsymbol{U}}_k = (\hat{\boldsymbol{H}}_k\boldsymbol{V}_k\boldsymbol{V}_k^\dagger\hat{\boldsymbol{H}}_k^\dagger + \tilde{\boldsymbol{R}}_{\bar{k}})^{-1}\hat{\boldsymbol{H}}_k\boldsymbol{V}_k \tag{4.7}$$

and the MSE weights satisfy

$$\boldsymbol{W}_k = \omega_k(\boldsymbol{I} + \boldsymbol{T}_k^\dagger\boldsymbol{V}_k^\dagger\hat{\boldsymbol{H}}_k^\dagger\tilde{\mathbf{R}}_{\bar{k}}^{-1}\hat{\boldsymbol{H}}_k\boldsymbol{V}_k\boldsymbol{T}_k), \tag{4.8}$$

where $\boldsymbol{T}_k$ is an arbitrary unitary matrix. The $\tilde{\boldsymbol{R}}_{\bar{k}}$ is updated by letting $\boldsymbol{V}_i = \mathbf{0}$ for $i \notin \mathcal{U}_s$. Besides, $\tilde{\boldsymbol{U}}_k$ in MMSEWeights achieves the optimal data rate with a given $\boldsymbol{V}_k$.

Based on the equivalence relation between WSR maximization and weighted sum MSE minimization, an iterative algorithm can be derived to find a local WSR-optimum, as summarized in Figure 4.2. The algorithm alternatively updates the precoders $\boldsymbol{V}_k$'s, MSE weights $\boldsymbol{W}_k$'s and composed combiner $\tilde{\boldsymbol{U}}_k$'s, which solves a weighted sum MSE minimization problem in each iteration. As analyzed in [67], the algorithm will converge to a local WSR-optimum.

While the weighted sum MSE minimization problem can be solved by extending the algorithms in [67] and [71] to the formulated problem with per-AP power constraint, the

1. Initialization: $\boldsymbol{V}_k = \boldsymbol{V}_k^0$ for all $k \in \mathcal{U}_s$;
2. Repeat
3. Compute $\tilde{\boldsymbol{U}}_k$ using (4.7) for all $k \in \mathcal{U}_s$ for given $\boldsymbol{V}_k$'s;
4. Compute $\boldsymbol{W}_k$ using (4.8) for given $\boldsymbol{V}_k$'s and $\tilde{\boldsymbol{U}}_k$'s;
5. Update $\boldsymbol{V}_k$'s by solving problem (4.6);
6. until $\left| \sum_{k \in \mathcal{U}_s} \omega_k \hat{R}_k^{(n)} - \sum_{k \in \mathcal{U}_s} \omega_k \hat{R}_k^{(n-1)} \right| \leq \epsilon$.

Figure 4.2: Alternating optimization for WSR maximization

solutions provided by these iterative algorithms lack the ability to quickly deactivate links. Specifically, the solutions in [67] and [71] cannot decouple the precoder of the k^{th} user and its combiner, which can only gradually reduce the power allocated to some links as the algorithm iterates, eventually deactivating links but only after a sufficiently large number of iterations.

In order to further refine the selected users and determine the active streams of each user efficiently, we propose a different algorithm to solve the weighted sum MSE minimization problem. First, the Lagrangian of the weighted sum MSE minimization problem is given by

$$L = \sum_{k \in \mathcal{U}_s} Tr(\boldsymbol{W}_k \boldsymbol{E}_k) + \sum_{m=1}^{M} \mu_m (Tr(\boldsymbol{\Gamma}_m \boldsymbol{V}\boldsymbol{V}^\dagger) - P_m), \tag{4.9}$$

where $\mu_m \geq 0$ for $m = 1, \ldots, M$ are the Lagrange multipliers. The dual problem is given by

$$\max_{\boldsymbol{\mu}} \quad q(\boldsymbol{\mu}) \;\; s.t. \mu_m \geq 0, \text{for } m = 1, \ldots, M, \tag{4.10}$$

where $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_M)^T$ and $q(\boldsymbol{\mu}) = \min_{\boldsymbol{V}} \; L(\boldsymbol{V}, \boldsymbol{\mu})$ is the Lagrangian dual function. The dual problem can be solved iteratively, where in each iteration the precoder matrix $\boldsymbol{V}$ can be solved by using the KKT conditions for a fixed set of Lagrange multipliers, and the master problem is solved to find the Lagrange multipliers.

Based on the KKT conditions, the gradient of L with respect to $\boldsymbol{V}_k^\dagger$ should be zero,

which yields the following equation

$$\begin{aligned}\hat{\boldsymbol{H}}_k^{\dagger}\tilde{\boldsymbol{U}}_k\boldsymbol{W}_k &= \hat{\boldsymbol{H}}_k^{\dagger}\tilde{\boldsymbol{U}}_k\boldsymbol{W}_k\tilde{\boldsymbol{U}}_k^{\dagger}\hat{\boldsymbol{H}}_k\boldsymbol{V}_k + \\ &\sum_{l\in\mathcal{U}_s, l\neq k}\hat{\boldsymbol{H}}_l^{\dagger}\tilde{\boldsymbol{U}}_l\boldsymbol{W}_l\tilde{\boldsymbol{U}}_l^{\dagger}\hat{\boldsymbol{H}}_l\boldsymbol{V}_k + \sum_{m=1}^{M}\mu_m\boldsymbol{\Gamma}_m\boldsymbol{V}_k.\end{aligned} \tag{4.11}$$

The above equation can only provide the precoder $\boldsymbol{V}_k$ as a function of its combiner $\boldsymbol{U}_k$. To overcome this problem, we solve the precoder using (4.11) and (4.7). Let us first perform the following compact singular value decomposition (SVD), we introduce another set of equations by rewriting (4.7) into

$$\hat{\boldsymbol{H}}_k\boldsymbol{V}_k = \hat{\boldsymbol{H}}_k\boldsymbol{V}_k\boldsymbol{V}_k^{\dagger}\hat{\boldsymbol{H}}_k\tilde{\boldsymbol{U}}_k + \tilde{\boldsymbol{R}}_{\bar{k}}\tilde{\boldsymbol{U}}_k. \tag{4.12}$$

To solve the precoder in (4.11), let us first perform the following compact SVD,

$$\tilde{\boldsymbol{R}}_{\bar{k}}^{-1/2}\hat{\boldsymbol{H}}_k\boldsymbol{\Pi}_{\bar{k}}^{-1/2} = \boldsymbol{F}_k\boldsymbol{D}_k\boldsymbol{G}_k^{\dagger}, \tag{4.13}$$

where

$$\boldsymbol{\Pi}_{\bar{k}} = \sum_{l=\in\mathcal{U}_s, l\neq k}\hat{\boldsymbol{H}}_l^{\dagger}\tilde{\boldsymbol{U}}_l\boldsymbol{W}_l\tilde{\boldsymbol{U}}_l^{\dagger}\hat{\boldsymbol{H}}_l + \sum_{m=1}^{M}\mu_m\boldsymbol{\Gamma}_m; \tag{4.14}$$

and $\boldsymbol{D}_k \in \mathbb{R}^{N_{r,k}\times N_{r,k}}$ is a diagonal matrix containing the singular values of $\tilde{\boldsymbol{R}}_{\bar{k}}^{-1/2}\hat{\boldsymbol{H}}_k\boldsymbol{\Pi}_{\bar{k}}^{-1/2}$ ordered in decreasing order; $\boldsymbol{F}_k \in \mathbb{C}^{N_{r,k}\times N_{r,k}}$ and $\boldsymbol{G}_k \in \mathbb{C}^{N_t\times N_{r,k}}$ are the corresponding left and right singular vectors of $\tilde{\boldsymbol{R}}_{\bar{k}}^{-1/2}\hat{\boldsymbol{H}}_k\boldsymbol{\Pi}_{\bar{k}}^{-1/2}$.

Based on [72] and [70], for given μ_m's, the precoding matrix for receiver k that solves (4.11) and (4.12) is given by

$$\boldsymbol{V}_k = \boldsymbol{\Pi}_{\bar{k}}^{-1/2}\boldsymbol{G}_k\boldsymbol{\Psi}_k, \tag{4.15}$$

$$\tilde{\boldsymbol{U}}_k = \tilde{\boldsymbol{R}}_{\bar{k}}^{-1/2}\boldsymbol{F}_k\boldsymbol{\Phi}_k, \tag{4.16}$$

Theorem 1. *The matrices $\boldsymbol{\Psi}_k$ and $\boldsymbol{\Phi}_k$ is given by*

$$\begin{aligned}\boldsymbol{\Psi}_k &= \left(\boldsymbol{W}_k^{1/2}\boldsymbol{D}_k^{-1} - \boldsymbol{D}_k^{-2}\right)_+^{1/2}, \\ \boldsymbol{\Phi}_k &= \boldsymbol{W}_k^{-1/2}\boldsymbol{\Psi}_k\ ,\end{aligned} \tag{4.17}$$

where $(\cdot)_+$ 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.

Proof. Premultiplying (4.11) and (4.12) with $\boldsymbol{V}_k^\dagger$ and $\tilde{\boldsymbol{U}}_k^\dagger$, respectively, and using the expressions in (4.15), (4.16) and (4.13), we obtain the following equations,

$$\boldsymbol{\Psi}_k^\dagger \boldsymbol{D}_k \boldsymbol{\Phi}_k \boldsymbol{W}_k = \boldsymbol{\Psi}_k^\dagger \boldsymbol{D}_k \boldsymbol{\Phi}_k \boldsymbol{W}_k \boldsymbol{\Phi}_k^\dagger \boldsymbol{D}_k \boldsymbol{\Psi}_k + \boldsymbol{\Psi}_k^\dagger \boldsymbol{\Psi}_k, \tag{4.18}$$

$$\boldsymbol{\Phi}_k^\dagger \boldsymbol{D}_k \boldsymbol{\Psi}_k = \boldsymbol{\Phi}_k^\dagger \boldsymbol{D}_k \boldsymbol{\Psi}_k \boldsymbol{\Psi}_k^\dagger \boldsymbol{D}_k \boldsymbol{\Phi}_k + \boldsymbol{\Phi}_k^\dagger \boldsymbol{\Phi}_k. \tag{4.19}$$

Let

$$\begin{aligned}\boldsymbol{\Theta_1} &= \boldsymbol{\Psi}_k^\dagger \boldsymbol{D}_k \boldsymbol{\Phi}_k \\ \boldsymbol{\Theta_2} &= \boldsymbol{\Psi}_k^\dagger \boldsymbol{\Psi}_k \\ \boldsymbol{\Theta_3} &= \boldsymbol{\Phi}_k^\dagger \boldsymbol{\Phi}_k.\end{aligned} \tag{4.20}$$

Based on the fact that $\boldsymbol{\Theta_1}$, $\boldsymbol{\Theta_2}$ and $\boldsymbol{\Theta_3}$ are real-valued diagonal matrices [72], we have

$$\begin{aligned}\boldsymbol{\Theta_1}\boldsymbol{W}_k &= \boldsymbol{\Theta_1}\boldsymbol{W}_k\boldsymbol{\Theta_1} + \boldsymbol{\Theta_2} \\ \boldsymbol{\Theta_1} &= \boldsymbol{\Theta_1^2} + \boldsymbol{\Theta_3} \\ \boldsymbol{\Theta_1^2} &= \boldsymbol{\Theta_2}\boldsymbol{D}_k^2\boldsymbol{\Theta_3}.\end{aligned} \tag{4.21}$$

Solving the equations above, we obtain

$$\boldsymbol{\Theta}_1 = (\boldsymbol{I} - \boldsymbol{D}_k^{-1}\boldsymbol{W}_k^{-1/2})_+. \tag{4.22}$$

Then $\mathbf{\Theta}_2$ and $\mathbf{\Theta}_3$ can be solved by substituting (4.22) to (4.21), which will give the results in (4.17). □

Note that if the precoder is given by (4.15), the $\tilde{\boldsymbol{U}}_k$'s in (4.16) is equivalent to the composed combiner given in (4.7). Different from the solution in [67] and [71], the power allocated to each stream for the k^{th} user given by (4.15) is determined by the received interference plus noise, the interference to other receivers, and the available power. It implies that the decision on whether to activate or deactivate the streams is based on the current state of the network, instead of on whether the stream was active or inactive previously.

To find the Lagrange multiplier μ_m, the ellipsoid method or sub-gradient method can be used. The solution given by the proposed WSR maximization algorithm can set the active streams with a small number of iterations. Therefore, it can quickly remove redundant users with inactive precoders and eliminate their effects on other users.

4.3.3 Joint algorithm for WSR maximization

Due to the properties of the proposed WSR maximization algorithm, it can also be implemented without the pre-user selection procedure. In this case, the user selection is jointly determined with the MIMO weights, where a user is active when its precoder is active with non-zero power.

Especially when the number of users is relatively small, the proposed WSR maximization algorithm can be performed to solve (4.4) directly, which will not generate much higher computational complexity than the combined algorithm.

4.4 Algorithm Implementation

Prior to the calculation of precoders, the central controller needs to collect the up-to-dated CSI from the cooperative APs. With the compressed CSI feedback mechanism discussed in Section 2.3.4, the precoding matrices $\boldsymbol{V}_k$'s and the composed combiner $\tilde{\boldsymbol{U}}_k$'s are jointly

optimized by the central controller as discussed in Chapter 4.3.2. The combiner for the k^{th} receiver is given by $\boldsymbol{U}_k = \boldsymbol{A}_k \tilde{\boldsymbol{U}}_k$.

Theorem 2. *The solution to $\tilde{\boldsymbol{U}}_k$ given by (4.7) is equivalent to the MMSE receiver with perfect CSI, while ignoring the quantization error.*

Proof. Since we have $\tilde{\boldsymbol{U}}_k = \boldsymbol{A}_k^\dagger \boldsymbol{U}$ and $\boldsymbol{A}_k \boldsymbol{A}_k^\dagger = \boldsymbol{I}$, we can rewrite it into

$$\begin{aligned} \boldsymbol{U}_k &= \boldsymbol{A}_k \tilde{\boldsymbol{U}}_k \\ &= \boldsymbol{A}_k (\hat{\boldsymbol{H}}_k \boldsymbol{V}_k \boldsymbol{V}_k^\dagger \hat{\boldsymbol{H}}_k^\dagger + \tilde{\boldsymbol{R}}_{\bar{k}})^{-1} \hat{\boldsymbol{H}}_k \boldsymbol{V}_k \\ &= \left(\boldsymbol{A}_k (\hat{\boldsymbol{H}}_k \boldsymbol{V}_k \boldsymbol{V}_k^\dagger \hat{\boldsymbol{H}}_k^\dagger + \tilde{\boldsymbol{R}}_{\bar{k}}) \boldsymbol{A}_k^\dagger \right)^{-1} \boldsymbol{A}_k \hat{\boldsymbol{H}}_k \boldsymbol{V}_k \\ &= \left(\boldsymbol{H}_k \boldsymbol{V}_k \boldsymbol{V}_k^\dagger \boldsymbol{H}_k^\dagger + \boldsymbol{R}_{\bar{k}} \right)^{-1} \boldsymbol{H}_k \boldsymbol{V}_k \ , \end{aligned} \tag{4.23}$$

if there is no quantization error, that is, $\hat{\boldsymbol{H}} = \bar{\boldsymbol{H}}$. $\boldsymbol{R}_{\bar{k}} = \sum_{l=1, l\neq k}^{K} \boldsymbol{H}_k \boldsymbol{V}_l \boldsymbol{V}_l^\dagger \boldsymbol{H}_k^\dagger + \sigma_k^2 \boldsymbol{I}$ is the covariance matrix of the interference plus noise. Therefore, the solution to $\tilde{\boldsymbol{U}}_k$ obtained from the proposed WSRM algorithm is equivalent to the MMSE receiver. □

The simplest way to implement the optimized MMSE combiner is to select a head AP and let it broadcast the $\tilde{\boldsymbol{U}}_k$'s and the corresponding client IDs to the receivers before data transmission. The receivers can extract its combiner from $\tilde{\boldsymbol{U}}_k$ as $\boldsymbol{U}_k = \boldsymbol{A}_k \tilde{\boldsymbol{U}}_k$. The overhead caused by distributing the calculated $\tilde{\boldsymbol{U}}_k$'s depends on the matrix compression method. For example, assuming the compressed Given's rotation method as used for CSI compression, the broadcasting overhead is jointly determined by the number of active users, the number of receive antennas and the quantization bits.

Alternatively, to avoid the quantization error of the combiner, the MMSE combiner can be calculated at the receiver side. Recall that the MMSE combiner is given by (4.23), which can be rewritten into

$$\begin{aligned} \boldsymbol{U}_k &= \left(\boldsymbol{H}_k \boldsymbol{V}_k \boldsymbol{V}_k^\dagger \boldsymbol{H}_k^\dagger + \boldsymbol{R}_{\bar{k}} \right)^{-1} \boldsymbol{H}_k \boldsymbol{V}_k \\ &= \boldsymbol{J}_k^{-1} \boldsymbol{H}_k \boldsymbol{V}_k \ , \end{aligned}$$

where $\boldsymbol{J}_k$ is the covariance matrix of the total received signal at the k^{th} receiver. Therefore, the receiver can calculate its own MMSE combiner if the effective channel $\boldsymbol{H}_k\boldsymbol{V}_k$ is known. This can be achieved by a second-stage channel sounding procedure with calculated precoders at the AP side.

Specifically, after finishing the joint optimization of $\boldsymbol{V}_k$'s and $\tilde{\boldsymbol{U}}_k$'s, the central controller initiates the second-stage channel sounding and enables the receivers to estimate its effective channel $\boldsymbol{H}_k\boldsymbol{V}_k$, which can be probed by precoding the known data sequences and sending the precoded sequences from the cooperative APs.

4.5 Simulation Results

In this section, simulation results that evaluate the performance of our proposed scheme are reported. For all simulations, we assume a quasi-static Rayleigh flat-fading channel, which is considered constant for the duration of a burst that appears randomly in time. The path-loss exponent is set to 3 and the noise power is -80 dBm. We consider that the APs are located in a line with an interval of X meters. The maximum transmit power for each AP is set to 23dBm. We uniformly distribute k_m users around the m^{th} AP within a radius of Y meters. The total number of users is given by $\sum_{m=1}^{M} k_m = K$. Unless otherwise specified, we consider 3 APs, each with four antennas, and two antennas for each user. The user weights are randomly generated within the range of $[0, 1]$.

