Modeling, Analysis, and Real-Time Design of Many-Antenna MIMO Networks

Yongce Chen

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Electrical Engineering

Y. Thomas Hou, Chair
Richard M. Buehrer
Lingjia Liu
Wenjing Lou
Jeffrey H. Reed

August 12, 2021
Blacksburg, Virginia

Keywords: Wireless communications, 5G, MIMO, many antennas, degree of freedom, resource scheduling, real time, GPU.

Modeling, Analysis, and Real-Time Design of Many-Antenna MIMO Networks

Yongce Chen

(ABSTRACT)

Among the many advances and innovations in wireless technologies over the past twenty years, MIMO is perhaps among the most successful. MIMO technology has been evolving over the past two decades. Today, the number of antennas equipped at a base station (BS) or an access point (AP) is increasing, which forms what we call "many-antenna" MIMO systems. Many-antenna MIMO will have significant impacts on modern wireless communications, as it will allow numerous wireless applications to operate on the vastly underexplored mid-band and high-band spectrum and is able to deliver ultra-high throughput.

Although there are considerable efforts on many-antenna MIMO systems, most of them came from physical (PHY) layer information-theoretic exploitation. There is a lack of investigation of many-antenna MIMO from a networking perspective. On the other hand, new knowledge and understanding begin to emerge at the PHY layer, such as the rank-deficient channel phenomenon. This calls for new theories and models for many-antenna MIMO in a networking environment. In addition, the problem space for many-antenna MIMO systems is much broader and more challenging than conventional MIMO. Reusing existing solutions designed for conventional MIMO systems may suffer from inferior performance or require excessive computation time.

The goal of this dissertation is to advance many-antenna MIMO techniques for networking research. We focus on the following two critical areas in the context of many-antenna MIMO networks: (i) DoF-based modeling and (ii) real-time optimization. This dissertation consists

of two parts that study these two areas. In the first part, we aim to develop new DoF models and theories under general channel rank conditions for many-antenna MIMO networks, and we explored efficient DoF allocation based on our new DoF model. The main contributions of this part are summarized as follows.

- **New DoF models and theories under general channel rank conditions:** Existing DoF-based models in networking community assume that the channel matrix is of full rank. However, this assumption no longer holds when the number of antennas becomes many and the propagation environment is not ideal. In this study, we develop a novel DoF model under general channel rank conditions. In particular, we find that for IC, shared DoF consumption at both transmit and receive nodes is most efficient for DoF allocation, which is contrary to existing unilateral IC models based on full-rank channel assumption. Further, we show that existing DoF models under the full-rank assumption are a special case of our generalized DoF model. The findings of this study pave the way for future research of many-antenna networks under general channel rank conditions.

- **Efficient DoF utilization for MIMO networks:** We observes that, in addition to the fact that channel is not full-rank, the strength of signals on different directions in the eigenspace is extremely uneven. This offers us new opportunities to efficiently utilize DoFs in a MIMO network. In this study, we introduce a novel concept called "effective rank threshold". Based on this threshold, DoFs are consumed only to cancel strong interferences in the eigenspace while weak interferences are treated as noise in throughput calculation. To better understand the benefits of this approach, we study a fundamental trade-off between network throughput and effective rank threshold for an MU-MIMO network. Our simulation results show that network throughput under optimal rank threshold is significantly higher than that under existing DoF IC models.

In the second part, we offered real-time designs and implementations to solve many-antenna MIMO problems for 5G cellular systems. In addition to maximizing a specific optimization objective, we aim at offering a solution that can be implemented in sub-ms to meet requirements in 5G standards. The main contributions of this part are summarized as follows.

- **Turbo-HB—A novel design and implementation for ultra-fast hybrid beamforming:** We investigate the beamforming problem under hybrid beamforming (HB) architecture. A major practical challenge for HB is to obtain a solution in 500 μs, which is an extremely stringent but necessary time requirement for its deployment in the field. To address this challenge, we present Turbo-HB—a novel beamforming design under the HB architecture that can obtain the beamforming matrices in about 500 μs. The key ideas of Turbo-HB are two-fold. First, we develop low-complexity SVD by exploiting randomized SVD technique and leveraging channel sparsity at mmWave frequencies. Second, we accelerate the overall computation time through large-scale parallel computation on a commercial off-the-shelf (COTS) GPU platform, with special engineering efforts for matrix operations and minimized memory access. Experimental results show that Turbo-HB is able to obtain the beamforming matrices in 500 μs for an MU-MIMO cellular system while achieving similar or better throughput performance by those state-of-the-art algorithms.

- **mCore+—A sub-millisecond scheduler for 5G MU-MIMO systems:** We study a scheduling problem in a 5G NR environment. In 5G NR, an MU-MIMO scheduler needs to allocate RBs and assign MCS for each user at each TTI. In particular, multiple users may be co-scheduled on the same RB under MU-MIMO. In addition, the real-time requirement for determining a scheduling solution is at most 1 ms. In this study, we present a novel scheduler mCore+ that can meet the sub-ms real-time

requirement. mCore+ is designed through a multi-phase optimization, leveraging large-scale parallelism. In each phase, mCore+ either decomposes the optimization problem into a large number of independent sub-problems, or reduces the search space into a smaller but more promising subspace, or both. We implement mCore+ on a COTS GPU platform. Experimental results show that mCore+ can obtain a scheduling solution in $\sim$500 μs. Moreover, mCore+ can achieve better throughput performance than the state-of-the-art algorithms.

- **$\mathbf{M}^3$—A sub-millisecond scheduler for multi-cell MIMO networks under C-RAN architecture:** We investigate a scheduling problem for a multi-cell environment. Under Cloud Radio Access Network (C-RAN) architecture, the signal processing can be performed cooperatively for multiple cells at a centralized baseband unit (BBU) pool. However, a new resource scheduler is needed to jointly determine RB allocation, MCS assignment, and beamforming matrices for all users under multiple cells. In addition, we aim at finding a scheduling solution within each TTI (i.e., at most 1 ms) to conform to the frame structure defined by 5G NR. To do this, we propose $\mathbf{M}^3$—a GPU-based real-time scheduler for a multi-cell MIMO system. $\mathbf{M}^3$ is developed through a novel multi-pipeline design that exploits large-scale parallelism. Under this design, one pipeline performs a sequence of operations for cell-edge users to explore joint transmission, and in parallel, the other pipeline is for cell-center users to explore MU-MIMO transmission. For validation, we implement $\mathbf{M}^3$ on a COTS GPU. We showed that $\mathbf{M}^3$ can find a scheduling solution within 1 ms for all tested cases, while it can significantly increase user throughput by leveraging joint transmission among neighboring cells.

Modeling, Analysis, and Real-Time Design of Many-Antenna MIMO Networks

Yongce Chen

(GENERAL AUDIENCE ABSTRACT)

MIMO is widely considered to be a major breakthrough in modern wireless communications. MIMO comes in different forms. For conventional MIMO, the number of antennas at a base station (BS) or access point (AP) is typically small (< 8). Today, the number of antennas at a BS/AP is typically ranging from 8 to 64 when the carrier frequency is below 24 GHz. When the carrier frequency is above 24 GHz (e.g., mmWave), the number of antennas can be even larger (> 64). We call today's MIMO systems (typically with ≥ 8 antennas at some nodes) as "many-antenna" MIMO systems, and this will be the focus of this dissertation.

Although there exists a considerable amount of works on many-antenna MIMO techniques, most efforts focus on physical (PHY) layer for information-theoretic exploitation. There is a lack of investigation on how to efficiently and effectively utilize many-antenna MIMO from a networking perspective.

The goal of this dissertation is to advance many-antenna MIMO techniques for networking research. We focus on the following two critical areas in the context of many-antenna MIMO networks: (i) degree-of-freedom (DoF)–based modeling and (ii) real-time optimization. In the first part, we investigate a novel DoF model under general channel rank conditions for many-antenna MIMO networks. The main contributions of this part are summarized as follows.

- **New DoF models and theories under general channel rank conditions:** In this study, we develop a novel DoF model under general channel rank conditions. We show that existing works claiming that unilateral DoF consumption is optimal no longer hold when channel rank is deficient (not full-rank). We find that for IC, shared DoF consumption at both Tx and Rx nodes is the most efficient scheme for DoF allocation.

- **Efficient DoF utilization for MIMO networks:** In this study, we proposed a new approach to efficiently utilize DoFs in a MIMO network. The DoFs used to cancel interference are conserved by exploiting the interference signal strength in the eigenspace. Our simulation results show that network throughput under our approach is significantly higher than that under existing DoF IC models.

In the second part, we offer real-time designs and implementations to solve many-antenna MIMO problems for 5G cellular systems. The timing performance of these designs is tested in actual wall-clock time.

- **A novel design and implementation for ultra-fast hybrid beamforming:** We investigate a beamforming problem under the hybrid beamforming (HB) architecture. We propose Turbo-HB—a novel beamforming design under the HB architecture that can obtain the beamforming matrices in about 500 μs. At the same time, Turbo-HB can achieve similar or better throughput performance by those state-of-the-art algorithms.

- **A sub-millisecond scheduler for 5G multi-user (MU)-MIMO systems:** We study a resource scheduling problem in 5G NR. We present a novel scheduler called mCore+ that can schedule time-frequency resources to MU-MIMO users and meet the $\sim$500 μs real-time requirement in 5G NR.

- **A sub-millisecond scheduler for multi-cell MIMO networks under C-RAN architecture:** We investigate the scheduling problem for a multi-cell environment under a centralized architecture. We present $\mathbf{M}^3$—a GPU-based real-time scheduler that jointly determines a scheduling solution among multiple cells. $\mathbf{M}^3$ can find the scheduling solution within 1 ms.

Dedication

To my beloved parents.

Acknowledgments

It would have been a much more difficult journey for me to get my Ph.D. degree without the help of many around me. I owe too many people for their unconditional support and encouragement.

First and foremost, I would like to give my deepest gratitude to my advisor, Prof. Tom Hou, the Bradley Distinguished Professor of ECE at Virginia Tech. Throughout my entire Ph.D. study in the past five years, Prof. Hou has provided me with valuable guidance, support, and confidence in me and for my research. He leaves me a lifetime unforgettable memory of diligence, patience, and erudition. He has spent days and nights going through my papers word by word, continually training me on how to think critically and write logically. He always encouraged me to practice my English speaking skills and presentation skills. It would be impossible for me to make great improvements in communication and presentation skills without his enlightening instructions and encouragements. I sincerely appreciate that he has offered so many precious opportunities to cultivate my professional skills. He suggested and sponsored me to attend a 4-day CUDA training program in Calgary, Canada. He also supported me for many academic conference trips, both domestically and overseas. Those experiences are critical for Ph.D. study and will profoundly benefit my career development beyond Ph.D. I have learned so much in writing this dissertation with the guidance and help from Prof. Hou. This is a great treasure that I will cherish not only in my future career but also in my whole life.

My deepest gratitude extends to the rest of my committee members: Prof. Wenjing Lou, Prof. Jeffrey Reed, Prof. Michael Buehrer, and Prof. Lingjia Liu for making valuable comments on this dissertation. Their insightful feedback has helped me improve this dissertation

in every respect.

My sincere thanks also go to my previous colleagues Dr. Xiaoqi Qin and Dr. Yan Huang in the CNSR Lab. They were my first mentors on how to start my Ph.D. study and life in a foreign country back in August 2016. I wish to thank my current labmates Dr. Qingyu Liu, Shaoran Li, Chengzhang Li, Yubo Wu, Shiva Acharya, Shabnam Wahed and Naru Jai. Through our numerous meetings and discussions, they gave me ingenious suggestions and sharpened my thoughts on my research problems. Thanks should also go to CNSR visiting scholars Yue Wang, Wei Teng and Yan Han for their friendship and support. I will always remember our time together sharing life experiences. The years of 2020 and 2021 are very challenging—we had to spend most of our time physically apart due to the pandemic of COVID-19. But CNSR members stayed together at heart. It is their care and support that helped me go through the final stage of my Ph.D. journey.

I am also grateful to my friends Ashish Agrawal, Jack Hartley, Teng Hu, Lisi Jiang, Yin Liu, Yuyan Liu and Wei Wang. During the past five years pursuing my Ph.D. degree, I have experienced negative emotions from time to time such as frustration, loneliness and anxiety. It is these awesome people who always trusted in me and encouraged me to keep going.

Last but not least, I want to express my deepest gratitude to my parents for their unconditional support and love. They always worked diligently and passionately to provide a better life for me in every possible way. They understood and supported my decision to study and live in a foreign country that is thousands of miles away. It would be impossible for me to have this dissertation without their unconditional love. I must also thank other family members for their unwavering understanding and assistance. There aren't enough words for me to express how much I owe to my family.

Funding Acknowledgments

This research was supported in part by the National Science Foundation (NSF) under Grants CNS-1343222, CNS-1443889, CNS-1617634, CNS-1642873 and CNS-1800650, Virginia Commonwealth Cyber Initiative (CCI), and the NVIDIA AI Lab (NVAIL) in Santa Clara, CA for its unrestricted gift and equipment donation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Objective

Among the many advances and innovations in wireless technologies over the past 20 years, MIMO is perhaps among the most successful. In the commercial sector, MIMO is the core technology in wireless standards such as Wi-Fi (802.11n [1], IEEE 802.11ac [2]) and cellular (4G LTE [3], 5G NR [4]). In the research community, MIMO continues to be a centerpiece of wireless communications and networking.

The paradigms and applications of MIMO have been evolving. One of the most noticeable evolutions is that the numbers of antennas at a base station (BS), an access point (AP), and a mobile device, are continuously increasing. For conventional MIMO, the number of antennas at a BS/AP is typically small (< 8) and the number of antennas at a user device is even fewer. Today, the number of antennas at a BS/AP typically ranges from 8 to 64 when the carrier frequency is below 24 GHz. When the carrier frequency is above 24 GHz (e.g., mmWave), the number of antennas can be even larger (> 64). We call today's MIMO systems (with ≥ 8 antennas) as "many-antenna" MIMO systems. Many-antenna MIMO allows numerous wireless applications to operate on the vastly underexplored mid-band and high-band spectrum and is able to deliver ultra-high throughput.

Although many-antenna MIMO is critical for high-performance wireless networks, most

research efforts focus on physical (PHY) layer studies for information-theoretic exploitation (e.g., [5, 6, 27, 34, 35, 72, 80, 81, 82, 84, 101, 162, 163]). There is a lack of investigation of many-antenna MIMO from a networking perspective. On the other hand, new knowledge and understanding begin to emerge at the PHY layer, such as the rank-deficient channel phenomenon. This calls for new theories and models for many-antenna MIMO in a networking environment. In addition, the problem space for many-antenna MIMO systems is much broader and more challenging than conventional MIMO. Reusing existing solutions designed for conventional MIMO systems may suffer from inferior performance or require excessive computation time.

To make a concrete step towards advancing many-antenna MIMO technologies for networking research, this dissertation identifies and focuses on the following two areas: (i) Degree-of-Freedom (DoF)–based modeling and (ii) real-time optimization. Our motivation and the limitations of existing works in these areas are summarized as follows:

(i) *DoF-based Modeling for Many-Antenna MIMO Networks.* DoF based models have become very popular in the research community for modeling, analysis, and optimization of MIMO networks [14, 15, 16, 17, 18, 19, 20, 21, 22, 50, 51, 52]. Due to their simple abstraction of MIMO's capabilities in spatial multiplexing (SM) and interference cancellation (IC) [10, 11, 23, 60, 61], a DoF-based model can be used for resource allocation for SM and IC, with simple "+/-" arithmetic calculations. By avoiding complex matrix manipulation in resource allocation, DoF-based models are powerful and tractable tools to analyze MIMO's behavior in a network setting. A common characteristic among existing DoF-based models is that they all assume the channel matrix is of full rank (see, e.g., [14, 15, 16, 17, 18, 19, 20, 21, 22, 50, 51, 52]). This assumption is mostly valid for conventional MIMO (with a small number of antennas) in the rich-scattering environment. But when the number of antennas becomes many

and the propagation environment is not ideal (e.g., lack of rich scattering or presence of key-hole effect [30, 31]), this assumption no longer holds. As expected, a rank-deficient channel will hinder many-antenna MIMO's SM capability. Further, it undermines the viability of existing DoF-based IC models. Although channel rank deficiency has been recognized and studied for many-antenna MIMO [5, 6, 27, 34, 35, 72, 73], those efforts have been mainly at the PHY layer. Little progress has been made so far for networking research. As a result, there is hardly any result available on how to address rank deficiency in the context of DoF models for many-antenna MIMO networks.

(ii) *Real-Time Optimization for Many-Antenna Cellular Networks.* For practical MIMO systems, the available time to compute an optimal (or near-optimal) solution to a scheduling problem can be very limited. In particular, the allowed computation time is constrained by the physical properties of wireless channels. For example, the channel coherence time for mmWave systems can be as short as 1 ms for a mobile user moving at a speed of 20 km/h. Therefore, a practical beamforming solution must be offered within $\sim$500 μs (i.e, half of the channel coherence time) to be useful. Beyond channel coherence time, the beamforming solution may lead to poor performance due to the fast varying channel conditions. As a result, in modern cellular systems, the standardization bodies have imposed stringent timing requirements in their radio interface. For example, in 5G NR, one transmission time interval (TTI) is 1 ms under numerology 0 [91]. Since the radio resources are allocated in each TTI in 5G, the solution of resource allocation and beamforming matrices must be found within 1 ms to meet the requirement. To support ultra-low latency applications, an even shorter TTI may be needed (e.g., 500 μs under numerology 1). Such stringent timing requirement becomes a serious challenge for many-antenna MIMO systems, due to the extremely large solution space and high-dimensional matrix operations. On the other hand, most

existing research has been largely limited to asymptotic complexity analysis (i.e., in $O(\cdot)$). Although such complexity analysis is of interest from a theoretical perspective, it does not give any indication on how much actual time ("real-time") is needed when it is implemented on a given hardware platform. For a real-world cellular system, the benchmark is real-time performance (as measured in wall-clock time in terms of μs or ms), as there is a well-defined frame structure for data transmission.

The goal of this dissertation is to address the above challenges so as to advance research in many-antenna MIMO networks. Specifically, we aim to:

(i) *Develop new DoF models and theories under general channel rank conditions.* The presence of rank-deficient channels fundamentally changes the current understanding of DoF models for many-antenna MIMO networks. We aim to address this problem by developing a novel DoF model that can identify a feasible DoF region of any multi-link MIMO networks under general channel rank conditions. Further, we explore efficient DoF allocation based on our new DoF model.

(ii) *Offer real-time designs and implementations for 5G cellular networks.* In this dissertation, we focus on critical MIMO problems in modern cellular networks, such as hybrid beamforming, MU-MIMO scheduling, and joint transmission under C-RAN architecture. In addition to maximizing the optimization objective, we want to offer real-time (sub-ms) solutions so that they can be used in practice. We will pursue implementation on real-world hardware, so that we can measure their actual running time performance.

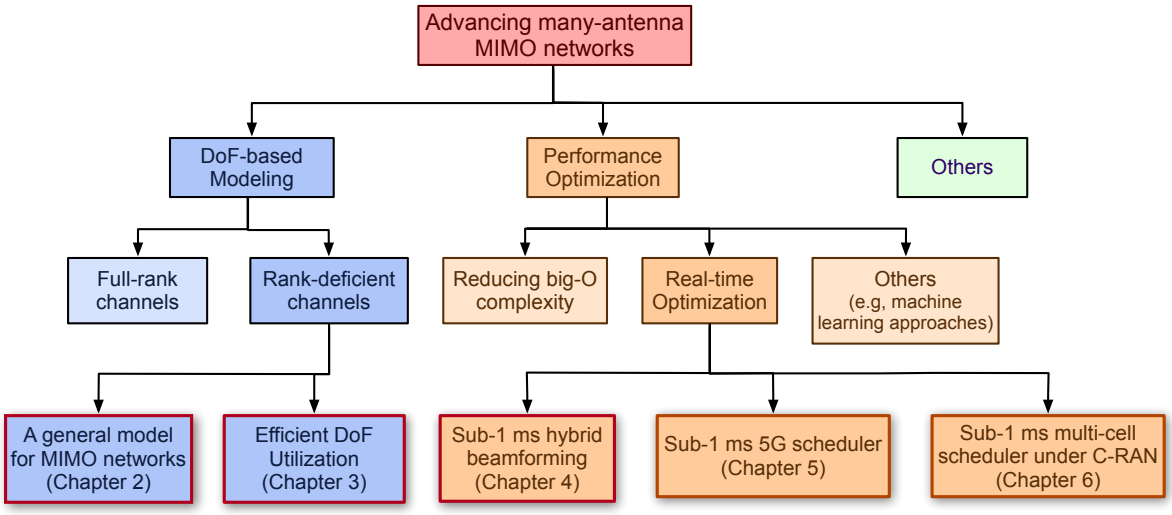


Figure 1.1: Dissertation structure.

1.2 Dissertation Outline and Contributions

The goal of this dissertation is to make a concrete step towards advancing many-antenna MIMO techniques in the networking research community. This dissertation consists of two parts. In the first part (Chapters 2 and 3), we investigate DoF models and theories and their utilization under general channel rank conditions. The second part (Chapters 4, 5 and 6) offers real-time designs and implementations for many-antenna MIMO problems. The structure of this dissertation is shown in Fig. 1.1. The main contributions of this dissertation are summarized as follows.

- In Chapter 2, we investigate the DoF-based model under general channel rank conditions. Existing DoF-based models in networking community assume that the channel matrix is of full rank. However, this assumption no longer holds when the number of antennas becomes many and the propagation environment is not ideal. In this chapter, we start with a fundamental understanding on how MIMO's DoFs are consumed at each node for SM and IC in the presence of rank-deficient channels. Based on this

understanding, we develop a DoF model that can be used for identifying a feasible DoF region of a multi-link MIMO network and for studying DoF scheduling in MIMO networks under general channel rank conditions. In particular, we find that for IC, shared DoF consumption at both transmit (Tx) and receive (Rx) nodes is critical for efficient DoF allocation. Further, we show that existing DoF models under the full-rank assumption become a special case of our generalized DoF model. Based on case studies, we show that the general IC model can achieve larger feasible DoF regions or improved objective values than existing unilateral IC models.

- In Chapter 3, we study DoF conservation in MIMO IC and exploit the difference in interference signal strength in the eigenspace. Chapter 2 addresses the problem of how DoFs should be allocated between Tx and Rx nodes to support SM and IC with given channel ranks. In this chapter, we address a parallel question on how to set channel ranks and efficiently utilize DoFs. We introduce a novel concept called "effective rank threshold" to differentiate signal strength on an interference link. Based on this threshold, DoFs are consumed only to cancel strong interfering signals in the eigenspace while weak interfering signals are treated as noise in throughput calculation. To better understand the benefits of this approach, we study a fundamental trade-off between network throughput and effective rank threshold for an MU-MIMO network. Our simulation results show that network throughput under optimal rank threshold is significantly higher than that under existing DoF IC models. To ensure the new DoF IC model is feasible at PHY layer, we propose an algorithm to set the weights for all nodes that can offer our desired DoF allocation.

- In Chapter 4, we focus on a beamforming problem under the hybrid beamforming (HB) architecture. A major practical challenge for HB is to obtain a solution in 500 μs, which is an extremely stringent but necessary time requirement for its deployment in the

field. We present Turbo-HB—a novel beamforming design under the HB architecture that can obtain the beamforming matrices in about 500 μs. The key ideas of Turbo-HB are two-fold. First, we identify the bottleneck of computation time is attributed to the high-dimensional SVD operations. Our design cuts down the computational complexity by utilizing randomized SVD technique and leveraging channel sparsity at mmWave frequencies. Second, we propose to accelerate the overall computation time through large-scale parallel computation on a commercial off-the-shelf (COTS) GPU platform. Our design incorporates a large number of matrix transformations and special engineering efforts such as minimized memory access. Experimental results show that Turbo-HB is able to obtain the beamforming matrices in 500 μs for an MU-MIMO cellular system while achieving similar or better throughput performance by those state-of-the-art algorithms.

- In Chapter 5, we study a scheduling problem for 5G MU-MIMO systems. Per 5G specifications, an MU-MIMO scheduler needs to determine RBs allocation and MCS assignment to each user for each TTI. Under MU-MIMO, multiple users may be co-scheduled on the same RB and each user may have multiple data streams simultaneously. In addition, the scheduler must meet the stringent real-time requirement (at most 1 ms) during decision making to be useful. We present mCore+—the first 5G MU-MIMO scheduler design and implementation that can meet the sub-ms real-time requirement. The key idea of mCore+ is to perform a multi-phase optimization, leveraging large-scale parallel computation. In each phase, mCore+ either decomposes the optimization problem into a number of independent sub-problems, or reduces the search space into a smaller but most promising subspace, or both. mCore+ is implemented and validated on a COTS GPU platform with meticulous engineering considerations. Experimental results show that mCore+ can offer a scheduling solution, as well as corresponding

beamforming matrices, in ∼500 μs for up to 100 RBs, 100 users, 29 MCS levels and 4×12 MIMO scheduling. Moreover, mCore+ can achieve better throughput performance than the state-of-the-art algorithms.

- In Chapter 6, we investigate a scheduling problem for a multi-cell MIMO system under C-RAN architecture. C-RAN is a novel centralized architecture for cellular networks, which can significantly improve spectrum efficiency by cooperative signal processing for multiple cells at a centralized baseband unit (BBU) pool. However, a new resource scheduler is needed before we can take advantage of C-RAN's multi-cell processing capability. The problem is how to jointly determine RB allocation, MCS assignment, and beamforming matrices for all users under all covering cells so that the PF objective can be maximized. In addition, the real-time requirement to determine a solution is 1 ms in order to conform to the frame structure defined by 5G NR. We propose $\mathbf{M}^3$—a GPU-based real-time scheduler for a multi-cell MIMO system. $\mathbf{M}^3$ exploits independency and parallelism through a multi-pipeline design. Specifically, $\mathbf{M}^3$ performs two independent parallel pipelines, where one pipeline performs a sequence of operations for cell-edge users to explore joint transmission, and in parallel, the other pipeline is for cell-center users to explore MU-MIMO transmission. We implemented $\mathbf{M}^3$ on a COTS Nvidia DGX Station. Through extensive experiments, we show that $\mathbf{M}^3$ can find a scheduling solution within 1 ms for all tested cases.

Chapter 2

A General Model for DoF-based Interference Cancellation with Rank-deficient Channels

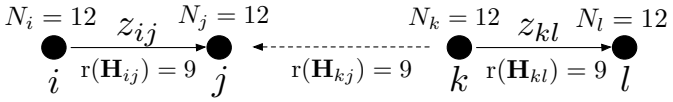
2.1 Introduction

Degree-of-freedom (DoF) models have become widely used to study MIMO network performance [9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 50]. The concept of DoF was first introduced by the information theory community (IT) to represent the multiplexing gain of a MIMO channel [10, 11, 12, 13]. It was then extended by the wireless networking community to characterize a node's spatial freedom. In particular, DoF-based models (e.g., [14, 15, 16, 17, 18, 19, 20, 21, 22, 50]) leverage DoFs at nodes to characterize MIMO's spatial multiplexing (SM) and interference cancellation (IC) capabilities. For IC, zero-forcing (ZF) precoding technique is used to create interference-free signals through beamforming in the null space of interference signals [23, 24]. Using DoF as a metric, the so-called DoF region can be used to characterize the performance envelope of SM for a set of links that transmit simultaneously (free of interference) [25, 26]. Although not without limitations, DoF-based models have served the wireless networking community well. By getting around complex matrix manipulation, it is a simple and tractable tool to analyze MIMO's SM and

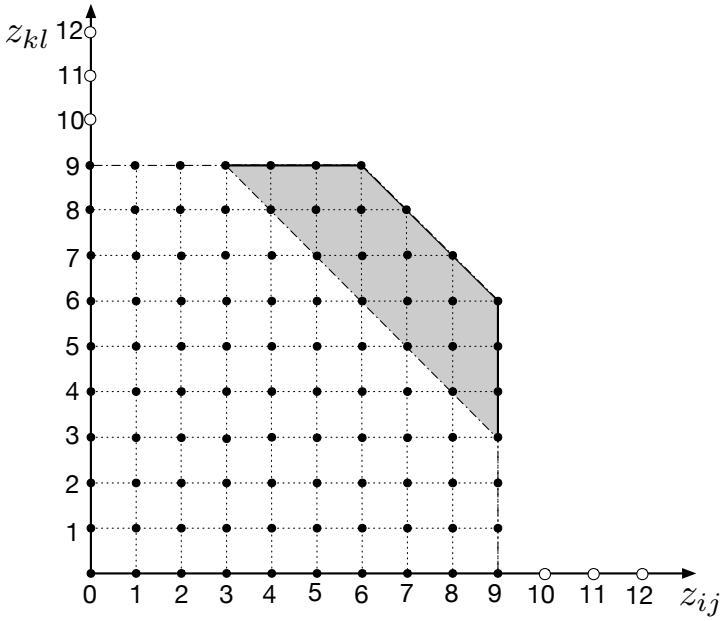
IC capability.

However, existing DoF-based models in the literature do suffer from one serious limitation. They assume the channel matrix is of full-rank (see, e.g., [14, 15, 16, 17, 18, 19, 20, 21, 22, 50]) which is typically one would encounter when the number of antennas is small and the propagation environment is ideal (i.e., rich scattering). But such an assumption quickly falls apart as the number of antennas increases and the propagation environment is not close to ideal (i.e., lack of rich scattering or presence of key-hole effect [27, 28, 29, 30, 31]). As expected, a rank-deficient channel will hinder MIMO's SM capability and undermine the validity of existing DoF-based IC models, which all assume full-rank channels. With FCC's recent interest in communications in midband spectrum (between 3.7 and 24 GHz) [32, 33], which is the spectrum where we expect to see *many-antenna* MIMO (typically ranges from 12 to 64), issues associated with rank deficiency will become even more critical and significant.

We use an example to illustrate issues with rank-deficient channels and motivate the need of our research in this chapter. Consider two active transmissions in Fig. 2.1(a), where Tx node i transmits z_{ij} data streams to Rx node j while Tx node k transmits z_{kl} data streams to Rx node l. Rx node j is interfered with by Tx node k. Suppose all the nodes have 12 antennas. Denote $\mathbf{H}_{ij}, \mathbf{H}_{kl}$ and $\mathbf{H}_{kj}$ as channel matrices of $i \to j$, $k \to l$ and $k \to j$, respectively and let the ranks of $\mathbf{H}_{ij}, \mathbf{H}_{kl}$ and $\mathbf{H}_{kj}$ all be 9 (< 12, i.e., rank-deficient). Under these rank-deficient channels, SM on links $i \to j$ and $k \to j$ are now each upper limited to 9 (instead of 12). So it is infeasible to have z_{ij} or z_{kl} to carry 12 data streams as under full-rank assumption. To find the DoF region of the two links (i.e., feasible data streams that can be carried on links $i \to j$ and $k \to j$ simultaneously), we need to consider how the interference (from Tx node k to Rx node j) is cancelled. It was well understood that for full-rank channels, DoF consumption for IC is most efficiently done by either Tx node



(a) An interference link between two active transmissions.



(b) DoF regions under different IC models

Figure 2.1: A motivating example showing different DoF regions for a two-link network.

k or Rx node j, but *not* both nodes. That is, either Rx node j (consuming z_{kl} DoFs) or Tx node k (consuming z_{ij} DoFs) can be used to cancel the interference from node k to j [15, 16, 17, 21, 22]. This will result in a DoF region that is bounded by the inner pentagon (dash lines) in Fig. 2.1(b). However, as we shall show in this chapter, such unilateral DoF consumption for IC (at either Tx node or Rx node, but not both) is *inefficient* for general rank-deficient channels. In fact, to maximize efficiency, DoF consumption must be shared between Tx node k and Rx node j to cancel the interference from k to j. We will show that through shared DoF consumption by both Tx node k and Rx node j for IC, a larger DoF region can be achieved, as shown in the outer pentagon in Fig. 2.1(b), where the shaded area is the gain in the feasible DoF region.

The existence of rank-deficient channels for MIMOs with many antennas calls for a deeper understanding of DoF-based IC models. Unfortunately, there is hardly any research results available on this important problem in the wireless networking community. The goal of this chapter is to explore this important area by developing a unified theory on DoF consumption for SM and IC under general channel rank conditions. The main contributions of this chapter are the following:

- Based on a rigorous analytical method for accounting of DoF consumption at a node, we offer a theory on how DoFs at a node are consumed for SM and IC under general channel rank conditions. Our theoretical development starts with a single SM link and IC on a single interference link and then extends to multiple links. In particular, we find that a shared DoF consumption for IC at both transmit and receive nodes is most efficient for DoF allocation under rank-deficient conditions. This result is in contrast to existing DoF models under full-rank conditions.

- We show that in the special case when channels are of full ranks, existing unilateral IC models present themselves as a special case under our new model for general channel rank conditions. That is, our general DoF model remains valid for both full-rank and rank-deficient channel conditions.

- We further extend the general DoF model to analyze multi-link MIMO networks by developing a set of mixed integer linear constraints. This allows our DoF model to be used for identifying the DoF region of a multi-link MIMO network as well as for studying DoF scheduling problems in MIMO networks.

- Through numerical studies, we show that the general DoF model can achieve larger feasible DoF regions or improved objective values than existing unilateral IC models under general channel rank conditions.

The remainder of this chapter is organized as follows. In Section 2.2, we review existing works on DoF-based IC models. In Section 2.3, we present a DoF IC model under general channel rank conditions. In Section 2.4, we revisit previous DoF models that assume full-rank conditions and show that they are a special case under our general DoF model. In Section 2.5, we develop a DoF scheduling model for multi-link MIMO networks. Section 2.6 presents case studies and demonstrate the efficacy of our DoF model under general channel conditions. Section 2.7 concludes this chapter.

2.2 Related Work

DoF-based IC models have been widely studied in the networking community. However, these DoF models have been mainly established under the assumption of full-rank channels [14, 15, 16, 17, 18, 19, 20, 21, 22, 50]. In [14], Bhatia and Li proposed to cancel interference by consuming DoFs on both Tx and Rx nodes. But it is easy to show that consuming DoFs at both sides only results in duplication in IC and is wasteful in DoF resources. In [50], Sundaresan *et al.* proposed that IC could be done by consuming DoFs only at the Rx node. This approach failed to explore the possibility of consuming DoFs at a Tx node and thus led to a smaller solution space. In [15], Blough *et al.* showed that it is sufficient to consume DoFs at either the Tx node or the Rx node to cancel the interference, but not both. Most DoF-based IC models (e.g. [16, 17, 21, 22]) were developed along this "unilateral" approach, which is the most efficient DoF allocation under full-rank channel assumption. As we shall show in this chapter, under general channel rank conditions (i.e., in the presence of rank-deficient channels), a unilateral IC approach is no longer the most efficient and instead, a shared DoF consumption at both Tx and Rx nodes is more efficient. Further, the existing (unilateral) IC scheme for full-rank channels can be regarded as a special case of our new IC

model when channels are of full ranks.

In the IT community, there have been some active research activities to understand MIMO's behavior under rank-deficient channels [25, 27, 34, 35, 36, 37, 39]. The focus there has been to derive closed-form expressions of achievable/outer-bound DoF region for specific link topology and rank settings. Some representative research includes achievable DoFs for point-to-point MIMO channels with an arbitrary number of antennas and channel rank [27], 3-link MIMO with symmetric antenna number and channel rank [25, 27, 34], K-link MIMO with symmetric antenna number and/or channel rank [27, 37], K-link MIMO with rank-deficient channels only on interference links [35]. A few works also considered simple multi-hop networks. In [36], Sun *et al.* studied the upper bound of the DoF region under $2 \times 2 \times 2$ link topology. In [38], Chae *et al.* showed the achievable DoFs under relay-assisted K-link topology with specific rank settings. In [39], Fanjul *et al.* proposed a scheme to construct beamforming matrices given that the DoFs allocated for SM at each link are known and feasible. None of these results can be used for DoF allocation for arbitrary network topology and general channel rank conditions. In other words, there is a lack of study of rank-deficient from a networking perspective, i.e., a lack of results that can be used for DoF allocation in a MIMO network, which is the primary interest in the wireless networking community.

In the literature, another related interference management technique is known as interference alignment (IA) (see, e.g., [40, 41, 42]). The focus of IA is to jointly construct the signals so that multiple interfering signals are aligned in the same direction at an unintended receiver. In other words, the focus of IA is on signal alignment. In contrast, our proposed scheme (DoF-based IC) focuses on how to eliminate interference with the fewest number of DoFs without exploiting signal alignment. To better understand the relationship between IA and DoF-based IC (our scheme), let's compare the solution spaces, multiplexing gains and tractability of these two schemes. First, since IA requires additional constraints (for

signal alignment), the solution space for IA is a subspace of that for IC. Second, for both IA and IC, the optimal multiplexing gain that can be achieved under a general multi-link MIMO network remains an open problem. But generally speaking, a feasible IA scheme provides higher multiplexing gain, as fewer DoFs are needed to cancel interference when multiple interferences are aligned. Third, IA is much more complex than IC from a theoretical perspective. To date, how to design a feasible IA scheme for general topologies (under general rank conditions) remains unknown. IA is only understood for certain topologies (e.g., [36, 39, 40]). On the other hand, this chapter offers a tractable approach for IC under arbitrary topologies and general rank conditions.

2.3 DoF Consumption under General Channel Rank Conditions: A Theory

In this section, we present a DoF IC model under general rank conditions. We say a channel is under general channel rank condition if the channel is either rank-deficient or full rank. Throughout our exposition, we assume general channels unless we make an explicit distinction between rank-deficient and full-rank conditions.

The concept of DoF was originally developed to represent the multiplexing gain of a MIMO channel. For a multi-link network, the sum of DoFs in the network represents the total number of data streams that can be transmitted simultaneously (free of interference) in the network. This DoF concept was then extended to characterize a node's spatial freedom by its multiple antennas. DoFs at a node can be used for SM and IC.

As more and more DoFs are used for SM and IC at a node, its spatial freedom diminishes. In this section, we develop a rigorous accounting method for DoF consumption at a node

Table 2.1: Notations in Chapter 2

Symbol	Definition
$\mathbb{C}$	A complete set of complex numbers
d_{ij}^{R}	Number of DoFs consumed by Rx node j to cancel interference from Tx node i to Rx node j
d_{ij}^{T}	Number of DoFs consumed by Tx node i to cancel interference from Tx node i to Rx node j
$\mathbf{H}_{ij}$	Channel matrix from Tx node i to Rx node j
$\mathbf{I}_m$	Identity matrix with dimension $m \times m$.
$\mathcal{I}_i$	Set of nodes within node i's interference range
$\mathcal{K}$	Set of nodes in the network
N_i	Number of antennas at node i
r_{ij}	Rank of $\mathbf{H}_{ij}$
$\mathcal{T}_i$	Set of nodes within node i's transmission range
$\mathbf{U}_i$	Weight matrix at Tx node i
$\mathbf{V}_j$	Weight matrix at Rx node j
$x_i(t)$	A binary variable to indicate whether node i is a Tx node at time t
$y_i(t)$	A binary variable to indicate whether node i is an Rx node at time t
z_{i*}	Total number of outgoing data streams at Tx node i
z_{*j}	Total Number of incoming data streams at Rx node j
z_{ij}	Number of data streams from Tx node i to Rx node j
$\mathbf{1}_{ij}^{\mathrm{R}}$	A binary variable to indicate whether Rx node j consumes DoFs for IC from i to j
$\mathbf{1}_{ij}^{\mathrm{T}}$	A binary variable to indicate whether Tx node i consumes DoFs for IC from i to j
$\mathbf{X}^{\dagger}$	Hermitian transpose of matrix $\mathbf{X}$

that is tightly related to the number of constraints at the node. Based on this accounting method, we offer a theory on how DoFs at a node are consumed for SM and IC under general channel rank conditions. Our theoretical development begins with a fundamental understanding of SM on a single link and IC on a single interference link. Then our theory generalizes to multi-link MIMO networks via an additivity property.

We consider a multi-link MIMO network with an arbitrary topology. Some of the key assumptions that we made in this chapter include the following. We assume channel state information (CSI) is known at both Tx and Rx nodes, and the set of interfering nodes at a node is also given. All channels are assumed to be generic. That is, the channel matrices are randomly and independently generated from continuous distributions subject to rank-constraints, without any special structure. Further, we do not consider diversity gain and multi-cast channels.

2.3.1 DoF Consumption at Node

To quantify the DoFs consumption at a node, it is necessary to have an analytical method for DoF accounting, which we formally describe as follows.

Assume node i has N_i antennas. Denote $\mathbf{x}_{ij} \in \mathbb{C}^{N_i \times 1}$ as the weight vector at node i for the j-th stream, where $\mathbb{C}^{m \times n}$ denotes a complex set with dimension $m \times n$. With N_i antennas, there can be at most N_i streams. Assume node i transmits or receives n_s streams (where $n_s \leq N_i$). Then its weight matrix $\mathbf{X}_i = [\mathbf{x}_{i1}, \mathbf{x}_{i2}, ..., \mathbf{x}_{in_s}] \in \mathbb{C}^{N_i \times n_s}$. We first introduce the definitions of total number of DoFs at a node and DoF consumption at a node. Then we derive the remaining available DoFs at a node.

Definition 2.1. The total number of DoFs at node i is the maximum number of dimensions that can be spanned by $\mathbf{x}_{i1}, \mathbf{x}_{i2}, ..., \mathbf{x}_{in_s}$.

Definition 2.2. The number of consumed DoFs at node i is the number of linearly independent constraints imposed on $\mathbf{X}_i$.

The number of a node's total DoFs is directly tied to its number of antennas. Initially, when there is no constraint on $\mathbf{X}_i$, each of its elements is undetermined and can be set arbitrarily. There is a feasible region (a space) that includes all possible values by such an unconstrained matrix. The initial DoFs of this feasible region is equal to the number of $\mathbf{X}_i$'s rows (or the number of antennas at the node), i.e., N_i, since N_i is the maximum number of dimensions spanned by $\mathbf{x}_{i1}, \mathbf{x}_{i2}, ..., \mathbf{x}_{in_s}$. Thus, a node's total number of DoFs is the number of antennas at this node.

To perform SM and IC, a node's weight matrix must satisfy certain constraints to achieve interference-free transmission. Thus, some DoFs at the node will be consumed for SM or IC. Assume some constraints are imposed on $\mathbf{X}_i$ in the form $\mathbf{A}\mathbf{X}_i = \mathbf{B}$, where $\mathbf{A} \in \mathbb{C}^{M \times N_i}$ and $\mathbf{B} \in \mathbb{C}^{M \times n_s}$. That is, M linear constraints are imposed on each $\mathbf{x}_{ij}$. Denote $\mathbf{\Phi}$ as the union solution space of each $\mathbf{x}_{ij}$ to problem $\mathbf{A}\mathbf{X}_i = \mathbf{B}$, i.e., $\mathbf{\Phi} = \{\boldsymbol{\phi}_1 \cup \boldsymbol{\phi}_2 \cup \cdots \cup \boldsymbol{\phi}_{n_s} | \mathbf{A}[\boldsymbol{\phi}_1 \ \boldsymbol{\phi}_2 \ \cdots \ \boldsymbol{\phi}_{n_s}] = \mathbf{B}\}$. Then the remaining available DoF at node i is the free dimension of $\mathbf{X}_i$, namely $\dim(\mathbf{\Phi})$.

Lemma 2.3. *Suppose node i has N_i antennas and its weight matrix $\mathbf{X}_i$ is constrained by $\mathbf{A}\mathbf{X}_i = \mathbf{B}$. If* $\text{rank}([\mathbf{A}\ \mathbf{B}]) = \text{rank}(\mathbf{A})$*, then the number of consumed DoFs at node i is equal to* $\text{rank}([\mathbf{A}\ \mathbf{B}])$*, and the remaining available DoFs at node i is $N_i -$* $\text{rank}([\mathbf{A}\ \mathbf{B}])$*. If* $\text{rank}([\mathbf{A}\ \mathbf{B}]) \neq \text{rank}(\mathbf{A})$*, then there is no feasible solution to $\mathbf{A}\mathbf{X}_i = \mathbf{B}$.*

Proof. Initially, all the elements in $\mathbf{X}_i$ are undetermined and can be set arbitrarily. N_i is the maximum number of dimensions spanned by $\mathbf{x}_{i1}, \mathbf{x}_{i2}, ..., \mathbf{x}_{in_s}$, i.e., the number of initial available DoFs provided by $\mathbf{X}_i$ is N_i.

Let $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_{n_s}]$. For any $\mathbf{A} \in \mathbb{C}^{M \times N_i}$ and $\mathbf{b}_j \in \mathbb{C}^{M \times 1}$, where $j \in \{1, ..., n_s\}$,

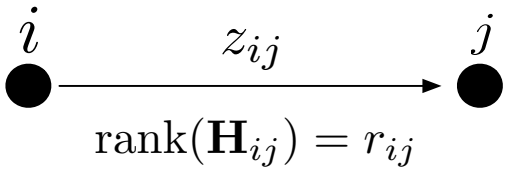


Figure 2.2: Spatial multiplexing on a link.

if $\text{rank}([\mathbf{A}\ \mathbf{b}_j]) = \text{rank}(\mathbf{A})$, then the set of solutions to the non-homogeneous linear system $\mathbf{A}\mathbf{x}_{ij} = \mathbf{b}_j$ is an affine subspace of $\mathbb{C}^{N_i \times 1}$, denoted as $\mathbf{\Phi}_j$. Since the solution dimension of a non-homogeneous linear system is the same as its corresponding homogeneous linear system, we have $\dim(\mathbf{\Phi}_j) = \dim(\text{nullspace}(\mathbf{A})) = N_i - \text{rank}([\mathbf{A}\ \mathbf{b}_j])$. Note that non-homogeneous linear systems $\mathbf{A}\mathbf{x}_{ij} = \mathbf{b}_j$ $(j = 1, ..., n_s)$ are sharing the same corresponding homogeneous linear system $\mathbf{A}\mathbf{x}_{ij} = \mathbf{0}$ (thus share the same homogeneous solutions). We conclude $\dim(\mathbf{\Phi}) = \dim(\mathbf{\Phi}_j) = N_i - \text{rank}([\mathbf{A}\ \mathbf{B}])$. If $\text{rank}([\mathbf{A}\ \mathbf{b}_j]) \neq \text{rank}(\mathbf{A})$, then there is no feasible solution to $\mathbf{A}\mathbf{x}_{ij} = \mathbf{b}_j$. Consequently, if $\text{rank}([\mathbf{A}\ \mathbf{B}]) \neq \text{rank}(\mathbf{A})$, then there is no feasible solution to $\mathbf{A}\mathbf{X}_i = \mathbf{B}$. □

Lemma 2.3 shows that the number of consumed DoFs at a node is determined by the number of linearly independent constraints imposed on its weight matrix. In particular, SM and IC will appear in the form of constraints that are to be imposed on a node's weight matrix and will consume DoFs. That is, one DoF is consumed for each linearly independent constraint imposed on $\mathbf{X}_i$. The number of linearly independent constraints (DoFs consumed) on $\mathbf{X}_i$ is equal to $\text{rank}([\mathbf{A}\ \mathbf{B}])$, and the remaining available DoFs at node i is $\mathbf{X}_i$ to be $N_i - \text{rank}([\mathbf{A}\ \mathbf{B}])$. Based on this understanding, we study DoF consumption by SM and IC separately under general channel rank conditions in the following two sections.

2.3.2 DoF Consumption for SM under General Channel Rank Conditions

Consider the single transmission link in Fig. 2.2, where the number of data streams transmitted from Tx node i to Rx node j is z_{ij}, and $\text{rank}(\mathbf{H}_{ij}) = r_{ij}$. Then some DoFs at Tx node i and Rx node j will be consumed for SM. As expected, the number of data streams transmitted on channel $\mathbf{H}_{ij}$ cannot exceed the rank of this channel.

Lemma 2.4. *For transmission on a single link where node i is a transmitter and node j is a receiver, z_{ij} data streams can be transmitted free of interference only if $z_{ij} \leq r_{ij}$. Further, the number of DoFs consumed by SM at node i and node j are both z_{ij}.*

Proof. Denote $\mathbf{U}_i$ and $\mathbf{V}_j$ as the weight matrices at Tx node i and Rx node j, respectively. Since the number of data streams transmitted from Tx node i to Rx node j is z_{ij}, $\mathbf{U}_i^\dagger$ and $\mathbf{V}_j$ can be represented as $[\mathbf{u}_{i1}\ \mathbf{u}_{i2}\ ...\mathbf{u}_{iz_{ij}}]^\dagger$ and $[\mathbf{v}_{j1}\ \mathbf{v}_{j2}\ ...\mathbf{v}_{jz_{ij}}]$, respectively. To ensure interference-free transmission of z_{ij} data streams, the following constraint must be satisfied:

$$\underset{z_{ij}\times N_i}{\mathbf{U}_i^\dagger} \cdot \underset{N_i\times N_j}{\mathbf{H}_{ij}} \cdot \underset{N_j\times z_{ij}}{\mathbf{V}_j} = \mathbf{I}_{z_{ij}}, \tag{2.1}$$

where $\mathbf{I}_{z_{ij}}$ denotes identity matrix with dimension $z_{ij} \times z_{ij}$.

We first consider DoF consumption at Rx node j. We have

$$\text{rank}\left(\left[\underset{z_{ij}\times N_i}{\mathbf{U}_i^\dagger} \cdot \underset{N_i\times N_j}{\mathbf{H}_{ij}} \quad \mathbf{I}_{z_{ij}}\right]\right) = z_{ij}. \tag{2.2}$$

Note that $\text{rank}(\mathbf{H}_{ij})$ must be at least z_{ij}. Otherwise, $\text{rank}(\mathbf{U}_i^\dagger\mathbf{H}_{ij}) \leq \min\{\text{rank}(\mathbf{U}_i^\dagger), \text{rank}(\mathbf{H}_{ij})\} < z_{ij} = \text{rank}([\mathbf{U}_i^\dagger\mathbf{H}_{ij}\ \mathbf{I}_{z_{ij}}])$, and Eq. (2.1) will have no solution. This means z_{ij} data streams can be transmitted only if $z_{ij} \leq r_{ij}$ is satisfied. By (2.1), (2.2) and Lemma 2.3,

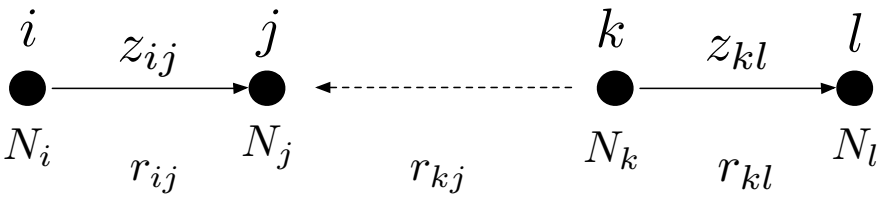


Figure 2.3: Interference cancellation between two nodes.

the number of DoFs consumed by SM at Rx node j is z_{ij}.

Following the same token, one can show that at Tx node i, the number of DoFs consumed for SM is also z_{ij}. □

2.3.3 DoF Consumption for IC under General Channel Rank Conditions

Consider a single-interference case as shown in Fig. 2.3, Tx nodes i and k are transmitting z_{ij} and z_{kl} data streams to Rx nodes j and l, respectively, where $z_{ij} \geq 1, z_{kl} \geq 1$, and Rx node j is interfered with by Tx node k, $\text{rank}(\mathbf{H}_{kj}) = r_{kj}$. Suppose channel matrix $\mathbf{H}_{kj}$ is of general rank condition, (i.e., $\mathbf{H}_{kj}$ may be rank deficient). Then how to cancel the interference from k to j so that data streams z_{ij} can be received at Rx node j free of interference?

Denote $\mathbf{1}_{kj}^{\mathrm{T}}$ and $\mathbf{1}_{kj}^{\mathrm{R}}$ as binary variables with the following definitions:

$$\mathbf{1}_{kj}^{\mathrm{T}} = \begin{cases} 1 & \text{if Tx node } k \text{ consumes DoFs for IC from } k \text{ to } j, \\ 0 & \text{otherwise,} \end{cases}$$

$$\mathbf{1}_{kj}^{\mathrm{R}} = \begin{cases} 1 & \text{if Rx node } j \text{ consumes DoFs for IC from } k \text{ to } j, \\ 0 & \text{otherwise.} \end{cases}$$

Then following theorem shows how the interference is cancelled by consuming DoFs at Tx

node k and Rx node j.

Theorem 2.5. *For the single-interference case, let Tx node k consume d_{kj}^{T} DoFs and Rx node j consume d_{kj}^{R} DoFs for IC. Then interference from Tx node k to Rx node j is cancelled if*

$$d_{kj}^{\mathrm{R}}\mathbf{1}_{kj}^{\mathrm{R}} + d_{kj}^{\mathrm{T}}\mathbf{1}_{kj}^{\mathrm{T}} = \min\left\{z_{kl}\mathbf{1}_{kj}^{\mathrm{R}} + z_{ij}\mathbf{1}_{kj}^{\mathrm{T}},\quad r_{kj}\right\}, \tag{2.3a}$$

$$\left(\mathbf{1}_{kj}^{\mathrm{R}}, \mathbf{1}_{kj}^{\mathrm{T}}\right) \neq (0,0). \tag{2.3b}$$

Proof. To guarantee interference-free transmission, the following constraint must be satisfied:

$$\underset{z_{kl}\times N_k}{\mathbf{U}_k^{\dagger}} \cdot \underset{N_k\times N_j}{\mathbf{H}_{kj}} \cdot \underset{N_j\times z_{ij}}{\mathbf{V}_j} = \underset{z_{kl}\times z_{ij}}{\mathbf{0}}. \tag{2.4}$$

The theorem can be proved by enumerating all possibilities of $\left(\mathbf{1}_{kj}^{\mathrm{R}}, \mathbf{1}_{kj}^{\mathrm{T}}\right)$.

Case I: Only Rx node j consumes DoFs for IC, i.e., $\left(\mathbf{1}_{kj}^{\mathrm{R}}, \mathbf{1}_{kj}^{\mathrm{T}}\right) = (1,0)$. This means we impose constraint (2.4) on $\mathbf{V}_j$. We have

$$\operatorname{rank}\left(\left[\underset{z_{kl}\times N_k}{\mathbf{U}_k^{\dagger}} \cdot \underset{N_k\times N_j}{\mathbf{H}_{kj}} \quad \underset{z_{kl}\times N_j}{\mathbf{0}}\right]\right) \leq \min\{z_{kl}, r_{kj}\}. \tag{2.5}$$

This indicates node j may use fewer DoFs than $\min\{z_{kl}, r_{kj}\}$ to cancel interference. But since $\mathbf{H}_{kj}$ is generic, without "special treatment" (case III) on $\mathbf{U}_k$, we have to consider an upper bound $\min\{z_{kl}, r_{kj}\}$ to guarantee interference-free transmission. Thus according to Lemma 2.3, the number of DoFs consumed for IC at Rx node j is $\min\{z_{kl}, r_{kj}\}$.

Case II: Only Tx node k consumes DoFs for IC, i.e., $\left(\mathbf{1}_{kj}^{\mathrm{R}}, \mathbf{1}_{kj}^{\mathrm{T}}\right) = (0,1)$. The proof is similar to Case I and we omit the details to conserve space. The number of DoFs consumed for IC at Tx node j is $\min\{z_{ij}, r_{kj}\}$.

Case III: Let both Tx node k and Rx node j consume DoFs for IC. i.e., $\left(\mathbf{1}_{kj}^{\mathrm{R}}, \mathbf{1}_{kj}^{\mathrm{T}}\right) = (1,1)$.

Obviously, if $z_{kl} + z_{ij} \leq r_{kj}$, Theorem 2.5 is trivial and can be proved based on the same analysis as in Case I and Case II. Now we prove Theorem 2.5 when $z_{kl} + z_{ij} > r_{kj}$.

By singular value decomposition (SVD) of the interference channel, we have $\mathbf{H}_{kj} = \mathbf{U}_k'^{\dagger} \mathbf{\Lambda}_{kj} \mathbf{V}_j'$, where $\mathbf{U}_k'^{\dagger}$ and $\mathbf{V}_j'$ are $N_k \times N_k$ and $N_j \times N_j$ unitary matrices, respectively. $\mathbf{\Lambda}_{kj}$ is a $N_k \times N_j$ diagonal matrix with singular values of $\mathbf{H}_{kj}$ on the main diagonal and zeros elsewhere. Denote $\bar{\mathbf{U}}_k = \mathbf{U}_k' \mathbf{U}_k$, $\bar{\mathbf{V}}_j = \mathbf{V}_j' \mathbf{V}_j$. Note that it is just linear transformation from $\mathbf{U}_k$ to $\bar{\mathbf{U}}_k$, $\mathbf{V}_j$ to $\bar{\mathbf{V}}_j$, maintaining the same number of dimensions (the same rank). Also, $\mathbf{U}_k$ can be easily derived by $\mathbf{U}_k'^{-1} \bar{\mathbf{U}}_k$. Therefore, it is equivalent to use $\bar{\mathbf{U}}_k$, $\bar{\mathbf{V}}_j$ and $\mathbf{\Lambda}_{kj}$ in the proof.

We can write $\mathbf{\Lambda}_{kj}$ and $\bar{\mathbf{U}}_k$ as

$$\mathbf{\Lambda}_{kj} = \begin{bmatrix} \lambda_1 & & & & & \\ & \lambda_2 & & & & \\ & & \ddots & & & \\ & & & \lambda_{r_{kj}} & & \\ & & & & 0 & \\ & & & & & \ddots \\ & & & & & 0 \end{bmatrix}_{N_k \times N_j},$$

$$\bar{\mathbf{U}}_k = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_{N_k} \end{bmatrix}_{N_k \times z_{kl}}.$$

According to Sylvester's rank inequality: if $\mathbf{A}$ is an $m \times n$ matrix and $\mathbf{B}$ is an $n \times k$

matrix, then

$$\text{rank}(\mathbf{AB}) \geq \text{rank}(\mathbf{A}) + \text{rank}(\mathbf{B}) - n. \tag{2.6}$$

Thus we have $\text{rank}\left(\begin{bmatrix} \bar{\mathbf{U}}_k^\dagger \\ {\scriptstyle z_{kl}\times N_k} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{\Lambda}_{kj} \\ {\scriptstyle N_k \times N_j} \end{bmatrix}\right) \geq z_{kl} + r_{kj} - N_k$. We can force the rank of $\begin{bmatrix} \bar{\mathbf{U}}_k^\dagger \\ {\scriptstyle z_{kl}\times N_k} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{\Lambda}_{kj} \\ {\scriptstyle N_k \times N_j} \end{bmatrix} = [\lambda_1 \mathbf{u}_1^\dagger \ \lambda_2 \mathbf{u}_2^\dagger \ \cdots \ \lambda_{r_{kj}} \mathbf{u}_{r_{kj}}^\dagger \ \mathbf{0} \cdots \mathbf{0}]$ to be at most $r', (z_{kl} + r_{kj} - N_k \leq r')$, by adding the following $r_{kj} - r'$ linear independent constraints on $\bar{\mathbf{U}}_k^\dagger$:

$$\begin{aligned}
\lambda_{r'+1}\mathbf{u}_{r'+1}^\dagger &= \omega_{11}\lambda_1\mathbf{u}_1^\dagger + \omega_{12}\lambda_2\mathbf{u}_2^\dagger + ... + \omega_{1r'}\lambda_{r'}\mathbf{u}_{r'}^\dagger, \\
\lambda_{r'+2}\mathbf{u}_{r'+2}^\dagger &= \omega_{21}\lambda_1\mathbf{u}_1^\dagger + \omega_{22}\lambda_2\mathbf{u}_2^\dagger + ... + \omega_{2r'}\lambda_{r'}\mathbf{u}_{r'}^\dagger, \\
&\vdots \\
\lambda_{r_{kj}}\mathbf{u}_{r_{kj}}^\dagger &= \omega_{(r_{kj}-r')1}\lambda_1\mathbf{u}_1^\dagger + \omega_{(r_{kj}-r')2}\lambda_2\mathbf{u}_2^\dagger + ... \\
&\quad + \omega_{(r_{kj}-r')r'}\lambda_{r'}\mathbf{u}_{r'}^\dagger,
\end{aligned} \tag{2.7}$$

where scalars $\omega_{i1}, \omega_{i2}, ..., \omega_{ir'}$ are not all zeros for $1 \leq i \leq r_{kj} - r'$.