4.5.1 Impact of compressed CSI feedback

First, we investigate the achievable WSR of the proposed algorithm with different CSI quantization bits. As we mentioned in Chapter 2.3.4, there are typically two types of quantization for the right singular value of the channel matrix. Type I uses 12 bits for each pair of angles (7 bits for ϕ and 5 bits for ψ) and Type II uses 16 bits for each pair of angles (9 bits for ϕ and 7 bits for ψ). The quantization bits determines the accuracy of the CSI available at the AP side and, thus, affect the performance of the optimized precoders. In Figure 4.3, the

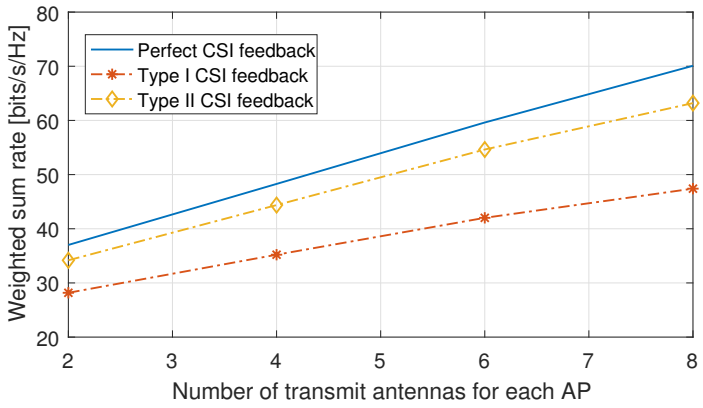

Figure 4.3: Achievable WSR as a function of the number of transmit antennas with different CSI accuracy.

achievable WSR is evaluated for different types of CSI feedback. The number of transmit antennas for each AP varies from 2 to 8, which leads to different dimensions of the channel matrix. The perfect CSI feedback achieves the highest WSR performance, which, however, is impractical. Another observation is that channel matrix with larger dimension is more sensitive to the CSI accuracy. As shown in Figure 4.3, the WSR performance loss gets larger for both type I and type II CSI feedback as the number of transmit antenna increases. More quantization bits used by type II CSI feedback contribute to about 30% higher WSR than type I CSI feedback, at the price of larger feedback overhead. Overall, the type II CSI feedback can achieve more than 90% WSR of the perfect CSI feedback, with up to 8 transmit antennas per AP.

4.5.2 WSR and computational complexity performance

Then, we study the WSR performance with our proposed algorithms. Both the proposed combined algorithm with orthogonality-based user selection and the joint algorithm without user selection are evaluated. We also compare our approaches to the WMMSE method [67], DPC [73], and the BD algorithm [59]. Since there is no existing work on the DPC with per-AP power constraint, we use the DPC algorithm with sum power constraint to serve as the upper bound of the actual DPC. Since most of the algorithms for comparison require

the perfect knowlege of CSI at the transmitter side, we ignore the quantization error of the CSI feedback for the WSR evaluation in this section.

Figure 4.4 shows the WSR and computation time of different algorithms as a function of the number of users with $X = 30$ m and $Y = 50\ m$. The upper bound is provided by the DPC scheme with sum power constraint. The c-algorithm for BD precoding scheme in [59] is extended to the WSR maximization problem. Comparing the combined algorithm with pre-user selection with $K_0 = 8$ to the joint algorithm, the performance loss due to the pre-user selection is less than 5%, while the computation time is significantly reduced and becomes almost independent of the number of users. Our proposed algorithm with pre-user selection achieves about 25% higher WSR than WMMSE and 40% higher than BD precoding scheme. Moreover, the c-algorithm-based BD precoding requires the highest computation time of all approaches, taking more time than even our proposed WSR maximization algorithm *without* user selection.

Figure 4.5 shows the WSR as a function of the circle radius Y. The number of users is 30 and the inter-AP spacing is 30 m. Smaller radius suggests that the users are more densely distributed around each AP, indicating high average received SNR. Thus, the WSR achieved by these algorithms increases as the radius decreases. The gap between the upper bound (DPC) and our proposed algorithm is less than 10%, and is caused by both the nonlinear precoding technique and the relaxed power constraint. WMMSE performs worse at low SNR values than with high SNR, while both of our proposed algorithms achieve about 25% higher WSR than that of WMMSE. When the radius is small enough, the BD algorithm performs close to our proposed algorithms, outperforming the WMMSE algorithm.

To demonstrate the performance gain from full cooperation, the WSRs at different levels of cooperation are illustrated in Figure 4.6 with $Y = 50$. Besides the considered full cooperation, we evaluate three other cases, namely interference coordination (IC), non-cooperation across APs, and orthogonal channels for APs. The WSR is plotted as a function of the AP separation in Figure 4.6, while the circle radius Y is fixed to 50 m. The

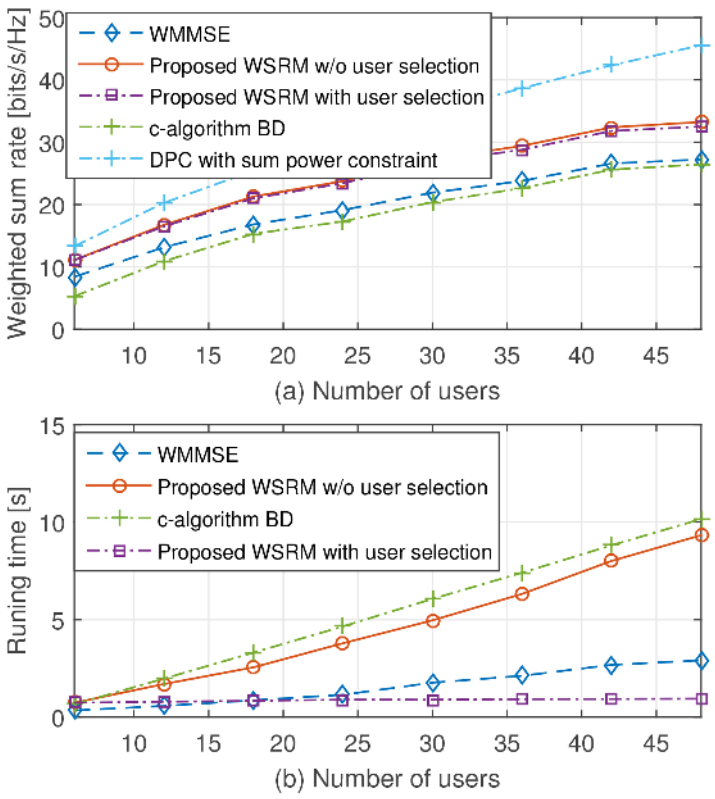

Figure 4.4: WSR and running time as a function of the number of users.

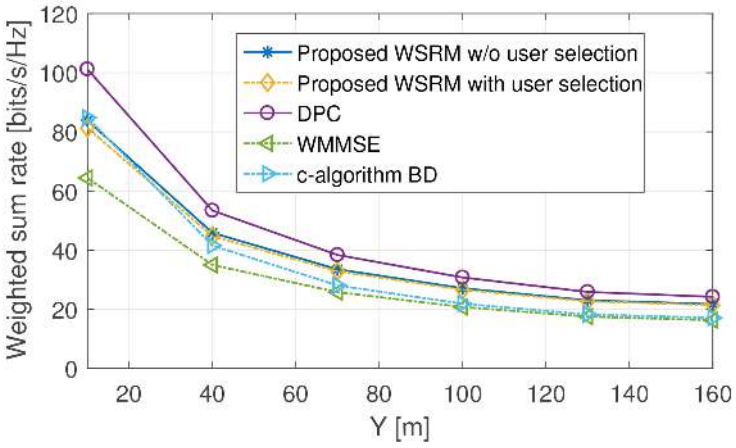

Figure 4.5: Achieved WSR as a function of the radius of user distribution.

aggregate performance of different levels of cooperation will converge to the same point with a sufficiently large X. More closely distributed APs result in higher WSR for the full cooperation case, while producing lower WSR for IC and non-cooperative cases. This is because decreasing the AP separation increases the interference, which negatively impacts the IC and non-cooperation solutions, while these interfering channels are turned into useful channels with full cooperation.

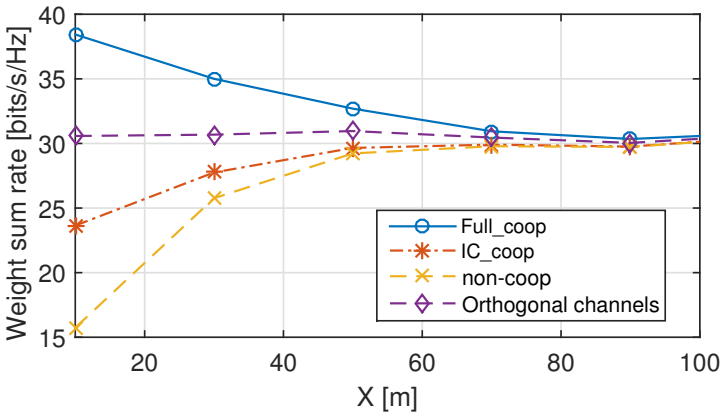

Figure 4.6: Achieved WSR of different multiuser MIMO networks.

We also evaluate the WSR as a function of the number of APs in Figure 4.7, where the number of users is fixed to 20. All APs and receivers are randomly located within a circle of radius 100 m. We also include the performance with non-overlapping time slot assignment for APs, so that only one AP can serve all users at a given time slot. As more APs are deployed in the area, more power is provided to improve the average received SINR, which leads to higher WSR for both full cooperation and IC, while assigning non-overlapping time slots to APs will not change the WSR. Note that, as the interference gets more severe, the performance gap between full cooperation and IC becomes greater because of full cooperation's ability to utilize the interfering channels across different APs. The results also show that wireless performance with full cooperation scales linearly with the number of APs increasing from 2 to 10.

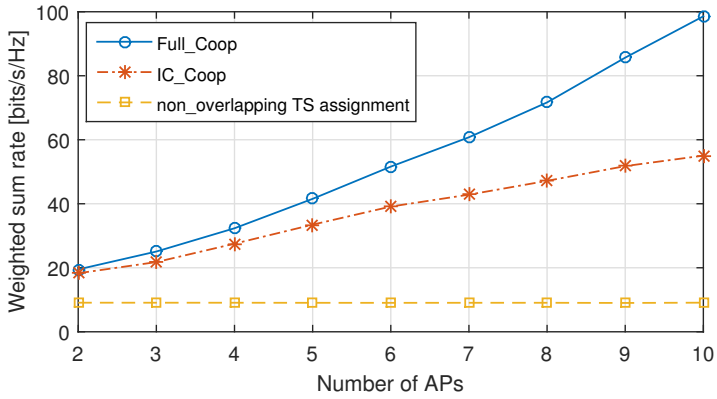

Figure 4.7: Achieved WSR as a function of the number of APs.

4.5.3 Convergence properties

We also investigate the convergence performance by comparing our proposed WSR maximization algorithm with the WMMSE algorithm in [67]. We set the same starting point for both algorithms and two random trials of experiments are performed for $K = 8$ and $K = 18$. Figure 4.8 shows the WSR as a function of the number of iterations. Although the convergence speed varies for different channel realizations, the results indicate that our proposed approach converges much faster than the WMMSE method with respect to the number of iterations. This occurs because the WMMSE method gradually reduces the power of some streams, requiring many iterations to deactivate streams, while our algorithm is able to completely deactivate some poor links in a single iteration. To validate the property of our proposed algorithm of deactivating streams efficiently, the number of active streams is plotted as a function of the number of iterations in Figure 4.8. Streams with non-zero power are deemed as active streams. For different number of users, our proposed WSRM algorithm quickly reduces the number of active streams to the supportable number and becomes stable in less than 10 iterations, while the WMMSE needs more than 1000 iterations to completely deactivate the unnecessary streams. These results validate that the proposed WSR maximization algorithm can eliminate redundant users and determine the active streams efficiently.

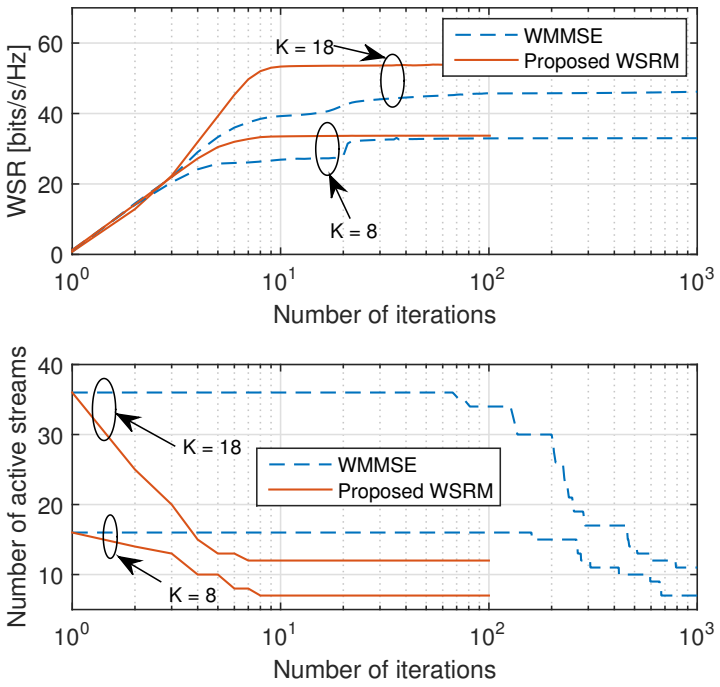

Figure 4.8: Convergence rate of the iterative WSRM algorithm

4.6 Chapter Summary

An approach to maximize performance in dense wireless networks with AP cooperation was presented and evaluated in this Chapter. The proposed algorithm is designed to maximize the weighted sum rate (WSR) with per-AP power constraint. The proposed approach was shown to outperform previous approaches to the problem while having significantly lower running time for a moderate to large number of users.

CHAPTER 5
HIGH THROUGHPUT AND FAIR SCHEDULING FOR MULTI-AP MULTIUSER MIMO

5.1 Introduction

In dense wireless networks, there are many users that cannot be accommodated in a single time slots. Therefore, we must extend the proposed solution in the previous chapter to multiple slots and consider how to satisfy the demands over all users. In this chapter, we study the fair scheduling problem for dense wireless networks with AP cooperation and MIMO links. We mainly focus on indoor wireless communications, where most users are expected to be stationary for significant periods of time with intermittent shorter periods of mobility. This scenario is consistent with most enterprise environments. Since there are many users sharing the limited resources of the wireless network, MIMO link scheduling arises as a key problem, i.e. determining how to activate MIMO links for a given scheduling period to meet the desired organizational requirements.

Prior work has considered the fairness issue either with multi-user MIMO with a single AP [38, 37], or with multiple APs but a single user per AP [74, 75, 48]. The works of [38, 37] primarily consider the problem of user selection to maximize sum rate but [38] 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 [74] and [75] consider how to associate users to APs to achieve fairness objectives. In [74], the association is done assuming that APs operate on different channels so interference between users is not a consideration while [75] accounts for interference in its evaluation. Finally, [48] shows how interference introduced by scheduling multiple users concurrently across APs distorts fairness and it proposes a

scheduling algorithm that achieves fairness comparable to the interference-free case. None of the above-cited works consider a scenario with both multi-user MIMO and multiple APs, as we address in this chapter.

Different from the single-user MIMO links, scheduling simultaneous transmission from APs to multiple users requires suppression of inter-user interference. For MIMO interference channels, interference alignment (IA) schemes are presented in [76, 77, 78, 42]. A capacity-optimal achievable IA scheme is proposed in [76] in a high SNR regime. However, IA is known to be a suboptimal strategy at lower SNRs [77, 78]. Three generations of IA are proposed in [42], including minimum interference leakage, joint mean square error minimization (MMSE) and maximum SNR algorithm. However, all of these works require a priori specification of which transmitters should transmit as well as how many streams each transmitter should transmit.

In general, throughput and fairness are two fundamental objectives in wireless networks that cannot be maximized simultaneously. This motivates the investigation of inherent tradeoffs between the two objectives, where a common approach is to maximize performance subject to some fairness constraints. We adopt the widely-used notion of time-based fairness [79][80][69], which avoids the performance anomaly associated with rate-based fairness in multi-rate wireless networks [81]. 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 [79].

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 spectrum frequency and employing multi-user MIMO. Our contributions are as follows:

1. we provide the first mathematical formulation of a maximum sum rate 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,

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 joint optimization algorithm produces significantly higher sum rate than all 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.

5.2 System Model and Problem Description

We consider a scenario in which a small number of APs forms a cluster and can cooperate with each other to serve the users. We expect that most users are stationary for significant periods of time with intermittent shorter periods of mobility. This is a common scenario for most enterprise WLAN settings, which typically covers office-type environments. 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. Scheduling downlink or uplink traffic only within single time slot helps reduce channel estimation overhead as shown in [69]. We include scenarios with both downlink and uplink traffic in our simulation results to evaluate the impact of our optimizations on overall network performance.

We assume that there is a single entity for one 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 users active within each slot, as shown in Figure 5.1. This entity, also referred as central controller, is connected to the APs within the cluster via high-speed wired links.

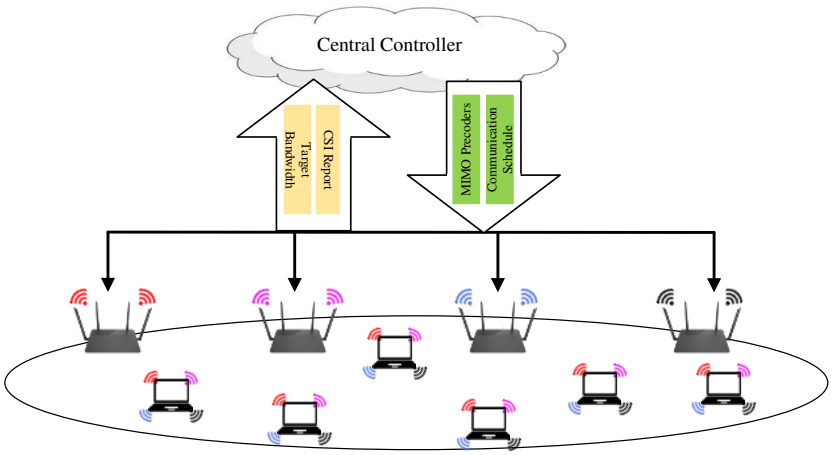

Figure 5.1: An example of the clustered overlapping APs.

Assume there are M cooperative access points (APs) in one cluster, where the m^{th} AP is equipped with $N_{t,m}$ antennas. We assume that there are K users with $N_{r,k}$ antennas for the k^{th} user. The user set is denoted by $\mathcal{K} = \{1, \ldots, K\}$. Let $N_t = \sum_{m=1}^{M} N_{t,m}$ and $N_r = \sum_{k=1}^{K} N_{r,k}$ 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 $\boldsymbol{H}_k \in \mathbb{C}^{N_{r,k} \times N_t}$, which is assumed to be time-invariant within a scheduling period. We assume that one scheduling period contains T time slots, each of which has the same duration. 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 precoding matrix $\boldsymbol{V}(t) = [\boldsymbol{V}_1(t), \ldots, \boldsymbol{V}_K(t)]$ for time slot t. $\boldsymbol{x}_k(t) \in \mathbb{C}^{N_{r,k}}$ 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 $N_{r,k}$ data streams, although some of the streams can have a rate of zero. $\boldsymbol{V}_k(t) \in \mathbb{C}^{N_t \times N_{r,k}}$ is the partition of $\boldsymbol{V}(t)$ applied at the APs to precode the signals of user k.

Moreover, we assume the modified explicit CSI feedback mechanism as discussed in Section 2.3.4. At the beginning of each scheduling period, the central controller initiates the channel sounding process and then the users send the compressed CSI to the AP side. For a certain scheduling period, with the SVD of the channel matrix $\boldsymbol{H}_k = \boldsymbol{A}_k \boldsymbol{S}_k \boldsymbol{B}_k^\dagger$, the quantized $\boldsymbol{S}_k$ and $\boldsymbol{B}_k$ are fed back to the APs, denoted by $\hat{\boldsymbol{S}}_{\boldsymbol{k}}$ and $\hat{\boldsymbol{B}}_k$. Details can be found in Section 2.3.4.

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

$$\boldsymbol{y}_k(t) = \boldsymbol{H}_k \boldsymbol{V}_k(t) \boldsymbol{x}_k(t) + \sum_{l=1, l \neq k}^{K} \boldsymbol{H}_k \boldsymbol{V}_l(t) \boldsymbol{x}_l(t) + \boldsymbol{n}_k \ , \tag{5.1}$$

where $\boldsymbol{n}_k$ is the vector of Gaussian noise at the k^{th} user with covariance matrix $\sigma_k^2 \boldsymbol{I}$.

The achievable data rate of the k^{th} user over time slot t can be evaluated by the central controller as

$$\hat{R}_k(t) \ = \log_2 \left| \boldsymbol{I} + \left(\tilde{\boldsymbol{U}}_k(t)^\dagger \hat{\boldsymbol{R}}_{\bar{k}}(t) \tilde{\boldsymbol{U}}_k(t) \right)^{-1} \tilde{\boldsymbol{U}}_k^\dagger(t) \hat{\boldsymbol{H}}_k \boldsymbol{V}_k(t) \boldsymbol{V}_k(t)^\dagger \hat{\boldsymbol{H}}_k^\dagger \tilde{\boldsymbol{U}}_k(t) \right| \ , \tag{5.2}$$

where

$$\hat{\boldsymbol{R}}_{\bar{k}}(t) = \sum_{l=1, l \neq k}^{K} \hat{\boldsymbol{H}}_k \boldsymbol{V}_l(t) \boldsymbol{V}_l(t)^\dagger \hat{\boldsymbol{H}}_k^\dagger + \sigma_k^2 \boldsymbol{I} \ , \tag{5.3}$$

and $\hat{\boldsymbol{H}}_k = \hat{\boldsymbol{S}}_k \hat{\boldsymbol{B}}_k^\dagger$. Since the central controller lacks the full knowledge of the channel matrix, the combiner and the left singular matrix are considered as a whole, namely, $\tilde{\boldsymbol{U}}_k = \boldsymbol{A}_k^\dagger \boldsymbol{U}_k$. $\boldsymbol{U}_k \in \mathbb{C}^{N_{r,k} \times N_{r,k}}$ is the combiner applied at the k^{th} receiver. Instead of optimizing

$\boldsymbol{U}_k(t)$, the central controller optimizes $\tilde{\boldsymbol{U}}_k(t)$.

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:

$$\begin{aligned} \max_{\substack{\{\boldsymbol{V}_k(t), \tilde{\boldsymbol{U}}_k(t)\}_{k\in\mathcal{K}} \\ t\in\mathcal{T}}} \quad & \frac{1}{T}\sum_{t=1}^{T}\sum_{k=1}^{K}\hat{R}_k(t) \\ s.t. \quad & \mathrm{Tr}(\Gamma_m \sum_{k=1}^{K} \boldsymbol{V}_k(t)\boldsymbol{V}_k(t)^{\dagger}) \leq P_m, m = 1, \ldots, M,\ t = 1, \ldots, T \\ & \sum_{t=1}^{T}\hat{R}_k(t) = b_k \sum_{t=1}^{T}\sum_{k=1}^{K}\hat{R}_k(t),\ \forall k \in \mathcal{K} \end{aligned} \tag{5.4}$$

where P_m is the maximum transmit power of the m^{th} AP and $\mathrm{Tr}(\cdot)$ denotes the trace of a matrix $(\cdot)$. 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 $b_k = 1/K, \forall k$, which aims to achieve same throughput for all users.

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 (5.4) 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 a cross-layer optimization problem with fairness constraints for multi-AP MIMO networks.

5.3 Multiuser MIMO Fair Scheduling with Joint Optimization

In this section, we propose an alternating algorithm to solve the formulated problem, which jointly determines the active user subset for each time slot and the MIMO weights of all active users. In order to make the problem tractable, we propose several transformations to Problem (5.4) 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 $c \leq \sum\limits_{k=1}^{K} \sum\limits_{t=1}^{T} R_k(t)$. Then, the equality constraints can be converted into a set of inequality constraints, i.e., $b_k c \leq \sum_{t=1}^{T} R_k(t), \forall k$. Thus, the optimization problem (5.4) can be reformulated as,

$$\begin{aligned} \max_{c,\{\boldsymbol{V}_k(t),\tilde{\boldsymbol{U}}_k(t)\}_{\substack{k\in\mathcal{K}\\ t\in\mathcal{T}}}} \quad & c \\ s.t. \quad & \mathrm{Tr}(\Gamma_m \sum_{k=1}^{K} \boldsymbol{V}_k(t)\boldsymbol{V}_k(t)^{\dagger}) \leq P_m, \forall m = 1,\ldots,M,\ t = 1,\ldots,T \\ & \sum_{t=1}^{T} R_k(t) \geq b_k c,\ \forall k \in \mathcal{K}\ , \end{aligned} \tag{5.5}$$

By driving c toward $c = \sum\limits_{k=1}^{K} \sum\limits_{t=1}^{T} R_k(t)$, a solution to Problem (5.5) will also solve Problem (5.4). Note also that the constraint $c \leq \sum\limits_{k=1}^{K} \sum\limits_{t=1}^{T} R_k(t)$ is implicitly satisfied when the K constraints $\sum\limits_{t=1}^{T} R_k(t) \geq b_k c,\ \forall k \in \mathcal{K}$ are met, since we have $\sum\limits_{k=1}^{K} b_k = 1$. Therefore, we can omit the constraint $c \leq \sum\limits_{k=1}^{K} \sum\limits_{t=1}^{T} R_k(t)$ in Problem (5.5).

Table 5.1: 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 $\mathrm{Tr}(\Gamma_m \sum_{k=1}^{K} \boldsymbol{V}_k(t)\boldsymbol{V}_k(t)^\dagger) \leq P_m, \forall m, \forall t \in \mathcal{T}$
2.	***while*** $I \leq I_{max}$
3.	$I \leftarrow I + 1$
4.	***for*** t from 1 to T
5.	update $\boldsymbol{V}_k(t)$ and $\tilde{\boldsymbol{U}}_k(t)$ and c for $k \in \mathcal{K}$ by solving problem (5.6)
6.	***Qiut if*** $\lvert \sum_{t=1}^{T}\sum_{k=1}^{K} R_k^I(t) - \sum_{t=1}^{T}\sum_{k=1}^{K} R_k^{I-1}(t)\rvert \leq \varepsilon$

The solution to problem (5.5) involves $2K \times T$ variables, i.e., the precoder $\boldsymbol{V}_k(t)$ and $\tilde{\boldsymbol{U}}_k(t)$ for each user in each time slot. 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. The number of variables will therefore make the solution of Problem (5.5) quite complex. To reduce the number of variables, Problem (5.5) 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:

$$\begin{aligned} \max_{c,\{\boldsymbol{V}_k(t),\tilde{\boldsymbol{U}}_k(t)\}_{k\in\mathcal{K}}} \quad & c \\ s.t. \quad & \mathrm{Tr}(\Gamma_m \sum_{k=1}^{K} \boldsymbol{V}_k(t)\boldsymbol{V}_k(t)^\dagger) \leq P_m, m = 1, \ldots, M \\ & R_k(t) \geq b_k c - \sum_{s=1, s\neq t}^{T} R_k(s),\ \forall k \in \mathcal{K} \end{aligned} \tag{5.6}$$

The T subproblems can be solved iteratively to find a suboptimal solution to the problem (5.5).

The iterative algorithm is summarized in Table 5.1. In each iteration, it solves the problem (5.6) for each time slot sequentially. Since the problem (5.5) is non-convex, a global optimum cannot be found using alternating optimization. However, since alternating optimization provides monotonously non-decreasing solutions c to Problem (5.5) and variable c is upper bounded, the alternating optimization solution will converge to a local optimum

of problem (5.5). The convergence of the alternating algorithm is proved as follows.

Let

$$\{\boldsymbol{V}_k(1)^{(I)}, \tilde{\boldsymbol{U}}_k(1)^{(I)}, \ldots, \boldsymbol{V}_k(T)^{(I)}, \tilde{\boldsymbol{U}}_k(T)^{(I)}\}_{k\in\mathcal{K}}$$

be the optimized precoders and combiners after the I^{th} iteration, which corresponds to $\{\hat{R}_k(1)^{(I)}, \ldots, \hat{R}_k(T)^{(I)}\}_{k\in\mathcal{K}}$ and $c^{(I)}$. During the $I+1^{\text{th}}$ iteration, we will solve the T subproblems sequentially to update the precoders and combiners. The solution to each subproblem is to maximize the objective c. The solution to the t^{th} subproblem serves the starting point of the $t+1^{\text{th}}$ subproblem. Therefore, we have

$$\begin{aligned}
&c^{I+1}\left(\{\boldsymbol{V}_k(1)^{(I+1)}, \tilde{\boldsymbol{U}}_k(1)^{(I+1)}, \ldots, \boldsymbol{V}_k(T)^{(I+1)}, \tilde{\boldsymbol{U}}_k(T)^{(I+1)}\}_{k\in\mathcal{K}}\right)\\
&\geq c^{I+1}\left(\{\boldsymbol{V}_k(1)^{(I+1)}, \tilde{\boldsymbol{U}}_k(1)^{(I+1)}, \ldots, \boldsymbol{V}_k(T-1)^{(I+1)}, \tilde{\boldsymbol{U}}_k(T-1)^{(I+1)}, \boldsymbol{V}_k(T)^{(I)}, \tilde{\boldsymbol{U}}_k(T)^{(I)}\}_{k\in\mathcal{K}}\right)\\
&\geq \cdots \geq c^{I+1}\left(\{\boldsymbol{V}_k(1)^{(I+1)}, \tilde{\boldsymbol{U}}_k(1)^{(I+1)}, \ldots, \boldsymbol{V}_k(t)^{(I)}, \tilde{\boldsymbol{U}}_k(t)^{(I)}, \ldots, \boldsymbol{V}_k(T)^{(I)}, \tilde{\boldsymbol{U}}_k(T)^{(I)}\}_{k\in\mathcal{K}}\right)\\
&\geq \cdots \geq c^{I}\left(\{\boldsymbol{V}_k(1)^{(I)}, \tilde{\boldsymbol{U}}_k(1)^{(I)}, \ldots, \boldsymbol{V}_k(T)^{(I)}, \tilde{\boldsymbol{U}}_k(T)^{(I)}\}_{k\in\mathcal{K}}\right).
\end{aligned}$$

Thus, the alternating maximization process leads to monotonous increase of the objective c. With the fact of the variable c representing the total throughput over T time slots is upper bounded by the transmit power contraints, we can conclude that the alternating algorithm converges to a local maximum.

Lemma 1. *If we have a locally optimal solution*

$$\boldsymbol{X} = \left(\{\boldsymbol{V}_k(1), \tilde{\boldsymbol{U}}_k(1)\}_{k\in\mathcal{K}}, \cdots, \{\boldsymbol{V}_k(T), \tilde{\boldsymbol{U}}_k(T)\}_{k\in\mathcal{K}}\right)$$

to problem (5.5), it is also locally optimal for problem (5.4).

Proof. Let $f(c, \boldsymbol{X}, \boldsymbol{\lambda})$ be the Lagrangian of problem (5.5) and $h(\boldsymbol{X}, \boldsymbol{\lambda})$ be the Lagrangian

of problem (5.4), where $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers. We have

$$\begin{aligned} f(c, \boldsymbol{X}, \boldsymbol{\lambda}) = \; & -c + \sum_{m=1}^{M} \lambda_m (\mathrm{Tr}\left(\Gamma_m \sum_{k=1}^{K} \boldsymbol{V}_k(t)\boldsymbol{V}_k(t)^\dagger\right) - P_m\Big) \\ & + \sum_{k=1}^{K} \lambda_{M+k} \left(b_k c - \sum_{t=1}^{T} R_k(t)\right) \end{aligned}$$

and

$$\begin{aligned} h(\boldsymbol{X}, \boldsymbol{\lambda}) = \; & -\sum_{t=1}^{T}\sum_{k=1}^{K} \hat{R}_k(t) + \sum_{m=1}^{M} \lambda_m (\mathrm{Tr}\left(\Gamma_m \sum_{k=1}^{K} \boldsymbol{V}_k(t)\boldsymbol{V}_k(t)^\dagger\right) - P_m\Big) \\ & + \sum_{k=1}^{K} \lambda_{M+k} \left(b_k \sum_{t=1}^{T}\sum_{k=1}^{K} \hat{R}_k(t) - \sum_{t=1}^{T} R_k(t)\right) \end{aligned}$$

If

$$\boldsymbol{X} = \left(\{\boldsymbol{V}_k(1), \tilde{\boldsymbol{U}}_k(1)\}_{k\in\mathcal{K}}, \cdots, \{\boldsymbol{V}_k(T), \tilde{\boldsymbol{U}}_k(T)\}_{k\in\mathcal{K}}\right)$$

is a optimal point of problem (5.5), i.e., $\nabla_c f = 0$, $\nabla_{\boldsymbol{X}} f = \mathbf{0}$ and $\nabla_{\boldsymbol{\lambda}} f = \mathbf{0}$, we will have $\nabla_{\boldsymbol{X}} h = \mathbf{0}$ and $\nabla_{\boldsymbol{\lambda}} h = \mathbf{0}$. Thus, it is also a optimal point for problem (5.4). □

Based on ***Lemma 1***, we can further conclude that the alternating optimization also approaches a local optimum of problem (5.4).

To solve each of the T subproblems, we utilize the Lagrangian dual method. The dual function of (5.6) is given by

$$g(\boldsymbol{\lambda}(t)) \;\; = \min_{\mathrm{Tr}(\Gamma_m \boldsymbol{V}(t)\boldsymbol{V}(t)^\dagger) \le P_m} L(c, \{\boldsymbol{V}_k, \tilde{\boldsymbol{U}}_k\}_{k\in\mathcal{K}}, \boldsymbol{\lambda}(t)) \;, \tag{5.7}$$

where the Lagrangian of (5.6) is

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

and $\boldsymbol{\lambda}(t) = \{\lambda_1(t), \ldots, \lambda_K(t)\}$ with $\lambda_k(t) \ge 0, \forall k \in \mathcal{K}$ are the Lagrange multipliers. Since a linear function is bounded below only when it is identically zero, it is straightforward to

prove that $g(\boldsymbol{\lambda}(t)) = -\infty$ except when $\sum_{k=1}^{K} b_k \lambda_k(t) - 1 = 0$.

The dual problem is then given by

$$\begin{aligned} \max_{\boldsymbol{\lambda}(t)} \quad & \left\{ \min_{\mathrm{Tr}(\Gamma_m \boldsymbol{V}(t)\boldsymbol{V}(t)^\dagger) \le P_m} - \sum_{k=1}^{K} \lambda_k(t) \hat{R}_k(t) \right\} \\ s.t. \quad & \sum_{k=1}^{K} b_k \lambda_k(t) = 1, \quad \lambda_k(t) \ge 0, \forall k \in \mathcal{K}. \end{aligned} \tag{5.8}$$

The dual problem can be solved iteratively: the precoders and combiners ($\boldsymbol{V}_k(t)$'s and $\tilde{\boldsymbol{U}}_k(t)$'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. The Lagrange multipliers for the i^{th} iteration are given by

$$\lambda_k^{(i)}(t) = \max\left(\lambda_k^{(i-1)}(t) + \alpha_i \left(c b_k - \sum_{t=1}^{T} R_k(t) \right), 0 \right) \tag{5.9}$$

where α_i is the step size for the i^{th} iteration. To meet the equality constraint of $\boldsymbol{\lambda}$, the multipliers need to be further normalized as $\lambda_k = \lambda_k / \sum_{k=1}^{K} b_k \lambda_k$.

To solve for the $\boldsymbol{V}_k(t)$'s and $\tilde{\boldsymbol{U}}_k(t)$'s with given Lagrange multipliers, the minimization problem in (5.8) can be rewritten as a WSRM problem under per-AP power constraint,

$$\begin{aligned} \max_{\{\boldsymbol{V}_k(t), \tilde{\boldsymbol{U}}_k(t)\}_{k \in \mathcal{K}}} & \sum_{k=1}^{K} \lambda_k \hat{R}_k(t) \\ \mathrm{Tr}(\Gamma_m \boldsymbol{V}(t)\boldsymbol{V}(t)^\dagger) & \le P_m \ . \end{aligned} \tag{5.10}$$

Note that the solution to problem (5.8) has been discussed in Section 4.3.2. The iterative algorithm described in Table 4.2 can be used to jointly solve the precoders and combiners in problem (5.8) for a certain time slot t.

5.4 Multiuser MIMO Fair Scheduling via a Two-stage Approach

The alternating optimization method proposed in Section 5.3 determines the active user set for each time slot by jointly optimizing the precoders and stream allocation of each user over T time slots. The active user set and the corresponding data rates are the solution to a WSRM problem. Based on this observation, we propose a lower-complexity heuristic approach to approximate the solution given by the alternating optimization algorithm.

Since the channels in our target scenarios are assumed to be fixed during T time slots, we can assign an optimized set of communications for each time slot to obtain high aggregate performance, while achieving fairness among the competing users. Our 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.

5.4.1 Communication sets generation

In this section, we present an efficient approach to generate multiple diverse and high-performance communication sets iteratively. In each iteration, one communication set is generated through a 2-step procedure. First, a WSRM problem is solved to determine the active user set and calculate the MIMO weights of the active users, which can also determine the stream allocation for each user. Second, 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. Repeat the 2-step procedure n times will produce n MU-MIMO communication sets. This approach can ensure user diversity across the communication sets and balance the probability of activating different users over a number of communication sets. After a specified number of communication sets are generated in

this iterative manner, a final group of single-user communication sets is added to ensure that there is a solution that satisfies the fairness constraints.

Solving weighted sum rate maximization problem

For the n^{th} iteration, let $\boldsymbol{V}_{k,n}$ and $\tilde{\boldsymbol{U}}_{k,n}$ be the MIMO precoder and composed combiner for the k^{th} user, and $w_{k,n}$ be the user weights for the k^{th} user. we solve a WSRM problem to determine the active users, as well as their MIMO weights. Recall that the general form of a WSRM problem with per-AP power constraint is given as follows:

$$\begin{aligned} \max_{\{\boldsymbol{V}_{k,n},\tilde{\boldsymbol{U}}_{k,n}\}_{k\in\mathcal{K}}} \quad & \sum_{k=1}^{K} w_{k,n}\hat{R}_{k,n} \\ s.t. \quad & \sum_{k=1}^{K} Tr(\boldsymbol{\Gamma}_m \boldsymbol{V}_{k,n}\boldsymbol{V}_{k,n}^{\dagger}) \leq P_m, \forall m \ , \end{aligned} \tag{5.11}$$

where $\hat{R}_{k,n}$ is the data rate of the k^{th} user in the n^{th} communication set.