Denote $\mathbf{\Omega} =$

$$\begin{bmatrix}
\omega_{11}\lambda_1 & \omega_{12}\lambda_2 & \cdots & \omega_{1r'}\lambda_{r'} \\
\omega_{21}\lambda_1 & \omega_{22}\lambda_2 & \cdots & \omega_{2r'}\lambda_{r'} \\
\vdots & \vdots & \ddots & \vdots \\
\omega_{(r_{kj}-r')1}\lambda_1 & \omega_{(r_{kj}-r')2}\lambda_2 & \cdots & \omega_{(r_{kj}-r')r'}\lambda_{r'}
\end{bmatrix}
\begin{bmatrix}
-\lambda_{r'+1} & 0 & \cdots & 0 & 0 & \cdots & 0 \\
0 & -\lambda_{r'+2} & \cdots & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & -\lambda_{r_{kj}} & 0 & \cdots & 0
\end{bmatrix}_{(r_{kj}-r')\times N_k}. \tag{2.8}$$

(2.7) is equivalent to

$$\mathbf{\Omega} \cdot \bar{\mathbf{U}}_k = \mathbf{0} \tag{2.9}$$

Obviously, the rank of $[\mathbf{\Omega}\ \mathbf{0}]$ is $r_{kj} - r'$ and thus $r_{kj} - r'$ DoFs are consumed at node k by Eq. (2.9) and Lemma 2.3.

Next, since the rank of $\underset{z_{kl}\times N_k}{\bar{\mathbf{U}}_k^\dagger} \cdot \underset{N_k \times N_j}{\mathbf{\Lambda}_{kj}} \quad \underset{z_{kl}\times z_{ij}}{\mathbf{0}}$ is at most r', we can use r' DoFs at node j to force $\mathbf{U}_k^\dagger \mathbf{H}_{kj} \mathbf{V}_j = \mathbf{0}$ according to Lemma 2.3. Thus we have $d_{kj}^{\mathrm{R}} + d_{kj}^{\mathrm{T}} = r_{kj}$.

□

Theorem 2.5 shows that to cancel interference from Tx node k to Rx node j, DoFs can be consumed either Tx node k, or Rx node j, or both nodes. The required number of DoFs consumed at Tx node k and Rx node j are related to the number of data streams and rank of the interference channel. By enumerating all possibilities of $\left(\mathbf{1}_{kj}^{\mathrm{R}}, \mathbf{1}_{kj}^{\mathrm{T}}\right)$ in (2.3b), IC can be done by one of the following three cases:

- Case I: $\left(\mathbf{1}_{kj}^{\mathrm{R}}, \mathbf{1}_{kj}^{\mathrm{T}}\right) = (1, 0)$, i.e., only Rx node j consumes DoFs for IC and the number of DoFs that Rx node j consumes is $\min\{z_{kl}, r_{kj}\}$.

- Case II: $\left(\mathbf{1}_{kj}^{\mathrm{R}}, \mathbf{1}_{kj}^{\mathrm{T}}\right) = (0, 1)$, i.e., only Tx node k consumes DoFs for IC and the number of DoFs that Tx node k consumes is $\min\{z_{ij}, r_{kj}\}$.

- Case III: $\left(\mathbf{1}_{kj}^{\mathrm{R}}, \mathbf{1}_{kj}^{\mathrm{T}}\right) = (1, 1)$, i.e., both Tx node k and Rx node j consume DoFs for IC. If $z_{kl} + z_{ij} \leq r_{kj}$, then $d_{kj}^{\mathrm{R}} + d_{kj}^{\mathrm{T}} = z_{kl} + z_{ij}$. That is, a total of $z_{kl} + z_{ij}$ DoFs are used for IC, which is more than that in the previous two cases (either transmitter or receiver). On the other hand, if $r_{kj} < z_{kl} + z_{ij}$, it is possible to design $\mathbf{U}_k^\dagger$ and $\mathbf{V}_j^\dagger$ such that $(r_{kj} - x)$ DoFs are consumed at Tx node k to guarantee the rank of $\left[\mathbf{U}_k^\dagger \cdot \mathbf{H}_{kj}\right]$ is at most x. Then Rx node j will consume x DoFs to cancel this interference. Thus

we have $d_{kj}^{\mathrm{R}} + d_{kj}^{\mathrm{T}} = r_{kj}$. This is the most interesting case and is quite surprising. It shows that a shared DoF consumption between Tx and Rx for IC is most efficient under rank-deficient conditions.

To fully understand and appreciate the significance of the third (and new) case, let's revisit the motivating example in Section 2.1 (see Fig. 2.1). First, $(z_{ij}, z_{kl}) = (5, 7)$ is a feasible solution and can be realized by Case I, i.e., $\left(\mathbf{1}_{kj}^{\mathrm{R}}, \mathbf{1}_{kj}^{\mathrm{T}}\right) = (1, 0)$, because Rx node j can consume 5 DoFs for SM and 7 DoFs for IC, and Tx node k uses 7 DoFs for SM. Second, $(z_{ij}, z_{kl}) = (5, 7)$ can also be designed under Case II, i.e., $\left(\mathbf{1}_{kj}^{\mathrm{R}}, \mathbf{1}_{kj}^{\mathrm{T}}\right) = (0, 1)$, where Tx node k consumes 7 DoFs for SM and 5 DoFs for IC, and Rx node j consumes 5 DoFs for SM. Following the same token, we can find a feasible region of the inner pentagon in Fig. 2.1(b). However, for $(z_{ij}, z_{kl}) = (8, 7)$, it is impossible to have only Rx node j (Case I) or Tx node k (Case II) alone to cancel this interference. But if we let Rx node j consumes 8 DoFs for SM and 4 DoFs for IC, and Tx node k consumes 7 DoFs for SM and 5 DoFs for IC, then the condition in Case III (i.e., $\left(\mathbf{1}_{kj}^{\mathrm{R}}, \mathbf{1}_{kj}^{\mathrm{T}}\right) = (1, 1)$) will be satisfied and we have a feasible solution. That is, *under channel rank-deficient condition, a shared DoF consumption between both Tx node k and Rx node j can offer more feasible solutions than unilateral IC by only Tx node or Rx node.* The outer pentagon in Fig. 2.1(b) shows the extra feasible DoF region. This finding is new and beyond the state-of-the-art.

2.3.4 Extension to Multiple Links and Additivity Property

The results in the previous two sections show DoF consumption for SM on a single link and IC between a Tx node and a Rx node. Using these results as basic building blocks, we explore DoF consumption for the general multiple-link case in this section. Consider Fig. 2.4, where Tx nodes $i_1, i_2, ..., i_P$ are transmitting $z_{i_1 j}, z_{i_2 j}, ..., z_{i_P j}$ data streams to Rx node j, respectively.

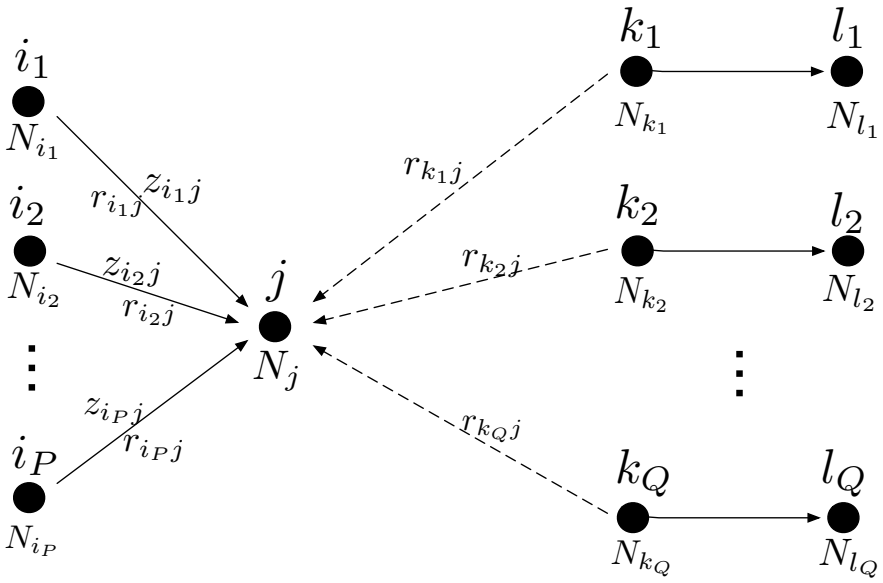


Figure 2.4: Additivity of DoF consumption at Rx node j.

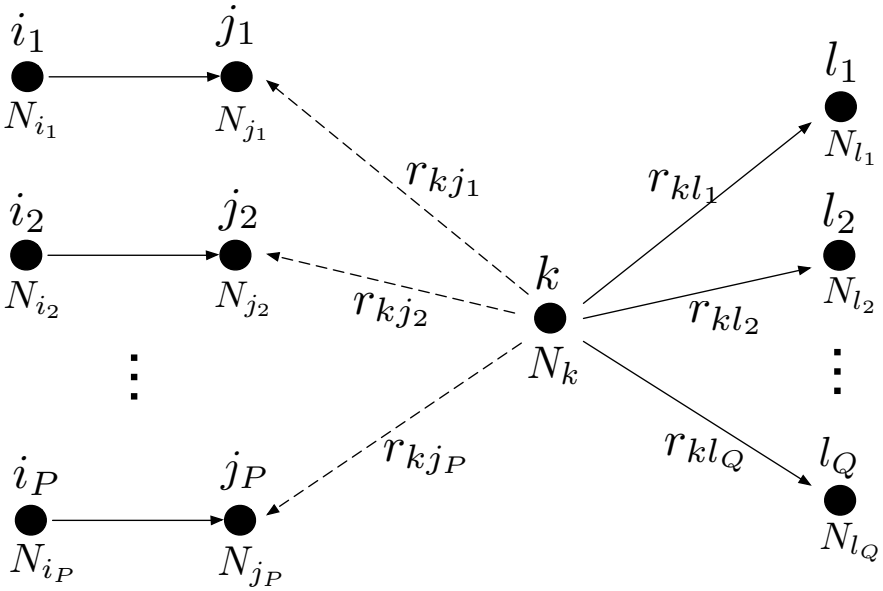


Figure 2.5: Additivity of DoF consumption at Tx node i.

Rx node j is also interfered with by Tx nodes $k_1, k_2, .., k_Q$ simultaneously. Suppose Tx nodes $k_1, k_2, .., k_Q$ are transmitting $z_{k_1 l_1}, z_{k_2 l_2}, ..., z_{k_Q l_Q}$ data streams to their respective receivers. Suppose the number of consumed DoFs at Rx node j for cancelling interference from k_n to j is $d^{\mathrm{R}}_{k_n j}$, where $d^{\mathrm{R}}_{k_n j}\mathbf{1}^{\mathrm{R}}_{k_n j} + d^{\mathrm{T}}_{k_n j}\mathbf{1}^{\mathrm{T}}_{k_n j} = \min\left\{z_{k_n l_n}\mathbf{1}^{\mathrm{R}}_{k_n j} + z_{*j}\mathbf{1}^{\mathrm{T}}_{k_n j},\ r_{k_n j}\right\}$, $\left(\mathbf{1}^{\mathrm{R}}_{k_n j}, \mathbf{1}^{\mathrm{T}}_{k_n j}\right) \neq (0,0)$, $k_n = k_1, k_2, ..., k_Q$. The following Lemma shows the required DoF consumption for SM and IC at Rx node j.

Lemma 2.6. *In a general multi-link case for a Rx node j, the number of consumed DoFs for SM and IC at Rx node j is additive and constrained by channel ranks. If $z_{i_m j} \leq r_{i_m j}$ for $m = 1, 2, ..., P$ are satisfied, then the number of consumed DoFs for SM at Rx node j is $\sum_{m=1}^{P} z_{i_m j}$. The total number of consumed DoFs for IC at Rx node j is $\sum_{n=1}^{Q} d^{\mathrm{R}}_{k_n j}$. The total number of consumed DoFs for SM and IC at Rx node j is $\sum_{m=1}^{P} z_{i_m j} + \sum_{n=1}^{Q} d^{\mathrm{R}}_{k_n j}$.*

Proof. Supposing Rx node j consumes $d^{\mathrm{R}}_{k_n j}$ DoFs to cancel interference from Tx node k_n to Rx node j, $n = 1, 2, ..., Q$. According to Lemma 2.3 and Theorem 2.5, we have $\mathrm{rank}([\mathbf{U}^{\dagger}_{k_1}\mathbf{H}_{k_1 j}]) \leq d^{\mathrm{R}}_{k_1 j}$, $\mathrm{rank}([\mathbf{U}^{\dagger}_{k_2}\mathbf{H}_{k_2 j}]) \leq d^{\mathrm{R}}_{k_2 j}$,..., $\mathrm{rank}([\mathbf{U}^{\dagger}_{k_Q}\mathbf{H}_{k_Q j}]) \leq d^{\mathrm{R}}_{k_Q j}$. Now consider the general multi-link case for a Rx node j (see Fig. 2.4). Weight matrix $\mathbf{V}_j$ of node j must satisfy

$$\begin{bmatrix} \mathbf{U}^{\dagger}_{i_1}\mathbf{H}_{i_1 j} \\ \mathbf{U}^{\dagger}_{i_2}\mathbf{H}_{i_2 j} \\ \vdots \\ \mathbf{U}^{\dagger}_{i_P}\mathbf{H}_{i_P j} \\ \mathbf{U}^{\dagger}_{k_1}\mathbf{H}_{k_1 j} \\ \mathbf{U}^{\dagger}_{k_2}\mathbf{H}_{k_2 j} \\ \vdots \\ \mathbf{U}^{\dagger}_{k_Q}\mathbf{H}_{k_Q j} \end{bmatrix} \mathbf{V}_j = \begin{bmatrix} \mathbf{I}_{z_{i_1 j}} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{z_{i_2 j}} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I}_{z_{i_P j}} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} \tag{2.10}$$

Thus we have

$$\begin{aligned}
&\text{rank}\left(\begin{bmatrix}
\mathbf{U}_{i_1}^{\dagger}\mathbf{H}_{i_1 j} & \mathbf{I}_{z_{i_1 j}} & \mathbf{0} & \cdots & \mathbf{0} \\
\mathbf{U}_{i_2}^{\dagger}\mathbf{H}_{i_2 j} & \mathbf{0} & \mathbf{I}_{z_{i_2 j}} & \cdots & \mathbf{0} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\mathbf{U}_{i_P}^{\dagger}\mathbf{H}_{i_P j} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I}_{z_{i_P j}} \\
\mathbf{U}_{k_1}^{\dagger}\mathbf{H}_{k_1 j} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\
\mathbf{U}_{k_2}^{\dagger}\mathbf{H}_{k_2 j} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\mathbf{U}_{k_Q}^{\dagger}\mathbf{H}_{k_Q j} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0}
\end{bmatrix}\right) \\
\leq & z_{i_1 j} + z_{i_2 j} ... + z_{i_P j} + d_{k_1 j}^{\mathrm{R}} + d_{k_2 j}^{\mathrm{R}} + ... + d_{k_Q j}^{\mathrm{R}} \\
= & \sum_{m=1}^{P} z_{i_m j} + \sum_{n=1}^{Q} d_{k_n j}^{\mathrm{R}}.
\end{aligned} \tag{2.11}$$

The first $\sum_{m=1}^{P} z_{i_m j}$ rows are with full rank $\sum_{m=1}^{P} z_{i_m j}$; the remaining rows may have rank lower than $\sum_{n=1}^{Q} d_{k_n j}^{\mathrm{R}}$ and we consider the upper bound in this chapter. According to (2.10), (2.11) and Lemma 2.3, the number of DoFs consumed for SM and IC at Rx node j is $\sum_{m=1}^{P} z_{i_m j} + \sum_{n=1}^{Q} d_{k_n j}^{\mathrm{R}}$. Note that $z_{i_m j} \leq r_{i_m j}, m = 1, 2, ..., P$ must be satisfied, and the channels are assumed to be generic (i.e., row space of $\mathbf{U}_{i_n}^{\dagger}\mathbf{H}_{i_n j}$ for $n = \{1, 2, \cdots, P\}$ are mutually linearly independent almost surely, and $\mathbf{U}_{i_n}^{\dagger}\mathbf{H}_{i_n j}$ and $\mathbf{U}_{k_n}^{\dagger}\mathbf{H}_{k_n j}$ are mutually linearly independent almost surely), otherwise Eq. (2.10) will have no feasible solution. □

Next we consider the case of SM and IC at Tx node k as shown in Fig. 2.5, where Tx node k is transmitting $z_{kl_1}, z_{kl_2}, ..., z_{kl_Q}$ data streams to Rx nodes $l_1, l_2, ..., l_Q$, respectively. Tx node k is also interfering with Rx nodes $j_1, j_2, .., j_P$. Suppose Rx nodes $j_1, j_2, .., j_P$ are receiving $z_{i_1 j_1}, z_{i_2 j_2}, ..., z_{i_P j_P}$ data streams from Tx nodes $i_1, i_2, ..., i_P$, respectively. Supposing the number of consumed DoFs at Tx node k for cancelling interference from k to j_n is $d_{k j_n}^{\mathrm{T}}$,

where

$$d^{\mathrm{R}}_{kj_n}\mathbf{1}^{\mathrm{R}}_{ij_n} + d^{\mathrm{T}}_{kj_n}\mathbf{1}^{\mathrm{T}}_{kj_n} = \min\left\{z_{k*}\mathbf{1}^{\mathrm{R}}_{kj_n} + z_{i_nj_n}\mathbf{1}^{\mathrm{T}}_{kj_n},\ r_{kj_n}\right\}, \left(\mathbf{1}^{\mathrm{R}}_{kj_n}, \mathbf{1}^{\mathrm{T}}_{kj_n}\right) \neq (0,0), \quad j_n = j_1, j_2, ..., j_P.$$

The following Lemma shows the required DoF consumption at Tx node k.

Lemma 2.7. *In a general multi-link case for a Tx node k, the number of consumed DoFs for SM and IC at Tx node k is additive and constrained by channel ranks. If $z_{kl_m} \leq r_{kl_m}$ for $m = 1, 2, ..., Q$, then the number of consumed DoFs for SM at Tx node k is $\sum_{m=1}^{Q} z_{kl_m}$. The number of consumed DoFs for IC at Tx node k is $\sum_{n=1}^{P} d^{\mathrm{T}}_{kj_n}$. The total number of consumed DoFs for SM and IC at Tx node k is $\sum_{m=1}^{Q} z_{kl_m} + \sum_{n=1}^{P} d^{\mathrm{T}}_{kj_n}$.*

The proof of Lemma 2.7 is similar to Lemma 2.6 and is omitted to conserve space.

2.4 A Special Case: Full-rank Channels

In this section, we show that, when channel has full rank, our general DoF model degenerates into the well-known unilateral DoF consumption model in the literature (e.g., [14, 15, 16, 17, 21, 22]). Therefore, the existing full-rank DoF model is a special case of our model.

For SM, consider a single link transmission (see Fig. 2.2). If the channel matrix is of full rank (i.e., $\mathrm{rank}(\mathbf{H}_{ij}) = \min\{N_i, N_j\}$), then at most $\min\{N_i, N_j\}$ data streams can be transmitted over this link, and the number of DoFs consumed by SM at Tx node i and Rx node j are both z_{ij}.

For IC, consider the single-interference link (see Fig. 2.3). When channel matrix $\mathbf{H}_{kj}$ has full rank, we have $\mathrm{rank}(\mathbf{H}_{kj}) = \min\{N_k, N_j\}$. We consider two existing IC schemes assuming full-rank channels and show that they are a special case of our model.

Scheme 1: IC by Tx or Rx node, but not both: As shown in the literature (e.g. [15, 16, 17, 21, 22]), for IC, DoFs can be consumed unilaterally at either Rx node j (with z_{kl} DoFs), or at Tx node k (with z_{ij} DoFs). Without loss of generality, assume $N_k \leq N_j$, then we have $r_{kj} = \min\{N_k, N_j\} = N_k$.

Case 1: Rx node j consumes DoFs for IC. Since $r_{kj} = N_k \geq z_{kl}$, this is consistent to Theorem 2.5 when $\left(\mathbf{1}_{kj}^{\mathrm{R}}, \mathbf{1}_{kj}^{\mathrm{T}}\right) = (1, 0)$.

Case 2: Tx node k consumes DoFs for IC. Since $z_{ij} \geq 1, z_{kl} \geq 1$, the available DoFs at Tx node k for IC is no more than $N_k - 1$. Consequently the number of data streams that can be received at Rx node j is no more than $N_k - 1$, i.e., $z_{ij} \leq N_k - 1$. We have $z_{ij} < r_{kj}$. This is consistent to Theorem 2.5 when $\left(\mathbf{1}_{kj}^{\mathrm{R}}, \mathbf{1}_{kj}^{\mathrm{T}}\right) = (0, 1)$.

Scheme 2: IC by both Tx and Rx nodes: In this case, interference is cancelled at Rx node j by consuming z_{kl} DoFs, and at Tx node k by consuming z_{ij} DoFs as in [14]. Obviously $z_{ij} + z_{kl} \geq \min\left\{z_{kl}\mathbf{1}_{kj}^{\mathrm{R}} + z_{ij}\mathbf{1}_{kj}^{\mathrm{T}}, r_{kj}\right\}$. We can let $d_{kj}^{\mathrm{R}} = z_{kl}, d_{kj}^{\mathrm{T}} = z_{ij}, \left(\mathbf{1}_{kj}^{\mathrm{R}}, \mathbf{1}_{kj}^{\mathrm{T}}\right) = (1, 1)$, which will satisfy the sufficient condition in Theorem 2.5. Although feasible, this scheme uses more DoFs than necessary and is considered wasteful.

The next question to ask is: in full-rank case, is it possible for Tx and Rx nodes to share DoF consumption for IC such that Tx node k consumes fewer than z_{ij} DoFs and Rx node j consumes fewer than z_{kl} DoFs as in rank-deficient case? The answer to this question is given in the following Lemma.

Lemma 2.8. *In full-rank case, in order to cancel interference from Tx node k to Rx node j (as shown in Fig. 2.3), it is impossible to have both Tx node k consume fewer than z_{ij} DoFs and Rx node j consume fewer than z_{kl} DoFs, where $z_{ij} \geq 1, z_{kl} \geq 1$.*

Proof. Suppose Tx node k consumes x DoFs and Rx node j consumes y DoFs to cancel interference from Tx node k to Rx node j, where $x < z_{ij}$ and $y < z_{kl}$. This means there are

x linearly independent constraints imposed on $\mathbf{U}_k$ and y linearly independent constraints imposed on $\mathbf{V}_j$. We must have

$$\operatorname{rank}\left(\begin{bmatrix} \underset{z_{ij}\times N_j}{\mathbf{V}_j^\dagger} & \cdot & \underset{N_j\times N_k}{\mathbf{H}_{kj}^\dagger} & \underset{z_{ij}\times z_{kl}}{\mathbf{0}} \end{bmatrix}\right) = x, \quad x < z_{ij}, \tag{2.12}$$

and

$$\operatorname{rank}\left(\begin{bmatrix} \underset{z_{kl}\times N_k}{\mathbf{U}_k^\dagger} & \cdot & \underset{N_k\times N_j}{\mathbf{H}_{kj}} & \underset{z_{kl}\times z_{ij}}{\mathbf{0}} \end{bmatrix}\right) = y, \quad y < z_{kl}. \tag{2.13}$$

On the other hand, since $\mathbf{H}_{kj}$ is of full rank, according to Sylvester's rank inequality [43], we have

$$\operatorname{rank}\left(\begin{bmatrix} \underset{z_{ij}\times N_j}{\mathbf{V}_j^\dagger} & \cdot & \underset{N_j\times N_k}{\mathbf{H}_{kj}^\dagger} & \underset{z_{ij}\times z_{kl}}{\mathbf{0}} \end{bmatrix}\right) \geq z_{ij} + \min\{N_k, N_j\} - N_j, \tag{2.14}$$

and

$$\operatorname{rank}\left(\begin{bmatrix} \underset{z_{kl}\times N_k}{\mathbf{U}_k^\dagger} & \cdot & \underset{N_k\times N_j}{\mathbf{H}_{kj}} & \underset{z_{kl}\times z_{ij}}{\mathbf{0}} \end{bmatrix}\right) \geq z_{kl} + \min\{N_k, N_j\} - N_k. \tag{2.15}$$

But (2.14) and (2.15) contradict (2.12) and (2.13).

In fact, the outcome of this cancellation is

$$\underset{z_{kl}\times N_k}{\mathbf{U}_k^\dagger} \cdot \underset{N_k\times N_j}{\mathbf{H}_{kj}} \cdot \underset{N_j\times z_{ij}}{\mathbf{V}_j} \xrightarrow[\text{operations}]{\text{elementary}} \underset{z_{kl}\times z_{ij}}{\begin{pmatrix} 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \mathbf{C} & \\ \underbrace{0 \cdots 0}_{x} & & & & \end{pmatrix}} \Bigg\} y \tag{2.16}$$

where $\mathbf{C}$ is a $(z_{kl} - y) \times (z_{ij} - x)$ matrix which is not guaranteed to be $\mathbf{0}$ and interference still exists.

Thus, it is infeasible to have Tx node k consume $x < z_{ij}$ DoFs and Rx node j consume

$y < z_{kl}$, where $z_{ij} \geq 1$ and $z_{kl} \geq 1$. □

Lemma 2.8 shows that in full-rank case, there is no benefit to have both Tx node and Rx node consume DoFs for IC, as doing so will incur more DoF consumption than necessary. That is precisely the reason why existing DoF IC models only consider using DoFs unilaterally at either Tx or Rx node for IC, but not both. But under general channel rank conditions (i.e., with rank deficiency), things become different. IC burden is better to be distributed between Tx and Rx nodes, as we have shown in Theorem 2.5.

2.5 DoF Scheduling in a Network

In Section 2.3, we developed a theory for DoF model under general channel rank conditions for the most basic topologies. In this section, we apply this model to develop a set of constraints that can be used to characterize a feasible DoF scheduling region for an *arbitrary* multi-link MIMO network. When outfitted with a proper objective function (e.g., the examples in Section 2.6.2), we will have a complete optimization problem involving a DoF scheduling model with general rank conditions.

Consider multi-link MIMO network with an arbitrary topology. Denote $\mathcal{K}$ as the set of nodes in the network. Denote $\mathcal{T}_i$ as the set of nodes that are within node i's transmission range and $\mathcal{I}_i$ as the set of nodes that are within node i's interference range, respectively. We consider a time-slot based scheduling (so that the model can be easily extended to multi-hop applications with additional flow balance constraints [21, 44]). We have the following three sets of constraints for DoF scheduling in a network.

Node Activity and SM Constraints We assume each node in the network is half-duplex, i.e., a node can be either a Tx node, an Rx node, or idle at any time. Define a

binary variable $x_i(t)$ to indicate whether or not node i is a Tx node at time t, i.e., $x_i(t) = 1$ if node i is transmitting at time t and 0 otherwise. Likewise, denote $y_i(t)$ as a binary variable to indicate whether or not node i is a Rx node at time t, i.e., $y_i(t) = 1$ if node i is receiving at time t and 0 otherwise. Then half-duplex constraint at node i can be modeled as:

$$x_i(t) + y_i(t) \leq 1, \quad i \in \mathcal{K}. \tag{2.17}$$

If node i is an active Tx node (i.e., $x_i(t) = 1$), then the total number of DoFs used for transmitting data streams cannot exceed the total number of antennas N_i at this node, i.e.,

$$x_i(t) \leq \sum_{j \in \mathcal{T}_i} z_{ij}(t) \leq N_i x_i(t), \quad i \in \mathcal{K}. \tag{2.18}$$

Similarly, if a node j is an active Rx node (i.e., $y_j(t) = 1$), then the total number of DoFs used for receiving data streams cannot exceed the total number of antennas N_j at this node, i.e.,

$$y_j(t) \leq \sum_{i \in \mathcal{T}_j} z_{ij}(t) \leq N_j y_j(t), \quad j \in \mathcal{K}. \tag{2.19}$$

Further, considering general channel rank condition, the number of data streams that can be sent over a channel must satisfy the following constraint (Lemma 2.4):

$$z_{ij}(t) \leq r_{ij}(t), \quad i \in \mathcal{K}, \ j \in \mathcal{K}, \ i \neq j. \tag{2.20}$$

IC Constraints Consider a Tx node k that interferes a Rx node j in a network (see Fig. 2.6). Tx node k may transmit multiple data streams to Rx nodes $l \in \mathcal{T}_k$ (other than j) while Rx node j may also receive multiple data streams from Tx nodes $i \in \mathcal{T}_j$ (other than k). To cancel interference from Tx node k to Rx node j, we apply Theorem 2.5 and Lemmas 2.6

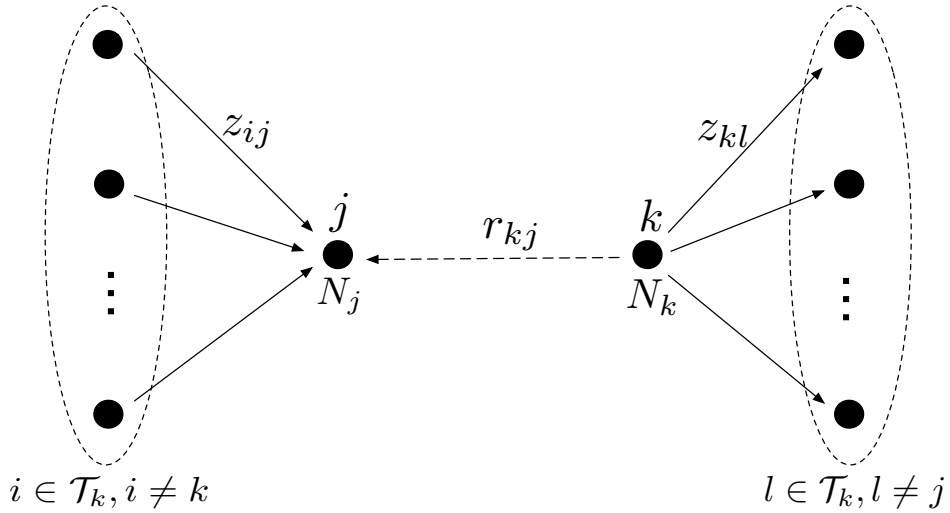


Figure 2.6: An interference link (k, j) in a network.

and 2.7 (with consideration of multiple outgoing data streams from node k and incoming data streams to node j). We have the following constraints:

For every $k \in \mathcal{K},\ j \in \mathcal{I}_k,$ if $x_k(t) = 1$ and $y_j(t) = 1,$ then

$$\begin{cases} d_{kj}^{\mathrm{T}}(t)\mathbf{1}_{kj}^{\mathrm{T}}(t) + d_{kj}^{\mathrm{R}}(t)\mathbf{1}_{kj}^{\mathrm{R}}(t) = \\ \quad \min\left\{ \mathbf{1}_{kj}^{\mathrm{R}}(t) \sum_{l \in \mathcal{T}_k}^{l \neq j} z_{kl}(t) + \mathbf{1}_{kj}^{\mathrm{T}}(t) \sum_{i \in \mathcal{T}_j}^{i \neq k} z_{ij}(t),\ r_{kj}(t) \right\}, & (2.21) \\ \left(\mathbf{1}_{kj}^{\mathrm{R}}(t), \mathbf{1}_{kj}^{\mathrm{T}}(t)\right) \neq (0, 0). & (2.22) \end{cases}$$

Node's DoF Constraints A node can use its DoFs for both SM and IC, as long as the total number of consumed DoFs does not exceed the total available DoFs at the node. If node k is an active Tx node, by Lemmas 2.6 and 2.7, we have

$$\text{If } x_k(t) = 1, \text{ then } \sum_{l \in \mathcal{T}_k} z_{kl}(t) + \sum_{j \in \mathcal{I}_k} d_{kj}^{\mathrm{T}}(t) \leq N_k, \quad k \in \mathcal{K}. \tag{2.23}$$

Similarly, if node j is an active Rx node, we have

$$\text{If } y_j(t) = 1, \text{ then } \sum_{i \in \mathcal{T}_j} z_{ij}(t) + \sum_{k \in \mathcal{I}_j} d_{kj}^{\mathrm{R}}(t) \leq N_j, \quad j \in \mathcal{K}. \tag{2.24}$$

For constraint (2.23), it can be reformulated by incorporating binary variable $x_k(t)$ into the expression as follows:

$$\sum_{l \in \mathcal{T}_k} z_{kl}(t) + \sum_{j \in \mathcal{I}_k} d_{kj}^{\mathrm{T}}(t) \leq N_k x_k(t) + (1 - x_k(t)) \cdot B_k, \quad k \in \mathcal{K}, \tag{2.25}$$

where B_k is a large constant, which can be set as $B_k = \sum_{j \in \mathcal{I}_k} N_j$ to ensure that B_k is an upper bound of $\sum_{j \in \mathcal{I}_k} d_{kj}^{\mathrm{T}}(t)$.

Similarly, constraint (2.24) can be reformulated as follows:

$$\sum_{i \in \mathcal{T}_j} z_{ij}(t) + \sum_{k \in \mathcal{I}_j} d_{kj}^{\mathrm{R}}(t) \leq N_j y_j(t) + (1 - y_j(t)) \cdot B_j, \quad j \in \mathcal{K}, \tag{2.26}$$

where $B_j = \sum_{k \in \mathcal{I}_j} N_k$.

Reformulation of IC Constraints To make the IC constraints suitable for mathematical programming, we need to remove "if-then" statement for (2.21) and (2.22), non-linearity in (2.21), and the joint statement in (2.22). First, we can relax (2.21) by substituting "=" sign by "$\geq$" sign. To remove "if-then" statement for (2.21) and (2.22) and the joint statement in (2.22), we incorporate binary variables $x_k(t)$ and $y_j(t)$ into (2.21) and (2.22), we have:

$$\begin{aligned}&\text{For every } \ k \in \mathcal{K},\ j \in \mathcal{I}_k,\\&\left[2 - x_k(t) - y_j(t)\right] r_{kj}(t) + x_k(t) y_j(t) \left(d_{kj}^{\mathrm{T}}(t)\mathbf{1}_{kj}^{\mathrm{T}}(t) + d_{kj}^{\mathrm{R}}(t)\mathbf{1}_{kj}^{\mathrm{R}}(t)\right)\\&\geq \min\left\{ \mathbf{1}_{kj}^{\mathrm{R}}(t) \sum_{l\in\mathcal{T}_k}^{l\neq j} z_{kl}(t) + \mathbf{1}_{kj}^{\mathrm{T}}(t) \sum_{i\in\mathcal{T}_j}^{i\neq k} z_{ij}(t),\ r_{kj}(t) \right\},\end{aligned} \tag{2.27}$$

$$\mathbf{1}_{kj}^{\mathrm{R}}(t) + \mathbf{1}_{kj}^{\mathrm{T}}(t) \geq x_k(t) + y_j(t) - 1. \tag{2.28}$$

When $x_k(t) = 1$ and $y_j(t) = 1$, one can easily verify that (2.27) is a relaxation of (2.21) because of "$\geq$" sign. Note that such relaxation won't introduce any infeasible DoF allocation for IC. While (2.28) is equivalent to (2.22) by examining all possibilities of $\left(\mathbf{1}_{kj}^{\mathrm{R}}(t), \mathbf{1}_{kj}^{\mathrm{T}}(t)\right)$ other than $(0,0)$. For $x_k(t) \neq 1$ or $y_j(t) \neq 1$ (i.e., link $k \to j$ is not an interference link), (2.27) and (2.28) always hold (i.e., the associated variables are unconstrained for IC).

Next we show how to reformulate the "min" function in (2.27). First, (2.27) is equivalent to

$$\begin{aligned}&\left[2 - x_k(t) - y_j(t)\right] r_{kj}(t) + x_k(t) y_j(t) \left(d_{kj}^{\mathrm{T}}(t)\mathbf{1}_{kj}^{\mathrm{T}}(t) + d_{kj}^{\mathrm{R}}(t)\mathbf{1}_{kj}^{\mathrm{R}}(t)\right)\\&\qquad \geq \mathbf{1}_{kj}^{\mathrm{R}}(t) \sum_{l\in\mathcal{T}_k}^{l\neq j} z_{kl}(t) + \mathbf{1}_{kj}^{\mathrm{T}}(t) \sum_{i\in\mathcal{T}_j}^{i\neq k} z_{ij}(t),\end{aligned} \tag{2.29a}$$

or

$$\begin{aligned}&\left[2 - x_k(t) - y_j(t)\right] r_{kj}(t) + x_k(t) y_j(t) \left(d_{kj}^{\mathrm{T}}(t)\mathbf{1}_{kj}^{\mathrm{T}}(t) + d_{kj}^{\mathrm{R}}(t)\mathbf{1}_{kj}^{\mathrm{R}}(t)\right)\\&\qquad \geq r_{kj}(t).\end{aligned} \tag{2.29b}$$

To remove the "or" statement in (2.29), we introduce a set of binary variables $s_{kj}(t)$, and

(2.29) can be reformulated as

$$\begin{aligned}
&\mathbf{1}^{\mathrm{R}}_{kj}(t)\sum_{l\in\mathcal{T}_k}^{l\neq j} z_{kl}(t) + \mathbf{1}^{\mathrm{T}}_{kj}(t)\sum_{i\in\mathcal{T}_j}^{i\neq k} z_{ij}(t) - \big[2 - x_k(t) - y_j(t)\big] r_{kj}(t) \\
&- x_k(t)y_j(t)\left(d^{\mathrm{T}}_{kj}(t)\mathbf{1}^{\mathrm{T}}_{kj}(t) + d^{\mathrm{R}}_{kj}(t)\mathbf{1}^{\mathrm{R}}_{kj}(t)\right) \leq M_1 \cdot s_{kj}(t),
\end{aligned} \tag{2.30a}$$

$$\begin{aligned}
&r_{kj}(t) - \big[2 - x_k(t) - y_j(t)\big] r_{kj}(t) \\
&- x_k(t)y_j(t)\left(d^{\mathrm{T}}_{kj}(t)\mathbf{1}^{\mathrm{T}}_{kj}(t) + d^{\mathrm{R}}_{kj}(t)\mathbf{1}^{\mathrm{R}}_{kj}(t)\right) \leq M_2 \cdot (1 - s_{kj}(t)),
\end{aligned} \tag{2.30b}$$

where M_1 and M_2 are big constants to ensure M_1 is the upper bound of LHS of (2.30a), and M_2 is the upper bound of LHS of (2.30b). As an example, we can set $M_1 = N_k + N_j$ and $M_2 = r_{kj}(t)$. Therefore, when $s_{kj}(t) = 0$, (2.30a) becomes (2.29a) and (2.30b) holds trivially. Likewise, when $s_{kj}(t) = 1$, (2.30b) becomes (2.29b) and (2.30a) holds trivially.

Now only the non-linear terms in (2.30), i.e., products of variables, need to be reformulated. For this purpose, we employ the *Reformulated-Linearization Technique* (RLT) in [45, 46], which is specifically designed for this purpose. For non-linear terms $x_k(t)y_j(t)\mathbf{1}^{\mathrm{T}}_{kj}(t)$ and $x_k(t)y_j(t)\mathbf{1}^{\mathrm{R}}_{kj}(t)$ in (2.30), they can be linearized by introducing new variables and adding new linear constraints. To do this, define binary variables $\eta_{kj}(t) = x_k(t)y_j(t)\mathbf{1}^{\mathrm{T}}_{kj}(t)$ and $\theta_{kj}(t) = x_k(t)y_j(t)\mathbf{1}^{\mathrm{R}}_{kj}(t)$. Constraint (2.30) can be reformulated as

$$\begin{aligned}
&\mathbf{1}^{\mathrm{R}}_{kj}(t)\sum_{l\in\mathcal{T}_k}^{l\neq j} z_{kl}(t) + \mathbf{1}^{\mathrm{T}}_{kj}(t)\sum_{i\in\mathcal{T}_j}^{i\neq k} z_{ij}(t) - \Big[2 - x_k(t) - y_j(t)\Big] r_{kj}(t) \\
&\qquad - \eta_{kj}(t)d^{\mathrm{T}}_{kj}(t) - \theta_{kj}(t)d^{\mathrm{R}}_{kj}(t) \leq M_1 \cdot s_{kj}(t) \\
&r_{kj}(t) - \Big[2 - x_k(t) - y_j(t)\Big] r_{kj}(t) - \eta_{kj}(t)d^{\mathrm{T}}_{kj}(t) - \theta_{kj}(t)d^{\mathrm{R}}_{kj}(t) \\
&\qquad \leq M_2 \cdot (1 - s_{kj}(t)) \\
&\eta_{kj}(t) \geq x_k(t) + y_j(t) + \mathbf{1}^{\mathrm{T}}_{kj}(t) - 2,
\end{aligned} \tag{2.31}$$

$$
\begin{aligned}
&\eta_{kj}(t) \le \mathbf{1}^{\mathrm{T}}_{kj}(t),\\
&\eta_{kj}(t) \le x_k(t),\\
&\eta_{kj}(t) \le y_j(t),\\
&\theta_{kj}(t) \ge x_k(t) + y_j(t) + \mathbf{1}^{\mathrm{R}}_{kj}(t) - 2,\\
&\theta_{kj}(t) \le \mathbf{1}^{\mathrm{R}}_{kj}(t),\\
&\theta_{kj}(t) \le x_k(t),\\
&\theta_{kj}(t) \le y_j(t).
\end{aligned}
$$

Next, to remove non-linear terms $\eta_{kj}(t)d^{\mathrm{T}}_{kj}(t)$, $\theta_{kj}(t)d^{\mathrm{R}}_{kj}(t)$, $\mathbf{1}^{\mathrm{R}}_{kj}(t)\sum_{l\in\mathcal{T}_k}^{l\neq j} z_{kl}(t)$ and $\mathbf{1}^{\mathrm{T}}_{kj}(t)\sum_{i\in\mathcal{T}_j}^{i\neq k} z_{ij}(t)$ in (2.31), we introduce new variables and add new linear constraints. For $\eta_{kj}(t)d^{\mathrm{T}}_{kj}(t)$, define new variables $\alpha_{kj}(t) = \eta_{kj}(t)d^{\mathrm{T}}_{kj}(t)$. Since $\eta_{kj}(t) \in \{0,1\}$, and $0 \le d^{\mathrm{T}}_{kj}(t) \le N_k$, then the following constraints must hold:

$$
\begin{aligned}
&(\eta_{kj}(t) - 0)\cdot\left(d^{\mathrm{T}}_{kj}(t) - 0\right) \ge 0,\\
&(\eta_{kj}(t) - 0)\cdot\left(N_k - d^{\mathrm{T}}_{kj}(t)\right) \ge 0,\\
&(1 - \eta_{kj}(t))\cdot\left(d^{\mathrm{T}}_{kj}(t) - 0\right) \ge 0,\\
&(1 - \eta_{kj}(t))\cdot\left(N_k - d^{\mathrm{T}}_{kj}(t)\right) \ge 0.
\end{aligned}
\tag{2.32}
$$

Substituting $\alpha_{kj}(t)$ for $\eta_{kj}(t)d^{\mathrm{T}}_{kj}(t)$ in the above constraints, the new constraints among $\alpha_{kj}(t)$, $\eta_{kj}(t)$ and $d^{\mathrm{T}}_{kj}(t)$ are

$$
\begin{aligned}
&\alpha_{kj}(t) \ge 0,\\
&\alpha_{kj}(t) \le N_k \cdot \eta_{kj}(t),\\
&\alpha_{kj}(t) \le d^{\mathrm{T}}_{kj}(t),\\
&\alpha_{kj}(t) \ge N_k \cdot \eta_{kj}(t) + d^{\mathrm{T}}_{kj}(t) - N_k.
\end{aligned}
\tag{2.33}
$$

Similarly, by letting new variables $\beta_{kj}(t) = \theta_{kj}(t)d^{\mathrm{R}}_{kj}(t)$, $\gamma_{kj}(t) = \mathbf{1}^{\mathrm{R}}_{kj}(t)\sum_{l\in\mathcal{T}_k}^{l\neq j} z_{kl}(t)$ and $\delta_{kj}(t) = \mathbf{1}^{\mathrm{T}}_{kj}(t)\sum_{i\in\mathcal{T}_j}^{i\neq k} z_{ij}(t)$, all non-linear terms in (2.31) can be removed with additional linear constraints.

Therefore, (2.30) can be reformulated as a set of mixed integer linear constraints as follows:

$$
\begin{aligned}
&\text{For every } \ k \in \mathcal{K},\ j \in \mathcal{I}_k,\\
&\gamma_{kj}(t) + \delta_{kj}(t) - [2 - x_k(t) - y_j(t)]r_{kj}(t) - \alpha_{kj}(t) - \beta_{kj}(t)\\
&\qquad \leq (N_k + N_j)s_{kj}(t),\\
&r_{kj}(t) - [2 - x_k(t) - y_j(t)]r_{kj}(t) - \alpha_{kj}(t) - \beta_{kj}(t)\\
&\qquad \leq r_{kj}(t)(1 - s_{kj}(t)),\\
&\eta_{kj}(t) \geq x_k(t) + y_j(t) + \mathbf{1}^{\mathrm{T}}_{kj}(t) - 2,\\
&\eta_{kj}(t) \leq \mathbf{1}^{\mathrm{T}}_{kj}(t),\\
&\eta_{kj}(t) \leq x_k(t),\\
&\eta_{kj}(t) \leq y_j(t),\\
&\theta_{kj}(t) \geq x_k(t) + y_j(t) + \mathbf{1}^{\mathrm{R}}_{kj}(t) - 2,\\
&\theta_{kj}(t) \leq \mathbf{1}^{\mathrm{R}}_{kj}(t),\\
&\theta_{kj}(t) \leq x_k(t),\\
&\theta_{kj}(t) \leq y_j(t),\\
&\alpha_{kj}(t) \geq 0,\\
&\alpha_{kj}(t) \leq N_k \cdot \eta_{kj}(t),\\
&\alpha_{kj}(t) \leq d^{\mathrm{T}}_{kj}(t),\\
&\alpha_{kj}(t) \geq N_k \cdot \eta_{kj}(t) + d^{\mathrm{T}}_{kj}(t) - N_k,
\end{aligned}
\tag{2.34}
$$

$$\beta_{kj}(t) \geq 0,$$

$$\beta_{kj}(t) \leq N_j \cdot \theta_{kj}(t),$$

$$\beta_{kj}(t) \leq d_{kj}^{\mathrm{R}}(t),$$

$$\beta_{kj}(t) \geq N_j \cdot \theta_{kj}(t) + d_{kj}^{\mathrm{R}}(t) - N_j,$$

$$\gamma_{kj}(t) \geq 0,$$

$$\gamma_{kj}(t) \leq N_k \cdot \mathbf{1}_{kj}^{\mathrm{R}}(t),$$

$$\gamma_{kj}(t) \leq \sum_{l \in \mathcal{T}_k}^{l \neq j} z_{kl}(t),$$

$$\gamma_{kj}(t) \geq N_k \cdot \mathbf{1}_{kj}^{\mathrm{R}}(t) + \sum_{l \in \mathcal{T}_k}^{l \neq j} z_{kl}(t) - N_k,$$

$$\delta_{kj}(t) \geq 0,$$

$$\delta_{kj}(t) \leq N_j \cdot \mathbf{1}_{kj}^{\mathrm{T}}(t),$$

$$\delta_{kj}(t) \leq \sum_{i \in \mathcal{T}_j}^{i \neq k} z_{ij}(t),$$

$$\delta_{kj}(t) \geq N_j \cdot \mathbf{1}_{kj}^{\mathrm{T}}(t) + \sum_{i \in \mathcal{T}_j}^{i \neq k} z_{ij}(t) - N_j.$$

In summary, (2.17)-(2.20), (2.25), (2.26), (2.28) and (2.34) together constitute a set of feasibility constraints for SM and IC among all nodes in a network.

Existence of Weight Matrices If the above set of feasibility constraints for SM and IC are satisfied among all nodes, then there exists a set of weight matrices that can offer the desired data stream transmission free of interference almost surely. We formally state this result in the following lemma.

Lemma 2.9. *Given that the channels are generic (i.e., the channel matrices are randomly and independently generated from continuous distributions subject to rank-constraints), if*

DoF feasibility constraints (2.17)-(2.20), (2.25), (2.26), (2.28) and (2.34) are satisfied, then there exists a set of weight matrices, with probability 1, such that

(i) the intended data streams are transmitted, i.e.,

$$\begin{aligned}\mathbf{U}_i^{\dagger}\left[\mathbf{H}_{ij_1}\mathbf{V}_{j_1}\cdots\mathbf{H}_{ij_1}\mathbf{V}_{j_Q}\right]&=\mathbf{I}_{z_{i*}},\ \text{if}\ \ z_{ij_1},\cdots,z_{ij_Q}>0,\\ \begin{bmatrix}\mathbf{U}_{i_1}^{\dagger}\mathbf{H}_{i_1j}\\ \vdots\\ \mathbf{U}_{i_P}^{\dagger}\mathbf{H}_{i_Pj}\end{bmatrix}\mathbf{V}_j&=\mathbf{I}_{z_{*j}}\ \text{if}\ \ z_{i_1j},\cdots,z_{i_Pj}>0,\end{aligned}\tag{2.35}$$

and (ii) unintended interferences are cancelled, i.e.,

$$\begin{gathered}\mathbf{U}_k^{\dagger}\mathbf{H}_{kj}\mathbf{V}_j=\mathbf{0},\\ \text{if } k\in\mathcal{K},\ \ j\in\mathcal{I}_k,\ \ z_{k*}>0,\ \ z_{*j}>0 \text{ and } z_{kj}=0.\end{gathered}\tag{2.36}$$

A Proof Sketch. At Rx node j, since all channels are randomly generated from a continuous distribution (i.e., without any special structure), the row spaces of $\mathbf{U}_{i_1}^{\dagger}\mathbf{H}_{i_1j}$, $\mathbf{U}_{i_2}^{\dagger}\mathbf{H}_{i_2j},\cdots,\mathbf{U}_{i_P}^{\dagger}\mathbf{H}_{i_Pj}$ are linearly independent with each other with probability 1. Also, the row space of $\mathbf{U}_{i_n}^{\dagger}\mathbf{H}_{i_nj}$ (for any $i\in\{1,2,\cdots,P\}$) and the row space of $\mathbf{U}_k^{\dagger}\mathbf{H}_{kj}$ are linearly independent almost surely. Similar properties hold for Tx node i. Further, the DoF allocation satisfying constraints (2.17)-(2.20), (2.25), (2.26), (2.28) and (2.34) guarantees that the DoF resources are sufficient for SM and IC at each node. Then following the DoF consumption theory (Theorem 1, Lemmas 1, 2, 3 and 4) that we developed in Section 2.3, there exists a feasible solution (i.e., a feasible $\mathbf{U}_i$ or $\mathbf{V}_j$) at each node satisfying SM and IC constraints (2.35) and (2.36). □

Although Lemma 2.9 guarantees the existence of a feasible set of weight matrices, finding such a set of weight matrices is not trivial from a computational perspective, due to the inter-

dependency among Tx weights and Rx weights. One possible approach to design weight matrices will be given in Chapter 3. The approach in Chapter 3 can find weight matrices that offer desired DoF scheduling and suppress the unwanted interference signal strength close to zero (rather than absolute zero). For practice purposes, this approach is sufficient to meet our needs.

2.6 Case Studies

In this section, we use case studies to demonstrate the DoF regions obtained by our general model and compare them to those obtained by other models. We also apply our general model for DoF scheduling in MIMO networks and demonstrate its efficacy. For ease of reference, we define the following notations for the several models under comparison:

- *Rank-aware shared DoF consumption model, denoted as* $\pi(\mathrm{R}, \mathrm{S})$. This is our general model, where DoF consumption for IC is shared between Tx node and Rx node. This is the most efficient IC model under general channel rank conditions.

- *Rank-aware non-shared DoF consumption model, denoted as* $\pi(\mathrm{R}, \not{\mathrm{S}})$. Under this model, the number of DoFs consumed for IC takes into consideration of channel rank, i.e., the number of DoFs consumed for IC is no greater than the rank of interference channel. IC is done by consuming DoFs *unilaterally* at either Tx node or Rx node, but not both, as in existing models such as [15, 16, 17, 21, 22].

- *Rank-blind non-shared DoF consumption model, denoted as* $\pi(\not{\mathrm{R}}, \not{\mathrm{S}})$. Under this model, channels are considered as full rank even though they are not. IC is done by consuming DoFs *unilaterally* at either Tx node or Rx node, but not both, as in [15, 16, 17, 21, 22].

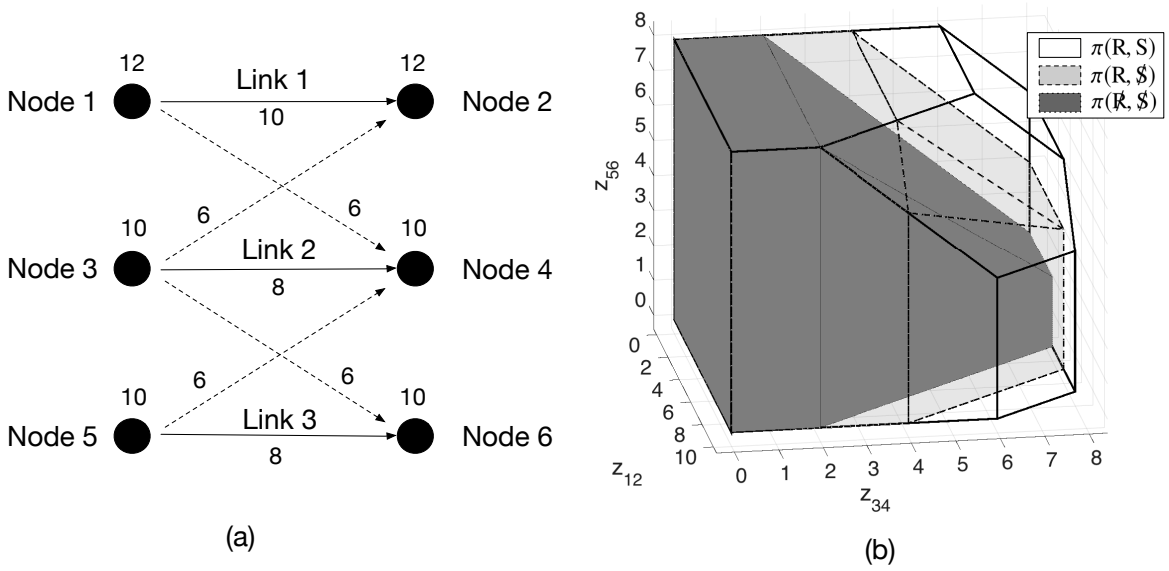


Figure 2.7: A study of DoF region for a three-link network. (a) Transmission and interference topology, number of antennas at each node, and rank of each link. (b) DoF region obtained under different models.

2.6.1 Comparison of DoF Regions

We now study and compare DoF regions for some cases. For the two-link example in Fig. 2.1, we showed that the general DoF model can expand the feasible DoF region. We now consider a few more cases.

Fig. 2.7(a) shows a three-link example, where links 1 and 2 are interfering with each other (in dashed lines) and links 2 and 3 are also interfering with each other (in dash lines). Suppose that the numbers of antennas at Tx nodes 1, 3 and 5 are 12, 10 and 10 while the number of antennas at Rx nodes 2, 4 and 6 are 12, 10 and 10, respectively. Also, suppose the rank of each channel is given as shown in Fig. 2.7(a).

By examining all possible solutions under our general DoF model, the DoF region obtained by $\pi(\mathrm{R}, \mathrm{S})$ is shown as the most outer polyhedron in Fig. 2.7(b). In the same figure, we also show the DoF regions of $\pi(\not{R}, \not{S})$ and $\pi(\mathrm{R}, \not{S})$, both of which are polyhedrons that

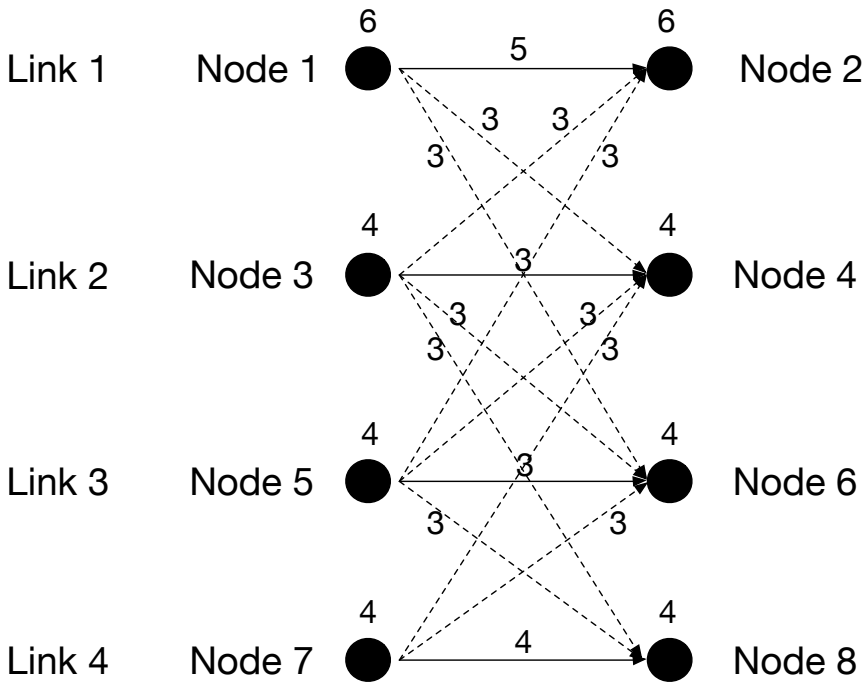


Figure 2.8: A study of DoF region for a four-link network.

are strictly contained inside the DoF region of $\pi(\mathrm{R},\mathrm{S})$. The DoF region by $\pi(\mathrm{R},\mathrm{S})$ is 51.2% and 14.3% larger than those under $\pi(\not{\mathrm{R}},\not{\mathrm{S}})$ and $\pi(\mathrm{R},\not{\mathrm{S}})$, respectively.

Next, we study DoF region for a four-link case as shown in Fig. 2.8, where the dashed lines represent interfering links. The number of antennas and rank of each channel are depicted in Fig. 2.8. Since the solution of DoF region is four-dimensional, which cannot be drawn, we use a table to present our results.

Table 2.2 lists all the boundary points of the DoF regions under models $\pi(\mathrm{R},\mathrm{S})$, $\pi(\mathrm{R},\not{\mathrm{S}})$ and $\pi(\not{\mathrm{R}},\not{\mathrm{S}})$. A boundary point is defined as a feasible point $(z_{12}^*, z_{34}^*, z_{56}^*, z_{78}^*)$ and there exists no other feasible point $(z_{12}, z_{34}, z_{56}, z_{78}) \neq (z_{12}^*, z_{34}^*, z_{56}^*, z_{78}^*)$ such that $z_{12} \geq z_{12}^*, z_{34} \geq z_{34}^*, z_{56} \geq z_{56}^*$ and $z_{78} \geq z_{78}^*$.

From the results in Table 2.2, we find that any boundary point achieved by $\pi(\not{\mathrm{R}},\not{\mathrm{S}})$ is inside the DoF region of $\pi(\mathrm{R},\not{\mathrm{S}})$. Further, $\pi(\mathrm{R},\mathrm{S})$ achieves the largest DoF region. Any

Table 2.2: Boundary points and hypervolumes of DoF regions for a four-link network

	$\pi(\mathrm{R},\mathrm{S})$	$\pi(\mathrm{R},\not{\mathrm{S}})$	$\pi(\not{\mathrm{R}},\not{\mathrm{S}})$
Boundary Points	$(1,2,3,0)$	$(2,1,3,1)$	$(2,1,3,1)$
	$(1,3,2,0)$	$(2,2,2,2)$	$(2,2,2,2)$
	$(2,1,3,1)$	$(2,3,1,1)$	$(2,3,1,1)$
	$(2,2,2,2)$	$(3,0,3,1)$	$(3,0,3,1)$
	$(2,3,1,1)$	$(3,1,2,2)$	$(3,1,2,2)$
	$(3,0,3,2)$	$(3,1,3,0)$	$(3,1,3,0)$
	$(3,1,2,2)$	$(3,2,1,2)$	$(3,2,1,2)$
	$(3,1,3,0)$	$(3,3,0,1)$	$(3,3,0,1)$
	$(3,2,1,2)$	$(3,3,1,0)$	$(3,3,1,0)$
	$(3,2,2,0)$	$(4,0,2,2)$	$(4,0,2,2)$
	$(3,3,0,2)$	$(4,1,1,3)$	$(4,1,1,3)$
	$(3,3,1,0)$	$(4,1,2,1)$	$(4,2,0,2)$
	$(4,0,2,3)$	$(4,2,0,2)$	$(5,0,1,3)$
	$(4,0,3,1)$	$(4,2,1,1)$	$(5,1,0,3)$
	$(4,1,1,3)$	$(5,0,1,3)$	
	$(4,1,2,1)$	$(5,1,0,3)$	
	$(4,2,0,3)$	$(5,1,1,2)$	
	$(4,2,1,1)$		
	$(4,3,0,1)$		
	$(5,0,1,3)$		
	$(5,0,2,2)$		
	$(5,1,0,3)$		
	$(5,1,1,2)$		
	$(5,2,0,2)$		
Hypervolume	77.3	61.5	59.5

boundary point achieved by $\pi(\mathrm{R},\not{\mathrm{S}})$ is inside the DoF region obtained by $\pi(\mathrm{R},\mathrm{S})$. Finally, by using Matlab-based Multi-Parametric Toolbox 3 [47], we can calculate the hypervolumes of these 4-dimensional regions, which we show in Table 2.2. The hypervolumes of the DoF region by $\pi(\mathrm{R},\mathrm{S})$ is 29.8% and 25.8% larger than those by $\pi(\not{\mathrm{R}},\not{\mathrm{S}})$ and $\pi(\mathrm{R},\not{\mathrm{S}})$, respectively.