With given user weights, the algorithm for solving problem (5.11) has been proposed in Section 4.3. The proposed combined user selection and MIMO weights calculation approach in Section 4.3 can solve the problem with a very low computational complexity and maintain good scalability for large number of users. The solution to problem (5.11) determines the precoders and combiners for each users with a certain specification of user weights. A user is activated if it has non-zero data rate.

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 $\hat{\boldsymbol{R}}_k$ be a $1 \times N$ vector that contains the data rates of the k^{th} user, i.e. $\hat{R}_{k,n}$ denotes the data rate of the k^{th} user in the n^{th} communication set. When generating the $(n+1)^{\text{th}}$ communication set after the first n 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 n

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

$$w_{k,n+1} \begin{cases} \geq w_{j,n+1} & \text{if } \ 0 \leq u_{k,n}/b_k \leq u_{j,n}/b_j \leq 1 \\ = 0 & \text{if } \ u_{k,n}/b_k \geq 1 \end{cases} \tag{5.12}$$

where $u_{k,n}$ is the bandwidth proportion of the kth user from previously computed n communication sets, given by

$$u_{k,n} = \sum_{i=1}^{n} \hat{R}_{k,i} / \sum_{k=1}^{K} \sum_{i=1}^{n} \hat{R}_{k,i} \ .$$

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 for the $n + 1^{\text{th}}$ iteration as follows:

$$w_{k,n+1} = \max(1 - u_{k,n}/b_k, 0). \tag{5.13}$$

As a result, the users that have already achieved their target bandwidth fractions are excluded from the current round of calculation.

Single-user MIMO communication sets

After the first two steps are iterated a specified number of times, generating N communication sets, 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

$$\hat{r}_k = \max_{\left\{\mathrm{Tr}(\boldsymbol{\Gamma}_m \boldsymbol{V}_k \boldsymbol{V}_k^\dagger) \le P_m\right\}} \log_2 \left| \boldsymbol{I} + \hat{\boldsymbol{R}}_{n_k}^{-1} \hat{\boldsymbol{H}}_k \boldsymbol{V}_k \boldsymbol{V}_k^\dagger \hat{\boldsymbol{H}}_k^\dagger \right| . \tag{5.14}$$

Let $\boldsymbol{Q}_k = \boldsymbol{V}_k \boldsymbol{V}_k^\dagger$ be the transmit covariance matrix. Since the problem (5.14) is convex over $\boldsymbol{Q}_k$, its optimal solution can be obtained through standard techniques. Therefore, in the $N + k^{\mathrm{th}}$ communication set, the k^{th} user has the data rate of $\hat{R}_{k,N+k} = \hat{r}_k$, while all other users have zero data rate as $\hat{R}_{i,N+k} = 0, \forall i \neq k$.

5.4.2 Scheduler calculation

After generating $N_{tot} = N + K$ communication sets as discussed in Section 5.4.1, our focus is on developing a schedule to achieve both high aggregate performance and the target fairness. Let $\hat{\boldsymbol{R}}_n = \{\hat{R}_{1,n}, \hat{R}_{2,n}, \ldots, \hat{R}_{K,n}\}$ be the data rates of the n^{th} communication set, where $\hat{R}_{k,n}$ is the data rate of the k^{th} user in the n^{th} communication set. If $\hat{R}_{k,n} = 0$, it indicates that the k^{th} user is inactive in the n^{th} communication set. Recall the original formulation of the scheduling problem in (5.4). The per-AP power constraints are already 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.

To reformulate the scheduling problem with a given set of communication sets, we introduce a set of binary variables $\rho_{n,t} \in \{0, 1\}$, which represents whether the n^{th} communication set is assigned to the t^{th} time slot. $\rho_{n,t} = 1$ indicates that the n^{th} communication set will communicate in the t^{th} time slot. Then we have,

$$\hat{R}_k(t) = \sum_{n=1}^{N_{tot}} \sum_{t=1}^{T} \rho_{n,t} \hat{R}_{k,n}.$$

Therefore, the scheduling problem (5.4) can be rewritten into

$$\begin{aligned}
\max_{\rho_{n,t}} \quad & \sum_{k=1}^{K}\sum_{n=1}^{N_{tot}}\sum_{t=1}^{T}\rho_{n,t}\hat{R}_{k,n} \\
s.t. \quad & \sum_{n=1}^{N_{tot}}\sum_{t=1}^{T}\rho_{n,t}\hat{R}_{k,n} = b_k\sum_{k=1}^{K}\sum_{n=1}^{N_{tot}}\sum_{t=1}^{T}\rho_{n,t}\hat{R}_{k,n}, \forall k \in \mathcal{K} \\
& \sum_{n=1}^{N_{tot}}\rho_{n,t} = 1, t = 1, \ldots, T \\
& \rho_{n,t} \in \{0,1\}, \forall k \in \mathcal{K}, t = 1, \ldots, K \ .
\end{aligned} \tag{5.15}$$

Problem (5.15) forms a binary integer programming problem with $N_{tot} \times T$ variables. To reduce the size of the optimization problem, the problem can be further simplified by setting $x_n = \sum_{t=1}^{T}\rho_{n,t}, n = 1, \ldots, N_{tot}$, which represents the number of time slots scheduled for the n^{th} communication sets. We can then convert Problem (5.15) into the following formulation

$$\begin{aligned}
\max_{x_n} \quad & \sum_{k=1}^{K}\sum_{n=1}^{N_{tot}}\hat{R}_{k,n}x_n \\
s.t. \quad & C1: \ \sum_{n=1}^{N_{tot}}\hat{R}_{k,n}x_n = b_k\sum_{k=1}^{K}\sum_{n=1}^{N_{tot}}\hat{R}_{k,n}x_n, \forall k \in \mathcal{K} \\
& C2: \ \sum_{n=1}^{N_{tot}}x_n = T \\
& C3: \ x_n \in \mathbb{Z}, n = 1, \ldots, N_{tot} \ .
\end{aligned} \tag{5.16}$$

The number of variables are reduced to N_{tot}, which is much smaller than that in Problem (5.15).

Note that the fairness constraint C_1 contains K equality constraints. 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

$$C_3 : d(1-\epsilon)b_k \leq \sum_{n=1}^{N_{tot}}\hat{R}_{k,n}x_n \leq d(1+\epsilon)b_k, k = 1, \ldots, K. \tag{5.17}$$

where $d = \sum_{k=1}^{K} \sum_{n=1}^{N_{tot}} \hat{R}_{k,n} x_n$. By replacing C_1 in (5.16). 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.

A general way to solve the formulated integer linear programming (ILP) problem (5.16) is to solve its LP relaxation and then round the entries of the solution. The LP relaxation of (5.16) with ϵ-approximate fairness is given by

$$\begin{aligned} \max_{\boldsymbol{x}} \quad & \sum_{k=1}^{K} \sum_{n=1}^{N_{tot}} \hat{R}_{k,n} x_n \\ s.t. \quad & d(1-\epsilon) b_j \leq \sum_{n=1}^{N_{tot}} \hat{R}_{k,n} x_n \leq d(1-\epsilon) b_j \\ & x_n \geq 0, n = 1, \ldots, N_{tot} \\ & \textstyle\sum_{n=1}^{N_{tot}} x_n \leq T. \end{aligned} \tag{5.18}$$

Note that the relaxed LP problem (5.18) provides an upper bound on sum rate for any feasible solution to the ILP problem (5.16). Since the problem (5.18) is a standard LP problem, it can be addressed by well-known techniques such as the interior-point method. The relaxed LP has at least one feasible solution, due to the inclusion of the single-user communication sets.

To solve the formulated scheduling problem, we introduce a simple interior-point method, called the barrier method [82]. First, we define a logarithmic barrier function $\phi(\boldsymbol{x})$

$$\begin{aligned} \phi(\boldsymbol{x}) \quad = & - \sum_{n=1}^{N_{tot}} \log(x_n) - \log(T - \sum_{n=1}^{N_{tot}} x_n) \\ & - \sum_{k=1}^{K} \log \left(d(1+\epsilon) b_k - \sum_{n=1}^{N_{tot}} \hat{R}_{k,n} x_n \right) \\ & - \sum_{k=1}^{K} \log \left(\sum_{n=1}^{N_{tot}} \hat{R}_{k,n} x_n - d(1-\epsilon) b_k \right) \end{aligned} \tag{5.19}$$

Table 5.2: Computing the Schedule for Given Communication Sets

Input: data rates in N_{tot} candidate communication sets $\{\hat{R}_{k,n}\}_{\forall k,\forall n}$, desired bandwidth proportion $\boldsymbol{b} = \{b_1, \ldots, b_K\}$
Output: optimized schedule $\boldsymbol{s}^* = \{s_1^*, \ldots, s_N^*\}$
1. Initialization: Given feasible $\boldsymbol{x} = \boldsymbol{x}_0$, $t := t_0 > 0$, $\mu > 0$, ε.
2. **Repeat**
3. Compute the optimal solution $\boldsymbol{x}^*(t)$ to (5.20) starting at $\boldsymbol{x}$.
4. Update $\boldsymbol{x} := \boldsymbol{x}^*(t)$.
5. Quit if $(2K + N + 1)/t \leq \varepsilon_b$.
6. Update $t := \mu t$.
7. $\boldsymbol{s}^* = \text{round}(\boldsymbol{x})$.

Next, we define an unconstrained minimization problem with parameter t,

$$\min_{\boldsymbol{x}} f_t(\boldsymbol{x}) = -t \sum_{j=1}^{K} \sum_{i=1}^{N_{tot}} r_{j,i} x_i + \phi(\boldsymbol{x}) \tag{5.20}$$

The optimal solution to problem (5.20) is an approximation of the optimal solution to problem (5.18), where $t > 0$ is a parameter that sets the accuracy of the approximation. As t increases, the approximation becomes more accurate. The outline of barrier method is summarized in Table 5.2. To solve the unconstrained minimization problem (5.20) in each iteration, Newton's method is used to compute the optimal solution. With a given t, the Newton step $\Delta \boldsymbol{x}_t$ at $\boldsymbol{x}$ is given by

$$\nabla^2 f_t(\boldsymbol{x}) \Delta \boldsymbol{x}_t = -\nabla f_t(\boldsymbol{x}) \ . \tag{5.21}$$

Where $\nabla^2 f_t(\boldsymbol{x})$ and $\nabla f_t(\boldsymbol{x})$ are the Hessian and the gradient of $f_t(\boldsymbol{x})$, respectively. Generally, the inverse of an $N_{tot} \times N_{tot}$ matrix $\nabla^2 f_t(\boldsymbol{x})$ requires $\mathcal{O}(N_{tot}^3)$ arithmetic operations, which can be reduced to $\mathcal{O}(N_{tot} K^2)$ using the fast barrier method proposed in [83], since K is typically much smaller than N_{tot}.

Once we have the optimal solution to the LP problem, we perform the following rounding procedure. First, sort the $res_i = x_i^* - \lfloor x_i^* \rfloor$ in descending order. Then, the solution x_i^*'s of the first I user sets with the higher res_i value will be rounded to $\lceil x_i^* \rceil$, while the remain-

ing x_i^*'s will be rounded to $\lfloor x_i^* \rfloor$, where $I = T - \sum_{i=1}^{K} \lfloor x_i^* \rfloor$. The rounded solution determines the number of time slots assigned to each communication set. Although the fairness constraint C_3 might be violated after the rounding procedure, it will be demonstrated in the simulations that any deviation from the targeted fairness is quite small.

5.5 Algorithm Implementation and Complexity Analysis

In this section, we discuss the implementation of the two proposed algorithms and analyze their complexity. To facilitate the discussion, we make the assumption that the cooperative APs within a single cluster are tied via high-speed links to a central processor, which is responsible for calculating the communication schedule.

5.5.1 Algorithm implementation

First, the central processor requests the CSI from the cooperative APs in one cluster. During this step, each AP takes a turn to send sounding packets and collect CSI from each user. Then the central processor computes the communication schedule for the following T time slots and the corresponding MIMO weights for each communication sets. For the two scheduling algorithms proposed in this paper, different operations can be implemented to improve the algorithm efficiency.

For the joint scheduling algorithm proposed in Section 5.3, the optimization of T subproblems iteratively requires significant computation effort. Since T is often chosen to be a large value, typically much larger than K, for stationary or limited mobility environment, the computational overhead of the joint algorithm can be significantly reduced via *time-slot aggregation*. Specifically, a number of adjacent time slots can be aggregated and deemed as a "virtual time window". The communication schedules for the time slots within one "virtual time window" are the same, i.e., the same user set is activated and these users are assigned with fixed MIMO weights during one "virtual time window". Based on this idea, if the T time slots are aggregated into T_w "virtual time windows", only T_w subproblems

instead of T need to be solved in each iteration.

The computational overhead of the heuristic algorithm proposed in Section 5.4 is mainly from the generation of N_{tot} communication sets, which involves intensive calculation of the corresponding MIMO weights for each communication set. The algorithm efficiency can be improved through parallel processing, i.e., distributing the workload to P parallel working processes. The main process first distributes the collected CSI to all helper processes. Each process then runs the communication sets generation algorithm in Section 5.4.1 to produce N_p communication sets. The generated PN_p communication sets and the MIMO weights are then passed to the main process. Finally, the link scheduling algorithm is performed by the main process to determine the number of time slots allocated to each communication set. Since the user weights need to be updated iteratively based on the previously calculated communication sets as discussed in Section 5.4.1, the performance of the parallel processing highly depends on the approach of local weights update for each process. The basic idea is to assign different initial user weights for each process and follow the user weights update approach proposed in Section 5.4.1 locally over each process. For example, the main process can start with identical user weights of $w_k = 1, \forall k$. Helper processes can employ a simple randomized method to initialize the binary user weights (i.e., $w_k = \{0, 1\}$). The basic idea is that the users with larger target bandwidth fractions have higher probability to be activated during the initialization stage.

5.5.2 Complexity analysis

In this section, we perform complexity analyses for the two proposed scheduling algorithms. The proposed algorithm using alternating optimization in Section 5.3 solves T subproblems in each main iteration. In terms of complexity per subproblem per iteration, it employs Lagrangian dual method to iteratively solve a WSRM problem and update the Lagrange multipliers using subgradient method. The number of iterations of the dual method is dominated by $O(1/\epsilon_s^2)$, where ϵ_s is the accuracy of the subgradient method. Therefore, in

each iteration of alternating optimization, it solves $O(T/\epsilon_s^2)$ WSRM problems for K users in (5.10).

The two-stage algorithm proposed in Section 5.4 involves communication sets generation and scheduler optimization. Generating N communication sets involves executing pre-user selection N times and solving N WSRM problems in (5.11) for K_0 users using the algorithm proposed in Chapter 4. The complexity of pre-user selection is dominated by $O(KK_0)$. For the scheduler optimization stage, the barrier method takes $O(\sqrt{m}\log\frac{m}{t_0\bar{\epsilon}_b})$ iterations, where $m = 2K + N + 1$. For each iteration, computing Newton step requires the inverse of a $(K+N)\times(K+N)$ matrix with $O((K+N)K^2)$ arithmetic operations.

Both of the proposed algorithms involve solving a set of WSRM problems, which requires the computationally expensive iterative algorithm of Table 3.1. The per-iteration complexity of updating the precoders and combiners is dominated by $O(U^2)$ for U users. Therefore, the complexity of solving WSRM for the two-stage method is significantly lower since it has $K_0 \ll K$ users, due to its pre-user selection.

In summary, the joint algorithm provides better aggregate performance with higher complexity. The two-stage algorithm has lower complexity, since N is generally on the same order as K. Moreover, unlike the joint algorithm, the complexity of the two-stage algorithm is independent of the number of times slots in one scheduling period.

5.6 Simulation Results

In this section, we report on simulation experiments to evaluate the performance of our proposed scheduling algorithms from Section 5.3 and Section 5.4 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 [84]

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 [69]. 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.

5.6.1 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 1000 random deployments. To evaluate fairness, we use the fairness index proposed in [80],

$$FI(\boldsymbol{u}, \boldsymbol{b}) = \exp\left(-\sum_{k=1}^{K} \left| \ln(u_k/b_k) \right| / K \right), \tag{5.22}$$

where u_k is the fraction of bandwidth allocated to the k^{th} user. The fairness index given by (6.8) takes values in $[0, 1]$, with 1 representing perfect fairness among users.

Different choices of parameter $\boldsymbol{b}$ achieve different fairness objectives, which can represent various QoS requirements. In our evaluations, we use the notion of time-based fairness since it has been shown to be particularly well-suited for multi-rate wireless networks. In [80], 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 $b_k = \hat{r}_k / \sum_{k=1}^{K} \hat{r}_k, \forall k$, where ρ_k is the interference-free data rate as discussed in Section 5.4.1. Once the precoders and combiners are calculated, instead of using the data rates given by the Shannons capacity formulas, the data rates are determined via the rate selection procedure discussed in Section 2.3.4.

5.6.2 Convergence properties

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

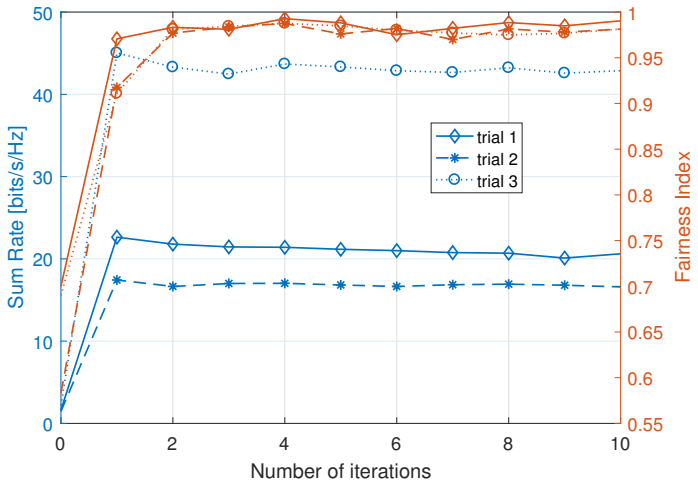

Figure 5.2: Sum-rate and fairness vs. number of iterations for alternating optimization method.

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 5.2. 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.

5.6.3 Performance with downlink traffic only

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

Sum-rate and fairness versus number of users

Figure 5.3 shows the achieved sum-rate and fairness of different algorithms as a function of the number of users. Note that the number of supportable users can be multiplied by the number of available orthogonal channels. Overall, **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 compar-

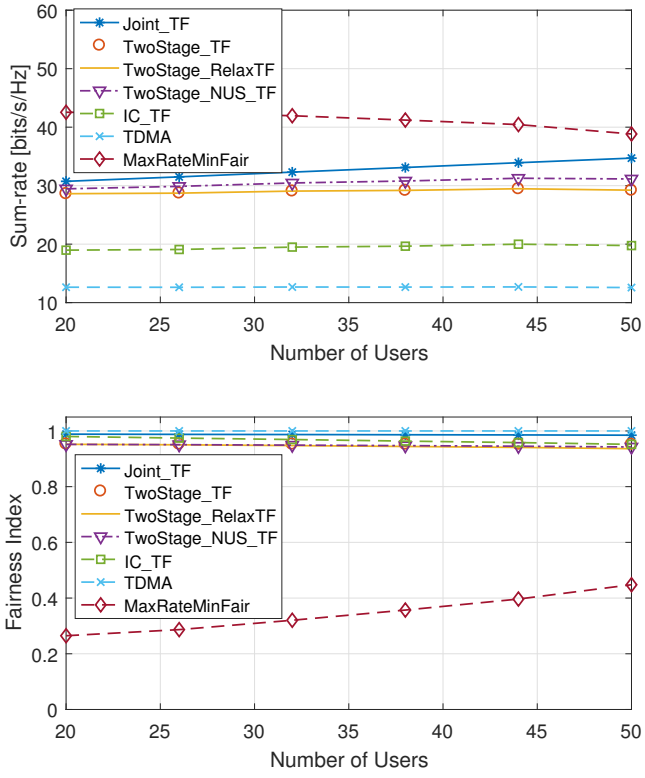

Figure 5.3: Sum-rate and fairness vs. number of users.

ing **TwoStage_TF** with **TwoStage_NUS_TF**. Although there is about a 5% sum-rate loss,

we will see in Section 5.6-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.

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 5.4, 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

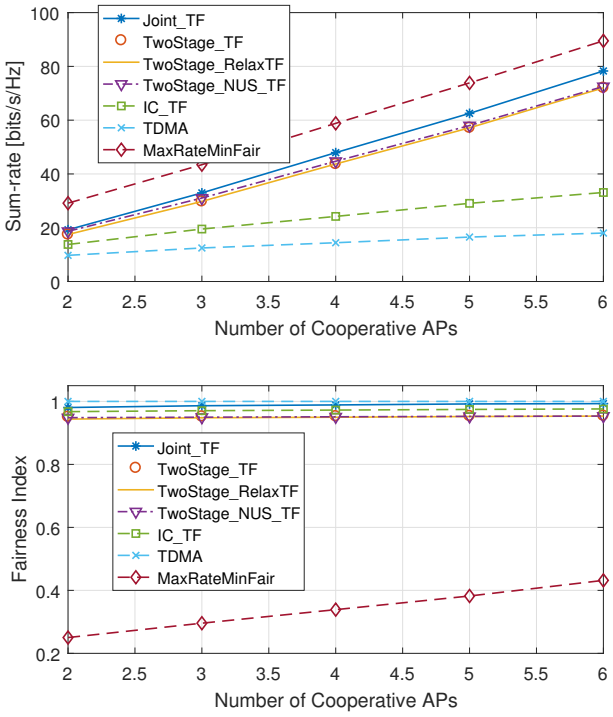

Figure 5.4: Sum-rate and fairness vs. number of APs

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 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.

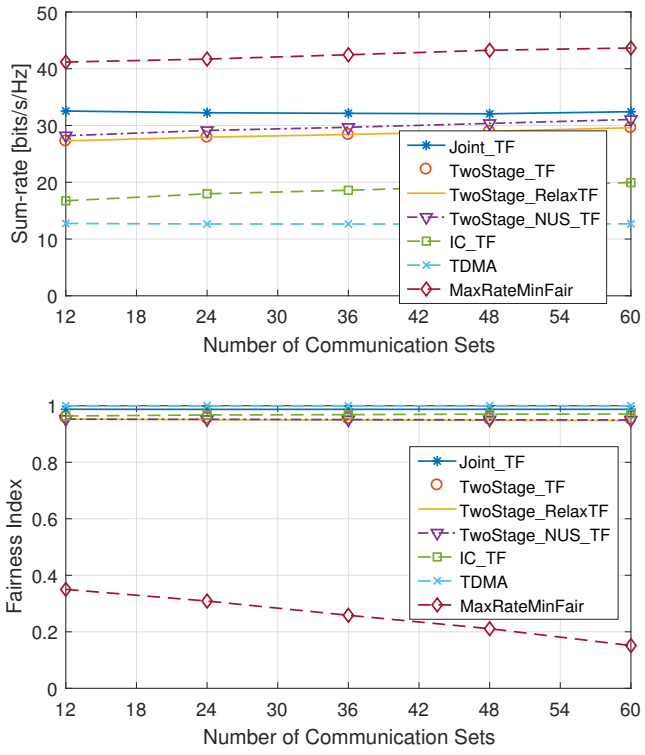

Figure 5.5: Sum-rate and fairness vs. number of communication sets.

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 5.5, 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.

Sum-rate and fairness versus fairness factor

We also present the results obtained with $K = 30$ at different choices of fairness factor ϵ. Figure 5.6 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 between **TwoStage_TF** and **TwoStage_NUS_TF** caused by pre-user selection is quite small. Figure 5.6 also illustrates how the two-stage algorithms allow for a performance-fairness tradeoff. Based on Figure 5.6, 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 5.6. The achieved fairness index can then be determined from the fairness curve. Any other op-

erating point in between the solutions to these two problems can also be determined from the plots.

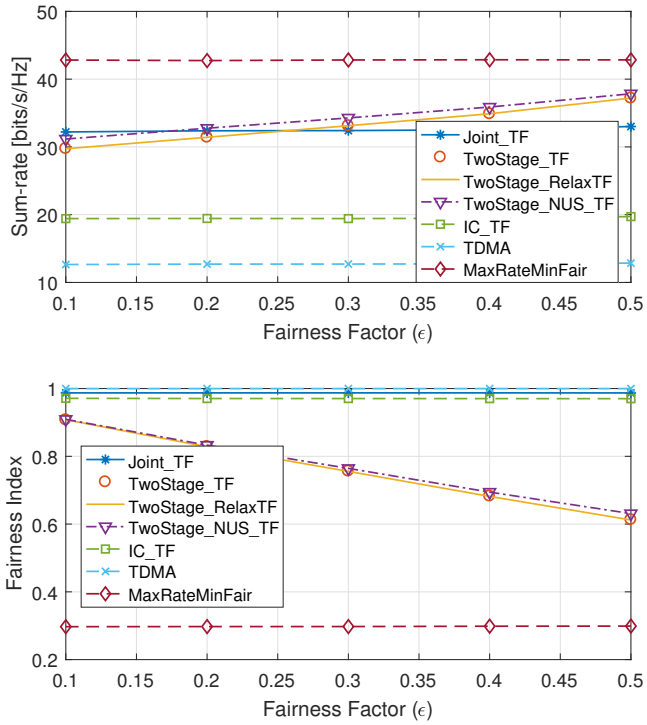

Figure 5.6: Sum-rate and fairness vs. fairness factor.

5.6.4 Performance with both downlink and uplink traffic

In this section, we evaluate the sum-rate and fairness performance of the proposed schedulers, when both downlink and uplink traffic is considered. In this section, we assume 20% traffic is on the uplink and $K = 30$. Since full cooperation among users is not possible, we assume only interference coordination is used by all algorithms on the uplink (except TDMA for which there is no interference).

Sum-rate and fairness versus number of users

In Figure 5.7, sum-rate and fairness achieved by different algorithms are plotted as a function of number of users. Since both IC and full cooperation gain advantage from multi-user

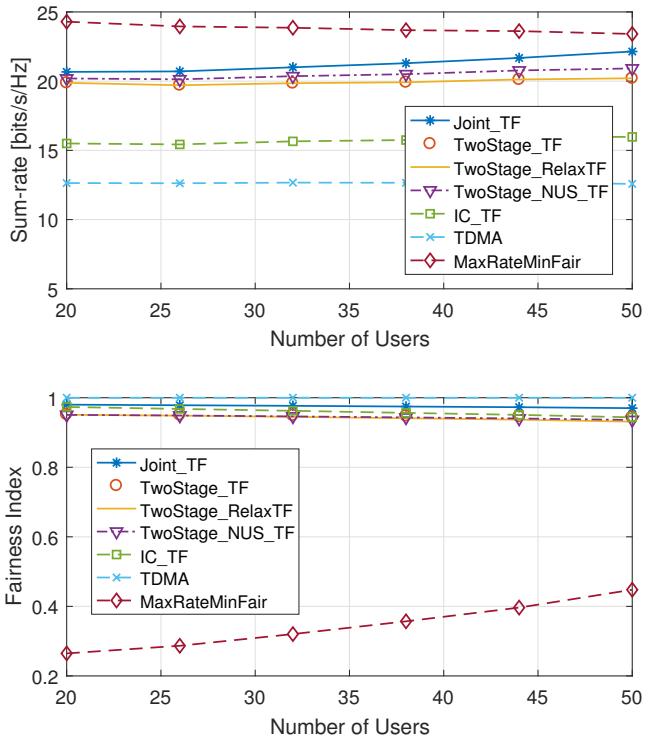

Figure 5.7: Sum-rate and fairness vs. number of users.

diversity, the overall sum-rate with both downlink and uplink traffic increases with the number of users. With 80% downlink traffic, the proposed schedulers with full cooperation still exhibit significant advantage in terms of aggregate performance compared to **IC_TF**. Since the uplink **IC_TF** also guarantees the target fairness, the achieved fairness of the proposed schedulers and **IC_TF** are always kept above 0.95. Thus, even though the proposed approaches target downlink transmissions, their benefits are still readily apparent when both downlink and uplink transmissions are considered in the typical scenario where traffic is heavier on the downlink.

Sum-rate and fairness versus number of APs

Figure 5.8 compares the sum-rate and fairness performance of different schedulers with different numbers of cooperative APs. We see that the sum-rates of the proposed schedulers

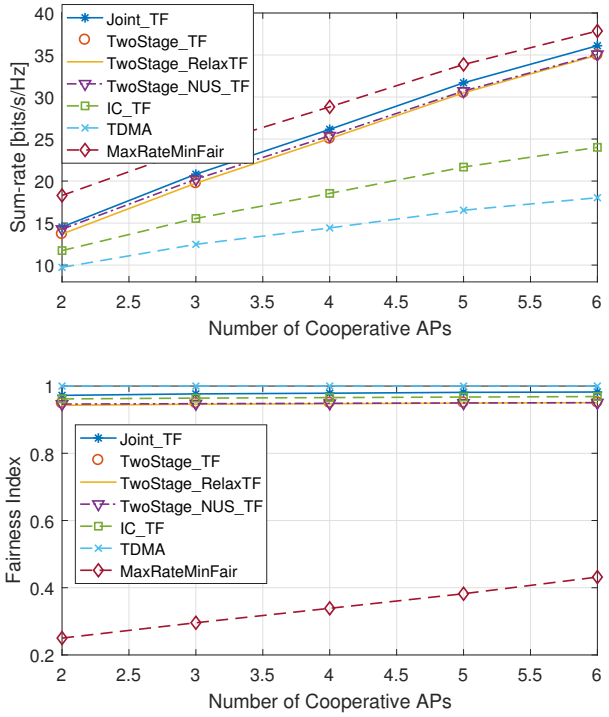

Figure 5.8: Sum-rate and fairness vs. number of APs.

still increase rapidly as the number of APs increases from 2 to 6. Note, however, that the increase is not quite linear due to the fact that the uplink transmissions cannot take advantage of joint transmission and have to rely solely on interference coordination, which does not scale linearly as was demonstrated in the previous section.

5.6.5 Running time evaluation

As mentioned earlier, our target environment is enterprise wireless networks where most users tend to be working and interacting in offices, cubicles, laboratories, and conference rooms. These environments are characterized by users that are stationary for periods of

time with intermittent short mobility periods. Due to their mostly stationary nature, these environments allow for a fairly complex scheduling algorithm to produce a schedule that can be in use for a moderate period of time, e.g. several seconds possibly even up to a few tens of seconds. 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 assumed stationary time of the network. In this subsection, we evaluate the execution times of the best performing of the scheduling algorithms evaluated in prior subsections. The algorithms are implemented in *Matlab* and run on an i7-2700K Intel CPU rated at 3.5 GHz with 32 GB RAM.

Figure 5.9 shows the running times of 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. We consider **Joint_TF**, **TwoStage_TF**, and **TwoStage_NUS_TF**, which were the top performers in terms of sum rate and fairness, i.e., A = **Joint_TF** with $T/T_w = 1$, B = **Joint_TF** with $T/T_w = 2$, C = **TwoStage_TF** with $P = 1$, D = **TwoStage_TF** with $P = 4$, E = **TwoStage_NUS_TF** with $P = 1$. For **Joint_TF**, we also consider the impact of aggregating multiple time slots together. 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_TF**, we also consider the impact of parallel execution, which can help speed up the communication set generation stage, which is the dominant factor in the running time.

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_TF** and **TwoStage_NUS_TF**, as indicated in Figure 5.5, generating $N = 60$ communication 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 basic **TwoStage_TF**, which employs pre-user selection, has running times from about 2.5 to 6 seconds, which is

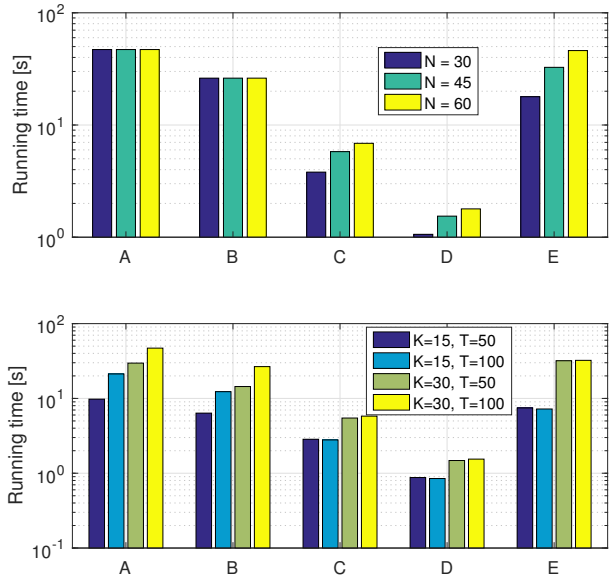

Figure 5.9: Running time of different algorithms.

more than 5 times faster than **TwoStage_NUS_TF**. For environments that need even shorter execution times, **TwoStage_TF** can be sped up further with parallel calculation of communication sets. Using four processors, **TwoStage_TF** only needs 1 to 2 seconds and is about 4 times faster than **TwoStage_TF**. While the complexity of the joint algorithm is independent of N, it is significantly affected by K and T. From Figure 5.9 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 5.9) 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.

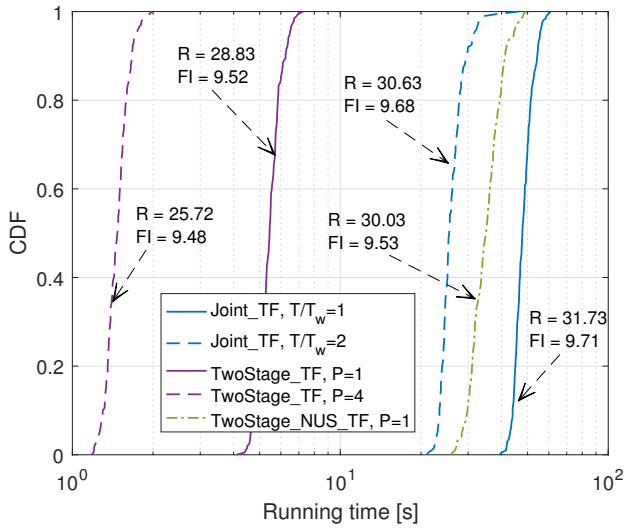

Figure 5.10: CDF of the running time for $K = 30, T = 100$.

It is also important to understand the trade-offs between sum rate performance and running time for the different algorithms under consideration. Figure 5.10 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 5.3. 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

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

Table 5.3: Sum-rate and running time performance for parallel processing.

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 $P = 4$.

5.7 Chapter Summary

In this chapter, we studied the MIMO link scheduling problem for a cluster of cooperative APs and a number of stationary users. We proposed alternative scheduling algorithms: the alternating optimization method and the two-stage method. The alternating optimization method jointly optimizes the MIMO weights and user selection for users over one scheduling period. The two-stage algorithm works in two phases: first, high-performance communication sets are generated via an iterative weighted sum-rate maximization procedure, and then an integer programming problem is solved through relaxation and rounding to produce a schedule that provides near-optimal performance for the chosen communication sets and given fairness constraint. Simulation results showed that the alternating optimization algorithm produces significantly higher aggregate throughput than all known approaches with a running time that is practical for scenarios with up to 50 users, while the two-stage algorithm produces aggregate throughput that is very close to existing heuristics while having significantly lower running time.

CHAPTER 6

MOBILITY-AWARE MULTI-USER MIMO LINK SCHEDULING

6.1 Introduction

This chapter focuses on developing a centralized schedule that achieves both high throughput and a target fairness criterion among users. Our previous research in chapter 5 that considers a similar problem targets only static network scenarios. While indoor WLANs, such as in office-type environments, are dominated by stationary clients, there is also limited mobility due to occasional device movements and environmental changes. Client mobility poses a unique challenges for the design of wireless protocols. In static environments, the wireless channels remain stable and past information can be relied on to optimize the performance. In contrast, the scheduler for mobile clients needs to accommodate frequent changes of wireless channels. Improvements have been investigated and tested by integrating the mobility-awareness into the design of client roaming, rate adaptation and frame aggregation scheme in WLANs [85, 86, 87]. In fact, bringing mobility hints into the scheduling algorithm can help sustain both good individual and overall performance. Applying an unified scheduling scheme to both static and mobile users lacks the ability to fully benifit from the throughput gain promised by multiuser MIMO techniques. Given the mix of users with diverse channel and mobility characteristics in next generation enterprise networks, different scheduling strategies are preferable for improving the overall performance.

In this chapter, we propose a mobility-aware multiuser MIMO link scheduling algorithm that distinguishes stationary and mobile users based on their CSI and applies different scheduling strategies within each user group. The central controller tracks the channel conditions of clients over time and applies a novel CSI similarity metric based on sub-

space collinearity to categorize users as either stationary or mobile. Our mobility-aware scheduling algorithm then separates static and mobile users into different time slots, and adaptively adjusts the number of time slots between the two categories to maintain fairness for both stationary and mobile users. The stationary user schedule is calculated in a highly optimized but fairly computationally expensive manner. However, since CSI does not change frequently for these users, this highly optimized schedule can be used for a significant number of scheduling periods. In contrast, the schedule for mobile users is done for each time slot using fresh CSI but in a highly efficient, less optimized fashion. The separation of users into two categories allows us to achieve the promise of expensive but very-high-performing scheduling algorithms that have been presented in the literature for stationary users, while still achieving reasonable performance for mobile users and ensuring fairness both across the two user categories and for individual users. Simulation results demonstrate that, when accounting for CSI feedback and scheduling overheads, our proposed scheduling algorithm with mobility awareness maintains very good fairness and provides substantial performance gains compared to conventional approaches that do not separate mobile and stationary users.

6.2 System Model and Problem Description

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. We focus on indoor environments, where most devices are stationary for a moderate amount of time between movements. When users' devices are not stationary, they move at low speeds (typically from walking with or rotating a hand-held device). This is a common scenario for most enterprise WLAN settings, such as most office-type environments. We focus on downlink transmissions since in typical indoor environments 80% or more of the traffic is on the downlink. We do not mix downlink and uplink traffic in one slot, because scheduling downlink or uplink traffic together helps reduce channel estimation overhead as shown

in [69].