In the last case study for DoF region, we consider a five-link network with random topology as shown in Fig. 2.9. The transmission and interference links, number of antennas at each node, and rank of each link are shown in the figure. Again, we find DoF regions under $\pi(\mathrm{R},\mathrm{S})$, $\pi(\mathrm{R},\not{\mathrm{S}})$ and $\pi(\not{\mathrm{R}},\not{\mathrm{S}})$, respectively. Table 2.3 show the boundary points and hypervol-

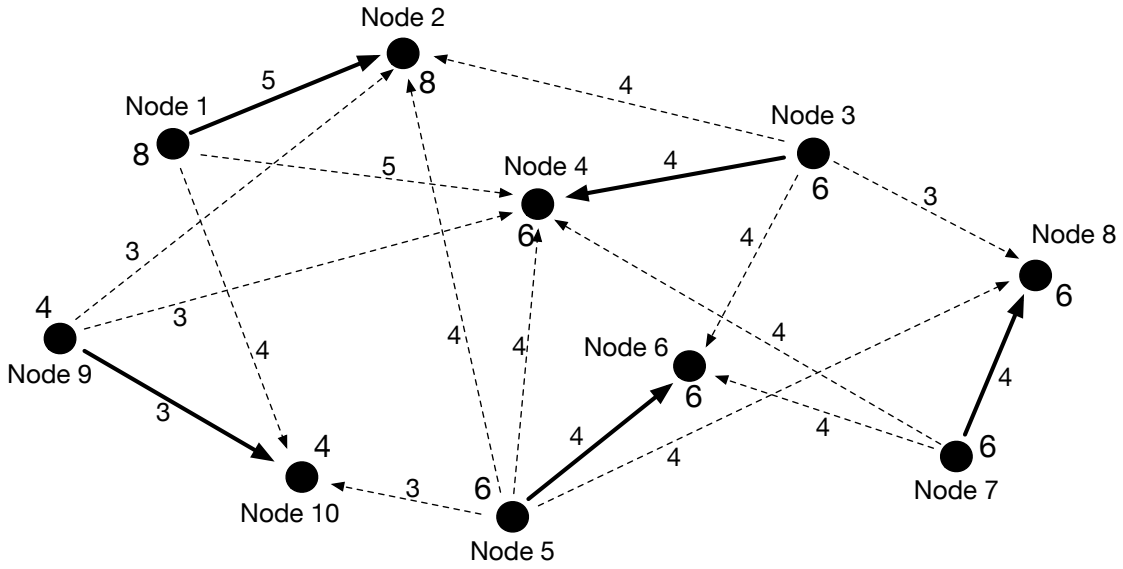


Figure 2.9: A study of DoF region for a five-link network with a random toplogy.

umes achieved under the three DoF IC models. We conclude that $\pi(\mathrm{R}, \mathrm{S})$ offers the largest feasible DoF region.

Table 2.3: Boundary points and hypervolumes of DoF regions for a five-link network with random topology.

	$\pi(\mathrm{R}, \mathrm{S})$	$\pi(\mathrm{R}, \not{\mathrm{S}})$	$\pi(\not{\mathrm{R}}, \not{\mathrm{S}})$
Boundary Points	$(0, 3, 3, 1, 2)$	$(1, 3, 3, 3, 1)$	$(1, 2, 4, 1, 1)$
	$(0, 3, 4, 1, 1)$	$(2, 0, 3, 3, 3)$	$(1, 3, 3, 3, 1)$
	$(0, 4, 3, 1, 1)$	$(2, 0, 4, 2, 2)$	$(2, 0, 3, 3, 3)$
	$(0, 4, 4, 0, 1)$	$(2, 1, 2, 4, 3)$	$(2, 0, 4, 2, 2)$
	$(1, 1, 4, 2, 3)$	$(2, 1, 3, 2, 3)$	$(2, 1, 2, 4, 3)$
	$(1, 2, 3, 3, 2)$	$(2, 1, 3, 3, 2)$	$(2, 1, 3, 2, 3)$

Continue on next page

Table 2.3 (Continued)

	$\pi(\mathrm{R},\mathrm{S})$	$\pi(\mathrm{R},\not{\mathrm{S}})$	$\pi(\not{\mathrm{R}},\not{\mathrm{S}})$
Boundary Points	$(1,3,2,3,2)$	$(2,1,4,1,2)$	$(2,1,3,3,2)$
	$(1,3,3,3,1)$	$(2,2,2,4,2)$	$(2,1,4,1,2)$
	$(1,3,4,0,1)$	$(2,2,4,0,2)$	$(2,2,2,4,2)$
	$(1,3,4,1,0)$	$(2,2,4,2,1)$	$(2,2,4,0,2)$
	$(1,4,1,2,3)$	$(2,3,0,3,3)$	$(2,2,4,2,0)$
	$(1,4,1,3,2)$	$(2,3,1,2,3)$	$(2,3,0,3,3)$
	$(1,4,3,1,0)$	$(2,3,1,3,2)$	$(2,3,1,2,3)$
	$(2,0,4,2,3)$	$(2,3,3,2,1)$	$(2,3,1,3,2)$
	$(2,1,3,3,3)$	$(2,3,3,3,0)$	$(2,3,3,2,1)$
	$(2,2,2,3,3)$	$(2,4,1,2,2)$	$(2,3,3,3,0)$
	$(2,2,3,2,2)$	$(2,4,2,1,2)$	$(2,4,0,2,2)$
	$(2,2,4,1,2)$	$(2,4,2,2,1)$	$(2,4,1,1,2)$
	$(2,2,4,2,1)$	$(3,0,2,4,3)$	$(2,4,2,0,2)$
	$(2,3,1,3,3)$	$(3,0,3,3,2)$	$(2,4,2,1,1)$
	$(2,3,1,4,2)$	$(3,1,2,3,3)$	$(2,4,2,2,0)$
	$(2,3,3,2,1)$	$(3,1,3,2,2)$	$(3,0,2,4,3)$
	$(2,3,3,3,0)$	$(3,1,4,2,1)$	$(3,0,3,3,2)$
	$(2,3,4,0,0)$	$(3,2,1,4,3)$	$(3,1,2,3,3)$
	$(2,4,0,3,2)$	$(3,2,3,1,2)$	$(3,1,2,4,2)$
	$(2,4,1,2,2)$	$(3,2,3,3,1)$	$(3,1,3,2,2)$
	$(2,4,2,1,2)$	$(3,2,4,1,1)$	$(3,1,4,2,1)$
	$(2,4,2,2,1)$	$(3,3,0,3,2)$	$(3,2,1,4,3)$
	$(2,4,3,0,1)$	$(3,3,1,2,2)$	$(3,2,2,4,1)$

Continue on next page

Table 2.3 (Continued)

	$\pi(\mathrm{R},\mathrm{S})$	$\pi(\mathrm{R},\not{\mathrm{S}})$	$\pi(\not{\mathrm{R}},\not{\mathrm{S}})$
	$(3,0,3,3,3)$	$(3,3,2,1,2)$	$(3,2,3,1,2)$
	$(3,0,4,3,2)$	$(3,3,2,3,1)$	$(3,2,3,3,1)$
	$(3,1,2,4,3)$	$(3,3,3,0,1)$	$(3,2,4,0,1)$
	$(3,1,3,2,3)$	$(3,4,1,2,1)$	$(3,2,4,1,0)$
	$(3,1,4,2,2)$	$(3,4,2,1,1)$	$(3,3,0,3,2)$
	$(3,1,4,3,1)$	$(4,0,4,2,1)$	$(3,3,1,2,2)$
	$(3,2,1,4,3)$	$(4,1,3,3,1)$	$(3,3,2,1,2)$
	$(3,2,2,2,3)$	$(4,1,4,1,1)$	$(3,3,2,3,1)$
	$(3,2,2,4,2)$	$(4,2,1,4,2)$	$(3,3,3,0,1)$
Boundary	$(3,2,3,3,1)$	$(4,2,2,2,2)$	$(3,4,1,2,1)$
Points	$(3,2,4,0,2)$	$(4,2,2,4,1)$	$(3,4,2,1,0)$
	$(3,2,4,1,1)$	$(4,2,4,0,1)$	$(4,0,2,4,2)$
	$(3,3,0,4,2)$	$(4,3,1,3,1)$	$(4,0,3,3,1)$
	$(3,3,1,3,2)$	$(4,3,2,1,1)$	$(4,0,4,2,0)$
	$(3,3,2,2,2)$	$(4,4,1,2,0)$	$(4,1,1,4,3)$
	$(3,3,2,3,1)$	$(4,4,2,1,0)$	$(4,1,2,3,2)$
	$(3,3,3,0,2)$	$(5,0,3,3,1)$	$(4,1,2,4,1)$
	$(3,3,3,1,1)$	$(5,1,1,4,3)$	$(4,1,3,2,1)$
	$(3,4,0,4,1)$	$(5,1,2,4,2)$	$(4,1,3,3,0)$
	$(3,4,1,3,1)$	$(5,1,3,2,1)$	$(4,1,4,1,0)$
	$(3,4,2,1,1)$	$(5,2,1,4,1)$	$(4,2,0,4,2)$
	$(3,4,2,2,0)$	$(5,2,2,2,1)$	$(4,2,1,3,2)$
	$(4,0,2,4,3)$	$(5,2,3,1,1)$	$(4,2,1,4,1)$

Continue on next page

Table 2.3 (Continued)

	$\pi(\mathrm{R}, \mathrm{S})$	$\pi(\mathrm{R}, \not{\mathrm{S}})$	$\pi(\not{\mathrm{R}}, \not{\mathrm{S}})$
	$(4,0,3,4,2)$	$(5,3,0,3,0)$	$(4,2,2,2,2)$
	$(4,0,4,2,2)$	$(5,3,1,2,0)$	$(4,2,2,4,0)$
	$(4,0,4,4,1)$	$(5,3,2,1,0)$	$(4,2,3,1,1)$
	$(4,1,2,3,3)$		$(4,2,4,0,0)$
	$(4,1,3,3,2)$		$(4,3,0,3,1)$
	$(4,1,3,4,1)$		$(4,3,1,2,1)$
	$(4,2,1,4,2)$		$(4,3,1,3,0)$
	$(4,2,2,3,2)$		$(4,3,2,1,1)$
	$(4,2,2,4,1)$		$(4,4,0,2,0)$
	$(4,2,3,1,2)$		$(4,4,1,1,0)$
Boundary	$(4,2,4,1,0)$		$(4,4,2,0,0)$
Points	$(4,3,1,4,1)$		$(5,0,1,4,3)$
	$(4,3,2,2,1)$		$(5,0,2,4,1)$
	$(4,3,3,0,1)$		$(5,0,3,3,0)$
	$(4,3,3,1,0)$		$(5,1,0,4,3)$
	$(4,4,0,2,1)$		$(5,1,1,4,2)$
	$(4,4,0,4,0)$		$(5,1,2,3,1)$
	$(4,4,1,1,1)$		$(5,1,2,4,0)$
	$(4,4,1,3,0)$		$(5,1,3,2,0)$
	$(4,4,2,1,0)$		$(5,2,0,4,1)$
	$(5,0,3,3,2)$		$(5,2,1,3,1)$
	$(5,0,3,4,1)$		$(5,2,1,4,0)$
	$(5,0,4,3,1)$		$(5,3,0,3,0)$

Continue on next page

Table 2.3 (Continued)

	$\pi(\mathrm{R}, \mathrm{S})$	$\pi(\mathrm{R}, \not{\mathrm{S}})$	$\pi(\not{\mathrm{R}}, \not{\mathrm{S}})$
Boundary Points	$(5, 1, 1, 4, 3)$		$(5, 3, 1, 2, 0)$
	$(5, 1, 2, 4, 2)$		
	$(5, 1, 3, 2, 2)$		
	$(5, 1, 3, 3, 1)$		
	$(5, 1, 4, 2, 1)$		
	$(5, 2, 0, 4, 2)$		
	$(5, 2, 1, 3, 2)$		
	$(5, 2, 1, 4, 1)$		
	$(5, 2, 2, 3, 1)$		
	$(5, 2, 3, 1, 1)$		
	$(5, 2, 4, 0, 1)$		
	$(5, 3, 0, 3, 1)$		
	$(5, 3, 1, 2, 1)$		
	$(5, 3, 1, 4, 0)$		
	$(5, 3, 2, 2, 0)$		
	$(5, 3, 3, 0, 0)$		
	$(5, 4, 0, 2, 0)$		
	$(5, 4, 1, 1, 0)$		
Hypervolume	664.8	541.1	511.3

2.6.2 DoF Scheduling for Multi-link Networks

To show how the new DoF model ($\pi(\mathrm{R}, \mathrm{S})$) can be used in a multi-link network, we study a throughput maximization problem using the DoF scheduling model developed in Section 2.5. In this study, we choose the objective of maximizing the minimum throughput (c_{min}) among

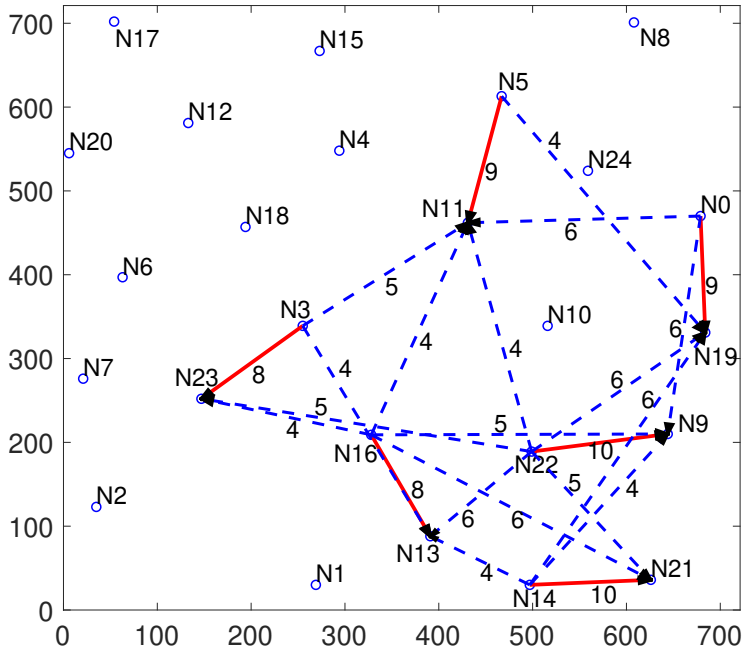


Figure 2.10: Topology of a 25-node network

a set of links $\mathcal{L}$ in a multi-link MIMO network. This objective aims to achieve fairness among the MIMO links while maximizing the number of SM data streams in the network. For ease of exposition, we assume that one data stream corresponds to one unit data rate, and use normalized unit for distance. The transmission and interference ranges are 180 and 360, respectively. The problem formulation becomes a mixed integer linear program (MILP) as follows:

$$\begin{aligned} \text{maximize} \quad & c_{min} \\ \text{s.t.} \quad & \text{Node activity and SM constraints:(2.17)} - \text{(2.20)}; \\ & \text{IC constraints:(2.28), (2.34)}; \\ & \text{Node's DoF constraints:(2.25), (2.26)}. \end{aligned}$$

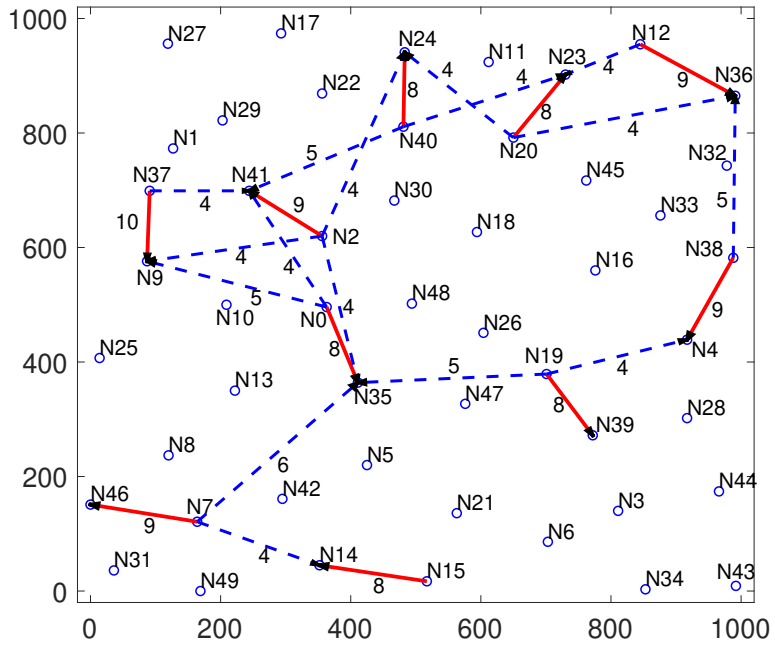


Figure 2.11: Topology of a 50-node network

This MILP problem is NP-hard in general. We use an off-the-shelf solver CPLEX to solve it. CPLEX applies branch-and-cut algorithm to find a solution [48]. In our experiment, it usually takes less than 1 second to obtain an optimal solution.

For the above throughput maximization problem, we consider a 25-node network topology (Fig. 2.10) and a 50-node network topology (Fig. 2.11), respectively. For the randomly generated 25-node network, we assume each node is equipped with 16 antennas. At time t, there are six links transmitting simultaneously. The rank of each transmitting or interfering channel is indicated next to each channel in the figure. The optimal objective value found by CPLEX is 8. The DoF allocation at each active node is given in Table 2.4(a). The number of DoFs consumed for SM at each active node is 8, number of DoF consumed for IC varies, but the total number of DoFs consumed for SM and IC is no more than 16. Table 2.4(b) shows the details of DoF allocation for IC on each interference link. One can easily verify

Table 2.4: DoF scheduling solution for the 25-node network

(a) DoF allocation at each active node

Active Node	Status	DoF Allocation		
		SM	IC	Total
N0	Tx node	8	8	16
N3	Tx node	8	8	16
N5	Tx node	8	4	12
N9	Rx node	8	8	16
N14	Tx node	8	8	16
N11	Rx node	8	6	14
N13	Rx node	8	8	16
N16	Tx node	8	8	16
N19	Rx node	8	8	16
N21	Rx node	8	6	14
N22	Tx node	8	5	13
N23	Rx node	8	4	12

(b) DoF scheduling results for IC

Interference from Tx node i to Rx node j	r_{ij}	$\left(d_{ij}^{\mathrm{T}}, d_{ij}^{\mathrm{R}}\right)$
$i = 0, j = 9$	6	$(2, 4)$
$i = 0, j = 11$	6	$(6, 0)$
$i = 3, j = 11$	5	$(4, 1)$
$i = 3, j = 13$	4	$(4, 0)$
$i = 5, j = 19$	4	$(4, 0)$
$i = 14, j = 9$	4	$(2, 2)$
$i = 14, j = 13$	4	$(2, 2)$
$i = 14, j = 19$	6	$(4, 2)$
$i = 16, j = 9$	5	$(3, 2)$
$i = 16, j = 11$	4	$(0, 4)$
$i = 16, j = 21$	6	$(5, 1)$
$i = 16, j = 23$	4	$(0, 4)$
$i = 22, j = 11$	4	$(3, 1)$
$i = 22, j = 13$	6	$(0, 6)$
$i = 22, j = 19$	6	$(0, 6)$
$i = 22, j = 21$	5	$(0, 5)$
$i = 22, j = 23$	5	$(5, 0)$

that Theorem 2.5 is satisfied for each interference link. In particular, on interference links $0 \to 9, 3 \to 11, 14 \to 9, 14 \to 13, 14 \to 19, 16 \to 9, 16 \to 21$ and $22 \to 11$, DoFs are consumed both at Tx and Rx nodes for IC; on interference links $0 \to 11, 3 \to 13, 5 \to 19$ and $22 \to 23$, DoFs are consumed only at Tx nodes while on interference links $16 \to 11, 16 \to 23, 22 \to 13, 22 \to 19$ and $22 \to 21$, DoFs are consumed only at Rx nodes. In contrast, the objective values achieved by $\pi(\not{\mathrm{R}}, \not{\mathrm{S}})$ and $\pi(\mathrm{R}, \not{\mathrm{S}})$ are 4 and 6, respectively. That is, c_{min} under $\pi(\mathrm{R}, \mathrm{S})$ is 100% and 33.3% more than that under $\pi(\not{\mathrm{R}}, \not{\mathrm{S}})$ and $\pi(\mathrm{R}, \not{\mathrm{S}})$, respectively.

For the randomly generated 50-node network (Fig. 2.11), we assume each node is equipped with 12 antennas. There are 10 concurrently active transmitting links. For this network, we find that the objective value achieved by our model ($\pi(\mathrm{R}, \mathrm{S})$) is 7. The DoF allocation at each active node is given in Table 2.5(a). Table 2.5(b) shows the details of DoF allocation

Table 2.5: DoF scheduling solution of a 50-node network

(a) DoF allocation at each active node

Active Node	Status	DoF Allocation		
		SM	IC	Total
N0	Tx node	7	4	11
N2	Tx node	7	4	11
N4	Rx node	7	4	11
N7	Tx node	7	4	11
N9	Rx node	7	4	11
N12	Tx node	7	4	11
N14	Rx node	7	4	11
N19	Tx node	7	4	11
N20	Tx node	7	4	11
N23	Rx node	7	4	11
N24	Rx node	7	5	12
N30	Tx node	7	4	11
N35	Rx node	7	2	9
N36	Rx node	7	4	11
N37	Tx node	7	4	11
N38	Tx node	7	4	11
N39	Rx node	7	0	7
N40	Tx node	7	5	12
N41	Rx node	7	4	11
N46	Rx node	7	0	7

(b) DoF scheduling results for IC

Interference from Tx node i to Rx node j	r_{ij}	$\left(d_{ij}^{\mathrm{T}}, d_{ij}^{\mathrm{R}}\right)$
$i = 0, j = 9$	5	$(4, 1)$
$i = 0, j = 41$	4	$(0, 4)$
$i = 2, j = 9$	4	$(1, 3)$
$i = 2, j = 24$	4	$(0, 4)$
$i = 2, j = 35$	4	$(3, 1)$
$i = 7, j = 14$	4	$(0, 4)$
$i = 7, j = 35$	6	$(4, 2)$
$i = 12, j = 23$	4	$(4, 0)$
$i = 19, j = 4$	4	$(0, 4)$
$i = 19, j = 35$	5	$(4, 1)$
$i = 20, j = 24$	4	$(3, 1)$
$i = 20, j = 36$	4	$(1, 3)$
$i = 37, j = 41$	4	$(4, 0)$
$i = 38, j = 36$	5	$(4, 1)$
$i = 40, j = 23$	4	$(0, 4)$
$i = 40, j = 41$	5	$(5, 0)$

for IC on each interference link. In contrast, the objective values achieved by $\pi(\not{\mathrm{R}}, \not{\mathrm{S}})$ and $\pi(\mathrm{R}, \not{\mathrm{S}})$ are 4 and 6, respectively. That is c_{min} under $\pi(\mathrm{R}, \mathrm{S})$ is 75% and 16.7% more than those under $\pi(\not{\mathrm{R}}, \not{\mathrm{S}})$ and $\pi(\mathrm{R}, \not{\mathrm{S}})$, respectively.

We have also generated other random topologies and all results are consistent, i.e., c_{min} under $\pi(\mathrm{R}, \mathrm{S})$ is larger than that under $\pi(\not{\mathrm{R}}, \not{\mathrm{S}})$ and $\pi(\mathrm{R}, \not{\mathrm{S}})$. This affirms the significance of using the new DoF model under general channel rank conditions.

2.7 Chapter Summary

Most existing DoF-based models assume channel matrix is of full-rank, which will not hold when more and more antennas are employed at a node and the channel condition is not ideal. This chapter addresses this fundamental limitation in existing DoF-based models by considering general channel rank conditions. We developed a general theory on how DoFs are consumed at Tx and Rx nodes for SM and IC in the presence of rank deficiency. In contrast to common belief developed for full-rank channels, we showed that a shared DoF consumption at both Tx and Rx nodes for IC is most efficient and can achieve a larger feasible DoF region than having only Tx or Rx node consume DoFs unilaterally for IC. We also showed that DoF consumption under the existing full rank assumption is a special case of our DoF model for general channel rank conditions. Based on this understanding, we explored DoF scheduling in a general multi-link MIMO network by developing a set of constraints to characterize a feasible DoF scheduling. Through extensive case studies on DoF regions and DoF scheduling problems, we confirmed the efficacy of the new DoF model for general channel rank conditions. Our findings in this chapter pave the way for further research on MIMO-based wireless networks under general channel rank conditions.

Chapter 3

On DoF Conservation in MIMO Interference Cancellation based on Signal Strength in the Eigenspace

3.1 Introduction

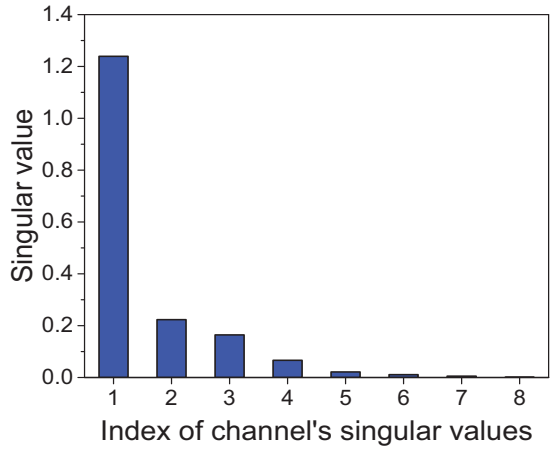
Degree-of-Freedom (DoF) based models have become widely popular in the research community for modeling, analysis, and optimization of MIMO networks [14, 15, 16, 17, 18, 19, 20, 21, 22, 50, 51, 52]. Due to their simple abstraction of MIMO's capabilities in spatial multiplexing (SM) and interference cancellation (IC) [10, 11, 23, 60, 61], a DoF-based model can be used for resource allocation for SM and IC with simple "+/-" arithmetic calculations. By avoiding complex matrix manipulation in resource allocation, DoF-based models are powerful and tractable tools to analyze MIMO's behavior in a network setting.

Under a DoF-based model, the total number of available DoFs at a node is the same as its number of antennas, and a node can use its DoFs for either SM or IC [14, 15, 16, 17, 18, 19, 20, 21, 22, 50, 51, 52]. Existing DoF IC models require to consume DoFs to cancel all interference in the channel, regardless of interference strength in different directions in the eigenspace. However, interference strength varies greatly in different directions in the

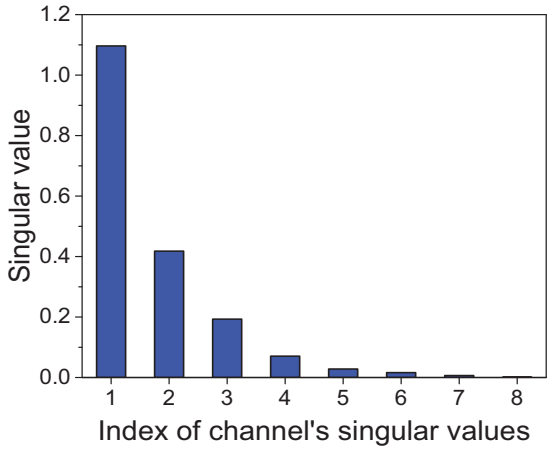
Figure 3.1: A portable 8-antenna wireless testbed.

eigenspace for the same link, as we shall see in the following experiment.

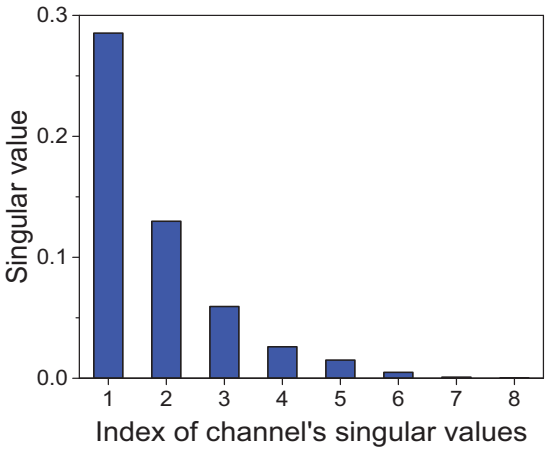
An Experiment We have conducted experiments to examine channel conditions in an indoor environment. In this experiment, we build two nodes to form an 8×8 MIMO channel. Each node is built with 8 USRP N210 devices [57], a OctoClock-G CDA-2990 device [58], a 10 GbE-switch, a desktop computer, and GNU radio software package [59]. The 8 USRP devices is connected to the 10 GbE-switch via CAT5E Ethernet cables and synchronized using the OctoClock-G CDA-2990 device (providing external 1 PPS and 10 MHz reference clock), as shown in Fig. 3.1. We install GNU Radio (in Ubuntu) on the desktop computer to control the USRP devices. Such a MIMO node can achieve 20 MHz of instantaneous bandwidth for wireless signal transmission and reception. We perform a set of experiments under LOS/NLOS and different antenna spacing settings (5 cm or 10 cm) in an indoor environment to measure the MIMO channel matrices. Then we perform singular



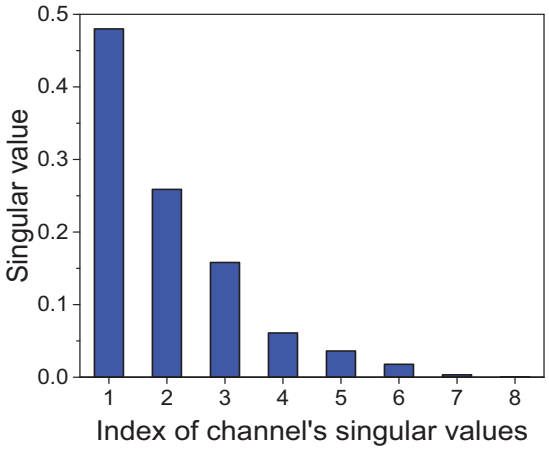
(a) LOS channel with 5 cm antenna spacing



(b) LOS channel with 10 cm antenna spacing



(c) NLOS channel with 5 cm antenna spacing



(d) NLOS channel with 10 cm antenna spacing

Figure 3.2: SVD of an 8×8 MIMO channel in our experiment. Carrier frequency is 5.8 GHz.

value decomposition (SVD) of measured 8×8 MIMO channel matrices. Fig. 3.2 presents the singular values in each direction under different settings. As shown in Fig. 3.2, strictly deficient channel rank (lower than the number of Tx/Rx antennas) can be seen throughout our experiments, i.e., zero or near-zero for the least singular value. More important, in many cases, we observe that the remaining singular values vary greatly. This means signals in some directions are much stronger than the others on the same link. This phenomenon is mainly due to the lack of rich multipath propagation and spatial separations, leading to correlations among the spatial channels within the MIMO link [62, 63]. As a result, the transmit power from a node is generally not uniformly distributed in all directions of the channel's eigenspace. □

Based on our observation from the experiment, we ask the following question: Can we exploit such disparity in singular values (interference signal strength) to conserve DoF in IC?

To answer this question, we must first re-examine state-of-the-art IC strategies in existing DoF models and understand their limitations. Under existing IC schemes, all interference at an interference channel are cancelled at either Tx side or Rx side [14, 15, 16, 17, 18, 19, 20, 21, 22, 50, 51, 52]. The number of DoFs consumed in IC is solely based on the number of interfering data streams, regardless of interference strength in different directions in the eigenspace. That is, given the number of transmitting data streams, the number of DoFs required by IC under a highly correlated interference channel would be exactly the same as that under a channel with uniformly distributed singular values, without any discrimination on channel conditions in different directions. The weakness of such an IC strategy is that it turns a "blind eye" on interference signal strength and considers the impact of a weak interference signal the same as a strong interference signal. As a result, the existing IC models may not utilize DoF resources in the most efficient manner.

In this chapter, we propose to make a major departure from the existing approach for DoF IC. We propose to exploit the differences in interference signal strength among different directions by examining singular values in the eigenspace and propose to expend DoFs only to cancel strong interference. In other words, we want to conserve precious DoFs from cancelling the weaker ones. Specifically, we introduce the concept called "effective rank threshold." If the singular value (i.e., the interference strength at the corresponding direction in the eigenspace) is greater than the threshold, then such interference will be cancelled with DoFs. But if the singular value is smaller than effective rank threshold, it will be treated as noise before IC. Although there might be throughput loss due to un-cancelled weak interference, precious DoFs can be saved to support more data streams which in return improves network throughput. The main contributions of this chapter are summarized as the following:

- This is the first work on DoF IC models that exploits interference signal strengths in the eigenspace. Existing DoF models cancel interference with precious DoFs on all directions in the eigenspace. Instead, we propose to perform IC with DoFs only on those directions with strong signals in the eigenspace.

- We introduced the concept of effective rank threshold to differentiate strong and weak interference in different directions in the eigenspace on an interference link. Based on this effective rank threshold, IC will only be performed for strong interference corresponding to large singular values in the eigenspace, while weak interference will be treated as noise in throughput calculation.

- We investigate the fundamental trade-off between throughput and effective rank threshold, using a general MU-MIMO network. Through simulation results, we show that there exists an optimal trade-off between throughput and effective rank threshold. We show that the network throughput under optimal effective rank threshold setting is

considerably higher than that under existing DoF models.

- To ensure our new IC model is feasible at the PHY layer, we propose an algorithm to determine weights for all Tx and Rx nodes that can offer our desired DoF allocation. Through an iterative process, our algorithm can successfully find the beamforming weights for all Tx and Rx nodes such that the strong interferences beyond the effective rank threshold can be suppressed close to zero, thus ensuring the feasibility of our new IC model.

The remainder of this chapter is organized as follows. In Section 3.2, we use a motivating example to illustrate our new IC idea. Section 3.3 shows how to determine the effective channel rank of a link. In Section 3.4, we present the DoF IC model based on effective channel rank. Section 3.5 analyzes the trade-off among total network throughput, DoFs for SM, and effective channel rank. In Section 3.6, we develop an algorithm that can find Tx and Rx weights at each node to ensure feasibility at PHY layer. In Section 3.7, we review related works on DoF IC models. Section 3.8 concludes this chapter.

3.2 A Motivating Example

In this section, we use a motivating example to illustrate our main idea. Considering a simple two-cell MIMO network shown in Fig. 3.3. There are two APs (AP1 and AP2) and two users (u_1 and u_2). Suppose each node (AP or user) is equipped with 12 antennas. AP1 transmits z_{11} data streams to user u_1 (marked with solid arrow lines) which interfere with user u_2 (marked with dashed arrow lines). Likewise, AP2 transmits concurrently z_{22} data streams to user u_2. For the time being, let's neglect the interference from AP2 to user u_1.[1]

[1]Such weak interference will be considered in throughput calculation (see Section 3.5).

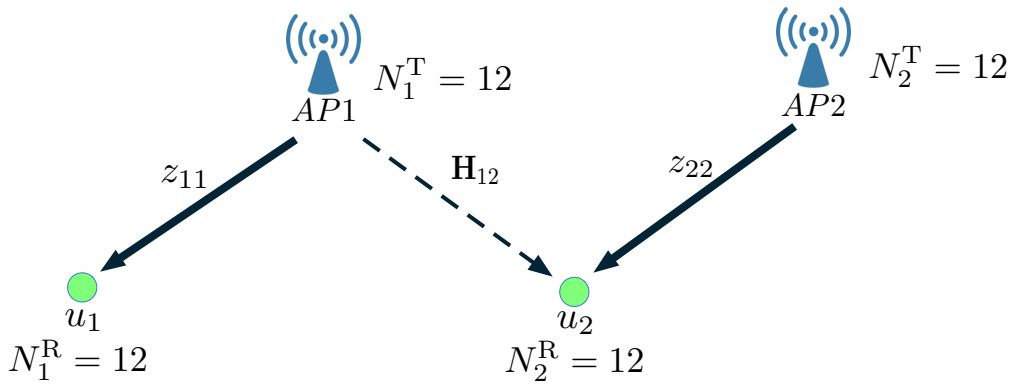


Figure 3.3: A motivating example with two APs and two users.

Consider the interference channel $\mathbf{H}_{12}$ in Fig. 3.3. We use the Kronecker channel model to characterize the channel correlations [63]. We can write $\mathbf{H}_{12}$ as $\mathbf{H}_{12} = \mathbf{R}_{tx}^{1/2}\mathbf{H}_w\mathbf{R}_{rx}^{1/2}$, where $\mathbf{H}_w$ is an 12×12 random matrix with zero-mean i.i.d. complex Gaussian entries, $\mathbf{R}_{tx}^{1/2}$ ($\mathbf{R}_{rx}^{1/2}$) is the 12×12 square root matrix of the transmit (receiver) antenna correlation matrix. The (i, j)-th element in the correlation matrix $\mathbf{R}_{tx}$ and $\mathbf{R}_{rx}$ is calculated as $\rho_{tx}^{|i-j|}$ and $\rho_{rx}^{|i-j|}$, where $\rho_{tx} \in [0, 1)$ and $\rho_{rx} \in [0, 1)$ represent the level of correlation between any two adjacent antennas (in a linear antenna array) at the respective Tx and Rx nodes [64, 65]).

For different values of ρ_{tx} and ρ_{rx}, we can simulate the expectations of singular values $\boldsymbol{\sigma}$ of $\mathbf{H}_{12}^{\dagger}\mathbf{H}_{12}$, which we show in Fig. 3.4. It is easy to see that for any given value of ρ_{tx} and ρ_{rx}, the expectations of singular values vary significantly, which is consistent with our experimental result for the 8×8 MIMO channel case in Fig. 3.2. Here, a high singular value indicates that a large portion of AP1's power is projected into the direction of the corresponding singular vector. Likewise, a close-to-zero singular value indicates a close-to-zero portion of AP1's power is projected into the direction of the corresponding singular vector. When the values of ρ_{tx} and ρ_{rx} increases (i.e., with increased channel correlation), more and more expectations of singular values diminish toward zero.[2]

[2]Apart from correlation, singular values can also be zero due to the presence of "key-hole" effect [30, 31].

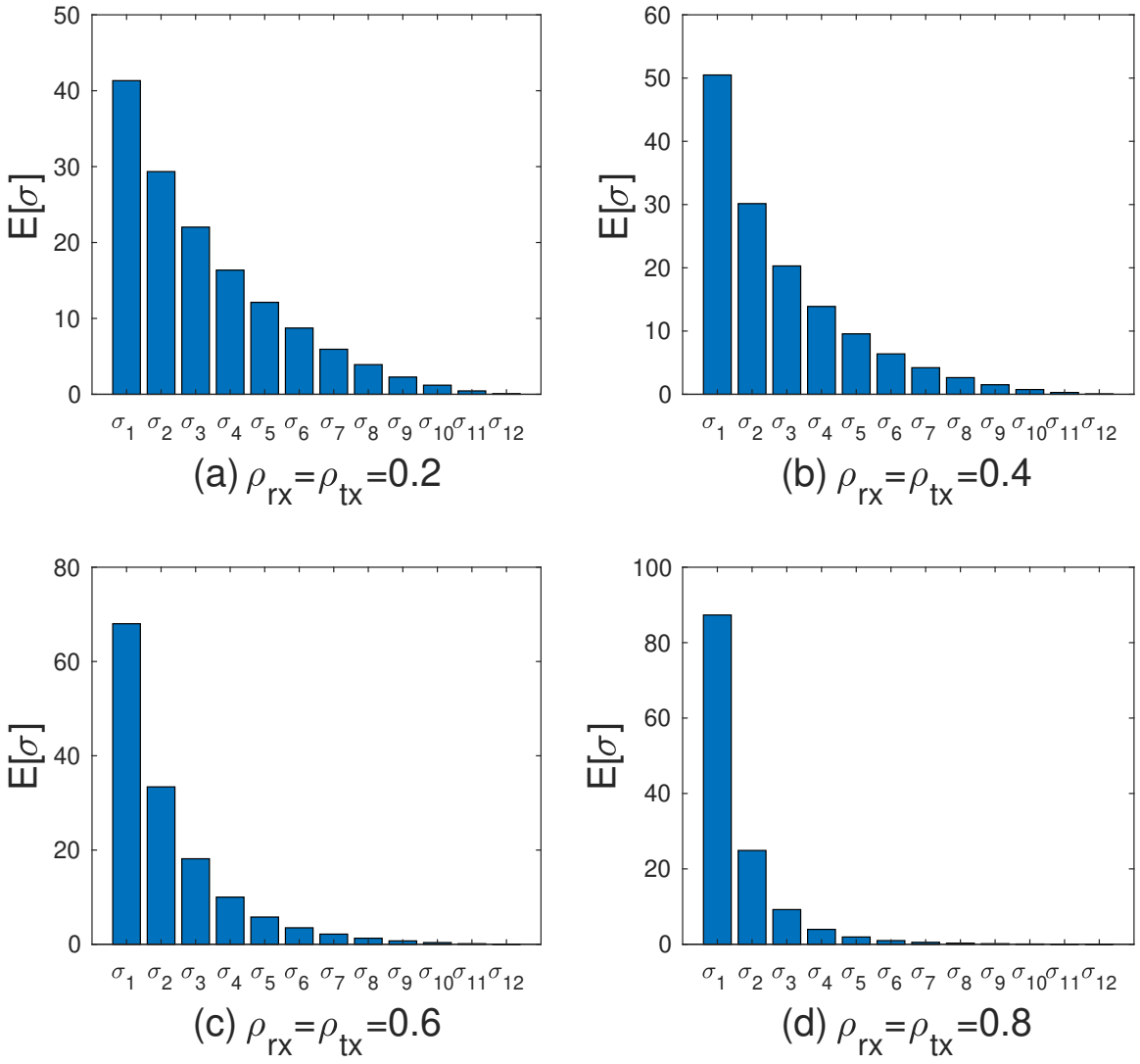


Figure 3.4: Simulation results of expectations of singular values $\mathbb{E}[\sigma]$ under different levels of correlation (ρ_{tx} and ρ_{rx}).

Figure 3.4 suggests that the interference strength varies significantly in different directions in its eigenspace. Under traditional IC scheme (see, e.g., [14, 15, 16, 17, 18, 19, 20, 21, 22, 50, 51, 52]), all interference from AP1 to u_2 shall be cancelled by either AP1 (Tx side, using z_{22} DoFs) or u_2 (Rx side, using z_{11} DoFs). This approach does not differentiate strong and weak interferences in different directions and thus blindly cancels them all with precious DoFs.

To explore this potential opportunity, we propose to exploit the difference of interference power strength in each direction and only cancel the strong interference with DoFs whiling treating the weak ones just as noise. In other words, by exploiting the disparity in interference signal strengths in the eigenspace, we could conserve precious DoFs from cancelling the weaker ones.

Specifically, as shown in Fig. 3.4 ((c) and (d) in particular), the vast majority interference power only appears in the directions corresponding to the high singular values of $\mathbf{H}_{12}$, which can be properly cancelled by using a small number of DoFs. But the remaining weak (small) interference power in these figures is better treated as noise, rather than to be cancelled with precious DoFs. Although there may be some throughput loss due to un-cancelled weak interference, the DoFs savings could be used to transport more data streams (SM). By judiciously exploiting the threshold used to differentiate strong and weak interference, one could achieve a better design objective (e.g., more data streams and/or higher throughput) than blindly cancelling all interferences (weak or strong) with DoFs, as in existing approaches [14, 15, 16, 17, 20, 21, 22, 50, 51, 52].

To show the potential benefits, suppose we set $z_{11} = 12$ in the example in Fig. 3.3. Following traditional IC approach (i.e., no differentiation between strong and weak interferences), AP2 cannot send any data stream to user u_2 as there is no DoF left at user u_2 to cancel interference from AP1. On the other hand, if u_2 treats the interference coming from AP1 in the direction corresponding to the least singular value of $\mathbf{H}_{12}$ as weak interference

and does not use a DoF to cancel it, then it only needs to use 11 DoFs for IC from AP1 to u_2 and use the remaining one to support one data stream transmission from AP2 to u_2. Following the same token, as more interferences from AP1 (corresponding to the least singular values) are treated as weak interferences and thus not to be cancelled with DoFs, more DoFs could be saved and be used to support SM from AP2 to u_2.

As shown in Fig. 3.5(a), by increasing interference threshold η (more on this notation in Section 3.3) to differentiate strong and weak interferences, more DoFs can be conserved from cancelling a fewer number of weak interferences at u_2 and more data streams (SM) can be sent from AP2 to u_2. Fig. 3.5(b) shows the total network throughput (in bits/s/Hz) on all data streams (from AP1 to u_1 and AP2 to u_2) as a function of interference threshold η. Clearly, there is a trade-off among total network throughput, DoFs for SM, and effective channel rank. In particular, there is an optimal knee point that offers the best trade-off between total throughput and effective channel rank (determined by interference threshold η).

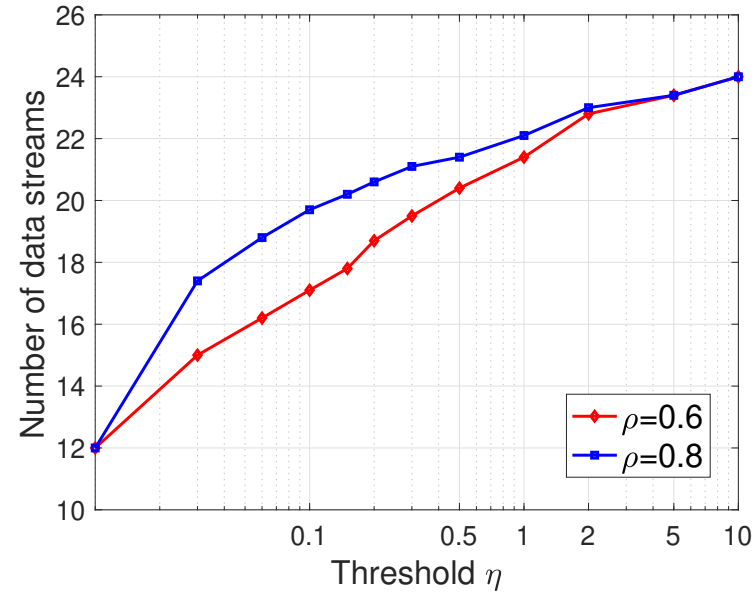
3.3 Determine Effective Channel Rank of a Link

In this section, we present the system model and introduce the concept of "effective channel rank." Consider a general MU-MIMO network (see Fig. 3.6) with a set $\mathcal{K}^{\mathrm{T}}$ of Tx nodes and a set $\mathcal{K}^{\mathrm{R}}$ of Rx nodes, respectively. Each Tx node $i \in \mathcal{K}^{\mathrm{T}}$ and Rx node $j \in \mathcal{K}^{\mathrm{R}}$ are equipped with N_i^{T} and N_j^{R} antennas, respectively. Under MU-MIMO, a Tx node is able to transmit to multiple Rx nodes concurrently while each Rx node can receive from at most one Tx node. For a Tx node $i \in \mathcal{K}^{\mathrm{T}}$, denote $\mathcal{K}_i^{\mathrm{R}}$ as the set of its Rx nodes. For an Rx node $j \in \mathcal{K}^{\mathrm{R}}$, denote $s(j)$ as its source Tx node. Table 3.1 lists key notations in this chapter.

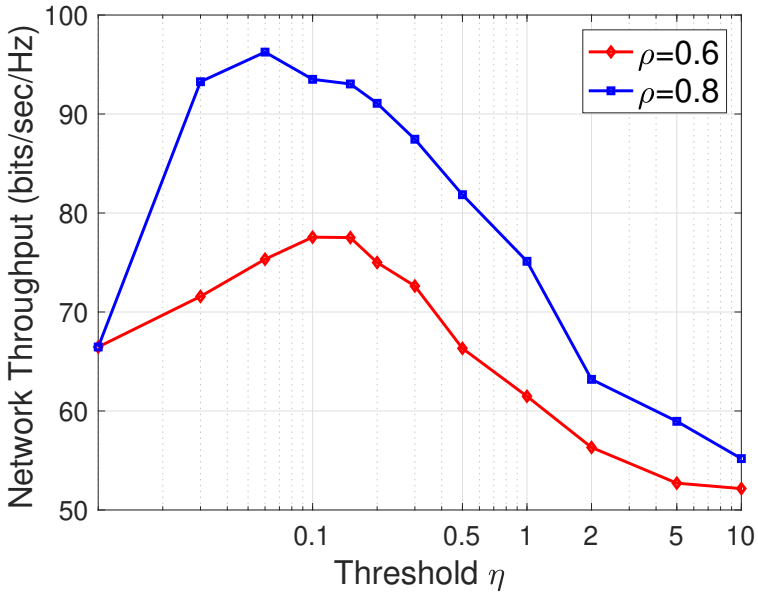
We assume all links in the network are controlled centrally and all channel state infor-

Table 3.1: Notations in Chapter 3

Symbol	Definition
d_{ij}^{R}	Number of DoFs consumed by Rx node j to cancel interference from AP i to Rx node j
d_{ij}^{T}	Number of DoFs consumed by Tx node i to cancel interference from Tx node i to Rx node j
$\mathbf{H}_{ij}$	Channel matrix from Tx node i to Rx node j
$\mathcal{K}^{\mathrm{T}}$	Set of Tx nodes
$\mathcal{K}^{\mathrm{R}}$	Set of Rx nodes
$\mathcal{K}_i^{\mathrm{R}}$	Set of Rx nodes for Tx node i
L_{ij}	Pathloss from Tx node i to Rx node j
N_j^{R}	Number of antennas at Rx node j
N_i^{T}	Number of antennas at Tx node i
P_i	Transmission power at Tx node i
r_{ij}	Effective rank of $\mathbf{H}_{ij}$
$s(j)$	Rx node j's serving Tx node
$\mathbf{U}_i$	Weight matrix at Tx node i
$\mathbf{V}_j$	Weight matrix at Rx node j
z_{i*}	Total number of outgoing data streams at Tx node i
z_{*j}	Total number of incoming data streams at Rx node j
z_{ij}	Number of data streams from Tx node i to Rx node j
η	Normalized effective rank threshold
$\mathbf{X}^{[*f]}$	The f-th column of matrix $\mathbf{X}$



(a)



(b)

Figure 3.5: Total DoFs for SM and throughput performance as a function of threshold setting (used to differentiate strong and weak interferences). (a) Total number of data streams in the network. (b) Network throughput.

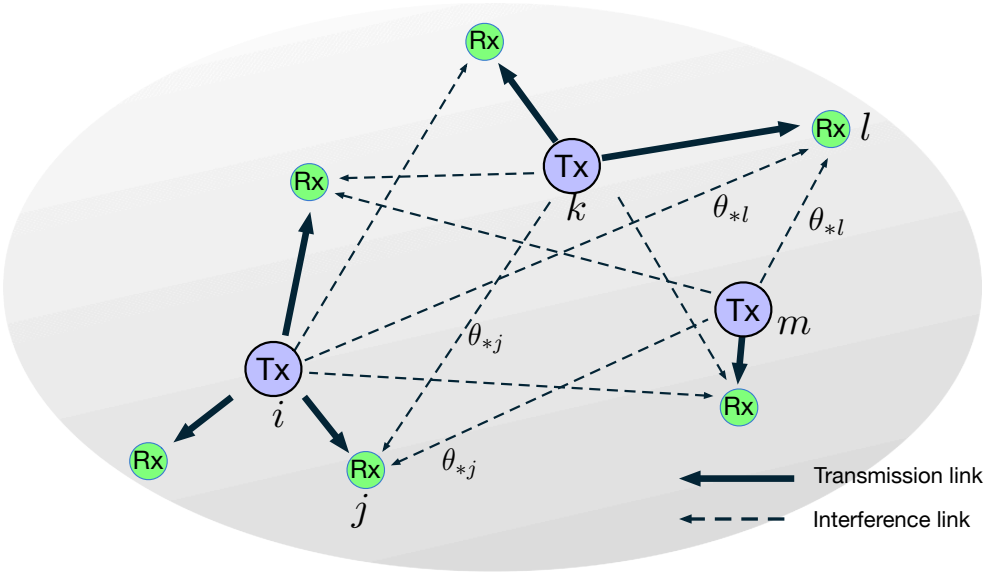


Figure 3.6: A general MU-MIMO network with multiple Tx nodes and Rx nodes.

mation (CSI) is sent to a central controller. CSI can be obtained by either explicit channel feedback and implicit channel feedback [2, 53, 54, 55, 56]. For explicit feedback, the CSI is compressed at each Rx node and then the compressed CSI is sent back to the Tx node. For implicit feedback, we can take advantage of channel reciprocity and use the backward CSI as the forward CSI; channel sounding can be conducted for the backward channel and a relative calibration is performed for each node to maintain channel reciprocity.

3.3.1 Effective Rank of A Single Interference Link

We first differentiate strong and weak interferences on a single interference link and use this differentiation to determine its effective rank. For a single interference link $k \to j$, instead of dealing directly with the fast fading channel matrix $\mathbf{H}_{kj} \in \mathbb{C}^{N_k^{\mathrm{T}} \times N_j^{\mathrm{R}}}$, we take into consideration of transmit power and path loss fading. Denote P_k as the transmit power at Tx node k and L_{kj} as the path loss from Tx node k to Rx node j. Define $\mathbf{Y}_{kj}$ as an $N_j^{\mathrm{R}} \times N_j^{\mathrm{R}}$ symmetric matrix by:

$$\mathbf{Y}_{kj} = \frac{P_k L_{kj}}{N_k^{\mathrm{T}}} \mathbf{H}_{kj}^{\dagger} \mathbf{H}_{kj}, \tag{3.1}$$

where $\mathbf{X}^{\dagger}$ is the conjugate transpose of $\mathbf{X}$. In matrix $\mathbf{Y}_{kj}$, each entry represents the received interference power on the corresponding channel on interference link $k \to j$. We will use $\mathbf{Y}_{kj}$ to determine the effective rank of interference link $k \to j$.

To differentiate strong and weak interferences, we employ the so-called *best rank-r approximation* of a matrix [66]. Under this approximation, $\mathbf{Y}_{kj}$ is decomposed through an SVD process and we retain only the first r largest singular values and their corresponding singular vectors and use them as an approximation.

Fact 1. For a matrix $\mathbf{A} \in \mathbb{C}^{m \times n} (m \geq n)$, denote $\tilde{\mathbf{A}}$ as a rank-r matrix approximation of $\mathbf{A}$ with $r \in \{1, 2, \cdots, n\}$. The optimal solution to minimum approximation error

$$\min_{\tilde{\mathbf{A}} \in \mathbb{C}^{m \times n}} \quad \left|\left| \mathbf{A} - \tilde{\mathbf{A}} \right|\right|_F, \qquad s.t. \;\; rank(\tilde{\mathbf{A}}) = r \tag{3.2}$$

where $|| \cdot ||_F$ denotes Frobenius norm, is

$$\tilde{\mathbf{A}} = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^{\dagger},$$

where σ_i, $\boldsymbol{u}_i$, and $\boldsymbol{v}_i$ are singular value, left and right singular vectors respectively from the SVD of $\mathbf{A}$, i.e., $\mathbf{A} = \sum_{i=1}^{n} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^{\dagger}$ and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$. The minimum approximation error (i.e., optimal objective value for (3.2)) is $\sqrt{\sum_{i=r+1}^{n} \sigma_i^2}$.

The SVD process in Fact 1 clearly shows the relative strength of interferences in different directions. The larger the singular value is, the stronger the interference in that direction. Based on the desired level of approximation error, we can approximate a rank-n matrix $\mathbf{A}$ by a rank-r matrix $\tilde{\mathbf{A}}$ with the r-strongest singular values of $\mathbf{A}$ through (1).

To apply best rank-r approximation on a single interference link $\mathbf{Y}_{kj}$, define θ as a threshold for singular values and denote r_{kj} as the effective channel rank of $\mathbf{H}_{kj}$. Then r_{kj} is given by

$$r_{kj} = \sum_{l=1}^{N_j^{\mathrm{R}}} \mathbb{1}\left\{\sigma_l(\mathbf{Y}_{kj}) \geq \theta\right\}, \tag{3.3}$$

where $\sigma_l(\mathbf{Y}_{kj})$ is the l-th singular value based on SVD of $\mathbf{Y}_{kj}$, and $\mathbb{1}\{\text{event}\}$ is an indicator function, which is 1 if event is true and 0 otherwise.

3.3.2 Interference Threshold at an Rx Node

Note that in a network with a set $\mathcal{K}^{\mathrm{T}}$ of Tx nodes and a set $\mathcal{K}^{\mathrm{R}}$ of Rx nodes, the interference threshold θ in (3.3) should be dependent upon the Rx node of this interference link. This is because the received intended signal power (from its intended transmitter) differs at each Rx node. As an example, consider Rx nodes j and l in Fig. 3.6. Rx node j is closer to its (intended) Tx node i than Rx node l to its (intended) Tx node k. For the same transmit power at i and k, Rx node j will receive a higher signal power (from its intended transmitter) and could thus tolerate a stronger interference. Then, for the interference links at Rx node j ($k \to j$ and $m \to j$), the threshold used to differentiate strong and weak interference should be larger than that used to differentiate stronger and weak interference on interference links ($i \to l$ and $m \to l$) for Rx node l. Based on the above discussion, for an Rx node j, denote θ_{*j} as the threshold for singular values on its interference link. Then we should have $\theta_{*j} > \theta_{*l}$.

In this chapter, instead of optimizing the settings of θ_{*j} for each individual Rx node j based on its (intended) received power level at Rx node j, we introduce a common scaling factor η across all receive nodes to normalize its received power and only optimize the setting of this scaling factor for the entire network. We define η as follows:

$$\theta_{*j} = \eta \frac{P_{s(j)} L_{s(j)j}}{N^{\mathrm{T}}_{s(j)}}.$$

Based on this definition of common scaling factor η, the effective rank r_{kj} of $\mathbf{H}_{kj}$ can be determined by the number of $\mathbf{Y}_{kj}$'s singular values that are greater than or equal to the threshold $\eta \frac{P_{s(j)} L_{s(j)j}}{N^{\mathrm{T}}_{s(j)}}$. That is,

$$r_{kj} = \sum_{l=1}^{N_j} \mathbb{1} \left\{ \sigma_l(\mathbf{Y}_{kj}) \geq \eta \frac{P_{s(j)} L_{s(j)j}}{N^{\mathrm{T}}_{s(j)}} \right\}, \; k \in \mathcal{K}^{\mathrm{T}}, j \in \mathcal{K}^{\mathrm{R}}, j \notin \mathcal{K}^{\mathrm{R}}_k. \tag{3.4}$$

Note that any negligible interference for IC will be treated as noise in the throughput calculation (see Section 3.5).

3.3.3 Effective Rank of An SM Link

For SM from node i to node j (intended transmission), the effective channel rank of $\mathbf{H}_{ij}$ can be determined by

$$r_{ij} = \sum_{l=1}^{N^{\mathrm{R}}_j} \mathbb{1} \left\{ \sigma_l \left(\mathbf{H}^{\dagger}_{ij} \mathbf{H}_{ij} \right) \geq \theta_{\mathrm{SM}} \right\}, \qquad i \in \mathcal{K}^{\mathrm{T}}, j \in \mathcal{K}^{\mathrm{R}}_i,$$

where θ_{SM} is the rank threshold for singular values on SM link $i \to j$. Note that the DoF savings by exploiting strong and weak interference can be made available for SM (more independent data streams) or diversity, both of which have the potential to increase the throughput. To focus on using DoFs for IC at interference links, we do not explore SM-diversity trade-off in this chapter. Therefore, we will try to transmit more data streams as long as we have DoFs available for SM and assume θ_{SM} is a given constant throughout the chapter.

3.4 IC Based on Effective Channel Rank

In the last section, we showed how to differentiate strong and weak interference at an Rx node by setting a threshold for singular value and use this threshold to determine effective channel rank. In this section, we show how to perform IC (for strong interference only) in an MU-MIMO network based on this effective channel rank.

Note that DoF allocation for IC cannot be done arbitrarily and must follow certain rules to be feasible. By "feasible", we mean that all the strong interference can be cancelled at the PHY layer. Section 3.6 will present details on PHY layer feasibility for our DoF allocation.

If DoF allocation for IC and SM is feasible at the PHY layer, then multiple data streams can be transmitted concurrently while all strong interference under best rank-r channels is cancelled. The remaining (un-cancelled) weak interference will be treated as noise and included in the throughput calculation in Section 3.5.

We employ the DoF-based IC model in Chapter 2 to perform DoF allocation. In Chapter 2, the rank of a channel is assumed to be given *a priori*. But in this chapter, the rank of a channel is a function of effective rank threshold.

3.4.1 Modeling of DoF Constraints

DoF Constraints for SM For an intended transmission from Tx node i to Rx node j, denote the number of data streams on this link as z_{ij}. Denote $x_i(t)$ as a binary variable to indicate whether Tx node i is active or not at time t, i.e., $x_i(t) = 1$ if Tx node i is transmitting at time t and 0 otherwise. Likewise, denote $y_j(t)$ as a binary variable to indicate whether Rx node j is active or not at time t, i.e., $y_j(t) = 1$ if Rx node j is receiving at time t and 0 otherwise.