6.2.1 PHY-layer model

Assume there are M access points (APs) in one cluster, which cooperatively serve K users. We denote the number of antenna elements on the m^{th} AP by $N_{t,m}$ and the number of antenna elements on the k^{th} user by N_r. The user set is denoted by $\mathcal{K} = \{1, \ldots, K\}$. Let $N_t = \sum_{m=1}^{M} N_{t,m}$ be the total numbers of antennas at the AP side. The matrix of complex channel gains between the cooperative APs and the antennas of the k^{th} user is denoted by $\boldsymbol{H}_k \in \mathbb{C}^{N_r \times N_t}$. The data vector $\boldsymbol{x} = \left[\boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_K^T\right]^T$ is jointly precoded by the M APs using the precoding matrix $\boldsymbol{V} = [\boldsymbol{V}_1, \ldots, \boldsymbol{V}_K]$. $\boldsymbol{x}_k \in \mathbb{C}^{N_r}$ is the transmit signal vector for receiver k, and $\boldsymbol{x}_k$ is assumed to be independently encoded Gaussian codebook symbols with $\mathbb{E}[\boldsymbol{x}_k \boldsymbol{x}_k^\dagger] = \boldsymbol{I}$, where $(\cdot)^\dagger$ is the conjugate transpose of $(\cdot)$. 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. $\boldsymbol{V}_k \in \mathbb{C}^{N_t \times N_r}$ is the partition of $\boldsymbol{V}$ applied at the APs to precode the signals of user k.

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

$$\boldsymbol{y}_k = \boldsymbol{H}_k \boldsymbol{V}_k \boldsymbol{x}_k + \sum_{l=1, l \neq k}^{K} \boldsymbol{H}_k \boldsymbol{V}_l \boldsymbol{x}_l + \boldsymbol{n}_k \ , \tag{6.1}$$

where $\boldsymbol{n}_k$ is the vector of Gaussian noise at the k^{th} user with covariance matrix $\sigma_k^2 \boldsymbol{I}$. Assume the received signal is equalized using the linear receive filter $\boldsymbol{U}_k \in \mathbb{C}^{N_{r,k} \times N_{r,k}}$. The received signal of the k^{th} receiver is given by $\hat{\boldsymbol{x}}_k = \boldsymbol{U}_k^\dagger \boldsymbol{y}_k$.

6.2.2 Time-variant MIMO channel model

To characterize the channels of mobile users, we follow the geometry-based stochastic channel modelling approach used in the generic WINNER II channel model [24, 88]. The physical parameters are determined in a stochastic manner based on the statistical distributions extracted from measurements. It is applicable for wireless systems operating at

2-6 GHz with up to 100 MHz bandwidth [88].

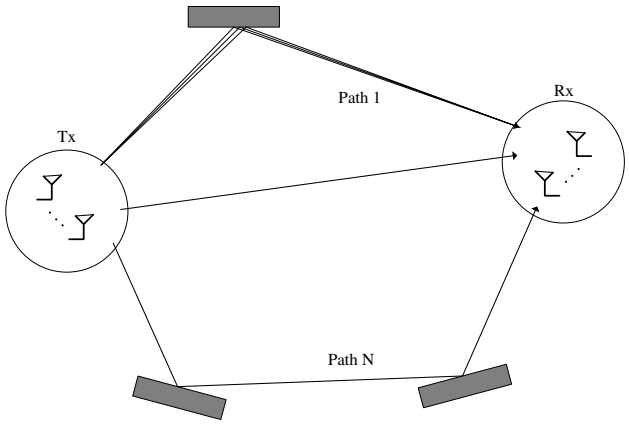

Figure 6.1: The multi-ray MIMO propagation channel

The impulse response of the channel is

$$\boldsymbol{H}(t,\tau) = \sum_{n=1}^{N} \boldsymbol{H}_n(t;\tau) \ ,$$

where N is the number of dominant scattering clusters, each of which is constituted by a number of rays, as shown in Figure 6.1. Assume there are M_n rays from the n^{th} scattering cluster. The channel from TX antenna element s to RX antenna element t for scattering cluster n

$$\begin{aligned} H_{u,s,n}(t;\tau) = & \sum_{m=1}^{M_n} \alpha_{n,m} \text{exp}(\text{j}2\pi\lambda_0^{-1}(\overrightarrow{\varphi_{m,n}} \cdot \overrightarrow{r_{rx,u}})) \\ & \times \text{exp}(\text{j}2\pi\lambda_0^{-1}(\overrightarrow{\phi_{m,n}} \cdot \overrightarrow{r_{rx,u}})) \\ & \times \text{exp}(j2\pi\upsilon_{m,n}t)\delta(\tau - \tau_{m,n}) \end{aligned}$$

where λ_0 is the wavelength of carrier frequency and $a_{n,m}$ is the complex gain of the n^{th} ray from the m^{th} scattering cluster. $\overrightarrow{\varphi_{n,m}}$ and $\overrightarrow{\phi_{n,m}}$ are the AoA and AoD unit vector respectively. $\overrightarrow{r_{rx,u}}$ and $\overrightarrow{r_{tx,s}}$ are the location vectors of receive antenna element u and transmit antenna element s. $\upsilon_{n,m}$ is the Doppler frequency component of the n^{th} ray from the m^{th} scattering cluster.

For time-variant channels, the aforementioned small scale parameters, such as AOD,

AOA and propagation delay, are time variance, i.e., function of t. To model these propagation parameters that vary over time, time evolution with smooth transitions between two quasi-stationary periods are discussed in [88].

6.2.3 MIMO link scheduling problem

In the targeted dense environment, there are many users competing for limited resources. Therefore, MIMO link scheduling that can achieve high throughput while maintaining fairness is an essential requirement.

Potential aggregate throughput

The achievable data rates of MU-MIMO users depend on the concurrent user group and the corresponding MIMO weights (precoders and combiners). There are $\sum_{i=1}^{N_t} \binom{K}{i}$ possible user groups, also referred as communication sets. Assume a certain communication set $\Pi = \{\pi_1, \pi_2, \ldots, \pi_I\}$ for concurrent transmission with I users. The data rate of user π_k in Π is given by

$$r_{\pi_k} = \log_2 \left| \boldsymbol{I} + (\boldsymbol{U}_{\pi_k}^{\dagger} \boldsymbol{R}_{\pi_k} \boldsymbol{U}_{\pi_k})^{-1} \boldsymbol{U}_{\pi_k}^{\dagger} \boldsymbol{H}_{\pi_k} \boldsymbol{V}_{\pi_k} \boldsymbol{V}_{\pi_k}^{\dagger} \boldsymbol{H}_{\pi_k}^{\dagger} \boldsymbol{U}_{\pi_k} \right| \ . \tag{6.2}$$

where $\boldsymbol{R}_{\pi_k}$ is the corresponding covariance matrix of the received interference plus noise is given by

$$\boldsymbol{R}_{\pi_k} = \sum_{l \in \Pi, l \neq \pi_k} \boldsymbol{H}_{\pi_k} \boldsymbol{V}_l \boldsymbol{V}_l^{\dagger} \boldsymbol{H}_{\pi_k}^{\dagger} + \sigma_{\pi_k}^2 \boldsymbol{I} \ . \tag{6.3}$$

As we assume the use of explicit CSI feedback, the central controller has the access to the compressed feedback of channel matrix $\boldsymbol{H}_k, \forall k$, which is given by $\hat{\boldsymbol{H}}_{\boldsymbol{k}} = \hat{\boldsymbol{S}}_k \hat{\boldsymbol{B}}_k^{\dagger}$, where $\hat{\boldsymbol{S}}_k$ and $\hat{\boldsymbol{B}}_k$ are the quantized diagonal matrix containing non-zero singular values and the right singular matrix of $\boldsymbol{H}_k$. With the limited CSI feedback, the central controller lacks the ability to calculate the combiner. Instead, as discussed in Chapter 4, it can compute the composed combiner $\tilde{\boldsymbol{U}}_k$ and estimate the user performance, where $\tilde{\boldsymbol{U}}_k = \boldsymbol{A}_k^{\dagger} \boldsymbol{U}_k$ and $\boldsymbol{A}_k$ is

the left singular matrix of $\boldsymbol{H}_k$. With calculated precoders and combiners, the SINR of the n^{th} stream of the k^{th} user in communication set Π is evaluated at the transmitter side is

$$\gamma_{k,n} = \frac{\tilde{\boldsymbol{u}}^{\dagger}_{\pi_k,n}\hat{\boldsymbol{H}}_{\pi_k}\boldsymbol{v}_{\pi_k,n}\boldsymbol{v}^{\dagger}_{\pi_k,n}\hat{\boldsymbol{H}}^{\dagger}_{\pi_k}\tilde{\boldsymbol{u}}_{\pi_k,n}}{\tilde{\boldsymbol{u}}^{\dagger}_{\pi_k,n}(\sigma_k^2\boldsymbol{I} + \sum\limits_{(\pi_{k'},j)\neq(\pi_k,n)}\hat{\boldsymbol{H}}_{\pi_k}\boldsymbol{v}_{\pi_{k'},j}\boldsymbol{v}^{\dagger}_{\pi_{k'},j}\hat{\boldsymbol{H}}^{\dagger}_{\pi_k})\tilde{\boldsymbol{u}}_{\pi_k,n}},$$

where $\tilde{\boldsymbol{u}}_{k,n}$ and $\boldsymbol{v}_{k,n}$ are the n^{th} column of $\tilde{\boldsymbol{U}}_k$ and $\boldsymbol{V}_k$. Then, the bit-rates can be determined via the rate selection process discussed in Section 2.3.4.

Scheduling problem description

Our focus is on building a fair scheduler for a single cluster with M cooperative APs and K users. Let $\mathcal{T} = \{t_1, \ldots, t_T\}$ be the scheduling period composed of T time slots of equal duration, $\boldsymbol{\Pi}_j = \{\pi_{1,j}, \ldots, \pi_{I_j,j}\}$ be the communication set scheduled in time slot t_j with I_j active users, and $\boldsymbol{r}_j = [r_{1,j}, \ldots, r_{K,j}]^T$ be the bit-rates of users in time slot t_j, where $r_{k,j} = 0$ if $k \notin \boldsymbol{\Pi}_j$. For a scheduling period $\mathcal{T}$, we need to schedule a communication set for each time slot that maximizes the throughput while satisfying a fairness constraint. Mathematically, it can be formulated as follows:

$$\begin{aligned} &\max\nolimits_{\{\boldsymbol{\Pi}_j\}_{j=1}^{T}} && \sum_{j=1}^{T}\sum_{k=1}^{K} r_{k,j} \\ &s.t. && \sum_{j=1}^{T} r_{k,j} = b_k \sum_{j=1}^{T}\sum_{k=1}^{K} r_{k,j} \end{aligned} \tag{6.4}$$

The fairness constraints require that each user achieves a bandwidth that is proportional to its target bandwidth share b_k. For example, the target bandwidth vector $\boldsymbol{b}$ can represent the QoS ratios of the competing users. In this paper, we are particularly interested in achieving time-based fairness, which has been shown in [89] to substantially improve the throughput compared to rate-based fairness in multi-rate WLANs. In [80], the idea of time-based fairness is extended to interfering MIMO channels. Following the idea in [80], the target bandwidth fraction of user k can be set to $b_k = \rho_k / \sum_{k=1}^{K}\rho_k$, where ρ_k is the

interference-free data rate of user k. These time-fair b_k's are used in the simulation results of Section 6.4.

6.3 Fair MIMO Link Scheduling Algorithm Using Mobility Hints

Stationary users' channels can be stable for hundreds of milliseconds or even longer. For these users, the scheduler can rely on a CSI measurement to remain valid over multiple communication slots. The mobile users, however, require more frequent CSI updates to capture the channel variations. Therefore, it is inefficient to schedule the stationary and mobile users together, especially for a large user population with only a few mobile users. To resolve this problem, we incorporate mobility awareness into our proposed scheduling algorithm. Since the mobility only affects the performance of mobile users and does not affect stationary users during downlink transmission [84], it is possible to enhance the MU-MIMO performance by separating the stationary and mobile users into different time slots.

6.3.1 High-level operation of proposed scheduling framework

The operational flow of the proposed scheduling framework is shown in Figure 6.2. The CU tracks CSI over time and uses it to classify the users into stationary and mobile groups. The number of time slots reserved for the two user groups in each scheduling period, denoted by T_s and T_m, are adaptively adjusted based on the fairness criterion and achieved bandwidth (discussed in Section 6.3.2). The scheduler first calculates an overall schedule for T_s time slots, including only stationary users. Upon completion of the stationary users' transmission, the scheduler executes a per-slot scheduling strategy based on fresh CSI of mobile users, measured for each slot. The detailed scheduling algorithm is elaborated in Section 6.3.3.

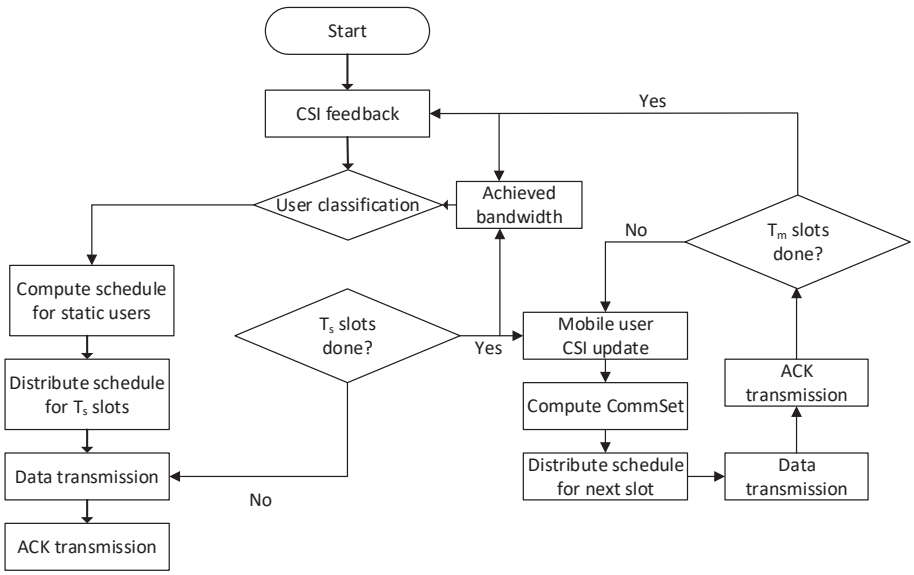

Figure 6.2: High-level flow chart of the mobility-aware scheduling framework

6.3.2 User mobility classification

To categorize stationary and mobile users, we track the CSI of each user across multiple measurements and identify the channels of stationary users based on CSI similarity. We propose to use subspace collinearity as a metric of CSI similarity. Subspace collinarity is a criterion that reflects the similarity between two matrix subspaces. In general, given two matrices $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$, their subspace collinearity can be represented as

$$col(\boldsymbol{M}_1, \boldsymbol{M}_2) = 1 - \frac{abs(tr(\boldsymbol{M}_1 \boldsymbol{M}_2^{\dagger}))}{||\boldsymbol{M}_1||_F ||\boldsymbol{M}_2||_F}$$

The value of subspace collinearity varies from 0 to 1. A larger collinearity indicates a lower similarity of the two matrix subspaces.

Let $\tilde{\boldsymbol{B}}_k(t)$ and $\tilde{\boldsymbol{B}}_k(t-\Delta t)$ be the feedback right singular value of the k^{th} user's channel at time instants t and $t-\Delta t$. The similarity between consecutive CSI values is estimated by $f_s(k,t) = col\left(\tilde{\boldsymbol{B}}_k(t), \tilde{\boldsymbol{B}}_k(t-\Delta t)\right)$. For each user, we maintain a moving average of

the CSI similarity to track the channel variation as follows:

$$\mathcal{S}(k,t) = \ (1-\beta_k)\mathcal{S}(k,t-\Delta t) + \beta_k f_s(k,t) \ .$$

If the value of $\mathcal{S}(k,t)$ for user k is smaller than a predefined threshold, user k is declared as a stationary user. Therefore, the stationary and mobile user groups are updated accordingly after each channel sounding stage.

Let $\mathcal{U}_s$ and $\mathcal{U}_m$ be the user sets containing stationary and mobile users, respectively. To maintain the fairness between the two user groups, the schedule duration portions reserved for stationary and mobile users should be proportional to their target bandwidth portions by factoring in their achieved bandwidth, which is given by:

$$\frac{T_s}{T_m} = \frac{\sum\limits_{i\in\mathcal{U}_s} b_i \exp(1-u_i/b_i)}{\sum\limits_{i\in\mathcal{U}_m} b_i \exp(1-u_i/b_i)} \ .$$

where T_s and T_m are the number of time slots to accommodate stationary and mobile users, respectively, which are adjusted upon the completion of each entire round of communications based on the achieved bandwidth portion $u_i = \bar{R}_i / \sum\limits_{i=1}^{K} \bar{R}_i$ with $\bar{R}_i$ representing the average achieved throughput of the i^{th} user. Without loss of generality, we assume $T = T_s + T_m$ is the number of time slots within one entire scheduling period. The objective of the adjustment is to roughly maintain a good fairness between stationary and mobile users. The fairness among each specific user group will be guaranteed by the proposed scheduler for each user group.

6.3.3 Calculating a schedule

For MU-MIMO transmission, the performance of the scheduler is largely dependent on the choice of communication sets and their MIMO weights. For the targeted dense environment, there are typically a large number of users and it is, therefore, computationally

prohibitive to explore all possible user combinations.

To balance the aggregate performance and processing overhead, the proposed scheduler works differently for stationary and mobile users. For stationary users, the scheduler calculates a number of high-performance communication sets and corresponding MIMO weights intensively and combines them into a schedule that maximizes throughput and satisfies the target fairness among stationary users. Compared to stationary users, mobile users are much more sensitive to stale CSI. The scheduler for mobile users requires frequent CSI update to accommodate channel variations. A general idea is to calculate a "good" communication set for each time slot with updated CSI and run a low-complexity MIMO weight calculation algorithm.

Scheduling stationary users

$\mathcal{U}_s$ is the stationary user set to be scheduled over a scheduling period $\mathcal{T}_s$ having T_s time slots. The CSI values of the stationary users are updated and expected to be stable for the period of $\mathcal{T}_s$. After collecting the CSI for stationary users, the CU first generates a number of high-performance communication sets and their corresponding MIMO weights and then schedules the communication sets over the slots in $\mathcal{T}_s$, as shown in Figure 6.2. With stationary channels, the scheduler for stationary users can fully reap the benefits of AP cooperation by performing a fairly expensive optimization procedure to produce the schedule and MIMO weights.

The schedule for the static users can be calculated by the proposed approaches in chapter 5. In particular, we use the two-stage method in Chapter 5 in the simulation, since it has lower computational complexity for a medium to large size user population. To be specific, we use an iterative algorithm to generate communication sets. In each iteration, we solve a weighted sum rate maximization problem. Then, the user weights are updated according to the previously generated communication sets. The user weight update procedure is designed to aid the scheduler in achieving the target fairness criterion. With generated

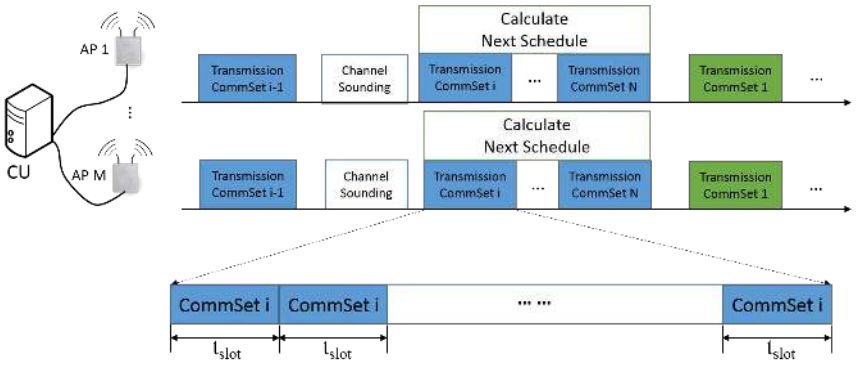

Figure 6.3: Flow of operations for AP cooperation with proposed scheduling framework

communication sets, the scheduler calculates the number of time slots assigned to each communication set that achieves maximum throughput while guaranteeing the target fairness among stationary users, as also proposed in [84].

Stationary users are less sensitive to processing overhead. If processing overhead becomes too high in certain scenarios, e.g. if the number of subcarrier groups is large, the CU can use parallel processing to speed up schedule calculation. In Section 6.4, we demonstrate that the computation time of the stationary scheduling procedure is small enough to achieve large throughput gains in practical scenarios.

Scheduling mobile users

Unlike stationary users, mobile users require more frequent CSI feedback to accommodate the channel variations caused by environmental changes and/or user mobility. The scheduling approach for stationary users is no longer suitable for mobile users, since the performance of mobile users will degrade as the CSI becomes outdated. Thus, mobile users cannot afford an intensive schedule calculation. Here, we need to develop a more efficient scheduling approach to improve CSI timeliness and accuracy.

Recall that there are T_m time slots, also referred to as *mobile slots*, reserved for $|\mathcal{U}_m|$ mobile users as discussed in Section III-B. The objective of scheduling mobile users within T_m mobile slots is to maximize their aggregate performance while meeting the fairness

requirements. Let $b_k^m = \frac{b_k}{\sum_{k\in\mathcal{U}_m} b_k}$ be the normalized target bandwidth portion for mobile user k. When scheduling mobile time slot $t+1$, we use $R_{k,t}$, which is the achieved sum-rate of user k during the first t slots. We have $R_{k,t} = R_{k,t-1} + r_{k,t}$. Let $\mu_{k,t} = \frac{R_{k,t}}{\sum_{k\in\mathcal{U}_m} R_{k,t}}$ be the achieved bandwidth of mobile user k during the first t mobile slots with $\mu_{k,0} = 1, \forall k \in \mathcal{U}_m$ as the initial value. Since the CSI information for mobile users is updated for each time slot, the scheduling problem can be solved for each time slot in sequence. Ultimately, we aim to approach the fairness constraint in problem (6.4) for mobile users, which can be rewritten as:

$$R_{k,t} = b_k^m \sum_{k\in\mathcal{U}_m} R_{k,t}, \forall k \in \mathcal{U}_m \tag{6.5}$$

Assuming the equality constraint is satisfied at time slot t, we have

$$\begin{aligned} & R_{k,t} - r_{k,t} = u_{k,t-1}(R_{k,t}/b_k^m - \sum_{k\in\mathcal{U}_m} r_{k,t}) \\ \Longrightarrow \quad & r_{k,t} = (1 - u_{k,t-1}/b_k^m)R_{k,t} + u_k \sum_{k\in\mathcal{U}_m} r_{k,t} \\ & \geq (1 - u_{k,t-1}/b_k^m)R_{k,t} \ . \end{aligned}$$

Thus, the sum rate maximization can be approached by solving a weighted sum rate maximization problem for each time slot, i.e., $\max \ \sum_{k\in\mathcal{U}_m} w_{k,t}R_{k,t}$, where $w_k \propto 1 - u_{k,t-1}/b_k^m$. s_k indicates that larger weights are assigned to users that are below their target bandwidth proportions when considering the previous $t-1$ time slots. Therefore, we can update the user weights as follows:

$$w_{k,t} = \max\left(1 - u_{k,t-1}/b_k^m, 0\right) \ . \tag{6.6}$$

Thus, any user that is at or above its desired bandwidth proportion is assigned with zero weight and is therefore excluded from the current round of transmission. To speed up the processing overhead of mobile users, a simple and computationally efficient precoding approach is utilized, namely, block diagonalization (BD). The optimization problem for

time slot t can be formulated as:

$$\begin{aligned} \max \quad & \sum_{k \in \mathcal{U}_m} w_{k,t} \log_2 \left| \boldsymbol{I} + \boldsymbol{R}_k^{-1} \boldsymbol{H}_k \boldsymbol{F}_k \boldsymbol{F}_k^{\dagger} \boldsymbol{H}_k^{\dagger} \right| \\ s.t. \quad & \sum_{k \in \mathcal{U}_m} Tr(\boldsymbol{\Gamma}_m \boldsymbol{F}_k \boldsymbol{F}_k^{\dagger}) \leq P_m, m = 1, \ldots, M \\ & \boldsymbol{H}_l \boldsymbol{F}_k = \boldsymbol{0}, l, k \in \mathcal{U}_m, l \neq k \ . \end{aligned} \tag{6.7}$$

The diagonal matrix $\boldsymbol{\Gamma}_m \in \mathbb{R}^{N_t \times N_t}$ is introduced for each AP to select the partition of $\boldsymbol{F}_k$ 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.

The maximum number of users in one slot is $\lceil N_t/N_r \rceil$. To reduce computational overhead, we select the $\lceil N_t/N_r \rceil$ users with highest weights for each time slot. The BD precoder can then be designed using QR decomposition with water-filling power loading, as analyzed in Chapter 3.

6.4 Simulation Results

We conduct simulations of our proposed scheduling algorithm using the WINNER II channel model for indoor office environments [24]. We uniformly distribute M APs and K users in a circular region with a radius of 50 meters. We set each AP to have 4 transmit antennas and each client to have 2 receive antennas. The noise power is -85 dBm and the transmit power of each AP is 23 dBm. Unless otherwise specified, we consider downlink transmission with $M = 3$ cooperative APs.

For comparison, we also consider the following schedulers:

- **Per-slot scheduler**: This is a scheduling algorithm that generates a communication set by solving a WSRM problem for each time slot. To meet the fairness requirement, the user weights are updated using (6.6) after the transmission of each time slot. The CSI values are assumed to be updated for each time slot $\tau_{slot} = 5\ ms$. The

performance of the basic per-slot scheduler is computed without accounting for the overhead of CSI feedback and processing overhead. This then forms an upper bound on the performance of other schedulers since it optimizes for each time slot and incurs zero overhead. The per-slot scheduler that accounts for CSI feedback and processing overhead is also evaluated and that algorithm is denoted by **Per-slot***.

- **One-shot scheduler**: This is a scheduling algorithm proposed in [84] for a completely static environment. In this paper, we implement this algorithm by treating all users as if they were stationary. Therefore, the channel variation of mobile users within one entire scheduling period will cause performance loss for this algorithm.

- **Conventional TDMA**: This is a basic time-fair TDMA scheduling algorithm, where the MIMO links are scheduled sequentially in a round robin manner. In other words, there is only one user scheduled in each time slot cooperatively served by M APs. The SU-MIMO transmission within each time slot can achieve the interference-free data rates using the optimal SVD MIMO weights.

To evaluate the achieved fairness, we use the fairness index proposed in [80],

$$FI(\boldsymbol{u}, \boldsymbol{b}) = \exp\left(-\sum_{k=1}^{K} \left| \ln(u_k / b_k) \right| / K \right), \tag{6.8}$$

where u_k is the fraction of bandwidth allocated to the k^{th} user. The fairness index given by (6.8) takes values in $[0, 1]$, with 1 representing perfect fairness among users.

6.4.1 Evaluation of CSI feedback overhead

We first evaluate the CSI feedback overhead of the proposed scheme by comparing with the conventional scheme without user classification. The percentage of mobile users is denoted by p_m. Figure 6.4 shows the CSI collection time versus the number of clients within for a period of 1 second. Without user classification, the CSI update period t_{fd} is identical for

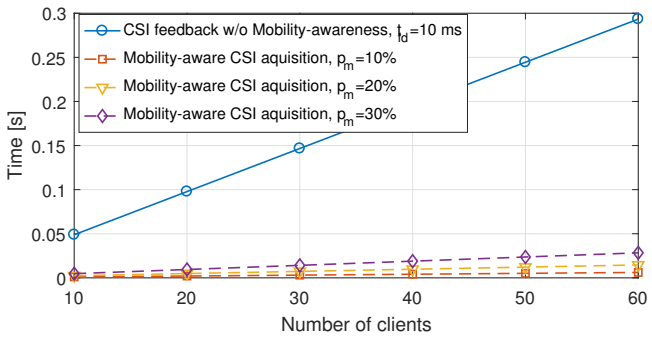

Figure 6.4: CSI collection time versus the number of clients within $T = 1\ s$.

all users. To guarantee the CSI accuracy of mobile users, the updated period is set to every 10 msec. In this case, the CSI collection time increases rapidly as the number of users increases. For example, it takes more than 0.3 seconds for CSI feedback for 50 users or more, which can overwhelm the data transmission time. The CSI feedback overhead will further scale up with the increase of subcarriers/subbands. By taking advantage of user classification, we can significantly lower the CSI update frequency for stationary users, since their channels can be stable for up to several seconds. In Figure 6.4, the CSI update period for stationary users is set to 1 second while for mobile users it remains at 10 msec. In this case, we can largely reduce the CSI feedback overhead, while guaranteeing the same CSI accuracy for the mobile users as the conventional scheme. Therefore, the mobility-aware scheme is a promising approach in terms of reducing CSI feedback for the scenarios with limited-mobility. For example, with 30% mobile users, the CSI collection requires about 0.03 seconds within a period of 1 second for 60 users.

6.4.2 Evaluation of user classification

In Figure 6.5, the cumulative distributions of the CSI similarity for stationary and mobile users are plotted using our subspace collinearity metric. The consecutive CSI samples are collected every 1 second. Clearly, the subspace collinearity metric is a reliable indicator to distinguish stationary and mobile users. For stationary users, the CSI similarity is very close to zero, while the mobile users generate much higher CSI similarity values. Based on

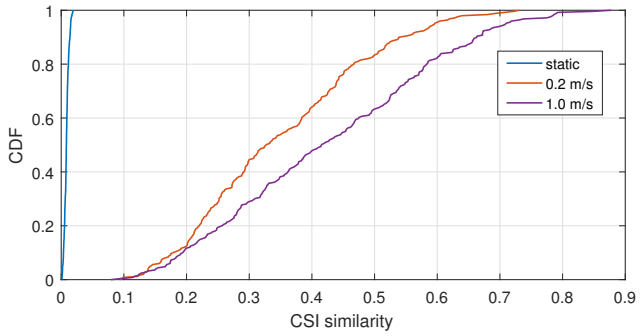

Figure 6.5: CDF of the CSI similarity.

these results, we set the similarity threshold $Thr = 0.05$ so that any $Thr > 0.05$ causes a user to be classified as mobile.

6.4.3 Evaluation of throughput and fairness

In Figure 6.6, we evaluate the throughput and fairness performances of the different scheduling algorithms versus the mobile user percentage. The average speed of mobile users is set to 1 m/s. The **per-slot scheduler** provides highest throughput and good fairness, because it neglects the CSI feedback and processing overheads. However, in practice, the intensive CSI feedback for the per-slot update would significantly reduce the data transmission time and lower the achievable throughput. Moreover, the CSI delay caused by the per-slot processing overhead for a large user population (e.g., $K = 45$) would also introduce large fairness loss, especially for mobile users.

Both CSI feedback and processing overhead are accounted for with **per-slot***, as well as with our proposed mobility-aware scheduling algorithm. With user classification, we are able to reduce the CSI overhead for stationary users, while reducing the processing overhead for the mobile users. Besides, the adaptive adjustment of the time slot assignment produces a good fairness among stationary and mobile users. Therefore, our proposed scheduling scheme achieves 25%-35% higher throughput than that of **per-slot***. The one-shot scheduler cannot meet the fairness requirement because the CSI for mobile users become outdated for data transmission. The conventional TDMA schedules a single user

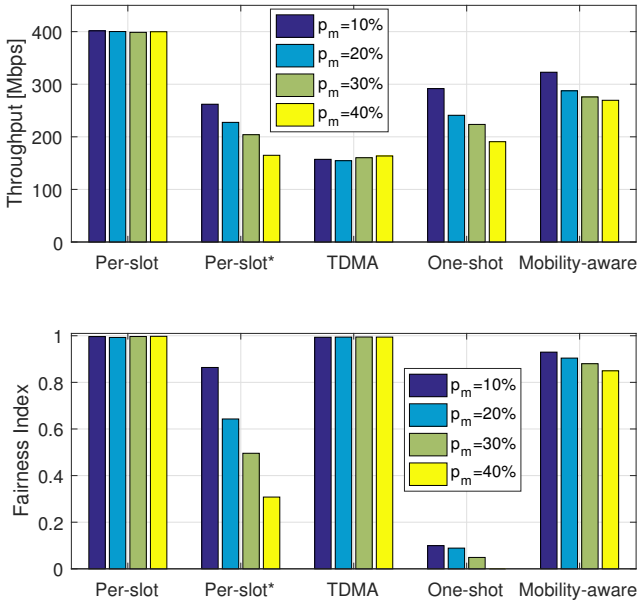

Figure 6.6: Throughout and fairness versus mobile user percentage for $K = 45$.

for each time slot and guarantees perfect fairness among all users but it fails to exploit the multi-user MIMO gain promised by AP cooperation. Thus, it can only achieve about 60% of the throughput of our proposed scheme.

In Figure 6.7, the achieved throughput and fairness performance is plotted as a function of the number of users. The mobile user percentage is fixed to 20% for all cases. The performance provided by per-slot scheduler without considering the CSI and processing overhead is deemed as the upper bound. With the increase of user numbers, the achievable throughput and fairness of per-slot* scheduler experience sharp decreases. By separating stationary and mobile users and highly optimizing stationary users, we are able to improve throughput and maintain good fairness. The performance of our algorithm actually increases with a large number of users getting to within about 20% of the upper bound throughput and achieving fairness of greater than 0.9.

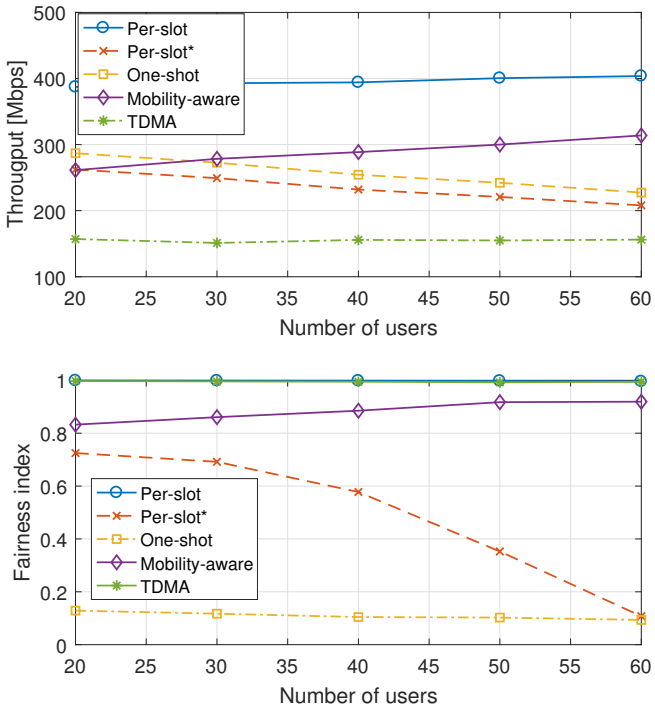

Figure 6.7: Throughout and fairness versus mobile user percentage for $K = 45$.

6.5 Chapter Summary

In this chapter, we presented a mobility-aware MIMO link scheduling scheme for a cluster of cooperative APs within a dense wireless network. The proposed approach tracks the user channel variation and separates stationary and mobile users into different time slots. Based on the characteristics of stationary and mobile users, different scheduling strategies are applied. On the one hand, for stationary users with slow-varying channels, we combine a set of pre-calculated high-performance communication sets into a high-throughput and fair schedule with sparse CSI update. On the other hand, the performance of mobile users are improved by utilizing timely updated CSI to produce a good communication set in an efficient way for each time slot. In the presence of limited mobility, our approach exhibits strong performance gains compared to conventional approaches that do not separate mobile and stationary users.

CHAPTER 7
CONCLUSIONS

7.1 Conclusions

With rapid proliferation of both access point and wireless devices, the wireless performance issues have been emerging for several years with respect to overall performance. However, simplily adding more APs does not scale well with the network size due to the sharing of limited unlicensed spectrum bands. AP cooperation together with advanced MIMO processing techniques are deemed as the key to break the performance bottleneck of wireless networks within unlicensed band. Different from traditional WLANs, a number of APs operating on the same frequency spectrum forms a cluster and cooperatively serve the clients by sharing the lower-layer parameters.

In this thesis, we aimed to establish the foundations for the realization of the WLANs with clustered coordinated-APs as we envisioned to improve the both individual and overall performance in dense environments. The contributions in each chapter are summarized as follows:

- In chapter 3, we proposed a novel user selection algorithm for block diagonalization. The proposed approach can store multiple high-performance user groups in a binary tree, which reduces the probability of dropping good user groups compared to conventional greedy methods. Moreover, our approach allows flexible adjustments of the number of stored user groups, so that permits tradeoffs between the computational cost and sum-rate performance. Additionally, our approach can further reduce the complexity for partially varying environments by reusing the pre-calculated information and performing fast update. The 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.

- In chapter 4, we proposed a combined user selection and MIMO weights optimization approach for solving a general weighted sum rate maximization (WSRM) problem. An novel pre-processing step that incorporates multiple decision factors is first developed to eliminate some undesired users and reduce the size of the input to the WSRM problem. Then, a modified WSRM algorithm is performed to determine the MIMO weights and further refine the user selection. The proposed approach was shown to outperform previous approaches while having significantly lower running time and better scalability for a moderate to large number of users.

- In chapter 5, we considered a specific problem for user scheduling to achieve high aggregate performance while maintaining fairness, which can operate across a small group of APs and employ multiuser MIMO. We first provide the mathematical formulation of a maximum throughput scheduling problem with fairness constraints in the multi-AP MIMO setting. We proposed alternative scheduling algorithms, that are alternating optimization method and two-stage method, to tackle the formulated problem. The alternating optimization method jointly optimizes the MIMO weights and user selection over one entire scheduling period. The two-stage method separates the optimization into two phases: (1) generating a number of high-performance multiuser communication sets and (2) calculating an overall schedule to determine the time slot assignment. The alternating optimization algorithm produces significantly higher aggregate throughput with a running time that is practical for a small user population, while the two-stage algorithm produces close aggregate throughput while having significantly lower running time.