If Tx node i is transmitting, then the total number of data streams transmitted to different receivers (under MU-MIMO) cannot exceed the total number of antennas at node i (i.e., N_i^{T}). We have

$$x_i(t) \leq \sum_{j \in \mathcal{K}_i^{\mathrm{R}}} z_{ij}(t) \leq N_i^{\mathrm{T}} x_i(t), \quad i \in \mathcal{K}^{\mathrm{T}}. \tag{3.5}$$

Similarly, if Rx node j is active at time t, then the total number of DoFs used for reception (from only one transmitter under MU-MIMO) cannot exceed the number of antennas at node j (i.e., N_j^{R}). We have

$$y_j(t) \leq z_{ij}(t) \leq N_j^{\mathrm{R}} y_j(t), \quad i \in \mathcal{K}^{\mathrm{T}}, j \in \mathcal{K}_i^{\mathrm{R}}. \tag{3.6}$$

Taking into consideration of the effective rank of the SM link $i \to j$, the number of data streams that can be sent on this SM link cannot exceed the link's effective rank (see Section 3.3). We have

$$z_{ij}(t) \leq r_{ij}(t), \quad i \in \mathcal{K}^{\mathrm{T}}, \ j \in \mathcal{K}_i^{\mathrm{R}}. \tag{3.7}$$

For Rx node l that is not Tx node i's intended receiver, i.e., $l \notin \mathcal{K}_i^{\mathrm{R}}$, the transmission at Tx node i is considered interference (instead of SM) and there is zero data streams over this link. We have

$$z_{il}(t) = 0, \quad k \in \mathcal{K}^{\mathrm{T}}, l \in \mathcal{K}^{\mathrm{R}}, l \notin \mathcal{K}_i^{\mathrm{R}}. \tag{3.8}$$

DoF Constraints for IC For interference from Tx node k to Rx node j, denote $d_{kj}^{\mathrm{T}}(t)$ as

the number of consumed DoFs at Tx node k and $d_{kj}^{\mathrm{R}}(t)$ as the number of consumed DoFs at Rx node j that are needed to cancel this interference. Based on Chapter 2, a collaborative DoF consumption at both interfering Tx node k and Rx node j is the most efficient approach for IC when the rank of the interference channel is not full, as in our case. Denote $\mathbf{1}_{kj}^{\mathrm{T}}$ and $\mathbf{1}_{kj}^{\mathrm{R}}$ as two binary variables to indicate whether Tx node i (or Rx node j) consumes any DoFs for IC from k to j. That is, $\mathbf{1}_{kj}^{\mathrm{T}} = 1$ if Tx node k consumes DoFs for IC from k to j, $\mathbf{1}_{kj}^{\mathrm{T}} = 0$ otherwise; $\mathbf{1}_{kj}^{\mathrm{R}} = 1$ if Rx node j consumes DoFs for IC from k to j, $\mathbf{1}_{kj}^{\mathrm{R}} = 0$ otherwise.

If $x_k(t) = 1$ and $y_j(t) = 1$, then

$$\begin{aligned} &d_{kj}^{\mathrm{T}}(t)\mathbf{1}_{kj}^{\mathrm{T}}(t) + d_{kj}^{\mathrm{R}}(t)\mathbf{1}_{kj}^{\mathrm{R}}(t) = \\ &\min\left\{ \mathbf{1}_{kj}^{\mathrm{R}}(t) \sum_{l\in\mathcal{K}_k^{\mathrm{R}}}^{l\neq j} z_{kl}(t) + \mathbf{1}_{kj}^{\mathrm{T}}(t) \sum_{i\in\mathcal{K}^{\mathrm{T}}}^{i\neq k} z_{ij}(t),\ r_{kj}(t) \right\}, \end{aligned} \tag{3.9a}$$

$$\left(\mathbf{1}_{kj}^{\mathrm{T}}(t), \mathbf{1}_{kj}^{\mathrm{R}}(t)\right) \neq (0,0),\ k \in \mathcal{K}^{\mathrm{T}}, j \in \mathcal{K}^{\mathrm{R}} \tag{3.9b}$$

That is, the interference from k to j can be cancelled by consuming DoFs on Tx node k only (when $\left(\mathbf{1}_{kj}^{\mathrm{T}}(t), \mathbf{1}_{kj}^{\mathrm{R}}(t)\right) = (1,0)$), Rx node only (when $\left(\mathbf{1}_{kj}^{\mathrm{T}}(t), \mathbf{1}_{kj}^{\mathrm{R}}(t)\right) = (0,1)$), or both Tx and Rx nodes (when $\left(\mathbf{1}_{kj}^{\mathrm{T}}(t), \mathbf{1}_{kj}^{\mathrm{R}}(t)\right) = (1,1)$). Constraint (3.9) can be reformulated as mixed integer linear (MIL) constraints, which is omitted here to conserve space.

DoF Constraints at A Node A node can use its DoFs for SM and/or IC, as long as the total number of consumed DoFs does not exceed the total available DoFs at the node. We consider DoF constraints at Tx and Rx nodes separately. If node i is an active Tx node, we have

$$\text{if } x_i(t) = 1,\ \text{then} \sum_{j\in\mathcal{K}_i^{\mathrm{R}}} z_{ij}(t) + \sum_{l\in\mathcal{K}^{\mathrm{R}}} d_{il}^{\mathrm{T}}(t)\mathbf{1}_{il}^{\mathrm{T}}(t) \leq N_i^{\mathrm{T}},\quad i \in \mathcal{K}^{\mathrm{T}}. \tag{3.10}$$

If node j is an active Rx node, we have

$$\text{if } y_j(t) = 1, \text{ then } \sum_{i \in \mathcal{K}^{\mathrm{T}}} z_{ij}(t) + \sum_{k \in \mathcal{K}^{\mathrm{T}}} d^{\mathrm{R}}_{kj}(t) \mathbf{1}^{\mathrm{R}}_{kj}(t) \leq N^{\mathrm{R}}_j, \quad j \in \mathcal{K}^{\mathrm{R}}. \tag{3.11}$$

For constraint (3.10), it can be reformulated by incorporating binary variable $x_i(t)$ into the expression as follows:

$$\sum_{j \in \mathcal{K}^{\mathrm{R}}_i} z_{ij}(t) + \sum_{l \in \mathcal{K}^{\mathrm{R}}} d^{\mathrm{T}}_{il}(t) \mathbf{1}^{\mathrm{T}}_{il}(t) \leq N^{\mathrm{T}}_i x_i(t) + (1 - x_i(t))B, \quad i \in \mathcal{K}^{\mathrm{T}}, \tag{3.12}$$

where B is a large constant, which can be set as $B = \sum_{i \in \mathcal{K}^{\mathrm{T}}} N^{\mathrm{T}}_i + \sum_{j \in \mathcal{K}^{\mathrm{R}}} N^{\mathrm{R}}_j$ to ensure that B is an upper bound of $\sum_{l \in \mathcal{K}^{\mathrm{R}}} d^{\mathrm{T}}_{il}(t)$.

Similarly, constraint (3.11) can be reformulated as follows:

$$\sum_{i \in \mathcal{K}^{\mathrm{T}}} z_{ij}(t) + \sum_{k \in \mathcal{K}^{\mathrm{T}}} d^{\mathrm{R}}_{kj}(t) \mathbf{1}^{\mathrm{R}}_{kj}(t) \leq N^{\mathrm{R}}_j y_j(t) + (1 - y_j(t))B, \quad j \in \mathcal{K}^{\mathrm{R}}. \tag{3.13}$$

Constraints (3.12) and (3.13) can be reformulated as mixed integer linear constraints, which are omitted here to conserve space.

3.4.2 An Example

As an example to illustrate the relationship between total achievable data streams (SM) in the network and η (the common scaling factor to differentiate strong and weak interference and effective channel rank), consider the simple MU-MIMO network in Fig. 3.7. Suppose our objective is to maximize the sum of log of all data streams (SM) in the network with the consideration of fairness [68]. Then we have the following optimization problem:

$$\begin{aligned} \max \quad & \sum_{i\in\mathcal{K}^{\mathrm{T}}}\sum_{j\in\mathcal{K}^{\mathrm{R}}} \log(z_{ij}) \\ \text{s.t.} \quad & \text{SM constraints: } (3.5)-(3.8); \\ & \text{IC constraints: } (3.9); \\ & \text{Node's DoF constraints: } (3.12), (3.13), \end{aligned}$$

where $z_{ij}, d_{kj}^{\mathrm{T}}, d_{kj}^{\mathrm{R}}, \mathbf{1}_{kj}^{\mathrm{T}}$ and $\mathbf{1}_{kj}^{\mathrm{R}}$ are variables while all other symbols are constants.

As discussed earlier, the constraints in the above formulation can be reformulated into mixed integer linear constraints. However, the objective function (sum of log) remains non-linear. Fortunately, the sum of log objective can be reformulated (along with the MIL constraints) as a second order conic program (SOCP) [69]. Off-the-shelf optimization tools, such as Gurobi [70], can solve this SOCP (with integer variables) optimally.

Some numerical results follow. Suppose the six Tx nodes in Fig. 3.7 are uniformly generated in a $400m \times 400m$ space, with a minimum of $90m$ distance between every two nodes. For each Tx node, there are two Rx nodes uniformly generated with a radius of $70m$ of the Tx node. Unless otherwise, all parameters are fixed as follows. Each Tx and Rx nodes are equipped with 16 and 12 antennas, respectively. Assume a fixed (constant) transmit power for each Tx node i, with SNR $P_i/n_0^2 = 80$ dB, where n_0^2 is the white noise power. Path loss is modeled as $L_{ij} = D_{ij}^{-3}$, with D_{ij} being the distance between Tx node i and Rx node j. Fast fading is modeled by Kronecker channel model, i.e., $\mathbf{H}_{ij} = \mathbf{R}_{tx}^{1/2}\mathbf{H}_w\mathbf{R}_{rx}^{1/2}$, where $\mathbf{R}_{rx}^{1/2}$ is an $N_j^{\mathrm{R}} \times N_j^{\mathrm{R}}$ matrix with each entry containing square root of the receive antenna correlation while $\mathbf{R}_{tx}^{1/2}$ is an $N_i^{\mathrm{T}} \times N_i^{\mathrm{T}}$ matrix with each entry containing square root of the transmit antenna correlation. $\mathbf{H}_w$ is an $N_i^{\mathrm{T}} \times N_j^{\mathrm{R}}$ random matrix with its entries containing zero-mean i.i.d. complex Gaussian random numbers. The (k,l)-th element of the correlation matrix

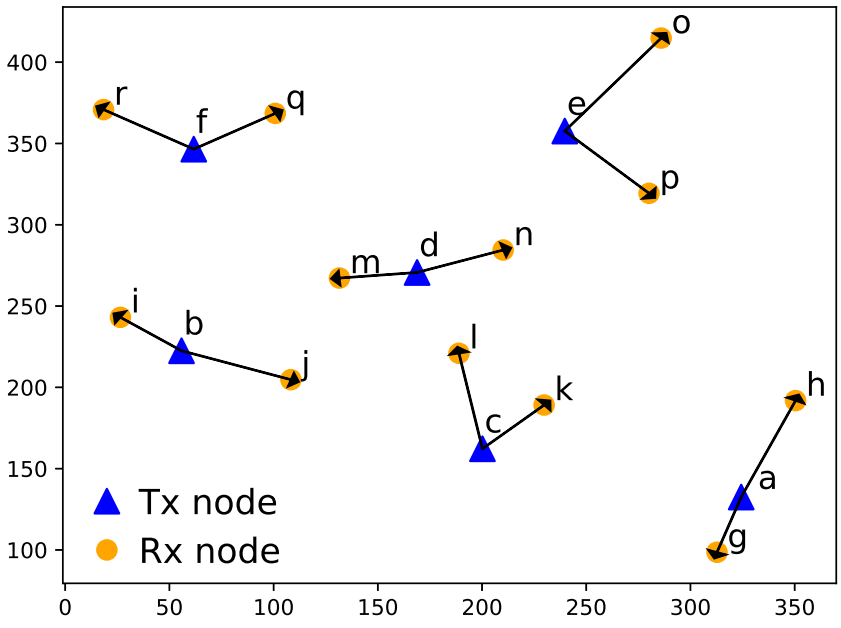


Figure 3.7: An instance of MU-MIMO network topology.

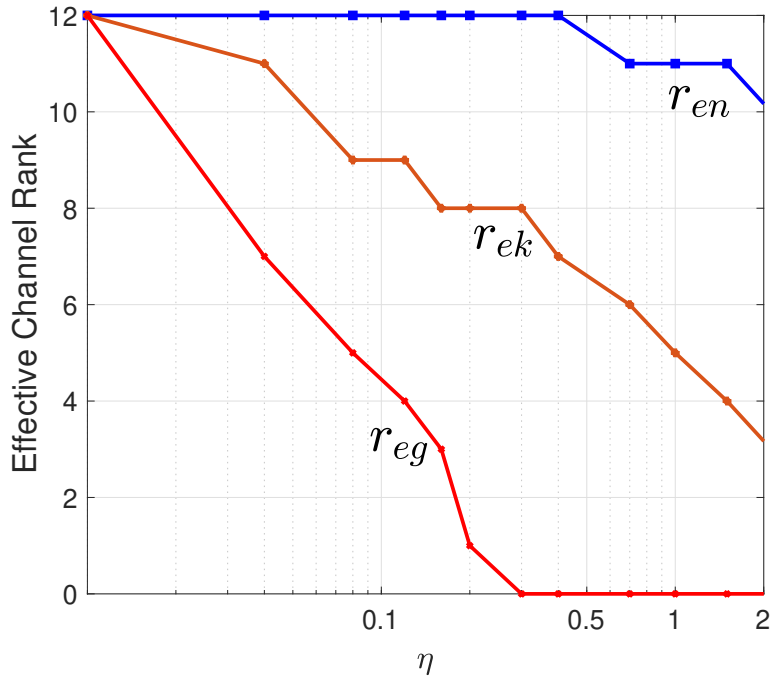
$\mathbf{R}_{rx}$ and $\mathbf{R}_{tx}$ is taken here as $\rho^{|k-l|}$ with $\rho \in \{0.2, 0.4, 0.6\}$. The rank threshold for SM links θ_{SM} is set to be 1.

Fig. 3.8 shows the effective ranks on three representative links ($e \to n, e \to k$ and $e \to g$) as a function of rank threshold scaling factor η (in log scale). We draw η in log scale since singular value distribution is more like a log-shape other than a linear shape (see Fig. 3.4). As expected, all effective channel ranks are decreasing steadily. For $\rho = 0.2$ shown in Fig. 3.8(a), note that r_{en} remains full rank until η becomes greater than 0.4 while r_{ek} and r_{eg} starts to decrease when η starts to increase from 0. This is because Rx node n is close to the interfering Tx node e than k and g and thus experience much stronger interference from Tx node e than k and g. On the other hand, r_{eg} drops very fast because Rx node g is further away from Tx node e than n and k. When η is greater than 0.3, $r_{eg} = 0$ and Rx node g is considered out of interference range of Tx node e. For $\rho = 0.4$ shown in Fig.

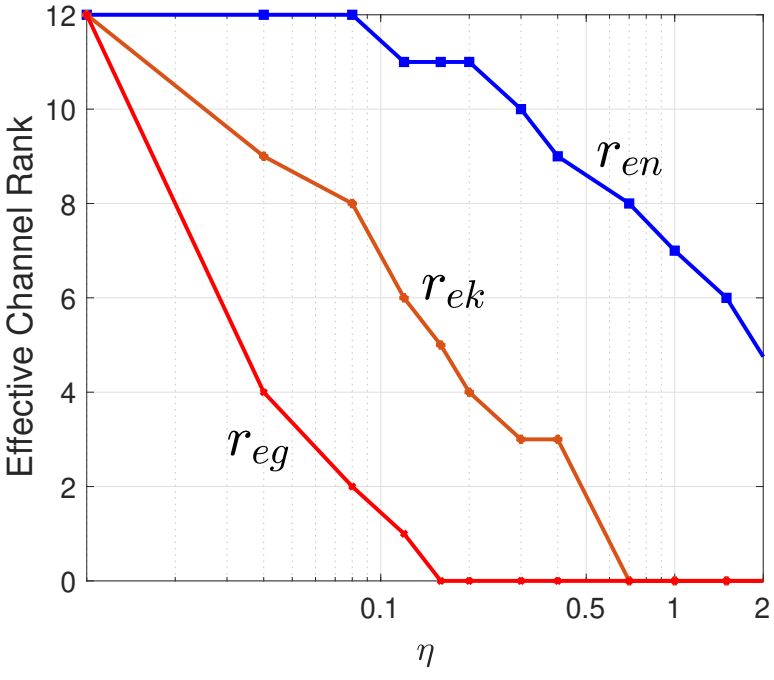
3.8(b), effective ranks have a similar trend but drop faster than those when $\rho = 0.2$, since the higher channel correlation causes interference strength more concentrated in few directions (see Fig. 3.4). A similar conclusion can be found for $\rho = 0.6$, and we omit the figure to conserve space. Clearly, the setting of rank threshold scaling factor η has different effect on different interference links in terms of effective rank determination.

Fig. 3.9 shows the total number of data streams in the network from our optimal objective (averaged over 10 random network instances similar to Fig. 3.7). As shown in this figure, for a given ρ, the total number of data streams steadily increases from 24 to 96 and then flattens out. This is because the higher the rank threshold scaling factor η, the lower the effective channel ranks on interference links in the network. As a result, fewer DoFs are needed to cancel interferences and more DoFs can be allocated for SM. When η is greater than 10, the number of data streams cannot be further increased, either there is no room to further decrease of effective ranks on interference links (all effective ranks are 0), or further decrease of effective ranks on interference links will not improve objective value, due to the bounds on effective ranks on SM links. We also observe that for the same rank threshold η, a higher number of data streams can be achieved for higher channel correlation level, due to lower effective ranks.

The above example demonstrates the impact of effective rank threshold setting on the number of data streams that can be transported in the network. However, a larger number of data streams in the network does not necessarily mean a higher throughput (in bits/s/Hz), due to un-cancelled interference (considered as noise) and channel hardening effect [71]. In the next section, we investigate the impact of effective rank threshold setting on achievable throughput in the network.



(a) $\rho = 0.2$



(b) $\rho = 0.4$

Figure 3.8: Effective ranks on interference links versus rank threshold scaling factor η.

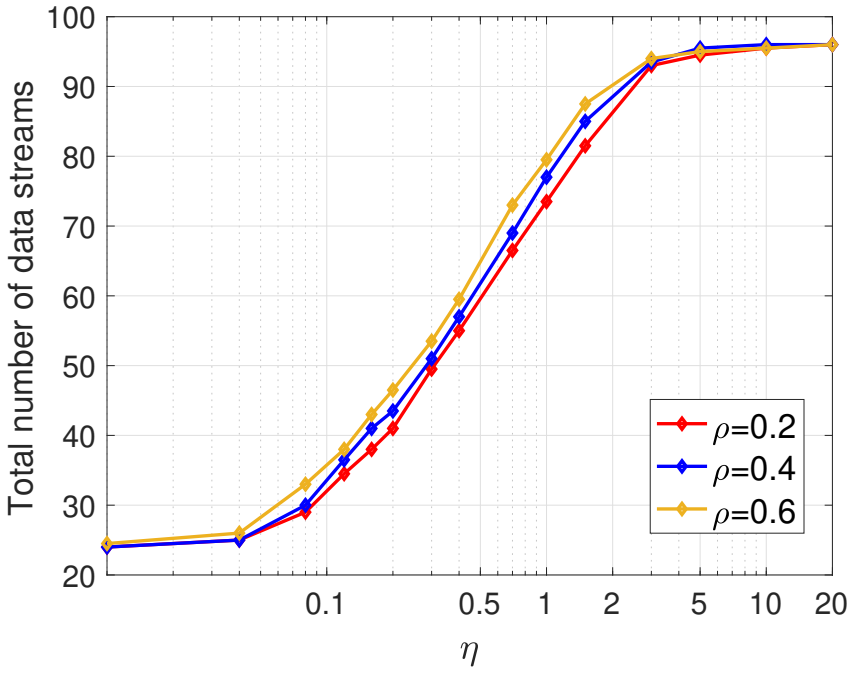


Figure 3.9: Total number of data streams in the network.

3.5 Throughput Calculation and Optimal Throughput-η trade-off

In this section, we calculate the actual throughput for a given DoF allocation for SM and IC. Then we explore the trade-off between throughput maximization and interference threshold scaling factor η.

3.5.1 Throughput Calculation

Assume a DoF allocation for SM and IC is feasible for an MU-MIMO network. Then the network throughput is the sum of the throughput achieved on each data stream under SM. So the key question is how to calculate throughput for each SM stream.

For each data stream, we can calculate its throughput by finding its SINR and then apply the Shannon capacity formula. The only subtlety here is that the SINR calculation should include all interferences that this data stream is suffering from, which includes all un-cancelled interference at PHY layer and white noise. To do this, we need to go to the PHY layer and work with the transmit and receive vectors for each data stream. Denote $\mathbf{U}_i \in \mathbb{C}^{N_i^{\mathrm{T}} \times z_{i*}}$ as the weight matrix at Tx node i with z_{i*} outgoing SM data streams and $\mathbf{V}_j \in \mathbb{C}^{N_j^{\mathrm{R}} \times z_{*j}}$ as the weight matrix at Rx node j with z_{*j} incoming SM data streams. Assume we have additive white Gaussian noise (AWGN) with zero mean and variance n_0^2. To satisfy the transmit power constraint at node i and decoding power constraint at node j, the weight matrices must satisfy

$$\mathrm{Tr}(\mathbf{U}_i\mathbf{U}_i^\dagger) = 1, \quad \mathrm{Tr}(\mathbf{V}_j\mathbf{V}_j^\dagger) = 1, \qquad (i \in \mathcal{K}^{\mathrm{T}}, j \in \mathcal{K}^{\mathrm{R}}).$$

In Section 3.6, we will show one implementation on how to derive $\mathbf{U}_i$ and $\mathbf{V}_j$ based on a DoF allocation while guaranteeing PHY layer feasibility. For now, let's assume the $\mathbf{U}_i$'s and $\mathbf{V}_j$'s are already found. Define the partition of matrix $\mathbf{U}_i$ as $[\mathbf{U}_{i,j_1} \;\; \mathbf{U}_{i,j_2} \;\; \cdots \;\; \mathbf{U}_{i,j_M}]$, where $j_1, j_2, \cdots, j_M$ are Tx node i's M recipients, i.e., $\{j_1, j_2, \cdots, j_M\} = \mathcal{K}_i^{\mathrm{R}}$, then $\mathbf{U}_{i,j_1}, \mathbf{U}_{i,j_2}, \cdots, \mathbf{U}_{i,j_M}$ are sub-weights corresponding to Rx nodes $j_1, j_2, ..., j_M$, with dimensions $N_i^{\mathrm{T}} \times z_{ij_1}, N_i^{\mathrm{T}} \times z_{ij_2}, \cdots, N_i^{\mathrm{T}} \times z_{ij_M}$ ($z_{i*} = \sum_{n=1}^{M} z_{ij_n}$), respectively.

For any $j \in \mathcal{K}_i^{\mathrm{R}}$, the signal-to-interference-plus-noise ratio (SINR) of the f-th stream on link $i \to j$ is then given by

$$\mathrm{SINR}_{ij}^f = \frac{\gamma_{ij}^f}{\mathbf{V}_j^{[*f]\dagger}\mathbf{Q}_j\mathbf{V}_j^{[*f]} - \gamma_{ij}^f}, \tag{3.14}$$

where $(\cdot)^{[*f]}$ is the f-th column of $(\cdot)$ and

$$\gamma_{ij}^{f} = P_i L_{ij} \mathbf{V}_j^{[*f]\dagger} \mathbf{H}_{ij}^{\dagger} \mathbf{U}_{i,j}^{[*f]} \mathbf{U}_{i,j}^{[*f]\dagger} \mathbf{H}_{ij} \mathbf{V}_j^{[*f]},$$
$$\mathbf{Q}_j = n_0^2 \mathbf{I}_{N_j} + \sum_{k \in \mathcal{K}^{\mathrm{T}}} P_k L_{kj} \mathbf{H}_{kj}^{\dagger} \mathbf{U}_k \mathbf{U}_k^{\dagger} \mathbf{H}_{kj}.$$

Finally, the network throughput in bits/sec/Hz is given by

$$C = \sum_{i \in \mathcal{K}^{\mathrm{T}}} \sum_{j \in \mathcal{K}_i^{\mathrm{R}}} \sum_{f=1}^{z_{ij}} \log_2 \left(1 + \mathrm{SINR}_{ij}^{f}\right). \tag{3.15}$$

3.5.2 Optimal Throughput-η Trade-off

From the network throughput expression (3.15), it is evident that there exists a trade-off between throughput and η. When η increases, more DoFs will be made available to support a larger number of SM data streams z_{ij} (as shown in Section 3.4) and we have a larger value of z_{ij} in (3.15) to increase throughput. On the other hand, higher η means more weak interferences are not cancelled and left in the network. This will decrease the SINR term in (3.15) and decrease throughput. Thus, we have a trade-off. Unfortunately, due to the non-convex nature of (3.15), a closed-form expression to explore optimal throughput-η trade-off remains unknown. In the rest of this section, we use simulation study to explore an optimal throughput-η trade-off and gain insights.

We use the same MU-MIMO network setting in Section 3.4.2. We randomly generate 10 instances and evaluate the average performance among the 10 instances. Fig. 3.10 shows network throughput vs. η under different channel correlation levels ρ. Note that $\eta = 0$ stands for traditional DoF IC which uses DoFs to cancel interference indiscriminately in all directions in the eigenspace. For $\rho = 0.2$, we can see network throughput keep increasing until threshold $\eta = 0.3$, as more data streams are supported (see Fig. 3.9) while weak (un-

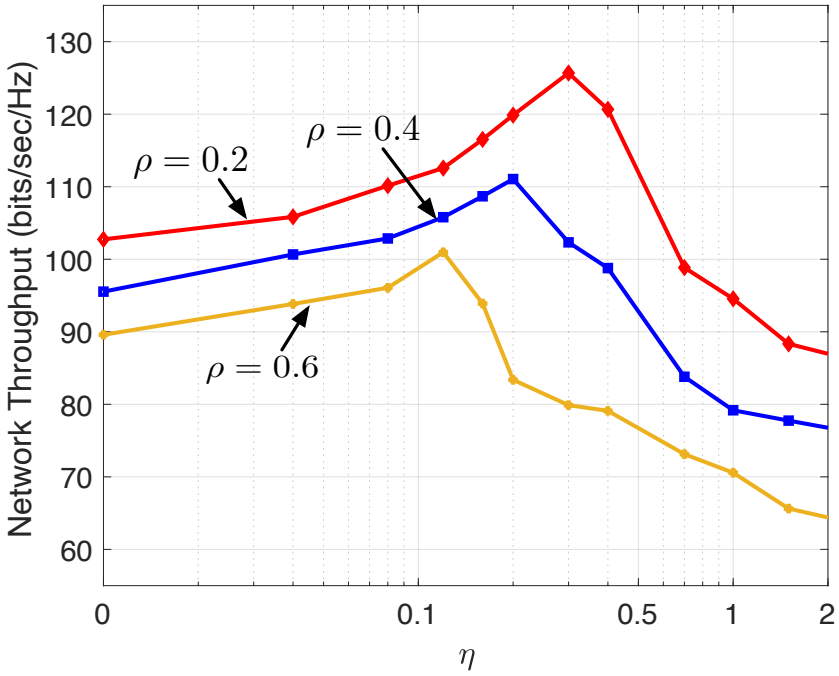


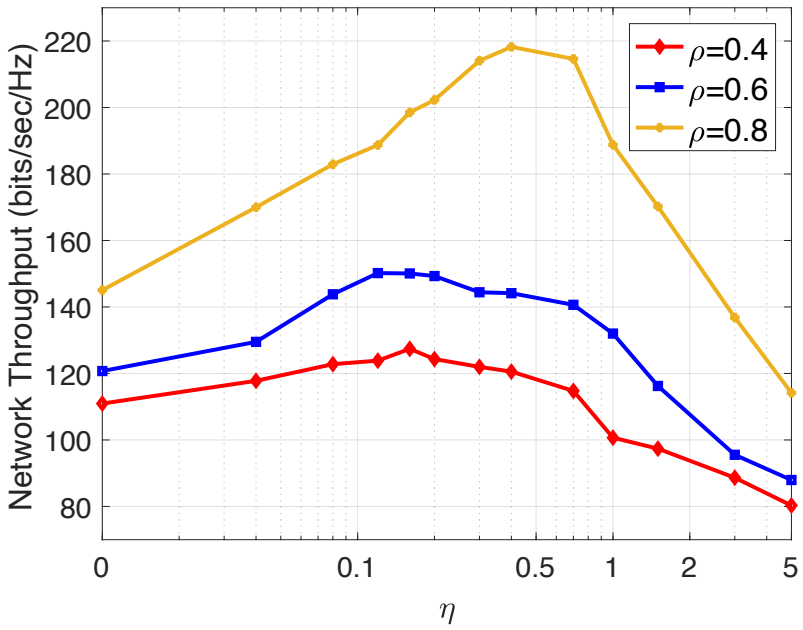
Figure 3.10: Performance of network throughput under increasing threshold η. Kronecker model for both intended and interference channels.

cancelled) interference has negligible impact (Section 3.6 will show the interference level versus η). However, as we further increase η, throughput decreases due to un-cancelled interference. Throughput under $\eta = 0.6$ can be as good as that with traditional IC (i.e., $\eta = 0$). By increasing η larger than 0.6, even though more DoFs can be made available for SM, un-cancelled interference will play a dominant role and will result in worse performance than traditional IC. For $\rho = 0.4$ and 0.6, we can see a similar trade-off. For this network setting, the optimal effective rank threshold η should be set to $\eta = 0.3, 0.2$ and 0.12 for $\rho = 0.2, 0.4$ and 0.6, respectively. The peak throughput (achieved at optimal η) is 22.3%, 16.25%, 12.71% more than that achieved at $\eta = 0$ for $\rho = 0.2, 0.4$ and 0.6, respectively. We also note that with a higher channel correlation level ρ, network throughput becomes lower. This is because high channel correlation also hinders MIMO's SM capability, which results in a lower throughput performance.

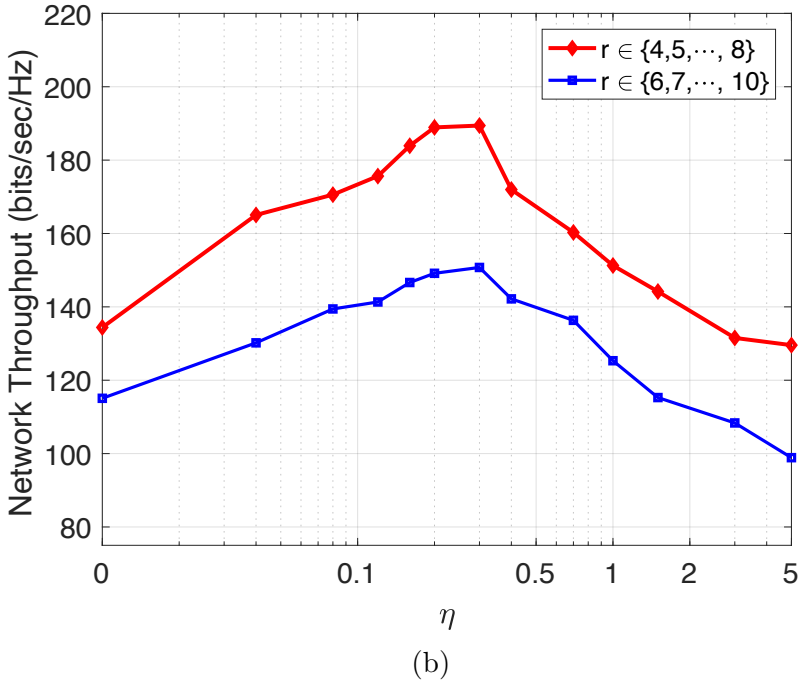
In the scenarios where intended links present low correlations while interference links present high correlations (e.g., high correlation caused by poor scattering or "key-hole" effect [30, 31]), our rank-based IC can be even more beneficial. To demonstrate this, we consider two different scenarios. First, fast fading for intended links is modeled by Rayleigh channel while fast fading for interference links is modeled by Kronecker model. Second, fast fading for intended links is modeled by Rayleigh channel while fast fading for interference links is modeled by reduced-rank model [28, 72, 73].

For Fig. 3.11(a), fast fading for intended links is modeled by Rayleigh channel, i.e., $\mathbf{H}_{ij} = \mathbf{H}_w$ $(i \in \mathcal{K}^{\mathrm{T}}, j \in \mathcal{K}_i^{\mathrm{R}})$, while fast fading for interference links is modeled by $\mathbf{H}_{ij} = \mathbf{R}_{tx}^{1/2}\mathbf{H}_w\mathbf{R}_{rx}^{1/2}$ $(i \in \mathcal{K}^{\mathrm{T}}, j \in \mathcal{K}^{\mathrm{R}}, j \notin \mathcal{K}_i^{\mathrm{R}})$, with $\rho \in \{0.4, 0.6, 0.8\}$. As shown in Fig. 3.11(a), network throughput follows a similar trend as Fig. 3.10 as we increase effective rank threshold η. However, we observe that for a higher channel correlation level ρ at interference links, we obtain a much higher throughput gain by setting optimal effective rank threshold η. Specifically, the peak throughput (achieved at optimal η) is 15.82%, 24.44%, 50.48% more than that achieved at $\eta = 0$ for $\rho = 0.4, 0.6$ and 0.8, respectively. This is because well-conditioned intended channels have the capability to achieve higher throughput when carrying more data streams, thus can fully benefit from exploiting interference signal strength in the eigenspace on correlated interference channels.

Different from Fig. 3.11(a), Fig. 3.11(b) shows the results that fast fading for intended links is still modeled by Rayleigh channel while interference links are modeled by reduced-rank channel model [72, 73]. Reduced-rank channel model generates channels by letting $\mathbf{H}_{ij} = \mathbf{AB}$, where $\mathbf{A}$ is an $N_i^{\mathrm{T}} \times r$ full-rank matrix with its entries containing zero-mean i.i.d. complex Gaussian random variables, and $\mathbf{B}$ is a $r \times N_j^{\mathrm{R}}$ full-rank rectangular unitary matrix, where $r \leq \min\{N_i^{\mathrm{T}}, N_j^{\mathrm{R}}\}$. This model can guarantee that the channel is of rank r (with probability 1). In our simulation experiment, the rank of an interference channel



(a)



(b)

Figure 3.11: Performance of network throughput under increasing threshold η. (a) Kronecker model for interference channels and Rayleigh model for intended channels. (b) Rank-reduced channel model for interference channels and Rayleigh model for intended channels.

r is randomly chosen from $\{4, 5, \cdots, 8\}$ and $\{6, 7, \cdots, 10\}$, respectively. Fig. 3.11(b) also presents the throughput-η trade-off. We observe that the highest network throughput is obtained when effective rank threshold η is equal to 0.3 in both setting, which is 40.95% and 31.00% more than that achieved at $\eta = 0$ for $r \in \{4, 5, \cdots, 8\}$ and $r \in \{6, 7, \cdots, 10\}$, respectively. The trade-off in Fig. 3.10 and 3.11 reaffirms that blind IC in all its directions is not efficient from a throughput perspective.

3.6 Physical Layer Feasibility

In Section 3.5 we assumed feasible weight matrices $\mathbf{U}_i$ and $\mathbf{V}_j$ at the PHY layer are given *a priori* corresponding to a particular DoF allocation. In this section, we show how to find such weight matrices at each node.

As expected, finding these feasible at the PHY layer for an MU-MIMO network is not trivial. First and foremost, the Tx weights and Rx weights are interdependent on each other. That is, the Tx weights for IC depend on the corresponding Rx weights, while the Rx weights for IC also on the corresponding Tx weights. There is no established guideline in the literature on how to find feasible weight matrices corresponding to a DoF allocation such that interference can be cancelled completely. Second, since we are exploring effective channel ranks in this chapter and some weak interferences are not cancelled by DoFs, one cannot guarantee the existence of feasible $\mathbf{U}_i$ and $\mathbf{V}_j$ to achieve perfect (100%) interference-free transmission. This makes finding feasible weight matrices even more challenging.

In the rest of this section, we propose an iterative algorithm that is able to implement the DoF allocation (based on the DoF solution for a specific objective as shown in Section 3.4), where the strong interferences in best rank-r channels are "almost" cancelled. By "almost", we mean the remaining signal strength in the directions of strong interferences is close to

zero.

3.6.1 Basic Idea

The main idea of our algorithm is as follows. For a given DoF allocation, we have the data stream allocation (i.e. z_{ij}) on each SM link in the network, which we can use to determine the dimension for each $\mathbf{U}_i$ and $\mathbf{V}_j$. Then, under the original channel matrix $\mathbf{H}_{ij}$, to cancel all the inter-stream and inter-node interference, we must have

$$\mathbf{U}_i^{\dagger}\left[\mathbf{H}_{ij_1}\mathbf{V}_{j_1} \quad \mathbf{H}_{ij_2}\mathbf{V}_{j_2}\cdots\right] = \Lambda_{z_{i*}}, i \in \mathcal{K}^{\mathrm{T}}, j_1, j_2 ... \in \mathcal{K}_i^{\mathrm{R}}, \tag{3.16}$$

$$\mathbf{U}_i^{\dagger}\mathbf{H}_{ij}\mathbf{V}_j = \mathbf{0}, \quad i \in \mathcal{K}^{\mathrm{T}}, j \in \mathcal{K}^{\mathrm{R}}, j \notin \mathcal{K}_i^{\mathrm{R}}, \tag{3.17}$$

where $\Lambda_{z_{i*}}$ is a $z_{i*} \times z_{i*}$ diagonal matrix with z_{i*} non-zero diagonal elements.

Although (3.16) can always be satisfied for all SM links by standard ZF design, (3.17), however, cannot be satisfied for all $i \in \mathcal{K}^{\mathrm{T}}, j \in \mathcal{K}^{\mathrm{R}}, j \notin \mathcal{K}_i^{\mathrm{R}}$ if there are not enough remaining DoFs to cancel those weak interference on some links. Recognizing that not all interference can be perfectly cancelled, we focus our goal on cancelling all the strong interference, which is based on the best rank$-r$ approximate channel $\tilde{\mathbf{H}}_{ij} = \sum_{l=1}^{r_{ij}} \sigma_l \boldsymbol{u}_l \boldsymbol{v}_l^{\dagger}$ via SVD of $\mathbf{H}_{ij}$. That is, we want to have

$$\mathbf{U}_i^{\dagger}\tilde{\mathbf{H}}_{ij}\mathbf{V}_j = \mathbf{0}, \quad \text{for } i \in \mathcal{K}^{\mathrm{T}}, j \in \mathcal{K}^{\mathrm{R}}, j \notin \mathcal{K}_i^{\mathrm{R}}. \tag{3.18}$$

The weak (un-cancelled) interference will reduce network throughput and will be taken into account in throughput calculation (as we did in Section 3.5).

Equations (3.16) and (3.18) constitute a system of bilinear equations and a general solution to bilinear equations remains unknown [74]. Instead of finding a feasible solution to

(3.16) and (3.18), we propose to minimize the LHS of (3.18) for all $i \in \mathcal{K}^{\mathrm{T}}, j \in \mathcal{K}^{\mathrm{R}}, j \notin \mathcal{K}_i^{\mathrm{R}}$, subject to (3.16). Denote Δ_{LI} as the leakage interference in the network,[3] which is defined as

$$\Delta_{\mathrm{LI}} = \sum_{i \in \mathcal{K}^{\mathrm{T}}} \sum_{j \in \mathcal{K}^{\mathrm{R}}}^{j \notin \mathcal{K}_i^{\mathrm{R}}} P_i L_{ij} \left|\left| \mathbf{U}_i^{\dagger} \tilde{\mathbf{H}}_{ij} \mathbf{V}_j \right|\right|_F^2. \tag{3.19}$$

The problem to solve is to minimize Δ_{LI} subject to (3.16).

To do this, we propose a simple yet effective approach to address the dependency between Tx weight matrices $\mathbf{U}_i$ and Rx weight matrices $\mathbf{V}_j$ by updating each in an alternating fashion (i.e., fixing $\mathbf{U}_i$ and update $\mathbf{V}_j$ and vice versa). Specifically, in each iteration, Tx weight matrices $\mathbf{U}_i$ are optimized first with given Rx weight matrices $\mathbf{V}_j$ and channel information. Then we optimize Rx weight matrices $\mathbf{V}_j$ with given Tx weight matrices $\mathbf{U}_i$ and channel information. For each weight matrix (either at Tx or Rx node) optimization, the weight matrix is updated by solving a minimization problem with the objective Δ_{LI} and the updated set of constraints. The iteration terminates if we find no improvement after a number of consecutive iterations.

3.6.2 Algorithm Details

Now we describe in detail on how to find weight matrices.

Step 1: Initialization. Initially all the Tx and Rx weight matrices can be set arbitrarily but have to be full rank matrices with dimension $N_i^{\mathrm{T}} \times z_{i*}$ and $N_j^{\mathrm{R}} \times z_{*j}$, respectively.

Step 2: Optimizing Tx Weights. In this step, channel information $\tilde{\mathbf{H}}_{ij}$ and Rx weight matrices $\mathbf{V}_j$ are given. We optimize Tx weight matrices $\mathbf{U}_i$ so as to minimize leakage

[3] Incidentally, a similar definition of leakage interference involving only channel matrix $\mathbf{H}_{ij}$ is given in [75, 76].

interference. Denote $\Delta_{\mathrm{LI},i}^{\mathrm{T}} = \sum_{j\in\mathcal{K}^{\mathrm{R}}}^{j\notin\mathcal{K}_i^{\mathrm{R}}} P_i L_{ij} \left|\left|\mathbf{U}_i^{\dagger}\tilde{\mathbf{H}}_{ij}\mathbf{V}_j\right|\right|_F^2$ as the leakage interference at Tx node i, then

$$\min_{\mathbf{U}_1,\mathbf{U}_2,\ldots,\mathbf{U}_{|\mathcal{K}^{\mathrm{T}}|}} \Delta_{\mathrm{LI}} = \min_{\mathbf{U}_1,\mathbf{U}_2,\ldots,\mathbf{U}_{|\mathcal{K}^{\mathrm{T}}|}} \sum_{i\in\mathcal{K}^{\mathrm{T}}} \Delta_{\mathrm{LI},i}^{\mathrm{T}} = \sum_{i\in\mathcal{K}^{\mathrm{T}}} \min_{\mathbf{U}_i} \Delta_{\mathrm{LI},i}^{\mathrm{T}}. \tag{3.20}$$

It follows that $\min \Delta_{\mathrm{LI}}$ can be solved separately by solving $|\mathcal{K}^{\mathrm{T}}|$ independent sub-problems $\min_{\mathbf{U}_i} \Delta_{\mathrm{LI},i}^{\mathrm{T}}$, i.e., one sub-problem for each Tx node. (Note that $\Delta_{\mathrm{LI}} = \sum_{i\in\mathcal{K}^{\mathrm{T}}} \Delta_{\mathrm{LI},i}^{\mathrm{T}}$ and $\Delta_{\mathrm{LI},i}^{\mathrm{T}}$'s are independent among each other). The constraints of sub-problem i are based on Tx node i's IC responsibilities (i.e., the number of DoFs needed to cancel interference from i to j at Tx node i (d_{ij}^{T}) per our discussion in Section 3.4). For Tx node i (sub-problem i), we have the following three cases to determine the sets of constraints to optimize $\mathbf{U}_i$:

- $d_{ij}^{\mathrm{T}} = 0, j \in \mathcal{K}^{\mathrm{R}}$. In this case, Tx node i is not responsible for cancelling interference from Tx node i to Rx node j. Thus no constraint is needed in this case.

- $d_{ij}^{\mathrm{T}} = z_{*j}$ and $d_{ij}^{\mathrm{T}} < r_{ij}, j \in \mathcal{K}^{\mathrm{R}}$. In this case, Tx node i is responsible for cancelling all the interference from Tx node i to Rx node j. Denote $\mathcal{D}_i^{\mathrm{T}}$ as the set of Rx nodes that Tx node i is responsible for cancelling all its interference, i.e., $\mathcal{D}_i^{\mathrm{T}} = \{j : d_{ij}^{\mathrm{T}} = z_{*j},\ d_{ij}^{\mathrm{T}} < r_{ij}, j \in \mathcal{K}^{\mathrm{R}}, j \notin \mathcal{K}_i^{\mathrm{R}}\}$. Then the following set of constraints is needed for optimizing $\mathbf{U}_i$:

$$\mathbf{U}_i^{\dagger}\tilde{\mathbf{H}}_{ij}\mathbf{V}_j = \mathbf{0}, \quad j \in \mathcal{D}_i^{\mathrm{T}}, i \in \mathcal{K}^{\mathrm{T}}.$$

- $d_{ij}^{\mathrm{T}} < z_{*j}$ or $d_{ij}^{\mathrm{T}} = r_{ij}, j \in \mathcal{K}^{\mathrm{R}}$. In this case, the number of DoFs consumed to cancel interference from Tx node i to Rx node j is shared between nodes i and j. That is, Tx node i uses d_{ij}^{T} DoFs to cancel interferences from d_{ij}^{T} directions in the eigenspace, and the remaining interferences (from $d_{ij}^{\mathrm{R}} = r_{ij} - d_{ij}^{\mathrm{T}}$ directions in the eigenspace) will be cancelled by Rx node j (when optimizing Rx weights later in Step 3), which guarantees

the interference channel (based on best rank-r approximation) is cleared for data transmission. Note that in this case Rx weight matrix $\mathbf{V}_j$ is not needed in the constraints to update $\mathbf{U}_i$; only the channel matrix $\tilde{\mathbf{H}}_{ij}$ is needed. Let $\tilde{\mathbf{H}}_{ij}^{[m,n]} = \sum_{l=m}^{n} \sigma_l \boldsymbol{u}_l \boldsymbol{v}_l^\dagger$ which represents the channel information at directions corresponding to the m-th to the n-th largest eigenvalues (recall that the SVD of $\mathbf{H}_{ij}$ is $\mathbf{H}_{ij} = \sum_{l=1}^{N_j^{\mathrm{R}}} \sigma_l \boldsymbol{u}_l \boldsymbol{v}_l^\dagger$). Denote $\widetilde{\mathcal{D}}_i^{\mathrm{T}} = \{j : 0 < d_{ij}^{\mathrm{T}} < z_{*j} \text{ or } d_{ij}^{\mathrm{T}} = r_{ij}, j \in \mathcal{K}^{\mathrm{R}}, j \notin \mathcal{K}_i^{\mathrm{R}}\}$ as the set of Rx nodes that Tx node i is partially responsible for cancelling its interference. Then we have the following set of constraints:

$$\tilde{\mathbf{H}}_{ij}^{[1,d_{ij}^{\mathrm{T}}]\dagger} \mathbf{U}_i = \mathbf{0}, \quad j \in \widetilde{\mathcal{D}}_i^{\mathrm{T}}, i \in \mathcal{K}^{\mathrm{T}}.$$

In addition, as a necessary condition to distinguish different data streams, Tx weight matrix $\mathbf{U}_i$ must have linearly independent columns. We consider the following constraint to guarantee the independency among the columns of $\mathbf{U}_i$:

$$\mathbf{U}_i^\dagger \mathbf{U}_i = \mathbf{I}.$$

Putting together the objective function and all the constraints above, for each Tx node $i \in \mathcal{K}^{\mathrm{T}}$, we have the following optimization problem for Tx weight matrix $\mathbf{U}_i$:

$$\begin{aligned}
\textbf{OPT-Tx-}i \qquad \min_{\mathbf{U}_i \in \mathbb{C}^{N_i^{\mathrm{T}} \times z_{i*}}} \quad & \Delta_{\mathrm{LI},i}^{\mathrm{T}} = \sum_{j \in \mathcal{K}^{\mathrm{R}}}^{j \notin \mathcal{K}_i^{\mathrm{R}}} P_i L_{ij} \left|\left| \mathbf{U}_i^\dagger \tilde{\mathbf{H}}_{ij} \mathbf{V}_j \right|\right|_F^2, \\
\text{s.t.} \quad & \mathbf{U}_i^\dagger \mathbf{U}_i = \mathbf{I}, \\
& \mathbf{V}_j^\dagger \tilde{\mathbf{H}}_{ij}^\dagger \mathbf{U}_i = \mathbf{0}, \quad j \in \mathcal{D}_i^{\mathrm{T}}, \\
& \tilde{\mathbf{H}}_{ij}^{[1,d_{ij}^{\mathrm{T}}]\dagger} \mathbf{U}_i = \mathbf{0}, \quad j \in \widetilde{\mathcal{D}}_i^{\mathrm{T}}.
\end{aligned}$$

where $\mathcal{D}_i^{\mathrm{T}} = \{j : d_{ij}^{\mathrm{T}} = z_{i*},\, d_{ij}^{\mathrm{T}} < r_{ij}, j \in \mathcal{K}^{\mathrm{R}}\}$, $\widetilde{\mathcal{D}}_i^{\mathrm{T}} = \{j : d_{ij}^{\mathrm{T}} < z_{i*} \text{ or } d_{ij}^{\mathrm{T}} = r_{ij}, j \in \mathcal{K}^{\mathrm{R}}\}$ and $\tilde{\mathbf{H}}_{ij}^{[m,n]} = \sum_{l=m}^{n} \sigma_l \boldsymbol{u}_l \boldsymbol{v}_l^{\dagger}$.

The optimal solution to problem OPT-Tx-i is given by the following lemma.

Lemma 3.1. *The optimal solution to problem* **OPT-Tx-***i* *is*

$$\mathbf{U}_i = \begin{cases} \text{nullspace}^{[1,z_{i*}]}\left(\begin{bmatrix}\mathbf{BAB}\\ \mathbf{C}\end{bmatrix}\right), & \text{if } z_{i*} \le N_i^{\mathrm{T}} - c \\ \begin{bmatrix}\text{nullspace}\left(\begin{bmatrix}\mathbf{BAB}\\ \mathbf{C}\end{bmatrix}\right) & \text{eig}^{[N_i^{\mathrm{T}}-p+1,z_{i*}+c-p]}(\mathbf{BAB})\end{bmatrix}, & \\ & \text{if } z_{i*} > N_i^{\mathrm{T}} - c \end{cases}$$

where $p = \text{rank}(\mathbf{BAB})$, $c = \text{rank}\left(\left[\begin{smallmatrix}\mathbf{BAB}\\ \mathbf{C}\end{smallmatrix}\right]\right)$, $\mathbf{A} = \sum_{j\in\mathcal{K}^R}^{j\notin\mathcal{K}_i^R} P_i L_{ij} \tilde{\mathbf{H}}_{ij} \mathbf{V}_j \mathbf{V}_j^{\dagger} \tilde{\mathbf{H}}_{ij}^{\dagger}$, $\mathbf{B}$ *is a projection matrix given by* $\mathbf{B} = \mathbf{I}_{z_{i*}} - \mathbf{C}^{\dagger}(\mathbf{C}\mathbf{C}^{\dagger})^{-1}\mathbf{C}$, *and* $\mathbf{C}$ *is given by*

$$\mathbf{C} = \begin{bmatrix} \mathbf{V}_{\bar{j}_1}^{\dagger} \tilde{\mathbf{H}}_{i\bar{j}_1}^{\dagger} \\ \mathbf{V}_{\bar{j}_2}^{\dagger} \tilde{\mathbf{H}}_{i\bar{j}_2}^{\dagger} \\ \vdots \\ \tilde{\mathbf{H}}_{i\hat{j}_1}^{[1,d_{i\bar{j}_1}^{\mathrm{T}}]\dagger} \\ \tilde{\mathbf{H}}_{i\hat{j}_2}^{[1,d_{i\bar{j}_2}^{\mathrm{T}}]\dagger} \\ \vdots \end{bmatrix} \text{ with } \begin{matrix} \{\bar{j}_1, \bar{j}_2, \cdots\} = \mathcal{D}_i^{\mathrm{T}} \\ \{\hat{j}_1, \hat{j}_2, \cdots\} = \widetilde{\mathcal{D}}_i^{\mathrm{T}} \end{matrix}.$$

$\text{nullspace}^{[1,z_{i*}]}(\mathbf{X})$ *denotes* z_{i*} *orthonormal vectors in the nullspace of* $\mathbf{X}$, *and* $\text{eig}^{[a,b]}(\mathbf{X})$ *is the eigenvectors of* $\mathbf{X}$ *corresponding to the* a*-th smallest to the* b*-th smallest eigenvalues. Further, the optimal objective value is given by*

$$\sum_{l=c-p+1}^{z_{i*}+c-p} \lambda_l(\mathbf{BAB}).$$

where $\lambda_l(\mathbf{X})$ is the l-th smallest eigenvalue of matrix $\mathbf{X}$.

Proof. The objective function of OPT-Tx-i can be rewritten as

$$\begin{aligned}
&\sum_{j\in\mathcal{K}^{\mathrm{R}}}^{j\notin\mathcal{K}_i^{\mathrm{R}}} P_i L_{ij} \left|\left|\mathbf{U}_i^{\dagger}\mathbf{H}_{ij}\mathbf{V}_j\right|\right|_F^2 \\
=&\mathrm{Tr}\left(\mathbf{U}_i^{\dagger}\left(\sum_{j\in\mathcal{K}^{\mathrm{R}}}^{j\notin\mathcal{K}_i^{\mathrm{R}}} P_i L_{ij}\mathbf{H}_{ij}\mathbf{V}_j\mathbf{V}_j^{\dagger}\mathbf{H}_{ij}^{\dagger}\right)\mathbf{U}_i\right) \\
=&\sum_{l=1}^{z_{i*}} \boldsymbol{w}_l^{\dagger}\left(\sum_{j\in\mathcal{K}^{\mathrm{R}}}^{j\notin\mathcal{K}_i^{\mathrm{R}}} P_i L_{ij}\mathbf{H}_{ij}\mathbf{V}_j\mathbf{V}_j^{\dagger}\mathbf{H}_{ij}^{\dagger}\right)\boldsymbol{w}_l,
\end{aligned} \tag{3.21}$$

where $\boldsymbol{w}_l$ is the l-th column of matrix $\mathbf{U}_i$. For ease of exposition, let matrix $\mathbf{A}$ be $\mathbf{A} = \sum_{j\in\mathcal{K}^{\mathrm{R}}}^{j\notin\mathcal{K}_i^{\mathrm{R}}} P_i L_{ij}\mathbf{H}_{ij}\mathbf{V}_j\mathbf{V}_j^{\dagger}\mathbf{H}_{ij}^{\dagger}$, and matrix $\mathbf{C}$ be

$$\mathbf{C} = \begin{bmatrix} \mathbf{V}_{\bar{j}_1}^{\dagger}\tilde{\mathbf{H}}_{i\bar{j}_1}^{\dagger} \\ \mathbf{V}_{\bar{j}_2}^{\dagger}\tilde{\mathbf{H}}_{i\bar{j}_2}^{\dagger} \\ \vdots \\ \tilde{\mathbf{H}}_{i\hat{j}_1}^{[1,d_{i\hat{j}_1}^T]\dagger} \\ \tilde{\mathbf{H}}_{i\hat{j}_2}^{[1,d_{i\hat{j}_2}^T]\dagger} \\ \vdots \end{bmatrix} \; \textit{with} \; \begin{matrix} \{\bar{j}_1,\bar{j}_2,\cdots\} = \mathcal{D}_i^T \\ \{\hat{j}_1,\hat{j}_2,\cdots\} = \widetilde{\mathcal{D}}_i^T \end{matrix}.$$

Then for each term in the summation of objective function (3.21) (i.e., $\boldsymbol{w}_l^{\dagger}\mathbf{A}\boldsymbol{w}_l$), along with the constraints of OPT-Tx-i, we have the following Lagrangian function

$$\phi(\boldsymbol{w}_l,\lambda_l,\boldsymbol{\beta}) = \boldsymbol{w}_l^{\dagger}\mathbf{A}\boldsymbol{w}_l - \lambda_l(\boldsymbol{w}_l^{\dagger}\boldsymbol{w}_l - 1) + 2\boldsymbol{\beta}^T\mathbf{C}\boldsymbol{w}_l\ , \tag{3.22}$$

where $\lambda_l, \boldsymbol{\beta}$ are Lagrangian multipliers. To find the KKT points of (3.22), we differentiate (3.22) with respect to $\boldsymbol{w}_l$ and let $\frac{\partial\phi}{\partial\boldsymbol{w}_l} = 0$. We have

$$\mathbf{A}\boldsymbol{w}_l - \lambda_l \boldsymbol{w}_l + \mathbf{C}^\dagger \boldsymbol{\beta} = \mathbf{0}. \tag{3.23}$$

By multiplying $(\mathbf{C}\mathbf{C}^\dagger)^{-1}\mathbf{C}$ on both sides of (3.23) and using the constraint of $\mathbf{C}\boldsymbol{w}_l = \mathbf{0}$, we have

$$\boldsymbol{\beta} = -(\mathbf{C}\mathbf{C}^\dagger)^{-1}\mathbf{C}\mathbf{A}\boldsymbol{w}_l. \tag{3.24}$$

By substituting (3.24) into (3.23), we obtain

$$\mathbf{B}\mathbf{A}\boldsymbol{w}_l = \lambda_l \boldsymbol{w}_l, \tag{3.25}$$

where $\mathbf{B} = \mathbf{I}_{N_i^{\mathrm{T}}} - \mathbf{C}^\dagger(\mathbf{C}\mathbf{C}^\dagger)^{-1}\mathbf{C}$. Note that $\mathbf{B}\boldsymbol{w}_l = \left(\mathbf{I}_{N_i^{\mathrm{T}}} - \mathbf{C}^\dagger(\mathbf{C}\mathbf{C}^\dagger)^{-1}\mathbf{C}\right)\boldsymbol{w}_l = \boldsymbol{w}_l$. We have

$$\mathbf{B}\mathbf{A}\mathbf{B}\boldsymbol{w}_l = \mathbf{B}\mathbf{A}\boldsymbol{w}_l = \lambda_l \boldsymbol{w}_l.$$

This suggests that at the KKT points, λ_l is the eigenvalue of $\mathbf{BAB}$ and $\boldsymbol{w}_l$ is the eigenvector of $\mathbf{BAB}$. Further, noting that $\mathbf{B}\boldsymbol{w}_l = \boldsymbol{w}_l$, we have

$$\boldsymbol{w}_l^\dagger \mathbf{A}\boldsymbol{w}_l = \boldsymbol{w}_l^\dagger \mathbf{B}\mathbf{A}\mathbf{B}\boldsymbol{w}_l = \lambda_l \boldsymbol{w}_l^\dagger \boldsymbol{w}_l = \lambda_l \ . \tag{3.26}$$

Eq. (3.26) suggests that at the KKT points, the objective value is $\sum_{l=1}^{z_{i*}} \boldsymbol{w}_l^\dagger \mathbf{A}\boldsymbol{w}_l = \sum_{l=1}^{z_{i*}} \lambda_l$. Therefore, to minimize $\sum_{l=1}^{z_{i*}} \boldsymbol{w}_l^\dagger \mathbf{A}\boldsymbol{w}_l$, it is equivalent to find the eigenvectors of $\mathbf{BAB}$ corresponding to z_{i*} smallest eigenvalues, while satisfying constraint $\mathbf{C}\mathbf{U}_i = \mathbf{0}$.

Denote $p = \mathrm{rank}(\mathbf{BAB})$ and $c = \mathrm{rank}\left(\left[\begin{smallmatrix}\mathbf{BAB}\\ \mathbf{C}\end{smallmatrix}\right]\right) \geq p$. We have the following two cases:

i) $z_{i*} \leq N_i^{\mathrm{T}} - c$. In this case, the dimension of the nullspace of $\mathbf{BAB}$ is $N_i^{\mathrm{T}} - p \geq N_i^{\mathrm{T}} - c \geq z_{i*}$. Thus, the z_{i*} smallest eigenvalues are zeros and the corresponding eigenvectors can be found in the nullspace of $\mathbf{BAB}$. That is, the optimal solution $\mathbf{U}_i$ satisfies $\mathbf{BABU}_i = \mathbf{0}$.

In addition, we must satisfy constraint $\mathbf{C}\mathbf{U}_i = \mathbf{0}$. Therefore, the optimal solution $\mathbf{U}_i$ can be given by the first z_{i*} columns of nullspace of $\left[\begin{smallmatrix}\mathbf{BAB}\\ \mathbf{C}\end{smallmatrix}\right]$, and the optimal objective value of OPT-Tx-i is 0.

ii) $z_{i*} > N_i^{\mathrm{T}} - c$. In this case, the first $(N_i^{\mathrm{T}} - c)$ columns of $\mathbf{U}_i$ can be derived in the nullspace of $\left[\begin{smallmatrix}\mathbf{BAB}\\ \mathbf{C}\end{smallmatrix}\right]$ (corresponding to zero eigenvalues). Then the remaining $(z_{i*} - N_i^{\mathrm{T}} + c)$ columns of $\mathbf{U}_i$ are given by the eigenvectors of $\mathbf{BAB}$ corresponding to the $(z_{i*} - N_i^{\mathrm{T}} + c)$ smallest positive eigenvalues. Note that the constraints $\mathbf{C}\mathbf{U}_i = \mathbf{0}$ are already satisfied for these eigenvectors correponding to positive eigenvalues (multiplying $\mathbf{C}_i$ on both sides of (3.25)). Further, without loss of generality, we let $\lambda_1 \leq \lambda_2 \leq \cdots \lambda_{N_i^{\mathrm{T}}}$. Then the optimal objective value is given by $\sum_{l=N_i^{\mathrm{T}}-p+1}^{N_i^{\mathrm{T}}-p+1+(z_{i*}-N_i^{\mathrm{T}}+c)-1} \lambda_l = \sum_{l=N_i^{\mathrm{T}}-p+1}^{z_{i*}+c-p} \lambda_l = \sum_{l=c-p+1}^{z_{i*}+c-p} \lambda_l$.

In summary, the optimal solution to Problem OPT-Tx-i is given by

$$\mathbf{U}_i = \begin{cases} \text{nullspace}^{[1,z_{i*}]}\left(\begin{bmatrix}\mathbf{BAB}\\ \mathbf{C}\end{bmatrix}\right), & \text{if } z_{i*} \leq N_i^{\mathrm{T}} - c \\ \left[\text{nullspace}\left(\begin{bmatrix}\mathbf{BAB}\\ \mathbf{C}\end{bmatrix}\right) \quad \text{eig}^{[N_i^{\mathrm{T}}-p+1,z_{i*}+c-p]}(\mathbf{BAB})\right], & \\ & \text{if } z_{i*} > N_i^{\mathrm{T}} - c \end{cases}$$

with the optimal objective value

$$\sum_{l=c-p+1}^{z_{i*}+c-p} \lambda_l(\mathbf{BAB}).$$

□

To ensure our algorithm to converge, we let $\mathbf{U}_i$ be updated only when the current optimal

objective value is smaller than that in the last iteration, i.e., $\Delta^{\mathrm{T}}_{\mathrm{LI},i}(t) < \Delta^{\mathrm{T}}_{\mathrm{LI},i}(t-1)$. Otherwise $\mathbf{U}_i$ remains unchanged as in the last iteration until updates in future iterations.

Step 3: Optimizing Rx Weights. Similar to optimizing Tx weight matrices, we have $|\mathcal{K}^{\mathrm{R}}|$ independent sub-problems for $|\mathcal{K}^{\mathrm{R}}|$ Rx nodes, and each has three sets of constraints to optimize Rx weight matrix $\mathbf{V}_j$. Deriving these constraints is similar to Step 2. Then the optimization problem for Rx weight matrix $\mathbf{V}_j$ is:

$$\begin{aligned}
\textbf{OPT-Rx-}j \quad \min_{\mathbf{V}_j \in \mathbb{C}^{N^{\mathrm{R}}_j \times z_{*j}}} \Delta^{\mathrm{R}}_{\mathrm{LI},j} = & \sum_{i \in \mathcal{K}^{\mathrm{T}}}^{i \neq s(j)} P_i L_{ij} \left|\left| \mathbf{U}_i^{\dagger} \tilde{\mathbf{H}}_{ij} \mathbf{V}_j \right|\right|_F^2, \\
\text{s.t.} \quad & \mathbf{V}_j^{\dagger} \mathbf{V}_j = \mathbf{I}, \\
& \mathbf{U}_i^{\dagger} \tilde{\mathbf{H}}_{ij} \mathbf{V}_j = \mathbf{0}, \quad i \in \mathcal{D}^{\mathrm{R}}_j, \\
& \tilde{\mathbf{H}}_{ij}^{[d^{\mathrm{T}}_{ij}+1, r_{ij}]} \mathbf{V}_j = \mathbf{0}, \quad i \in \widetilde{\mathcal{D}}^{\mathrm{R}}_j,
\end{aligned}$$

where $\mathcal{D}^{\mathrm{R}}_j = \{i : d^{\mathrm{R}}_{ij} = z_{i*},\ d^{\mathrm{R}}_{ij} < r_{ij}, i \in \mathcal{K}^{\mathrm{T}}, j \notin \mathcal{K}^{\mathrm{R}}_i\}$ and $\widetilde{\mathcal{D}}^{\mathrm{R}}_j = \{i : 0 < d^{\mathrm{R}}_{ij} < z_{i*}$ or $d^{\mathrm{R}}_{ij} = r_{ij}, i \in \mathcal{K}^{\mathrm{T}}, j \notin \mathcal{K}^{\mathrm{R}}_i\}$.

Solving problem OPT-Rx-j is similar to that for problem OPT-Tx-i and we omit the details to conserve space. To guarantee convergence, $\mathbf{V}_j$ is updated only when current optimal objective value $\Delta^{\mathrm{R}}_{\mathrm{LI},j}(t)$ is smaller than $\Delta^{\mathrm{R}}_{\mathrm{LI},j}(t-1)$ of last iteration. Otherwise $\mathbf{V}_j$ remains the same until updates in future iterations.

Step 2 and Step 3 are iteratively performed until there is no improvement for W consecutive iterations, i.e., $\Delta_{\mathrm{LI}}(t-w-1) - \Delta_{\mathrm{LI}}(t-w) < \epsilon, w = 0, 1, ..., W-1$ is met for a given convergence threshold ϵ.

Step 4: Cancelling Intra-node Interference. Within an intended link, there may exist multiple data streams and they would also interfere with each other. In this step, we cancel such intra-node interference to decode the desired data streams. This can be done by

performing a linear transformation of Tx weight $\mathbf{U}_i$ by multiplying a matrix $\mathbf{F}_i$. Such a linear transformation can decode different intra-node data streams while not affecting inter-node IC.

To show how such a linear transformation works, let's denote

$$\mathbf{\Gamma}_i = [\mathbf{H}_{ij_1}\mathbf{V}_{j_1} \quad \mathbf{H}_{ij_2}\mathbf{V}_{j_2} \quad \cdots], \; j_1, j_2, ... \in \mathcal{K}_i^{\mathrm{R}}, i \in \mathcal{K}^{\mathrm{T}}.$$

Then we define $\mathbf{F}_i$ as

$$\mathbf{F}_i = (\mathbf{U}_i^{\dagger}\mathbf{\Gamma}_i)^{-1}, \quad i \in \mathcal{K}^{\mathrm{T}}.$$

To perform a linear transformation on Tx weight matrix $\mathbf{U}_i$, we multiply it by matrix $\mathbf{F}_i$. We have:

$$\mathbf{U}_i \leftarrow \mathbf{U}_i\mathbf{F}_i^{\dagger}, \quad i \in \mathcal{K}^{\mathrm{T}}. \tag{3.27}$$

It is easy to verify that after such a transformation, we have $\mathbf{U}_i^{\dagger}\mathbf{\Gamma}_i = \mathbf{I}_{z_{i*}}$.

Step 5: Power Allocation. We apply equal power allocation for each data stream, subject to the total power constraints $\mathrm{Tr}(\mathbf{U}_i\mathbf{U}_i^{\dagger}) = 1, \mathrm{Tr}(\mathbf{V}_j\mathbf{V}_j^{\dagger}) = 1$. We have

$$\begin{aligned}
\mathbf{U}_i^{[*f]} &\leftarrow \frac{1}{\sqrt{z_{i*}}} \frac{\mathbf{U}_i^{[*f]}}{\left|\left|\mathbf{U}_i^{[*f]}\right|\right|}, \forall i \in \mathcal{K}^{\mathrm{T}}, f = 1, 2, ..., z_{i*}, \\
\mathbf{V}_j^{[*f]} &\leftarrow \frac{1}{\sqrt{z_{*j}}} \frac{\mathbf{V}_j^{[*f]}}{\left|\left|\mathbf{V}_j^{[*f]}\right|\right|}, \forall j \in \mathcal{K}^{\mathrm{R}}, f = 1, 2, ..., z_{*j}.
\end{aligned} \tag{3.28}$$

A pseudocode of our proposed algorithm to compute Tx and Rx weights is given in Algorithm 3.1. A proof of the algorithm's convergence is given as following.

Proof. The objective function associated with Tx node i at iteration t in Step 2 is

$$\Delta_{\mathrm{LI},i}^{\mathrm{T}}(t) = \sum_{j \in \mathcal{K}^{\mathrm{R}}}^{j \notin \mathcal{K}_i^{\mathrm{R}}} \left|\left| \mathbf{U}_i^{\dagger} \mathbf{H}_{ij} \mathbf{V}_j \right|\right|_F^2 .$$

Thus the total leakage interference is given by the sum of $\Delta_{\mathrm{LI},i}^{\mathrm{T}}$ over all Tx nodes, i.e.,

$$\Delta_{\mathrm{LI}}(t) = \sum_{i \in \mathcal{K}^{\mathrm{T}}} \sum_{j \in \mathcal{K}^{\mathrm{R}}}^{j \notin \mathcal{K}_i^{\mathrm{R}}} \left|\left| \mathbf{U}_i^{\dagger} \mathbf{H}_{ij} \mathbf{V}_j \right|\right|_F^2 = \sum_{i \in \mathcal{K}^{\mathrm{T}}} \Delta_{\mathrm{LI},i}^{\mathrm{T}}(t).$$

Therefore, at iteration t, each Tx weight $\mathbf{U}_i$ computed in Step 2 to minimize $\Delta_{\mathrm{LI},i}^{\mathrm{T}}(t-1)$ also minimizes $\Delta_{\mathrm{LI}}(t-1)$.

On the other hand, the objective function associated with Rx node j in Step 3 is

$$\Delta_{\mathrm{LI},j}^{\mathrm{R}}(t) = \sum_{i \in \mathcal{K}^{\mathrm{T}}}^{i \neq s(j)} \left|\left| \mathbf{U}_i^{\dagger} \mathbf{H}_{ij} \mathbf{V}_j \right|\right|_F^2 .$$

Thus the total leakage interference can also be given by the sum of $\Delta_{\mathrm{LI},j}^{\mathrm{R}}$ over all Rx nodes, i.e.,

$$\Delta_{\mathrm{LI}}(t) = \sum_{j \in \mathcal{K}^{\mathrm{R}}} \sum_{i \in \mathcal{K}^{\mathrm{T}}}^{i \neq s(j)} \left|\left| \mathbf{U}_i^{\dagger} \mathbf{H}_{ij} \mathbf{V}_j \right|\right|_F^2 = \sum_{j \in \mathcal{K}^{\mathrm{R}}} \Delta_{\mathrm{LI},j}^{\mathrm{R}}(t).$$

Therefore, at iteration t, each Rx weight $\mathbf{V}_j$ computed in Step 3 to minimize $\Delta_{\mathrm{LI},j}^{\mathrm{R}}(t-1)$ also minimizes $\Delta_{\mathrm{LI}}(t-1)$.

Since Δ_{LI} is lower bounded by 0 and Δ_{LI} is monotonically non-increasing in each iteration, Algorithm 3.1 must converge to some value no less than 0. □

Although Algorithm 3.1 minimizes leakage interference in each iteration and is proven to converge, the objective value upon this convergence may only be sub-optimal. Nevertheless, we find that this algorithm is computationally efficient. The performance of the algorithm

Algorithm 3.1: Computing Tx and Rx Weights

input : $\mathbf{H}_{ij}$, r_{ij}, z_{ij}, d_{ij}^{R}, d_{ij}^{T}, P_i, L_{ij}
output : $\mathbf{U}_i$, $\mathbf{V}_j$
parameter: ϵ, W

Initialize: Start with arbitrary weight matrices:
    $\mathbf{U}_i$: $N_i^{\mathrm{T}} \times z_{i*}, \mathrm{rank}(\mathbf{U}_i) = z_{i*}$;
    $\mathbf{V}_j$: $N_j^{\mathrm{R}} \times z_{*j}, \mathrm{rank}(\mathbf{V}_j) = z_{*j}$;
    NonImproveIter = 0 ;
while *NonImproveIter* $< W$, **do**
  foreach $i \in \mathcal{K}^{\mathrm{T}}$ **do**
    Solve optimization Problem **OPT-Tx-**i ;
    if $\Delta_{\mathrm{LI},i}^{\mathrm{T}}(t) < \Delta_{\mathrm{LI},i}^{\mathrm{T}}(t-1)$ **then**
      $\mathbf{U}_i \leftarrow$ solution to Problem **OPT-Tx-**i ;
    end
  end
  foreach $j \in \mathcal{K}^{R}$ **do**
    Solve optimization Problem **OPT-Rx-**j ;
    if $\Delta_{\mathrm{LI},j}^{\mathrm{R}}(t) < \Delta_{\mathrm{LI},j}^{\mathrm{R}}(t-1)$ **then**
      $\mathbf{V}_j \leftarrow$ solution to Problem **OPT-Rx-**j ;
    end
  end
  if $\Delta_{\mathrm{LI}}(t-1) - \Delta_{\mathrm{LI}}(t) < \epsilon$ **then**
    $NonImproveIter \leftarrow NonImproveIter + 1$;
  else
    $NonImproveIter = 0$;
  end
end
foreach $j \in \mathcal{K}^{\mathrm{R}}, i \in \mathcal{K}^{\mathrm{T}}$ **do**
  $\mathbf{U}_i \leftarrow$ performing linear transformation by (3.27) ;
  $\mathbf{V}_j, \mathbf{U}_i \leftarrow$ performing equal power allocation by (3.28) ;
end

is presented in the following section.

3.6.3 Performance

In this section, we examine the effectiveness of Algorithm 3.1 in terms of cancelling the strong interference. For evaluation, we first introduce the metric of *normalized residual interference,* which is defined as the ratio of residual interference (i.e., the remaining portion of the strong interference after applying the weights at the PHY layer for IC) to the interference power before this IC.

Recall that $\tilde{\mathbf{H}}_{ij}$ is the best rank-r approximate of channel $\mathbf{H}_{ij}$ (defined in Section 3.6.1). After applying the Tx and Rx weights found by Algorithm 3.1, the residual interference power perceived at Rx node j is $\sum_{i\in\mathcal{K}^{\mathrm{T}}}^{i\neq s(j)} P_i L_{ij} \left|\left|\mathbf{U}_i^{\dagger}\tilde{\mathbf{H}}_{ij}\mathbf{V}_j\right|\right|_F^2$, which we hope to be close to 0 (if our Algorithm 3.1 is effective). The interference power before IC can be expressed as $\sum_{i\in\mathcal{K}^{\mathrm{T}}}^{i\neq s(j)} \frac{P_i}{N_i^{\mathrm{T}}} L_{ij} \left|\left|\mathbf{H}_{ij}\right|\right|_F^2$. Then the normalized residual interference at Rx node j, denoted as $\tilde{\delta}_j$, is

$$\delta_j = \frac{\sum_{i\in\mathcal{K}^{\mathrm{T}}}^{i\neq s(j)} P_i L_{ij} \left|\left|\mathbf{U}_i^{\dagger}\tilde{\mathbf{H}}_{ij}\mathbf{V}_j\right|\right|_F^2}{\sum_{i\in\mathcal{K}^{\mathrm{T}}}^{i\neq s(j)} \frac{P_i}{N_i^{\mathrm{T}}} L_{ij} \left|\left|\mathbf{H}_{ij}\right|\right|_F^2}.$$

Denote δ_{ave} as the average normalized residual interference over all Rx nodes. Then δ_{ave} is given by

$$\delta_{\mathrm{ave}} = \frac{1}{|\mathcal{K}^{\mathrm{R}}|} \sum_{j\in\mathcal{K}^{\mathrm{R}}} \delta_j.$$

We will use δ_{ave} as the primary performance metric to show the effectiveness of Algorithm 3.1.

We consider the same network setting as in Section 3.5. For the parameters of Algorithm 3.1, we set $\epsilon = 0.01$ and $W = 5$. Fig. 3.12 shows that, by applying Algorithm 3.1, the average normalized residual interference is close to zero (less than 0.025) under all different network settings ($\rho = 0.2, 0.4$ and 0.6) and different effective rank thresholds. This demonstrates

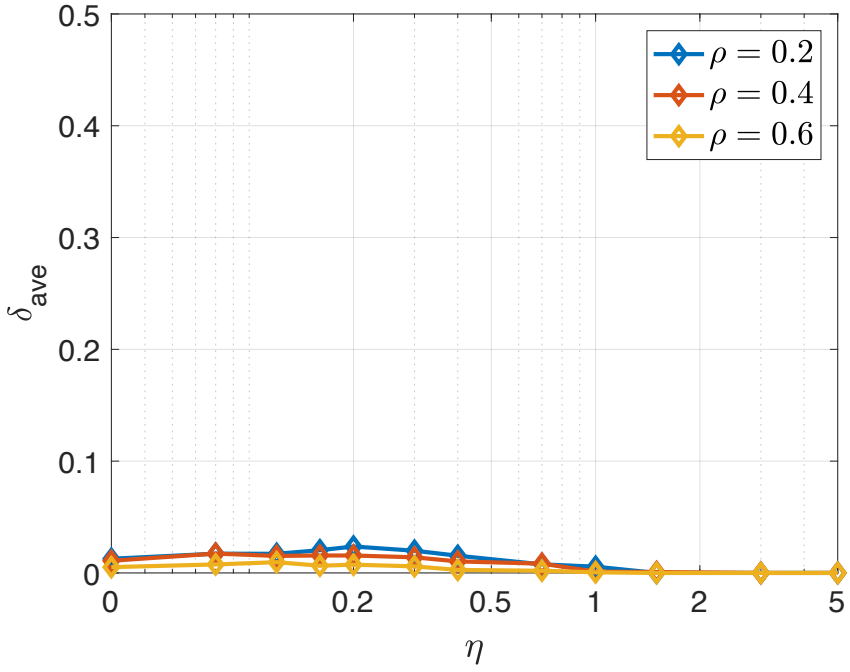


Figure 3.12: The average normalized residual interference under different rank thresholds.

that the Tx and RX weights assigned by Algorithm 3.1 can successfully suppress the strong interference close to zero in all cases. That is, for practical purpose, Algorithm 3.1 can guarantee feasibility at the PHY layer for a given DoF allocation by the DoF IC model in Section 3.4.

3.7 Related Work

DoF-based IC in MIMO networks has been widely studied in the literature. However, none of the existing DoF models differentiate strong and weak interference in different directions in the eigenspace per interference link, as we have done in this chapter.

In the Information Theory (IT) community, DoF characterizations are mainly based on idealized channel models, i.e., either full rank (e.g. [12, 77]) or rank-deficient with zero singular values (e.g. [25, 27, 34, 36]). Such idealized channel rank models do not exactly

capture what happens in reality, where singular values for weak interference are not exactly zero. As a result, they cannot closely represent channel behaviors in the real world.

In the networking community, most existing DoF-based models assume that channels are of full rank [14, 15, 16, 17, 18, 19, 20, 21, 22, 50, 51, 52]. To measure the footprint of interference (and its impact), the so-called protocol model (or disc model) has been widely used [14, 15, 16, 17, 20, 21, 22, 50, 51, 52], where an Rx node within a predetermined interference range is considered interfered and would require DoFs to cancel the interference, while an Rx node outside that range is considered to experience negligible interference (i.e., no IC is needed). The main issue with this model is that, for the same Rx node (inside the interference range), it does not differentiate interference strength in different directions in the eigenspace and thus would require DoFs to cancel interference in all directions (for the same Rx node) even though the signal strength in certain directions may be very weak. The weakness of these models is further amplified when the number of antennas at Tx/Rx nodes becomes large and channels exhibit high correlation. As a result, these models cannot exploit the full potential of MIMO networks. In contrast, instead of using a disc (or interference range), we differentiate interference strength by examining singular values in the eigenspace regardless of the location of the Rx node. Strong interferences (corresponding to large singular values) are cancelled by DoFs while weak interferences (corresponding to small singular values) are treated as noise in throughput calculation. This approach provides efficient DoF utilization that can offer higher throughput.

3.8 Chapter Summary

In this chapter, we developed a novel DoF IC strategy that exploited interference signal strengths among different directions in the eigenspace. By decomposing an interference

channel in its eigenspace and introducing an effective rank threshold to differentiate strong and weak interference, we showed that precious DoFs can be conserved if we only use DoFs to cancel those strong interference signals in the eigenspace. We investigated the trade-off between network throughput and effective rank threshold and showed that network throughput under the optimal effective rank threshold is significantly higher than that under existing DoF IC models. To ensure the new DoF IC model is feasible at the PHY layer, we proposed an algorithm to find the Tx and Rx weights such that the strong interferences beyond the effective rank threshold can be suppressed close to zero.

Chapter 4

A Novel Design and Implementation to Achieve Ultra-Fast Hybrid Beamforming

4.1 Introduction

Communication over mmWave frequencies is defining a new era of wireless communication, including the most recent cellular systems such as 5G NR [78, 79]. At mmWave frequencies, a base station (BS) typically needs to employ hundreds or more antennas to overcome the large path-loss fading. However, it is difficult to apply a dedicated RF chain for *each* antenna as traditional MIMO under 6 GHz, due to hardware complexity and energy consumption issues [79, 80]. To address this problem, the so-called "hybrid architecture" was proposed. As illustrated in Fig. 4.1, the hybrid architecture uses a much fewer number of shared RF chains to support a large number of antennas. This innovative design has attracted a lot of attention from both the academic communities and the industry sections [81, 82, 83, 84, 85, 86, 87, 88, 89, 90].

Although attractive, hybrid architecture faces a critical challenge. Specifically, it must be able to offer a beamforming solution in real-time to be practical. By real-time, we mean that a

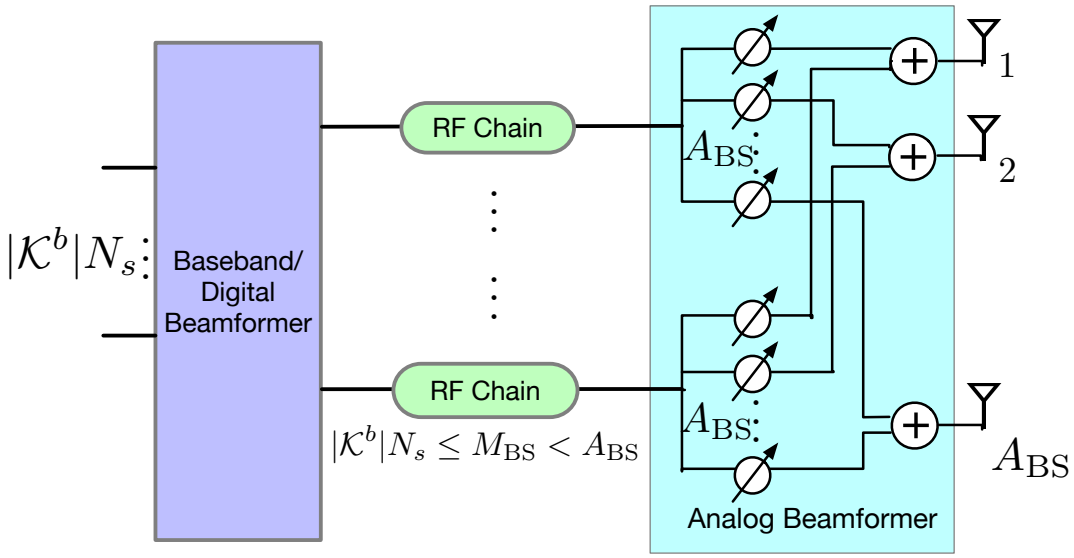


Figure 4.1: An HB architecture (BS side).

beamforming solution must be found within half of the channel coherence time.[1] At mmWave frequencies, this channel coherence time is extremely short, due to the severe Doppler effect. In 5G NR, new frame structures with shorter TTIs (compared to 4G LTE) are designed to support communications over short channel coherence time [91]. Specifically, under 5G NR numerology 0, a TTI is 1 ms, while the TTIs for numerologies 1, 2 and 3 are 500 μs, 250 μs and 125 μs, respectively. The shorter TTIs allow 5G to cope with extremely short coherence time at high frequencies and to support ultra-low latency applications. Therefore, for a hybrid architecture to work under 5G NR, an HB solution must be found within each TTI (corresponding to the applied numerology) to be useful. Further, a beamforming design must consider a large number of resource blocks (RBs), with each RB supporting multiple active users (MU-MIMO).

Although there exist a number of research works in the literature on HB design, few can meet the real-time requirement with high throughput performance. For instance, physical

[1]For efficiency, we break up the channel coherence time into two halves. Within each half, we transmit data based on beamforming matrices that are computed in the previous half and compute the beamforming matrices for the next half.

(PHY) layer research in this area attempted to jointly optimize analog and digital beamforming [81, 82, 83, 84]. Unfortunately, the iterative nature of these algorithms makes them difficult to be implemented in real-time. In addition, a joint design requires explicit antenna-to-antenna channel estimation and feedback, which involves a prohibitively high complexity and a large amount of CSI that is too difficult to obtain in practice [85].

To avoid the issues associated with a joint design, a new and practical direction for HB is to follow a sequential design [86, 87, 88, 89, 90]. Here, an analog beamforming is optimized first and then used as the input to optimize the digital beamforming. For analog beamforming, there have been successful designs and system demonstrations in the literature, which are based on beam sweeping/discovering techniques without explicit channel CSI [88, 89, 90]. After analog beamforming is applied to both the BS and a user's side, the effective channels seen at the baseband can be obtained through conventional channel estimation approaches.

However, how to properly design digital beamforming in a sequential design remains a challenge. Most existing works simply applied traditional beamforming methods such as ZF, MMSE and Block Diagonalization (BD) [23] as the digital beamformers [86, 87, 88, 89]. Although simple, ZF and MMSE typically experience inferior throughput performance for MU-MIMO and mmWave systems, particularly under ill-conditioned channels [88, 92, 93]. Although BD beamforming and its variants are shown to improve ZF/MMSE with a much better throughput performance [23, 94], it requires many high-dimensional matrix SVD operations, which are of high complexity and require significant computation time.

As expected, finding a beamforming scheme that can meet both real-time requirement and high throughput performance is not trivial. But recent advances in parallel architectures (based on the many-core technology) have shed new light on this problem. In particular, the general-purpose GPU-based platform (e.g., those from Nvidia) is particularly promising. Its

dedicated single-instruction-multiple-data (SIMD) architecture can solve a massive number of structurally-identical problems at an extremely fast speed. It also comes with highly programmable tools such as CUDA, making the real-time implementation feasible and flexible to many developers. A GPU-based parallel computing platform now offers a new possibility to tackle many hard problems whose real-time solutions are once considered elusive [96].

In this chapter, we present *Turbo-HB,*[2] a GPU-based novel design and implementation to achieve ultra-fast digital beamforming. The key ideas of Turbo-HB are twofold. First, we identify the bottleneck of computation time for BD-type beamforming, which attributes to high-dimensional SVD operations. Turbo-HB cuts down this computational complexity by utilizing randomized SVD technique. Second, Turbo-HB accelerates the overall computation time through large-scale parallel computation on a commercial off-the-shelf (COTS) GPU platform. It incorporates a large number of matrix transformations in parallel and special engineering efforts such as minimized memory access. The main contributions of this chapter are summarized as follows:

- This chapter presents Turbo-HB, the first successful HB design that can meet the sub-ms real-time requirement. This design considers a large number of RBs with MU-MIMO capability, which can be applied to 5G cellular systems. Our design only relies on a COTS GPU platform and does not require any customized hardware.

- Turbo-HB relieves the computational burden of SVD significantly by leveraging the sparsity at mmWave channels. Specifically, Turbo-HB is able to identify a small number of the most significant directions on a mmWave channel by exploiting randomized SVD technique. By limiting operations only to the key information of interests, high-dimensional SVD operations are transformed into lightweight lower-rank matrix op-

[2]By "Turbo," we mean fast and efficient.

erations. By judiciously choosing a proper target rank for lower-rank approximation, our design can reduce the computation time dramatically.

- Turbo-HB is capable of parallelizing the MU-MIMO beamforming for a large number of RBs and users. First, the MU-MIMO beamforming is transformed into a set of parallel single-user MIMO (SU-MIMO) beamforming. Second, with customized nullspace calculation based on Given's rotation method, Turbo-HB accelerates computation and fully utilizes GPU's processing cores. Third, by employing batched matrix operation with proper indexing method and utilizing shared memory, Turbo-HB achieves large-scale parallel matrix operations.

- We implement Turbo-HB on Nvidia DGX Station using the CUDA programming platform. Extensive experiments are performed to examine both the timing performance and throughput performance. Experimental results show that Turbo-HB is able to obtain the beamforming matrices far less than 1 ms for all tested cases. Specifically, Turbo-HB can meet the 125μs, 250 μs, and 500 μs timing requirement for 100 RBs with up to 4, 8, and 10 MU-MIMO users on each RB, respectively. Turbo-HB can also offer higher throughput performance for most cases compared to the state-of-the-art (non-real-time) algorithms.

4.2 System Model

We consider a cellular communication scenario where a BS serves a set $\mathcal{K}$ of users, as illustrated in Fig. 4.2. The BS is equipped with A_{BS} antennas and M_{BS} RF chains. Under HB architecture, $M_{\mathrm{BS}} < A_{\mathrm{BS}}$. Each user is equipped with A_{U} antennas and M_{U} RF chains, and $M_{\mathrm{U}} < A_{\mathrm{U}}$. Since the mathematical structure for uplink (UL) and downlink (DL) is symmetric, it is sufficient to study one of them. We focus on DL in this chapter.

Table 4.1: Notations in Chapter 4

Symbol	Definition
A_{BS}	Number of antennas at BS
A_{U}	Number of antennas at user
$\mathcal{B}$	A set of RBs to be allocated in a time slot
$\mathbf{F}_{\mathrm{BB}}$	Baseband precoder at BS
$\mathbf{F}_{\mathrm{RF}}$	Analog precoder at BS
$\mathcal{K}$	A set of users
$\mathcal{K}^b$	A subset of users using RB b
M_{BS}	Number of RF chains at BS
M_{U}	Number of RF chains at user
N_s	Number of data streams on a link
$\mathbf{W}_{\mathrm{BB},k}$	Baseband combiner at user k
$\mathbf{W}_{\mathrm{RF},k}$	Analog combiner at user k

Considering a typical cellular system (e.g., 4G LTE and 5G NR), we study time-slotted scheduling over a wide bandwidth. Within each time slot, there is a set $\mathcal{B}$ of RBs over the DL bandwidth. For each RB $b \in \mathcal{B}$, a subset of users $\mathcal{K}^b \subseteq \mathcal{K}$ is selected for MU-MIMO transmission, based on some RB allocation strategy (see, e.g., [97, 98]). For the ease of notation, suppose the BS sends N_s data streams to each user.[3] At the user side, since the number of received data streams cannot exceed the number of its RF chains, we have $N_s \leq M_{\mathrm{U}}$. Likewise, at the BS we have $|\mathcal{K}^b| N_s \leq M_{\mathrm{BS}}$.

Under the HB architecture, beamforming is performed in both digital and analog domains, as shown in Fig. 4.1. At the BS side, the transmitted signal is first processed in the digital domain by an $M_{\mathrm{BS}} \times |\mathcal{K}^b| N_s$ baseband precoder $\mathbf{F}_{\mathrm{BB}}$. Subsequently, an $A_{\mathrm{BS}} \times M_{\mathrm{BS}}$ analog precoder $\mathbf{F}_{\mathrm{RF}}$ (also known as RF precoder) based on analog circuitry (phase shifters) is applied in the analog domain. Since complex matrix $\mathbf{F}_{\mathrm{RF}}$ is implemented with analog phase shifters, each element in the matrix has the same amplitude and differs in its phase, i.e, $|(\mathbf{F}_{\mathrm{RF}})_{i,j}| = \frac{1}{\sqrt{A_{\mathrm{BS}}}}$, where $(\cdot)_{i,j}$ denotes the (i,j)-th element of matrix $(\cdot)$. In addition, to

[3] With additional notation, our results can be extended to the case where the BS sends a different number of data streams to different users.

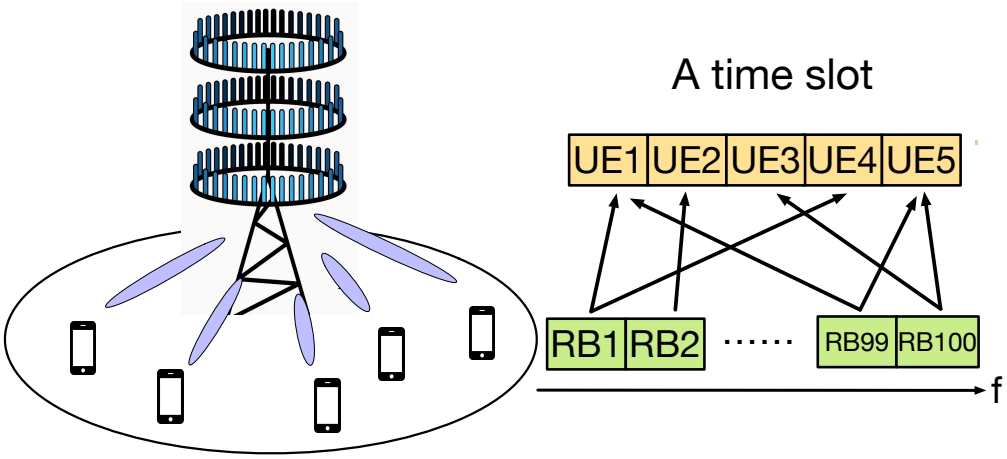


Figure 4.2: A cellular system consisting a large number of RBs (with MU-MIMO capability).

meet the total power constraint at the BS, $\mathbf{F}_{\mathrm{BB}}$ and $\mathbf{F}_{\mathrm{RF}}$ must satisfy $||\mathbf{F}_{\mathrm{RF}}\mathbf{F}_{\mathrm{BB}}||_F^2 \leq P_{\mathrm{T}}$, where P_{T} is the total power at the BS and $||\cdot||_F$ denotes the Frobenius norm.

For wireless channels, let $\mathbf{H}_k^b \in \mathbb{C}^{A_{\mathrm{U}} \times A_{\mathrm{BS}}}$ denote the channel matrix for user $k \in \mathcal{K}$ on RB $b \in \mathcal{B}$, and $\boldsymbol{n}_k^b$ is the $A_{\mathrm{U}} \times 1$ vector of i.i.d $\mathcal{CN}(0, \sigma^2)$ additive complex Gaussian noise. Let $\mathbf{F}_{\mathrm{BB}}^b$ and $\mathbf{F}_{\mathrm{RF}}^b$ denote the baseband precoder and analog precoder for RB b, respectively. Then the received signal of user k on RB b is given by

$$\boldsymbol{y}_k^b = \mathbf{H}_k^b \mathbf{F}_{\mathrm{RF}}^b \mathbf{F}_{\mathrm{BB}}^b \boldsymbol{s}^b + \boldsymbol{n}_k^b, \quad (k \in \mathcal{K}^b, b \in \mathcal{B}) \tag{4.1}$$

where $\boldsymbol{s}^b$ is the signal vector.

At the user side, a symmetric HB structure is employed except with a fewer number of antennas A_{U} and a fewer number of RF chains M_{U}. The received signal is first processed by an $A_{\mathrm{U}} \times M_{\mathrm{U}}$ analog combiner $\mathbf{W}_{\mathrm{RF},k}$ (subject to $|(\mathbf{W}_{\mathrm{RF},k})_{i,j}| = \frac{1}{\sqrt{A_{\mathrm{U}}}}$) in analog domain. Then an $M_{\mathrm{U}} \times N_s$ baseband combiner $\mathbf{W}_{\mathrm{BB},k}$ is applied.

Denote $\widehat{\mathbf{H}}_k^b$ as the effective channel seen at the baseband, i.e., $\widehat{\mathbf{H}}_k^b = \mathbf{W}_{\mathrm{RF},k}^{b\dagger} \mathbf{H}_k^b \mathbf{F}_{\mathrm{RF}}^b$. Denote $\mathbf{F}_{\mathrm{BB},k}^b$ as a sub-matrix of $\mathbf{F}_{\mathrm{BB}}^b = [\mathbf{F}_{\mathrm{BB},1}^b \cdots \mathbf{F}_{\mathrm{BB},k}^b \cdots \mathbf{F}_{\mathrm{BB},|\mathcal{K}^b|}^b]$, where $\mathbf{F}_{\mathrm{BB},k}^b$ consists of

N_s columns and corresponds to the baseband signal $\boldsymbol{s}_k^b$ for user k. Then at user k and on RB b, we have the following signal:

$$\begin{aligned}\tilde{\boldsymbol{y}}_k^b = \mathbf{W}_{\mathrm{BB},k}^{b\dagger}\widehat{\mathbf{H}}_k^b\mathbf{F}_{\mathrm{BB},k}^b\boldsymbol{s}_k^b + \sum_{i\in\mathcal{K}^b}^{i\neq k}\mathbf{W}_{\mathrm{BB},k}^{b\dagger}\widehat{\mathbf{H}}_k^b\mathbf{F}_{\mathrm{BB},i}^b\boldsymbol{s}_i^b \\ + \mathbf{W}_{\mathrm{BB},k}^{b\dagger}\mathbf{W}_{\mathrm{RF},k}^{b\dagger}\boldsymbol{n}_k^b, \qquad (k\in\mathcal{K}^b, b\in\mathcal{B})\end{aligned}$$

where $(\cdot)^\dagger$ denotes the conjugate transpose of a matrix.

Therefore, the network throughput in b/s/Hz is

$$\begin{aligned}C = \sum_{b\in\mathcal{B}}\sum_{k\in\mathcal{K}^b}\log\Big(\Big|\mathbf{I}_{N_s} + \\ (\mathbf{Q}_k^b)^{-1}\mathbf{W}_{\mathrm{BB},k}^{b\dagger}\widehat{\mathbf{H}}_k^b\mathbf{F}_{\mathrm{BB},k}^b\ \mathbf{F}_{\mathrm{BB},k}^{b\dagger}\widehat{\mathbf{H}}_k^{b\dagger}\mathbf{W}_{\mathrm{BB},k}^b\Big|\Big),\end{aligned} \tag{4.2}$$

where $(\mathbf{Q}_k^b)^{-1}$ is the covariance matrix of both interference and noise, which is given by

$$\begin{aligned}(\mathbf{Q}_k^b)^{-1} = \sum_{i\in\mathcal{K}^b}^{i\neq k}\mathbf{W}_{\mathrm{BB},k}^{b\dagger}\widehat{\mathbf{H}}_k^b\mathbf{F}_{\mathrm{BB},i}^b\mathbf{F}_{\mathrm{BB},i}^{b\dagger}\widehat{\mathbf{H}}_k^{b\dagger}\mathbf{W}_{\mathrm{BB},k}^b \\ + \sigma^2\mathbf{W}_{\mathrm{BB},k}^{b\dagger}\mathbf{W}_{\mathrm{RF},k}^{b\dagger}\mathbf{W}_{\mathrm{RF},k}^b\mathbf{W}_{\mathrm{BB},k}^b.\end{aligned}$$

Then the throughput optimization problem under the HB architecture can be stated as following:

OPT-HB

$$\max\ \ C\left(\mathbf{F}_{\mathrm{RF}}^b, \mathbf{F}_{\mathrm{BB}}^b, \mathbf{W}_{\mathrm{RF},k}^b, \mathbf{W}_{\mathrm{BB},k}^b\right)$$

s.t. Power constraint: $||\mathbf{F}_{\mathrm{RF}}^b\mathbf{F}_{\mathrm{BB}}^b||_F^2 \leq P_{\mathrm{T}}$;

Constant modulus constraints:

$$|(\mathbf{F}_{\mathrm{RF}}^b)_{i,j}| = \frac{1}{\sqrt{A_{\mathrm{BS}}}},\ |(\mathbf{W}_{\mathrm{RF},k}^b)_{m,n}| = \frac{1}{\sqrt{A_{\mathrm{U}}}};$$

$$\text{Index range: } b \in \mathcal{B},\ k \in \mathcal{K},$$
$$i \in \{1, 2, \cdots, A_{\text{BS}}\},\ j \in \{1, 2, \cdots, M_{\text{BS}}\},$$
$$m \in \{1, 2, \cdots, A_{\text{U}}\},\ n \in \{1, 2, \cdots, M_{\text{U}}\}.$$

In problem OPT-HB, the variables are digital and analog beamformers $\mathbf{F}^b_{\text{RF}}, \mathbf{F}^b_{\text{BB}}, \mathbf{W}^b_{\text{RF},k}$ and $\mathbf{W}^b_{\text{BB},k}$, while $P_{\text{T}}, A_{\text{BS}}, A_{\text{U}}, M_{\text{BS}}, M_{\text{U}}$ are constants and $\mathcal{B}$ and $\mathcal{K}^b$ are given sets.

Ideally, a joint optimization of all digital and analog beamformers is required to find a global optimal solution. However, several practical issues make such a joint design infeasible. For example, the amount of CSI required is prohibitively large; it is unclear how to estimate the antenna-to-antenna channel $\mathbf{H}^b_k$ through the lens of the RF precoding and combining [85]. A new and practical direction to address HB optimization is to follow a sequential design. Under this approach, analog domain is optimized first and then used as input to optimize the digital design [86, 87, 88, 89, 90]. It has been shown that such a sequential approach can offer a competitive performance (compared to those heuristics attempting to solve joint optimization [82, 84, 86, 100]).

Even with a sequential method, for MU-MIMO systems, it would still require enormous computational efforts to find a local optimum [101], due to the high complexity of high-dimensional matrix operations (in addition to non-convex programming). We discuss this problem in detail in the following section.

4.3 Real-Time Requirement

In 5G NR, the frame structure is designed to be scalable to accommodate diverse services and channel conditions. Under 5G frame structures, a beamforming solution (for all users on all RBs) must be obtained within 1 ms (numerology 0), 500 μs (numerology 1), 250 μs (numerology 2), or 125 μs (numerology 3). A shorter TTI can support applications with shorter coherence time and more stringent latency requirement.

Note that under the HB architecture, analog beamforming is meant to overcome path-loss fading by leveraging the large number of antennas [86, 90]. This part is done on a much larger time scale. In contrast, digital beamforming can optimize capacity by managing interference among data streams, which heavily depends on fast fading. This part has a much stringent timing requirement. Therefore, under a sequential design, the stringent sub-ms real-time requirement mainly comes from digital beamforming.

Technical Challenge Digital beamforming for MU-MIMO involves complex operations of matrices with a large number of elements. Traditional techniques such as ZF and MMSE typically experience inferior throughput performance for MU-MIMO and mmWave systems, particularly under ill-conditioned channels [88, 92, 93]. On the other hand, BD-type beamforming is shown to achieve much better throughput performance compared to ZF/MMSE [23]. But BD involves high-dimensional matrix SVD operations, whose computational complexity makes BD unsuitable for practical use.

Objective The objective of this chapter is to determine digital beamformers ($\mathbf{F}^b_{\mathrm{BB},k}$ and $\mathbf{W}^b_{\mathrm{BB},k}$) in real-time. Specifically, we want to develop a design that can meet the stringent sub-ms timing requirement while offering comparable (or better) throughput performance than state-of-the-art approaches.

4.4 A Novel Design for Real-time Beamforming

4.4.1 Main Ideas

Our main ideas consist of two parts.

Low-complexity SVD with high throughput First, we show the high computation time for BD-type beamforming is attributed to the high-dimensional SVD operations. Then we propose to reduce this complexity by identifying only a small number for the most significant dimensions, leveraging the sparsity of mmWave channels. Specifically, for a $(|\mathcal{K}^b|-1)M_{\mathrm{U}} \times M_{\mathrm{BS}}$ matrix (for BD beamforming), a standard SVD algorithm takes $O\left(\left[(|\mathcal{K}^b|-1)M_{\mathrm{U}}\right]^2 M_{\mathrm{BS}}\right)$ floating-point operations (flops) [108, 109]. Thus, applying BD beamforming for $|\mathcal{B}|$ RBs and $|\mathcal{K}^b|$ users at each RB yields at least $O\left(|\mathcal{B}||\mathcal{K}^b|\left[(|\mathcal{K}^b|-1)M_{\mathrm{U}}\right]^2 \cdot M_{\mathrm{BS}}\right)$ flops. To reduce this high complexity, we propose to utilize randomized SVD [108] to cut down the complexity to $O\left(|\mathcal{B}||\mathcal{K}^b| \cdot r^2 \cdot \left[(|\mathcal{K}^b|-1)M_{\mathrm{U}} + M_{\mathrm{BS}}\right]\right)$, where r is much smaller than $(|\mathcal{K}^b|-1)M_{\mathrm{U}}$. In essence, randomized SVD is a lower rank SVD approximation method. The reason why it works extremely well here is because of the limited number of scatterers at mmWave frequencies and thus highly correlated channels. In addition, by limiting the operations to the key information of our interest and applying the parallelizable Given's rotation method, the lower rank SVD can be done extremely fast in our implementation.

Interestingly, although Turbo-HB employs a lower rank SVD approximation, it does not mean the throughput performance needs to deteriorate. Rather, Turbo-HB appears to offer higher throughput performance in most cases. The science behind this behavior is attributed to the following. First, since mmWave channels exhibit a high correlation property, a small set of singular vectors in the lower rank SVD approximation is sufficient to capture the directions of the most significant signals or interferences. Second, an exact $(|\mathcal{K}^b|-1)M_{\mathrm{U}} \times M_{\mathrm{BS}}$ matrix

SVD (as in standard BD) aims to cancel *all* inter-user interference exactly (regardless of how small it is). But canceling all inter-user interference requires to project users' signals onto mutually orthogonal subspaces. To achieve such orthogonality, the perceived strength of desired signals at a user is reduced in the process. Since throughput is a function of SINR, it does not help if the perceived strength of desired signals at a user is reduced (for perfect orthogonality). On the other hand, a lower rank SVD approximation allows a certain level of overlapping subspace of different users (as only a small number of major signals preserve mutual orthogonality), which in return preserves greater desired signal strength. This offers us an opportunity to explore the promising beamforming space that is missed by the BD technique.

Fully functioning parallelism We argue that the asymptotic complexity analysis (i.e., those expressed in the big-O notation) does not directly translate into actual computation time as measured by a wall clock for our problem. The latter heavily depends on the underlying problem structure, actual input size, convergence speed, memory access time, among others. This motivates us to our second idea, which is to accelerate overall computing process in real-time, rather than focusing on $O(\cdot)$ analysis. We propose to design a beamforming algorithm with parallelizable implementation, incorporating special engineering efforts such as minimizing memory access.

Specifically, the MU-MIMO beamforming is first transformed into a set of parallel SU-MIMO beamforming. Then a large number of matrix operations are executed through batch computing. To achieve batched matrix operations (for a large number of RBs and users), Turbo-HB generates a large number of threads that fully occupy a GPU's processing cores and thus reaps the full benefits of GPU's parallel processing capability. At each step throughout our implementation, we meticulously minimize memory accesses to reduce time. For example, batched matrix operations such as QR factorization and matrix multiplications are

optimized with the use of fast on-chip shared memory. We carefully organize the storage of a large number of matrices with proper indexing. By managing consecutive GPU threads to read consecutive (and aligned) memory, multiple memory accesses can be combined into a single transaction. Further, Turbo-HB limits operations to the key information of our interests (e.g., certain singular vectors) and thus eliminates unnecessary calculations, parameter passing and memory access.

4.4.2 Design Details

The task of computing beamforming matrices can be split naturally into three computational stages. The first is to transform the MU-MIMO channel into a set of parallel SU-MIMO channels. The second is to apply randomized SVD with low computation complexity to obtain certain singular vectors for beamforming. The third is to construct the final digital beamforming matrices based on obtained singular vectors. Specifically, the objective of each stage is described as follows.

- **Stage A:** Given the partial CSI $\mathbf{V}_k^b$ and $\mathbf{\Sigma}_k^b$ (from $\widehat{\mathbf{H}}_k^b = \mathbf{U}_k^b \mathbf{\Sigma}_k^b \mathbf{V}_k^{b\dagger}$) that are computed and fed back by each user, we construct matrices $\overline{\mathbf{H}}_k^b$ and $\widetilde{\mathbf{H}}_k^b$ such that $\overline{\mathbf{H}}_k^b$ and $\widetilde{\mathbf{H}}_k^b$ contain all the information that is needed to compute beamforming matrices $\mathbf{F}_{\mathrm{BB},k}^b$ corresponding to user k. After this stage, the MU-MIMO channel is transformed into a set of parallel SU-MIMO channels.

- **Stage B:** Given matrix $\widetilde{\mathbf{H}}_k^b$, we apply randomized SVD technique for lower rank matrix approximation (with lower computational complexity). Then we obtain $\widetilde{\mathbf{V}}_k^{b(-)}$, which contains the necessary singular vectors to cancel inter-user interference.

- **Stage C:** With matrices $\overline{\mathbf{H}}_k^b$ and $\widetilde{\mathbf{V}}_k^{b(-)}$, we construct the final digital beamforming matrices $\mathbf{F}_{\mathrm{BB},k}^b$.

In the rest of this section, we offer details of each stage.

Stage A. Each user k estimates the effective channel $\widehat{\mathbf{H}}_k^b$ and computes its SVD as $\widehat{\mathbf{H}}_k^b = \mathbf{U}_k^b \mathbf{\Sigma}_k^b \mathbf{V}_k^{b\dagger}$. User k uses the first N_s columns of $\mathbf{U}_k^b$ as its digital combiner, i.e., $\mathbf{W}_{\mathrm{BB},k}^b$ is set to the first N_s columns of $\mathbf{U}_k^b$. Then to help form digital precoder at BS side, only partial CSI, i.e., $\mathbf{V}_k^b$ and $\mathbf{\Sigma}_k^b$, are required to feed back to the BS (note that $\mathbf{\Sigma}_k^b$ is diagonal and $\mathbf{V}_k^b$ is unitary and thus can be efficiently compressed [2]). Let

$$\overline{\mathbf{H}}_k^b = \mathbf{\Sigma}_k^b \mathbf{V}_k^b.$$

Then for our beamforming purpose, $\overline{\mathbf{H}}_k^b$ (an $M_{\mathrm{U}} \times M_{\mathrm{BS}}$ matrix) captures sufficient information of the intended channel from the BS to user k.

Denote $\widetilde{\mathbf{H}}_k^b$ as the concatenation of $\overline{\mathbf{H}}_k^b$'s of all users in $\mathcal{K}^b$ except intended user k, i.e., if $\mathcal{K}^b = \{k\} \bigcup \{1, \cdots, k-1, k+1, \cdots, |\mathcal{K}^b|\}$, then

$$\widetilde{\mathbf{H}}_k^b = \left[\overline{\mathbf{H}}_1^{b\dagger} \cdots \overline{\mathbf{H}}_{k-1}^{b\dagger} \;\; \overline{\mathbf{H}}_{k+1}^{\dagger} \cdots \overline{\mathbf{H}}_{|\mathcal{K}^b|}^{b\dagger}\right]^{\dagger}$$

is a $(|\mathcal{K}^b| - 1)M_{\mathrm{U}} \times M_{\mathrm{BS}}$ matrix that captures information of interference channels corresponding to user k.

As $\overline{\mathbf{H}}_k^b$ and $\widetilde{\mathbf{H}}_k^b$ are sufficient to construct the beamforming matrices $\mathbf{F}_{\mathrm{BB},k}^b$ corresponding to user k, the MU-MIMO channel is transformed into a set of $|\mathcal{K}^b|$ parallel SU-MIMO channels on each RB. Consequently, the remaining Stage B and Stage C can be processed in $\sum_{b \in \mathcal{B}} |\mathcal{K}^b|$ parallel flows, each of which contributes to one beamforming matrix for one user per RB.

Stage B. To construct beamforming matrix $\mathbf{F}_{\mathrm{BB},k}^b$ corresponding to user k's signal, we need to make sure that by applying $\mathbf{F}_{\mathrm{BB},k}^b$ most (if not all) of the interference to user k can

be canceled. This can be realized with the help of SVD of interference channel $\widetilde{\mathbf{H}}_k^b$. Let

$$\widetilde{\mathbf{H}}_k^b = \widetilde{\mathbf{U}}_k^b \begin{bmatrix} \widetilde{\mathbf{\Sigma}}_k^b & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} [\widetilde{\mathbf{V}}_k^{b(+)} \ \ \widetilde{\mathbf{V}}_k^{b(-)}]^\dagger, \tag{4.3}$$

where $\widetilde{\mathbf{V}}_k^{b(-)}$ is the last $(M_{\mathrm{BS}} - r)$ columns of the right singular matrix corresponding to the smallest $(M_{\mathrm{BS}} - r)$ singular values of $\widetilde{\mathbf{H}}_k^b$, $\widetilde{\mathbf{V}}_k^{b(+)}$ is the remaining r columns of the right singular matrix, and r is a constant.

Then, if the eigenvalues corresponding to $\widetilde{\mathbf{V}}_k^{b(-)}$ are close to zero, we have

$$\widetilde{\mathbf{H}}_k^b \widetilde{\mathbf{V}}_k^{b(-)} \approx \mathbf{0}, \quad (b \in \mathcal{B}, k \in \mathcal{K}^b).$$

It follows that

$$\widehat{\mathbf{H}}_j^b \widetilde{\mathbf{V}}_k^{b(-)} \overline{\mathbf{V}}_k^{b(+)} \approx \mathbf{0}, \quad \text{for } j \neq k,$$

for any $\overline{\mathbf{V}}_k^{b(+)}$ (which is used to differentiate data streams within a user and will be determined later). Therefore, by constructing $\mathbf{F}_{\mathrm{BB},k}^b$ as

$$\mathbf{F}_{\mathrm{BB},k}^b = \widetilde{\mathbf{V}}_k^{b(-)} \overline{\mathbf{V}}_k^{b(+)}, \tag{4.4}$$

most of the inter-user interference can be suppressed.

Now we have a real-time challenge. Stage B is computation-intensive as a high-dimensional SVD (i.e., Eq. (4.3)) is required. $\widetilde{\mathbf{H}}_k^b$ is a $(|\mathcal{K}^b| - 1)M_{\mathrm{U}} \times M_{\mathrm{BS}}$ matrix with standard SVD complexity of $O\left(|\mathcal{B}||\mathcal{K}^b| \cdot \left[(|\mathcal{K}^b| - 1)M_{\mathrm{U}}\right]^2 M_{\mathrm{BS}}\right)$ for $|\mathcal{B}|$ RBs. Its computation time can take more than 70% of the total time when not optimized (from our experiment).

In fact, the computation time of matrix SVD (power method) is tightly related to the

decaying speed of singular values [110]. For instance, suppose we have a matrix with 4 decreasing singular values $\sigma_1, \sigma_2, \sigma_3$ and σ_4. If $\sigma_1 \gg \sigma_2 \gg \sigma_3 \approx \sigma_4 \approx 0$, then it is computationally fast to obtain the first two singular values (and associated singular vectors), whereas it would take much longer to obtain the last two singular values. This observation is especially important, since at mmWave frequencies, most signal strength will be concentrated at a few directions due to the limited number of scatterers. As a consequence, it is likely that we encounter several non-zero but close-to-zero singular values. Finding those small singular values would take a long time and it does not help much in terms of throughput performance (as we shall see in Section 4.4.3).

To verify the singular values of $\widetilde{\mathbf{H}}_k^b$, we conduct the following experiment. We generate 100 instances of $\widetilde{\mathbf{H}}_k^b$ based on mmWave channel model to have $\mathbf{H}_k^b$'s (using the widely adopted mmWave channel model as described in [82]). For analog beamforming, we adopt the well-known DFT-codebook based method [86, 111]. We set $A_{\mathrm{BS}} = 128$, $A_{\mathrm{U}} = 8$, $M_{\mathrm{BS}} = 20$, $M_{\mathrm{U}} = 4$ and $|\mathcal{K}^b| = 5$, thus $\widetilde{\mathbf{H}}_k^b$ is a 16×20 matrix. We investigate two different scattering scenarios: (a) The number of clusters L_{cl} and the number of rays within each cluster L_{ray} are both set to 3; (b) L_{cl} and L_{ray} are both 6 (as typical number of paths for practical mmWave channels [80, 88, 90, 112]). Averaged by 100 instances, the singular values of $\widetilde{\mathbf{H}}_k^b \widetilde{\mathbf{H}}_k^{b\dagger}$ are plotted in Fig. 4.3. As we expected, the singular values are decaying fast in the beginning but then flatten out. The decaying speed is faster when the number of paths is smaller. More importantly, the last several singular values are pretty small but very close. This means the corresponding directions in the eigenspace have very weak signals but consume much computational effort to differentiate them, which is wasteful.

Following the above analysis, our next objective is to implement a lower rank SVD approximation with lower computational complexity. To this end, we apply randomized SVD technique [108]. The key idea of randomized SVD is that with the help of a random

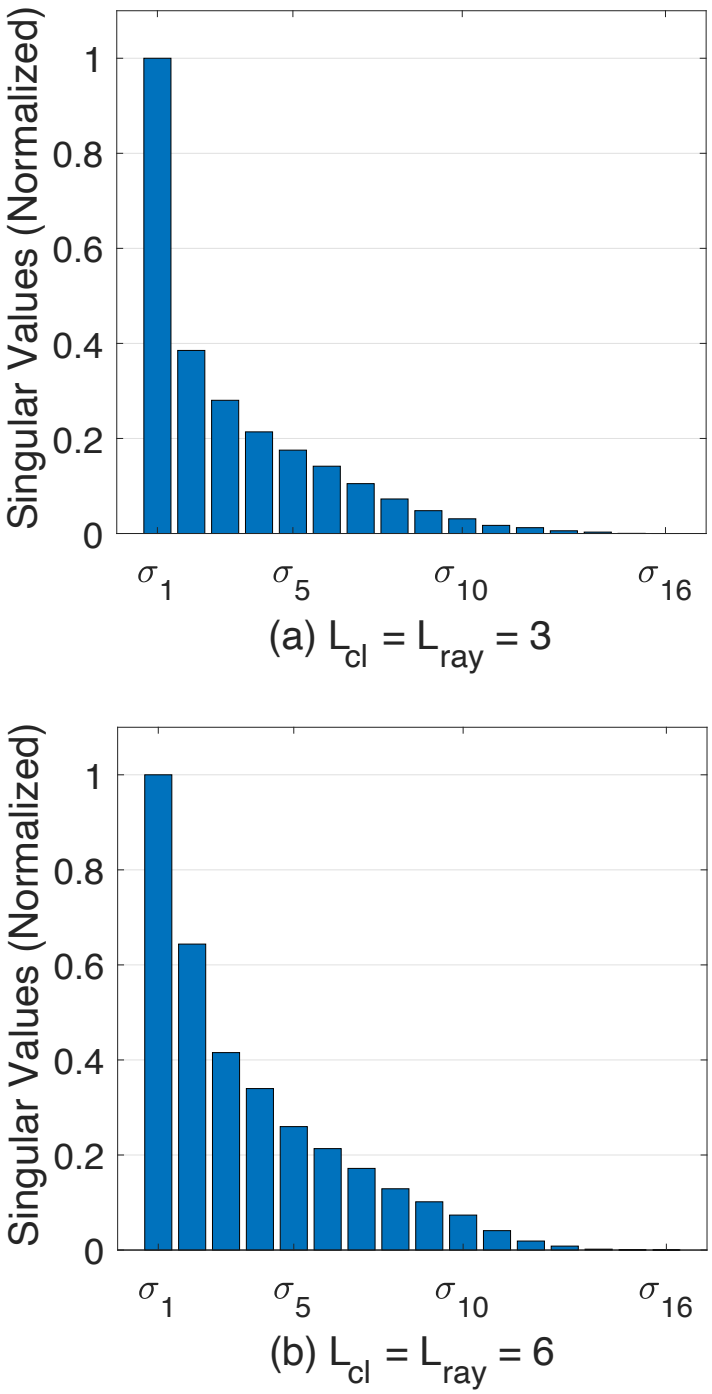


Figure 4.3: Singular values of $\widetilde{\mathbf{H}}_k^b$ (averaged over 100 instances) under different number of scatterers based on mmWave channel modelling.

Algorithm 4.1: Raw Randomized SVD

Given an $m \times n$ matrix $\mathbf{A}$, a target approximation rank r, and an exponent q (say $q = 1$ or $q = 2$), this procedure computes an approximate rank-r factorization $\mathbf{A} \approx \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\dagger$:

1 Generate an $n \times r$ Gaussian matrix $\mathbf{\Omega}$.
2 Form the $m \times r$ matrix $\mathbf{Y} = (\mathbf{A}\mathbf{A}^\dagger)^q\mathbf{A}\mathbf{\Omega}$ by multiplying alternately with $\mathbf{A}$ and $\mathbf{A}^\dagger$.
3 Construct an $m \times r$ matrix $\mathbf{P}$ whose columns form an orthonormal basis for the range of $\mathbf{Y}$.
4 Form the $r \times n$ matrix $\mathbf{B} = \mathbf{P}^\dagger\mathbf{A}$.
5 Compute an SVD of the small matrix: $\mathbf{B} = \check{\mathbf{U}}\mathbf{\Sigma}\mathbf{V}^\dagger$.
6 Set $\mathbf{U} = \mathbf{P}\check{\mathbf{U}}$.