- In chapter 6, we proposed a mobility-aware MIMO link scheduling scheme for a cluster of cooperative APs. The proposed scheme performs a user differentiation procedure to separate stationary and mobile users into different time slots and ap-

plies different scheduling strategies. For stationary users with static or slow-varying channels, a set of pre-calculated high-performance communication sets are incorporated into a high-throughput and fair schedule with sparse CSI update. For mobile users, the performance are improved by utilizing timely updated CSI to produce a good communication set in an efficient way for each time slot. In the presence of limited mobility, our approach outperforms conventional approaches without using the mobilty hints, due to a better balance of protocol overhead and aggregate performance.

The theoretical analysis and simulation results provided in this thesis lay out the foundation for the realization of the high-performance WLAN networks with clustered cooperative APs.

7.2 Future Work

Throughout this thesis, we focus on optimizing the performance of a single cluster with the aid of cooperation between a small number of APs. While the single cluster with a limited number of cooperative APs can be built on our prior research, there will be unaccounted-for interference from neighboring clusters that can drive down the per-cluster performance. In our future research, we will study the approaches that can scale our proposed per-cluster solutions to operate across a large enterprise network. Simply applying the solutions beyond a small number of cooperative APs is challenging due to the overheads associated with the channel measurements, CSI exchange and schedule computation. Therefore, we will investigate hierarchical cooperation approaches that can scale our proposed solution to a large number of APs.

Since the inter-cluster interference mainly affects the performance of edge nodes, which are near the boundaries of two adjacent clusters, we aim to improve the performance of these edge nodes and treat the interference to the far-away nodes as noise. We will investigate loose coordination schemes across neighboring clusters. First, we will study the

uplink/downlink transmission alignment across neighboring clusters to avoid the strong client-to-client interference between two nearby edge nodes. The basic idea is to ensure the alignment of downlink transmissions of the interfering clusters. Second, to further improve the performance of edge nodes, we will allow the exchange of the schedule between neighboring clusters. These clusters can negotiate and adjust their schedule to avoid the simultaneous transmission of highly interfering communication sets. Finally, we will investigate the methods for partial recomputation of the schedule by taking into account the inter-cluster interference on edge nodes.

7.3 Publications

As part of the research conducted in this dissertation, we have written several documents that are either published, submitted, or in progress as follows:

- M. Ge and D. M. Blough, "High Throughput and Fair Scheduling for Multi-AP Multiuser MIMO in Dense Wireless Networks," submitted to IEEE Transactions on Networking, 2018.
- M. Ge and D. M. Blough, "Mobility-aware multi-user MIMO link scheduling for AP cooperation," accepted by IEEE International Conference on Communications (ICC), 2018.
- M. Ge and D. M. Blough,"PBUS: Efficient User Selection for Block Diagonalization in Dense Wireless Networks," in Proceeding of IEEE Global Communications Conference (Globecom), 2017.
- M. Ge and D. M. Blough, "High-Throughput and Fair Scheduling for Access Point Cooperation in Dense Wireless Networks," in Proceeding of IEEE Wireless Communications and Networking Conference (WCNC), 2017.
- M. Ge, J. R. Barry, and D. M. Blough, "Combined User Selection and MIMO Weight

Calculation for AP Cooperation in Dense Wireless Networks," in Proceeding of IEEE Wireless Communications and Networking Conference (WCNC), 2017.

REFERENCES

[1] B. Bellalta, "Ieee 802.11ax: High-efficiency WLANs," *IEEE Wireless Commun.*, vol. 23, no. 1, pp. 38–46, 2016.

[2] S. Gollakota, S. D. Perli, and D. Katabi, "Interference alignment and cancellation," in *ACM SIGCOMM*, vol. 39, 2009, pp. 159–170.

[3] H. S. Rahul, S. Kumar, and D. Katabi, "JMB: Scaling wireless capacity with user demands," in *Proc. ACM SIGCOMM*, 2012, pp. 235–246.

[4] S. Kumar, D. Cifuentes, S. Gollakota, and D. Katabi, "Bringing cross-layer MIMO to today's wireless lans," in *Proc. ACM SIGCOMM*, vol. 43, 2013, pp. 387–398.

[5] X. Zhang, K. Sundaresan, M. A. A. Khojastepour, S. Rangarajan, and K. G. Shin, "NEMOx: Scalable network MIMO for wireless networks," in *Proc. Int'l. Conf. Mobile Comput. & Netw.*, 2013, pp. 453–464.

[6] H. Yu, O. Bejarano, and L. Zhong, "Combating inter-cell interference in 802.11 ac-based multi-user MIMO networks," in *Proc. Annual Int'l. Conf. Mobile Comput. Netw.*, 2014, pp. 141–152.

[7] G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless Pers. Commun.*, vol. 6, no. 3, pp. 311–335, 1998.

[8] E. Telatar, "Capacity of multi-antenna Gaussian channels," *Trans. Emerg. Telecommun. Technol.*, vol. 10, no. 6, pp. 585–595, 1999.

[9] D. Gesbert, M. Kountouris, R. W. H. Jr., C. b. Chae, and T. Salzer, "Shifting the MIMO paradigm," *IEEE Signal Process. Mag.*, vol. 24, no. 5, pp. 36–46, 2007.

[10] O. Bejarano, E. W. Knightly, and M. Park, "IEEE 802.11 ac: From channelization to multi-user MIMO," *IEEE Commun. Mag.*, vol. 51, no. 10, pp. 84–90, 2013.

[11] L. Liu, R. Chen, S. Geirhofer, K. Sayana, Z. Shi, and Y. Zhou, "Downlink MIMO in LTE-advanced: SU-MIMO vs. MU-MIMO," *IEEE Commun. Mag.*, vol. 50, no. 2, pp. 140–147, 2012.

[12] A. Ghosh, R. Ratasuk, B. Mondal, N. Mangalvedhe, and T. Thomas, "LTE-advanced: Next-generation wireless broadband technology," *IEEE Wireless Commun.*, vol. 17, no. 3, pp. 10–22, 2010.

[13] C. Lim, T. Yoo, B. Clerckx, B. Lee, and B. Shim, "Recent trend of multiuser MIMO in LTE-advanced," *IEEE Commun. Mag.*, vol. 51, no. 3, pp. 127–135, 2013.

[14] R. Liao, B. Bellalta, M. Oliver, and Z. Niu, "MU-MIMO MAC protocols for wireless local area networks: A survey," *IEEE Commun. Surveys & Tuts.*, vol. 18, no. 1, pp. 162–183, 2016.

[15] M. X. Gong, B. Hart, and S. Mao, "Advanced wireless LAN technologies: IEEE 802.11 ac and beyond," *GetMobile: Mobile Comput. Commun.*, vol. 18, no. 4, pp. 48–52, 2015.

[16] S. L. Loyka, "Channel capacity of MIMO architecture using the exponential correlation matrix," *IEEE Commun. Lett.*, vol. 5, no. 9, pp. 369–371, 2001.

[17] B. Holter, "On the capacity of the MIMO channel: A tutorial introduction," in *Proc. IEEE Norwegian Symp. Signal Process.*, 2001, pp. 167–172.

[18] R. B. Ertel, P. Cardieri, K. W. Sowerby, T. S. Rappaport, and J. H. Reed, "Overview of spatial channel models for antenna array communication systems," *IEEE Pers. Commun.*, vol. 5, no. 1, pp. 10–22, 1998.

[19] M. Debbah and R. R. Muller, "MIMO channel modeling and the principle of maximum entropy," *IEEE Trans. Inf. Theory*, vol. 51, no. 5, pp. 1667–1690, 2005.

[20] P. Almers, E. Bonek, A Burr, N. Czink, M. Debbah, V. Degli-Esposti, H. Hofstetter, P. Kyösti, D. Laurenson, G. Matz, *et al.*, "Survey of channel and radio propagation models for wireless MIMO systems," *EURASIP J. Wireless Commun. and Netw.*, vol. 2007, no. 1, pp. 56–56, 2007.

[21] O. Stabler and R. Hoppe, "MIMO channel capacity computed with 3D ray tracing model," in *Proc. EuCAP*, 2009, pp. 2271–2275.

[22] J. W. Wallace and M. A. Jensen, "Modeling the indoor MIMO wireless channel," *IEEE Trans. Antennas Propag.*, vol. 50, no. 5, pp. 591–599, 2002.

[23] C.-C. Chong, C.-M. Tan, D. I. Laurenson, S. McLaughlin, M. A. Beach, and A. R. Nix, "A new statistical wideband spatio-temporal channel model for 5-GHz band WLAN systems," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 2, pp. 139–150, 2003.

[24] J. Meinilä, P. Kyösti, T. Jämsä, and L. Hentilä, "WINNER II channel models," *Radio Technol. Concepts for IMT-Advanced*, pp. 39–92, 2009.

[25] J.-P. Kermoal, L. Schumacher, K. I. Pedersen, P. E. Mogensen, and F. Frederiksen, "A stochastic mimo radio channel model with experimental validation," *IEEE J. Sel. Areas Commun.*, vol. 20, no. 6, pp. 1211–1226, 2002.

[26] D. McNamara, M. Beach, P. Fletcher, and P Karlsson, "Initial investigation of multiple-input multiple-output (mimo) channels in indoor environments," in *Proc. IEEE Symp. Commun. Veh. Technol.*, IEEE, 2000, pp. 139–143.

[27] W. Weichselberger, M. Herdin, H. Ozcelik, and E. Bonek, "A stochastic MIMO channel model with joint correlation of both link ends," *IEEE Trans. Wireless Commun.*, vol. 5, no. 1, pp. 90–100, 2006.

[28] A. F. Molisch, *Wireless Communications*. John Wiley & Sons, 2012, vol. 34.

[29] D. Gesbert, M. Shafi, D.-s. Shiu, P. J. Smith, and A. Naguib, "From theory to practice: An overview of MIMO space-time coded wireless systems," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 3, pp. 281–302, 2003.

[30] G. Caire and S. Shamai, "On achievable rates in a multi-antenna broadcast downlink," in *Proc. Annu. Allerton Conf. Commun. Control Comput.*, 2000, pp. 1188–1193.

[31] W. Yu and J. M. Cioffi, "Trellis precoding for the broadcast channel," in *Proc. IEEE GLOBECOM*, vol. 2, 2001, pp. 1344–1348.

[32] H. Weingarten, Y. Steinberg, and S. S. Shamai, "The capacity region of the Gaussian multiple-input multiple-output broadcast channel," *IEEE Trans. Inf. Theory*, vol. 52, no. 9, pp. 3936–3964, 2006.

[33] S. Vishwanath, N. Jindal, and A. Goldsmith, "Duality, achievable rates, and sum-rate capacity of gaussian MIMO broadcast channels," *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2658–2668, 2003.

[34] 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, 2006.

[35] 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, 2004.

[36] A. Bayesteh and A. K. Khandani, "On the user selection for MIMO broadcast channels," *IEEE Trans. Inf. Theory*, vol. 54, no. 3, pp. 1086–1107, 2008.

[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*, 2015, pp. 451–459.

[38] X. Xie and X. Zhang, "Scalable user selection for MU-MIMO networks," in *Proc. IEEE INFOCOM*, 2014, pp. 808–816.

[39] 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*, 2007, pp. 3295–3300.

[40] B. Bandemer, M. Haardt, and S. Visuri, "Linear MMSE multi-user MIMO downlink precoding for users with multiple antennas," in *Proc. IEEE PIMRC*, IEEE, 2006, pp. 1–5.

[41] A. Tolli, M. Codreanu, and M. Juntti, "Minimum SINR maximization for multiuser mimo downlink with per bs power constraints," in *Proc. IEEE WCNC*, 2007, pp. 1144–1149.

[42] S. W. Peters and R. W. Heath, "Cooperative algorithms for MIMO interference channels," *IEEE Trans. Veh. Tech.*, vol. 60, no. 1, pp. 206–218, 2011.

[43] R. Irmer, H. Droste, P. Marsch, M. Grieger, G. Fettweis, S. Brueck, H. Mayer, L. Thiele, and V. Jungnickel, "Coordinated multipoint: Concepts, performance, and field trial results," *IEEE Commun.*, vol. 49, no. 2, pp. 102–111, 2011.

[44] K. Hosseini, W. Yu, and R. S. Adve, "Large-scale MIMO versus network MIMO for multicell interference mitigation," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 930–941, 2014.

[45] H. Huang, M. Trivellato, A. Hottinen, M. Shafi, P. J. Smith, and R. Valenzuela, "Increasing downlink cellular throughput with limited network MIMO coordination," *IEEE Trans. Wireless Commun.*, vol. 8, no. 6, 2009.

[46] J. Hoydis, M. Kobayashi, and M. Debbah, "On the optimal number of cooperative base stations in network MIMO," *IEEE Trans. Signal Process.*, 2009.

[47] 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.

[48] E. Hamed, H. Rahul, M. A. Abdelghany, and D. Katabi, "Real-time distributed MIMO systems," in *Proc. ACM SIGCOMM*, 2016, pp. 412–425.

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

[50] "IEEE draft standard for IT - telecommunications and information exchange between systems - LAN/MAN - specific requirements - part 11: Wireless LAN medium access control and physical layer specifications - amd 4: Enhancements for very high

throughput for operation in bands below 6GHz," *IEEE P802.11ac/D6.0, July 2013*, pp. 1–446, 2013.

[51] S. Biaz and S. Wu, "Rate adaptation algorithms for IEEE 802.11 networks: A survey and comparison," in *Proc. IEEE ISCC*, 2008, pp. 130–136.

[52] M. Lacage, M. H. Manshaei, and T. Turletti, "Ieee 802.11 rate adaptation: A practical approach," in *Proc. ACM MSWiM*, 2004, pp. 126–134.

[53] J. C. Bicket, "Bit-rate selection in wireless networks," PhD thesis, Massachusetts Institute of Technology, 2005.

[54] D. Xia, J. Hart, and Q. Fu, "Evaluation of the minstrel rate adaptation algorithm in ieee 802.11 g WLANs," in *Proc. ICC*, IEEE, 2013, pp. 2223–2228.

[55] J. Fang, K. Tan, Y. Zhang, S. Chen, L. Shi, J. Zhang, Y. Zhang, and Z. Tan, "Fine-grained channel access in wireless LAN," *IEEE/ACM Trans. Netw.*, vol. 21, no. 3, pp. 772–787, 2013.

[56] V. Garg, *Wireless communications & networking*. Morgan Kaufmann, 2010.

[57] C. F. Shih, B. Krishnaswamy, and R. Sivakumar, "Rhythm: Achieving scheduled WiFi using purely distributed contention in WLANs," in *Proc. IEEE GLOBECOM*, 2015, pp. 1–7.

[58] 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.

[59] 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, 2006.

[60] 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.

[61] 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, 2008.

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

[63] 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. Inf. Theory*, vol. 51, no. 4, pp. 1570–1580, 2005.

[64] S. You, L. Chen, and Y. Liu, "Convex-concave procedure for weighted sum-rate maximization in a MIMO interference network," in *Proc. IEEE GLOBECOM*, 2014, pp. 4060–4065.

[65] S. Shim, J. S. Kwak, R. Heath, and J. Andrews, "Block diagonalization for multi-user MIMO with other-cell interference," *IEEE Trans. Wireless Commun*, vol. 7, no. 7, pp. 2671–2681, 2008.

[66] D. Nguyen and T. Le-Ngoc, "Sum-rate maximization in the multicell MIMO broadcast channel with interference coordination," *IEEE Trans. Signal Process.*, vol. 62, no. 6, pp. 1501–1513, 2014.

[67] S. Christensen, R. Agarwal, E. Carvalho, and J. Cioffi, "Weighted sum-rate maximization using weighted MMSE for MIMO-BC beamforming design," *IEEE Trans. Wireless Commun.*, vol. 7, no. 12, pp. 4792–4799, 2008.

[68] J. Escudero Garzas, M. Hong, A. Garcia, and A. Garcia-Armada, "Interference pricing mechanism for downlink multicell coordinated beamforming," *IEEE Trans. Commun.*, vol. 62, no. 6, pp. 1871–1883, 2014.

[69] L. Cortes-Pena and D. Blough, "MIMO link scheduling for interference suppression in dense wireless networks," in *Proc. IEEE WCNC*, 2015, pp. 1225–1230.

[70] L. Cortes-Pena, J. Barry, and D. 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, 2015.

[71] F. Negro, S. Shenoy, I. Ghauri, and D. Slock, "On the MIMO interference channel," in *Proc. Inf. Theory Appl. Workshop*, 2010, pp. 1–9.

[72] 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, 2001.

[73] R. Bohnke and K.-D. Kammeyer, "Weighted sum rate maximization for the MIMO-downlink using a projected conjugate gradient algorithm," in *Proc. IEEE IWCLD*, 2007, pp. 82–85.

[74] L. Li, M. Pal, and Y. R. Yang, "Proportional fairness in multi-rate wireless LANs," in *Proc. IEEE INFOCOM*, 2008, pp. 1004–1012.

[75] W. Li, S. Wang, Y. Cui, X. Cheng, R. Xin, M. A. Al-Rodhaan, and A. Al-Dhelaan, "AP association for proportional fairness in multirate WLANs," *IEEE/ACM Trans. Net.*, vol. 22, no. 1, pp. 191–202, 2014.

[76] V. R. Cadambe and S. A. Jafar, "Interference alignment and degrees of freedom of the k-user interference channel," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3425–3441, 2008.

[77] K. Gomadam, V. R. Cadambe, and S. A. Jafar, "Approaching the capacity of wireless networks through distributed interference alignment," in *Proc. IEEE GLOBECOM*, 2008, pp. 1–6.

[78] ——, "A distributed numerical approach to interference alignment and applications to wireless interference networks," *IEEE Trans. Inf. Theory*, vol. 57, no. 6, pp. 3309–3322, 2011.

[79] G. Tan and J. Guttag, "Time-based fairness improves performance in multi-rate WLANs," in *Proc. USENIX Conf.*, 2004, pp. 269–282.

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

[81] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Performance anomaly of 802.11b," in *Proc. IEEE INFOCOM*, vol. 2, 2003, pp. 836–843.

[82] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[83] M. Ge and S. Wang, "Fast optimal resource allocation is possible for multiuser OFDM-based cognitive radio networks with heterogeneous services," *IEEE Trans. Wireless Comm.*, vol. 11, no. 4, pp. 1500–1509, 2012.

[84] M. Ge and D. M. Blough, "High-throughput and fair scheduling for access point cooperation in dense wireless networks," in *Proc. IEEE WCNC*, 2017, pp. 1–6.

[85] S. Byeon, K. Yoon, O. Lee, S. Choi, W. Cho, and S. Oh, "MoFA: Mobility-aware frame aggregation in Wi-Fi," in *Proc. ACM Emerg. Netw. Experiments Technol.*, 2014, pp. 41–52.

[86] L. Sun, S. Sen, and D. Koutsonikolas, "Bringing mobility-awareness to WLANs using PHY layer information," in *Proc. Emerg. Netw. Experiments Technol.*, ACM, 2014, pp. 53–66.

[87] S. Byeon, K. Yoon, C. Yang, and S. Choi, "STRALE: Mobility-aware PHY rate and frame aggregation length adaptation in WLANs," in *Proc. IEEE INFOCOM*, 2017, pp. 1–9.

[88] Y. d. J. Bultitude and T. Rautiainen, *IST-4-027756 WINNER II D1. 1.2 V1. 2 WINNER II channel models*, 2007.

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