Gaussian matrix $\mathbf{\Omega}$ we form a rank-r basis $\mathbf{P}$, with $r < (|\mathcal{K}^b| - 1)M_{\mathrm{U}} < M_{\mathrm{BS}}$, that captures the dominant directions with the largest SVD singular values. Then the original matrix is projected onto a lower-dimensional subspace (based on basis $\mathbf{P}$) to compute a standard rank-r SVD. We summarize this procedure in Algorithm 4.1 and call it Raw Randomized SVD. $\widetilde{\mathbf{H}}_k^b$ will be used as input for Algorithm 4.1. As we see in Step 5 of Algorithm 4.1, due to the lower rank r, only a small-scale SVD is required. The complexity of Step 5 is $O\left(r^2 \cdot \left[(|\mathcal{K}^b| - 1)M_{\mathrm{U}} + M_{\mathrm{BS}}\right]\right)$, which has been reduced from $O\left(\left[(|\mathcal{K}^b| - 1)M_{\mathrm{U}}\right]^2 \cdot M_{\mathrm{BS}}\right)$. How to choose a proper value of r will be discussed in the next section.

We now customize the Raw Randomized SVD to further expedite computation time. Note that from Eq. (4.3) and Eq. (4.4) in Stage B, our interest is $\widetilde{\mathbf{V}}_k^{b(-)}$, the last $(M_{\mathrm{BS}} - r)$ columns of the right singular matrix corresponding to the smallest $(M_{\mathrm{BS}} - r)$ singular values of $\widetilde{\mathbf{H}}_k^b$, while the singular values and left singular matrix are not necessary for the beamforming design. However, by examining Algorithm 4.1, a thorough lower rank SVD is performed, including the calculation of unnecessary singular vectors and singular values (see Steps 3 to 6). Therefore, we can customize Steps 3 to 6 to reduce time. First, note that $\widetilde{\mathbf{V}}_k^{b(-)}$ is also the nullspace of $\mathbf{Y}^\dagger\mathbf{A}$. The calculation of the orthogonalization and normalization of the range of $\mathbf{Y}$ in Step 3 is not required in our case. Second, it is redundant to perform a complete

SVD as in Step 5 to obtain the nullspace. Therefore, we apply the QR factorization based on the Given's rotation method [114] to directly obtain the nullspace of $\mathbf{Y}^{\dagger}\mathbf{A}$. Moreover, we implement Given's rotation based QR by exploiting parallelism. This is done by leveraging the following two characteristics of Given's rotation: i) at each iteration, all the rotated elements (elements of two columns of $\mathbf{Y}$) can be updated simultaneously; ii) at each iteration, only two columns of $\mathbf{Y}$ are dependent. It is worth noting that the mathematical complexity (in $O(\cdot)$) for a complete SVD and Given's rotation based QR may be the same, but in practice the QR method can lead to dramatic acceleration for our problem in real-time. This is because the QR method (with parallel implementation) can save a lot of redundant calculations and memory write/read caused by operations such as column interchanges and computing variables that are not of our interest. The revised algorithm is summarized in Algorithm 4.2.

Algorithm 4.2 for Stage B significantly reduces the computation time of standard SVD operations—the main bottleneck in BD beamforming. The only additional cost is a few more matrix multiplications, which, fortunately, can be parallelized and computed efficiently (more details in Section 4.5). Although randomized SVD is an approximation method, we will not analyze its performance here, since it is only an intermediate step for beamforming. Instead, we will discuss and show both the timing and throughput performance by applying randomized SVD in Section 4.5.

Stage C. In this stage, we construct the digital beamforming matrices $\mathbf{F}_{\mathrm{BB},k}^{b}$. For given matrices $\overline{\mathbf{H}}_{k}^{b}$ and $\widetilde{\mathbf{V}}_{k}^{b(-)}$, the product of $\overline{\mathbf{H}}_{k}^{b}$ and $\widetilde{\mathbf{V}}_{k}^{b(-)}$ effectively forms user k's channel with no (or minor) inter-user interference (recall Eq. (4.4.2)). Therefore, the optimal beamforming strategy regarding the effective $M_{\mathrm{U}} \times (M_{\mathrm{BS}} - r)$ channel $\overline{\mathbf{H}}_{k}^{b}\widetilde{\mathbf{V}}_{k}^{b(-)}$ can be realized based on

Algorithm 4.2: Lightweight Nullspace Computation

Given an $m \times n$ matrix $\mathbf{A}$, a target approximation rank r, and an exponent q (say $q = 1$ or $q = 2$), this procedure computes an approximate last $n - r$ right singular vectors of $\mathbf{A}$, denoted as $\mathbf{V}$:

1 Generate an $n \times r$ Gaussian matrix $\mathbf{\Omega}$.
2 Form the $m \times r$ matrix $\mathbf{Y} = (\mathbf{A}\mathbf{A}^\dagger)^q \mathbf{A}\mathbf{\Omega}$ by multiplying alternately with $\mathbf{A}$ and $\mathbf{A}^\dagger$.
3 Form a $r \times n$ matrix $\mathbf{B} = \mathbf{Y}^\dagger \mathbf{A}$.
4 Compute QR decomposition of $\mathbf{B}^\dagger$ based on Given's rotation: $\mathbf{B}^\dagger = \mathbf{Q}\mathbf{R}$.
5 Set $\mathbf{V}$ as the last $n - r$ columns of $\mathbf{Q}$.

its SVD, which is given by:

$$\overline{\mathbf{H}}_k^b \widetilde{\mathbf{V}}_k^{b(-)} = \overline{\mathbf{U}}_k^b \overline{\mathbf{\Sigma}}_k^b \left[\overline{\mathbf{V}}_k^{b(+)} \quad \overline{\mathbf{V}}_k^{b(-)} \right]^\dagger, \tag{4.5}$$

where $\overline{\mathbf{V}}_k^{b(+)}$ is the first N_s columns of right singular matrix and $\overline{\mathbf{V}}_k^{b(-)}$ is the remaining columns. Finally, the digital beamforming matrix $\mathbf{F}_{\mathrm{BB},k}^b$ is given by

$$\mathbf{F}_{\mathrm{BB},k}^b = \widetilde{\mathbf{V}}_k^{b(-)} \overline{\mathbf{V}}_k^{b(+)}. \tag{4.6}$$

In Eq. (4.5), we encounter another SVD computation. Luckily, the dimension of this to-be-factorized matrix is tied to the number of RF chains at one user, namely M_{U}, which is typically small (e.g., 1 to 4). $\overline{\mathbf{V}}_k^{b(+)}$ can be derived with the help of Hermitian symmetric matrix ED and matrix multiplication, through the following steps:

- Form an $M_{\mathrm{U}} \times (M_{\mathrm{BS}} - r)$ matrix $\mathbf{A}_k^b = \overline{\mathbf{H}}_k^b \widetilde{\mathbf{V}}_k^{b(-)}$;
- Form an $M_{\mathrm{U}} \times M_{\mathrm{U}}$ matrix $\mathbf{B}_k^b = \mathbf{A}_k^b \mathbf{A}_k^{b\dagger}$;
- Compute ED of the Hermitian matrix: $\mathbf{B}_k^b = \mathbf{U}_k^b \mathbf{\Lambda}_k^b \mathbf{U}_k^{b\dagger}$;
- Set $\overline{\mathbf{V}}_k^{b(+)}$ to the first N_s columns of $\mathbf{A}_k^{b\dagger} \mathbf{U}_k^b$.

Note that when $M_{\mathrm{U}} = 1$ or 2, simple and exact closed-form solution for SVD exists [113] and hence this stage can be completed very fast.

4.4.3 Approximation with Lower Rank

As we discussed in Section 4.4.2, Turbo-HB applies lower rank approximation to reduce computational complexity. Interestingly, in most cases, our approximation does not sacrifice throughput performance. In this section, we offer some intuition behind it. Then we address the last problem, which is how to choose a proper value for r.

Let's revisit the SVD of interference channel $\widetilde{\mathbf{H}}_k^b$ as in Eq. (4.3). In the M_{BS}-dimensional signal space $[\widetilde{\mathbf{V}}_k^{b(+)} \quad \widetilde{\mathbf{V}}_k^{b(-)}]$, $\widetilde{\mathbf{V}}_k^{b(-)}$ is an $(M_{\mathrm{BS}} - r)$-dimensional subspace corresponding to the $(M_{\mathrm{BS}} - r)$ smallest interference strengths, while $\widetilde{\mathbf{V}}_k^{b(+)}$ is a r-dimensional subspace corresponding to the r largest interference strengths. When standard SVD is performed, we have $r = (|\mathcal{K}^b| - 1)M_{\mathrm{U}}$ (as in conventional BD approach). Then $\widetilde{\mathbf{V}}_k^{b(-)}$ lies *exactly* in the nullspace of $\widetilde{\mathbf{H}}_k^b$, and therefore *all* inter-user interference will be cancelled when $\mathbf{F}_{\mathrm{BB},k}^b$ is constructed based on $\widetilde{\mathbf{V}}_k^{b(-)}$ (i.e., Eq. (4.6)). In addition to high complexity, there is another drawback of such a "perfect" interference cancellation. That is, to achieve mutual orthogonality, one has to project the desired signal onto a subspace with a small number of dimensions. As a result, the perceived desired signal strength at a user is reduced.

In Fig. 4.4, we use a simple example to illustrate this point. In a 3-dimensional signal space, we have a strong interference $\boldsymbol{f}_1$ along the z axis and a weak interference $\boldsymbol{f}_2$ along the y axis. Now we are going to project a desired signal $\boldsymbol{s}$ (originally in the xyz space) onto some subspace to avoid interference (via beamforming). If perfect interference cancellation is required, then $\boldsymbol{s}$ has to be projected along the x axis to achieve orthogonality to both $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$, leading to a smaller-strength signal $\boldsymbol{s}'$, as shown in Fig. 4.4(a). However, if only

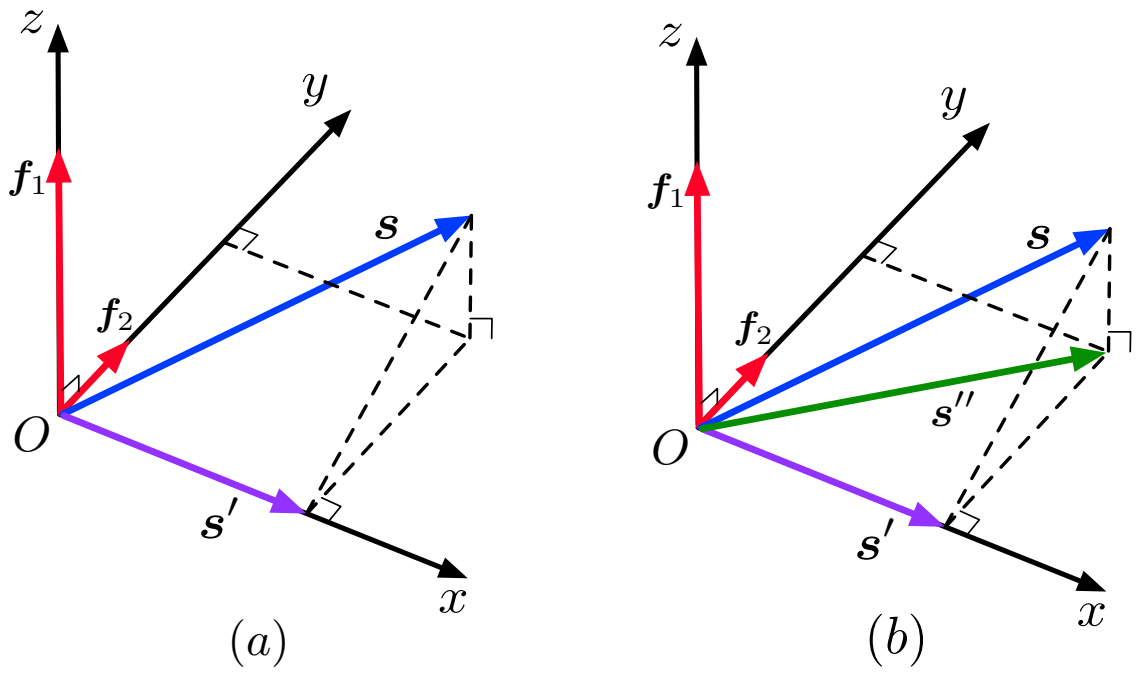


Figure 4.4: Signal is projected onto a lower dimensional subspace to avoid interference. (a) Signal $\boldsymbol{s}$ is projected along the x axis, resulting in $\boldsymbol{s}'$; (b) (a) Signal $\boldsymbol{s}$ is projected onto the xOy plane, resulting in $\boldsymbol{s}''$.

the strong interference $\boldsymbol{f}_1$ is required to be cancelled, then one can project $\boldsymbol{s}$ into a larger dimensional subspace, i.e., xOy plane, resulting in $\boldsymbol{s}''$ as shown in Fig. 4.4(b). Although $\boldsymbol{s}''$ is interfered with by a weak interference $\boldsymbol{f}_2$, $\boldsymbol{s}''$ can preserve higher signal strength than $\boldsymbol{s}'$, which will lead to a higher SINR (and throughput).

Turbo-HB is purposefully designed to explore such a design space by tolerating some level of weak interference. When lower rank SVD approximation is performed, it is meant to only identify r directions corresponding to r strongest interference. Without knowledge of how remaining interference presents, the desired signal will be projected onto a larger dimensional subspace only to avoid the identified interference, preserving greater desired signal strength. This approach is especially effective for scenarios where there is high correlation among the channels or SNR is low. Since in these scenarios, the last few singular values (i.e., corresponding weak interference strengths) are small compared to the power of white noise.

Then the dominant term in the denominator of SINR becomes the power of noise, which cannot be suppressed by interference cancellation. Thus, by tolerating weak interference, desired signal strength is preserved to overcome a bigger issue (the noise), leading to a higher SINR.

Now we address the question of how to choose a proper value for r. Since $0 < r \leq \text{rank}(\widetilde{\mathbf{H}}_k^b) = M_{\text{BS}} - M_{\text{U}}$ and r is an integer, we have $(M_{\text{BS}} - M_{\text{U}})$ possible values for r. If we choose r to be too large (i.e., close to $(M_{\text{BS}} - M_{\text{U}})$), then we will have to get into high-dimensional SVD operations, which are what we try to avoid. On the other hand, if we choose r to be too small, then we may experience serious sacrifice in throughput performance. So the goal is to find an optimal r that offers the best trade-off. Unfortunately, finding the optimal value of r (in terms of maximizing network throughput) is intractable, due to the large search space and non-convex objective function.

To gain some insight into what value of r should be, we conduct the following experiment. We randomly generate 1,000 channel instances under different settings following the mmWave channel model. For each instance, we enumerate all possible r's and calculate its corresponding throughput C. For the time being, we focus only on the objective function (throughput) and defer consideration of computation time till later. In the experiment, we set $N_s = 2$ and use the same settings as those used in Section 4.4.2 except that we now vary SNR values and the number of channel paths. Specifically, let's consider a low SNR scenario (5 dB) and a high SNR scenario (20 dB), each of which is in combination with a small number of clusters and rays ($L_{\text{cl}} = L_{\text{ray}} = 3$) or a large number of clusters and rays ($L_{\text{cl}} = L_{\text{ray}} = 10$). Fig. 4.5 shows the achieved network throughput as a function of approximation rank r under these four scenarios. Note that when $r = 16$, the achieved throughput value (the first blue bar in each figure) is what is achieved by standard SVD (as in traditional BD method). For the first three scenarios, where the channels are experiencing at least low SNR or high

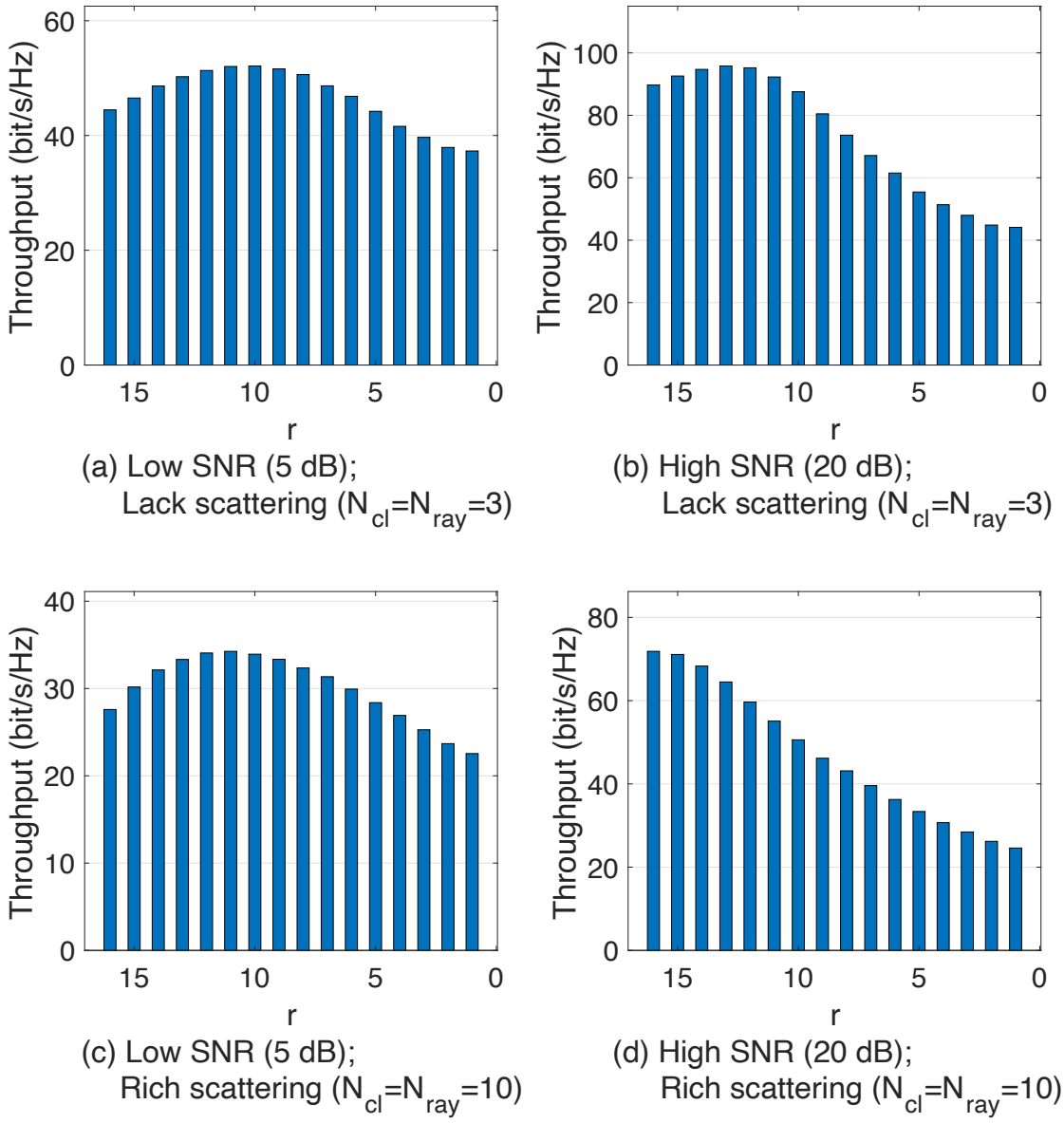


Figure 4.5: Achieved network throughput (averaged over 1,000 instances) as a function of approximation rank r under different SNR and number of channel paths.

correlations, we observe that the throughput goes up at first and then goes down as the value of r decreases. Only when the channels maintain both high SNR and low correlations (as in scenario (d)), the network throughput would strictly decline as the value of r decreases. However, scenario (d) is relatively rare for mmWave systems. This experiment suggests that the lower rank r indeed offers the opportunity for higher throughput, especially at low SNR or high correlation scenarios. Under this scenarios (a), (b) and (c), setting $r = \frac{M_{\text{BS}}}{2} = 10$ would offer better (or comparable) performance than that with $r = 16$ in most instances.

The results in Fig. 4.5 are averaged over 1,000 channel instances. However, our interest is on a particular channel instance, and the optimal choice of r based on averaging over 1,000 channel instances may not perform well in this particular instance. Therefore, we propose to employ multiple choices of promising r's in parallel and derive multiple beamforming candidates corresponding to these r's. That is, we execute several different lower rank approximations simultaneously, where the set of target rank is given by

$$\mathcal{R} = \{r - \delta, \cdots, r-1, r, r+1, \cdots, r+\delta\}, \tag{4.7}$$

where r is around $\frac{M_{\text{BS}}}{2}$ (which may be adjusted according to empirical statistics), and δ is a parameter to control the number of elements in $\mathcal{R}$. As $|\mathcal{R}|$ different lower rank approximations are implemented, we will have $|\mathcal{R}|$ different solutions of $\mathbf{F}_{\text{BB},k}^{b}$ for each user on each RB after Stage C. Among these $|\mathcal{R}|$ solutions, we evaluate their throughput performance (i.e., C in Eq. (4.2)) and choose the one that offers the largest objective value as the final beamforming matrix.

4.5 Implementation

In this section, we present the implementation of our design in Section 4.4. Our implementation is done on Nvidia DGX station—a COTS GPU platform. Our Nvidia DGX Station consists of 4 V100 GPU cards but we use only two of them. Each V100 card includes 80 streaming multiprocessors (SMs), and each SM has 64 CUDA cores. The CPU of our DGX station is Intel Xeon E5-2698 v4 2.2 GHz (20-core). The data communication between CPU and GPU is based on a PCIe V3.0 architecture [115]. CUDA programming tool (version 10.2) [117] is used to realize our algorithm and schedule the memory and processing cores.

For a successful implementation of Turbo-HB, we must have a thorough knowledge of the capability and limitation of the GPU and find a way to fit our problem optimally into the platform. In general, the more parallelism and less overhead in the implementation, the better the performance we can achieve. As such, we focus on the following two objectives in our implementation:

1. fully utilize GPU processing cores,

2. minimize memory access time.

In the rest of this section, we present the details of our implementation based on the above two objectives.

4.5.1 Workflow on GPU

The key to fully utilize GPU processing cores is to have a sufficient large amount of parallel workloads in flight to feed all the GPU cores. By our design in Section 4.4, the computations for beamforming matrice are independent among different RBs, different users, and different

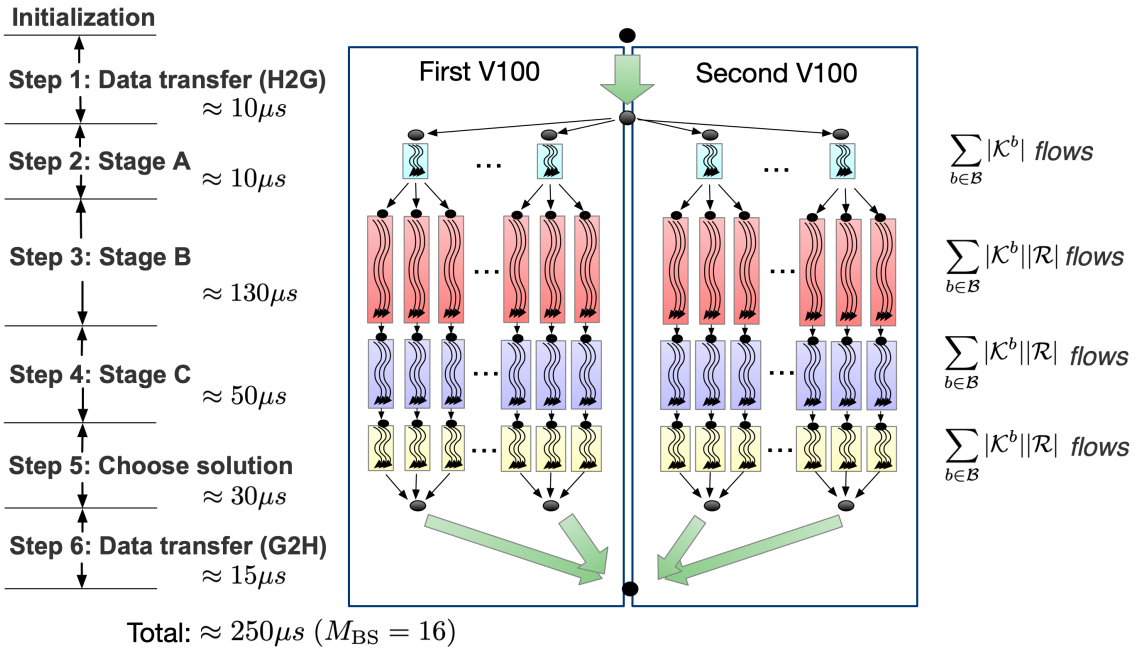


Figure 4.6: Workflow of implementing Turbo-HB on GPUs.

target ranks. Thus, we can spread out the computation tasks over all available processing cores. At each step in the implementation, the computation tasks are broken into a number of parallel processing flows. Each flow is a group of parallel threads that executes certain operations. All the flows shall be mutually data-independent and have the same computation procedures to take advantage of GPU's SIMD architecture. Based on the architecture of our GPU V100, every consecutive 32 parallel threads are assembled into a group called a *warp* for executing exactly the same instructions (while handling different data). Therefore, it is preferable that a flow consists of an integral multiple of (or close to an integral multiple of) 32 threads. As V100 has 80 SMs, the number of flows should be at least 80 to avoid idle SMs.

As illustrated in Fig. 4.6, our implementation includes the following key steps.

Step 0: Initialization. The system first sets up global parameters, including the number of RBs $|\mathcal{B}|$, user sets $\mathcal{K}^b$ on RB b, and the number of RF chains at BS M_{BS} and at

users M_{U}, etc. Then we calculate and allocate the memory space needed on GPU for storing the matrices and variables.

Step 1: Set up global parameters and transfer the compressed CSI from host to GPU. At the beginning of each time slot, the host transfers $\sum_{b\in\mathcal{B}}|\mathcal{K}^b|$ partial CSI (i.e., $\mathbf{V}_k^b$'s and $\mathbf{\Sigma}_k^b$'s) from host memory to GPU global memory (also known as device global memory). Since we use two V100 GPU cards, we divide the channel matrices into two halves. The first half corresponds to the first $\frac{|\mathcal{B}|}{2}$ RBs and is transferred to the first GPU card. The second half will be handled by the second GPU card.

Step 2: Execute Stage A. The objective of this step is to generate $\widetilde{\mathbf{H}}_k^b$ for every $k\in\mathcal{K}^b$ and $b\in\mathcal{B}$ on GPU. We generate a total number of $\sum_{b\in\mathcal{B}}|\mathcal{K}^b|$ parallel flows, where each flow corresponds to the beamforming matrix of one user on an RB. We program one thread to calculate one element of $\widetilde{\mathbf{H}}_k^b$, thus a total number of $\sum_{b\in\mathcal{B}}|\mathcal{K}^b|\cdot N_{\mathrm{thread}}$ threads are spawned in this step, where $N_{\mathrm{thread}}=(|\mathcal{K}^b|-1)M_{\mathrm{U}}\times M_{\mathrm{BS}}$ is the total number of elements in $\widetilde{\mathbf{H}}_k^b$.

Step 3: Execute Stage B. As we discussed in Section 4.4.2, the main task of this stage is to compute the approximate nullsapce of $\widetilde{\mathbf{H}}_k^b$. We generate a total number of $\sum_{b\in\mathcal{B}}|\mathcal{K}^b||\mathcal{R}|$ parallel flows, where each flow corresponds to the computation for one user on one RB with one target rank. Each flow executes Algorithm 4.2 to derive matrix $\widetilde{\mathbf{V}}_k^{b(-)}$. In particular, for the QR decomposition in Step 4 of Algorithm 4.2, we use Given's rotation method. The computation requires multiple iterations and each iteration would overwrite the processing matrix. To reduce the memory access time for repeated accesses, we first transfer the input matrix from GPU's global memory to the fast on-chip shared memory, then the iterative computations are performed based on shared memory access. The output matrix is transferred back to global memory after QR decomposition is completed. Step 3 is the most computation intensive step in our implementation. It consumes around 130 μs (for

$M_{\mathrm{BS}} = 16$, $M_{\mathrm{U}} = 2$ and $|\mathcal{K}^b| = 8$) after our optimization.

Step 4: Execute Stage C. In this step, $\sum_{b\in\mathcal{B}} |\mathcal{K}^b||\mathcal{R}|$ parallel flows are generated to calculate $\mathbf{F}^b_{\mathrm{BB},k}$. This step includes a small dimensional SVD operation. Note that when $M_{\mathrm{U}} = 1$ or 2, simple closed-form expressions can be directly applied for SVD computation.

Step 5: Choose the best solution. After Step 4, we obtain $|\mathcal{R}|$ beamforming candidates for each user on each RB. In $\sum_{b\in\mathcal{B}} |\mathcal{K}^b||\mathcal{R}|$ parallel flows, we evaluate their throughput performance as in Eq. (4.2) for every beamforming candidate. The best $\mathbf{F}^b_{\mathrm{BB},k}$ that provides the highest objective value C in Eq. (4.2) will be chosen as the final solution. To speed up comparison, parallel reduction technique [116] is employed.

Step 6: Transfer beamforming solution from GPU to host. Once Step 5 is accomplished, the final beamforming solution (i.e., $\mathbf{F}^b_{\mathrm{BB},k}$ for every $k \in \mathcal{K}^b$ and $b \in \mathcal{B}$) is transferred from GPU memory to the host memory.

4.5.2 Speed-Up Techniques

Now we discuss two specific techniques that we have employed in Turbo-HB to enhance parallelism and reduce memory access time.

Batching Batched matrix operations are critical to our problem, as we have to execute a large number of independent matrix operations following the same procedure. As an example, suppose we need to execute hundreds or even thousands of matrix multiplications simultaneously. The programmer needs to generate a kernel with a sufficient number of threads and divide these threads into a number of groups. Then each group computes one or a few matrix multiplications, such that this kernel is able to perform batched matrix multiplications. Similarly, other matrix operations (following the same procedure), such as a large number of independent matrix ED operations, should be programmed in a batched

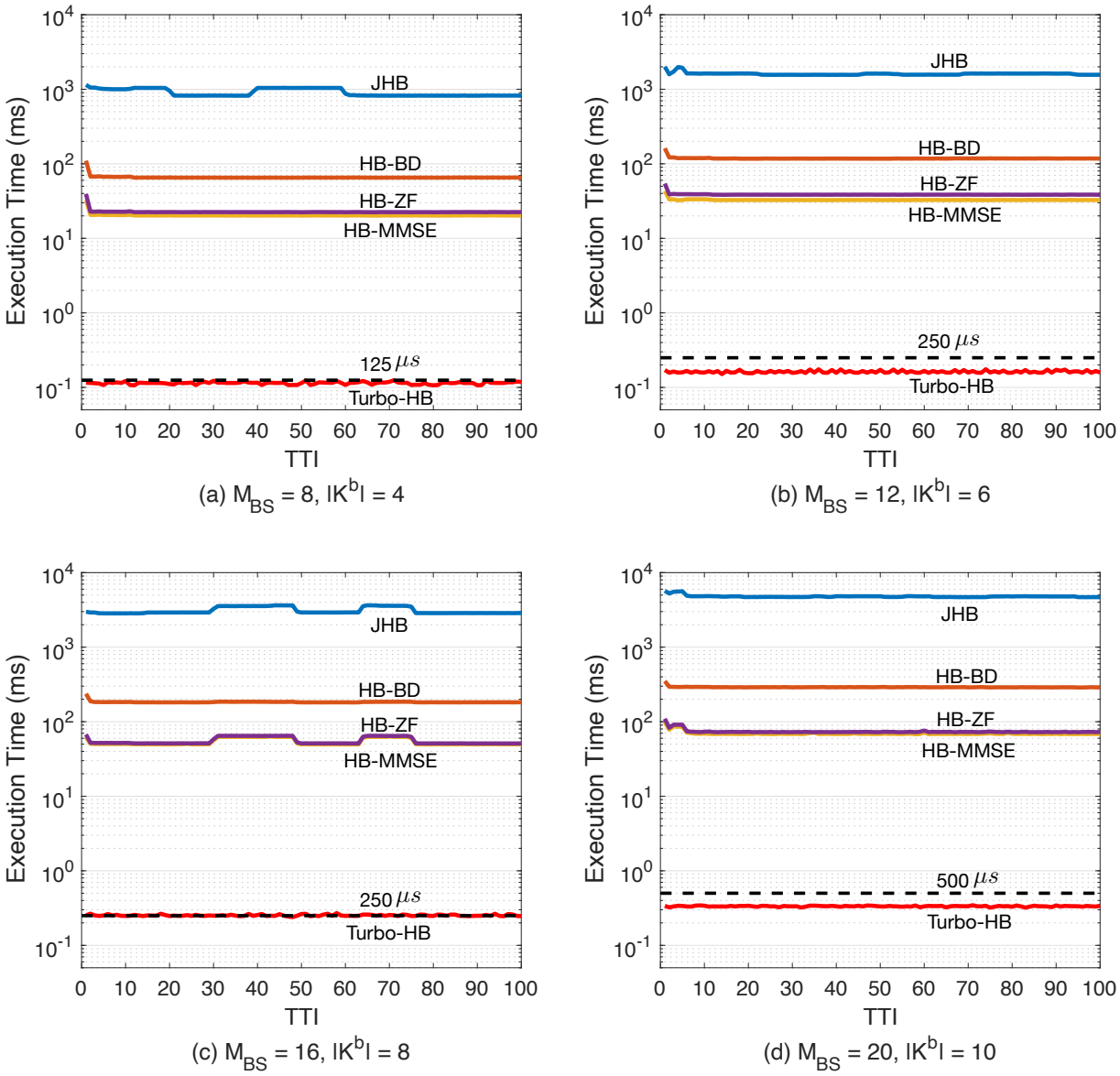


Figure 4.7: Comparison of execution time of different schemes under different MU-MIMO scenarios.

manner to fully occupy the processing cores.

Minimizing global memory access Compared to other types of memory access, accessing global memory is much more time-consuming. We identify two techniques that can help minimize global memory access in our problem.

First, the programmer should carefully coalesce memory access, i.e., consolidating multiple memory accesses into a single transaction. This is particularly important when we handle a large number of matrix operations. The key to memory coalescing is to store the matrices consecutively in the memory with proper indexing. Then the programmer can allow consecutive threads to read consecutive (and aligned) memory and minimize the number of transactions.

Second, instead of global memory accesses, which is more time-consuming, we can use on-chip shared memory accesses, which is much faster (but with limited storage space). Suppose we want to compute a matrix multiplication $\mathbf{C}_{m\times n} = \mathbf{A}_{m\times l}\mathbf{B}_{l\times n}$. A straightforward approach for parallelism is to program each thread to take care of one element of $\mathbf{C}$. Then we need to read $\mathbf{A}$ n times from the global memory and $\mathbf{B}$ m times. In contrast, if matrix multiplication is based on shared memory [117], we only need to read $\mathbf{A}$ for (n / block size) times from the global memory and $\mathbf{B}$ for (m / block size) times. The remaining computations are done by accessing the shared memory.

4.6 Experimental Validation

In this section, we present our experimental results, with a focus on timing and throughput performance. We also compare with other state-of-art sequential HB schemes. For analog beamforming part, we apply the widely adopted DFT-codebook based method [86, 111] for

all schemes. For digital beamforming schemes, we choose HB-BD [86], HB-MMSE and HB-ZF for comparison. We also include one joint analog and digital HB method (JHB) [83] to show its timing performance.

Experiment Setup We consider a cellular communication scenario with one BS and a number of users. The number of available RBs is up to 100. The BS is equipped with 128 antennas and each user is equipped with 16 antennas (a typical number for hybrid architecture at mmWave frequencies [80, 82, 86]). The number of RF chains at the BS varies from 8 to 20, while the number of RF chains at a user is 2. Each active link is assumed to transmit $N_s = 2$ data streams. The number of active users for MU-MIMO transmission on each RB (i.e., $|\mathcal{K}^b|$) varies in this study. For the wireless channels, we use the widely considered cluster-based mmWave channel model [82]. The number of clusters L_{cl}, the number of propagation paths L_{ray} caused by each cluster and SNR (i.e., $\frac{P_{\text{T}}}{\sigma^2}$) will be given under different settings. The angle spread σ_{AS} is set to 5 degrees. We set parameter δ (as defined in Section 4.4.3) to 2.

Timing Performance We first verify that Turbo-HB can indeed offer the beamforming solution in less than 1 ms for all settings in our experiments and even achieves as little as 125 μs execution time in some settings. Note that the time consumed for data transfer between CPU and GPU is included in Turbo-HB's total execution time.

We first run the experiments for 100 consecutive TTIs under different settings as following: (a) $M_{\text{BS}} = 8$, $|\mathcal{K}^b| = 4$, (b) $M_{\text{BS}} = 12$, $|\mathcal{K}^b| = 6$, (c) $M_{\text{BS}} = 16$, $|\mathcal{K}^b| = 8$ and (d) $M_{\text{BS}} = 20$, $|\mathcal{K}^b| = 10$. For the sequential algorithms (Turbo-HB, HB-BD, HB-MMSE and HB-ZF), we only count the computation time of digital beamforming part. But for the joint algorithm (JHB), we have to count time consumed both for its digital beamforming and analog beamforming since they are inseparable. Our GPU-based algorithm is run on CUDA platform while others are run on Matlab platform. Fig. 4.7 shows the results of execution

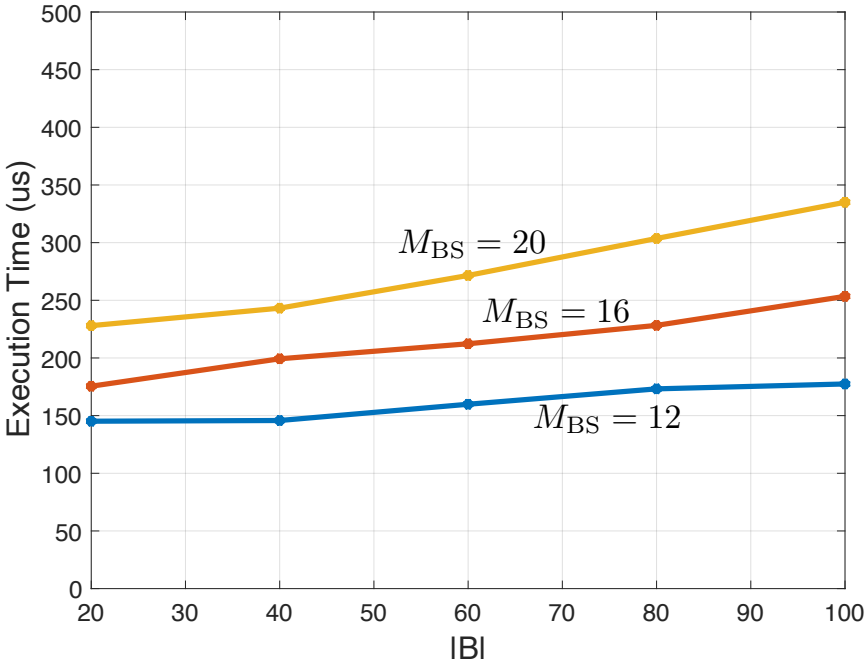


Figure 4.8: Average execution time of Turbo-HB vs. the number of available RBs $|\mathcal{B}|$ under different M_{BS} settings.

time by different schemes. JHB, HB-BD, HB-MMSE and HB-ZF require a computation time on the order of 10^3 ms, 10^2 ms, 10^1 ms and 10^1 ms, respectively. Our experiments show that Turbo-HB finds beamforming solution in 114 μs, 162 μs, 250 μs, and 335 μs averaged by 100 TTIs under settings (a), (b), (c) and (d), respectively. Based on the numerologies defined in 5G NR, Turbo-HB can meet the timing requirement for numerology 3 (125 μs TTI), numerology 2 (250 μs TTI) and numerology 1 (500 μs TTI) for 100 RBs with up to 4, 8, and 10 MU-MIMO users on each RB, respectively.

Next, we conduct experiments to examine Turbo-HB's total execution time under different numbers of available RBs $|\mathcal{B}|$. We consider the following settings: (a) $M_{\mathrm{BS}} = 12, |\mathcal{K}^b| = 6$, (b) $M_{\mathrm{BS}} = 16, |\mathcal{K}^b| = 8$. and (c) $M_{\mathrm{BS}} = 20, |\mathcal{K}^b| = 10$. Fig. 4.8 shows Turbo-HB's execution time performance (with value for each point being average over 100 TTIs) for the three settings. Note that the execution time increases slowly (and close to linear) as the number

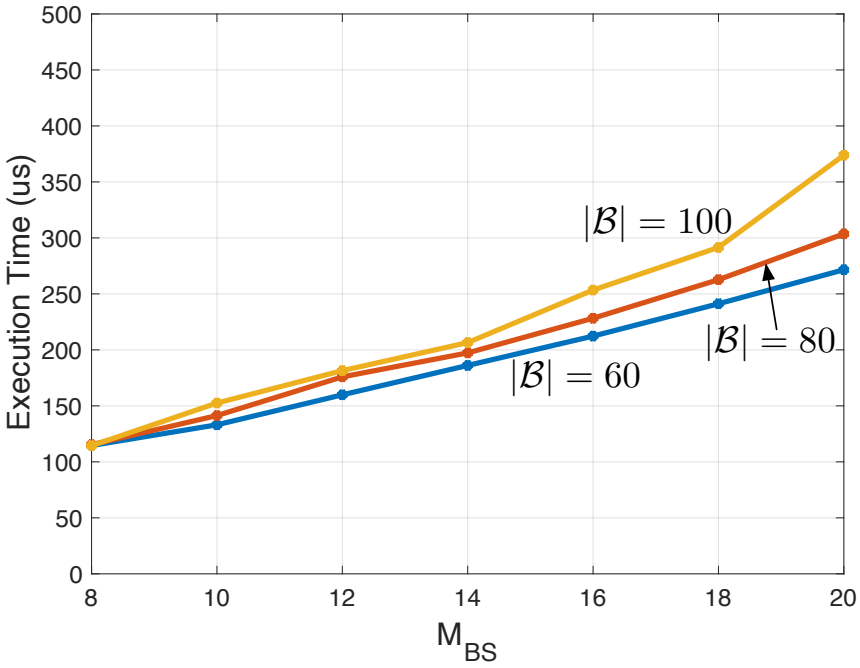


Figure 4.9: Average execution time of Turbo-HB vs. the number of RF chains M_{BS} at the BS under different settings of available RBs $|\mathcal{B}|$.

of RBs increases. This is because under Turbo-HB, computation among different RBs is executed in parallel and is not very sensitive to the number of RBs. For a given M_{BS}, the network operator can set the upper bound for the number of RBs to meet a certain 5G numerology. For example, when $M_{\text{BS}} = 16$, if the number of RBs is no more than 95 (a large number), we can meet 5G numerology 2 requirement (250 μs).

In Fig. 4.9, we vary the number of RF chains at the BS (i.e., M_{BS}) to show its impact on Turbo-HB's execution time. For this study, we consider the settings of $|\mathcal{B}| \in \{60, 80, 100\}$ and $|\mathcal{K}^b| = \frac{M_{\text{BS}}}{N_s}$, while M_{BS} varies from 8 to 20. As expected, the results in Fig. 4.9 show that Turbo-HB's average execution time is increasing with M_{BS}. Compared with varying $|\mathcal{B}|$, Turbo-HB is more sensitive to the change of M_{BS}. This is because the larger the M_{BS}, the higher dimensional matrix operations will be required, which leads to more computation time. However, Turbo-HB is able to complete the computation in real-time, thanks to its

design based on randomized SVD.

Throughput Performance We first evaluate throughput performance achieved by different schemes under varying SNR value. We consider two different settings: (a) $M_{\text{BS}} = 10$, $|\mathcal{K}^b| = 4$, and (b) $M_{\text{BS}} = 20$, $|\mathcal{K}^b| = 8$. We set $N_{\text{cl}} = N_{\text{ray}} = 3$ and SNR varies from -5 dB to 25 dB in both cases. Fig. 4.10 shows that in both cases, throughput under conventional HB-MMSE and HB-ZF methods are below the others, as MMSE and ZF are not designed for mmWave systems and the poorly conditioned channel greatly degrades MMSE/ZF's performance [88, 92, 93]. In Fig. 4.10, Turbo-HB is able to achieve similar performance as the classical HB-BD and is better than the others.

Next, we vary the channel correlation condition (by varying the number of propagation clusters L_{cl}) and study its impact on throughput performance. We fix SNR = 20 dB, $M_{\text{BS}} = 20$, $|\mathcal{K}^b| = 8$, and $N_{\text{ray}} = 3$. We vary L_{cl} from 1 to 7. Fig. 4.11 shows the throughput achieved by different schemes as a function of N_{cl}. The results show that the performance by HB-MMSE and HB-ZF is significantly lower than the others, especially when the number of clusters is small (and thus the channels are highly correlated). On the other hand, Turbo-HB is able to achieve similar performance as HB-BD and offers high throughput than HB-MMSE and HB-ZF. This is because both Turbo-HB and HB-BD are SVD-based and are capable of identifying the best signal directions for beamforming. When the number of channel paths is small, Turbo-HB is able to obtain even better performance than HB-BD. The reason behind this was given in our discussions in Section 4.4.3.

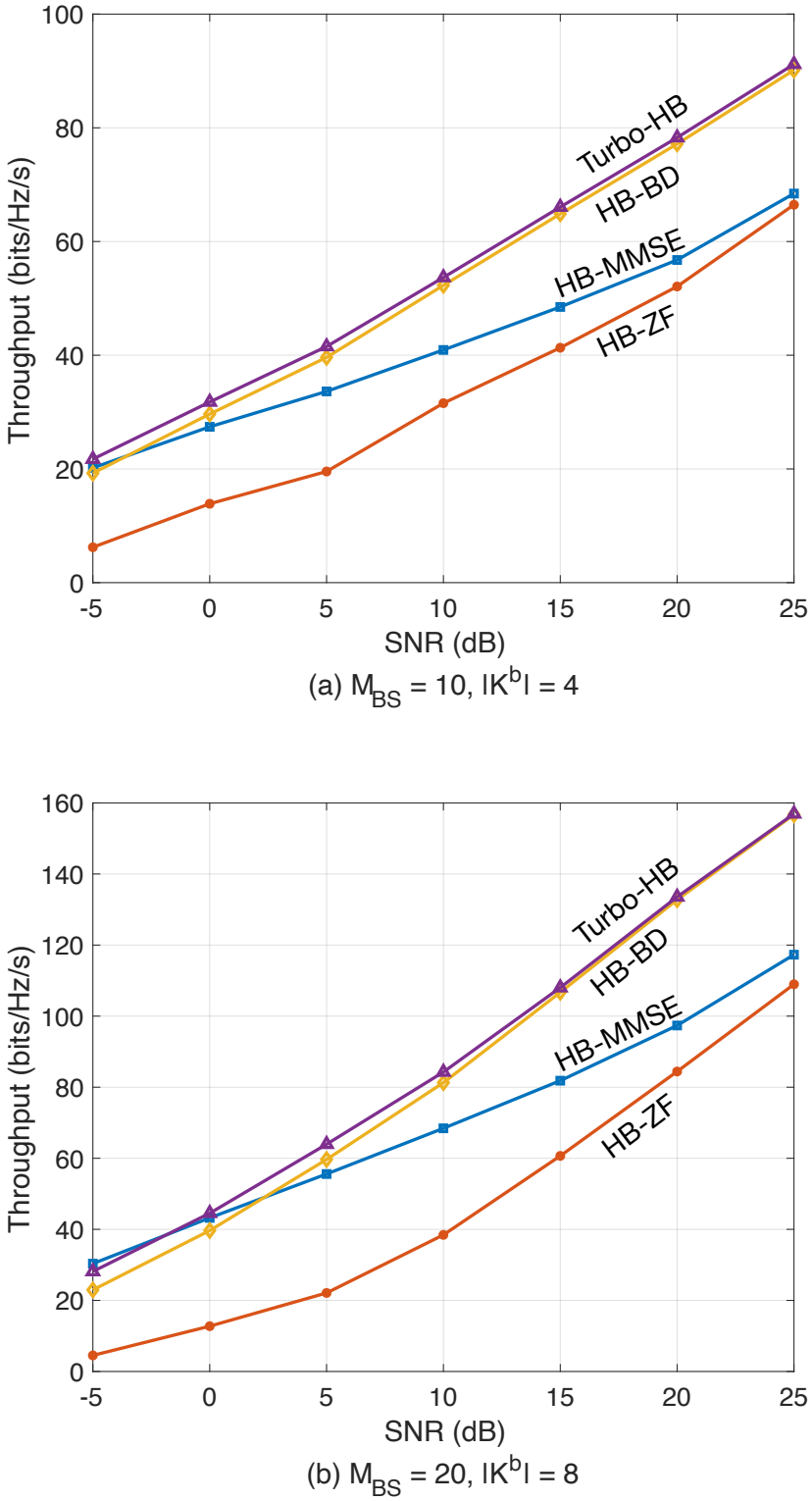


Figure 4.10: Comparison of throughput achieved by different schemes as a function of SNR under different MU-MIMO scenarios.

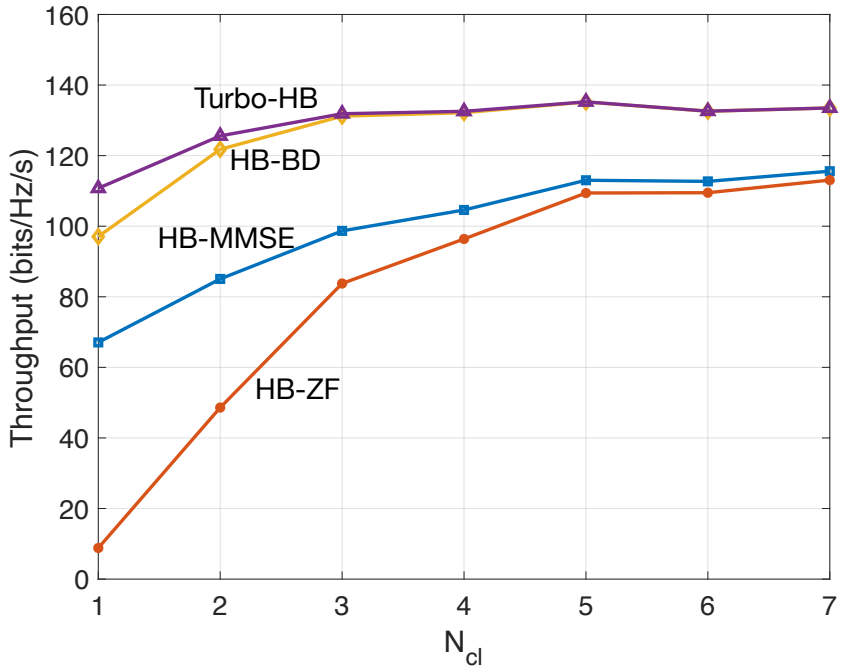


Figure 4.11: Comparison of the throughput achieved by different schemes under different channel propagation conditions.

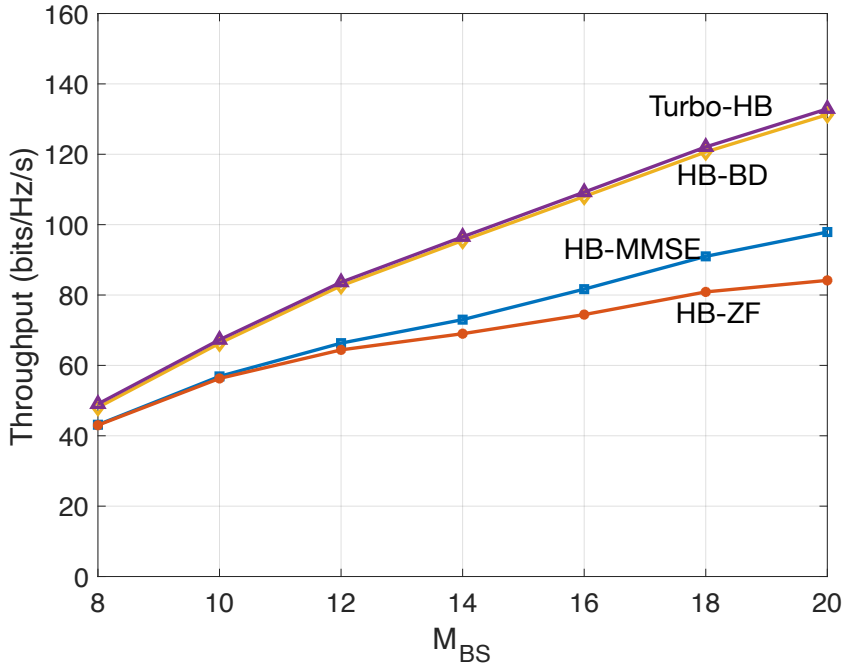


Figure 4.12: Comparison of throughput achieved by different schemes under different number of RF chains at the BS.

Finally, we present throughput performance under different numbers of RF chains M_{BS} at the BS. We consider the setting of SNR = 20 dB and $N_{\text{cl}} = N_{\text{ray}} = 3$. M_{BS} is chosen from $\{8, 10, 12, 14, 16, 18, 20\}$, and $|\mathcal{K}^b|$ is chosen from $\{2, 3, 4, 5, 6, 7, 8\}$ accordingly. In Fig. 4.12, the results show that the network throughput is increasing with M_{BS} for all schemes as more users can be supported. The performance gap between Turbo-HB/HB-BD and HB-MMSE/HB-ZF is also increasing with M_{BS} as the SVD-based approaches can better reap the benefits provided by additional RF chains. Again, we find that Turbo-HB can offer similar performance as HB-BD and outperforms other schemes.

Summary of Results The experimental results show that Turbo-HB can meet the 1-ms real-time requirement under all tested settings and can meet the 5G requirement with appropriate numerology. On the other hand, all other schemes incur a computation time that is of orders of magnitude higher than Turbo-HB and none of them can meet the 5G timing requirement. Further, Turbo-HB is able to offer a throughput performance that is better or comparable to the state-of-the-art algorithms.

4.7 Related Work

Hybrid beamforming design is an active research area and has attracted lots of research efforts. However, most existing research has been largely limited to asymptotic complexity analysis (i.e., in $O(\cdot)$). Although such complexity analysis is of interest from theoretical perspective, it does not give any indication on how much actual time ("real-time") is needed when it is implemented on a given hardware platform. On the other hand, for a real-world 5G system, the ultimate benchmark is real-time performance, as there is a stringent timing requirement under its numerology.

In the literature, all kinds of HB designs involve some level of heuristics. One line of

research is to jointly optimize analog and digital beamforming to offer a near-optimal solution (see, e.g., [81, 82, 83, 84]). A common feature of these designs is that their algorithms must run iteratively to update digital beamformers and analog beamformers. Due to a large number of iterations that are needed in these designs, none of them can offer real-time solutions under 5G requirement (sub-ms).

On the other hand, sequential designs are proposed to reduce the complexity by decoupling the analog domain and digital domain (see, e.g., [86, 87, 88, 89, 90]). However, the mainstream of existing research works heavily relies on reducing the asymptotic complexity (in $O(\cdot)$) in their algorithms. Since asymptotic complexity analysis of an algorithm is only concerned with when the input size n is sufficiently large (approaches to infinity), it does not reflect how much actual time it will need when input data is finite, as in 5G. As a result, these sequential algorithms do not meet the sub-ms timing requirement when they are tested by a real timer. In addition, algorithms designed with extremely simple digital beamforming such as ZF/MMSE may also suffer from considerable throughput loss at mmWave frequencies.

Recently, there has been a number of successful research works applying parallel techniques to wireless networking and signal processing problems. Some representative works include [95, 118, 119, 120, 121, 122]. Specifically, the authors in [95, 118, 119] implemented real-time designs to address scheduling problems in 4G/5G networks. In [120], the authors proposed MIMO detection algorithms that utilize parallelism to achieve high-performance detection. The study in [121, 122] applied parallel processing to accelerate LDPC decoding. These approaches were demonstrated on a GPU or FPGA platform. Among them, the designs based on general-purpose GPU platform (e.g., those from [95, 118, 121]) provided high level of parallelism and flexibility, thanks to GPU's large-scale SIMD architecture and highly programmable tools such as CUDA. However, these algorithms are designed to address scheduling or decoding problems, and their approaches cannot be applied in solving a

complex beamforming problem under hybrid architecture, which is the focus of this chapter.

4.8 Chapter Summary

This chapter presents Turbo-HB, the first design and implementation that addresses the real-time challenge for beamforming under HB architecture. To reduce computation time, Turbo-HB exploits randomized SVD technique by leveraging channel sparsity at mmWave frequencies. Further, Turbo-HB exploits large-scale parallel processing, with optimized matrix operations and minimized memory accesses. We implemented Turbo-HB on COTS Nvidia DGX Station with CUDA programming platform. Through extensive experimental studies, we found that Turbo-HB is able to find beamforming matrices successfully under 1 ms for all tested cases. Specifically, Turbo-HB can meet the 125μs (numerology 3), 250 μs (numerology 2) and 500 μs (numerology 1) timing requirements for 100 RBs with up to 4, 8, and 10 MU-MIMO users on each RB, respectively. In the meanwhile, Turbo-HB offers competitive throughput performance compared to the state-of-the-art algorithms.

Chapter 5

A Sub-millisecond Scheduler for 5G MU-MIMO Systems

5.1 Introduction

In 5G NR, MU-MIMO is one of the most powerful technologies to increase network throughput [123, 124, 125, 126, 127]. Under MU-MIMO, a base station (BS) is able to transmit signals to multiple users simultaneously on the same frequency band. Compared to traditional cellular networks such as 4G LTE, where BSs are typically equipped with a few number of antennas (e.g., < 8), BSs for 5G NR are likely to have a larger number of antennas (e.g., 12 antennas). Therefore, 5G NR can accommodate much more MU-MIMO users by exploiting the spatial diversity offered by many antennas. Per 5G specification[128], up to 12 data streams can be co-scheduled on the same RB. That is 12 MU-MIMO users when each user has one data stream. For a particular user, it may have up to 8 streams concurrently. In contrast, it is typical that only a small number of users and streams (e.g, 2-stream SU-MIMO or 2-user MU-MIMO) is considered under 4G LTE [129, 130].

However, a number of technical challenges in the design of a 5G scheduler need to be addressed before we can fully reap the fruit of MU-MIMO. Specifically, for a downlink scheduling problem at a 5G BS (see Fig. 5.1), we have the following challenges.

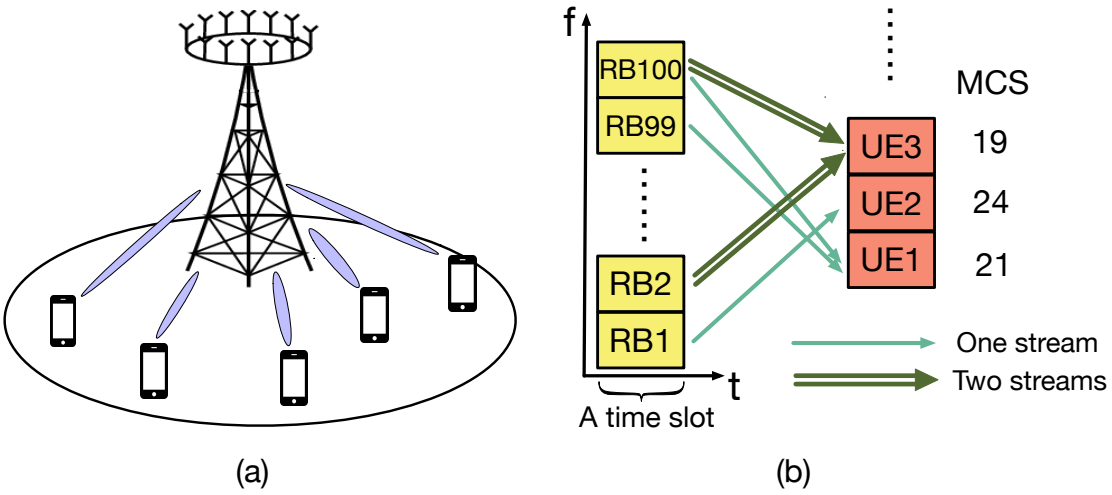


Figure 5.1: System model. (a) A 5G MU-MIMO BS serving a number of users. (b) Within each time slot, the BS determines RB allocation, number of data streams, and MCS assignment for each user.

- First, the scheduler should allocate a number of RBs to cell users for data transmission. Under MU-MIMO, one RB can be allocated to multiple users. These users can decode their desired signals by applying beamforming techniques. Note that a user's achieved SINR (and data rate) depends on the set of users that are co-scheduled with this user on the same RB.

- Second, the scheduler needs to determine the number of data streams transmitted from the BS to each user, exploiting the best tradeoff between diversity and spatial multiplexing offered by MIMO channels. One practical constraint in 5G NR is that the number of data streams used for a user must be identical across all RBs allocated to this user [128].

- Third, the scheduler needs to choose the modulation and coding scheme (MCS) for each user. Similar to the number of data streams, if a user is scheduled to receive data on multiple RBs, then the user must use the same MCS across all RBs scheduled to

her [4].[1]

Therefore, the scheduling problem tightly couples together RB allocation, stream number determination, and MCS selection. This makes the MU-MIMO scheduling problem NP-hard [131, 132, 133], with an extremely large solution space.

Further, we face a stringent real-time requirement for the 5G MU-MIMO scheduler. That is, the scheduler must be able to offer the scheduling solution within each TTI to be useful. In 5G NR, the frame structure is scalable to support a variety of 5G applications. Under 5G numerology 0, 1 TTI is 1 ms. To support the applications with higher latency requirement, the numerology 1 with 500 μs TTI may be considered. In this paper, we aim at offering the scheduling solution (including RB allocation, data stream number determination and MCS assignment) in $\sim$500 μs to meet the timing requirement of 5G numerology 1.

Designing a scheduler for cellular systems has been explored in the literature. However, none of the existing research can offer an MU-MIMO scheduler that meets the 5G real-time requirement. Some representative works include [95, 131, 132, 133, 134, 135, 136, 137, 138, 139]. The designs in [131, 132, 133, 134, 135, 136, 137, 138, 139] have one common feature—their algorithms must run a large number of iterations. Due to this iterative nature, none of these designs can offer a scheduling solution in real-time ($\sim$500 μs). Further, there is hardly any research that can jointly optimize the scheduling of RBs and MCS for MU-MIMO users. For instance, the authors in [95] implemented a real-time 5G scheduler, but their design only considered single-antenna deployment. Likewise, the designs in [133, 134, 136, 137] are also only applicable to non-MIMO settings. The works in [131, 132, 135, 138, 139] considered MIMO deployment to design schedulers. However, either MU-MIMO scheduling or MCS assignment is missing in their models. It is also fair to presume that extending these

[1]We consider a typical configuration of applying one transport block at each user. When a user is configured with two transport blocks, the data belonging to two different transport blocks may have two different MCSs.

algorithms to support the joint scheduling of RBs, MCS for MU-MIMO users would result in even longer computation time.

In this paper, we present a novel design and implementation for 5G MU-MIMO systems that can achieve ∼500 μs scheduling. We call our design "mCore+", which is our acronym for sub-millisecond scheduler with GPU cores, and the "+" denotes an improved version of our original design (mCore) in [119]. The success of mCore+ is built upon recent advances based on the general-purpose GPU-based platform (see, e.g., [95, 96, 120, 145, 146, 147]). Thanks to the massive computing cores offered by a GPU with the dedicated single-instruction-multiple-data (SIMD) architecture, GPU is capable of solving a large number of structurally-identical sub-problems at an extremely fast speed. A GPU-based platform offers a new possibility to solve complex optimization problems with stringent real-time requirements. In our design mCore+, we exploit GPU's parallel computing capability through a multi-phase optimization. At each phase, mCore+ either decomposes each problem into a large number of independent sub-problems to utilize large-scale parallelism, or selects the most promising search space leveraging channel conditions and user correlations, or performs both. mCore+ is implemented on a commercial off-the-shelf (COTS) GPU platform. Special engineering efforts are made to fit our problem into two GPU cards, such as minimizing the data transfer and synchronizations between GPU cards. The main contributions of mCore+ are summarized as follows:

- This paper presents the first design and implementation of a 5G MU-MIMO scheduler that can offer a scheduling solution, as well as corresponding beamforming matrices, in ∼500 μs. This design supports MU-MIMO transmission, allowing multiple users to share the same time-frequency resources, and each user may have multiple data streams concurrently. mCore+ can be applied to 5G NR numerology 0 and 1.

- Our design exploits large-scale parallelism through a dedicated multi-phase optimization design. Specifically, at each phase, mCore+ either decomposes the optimization problem into a number of independent sub-problems along with one type of variable, or restrict the search space for that type of variable into a smaller but most promising subspace, or both. mCore+ takes advantage of channel conditions and user correlations among MU-MIMO users to reduce the search space. In addition to the problem decomposition/reduction, the exploration of parallel computing is carried out throughout our design of mCore+, such as the user correlations and beamforming matrices are calculated in parallel among RBs.

- mCore+ is implemented on a COTS GPU platform NVIDIA DGX Station. We used two V100 GPU cards with the programming tool CUDA to perform our design. Special engineering efforts are performed to fit our problem into the GPU, which includes minimizing the data transfer and synchronization between GPU cards, exploiting streaming multi-processor's compute capability, practicing a proper indexing method, using shared memory wisely, etc. By taking advantage of GPU's massive-core architecture, mCore+ is able to accelerate the computation significantly.

- We validated mCore+'s performance through extensive experiments. The results show that mCore+ can offer a scheduling solution in $\sim$500 μs for a cellular system with up to 100 RBs, 100 users and 4×12 MIMO, which can meet the timing requirement for 5G numerology 1. Further, mCore+ can achieve better or comparable throughput performance compared with the state-of-the-art algorithms.

The rest of the paper is organized as follows. In Section 5.2, we review related work on cellular scheduler designs and GPU-based designs. In Section 5.3, we formulate the scheduling problem and state our objective. Section 5.4 presents our design of mCore+.

In Section 5.5, we offer the detailed implementation of mCore+ NVIDIA DGX Station. In Section 5.6, we show our experimental results to validate the performance of mCore+. Section 5.7 concludes this paper.

5.2 Related Work

We review the related work along the following two research lines.

Schedulers for Cellular Systems In the literature, there have been a number of research works that studied the design of cellular schedulers, including schedulers for single-antenna deployment and for MIMO systems.

In [95], the authors proposed GPF, which is a proportional fair (PF) scheduler design that allocates RBs and assigns MCS for users in a macro cell. This design can offer a scheduling solution in $\sim$100 μs for a user population size of up to 100 in a cell. Their experimental results show that GPF can achieve near-optimal performance per PF criterion. In [95, 133, 134, 136, 137], the authors proposed different heuristic schemes to allocate RB resources and determine MCS levels for each user. Unfortunately, all the designs in [95, 133, 134, 136, 137] do not consider multiple antennas at the BS or users. Thus, neither scheduling multiple users on an RB nor assigning multiple data streams for a user can be supported by these designs.

The designs in [131, 132, 135, 138, 139, 140, 141] designed schedulers for MIMO systems, which exploit the channel diversity and spatial multiplexing offered by MIMO channels. However, in [132, 135, 138], MU-MIMO transmission is not supported in their models. In [131, 139], their designs considered MU-MIMO transmission by allowing an RB to be shared by multiple users. But MCS assignment is not developed by their designs. In [140, 141],

the MU-MIMO scheduling problem is simplified as the problem is independent among RBs, because the authors do not apply the constraint of using the same MCS and the number of data streams across all RBs for a given user. In addition to the system models, all the proposed schemes in [131, 132, 135, 138, 139, 140, 141] require a large number of iterations to determine a solution. Due to the iterative nature of their algorithms, these designs cannot meet the sub-millisecond real-time requirement for 5G NR.

GPU-based Designs Recent advances in applying GPU to solve complex optimization problems have offered promising approaches to tackle the stringent real-time challenges. Different from most existing research that are largely limited to asymptotic complexity analysis (as in $O(\cdot)$), GPU-based designs are usually examined by actual "wall-clock" time, which is the ultimate benchmark in practice.

In the wireless communication community, GPU is employed in various areas of both PHY layer and MAC layer. In PHY layer, the authors in [120, 142] designed MIMO detectors that utilize GPU's parallel computing capability. In [146], the authors proposed parallel algorithms on GPU to accelerate LDPC decoding. The work [145] presented a GPU-based real-time solution to find digital beamforming weights for MU-MIMO users under hybrid architecture.

In MAC layer, the authors in [95] proposed a PF scheduler for 5G networks that can offer a near-optimal solution in $\sim$100 μs by leveraging GPU's massive parallel cores. In [143], the authors applied GPU to solve an age-of-information (AoI) minimization problem. The design in [144] solved a spectrum sharing problem based on chance-constrained programming, which employed GPU platform to meet 5G's timing requirement.

Unfortunately, all these GPU-based works are fundamentally different from the MU-MIMO scheduling problem we are studying in this paper. Their proposed approaches cannot

be applied to address our problem.

5.3 System Model

We consider a downlink (DL) scheduling problem for a 5G NR cellular system. As shown in Fig. 5.1, a BS serves a set $\mathcal{K}$ of users. The BS is equipped with N_{T} antennas and each user is equipped with N_{R} antennas ($N_{\mathrm{T}} > N_{\mathrm{R}}$). Table 5.1 gives the key notations that we use in this paper.

We consider time-slotted scheduling over a wide bandwidth. Within each time slot, there is a set $\mathcal{B}$ of RBs over the DL bandwidth. On each RB $b \in \mathcal{B}$, a subset of users $\mathcal{K}^b \subset \mathcal{K}$ is selected for MU-MIMO transmission[2]. Denote $x_k^b(t) \in \{0, 1\}$ as a binary variable indicating whether or not RB $b \in \mathcal{B}$ is scheduled by the BS for user $k \in \mathcal{K}$ in TTI t, i.e.,

$$x_k^b(t) = \begin{cases} 1, & \text{if RB } b \text{ is used for user } k \text{ in TTI } t, \\ 0, & \text{otherwise.} \end{cases}$$

For $x_k^b(t)$, we have the following constraint.

Constraint 1. The maximum number of users scheduled in an RB cannot exceed the number of antennas at the BS, i.e.,

$$\sum_{k \in \mathcal{K}} x_k^b(t) \leq N_{\mathrm{T}}. \qquad (b \in \mathcal{B}) \tag{5.1}$$

Further, each user $k \in \mathcal{K}$ may have multiple data streams (also known as "layers" in specifications [128]) simultaneously. However, to reduce the feedforward control signaling

[2]The smallest scheduling resolution in 5G can be a number of consecutive RBs, known as Resource Block Group (RBG). Our design can be easily extended to RBG-based scheduling.

Table 5.1: Notations in Chapter 5

Symbol	Definition
$\mathcal{B}$	The set of RBs to be allocated in a time slot
$\mathbf{F}_k^b(t)$	The precoding matrix for user k used on RB b in TTI t
$\mathbf{H}_k^b$	The channel matrix for user k on RB b
$\mathcal{K}$	The set of users
$\mathcal{K}^b$	A subset of users using RB b
$\mathcal{M}$	The set of MCSs
N_{R}	Number of antennas at each user
N_{T}	Number of antennas at BS
$r_k^{b,f,m}(t)$	The instantaneous achievable data rate of user k's f-th data stream on RB b with MCS m in TTI t
$R_k(t)$	The aggregate data rate of user k in TTI t
$\tilde{R}_k(t)$	The exponentially smoothed average data rate of user k up to TTI t
$x_k^b(t)$	A binary variable indicating whether or not RB b is scheduled for user k in TTI t
$y_k(t)$	Number of data streams for user k in TTI t
$z_k^m(t)$	A binary variable indicating whether or not MCS m is used for user k in TTI t

overhead and signal processing complexity, 5G NR imposes the following constraint.

Constraint 2. If a user is scheduled to receive signals on multiple RBs, then the user must have the same number of data streams across all RBs that are allocated to her [128].

Denote $y_k(t)$ (a non-negative integer) as the number of data streams for user k in TTI t (which is the same across all allocated RBs). As $y_k(t)$ cannot be greater than the number of receive antennas, we have

$$y_k(t) \leq N_{\mathrm{R}}. \qquad (k \in \mathcal{K})$$

Also, the total number of data streams on each RB for MU-MIMO transmission cannot

exceed the number of antennas at the BS. We have

$$\sum_{k\in\mathcal{K}} x_k^b(t) y_k(t) \leq N_{\mathrm{T}}. \qquad (b \in \mathcal{B}) \tag{5.2}$$

In each TTI t, a set $\mathcal{M}$ of MCS is available for users for data transmission. We have the following constraint for MCS selection.

Constraint 3. If a user is scheduled to receive data streams on multiple RBs, then the user must use the same MCS across all data streams on all scheduled RBs [4].

Denote $z_k^m(t) \in \{0, 1\}$ as a binary variable indicating whether or not MCS $m \in \mathcal{M}$ is used by the BS for user $k \in \mathcal{K}$ in TTI t, i.e.,

$$z_k^m(t) = \begin{cases} 1, & \text{if MCS } m \text{ is used for user } k \text{ in TTI } t, \\ 0, & \text{otherwise.} \end{cases}$$

To guarantee only one MCS is used across all scheduled RBs for user k, we have

$$\sum_{m\in\mathcal{M}} z_k^m = 1. \qquad (k \in \mathcal{K}) \tag{5.3}$$

The BS applies precoders to support MU-MIMO and multiple data streams. Let $\mathbf{F}_k^b(t)$ be an $N_{\mathrm{T}} \times x_k^b(t) y_k(t)$ precoding matrix for user k used by the BS scheduler on RB b. To meet the power constraint at the BS, we have $\sum_{k\in\mathcal{K}} ||\mathbf{F}_k^b(t)||_F^2 \leq P_{\mathrm{T}}$, where P_{T} is the total power for RB b at the BS and $||\cdot||_F$ denotes the Frobenius norm. Then the received signal of user k on RB b is given by

$$\boldsymbol{c}_k^b = \mathbf{H}_k^b \mathbf{F}_k^b \boldsymbol{s}_k^b + \mathbf{H}_k^b \sum_{i\in\mathcal{K}}^{i\neq k} \mathbf{F}_i^b \boldsymbol{s}_i^b + \boldsymbol{n}_k^b, \quad (k \in \mathcal{K}^b, b \in \mathcal{B})$$

where $\mathbf{H}_k^b \in \mathbb{C}^{N_\mathrm{R} \times N_\mathrm{T}}$ is the channel matrix for user $k \in \mathcal{K}$ on RB $b \in \mathcal{B}$, $\boldsymbol{n}_k^b$ is the $N_\mathrm{R} \times 1$ vector of i.i.d $\mathcal{CN}(0, n_0^2)$ additive complex Gaussian noise, $\boldsymbol{s}_k^b$ is the signal vector, and we omit the notation (t) for matrices for brevity.

Each user k computes the SVD of $\mathbf{H}_k^b = \mathbf{U}_k^b \, {}_k^b\mathbf{V}_k^{b\dagger}$, where $(\cdot)^\dagger$ denotes the conjugate transpose of a matrix. The ${}_k^b$ and $\mathbf{V}_k^b$ are further compressed, quantized and fed back to the BS (to assist precoding), and the leftmost y_k columns of $\mathbf{U}_k^b$, denoted as $\mathbf{U}_k^{b(y_k)}$, is used as the combiner at user k. After this combiner, we have the following signal:

$$\tilde{\boldsymbol{c}}_k^b = \mathbf{U}_k^{b(y_k)\dagger} \boldsymbol{c}_k^b = {}_k^{b\dagger}\mathbf{F}_k^b \boldsymbol{s}_k^b + {}_k^{b\dagger} \sum_{i \in \mathcal{K}^b}^{i \neq k} \mathbf{F}_i^b \boldsymbol{s}_i^b + \mathbf{U}_k^{b(y_k)\dagger} \boldsymbol{n}_k^b,$$

where ${}_k^b = [\sigma_k^b(1)\boldsymbol{v}_k^b(1), \cdots, \sigma_k^b(y_k)\boldsymbol{v}_k^b(y_k)]$ is an $N_\mathrm{T} \times y_k$ matrix, $\sigma_k^b(i)$ is the i-th eigenvalue of ${}_k^b$, and $\boldsymbol{v}_k^b(i)$ is the i-th column of $\mathbf{V}_k^b$. Thus, ${}_k^b$ is the effective channel after applying combiners at the users. Different precoding schemes can be used based on ${}_k^b$. In this paper, we apply ZF precoding scheme with equal power allocation for each data stream.[3]

For $k \in \mathcal{K}$, the signal-to-interference-plus-noise ratio (SINR) of the f-th stream on RB b is then given by

$$\mathrm{SINR}_k^{b,f} = \frac{\gamma_k^{b,f}}{\mathbf{Q}_k^{b,f} - \gamma_k^{b,f}}, \tag{5.4}$$

where

$$\gamma_k^{b,f} = {}_k^{b,f\dagger}\mathbf{F}_k^{b,f}\mathbf{F}_k^{b,f\dagger} \, {}_k^{b,f}$$
$$\mathbf{Q}_k^{b,f} = n_0^2 + \sum_{i \in \mathcal{K}} {}_k^{b,f\dagger}\mathbf{F}_i^b\mathbf{F}_i^{b\dagger} \, {}_k^{b,f},$$

and $(\cdot)_k^{b,f}$ is the f-th column of $(\cdot)_k^b$.

The instantaneous achievable data rate depends on the SINR of each stream and the selected MCS level. Specifically, for each user $k \in \mathcal{K}$, a higher MCS level m corresponds to a

[3]Other linear precoding schemes (e.g., MMSE) can also be applied.

higher data rate r^m in transmission. However, a certain level of SINR at each stream of user k is required in order to successfully decode the data. Denote θ^m as the SINR threshold for each data stream to successfully decode the data with MCS m. Then we have the following constraint.

Constraint 4. If the SINR of user k's f-th data stream on RB b is greater than or equal to θ^m, then the instantaneous achievable data rate of that stream is r^m; otherwise the achievable data rate drops to zero.

Denote $r_k^{b,f,m}(t)$ as the instantaneous achievable data rate of user k's f-th data stream on RB b with MCS m in TTI t. Then

$$r_k^{b,f,m}(t) = \begin{cases} r^m, & \text{if } \mathrm{SINR}_k^{b,f} \geq \theta^m, \\ 0, & \text{otherwise.} \end{cases} \tag{5.5}$$
$$(f = 1, \cdots, y_k(t), k \in \mathcal{K}, b \in \mathcal{B}, m \in \mathcal{M})$$

The aggregate achievable data rate of user k in TTI t can be given by

$$R_k(t) = \sum_{b \in \mathcal{B}} x_k^b(t) \sum_{f=1}^{y_k(t)} \sum_{m \in \mathcal{M}} z_k^m(t) r_k^{b,f,m}(t),$$

where we define $\sum_{f=1}^{y_k(t)}(\cdot) = 0$ if $y_k(t) = 0$.

Objective Function. To optimize the throughput performance with fairness consideration, we apply the widely used PF data rate as our performance objective. Denote $\tilde{R}_k$ as the long-term average data rate of user k. The PF objective function is then given by

$$\sum_{k \in \mathcal{K}} \log \tilde{R}_k. \tag{5.6}$$

Our real-time scheduler aims to maximize (5.6) by making scheduling decisions in each TTI t. A common approach is to maximize the sum of each user's instantaneous rate normalized by its exponentially smoothed average data rate over the past T_c TTIs [148, 149], i.e.,

$$\sum_{k\in\mathcal{K}} \frac{R_k(t)}{\tilde{R}_k(t-1)}, \tag{5.7}$$

where

$$\tilde{R}_k(t-1) = \frac{T_c - 1}{T_c}\tilde{R}_k(t-2) + \frac{1}{T_c}R_k(t-1).$$

It has been shown that maximizing (5.7) in each TTI is asymptotically approaching PF objective (5.6) when $T_c \to \infty$.

Problem Statement. Our objective is to allocate RBs, MCSs, assign the number of data streams, as well as compute precoding matrices for all users in each TTI, such that the PF objective function (5.7) is maximized. This 5G MU-MIMO scheduling problem can be written as follows.

OPT

$$\max \sum_{b\in\mathcal{B}}\sum_{k\in\mathcal{K}}\sum_{m\in\mathcal{M}}\sum_{f=1}^{y_k(t)} \frac{r_k^{b,f,m}(t)}{\tilde{R}_k(t-1)} x_k^b(t) z_k^m(t)$$

s.t. RB allocation constraint: (5.1);

Data stream allocation constraints: (5.2);

MCS assignemnt constraint: (5.3);

SINR and instantaneous data rate: (5.4), (5.5);

$x_k^b(t) \in \{0,1\}, \qquad (b\in\mathcal{B}, k\in\mathcal{K})$

$y_k(t) \in \{0,1,\cdots,N_{\mathrm{R}}\}, \quad (k\in\mathcal{K})$

$z_k^m(t) \in \{0,1\}. \qquad (k\in\mathcal{K}, m\in\mathcal{M})$

In problem OPT, $x_k^b(t)$, $y_k(t)$ and $z_k^m(t)$ are decision variables, $\mathbf{F}^b(t)$, $r_k^{b,f,m}(t)$ and $\tilde{R}_k(t-1)$ are intermediate variables, and the others are given constants.

Real-Time Challenge Problem OPT is NP-hard [131, 132], and the solution space is extremely large. The number of possible MCS assignments is $|\mathcal{M}|^{|\mathcal{K}|}$. Under the MU-MIMO setting, the number of possible RB allocations is $|\mathcal{B}|\left[\begin{pmatrix}|\mathcal{K}|\\1\end{pmatrix}+\cdots+\begin{pmatrix}|\mathcal{K}|\\N_{\mathrm{T}}\end{pmatrix}\right]$. Also, the number of possible data stream allocations is $N_{\mathrm{R}}^{|\mathcal{K}|}$. These give us a total number of $(|\mathcal{M}|N_{\mathrm{R}})^{|\mathcal{K}|}|\mathcal{B}|\left[\begin{pmatrix}|\mathcal{K}|\\1\end{pmatrix}+\cdots+\begin{pmatrix}|\mathcal{K}|\\N_{\mathrm{T}}\end{pmatrix}\right]$ possibilities in the solution space. In a typical 5G cellular system, this number can be as large as $\sim 10^{189}$ (when $|\mathcal{M}|=29, |\mathcal{K}|=100, |\mathcal{B}|=100, N_{\mathrm{T}}=8$ and $N_{\mathrm{R}}=2$).

On the other hand, we have a stringent timing requirement. A scheduler needs to find its scheduling solution within each TTI. In 5G NR, the longest time interval for a TTI is 1 ms (numerology 0) [91]. To support ultra-low latency applications, the numerology 1 with 500-μs TTI may be applied. In this paper, we set the real-time requirement to

$$T_{\mathrm{req}} = 500\ \mu\mathrm{s} \tag{5.8}$$

for a 5G scheduler. With the high-complexity problem and stringent timing requirement, none of the existing works can provide a solution to meet our goal.

5.4 mCore+: A Novel Design of Real-Time MU-MIMO Scheduler

5.4.1 Main Ideas and Road Map

The main ideas are twofold. First, mCore+ decomposes OPT into a large number of independent sub-problems. The goal is to leverage parallel computing resources to solve the sub-problems concurrently. Second, mCore+ judiciously reduces the large search space into a smaller but most promising search subspace, leveraging insights from channel conditions and correlations.

As expected, decomposing OPT or narrowing the search space is not trivial, as we have multiple sets of decision variables (i.e., $x_k^b(t)$'s, $y_k(t)$'s and $z_k^m(t)$'s) and they are tightly coupled with each other. The combinations make the solution space extremely large. Therefore, a simple parallel algorithm that exhaustively checks all possibilities is not possible. To address this problem, mCore+ judiciously solves OPT through a multi-phase optimization. At each phase, mCore+ focuses on one type of variable (i.e., $x_k^b(t)$'s, $y_k(t)$'s or $z_k^m(t)$'s). That is, each phase will either decompose the optimization problem into a number of independent sub-problems along with that type of variable, or restrict the search space for that type of variable into a smaller but most promising subspace, or both.

The multi-phase optimization is illustrated in Fig. 5.2. At Phase 1, problem OPT is decomposed along $z_k^m(t)$ variables, which corresponds to an MCS selection problem. From $|\mathcal{M}|^{|\mathcal{K}|}$ possible MCS assignments, mCore+ selects N_{P1} promising candidates, based on channel conditions. Thus problem OPT is split into N_{P1} independent sub-problems, named OPT-P1.

At Phase 2, we focus on reducing the search space along $x_k^b(t)$ variables, which implies a

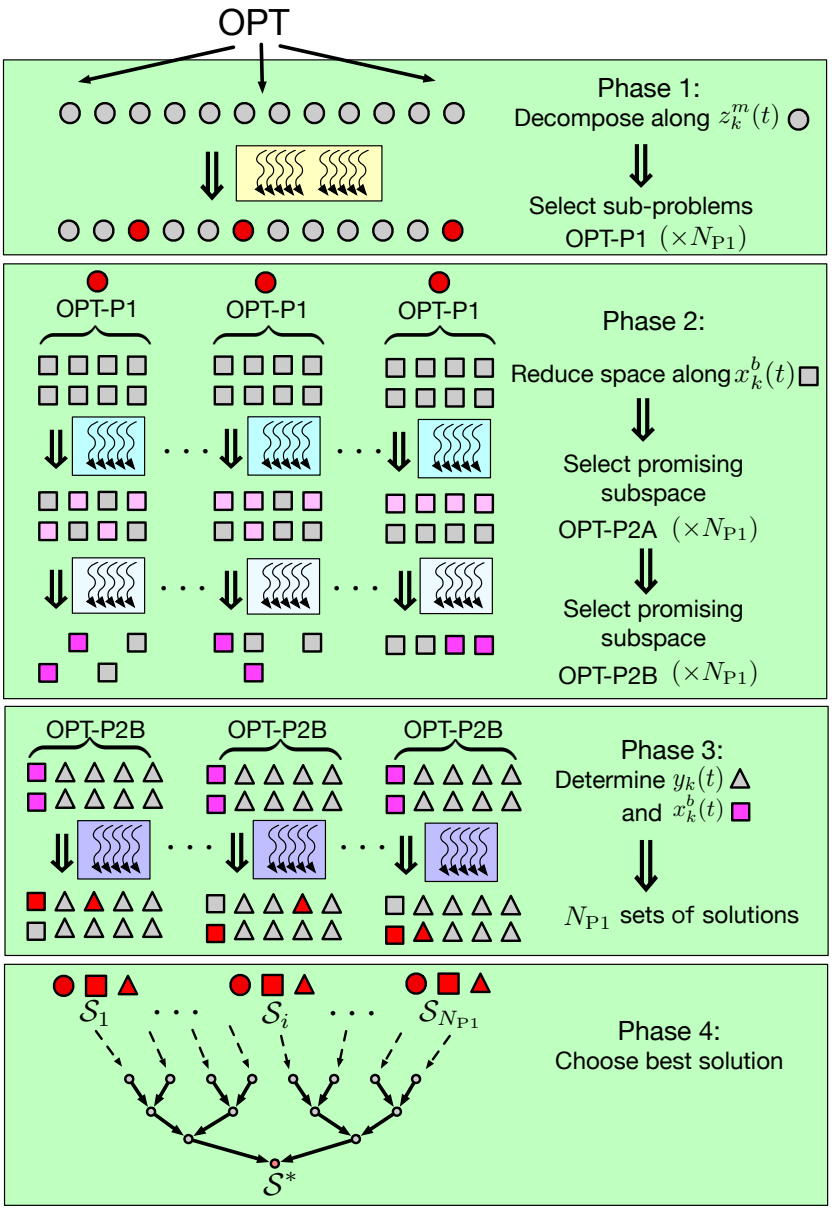


Figure 5.2: mCore+ solves OPT through a multi-phase process, leveraging parallel computation in each phase.

user selection problem for MU-MIMO transmission. Phase 2 is composed of two steps. Step A evaluates the channel qualities q_k^b normalized by long-term average $\tilde{R}_k(t-1)$. Then the allocation of RB b is restricted to a subset of users $\widetilde{\mathcal{K}}^b \subset \mathcal{K}$. In Step B of Phase 2, mCore+ measures the channel orthogonality among users $\widetilde{\mathcal{K}}^b$, then the RB allocation is further limited to a smaller user set $\mathcal{K}^b \subset \widetilde{\mathcal{K}}^b$ ($|\mathcal{K}^b| \leq N_{\mathrm{T}}$) that offers good orthogonality among the users within the set. After Phase 2, we still have N_{P1} independent sub-problems, named OPT-P2B, while the number of RB allocation possibilities is reduced from $|\mathcal{B}| \left[\binom{|\mathcal{K}|}{1} + \cdots + \binom{|\mathcal{K}|}{N_{\mathrm{T}}} \right]$ to $|\mathcal{B}| \left[\binom{|\mathcal{K}^b|}{1} + \cdots + \binom{|\mathcal{K}^b|}{|\mathcal{K}^b|} \right]$.

At Phase 3, we focus on determining the number of data streams for each user, which is to decide $y_k(t)$ variables. We also determine $x_k^b(t)$ variables in the meanwhile. Phase 3 has two steps. In Step A, we relax $y_k(t)$ to a set of $y_k^b(t)$'s by allowing the number of data streams to be different on different RBs. This effectively decouples problems OPT-P2B among different RBs. Thus problems OPT-P2B are decomposed into $N_{\mathrm{P1}}|\mathcal{B}|$ independent sub-problems, denoted as OPT-P3A. Each OPT-P3A is now a small problem that can be solved easily by checking all promising solutions, leveraging massive parallel computing resources. In Step B, we address the feasibility to the original problem OPT (i.e., those infeasible solutions due to relaxation of $y_k(t)$'s in Phase 3 Step A). This is done by another $N_{\mathrm{P1}}|\mathcal{K}|$ independent sub-problems (denoted as OPT-P3B). After Phase 3, we have N_{P1} sets of promising and feasible solutions as scheduling candidates.

Finally in Phase 4, among the N_{P1} intermediate best solutions, the best solution is chosen as the final scheduling solution to problem OPT.

5.4.2 Design Details

We have two main principles that will be carried out throughout the design of mCore+:

- Exploring parallelism: the decomposition should be able to generate a large number of independent sub-problems that can be fit into a given GPU platform. Also, each sub-problem should have an identical structure such that we can take advantage of GPU's single-instruction-multiple-data (SIMD) architecture for high efficiency.
- Finding the most promising search space: the search space is reduced into a smaller but most promising area, leveraging the insights from channel conditions and correlations.

In the rest of this section, we present the design details of mCore+ based on the above two principles.

Phase 1: MCS Selection. We first decompose problem OPT along $z_k^m(t)$ variables, which corresponds to fixing MCS in each sub-problem. If we enumerate all possibilities of $z_k^m(t)$'s, this will give us $|\mathcal{M}|^{|\mathcal{K}|}$ sub-problems in total, which is too large to be handled in real-time. We now show how mCore+ chooses a promising subset of sub-problems.

First, we identify the largest possible MCS that can be used by a user. On RB b, the largest possible SINR of user k happens when RB b is scheduled exclusively for user k (i.e., no co-scheduled MU-MIMO users) and only one data stream is transmitted to user k. In this case, user k gets all transmit power P_{T} exclusively and no inter-stream interference exists. By (5.4), the best possible SINR of user k on RB b is $\frac{P_{\mathrm{T}}\sigma_k^b(1)^2}{n_0^2}$. Denote the largest eigenvalue of user k's channels over all RBs as σ_k^*, i.e., $\sigma_k^* = \max\{\sigma_k^b(1)|b \in \mathcal{B}\}$. Then we must have the best possible SINR of user k's streams across all RBs is $\frac{P_{\mathrm{T}}\sigma_k^{*2}}{n_0^2}$, i.e.,

$$\mathrm{SINR}_k^{b,f} \leq \frac{P_{\mathrm{T}}\sigma_k^{*2}}{n_0^2}. \quad (b \in \mathcal{B}, f \in \{1, \cdots, N_{\mathrm{R}}\}) \tag{5.9}$$

Thus, the highest MCS that user k may use, denoted as $\overline{m}_k$, can be determined by the user k's best possible SINR. That is

$$\overline{m}_k = \max_{m \in \mathcal{M}} \left\{ m \left| \frac{P_{\mathrm{T}} \sigma_k^{*2}}{n_0^2} \geq \theta^m \right. \right\}. \qquad (k \in \mathcal{K})$$

On one hand, simply applying $\overline{m}_k$ to user k may not be the best choice, as a user may use a lower MCS to support more RBs and streams at the same time. On the other hand, choosing an MCS much lower than $\overline{m}_k$ is not helpful. The intuition is that applying an MCS much worse than channel quality is unlikely to improve the PF objective. Therefore, we consider user k's MCS to be chosen from the set $\mathcal{M}_k$ of top 10 highest MCS that can be applied to user k, where $\mathcal{M}_k$ is given by

$$\mathcal{M}_k = \{m \in \mathcal{M} | 0 \leq \overline{m}_k - m < 10\} \subset \mathcal{M}.$$

Let $\widetilde{\mathcal{M}}$ be the Cartesian product of sets $\mathcal{M}_1, \mathcal{M}_2 \cdots \mathcal{M}_k$, i.e.,

$$\widetilde{\mathcal{M}} = \mathcal{M}_1 \times \mathcal{M}_2 \times \cdots \times \mathcal{M}_{|\mathcal{K}|} \subset \mathcal{M}^{|\mathcal{K}|}.$$

Accordingly, the MCS selection for all users will be one of the elements in $\widetilde{\mathcal{M}}$. Now the important step is that, instead of picking one element from $\widetilde{\mathcal{M}}$ at a time, we propose to randomly select a number of elements from $\widetilde{\mathcal{M}}$ at once, as MCS candidates, to solve the problem. There are different approaches to have the random selection. One approach is based on the PDF extracted from previous experience and statistics. As the MCS sets $\mathcal{M}_k$'s for each user are already promising candidates, we apply a simple uniform selection in this paper, and show it will offer an adequate solution.

Suppose the number of elements we choose from $\widetilde{\mathcal{M}}$ is N_{P1}, which will be determined

based on GPU capability and PF performance in experiments. We then have N_{P1} independent sub-problems, each of which corresponds to a given MCS assignment. Denote the i-th random selection as $\boldsymbol{m}_i$ (a length $|\mathcal{K}|$ vector). Then in the i-th sub-problem, $z_k^m(t)$ is given by

$$z_k^m(t) = \begin{cases} 1, & \text{if } m = \boldsymbol{m}_i(k), \\ 0, & \text{otherwise}, \end{cases} \tag{5.10}$$

where $\boldsymbol{m}_i(k)$ is the k's element of $\boldsymbol{m}_i$.

Denote $\mathcal{S}_i^{\mathcal{Z}}$ as MCS solution for the i-th sub-problem. Note that for each $\mathcal{S}_i^{\mathcal{Z}}$ we still have $\sum_{m\in\mathcal{M}} z_k^m(t) = 1$. Denote m_k^* as the (one and only one) MCS that satisfies $z_k^{m_k^*}(t) = 1$. Then the problems we are going to solve after Phase 1 are N_{P1} independent sub-problems as follows.

OPT-P1 $(\times N_{\mathrm{P1}})$

$$\max \sum_{b\in\mathcal{B}}\sum_{k\in\mathcal{K}}\sum_{f=1}^{y_k(t)} \frac{r_k^{b,f,m_k^*}(t)}{\tilde{R}_k(t-1)} x_k^b(t)$$

s.t. Constraints (5.1), (5.2), (5.4), (5.5).

Phase 1 is designed to be performed in parallel, as illustrated in Fig. 5.3. mCore+ first generates $|\mathcal{B}||\mathcal{K}|$ parallel threads, each of which holds one value of $\sigma_k^b(1)$. Then by leveraging parallel reduction technique [116], we find the largest eigenvalue σ_k^* over all RBs for each user k. Next, through $|\mathcal{K}|$ independent threads, mCore+ calculates the best possible SINR $\frac{P_{\mathrm{T}}\sigma_k^{*2}}{n_0^2}$ and also determines the highest possible MCS $\overline{m}_k$ for each user. Now we have $\widetilde{\mathcal{M}}$ based on $\overline{m}_k$'s. mCore+ then spawns $N_{\mathrm{P1}}|\mathcal{K}|$ threads to randomly select MCS candidates. Specifically, every $|\mathcal{K}|$ threads are used to pick one $\boldsymbol{m}_i$ for $|\mathcal{K}|$ users in sub-problem i. All the $N_{\mathrm{P1}}|\mathcal{K}|$ MCS candidates (for all $N_{\mathrm{P1}}|$ sub-problems) are selected in parallel.

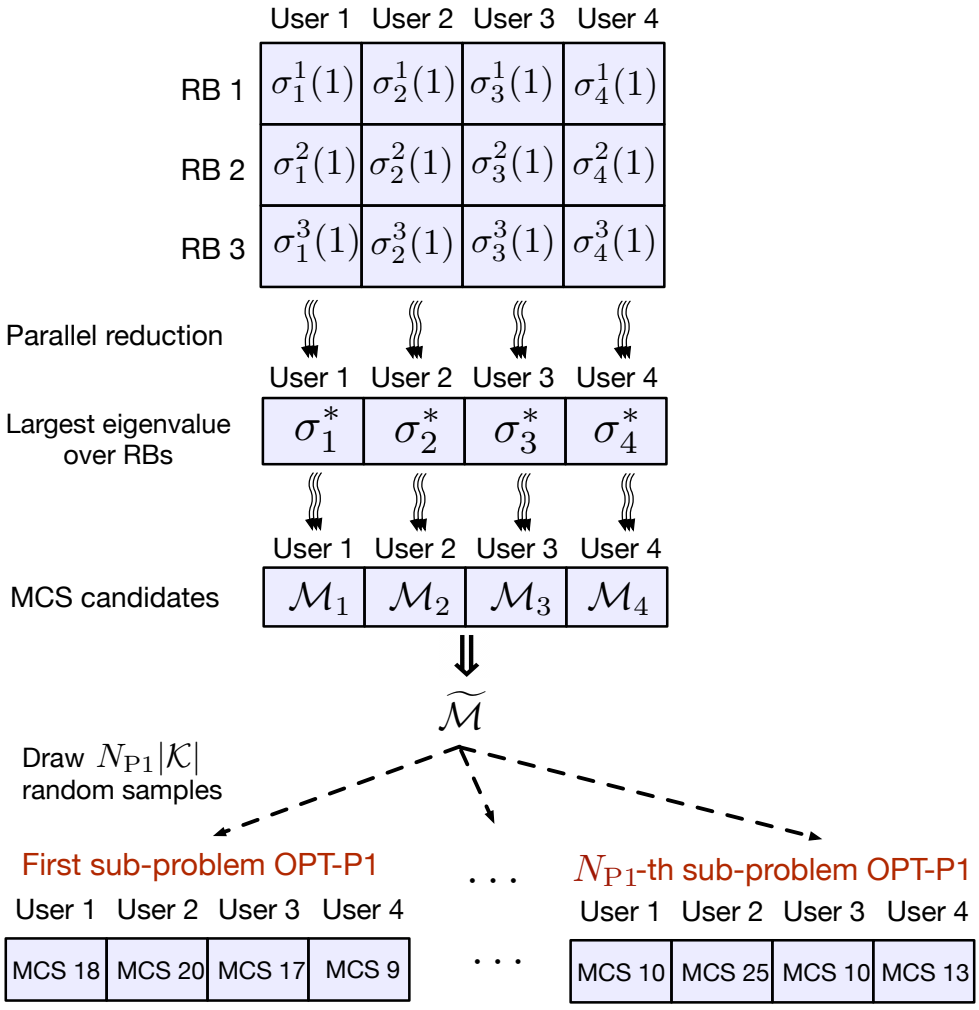


Figure 5.3: The illustration of the parallel design of Phase 1.

Phase 2: User Selection. In Phase 2, we focus on finding promising $x_k^b(t)$'s for problem OPT-P1. This means to select a group of users on each RB to form MU-MIMO transmission. For each $b \in \mathcal{B}$, at most N_{T} of $x_k^b(t)$'s can be non-zero (i.e., at most N_{T} users can be grouped for MU-MIMO on each RB). This gives us a total number of $|\mathcal{B}| \left[\binom{|\mathcal{K}|}{1} + \cdots + \binom{|\mathcal{K}|}{N_{\mathrm{T}}} \right]$ possibilities to assign $x_k^b(t)$'s.

Generally speaking, scheduling too many users on the same RB is not helpful in terms of improving data rate. This is because: 1) scheduling more users would cause higher correlations among the users, which will deteriorate SINR due to interference; 2) transmit power P_{T} will be split among the users, thus more users do not necessarily mean a higher sum rate. mCore+ identifies promising MU-MIMO users with the following two steps. In Phase 2-A, we pinpoint a subset of users $\widetilde{\mathcal{K}}^b \subset \mathcal{K}$ on each RB with better channel qualities at current TTI t that are likely to lead to a high data rate. In Phase 2-B, a smaller subset of users $\mathcal{K}^b \subset \widetilde{\mathcal{K}}^b$ is chosen as the candidates for MU-MIMO transmission, which is based on channel correlations. The number of possibilities to assign $x_k^b(t)$'s is then reduced to $|\mathcal{B}| \left[\binom{|\mathcal{K}^b|}{1} + \cdots + \binom{|\mathcal{K}^b|}{|\mathcal{K}^b|} \right]$ after Phase 2.

- *Phase 2-A.*

A subset of users $\widetilde{\mathcal{K}}^b \subset \mathcal{K}$ on each RB is selected based on channel qualities normalized by long-term average $\tilde{R}_k(t-1)$. Similar to Phase 1, the achievable data rate is directly tied to the channels' eigenvalues. We consider the largest eigenvalue $\sigma_k^b(1)$ of each user's channel on each RB. Then we determine $\widetilde{\mathcal{K}}^b$ based on the following metric:

$$q_k^b = \frac{\log\left(\frac{P_{\mathrm{T}}\sigma_k^b(1)^2}{n_0^2}\right)}{\tilde{R}_k(t-1)}. \tag{5.11}$$

Suppose we are going to select K_{P2A} users on each RB, i.e., $|\widetilde{\mathcal{K}}^b| = K_{\mathrm{P2A}}$ $(< |\mathcal{K}|)$. We

sort q_k^b's on each RB and $\widetilde{\mathcal{K}}^b$ is determined by choosing K_{P2A} users with the K_{P2A} highest q_k^b's. That is, let $\boldsymbol{\pi}^b$ be the descending ordering of $\{q_1^b, \cdots, q_{|\mathcal{K}|}^b\}$, and π_k^b is the order of q_k^b, then

$$\widetilde{\mathcal{K}}^b = \{k \in \mathcal{K} | \pi_k^b \leq K_{\text{P2A}}\}. \qquad (b \in \mathcal{B})$$

The intuition behind this selection is that, once the user is experiencing a better channel quality compared with its long-term average, it will have a higher chance to be scheduled. This behavior will maximize PF objective.

After this step, the search space regarding users on each RB is limited to $\widetilde{\mathcal{K}}^b \subset \mathcal{K}$, and the optimization problem can be written as:

OPT-P2A $(\times N_{\text{P1}})$

$$\max \quad \sum_{b\in\mathcal{B}} \sum_{k\in\widetilde{\mathcal{K}}^b} \sum_{f=1}^{y_k(t)} \frac{r_k^{b,f,m_k^*}(t)}{\bar{R}_k(t-1)} x_k^b(t)$$

$$\text{s.t.} \quad \text{Constraints (5.1), (5.2), (5.4), (5.5).}$$

- *Phase 2-B.*

This step determines a smaller user set $\mathcal{K}^b \subset \widetilde{\mathcal{K}}^b$ for possible MU-MIMO transmission on each RB. In Phase 2-A, we select users with plausible channel qualities. Another key impact on SINR performance is channel correlations among co-scheduled users, as multiple users co-scheduled in the same RB are mutually dependent. Generally speaking, the more orthogonality the scheduled users have, the higher SINR (and thus higher data rate) can be achieved after applying precoding schemes. This is because projecting co-scheduled signals onto mutually orthogonal subspaces (via precoding) to avoid interference would sacrifice more desire signal strength when the channels are less orthogonal.

In this step, we aim at identifying users with low channel correlations. To do this, we

first calculate channel correlations between every two users in $\widetilde{\mathcal{K}}^b$. mCore+ uses the metric *chordal distance* to measure the channel correlations. The chordal distance represents the *angle* between two subspaces $\mathbf{A}$ and $\mathbf{B}$, which is given by [150]

$$\frac{1}{\sqrt{2}}||\mathbf{A}_o\mathbf{A}_o^\dagger - \mathbf{B}_o\mathbf{B}_o^\dagger||_F, \tag{5.12}$$

where $\mathbf{A}_o$ and $\mathbf{B}_o$ are orthonormal bases for subspaces $\mathbf{A}$ and $\mathbf{B}$. One of the characteristics of chordal distance is that it can quantify the distance between multi-dimensional subspaces. Thus it is useful to measure the orthogonality between users when both the BS and users are with multiple antennas. Consequently, the chordal distance between two users can be given by

$$d_c^b(k_1, k_2) = \frac{1}{\sqrt{2}}||\mathbf{V}_{k_1}^b\mathbf{V}_{k_1}^{b\dagger} - \mathbf{V}_{k_2}^b\mathbf{V}_{k_2}^{b\dagger}||_F. \tag{5.13}$$

mCore+ calculates $d_c^b(k_1, k_2)$ for every two users $k_1, k_2 \in \widetilde{\mathcal{K}}^b$ on each RB $b \in \mathcal{B}$. Then mCore+ selects N_{S2B} users (i.e., $\mathcal{K}^b$) from $\widetilde{\mathcal{K}}^b$ on each RB as the candidates for MU-MIMO transmission.

$\mathcal{K}^b$ is determined by the following rules. First, on each RB b, mCore+ adds the first user (from $\widetilde{\mathcal{K}}^b$) to $\mathcal{K}^b$ that has the highest q_k^b (from Phase 2-A). Next, mCore+ adds users to $\mathcal{K}^b$ one by one by picking the largest average chordal distance to existing users in $\mathcal{K}^b$, until we have K_{P2B} users in $\mathcal{K}^b$. The process is computationally easy and can be simply expressed by the following:

While $|\mathcal{K}^b| < K_{\text{P2B}}$:

$\overline{k}^b = \arg\max_{k\in\widetilde{\mathcal{K}}^b} \frac{1}{|\mathcal{K}^b|}\sum_{k'\in\mathcal{K}^b} d_c^b(k, k')$;

$\mathcal{K}^b \leftarrow \mathcal{K}^b \cup \overline{k}^b$, $\widetilde{\mathcal{K}}^b \leftarrow \widetilde{\mathcal{K}}^b / \overline{k}^b$.

Now the candidate users for MU-MIMO on RB b are restricted to $\mathcal{K}^b$. Note that for

$k' \notin \mathcal{K}^b$, we have $x^b_{k'} = 0$. The final scheduled users on RB b will be determined in the next phase. After Phase 2-B, the remaining problem is given by

$$\textbf{OPT-P2B} \quad (\times N_{\mathrm{P1}})$$

$$\max \quad \sum_{b\in\mathcal{B}} \sum_{k\in\mathcal{K}^b} \sum_{f=1}^{y_k(t)} \frac{r_k^{b,f,m_k^*}(t)}{\widetilde{R}_k(t-1)} x_k^b(t)$$

$$\text{s.t.} \quad \text{Constraints (5.1), (5.2), (5.4), (5.5).}$$

The number of problem OPT-P2B we are going to solve is the same as that of OPT-P1, i.e., N_{P1}. After Phase 2, the number of RB allocation possibilities is reduced from $|\mathcal{B}|\left[\binom{|\mathcal{K}|}{1} + \cdots + \binom{|\mathcal{K}|}{N_{\mathrm{T}}}\right]$ to $|\mathcal{B}|\left[\binom{|\mathcal{K}^b|}{1} + \cdots + \binom{|\mathcal{K}^b|}{|\mathcal{K}^b|}\right]$.

Fig. 5.4 illustrates how we can execute Phase 2 in parallel. As the values of q_k^b's are independent among RBs and users, mCore+ can generate $|\mathcal{B}||\mathcal{K}|$ parallel threads to calculate them simultaneously. Next, every $|\mathcal{K}|$ threads are grouped as a block for cooperative operation (allowing certain information exchange within a block), and in total we have $|\mathcal{B}|$ independent blocks. Within b-th block, a sorting algorithm is performed on q_k^b's for RB b. The sorting results will determine $\widetilde{\mathcal{K}}^b$, which completes Phase 2-A. In Phase 2-B, the chordal distances are independent among RBs, and each distance only involves two users. Therefore, by using $|\mathcal{B}| \cdot \frac{1}{2}|\widetilde{\mathcal{K}}^b|^2$ parallel flows, all $d_c^b(k_1, k_2)$'s can be obtained at once. Finally, mCore+ applies $|\mathcal{B}||\widetilde{\mathcal{K}}^b|$ threads to find MU-MIMO candidates $\mathcal{K}^b$ on each RB. Similar to Phase 2-A, every $|\widetilde{\mathcal{K}}^b|$ threads are grouped in a block, and the b-th block is to find the best $|\mathcal{K}^b|$ orthogonal users for RB b.

Phase 3: Determining Stream Number. In this phase, we focus on determining $y_k(t)$ variables, and also decide $x_k^b(t)$ variables in the meanwhile. As a user k can transmit from 0 data streams to N_{R} data streams (across all scheduled RBs), the total number of

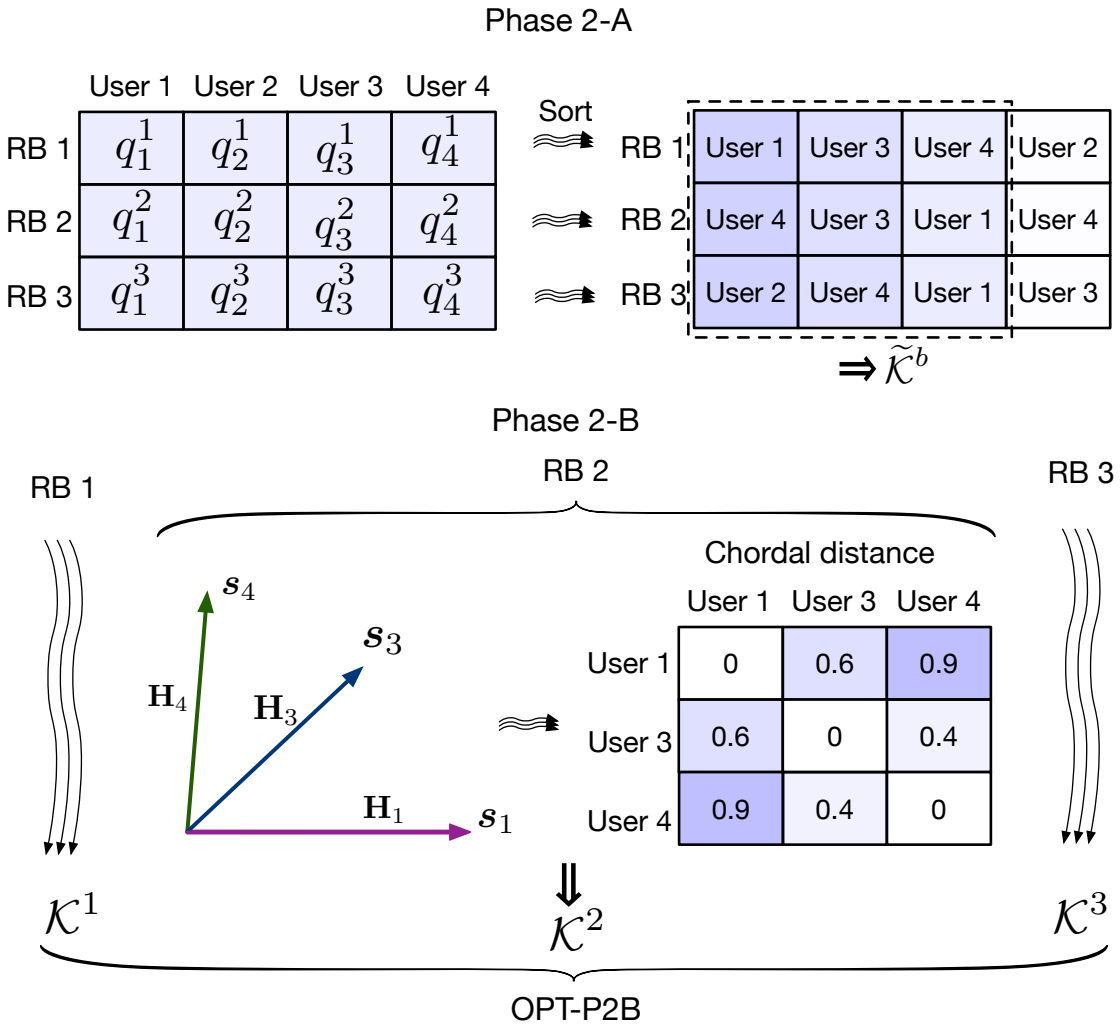


Figure 5.4: The illustration of the key steps of Phase 2, which is designed to be performed in parallel.

possibilities is $(N_{\mathrm{R}}+1)^{|\mathcal{K}|}$. This can be a huge number as $|\mathcal{K}|$ can be $\sim$100. We apply the following approach to reduce complexity.

We relax $y_k(t)$ by allowing user k to have a different number of data streams on different RBs. Denoted $y_k^b(t)$ as the number of user k's data streams on RB b, where $y_k^b(t)$ can be different from $y_k^{b'}(t)$ for $b \neq b'$. Then one problem OPT-P2B can be divided into $|\mathcal{B}|$ independent sub-problems. We first leverage parallel computing techniques to solve each sub-problem in Phase 3-A, then in Phase 3-B, we address the feasibility to the original problem OPT (i.e., to guarantee $y_k^b(t) = y_k^{b'}(t)$ for all $b \neq b'$ if $x_k^b(t) = x_k^{b'}(t) = 1$).

- *Phase 3-A.*

After the relaxation on $y_k(t)$'s, we have the following sub-problem for each $b \in \mathcal{B}$,

$$\textbf{OPT-P3A} \quad (\times N_{\mathrm{P1}}|\mathcal{B}|)$$

$$\max \quad \sum_{k\in\mathcal{K}^b}\sum_{f=1}^{y_k^b(t)} \frac{r_k^{b,f,m_k^*}(t)}{\tilde{R}_k(t-1)} x_k^b(t)$$

$$\text{s.t.} \quad \text{Constraints (5.1), (5.4), (5.5),}$$

$$y_k^b(t) \leq N_{\mathrm{R}},$$

$$\sum_{k\in\mathcal{K}} x_k^b(t) y_k^b(t) \leq N_{\mathrm{T}}.$$

In problem OPT-P3A, we have a total number of $(N_{\mathrm{R}}+1)^{|\mathcal{K}^b|}$ possibilities for $y_k^b(t)$'s. In practice, N_{R} typically ranges from 2 to 4. mCore+ will limit $y_k^b(t)$ to satisfy $0 \leq y_k^b(t) \leq 2$, as transmitting too many data streams on one user is not likely to improve sum rate [129, 130] (the constraint for total number of data streams on one RB remains the same as (5.2)). Then the total number possibilities for $y_k^b(t)$'s is $3^4 = 81$ when $|\mathcal{K}^b| = 4$. mCore+ conducts an exhaustive search in parallel to evaluate the objective values of problem OPT-P3A. Note that the assignment of $x_k^b(t)$ is implied by $y_k^b(t)$. That is, when $y_k^b(t)$ is assigned to be 0, then

$x_k^b(t) = 0$ (i.e., user k is not scheduled on RB b); when $y_k^b(t)$ is assigned to be 1 or 2, then $x_k^b(t) = 1$ (i.e., user k is scheduled on RB b). We denote the optimal solutions for $y_k^b(t)$ and $x_k^b(t)$ as $y_k^{b*}(t)$ and $x_k^{b*}(t)$, respectively.

- *Phase 3-B.*

Now we resolve the conflict of $y_k^{b*}(t) \neq y_k^{b'*}(t)$ for $b \neq b'$ if $x_k^{b*}(t) = x_k^{b'*}(t) = 1$ (i.e., to determine the final $x_k^b(t)$'s and $y_k(t)$'s). We apply the following heuristic to determine final $x_k^b(t)$'s and $y_k(t)$'s. The final scheduling solution will be determined independently among the users. When we determine a scheduling solution regarding user k, we fix all other users' solution by letting $x_{k'}^b(t) = x_{k'}^{b*}(t)$ and $y_{k'}(t) = y_k^{b*}(t)$ for all $k' \in \mathcal{K}, k' \neq k, b \in \mathcal{B}$. Then we choose the solution of $x_k^b(t)$ and $y_k(t)$ from those satisfying feasibility constraints in OPT (with MCS m_k^* from Phase 1). Denote $v\left(x_k^b, y_k, \{x_{k'}^{b*}\}/x_k^{b*}, \{y_{k'}^{b*}\}/y_k^{b*}\right)$ as the PF objective value achieved by $x_k^b(t)$, $y_k(t)$, and $x_{k'}^b(t) = x_{k'}^{b*}(t)$, $y_{k'}(t) = y_k^{b*}(t)$ for all $k' \in \mathcal{K}, k' \neq k$, which can be given by

$$v\left(x_k^b, y_k, \{x_{k'}^{b*}\}/x_k^{b*}, \{y_{k'}^{b*}\}/y_k^{b*}\right) = \sum_{b\in\mathcal{B}} \sum_{f=1}^{\mathcal{K}^b \ni k \; y_k(t)} \frac{r_k^{b,f,m_k^*}(t)}{\tilde{R}_k(t-1)} x_k^b(t) + \sum_{b\in\mathcal{B}} \sum_{k'\in\mathcal{K}^b}^{k'\neq k} \sum_{f=1}^{y_{k'}^{b*}(t)} \frac{r_{k'}^{b,f,m_{k'}^*}(t)}{\tilde{R}_{k'}(t-1)} x_{k'}^{b*}(t). \tag{5.14}$$

Note that the second term is a constant as all variables are being fixed. The only variables are $x_k^b(t)$ and $y_k(t)$ in the first term. To guarantee the feasibility constraints related to $x_k^b(t)$ and $y_k(t)$ in the original problem (i.e., constraints (5.1) and (5.2)), we impose $x_k^b(t) \leq x_k^{b*}(t)$ and $x_k^b(t)y_k(t) \leq x_k^{b*}(t)y_k^{b*}(t)$ for all $b \in \mathcal{B}$. As x_k^{b*} and $y_k^{b*}(t)$ are solutions to problem OPT-P3A, they must satisfy feasibility constraints (5.1) and (5.2) on any particular RB b. It is easy to verify such $x_k^b(t)$ and $y_k(t)$ will satisfy (5.1) and (5.2) with these imposed constraints. Formally, for each $k \in \mathcal{K}$, we have the following optimization problem to determine $x_k^b(t)$

and $y_k(t)$.

$$\textbf{OPT-P3B} \quad (\times N_{\mathrm{P1}}|\mathcal{K}|)$$

$$\max \quad \sum_{b\in\mathcal{B}} \sum_{f=1}^{\mathcal{K}^b \ni k \; y_k(t)} \frac{r_k^{b,f,m_k^*}(t)}{\tilde{R}_k(t-1)} x_k^b(t)$$

$$\text{s.t.} \quad \text{Constraints (5.4), (5.5)},$$

$$x_k^b(t) \le x_k^{b*}(t), \text{ for all } b \in \mathcal{B};$$

$$x_k^b(t) y_k(t) \le x_k^{b*}(t) y_k^{b*}(t), \text{ for all } b \in \mathcal{B}.$$

In Phase 1, we have N_{P1} different assignments for MCS. As problem OPT-P3B is designed to be independent for each user, we have $|\mathcal{K}| \times N_{\mathrm{P1}}$ independent problems of OPT-P3B. Each problem is fairly simple, as $y_k(t)$ has only three possibilities and $x_k^b(t)$ is restricted within a small set. Among the $|\mathcal{K}|N_{\mathrm{P1}}$ problems, there are $|\mathcal{K}|$ problems that corresponds to one MCS assignment $\mathcal{S}_i^{\mathcal{Z}}$ (from Phase 1). Denote the $|\mathcal{K}|$ solutions of $x_k^b(t)$'s and $y_k(t)$'s (for $|\mathcal{K}|$ users), corresponding to the i'th MCS assignment as $S_i^{\mathcal{X}}$ and $\mathcal{S}_i^{\mathcal{Y}}$, respectively. Then $S_i^{\mathcal{X}}$ and $\mathcal{S}_i^{\mathcal{Y}}$, along with corresponding MCS assignment $\mathcal{S}_i^{\mathcal{Z}}$, constitute a complete solution set. We denote the complete solution set as $\mathcal{S}_i = (\mathcal{S}_i^{\mathcal{X}}, \mathcal{S}_i^{\mathcal{Y}}, \mathcal{S}_i^{\mathcal{Z}})$. After solving OPT-P3B, mCore+ obtains N_{P1} sets of feasible solutions $\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_{N_{\mathrm{P1}}}$ to OPT.

Phase 3 is designed with the exploration to enable parallel implementation, as shown in Fig. 5.5. In Phase 3-A, thanks to the design of relaxed problem OPT-P3A, the calculations of SINR and objective values become independent among RBs. mCore+ uses $N_{\mathrm{P1}}|\mathcal{B}|$ parallel flows to solve $N_{\mathrm{P1}}|\mathcal{B}|$ problems of OPT-P3A independently (where each flow includes many threads to solve one problem). In Phase 3-B, the problems of OPT-P3B are constructed to be mutually independent among users. Such design allows us to use $N_{\mathrm{P1}}|\mathcal{K}|$ independent (but structurally identical) flows to solve each OPT-P3B.

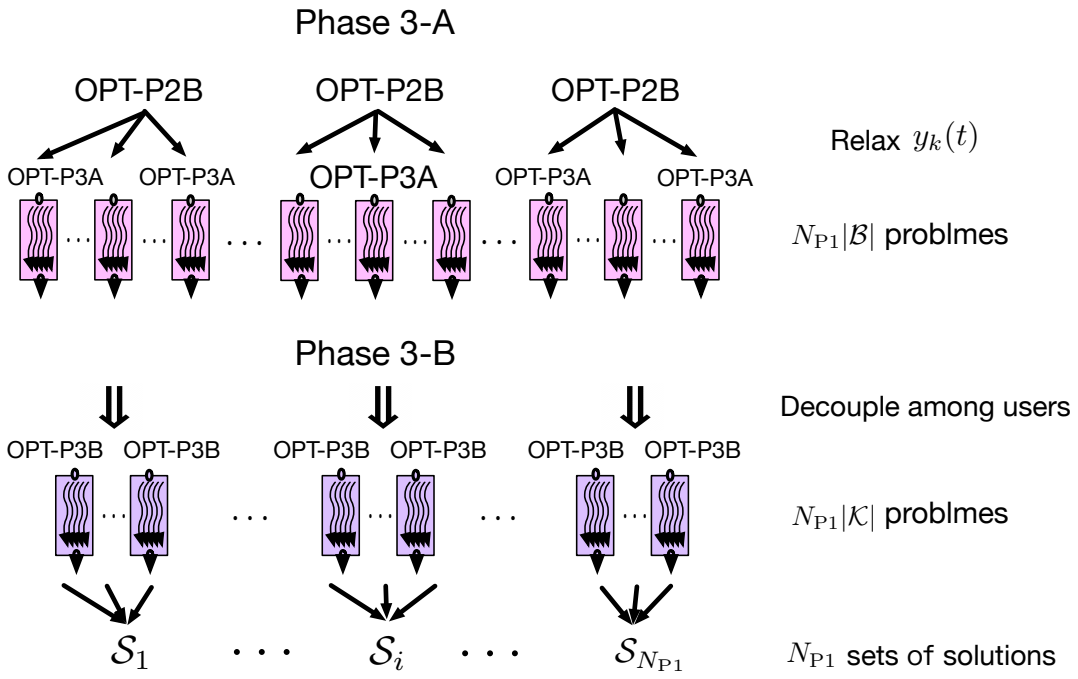


Figure 5.5: The illustration of the parallel design of Phase 3. It is designed to take advantage of parallel computation.

Phase 4: Comparison and Finding Best Solution. In this phase, we find the best MCSs $z_k^m(t)$'s, as well as corresponding $x_k^b(t)$'s and $y_k(t)$'s, that provide the best objective values. The approach is straightforward. Among all the promising solution sets $\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_{N_{\mathrm{P1}}}$, the one that can offer the highest PF objective value will be chosen as the final scheduling solution.

Note that all the objective values are already calculated in Phase 3. Here we just need to apply parallel techniques, such as parallel reduction, to compare and find the best solution.

5.5 Implementation

In this section, we present our implementation of mCore+. mCore+ is implemented on a COTS GPU platform—NVIDIA DGX Station. NVIDIA DGX Station includes 4 V100

GPU cards, and we only use two of them. Each V100 GPU card consists of an array of 80 streaming multiprocessors (SMs) with 5120 CUDA cores (64 cores per SM). Each SM includes 48 KB shared memory. The CPU of the DGX Station is Intel Xeon E5-2698 v4 2.2 GHz (20-core). The data communication between CPU and GPU is based on a PCIe V3.0 architecture, and data communication between different GPU cards is based on NVIDIA NVLink architecture [151]. Our programming platform is CUDA v10.2 [117].

5.5.1 Fitting mCore+ into the GPU

In addition to the design of a parallelizable algorithm, it is important for a designer to have full knowledge of the employed GPU and know how to fit the problems into the GPU to achieve high performance. Generally speaking, it is desired that all the GPU cores can keep busy calculating (to increase the so-called "achieved occupancy" [152]) with minimized memory access time. Following this principle, mCore+ is implemented with the following key considerations.

- *Execute all computation on the GPUs, with nearly zero support from CPU.* Our DGX Station is equipped with PCIe V3.0 and NVLink architecture—a relatively high-speed CPU-to-GPU, GPU-to-CPU, and GPU-to-GPU data transfer architecture. As mCore+ has decomposed problem OPT to a level that is suitable for massive parallel computation, we conclude that it is much faster to solely use GPUs for all computations. CPU will only be responsible for scheduling the kernels, controlling the data transfer, and synchronizing different flows as needed, while no actual computation for the problem is performed on CPU.

- *Minimize the data transfer between GPU cards.* Although high-speed data transfer between GPUs can be realized, the cost of corresponding synchronization can be

expensive. That is, the computation on both GPU cards may be suspended for sending/receiving data. Also, the extra scheduling overhead for synchronization may cause additional delay. To reduce such consumption, the best practice is to exploit independency between GPU cards and minimize the need for data transfer. For mCore+, most operations are independent among RBs. Therefore, the computation tasks are distributed between GPUs based on RBs. Only a small amount of data transfer between GPU cards is needed, such as sharing the MCS candidates with all RBs, and comparing the objective values on different RBs.

- *Exploit SM's compute capability.* A V100 GPU card consists of 80 SMs, each of which includes a set of computing resources: CUDA cores, registers, and shared memory, etc. SMs are responsible for creating, scheduling, and executing the parallel threads. Importantly, these threads are executed in groups, where a group is 32 consecutive threads known as a *warp*. Threads within a warp will execute exactly the same instructions simultaneously (while carrying different data). To fully utilize SM's compute capability, we need to achieve two goals here. First, generate a sufficiently large number of threads. Having a large number of threads in flight can keep all the GPU cores on each SM busy with calculations and therefore complete the mission in time. This number should be at least $80 \times 32 = 2560$, but can be much larger if the SM usage is not limited by factors such as the usage of registers and shared memory. Second, minimize the use of conditional branches (such as *if-then-else* statement) for threads within a warp. When conditional branches are used in a warp, only the threads in one of the branches can be executed in parallel, while threads in any other branches have to be suspended, due to the warp architecture. Therefore, it is essential to have every consecutive 32 threads (i.e., threads within a warp) follow the same instructions to achieve high efficiency.

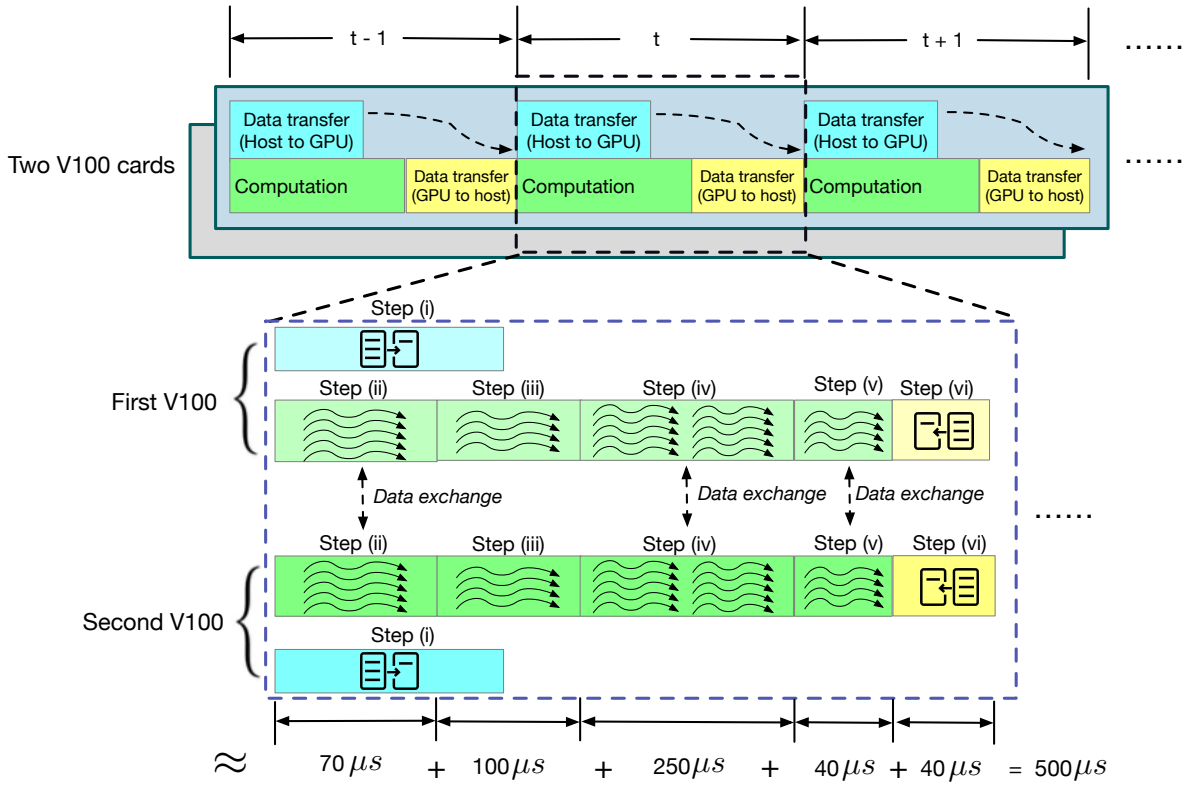


Figure 5.6: Implementation of mCore+ on two V100 GPU cards.

- *Use shared memory intelligently.* The on-chip shared memory is much faster than the global memory, but with limited storage space. In our problem, many operations that repeatedly acquire a small size of data can benefit from shared memory. For example, the parallel reduction technique with shared memory can be used to find the largest number in a group, and matrix inversion is also performed on shared memory.

5.5.2 Key Steps

mCore+ is implemented with the considerations in Section 5.5.1 throughout programming. As illustrated in Fig. 5.6, mCore+'s key steps in each TTI t is as follows.

- **Step (i): Transfer data from host device to GPU device.** To reduce the

time consumption for transferring the channel information to GPU, mCore+ exploits the parallelism between data transfer and GPU computation. Specifically, in TTI t, partial CSI $\frac{b}{k}$ and $\mathbf{V}_k^b$ for all users on all RBs that will be used in TTI $t+1$ are transferred from host device to GPU memory. The computation in current TTI t is based on the latest channel information that was transferred in TTI $t-1$. Therefore, the data transfer and GPU computation can be executed concurrently. To guarantee the channel information is still valid when we use it, the channel coherence time should be at least 3 TTIs (1 TTI for data transfer, 1 TTI for GPU computation, and 1 TTI for actual transmission), which can be satisfied for most communication scenarios under numerology 1. Further, since we use two GPU cards, we divide the channel matrices into two halves. As most operations are dependent among RBs by the design of mCore+, this division is based on RBs. That is, the $\frac{b}{k}$'s and $\mathbf{V}_k^b$'s corresponding to the first $\frac{|\mathcal{B}|}{2}$ are transferred to the first GPU card. The second half is transferred to the second GPU card.

• **Step (ii): Execute Phase 1.** On each GPU card, mCore+ generates $\frac{|\mathcal{B}|}{2}|\mathcal{K}|$ threads to obtain the largest eigenvalue of user k's channel over the first or second half of RBs. Then, the second GPU card sends the results to the first GPU card, so that the first GPU can obtain the largest eigenvalue σ_k^* over all RBs. Based on the σ_k^*'s, mCore+ generates $|\mathcal{K}|$ threads to calculate $\overline{m}_k$ (i.e., the highest MCS that user k can use) only using the first GPU card. The cuRAND library [153] is applied to draw N_{P1} random selections of $\boldsymbol{m}_i$'s from $\widetilde{\mathcal{M}}$ by $N_{\mathrm{P1}}|\mathcal{K}|$ threads. Subsequently, the first GPU card sends a copy of the MCS selection results $\boldsymbol{m}_i$'s to the second GPU card, which will be needed for future calculation on the second GPU. The value of N_{P1} can be determined empirically to strike an appropriate trade-off between computation time and PF performance. That is, we choose N_{P1} to be large enough (but not overly large) to obtain a satisfactory PF performance before consuming too much computation time. After this step, we have N_{P1} problems of OPT-P1.

• **Step (iii): Execute Phase 2.** The execution is independent between two GPU cards throughout Phase 2. In Phase 2-A, we generate a kernel with $\frac{|\mathcal{B}|}{2}|\mathcal{K}|$ threads to calculate q_k^b's on each GPU card in parallel. This kernel also finds the N_{S2A} highest q_k^b's in each block (for each RB), thus determines $\widetilde{\mathcal{K}}^b$. In Phase 2-B, to compute the chordal distances between every two users, a kernel with $\frac{|\mathcal{B}|}{2} \cdot |\widetilde{\mathcal{K}}^b| \cdot N_{\text{T}}^2$ first calculates $\mathbf{V}_k^b \mathbf{V}_k^{b\dagger}$ for each user. Then another kernel with $\frac{|\mathcal{B}|}{2} \cdot \frac{1}{2}|\widetilde{\mathcal{K}}^b|^2 \cdot N_{\text{T}}^2$ threads is created to calculate each element inside the Frobenius norm (5.13). Subsequently, within this kernel, the parallel reduction technique [116] is used to get the chordal distances $d_c^b(k_1, k_2)$ for every two users in $\widetilde{\mathcal{K}}^b, b \in \mathcal{B}$. Finally, another kernel with a size of $\frac{|\mathcal{B}|}{2} \times |\widetilde{\mathcal{K}}^b|$ (block by thread) on each GPU card is applied to find the best $|\mathcal{K}^b|$ orthogonal users on each RB.

• **Step (iv): Execute Phase 3.** To solve OPT-P3A, mCore+ first spawns $\frac{|\mathcal{B}|}{2}(2+1)^{|\widetilde{\mathcal{K}}^b|} \cdot N_{\text{T}} \cdot 2|\widetilde{\mathcal{K}}^b|$ threads on each GPU card to calculate each element of precoding matrices (with the dimension up to $N_{\text{T}} \times 2|\widetilde{\mathcal{K}}^b|$) for all $(2+1)^{|\widetilde{\mathcal{K}}^b|}$ possibilities on each RB. Then $\frac{|\mathcal{B}|}{2} \cdot 2|\widetilde{\mathcal{K}}^b| \cdot (2+1)^{|\widetilde{\mathcal{K}}^b|}$ threads compute the $\text{SINR}_k^{b,f}$ for up to $|\mathcal{B}| \cdot 2|\widetilde{\mathcal{K}}^b|$ data streams for all $(2+1)^{|\widetilde{\mathcal{K}}^b|}$ possibilities. After that, a kernel with $N_{\text{P1}}\frac{|\mathcal{B}|}{2}$ blocks and $(2+1)^{|\widetilde{\mathcal{K}}^b|}$ threads per block solves OPT-P3A, where one block finds the optimal solution to one OPT-P3A problem. To solve problem OPT-P3B, the data communication between GPU cards is needed. First, a kernel with $N_{\text{P1}}|\mathcal{K}|$ blocks and $\frac{|\mathcal{B}|}{2}$ threads per block is applied on each GPU card, where one block finds the best solution to one OPT-P3B problem w.r.t. the first or second half of RBs. Then the first GPU card calculates the optimal solution to OPT-P3B by accessing the second GPU's global memory. The first GPU card sends a copy of results to the second GPU card for further operations. As described in Section 5.4.2, we now have N_{P1} sets of feasible solutions $\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_{N_{\text{P1}}}$.

• **Step (v): Execute Phase 4.** First, each GPU card launches a kernel with $N_{\text{P1}}\frac{|\mathcal{B}|}{2}$ to obtain the sum of PF values over the first or second half of RBs. After the results are

collected to the first GPU card, a kernel with N_{P1} threads is launched only on the first GPU card to find the best solution $\mathcal{S}^*$ that offers the highest PF objective value, among the N_{P1} candidates of $\mathcal{S}_i$'s. Parallel reduction technique is used to accelerate the comparison process.

- **Step (vi): Transfer data from GPU to host device.** Once the final solution is determined, we transfer the scheduling solution $\mathcal{S}^*$ and corresponding precoding matrices from GPU devices to host device. The data transfer time will be counted toward mCore+'s total execution time.

5.6 Experimental Results

In this section, we validate the performance of mCore+ based on our implementation in Section 5.5.

5.6.1 Settings

We consider a DL scheduling problem in a 5G NR cellular environment, where one BS is serving a number of users. The number of users $|\mathcal{K}|$ is chosen from $\{50, 100\}$, the number of RBs is $|\mathcal{B}|$ is 100, and we have 29 different MCS levels in $\mathcal{M}$. The BS is equipped with up to 12 antennas, and each user has up to 4 antennas. For the wireless channel $\mathbf{H}_k^b(t)$, it is composed by large scale fading L_k and small scale fading $\bar{\mathbf{H}}_k^b(t)$, i.e., $\mathbf{H}_k^b(t) = L_k^{-1}\bar{\mathbf{H}}_k^b(t)$. Large scale fading L_k is uniformly chosen from [0dB 6dB] to reflect different user locations, which does not vary in frequencies and TTIs. The small scale fading $\bar{\mathbf{H}}_k^b(t)$ is modeled through Rayleigh fading [9], which differs among every user, RB and time slot. The SNR (i.e., P_{T}/n_0^2) is set to 10 dB. For the parameters of our algorithm, the number of possible MCS assignments N_{P1} (see Phase 1 in Section 5.4.2) is set to 256, the numbers of promising users

on each RB, $|\widetilde{\mathcal{K}}^b|$ and $|\mathcal{K}^b|$ (see Phase 2 in Section 5.4.2), are set to 10 and 4, respectively.

We compare our design with the following state-of-the-art PF schedulers: 1) SU-PF (SU-MIMO PF scheduler from [135]), 2) MU-PF (MU-MIMO PF scheduler from [131]), and 3) Unified-PF from [132]. None of the existing works developed their system models as comprehensive as ours. To compare with those algorithms, we made necessary extensions based on the nature of those algorithms. Specifically, we extended SU-PF and MU-PF to support MCS scheduling following their idea of choosing 1 or 2 data streams (diversity mode or multiplexing mode) for each user. We extended Unified-PF by allowing 2-user MU-MIMO based on how they iteratively select MCS and RBs.

All the algorithms are running on the same machine as we described in Section 5.5. mCore+ is run by CUDA platform while other algorithms are implemented on MATLAB. For any of the algorithms used for comparison, we terminate the algorithm once the execution time exceeds 5 hours. Therefore, for any partial curves, the last points of the curves represent the cutoff of 5-hour running time.

5.6.2 Case Studies

In this section, we use two network settings as case studies to validate mCore+'s timing performance and throughput performance. We consider the following two typical settings: (a) $|\mathcal{B}| = 100, |\mathcal{K}| = 50, N_{\mathrm{T}} = 8, N_{\mathrm{R}} = 2$; (b) $|\mathcal{B}| = 100, |\mathcal{K}| = 100, N_{\mathrm{T}} = 12, N_{\mathrm{R}} = 4$.

Results for network setting (a) We first examine the performance of mCore+ under the setting (a) $|\mathcal{B}| = 100, |\mathcal{K}| = 50, N_{\mathrm{T}} = 8, N_{\mathrm{R}} = 2$. In Fig. 5.7, we compare the execution time of mCore+ and the other three schedulers for 200 consecutive TTIs. The results show that mCore+ can offer the scheduling solution within 500 μs across all TTIs. The average running time is 411 μs. This demonstrates that mCore+ can meet the timing requirement of

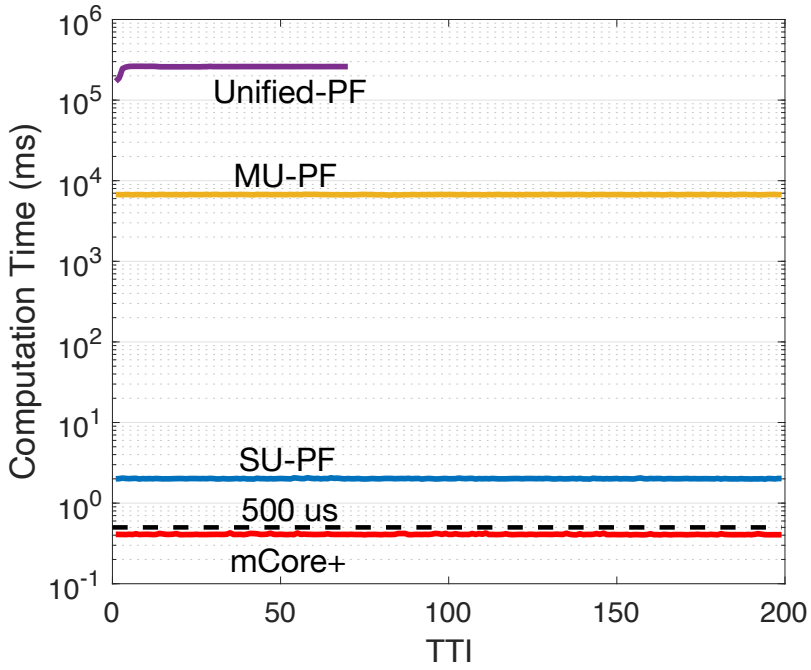


Figure 5.7: Timing performance comparison under different algorithms for setting (a) $|\mathcal{B}| = 100, |\mathcal{K}| = 50, N_{\mathrm{T}} = 8, N_{\mathrm{R}} = 2$.

5G numerology 1 under the setting (a). On the other hand, the average computation time of SU-PF, MU-PF, and Unified-PF is 2.5 ms, $\sim 6 \times 10^3$ ms, and $\sim 2 \times 10^5$ ms, respectively, which are far beyond 5G's sub-ms real-time requirement.

In Figs. 5.8 and 5.9, we evaluate mCore+'s throughput performance. We consider two important performance metrics for PF schedulers, including the PF objective $\sum_{k\in\mathcal{K}} \log_2(\tilde{R}_k(t))$, and the network throughput $\sum_{k\in\mathcal{K}} \tilde{R}_k(t)$. Figs. 5.8 and 5.9 show the achieved PF objective value and network throughput under different algorithms for 200 consecutive TTIs, respectively. The results suggest that mCore+ can obtain better or comparable PF values compared to other state-of-the-art algorithms. As expected, the algorithms that support MU-MIMO transmission (mCore+, MU-PF and Unified-PF) achieve better throughput performance than the algorithm SU-PF that only supports SU-MIMO scheduling.

Results for network setting (b) Now we evaluate mCore+'s performance under the

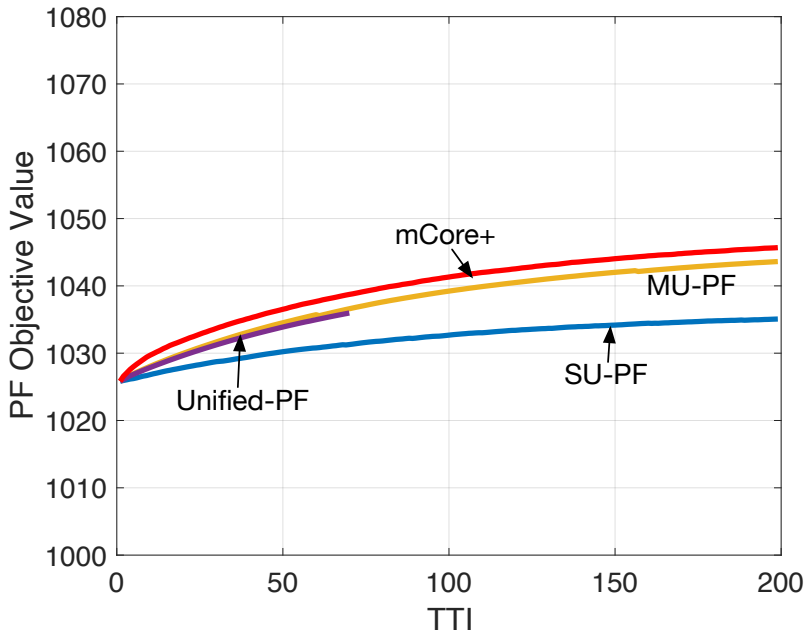


Figure 5.8: Achieved PF objective value under different algorithms for setting (a) $|\mathcal{B}| = 100, |\mathcal{K}| = 50, N_{\mathrm{T}} = 8, N_{\mathrm{R}} = 2$.

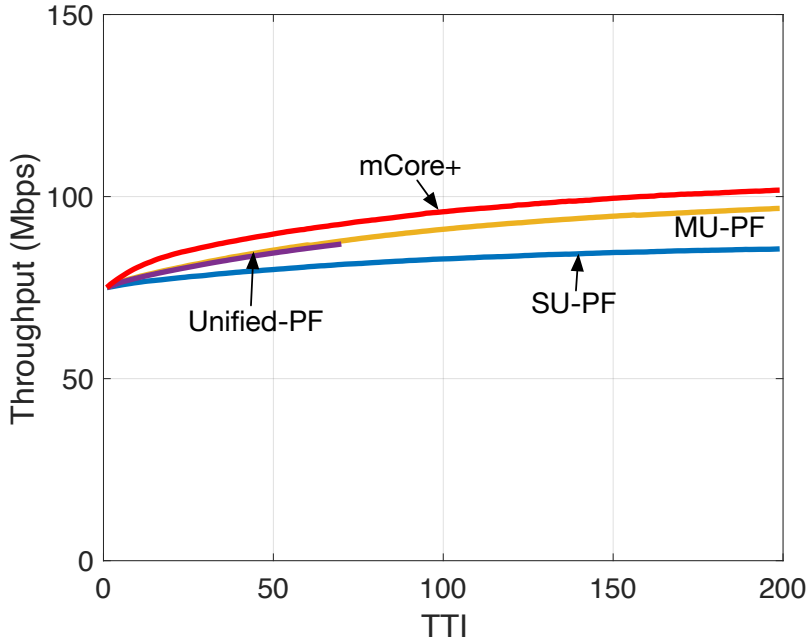


Figure 5.9: Network throughput under different algorithms for setting (a) $|\mathcal{B}| = 100, |\mathcal{K}| = 50, N_{\mathrm{T}} = 8, N_{\mathrm{R}} = 2$.

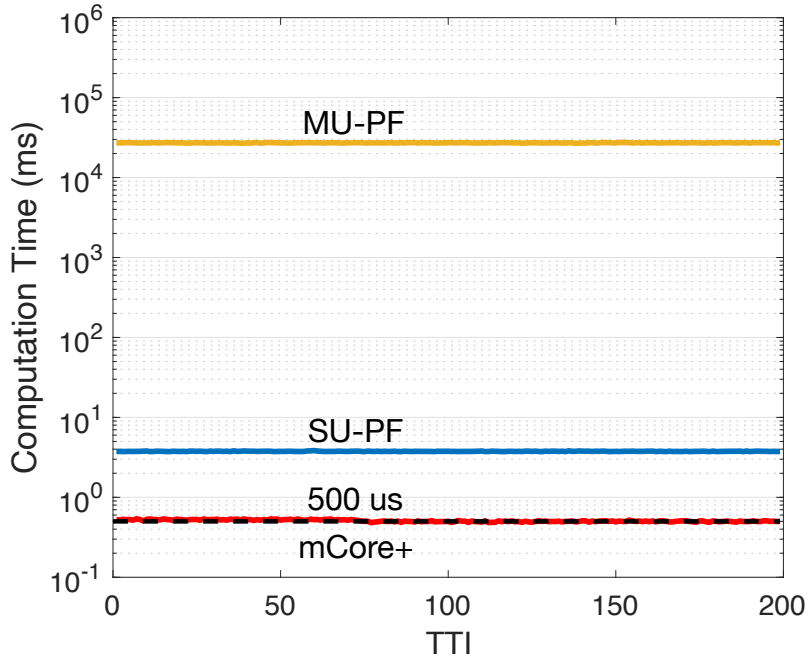


Figure 5.10: Timing performance comparison under different algorithms setting setting (b) $|\mathcal{B}| = 100, |\mathcal{K}| = 100, N_{\mathrm{T}} = 12, N_{\mathrm{R}} = 4$.

setting (b) $|\mathcal{B}| = 100, |\mathcal{K}| = 100, N_{\mathrm{T}} = 12, N_{\mathrm{R}} = 4$. Fig. 5.10 shows the timing performance under different algorithms for 200 consecutive TTIs. As shown in Fig. 5.10, the execution time of mCore+ is $\sim$500 μs. In contrast, the average computation time of SU-PF and MU-PF is $\sim$4 ms and $\sim 3 \times 10^4$ ms, respectively. Unified-PF requires $> 10^6$ ms to find a solution under this setting. Its computation time is too large to fit in the scale of the figure.

In Figs. 5.11 and 5.12, we show the achieved PF objective value and network throughput under different algorithms. As shown in Figs. 5.11 and 5.12, mCore+ can achieve the highest PF values and highest network throughput compared to other algorithms. Compared with the setting (a) (where the number of users and antennas are smaller), the performance gap between mCore+ and other algorithms becomes larger. This is because their algorithms can only schedule up to 2 streams per RB, while mCore+ can better take advantage of the spatial diversity brought by many antennas, thanks to the large-scale parallel design to support up

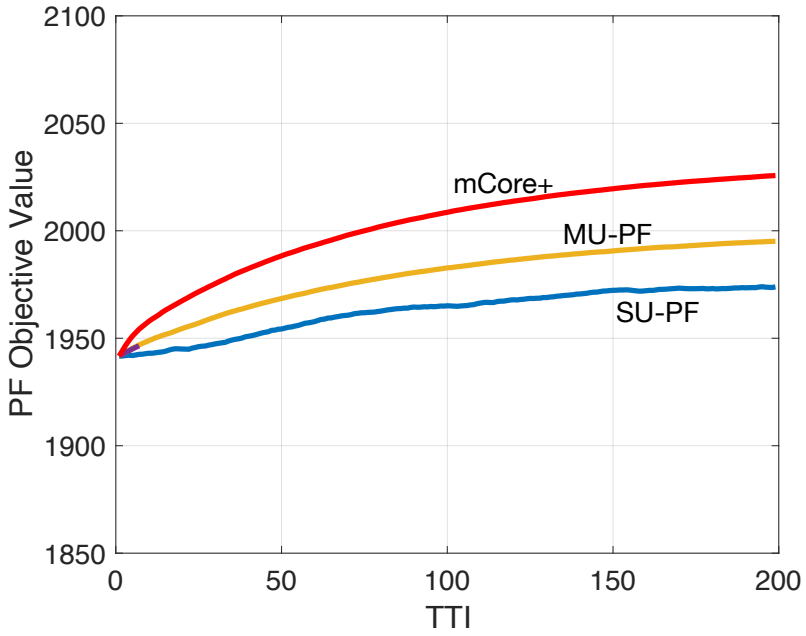


Figure 5.11: Achieved PF objective value under different algorithms for setting (b) $|\mathcal{B}| = 100, |\mathcal{K}| = 100, N_{\mathrm{T}} = 12, N_{\mathrm{R}} = 4$.

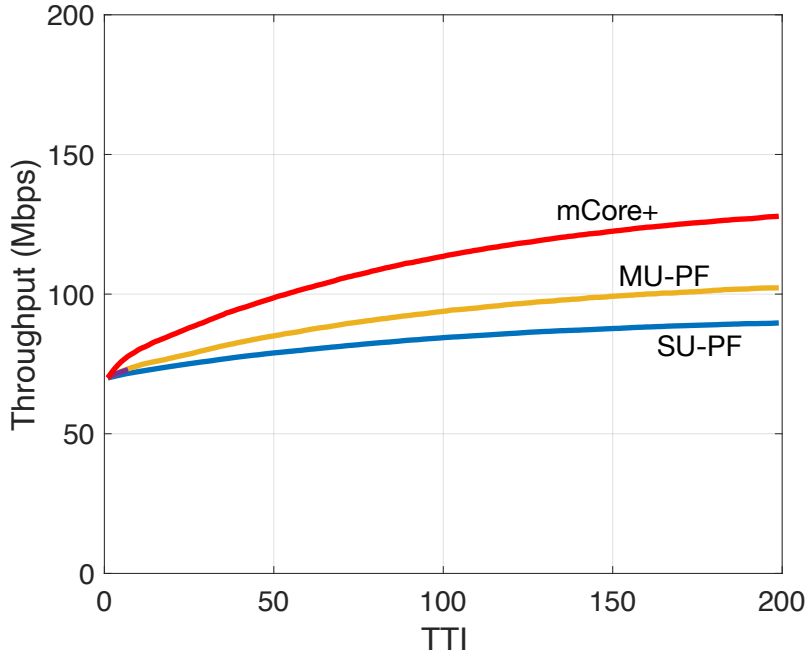


Figure 5.12: Network throughput under different algorithms for setting (b) $|\mathcal{B}| = 100, |\mathcal{K}| = 100, N_{\mathrm{T}} = 12, N_{\mathrm{R}} = 4$.

to 4-user MU-MIMO transmission. SU-PF achieves the least PF values as the algorithm only supports SU-MIMO scheduling.

5.6.3 Varying Network Parameters

In this section, we evaluate the behavior of mCore+ by varying different network parameters such as the number of RBs, users, and antennas.

In Fig. 5.13, we vary the number of RBs $|\mathcal{B}|$ to study its impact on mCore+'s execution time. We vary $|\mathcal{B}|$ from 20 to 100. The number of users is $|\mathcal{K}| = 100$. Fig. 5.13(a) shows mCore+'s execution time (mean, max and min values over 200 consecutive TTIs) as a function of RB numbers when $N_{\mathrm{T}} = 8, N_{\mathrm{R}} = 2$. The results demonstrate that mCore+'s execution time is well below 500 μs under all cases. The mean values are 277 μs, 333 μs, 393 μs, 411 μs, and 461 μs when $|\mathcal{B}|$ is 20, 40, 60, 80, and 100, respectively. In Fig. 5.13(b), we show mCore+'s execution time when $N_{\mathrm{T}} = 12, N_{\mathrm{R}} = 4$. Fig. 5.13(b) shows the mean values are 305 μs, 365 μs, 422 μs, 459 μs, and 509 μs when $|\mathcal{B}|$ is 20, 40, 60, 80, and 100, respectively. The execution time is slightly higher than the case when $N_{\mathrm{T}} = 8, N_{\mathrm{R}} = 2$, as the matrix operations become more intensive when the numbers of antennas are larger. Across 200 consecutive TTIs, the execution time is well below 500 μs for up to 80 RBs, and it is around 500 μs when $|\mathcal{B}| = 100$.

Next, we study the timing performance as a function of the number of users $|\mathcal{K}|$. We vary $|\mathcal{K}|$ from 20 to 100. The number of RBs $|\mathcal{B}|$ is 100. Fig. 5.14(a) shows mCore+'s average execution time with the maximum and minimum values over 200 consecutive TTIs when $N_{\mathrm{T}} = 8, N_{\mathrm{R}} = 2$. The results show that mCore+ is able to find the solution within 500 μs under all cases. We note that although the execution time increases with the number of users, the rate of increase is fairly slow as the number of users increases. This is because

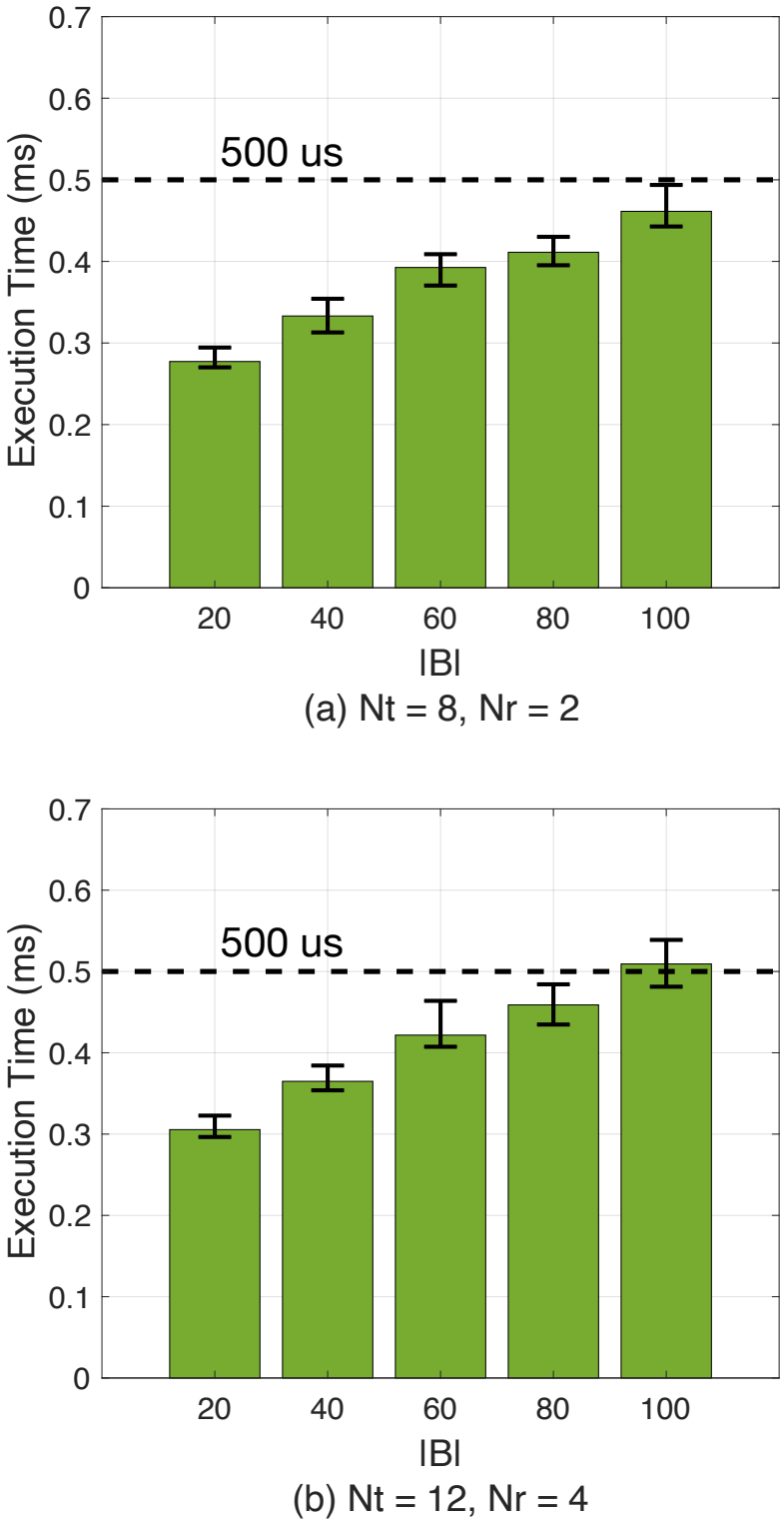


Figure 5.13: mCore+'s total execution time (mean, max and min values over 200 consecutive TTIs) as a function of the number of RBs. (a) $N_{\mathrm{T}} = 8, N_{\mathrm{R}} = 2$, (b) $N_{\mathrm{T}} = 12, N_{\mathrm{R}} = 4$.

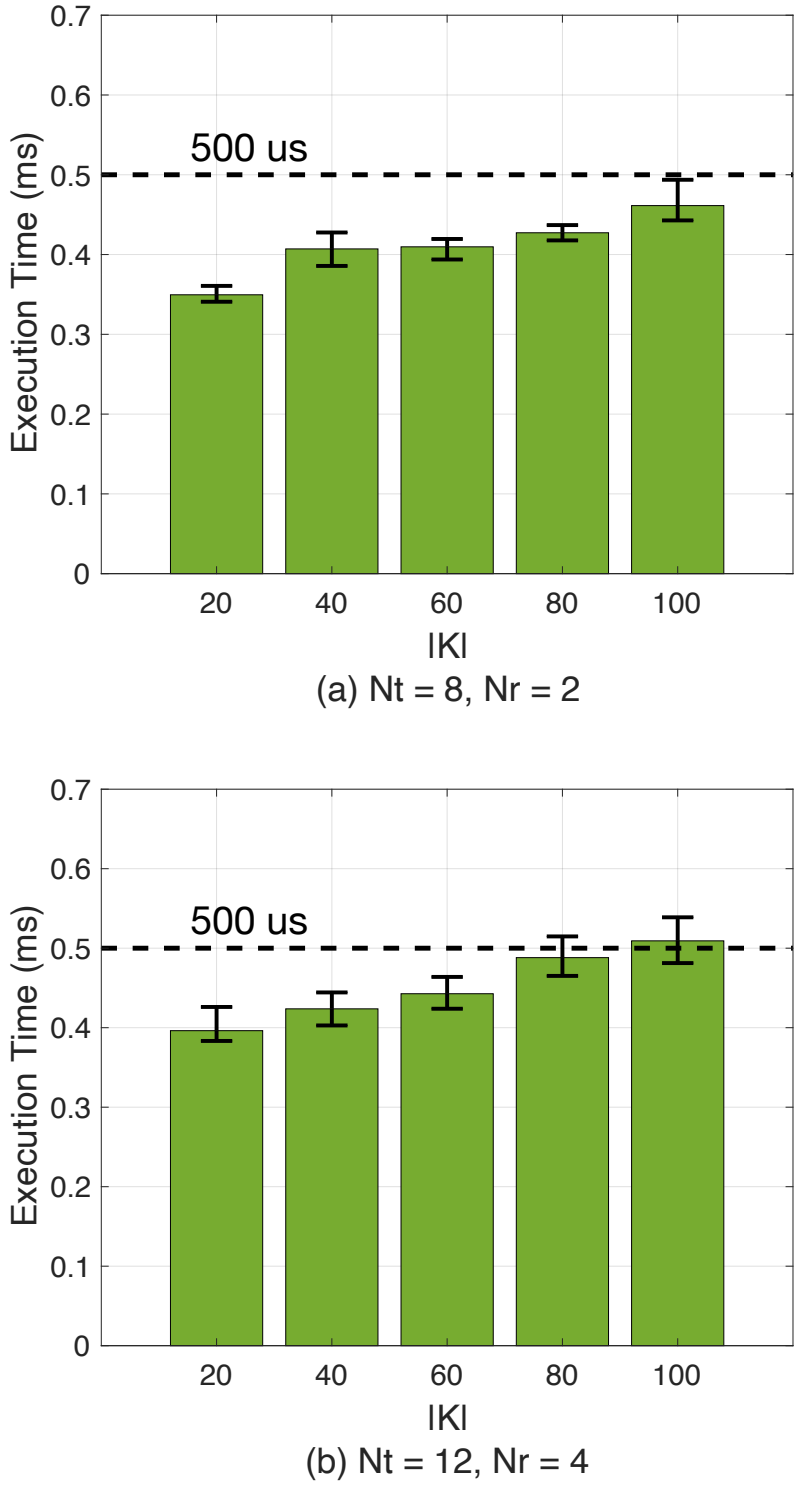


Figure 5.14: mCore+'s total execution time (mean, max and min values over 200 consecutive TTIs) as a function of the number of users. (a) $N_{\mathrm{T}} = 8, N_{\mathrm{R}} = 2$, (b) $N_{\mathrm{T}} = 12, N_{\mathrm{R}} = 4$.

that mCore+ can identify a small but most promising subset of users to form MU-MIMO transmission, and therefore the time-consuming operations (such as calculating beamforming matrices and SINR) only need to be performed for a small set of users. When $N_{\mathrm{T}} = 12, N_{\mathrm{R}} = 4$, Fig. 5.14(b) indicates that mCore+'s execution time is lower than 500 μs for up to 60 users, and it is around 500 μs for up to 100 users.

In Fig. 5.15, we present the network throughput performance $\sum_{k\in\mathcal{K}} \tilde{R}_k(t)$ as a function of the number of antennas N_{T} at the BS under different algorithms. N_{T} is varying from 6 to 12, and we consider two different settings: (a) (a) $N_{\mathrm{R}} = 2, |\mathcal{K}| = 50, |\mathcal{B}| = 100$, (b) $N_{\mathrm{R}} = 4, |\mathcal{K}| = 100, |\mathcal{B}| = 100$. We didn't include the performance of Unified-PF in the figure, as it is not able to converge to its long-term average throughput after 5 hours of running the algorithms. The results in in Fig. 5.15 show that under both settings (a) and (b), the network throughput just slightly increases with N_{T} under MU-PF and SU-PF. However, mCore+ can better take advantage of additional antennas to achieve much higher throughput. This is because mCore+ utilizes the knowledge of channel correlations among users and supports up to 4-user MU-MIMO transmission, and therefore it can better exploit the spatial diversity offered by many antennas compared with other algorithms.

In summary, the experimental results show that mCore+ is the only algorithm that can find the scheduling solution in ~500 μs under all tested cases (for up to 100 RBs, 100 users, 4 MIMO, and 4 users per RB). Further, the throughput performance achieved by mCore+ is better or comparable to other algorithms.

5.7 Chapter Summary

This paper presents the design and implementation of mCore+. mCore+ is the first MU-MIMO scheduler for 5G NR that achieves ~500-μs real-time scheduling. By the design of

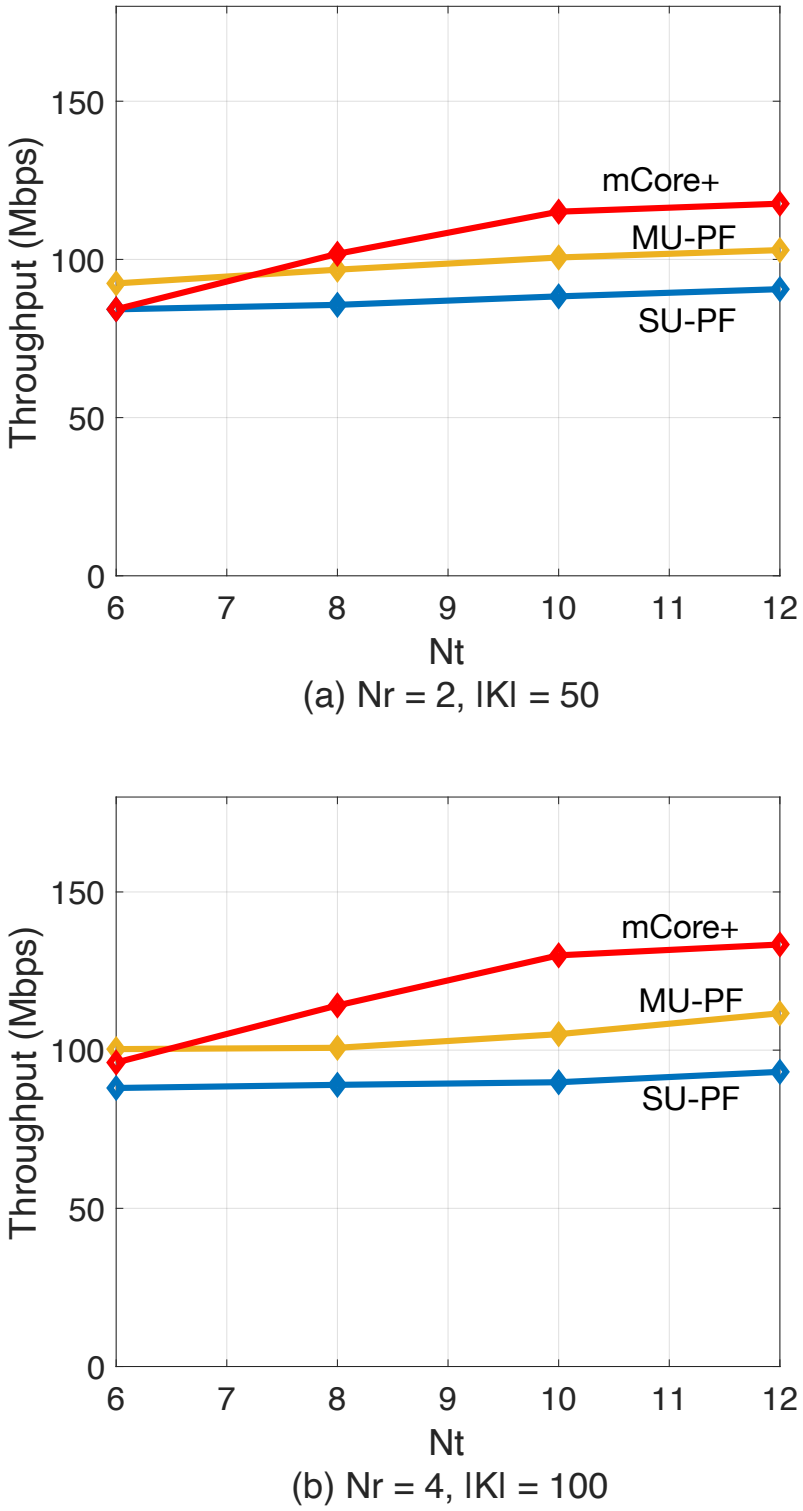


Figure 5.15: Comparison of throughput achieved by different algorithms as a function of the number of antennas at the BS. (a) $N_{\mathrm{R}} = 2, |\mathcal{K}| = 50$, (b) $N_{\mathrm{R}} = 4, |\mathcal{K}| = 100$.

mCore+, RB allocation, number of data stream determination and MCS assignment are jointly optimized. In particular, multiple users may share the same RB resources by our design. To address the real-time challenge, mCore+ employs a multi-phase optimization, with each phase exploiting large-scale parallelism. The search space is reduced through the knowledge of channel conditions and user correlations. We implemented mCore+ on a COTS GPU platform to examine its performance. Through extensive experiments, we show that mCore+ can obtain a scheduling solution with $\sim$500 μs for up to 100 RBs, 100 users, 4×12 MIMO systems. Moreover, mCore+ is able to offer a better or comparable throughput performance compared with other state-of-the-art algorithms.

Chapter 6

A Sub-millisecond Scheduler for Multi-Cell MIMO Networks under C-RAN Architecture

6.1 Introduction

To increase spectrum efficiency and reduce the operation cost for the next-generation cellular systems, the so-called "C-RAN architecture" has been explored [154, 155, 156, 157, 158]. As shown in Fig. 6.1, C-RAN is based on a centralized architecture—a baseband unit (BBU) pool located at a centralized site serving several remote radio heads (RRHs). The BBU pool is responsible for the data processing at upper PHY layer (i.e., baseband signal processing), MAC layer and network layer for all RRHs under its coverage. Each RRH is equipped with multiple antennas—they are responsible for lower PHY layer signal processing, i.e., performing the radio frequency functions and emitting the signals.[1] Connection between the BBU pool and the remote RRHs is through high-capacity, low-latency optical fronthauls. Per 5G specifications, the maximum allowed end-to-end one-way latency of functional spilt between upper PHY and lower PHY is 250 μs [160], which is sufficiently small compared

[1]The point of separation between BBU pool and RRHs in functionalities may vary and depends on different options in 5G [159].

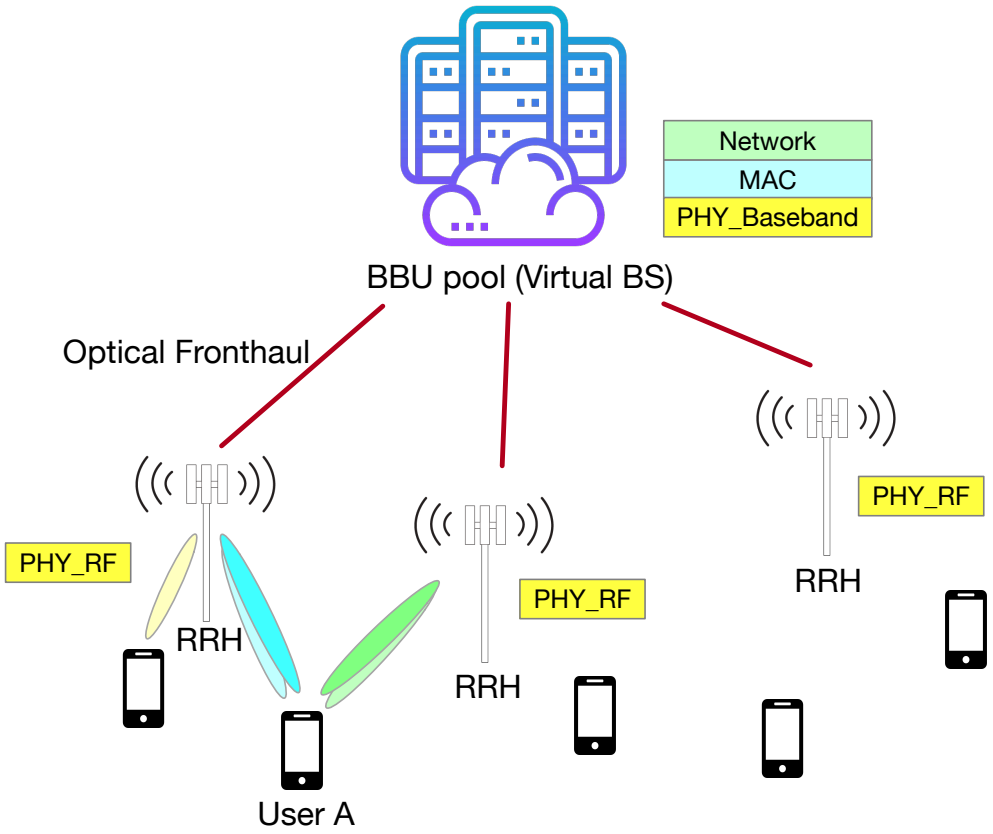


Figure 6.1: Under C-RAN architecture, a centralized BBU pool is scheduling resources for users covered by a set of RRHs. A user can receive its data from one or multiple RRHs at the same time.

with the channel coherence time for most communication scenarios. Therefore, the real-time scheduling can be performed at the center BBU pool.

A goal of C-RAN is to support *joint transmission*—a coordinated beamforming scheme that can significantly improve spectrum efficiency [154, 155, 156, 160, 161]. By "joint transmission", we mean a user can receive its data from multiple RRHs simultaneously (see user A in Fig. 6.1). As future networks become smaller and denser, there is opportunity from joint transmission. Likewise, effective management of inter-cell interference becomes even more important. Thanks to centralized architecture in C-RAN, the virtual BS (i.e., BBU pool) can ease the sharing of signaling, traffic data and channel state information (CSI) that

are needed for joint transmission from different cells.

However, significant challenges remain in the design of a C-RAN scheduler for joint transmission. To concretize our discussion, let's consider a downlink scheduling problem in C-RAN. We face the following critical challenges.

- First, the centralized scheduler must allocate a number of RBs and decide the beamforming matrices at each RRH. Under joint transmission, the same RBs but from neighboring RRHs may be transmitted to the same user. As such, in each TTI, solutions to RB allocation and beamforming matrices must be done jointly across different cells at the virtual BS.

- Second, the scheduler must assign MCS and number of data streams for each user. Under 5G NR [4], a user's receiver must have the same MCS level and number of data streams across all RBs that are allocated to her (even they come from different RRHs).

- Third, to achieve high throughput, MU-MIMO transmission should be used under C-RAN. Thus, an RB may be allocated to multiple users.

- In addition to the above challenges on the scheduler side, we also have a stringent timing requirement—the scheduler must find its scheduling solution within one TTI. Under 5G numerology 0, one TTI is 1 ms. Then the C-RAN scheduling solution must be found within 1 ms to be useful. To support ultra-low latency applications, an even shorter TTI may be needed (e.g., 500 μs under numerology 1).

To date, there has been a number of studies on scheduling or beamforming problems under C-RAN (see, e.g., [162, 163, 164, 165, 166, 167, 168, 169]). However, none of these studies considered real-time requirement for their proposed solutions. For example, the authors in [162, 163, 164, 165, 166] designed coordinated beamforming schemes for C-RAN, but these

designs are based on iterative optimization, which requires excessive amount of computation time. Further, none of these existing works jointly optimizes RB allocation, MCS assignment and beamforming matrices for a multi-cell system. For the studies in [162, 163, 164, 165, 166, 167, 168, 169], although coordinated beamforming is considered, either RB allocation or MCS assignment is missing. For example, the design in [163] is a representative research work that considers coordinated beamforming without the consideration of RB and MCS allocation. We made an experiment to run the algorithm in [163] (for a single RB without MCS selection) on Matlab platform. The execution time is ∼70 seconds per TTI on average. Such designs relying on iterative optimization cannot be used in real-time.

In this chapter, we present the design and implementation of $\mathbf{M}^3$—the first sub-Millisecond scheduler for Multi-cell MIMO networks under 5G C-RAN architecture. $\mathbf{M}^3$ is capable of finding a solution to RB allocation, MCS selection, data stream assignment, as well as the precoding matrices, in real-time for each TTI (at most 1 ms). We tackle the crucial timing problem through a novel parallel design on a commercial off-the-shelf (COTS) GPU platform. Our main contributions can be summarized as follows:

- $\mathbf{M}^3$ is the first C-RAN scheduler that can meet the 1 ms real-time requirement. The success of $\mathbf{M}^3$ is built upon a judicious parallel design and validated on a COTS GPU. The design of $\mathbf{M}^3$ is developed in accordance with the time-frequency resource structure defined by 5G NR, and it is applicable to centralized multi-cell systems.

- $\mathbf{M}^3$ exploits independency and parallelism through a multi-pipeline design. Specifically, $\mathbf{M}^3$ first divides all users into two groups: non-edge users and cell-edge users by leveraging their channel properties. Then $\mathbf{M}^3$ performs two independent parallel pipelines, with one pipeline focusing on a sequence of operations for cell-edge users (to explore joint transmission) and the other pipeline for non-edge users (to explore

MU-MIMO transmission). After both pipelines complete their operations in parallel, $\mathbf{M}^3$ determines the final solutions for all users.

- $\mathbf{M}^3$ achieves large-scale parallelism in addition to the multi-pipeline structure. Throughout our design of $\mathbf{M}^3$, the exploration of parallel computing is carried out by leveraging GPU's capability. For instance, within each pipeline, most operations are purposefully designed to be independent among RRHs and RBs. By taking advantage of massive parallel computation all the way, our design can reduce the computation time dramatically.

- $\mathbf{M}^3$ is implemented on a COTS GPU platform—Nvidia DGX Station. We conduct extensive experiments to verify $\mathbf{M}^3$'s timing performance as well as its throughput performance. Our experimental results show that $\mathbf{M}^3$ is able to offer the scheduling solution within 500 μs for a C-RAN system with 7 RRHs, 100 users, 100 RBs, and 2×8 MIMO. For a 2×12 MIMO system, $\mathbf{M}^3$ can also meet the 1 ms requirement under all tested cases. In the mean time, $\mathbf{M}^3$ achieves $\sim$40% throughput gain under joint transmission.

6.2 System Model

We consider a downlink (DL) scheduling problem under C-RAN architecture. As shown in Fig. 6.1, a centralized BBU pool is connected to a set $\mathcal{L}$ of RRHs, which serve a set $\mathcal{K}$ of users. Each RRH is equipped with N_{T} antennas while each user is equipped with N_{R} antennas and $N_{\mathrm{T}} > N_{\mathrm{R}}$. Table 6.1 gives the key notations that we use in this chapter.

Fig. 6.2 illustrates our scheduling problem, which we elaborate mathematically in the rest of this section.

Table 6.1: Notations in Chapter 6

Symbol	Definition
$\mathcal{B}$	A set of RBs to be allocated in a time slot
$\mathbf{F}_{l,k}^{b}(t)$	Precoding matrix for user k used by RRH l on RB b in TTI t
$\mathbf{H}_{l,k}^{b}$	Channel matrix from RRH l to user k on RB b
$\mathcal{K}$	The set of users from all RRHs
$\mathcal{K}^{\not\mathrm{E}}$	The subset of non-edge users in $\mathcal{K}$
$\mathcal{K}^{\mathrm{E}}$	The subset of cell-edge users in $\mathcal{K}$
$\mathcal{L}$	A set of RRHs
$\mathcal{M}$	A set of MCSs
N_{R}	Number of antennas at each user
N_{T}	Number of antennas at an RRH
$r_{k}^{b,f,m}(t)$	The instantaneous achievable data rate of user k's f-th data stream on RB b with MCS m in TTI t
$R_k(t)$	The aggregate data rate of user k in TTI t
$\tilde{R}_k(t)$	The exponentially smoothed average data rate of user k up to TTI t
$x_{l,k}^{b}(t)$	A binary variable indicating whether or not RRH l is transmitting data to user k on RB b in TTI t
$\hat{x}_{k}^{b}(t)$	A binary number indicating whether or not user k is receiving data from at least one RRH on RB b in TTI t
$y_k(t)$	Number of data streams for user k in TTI t
$z_{k}^{m}(t)$	A binary variable indicating whether or not MCS m is used for user k in TTI t

User Association and RB Allocation Consider a frequency reuse system, where a wide frequency band is reused at every RRH. Per 3GPP, the frequency band is divided into a set $\mathcal{B}$ of RBs. In each TTI, $\mathcal{B}$ is available at each RRH for DL transmission. Under C-RAN architecture, a user can receive its signals from one or multiple RRHs on an RB. Under joint transmission [156, 161], when user k is receiving its signals on RB b from multiple RRHs, the user data $\boldsymbol{s}_k^b$ is identical from these RRHs. Denote $x_{l,k}^b(t) \in \{0,1\}$ as a binary variable indicating whether or not RRH l is transmitting data to user k on RB b in TTI t, i.e.,

$$x_{l,k}^b(t) = \begin{cases} 1, & \text{if RRH } l \text{ is transmitting data} \\ & \text{to user } k \text{ on RB } b \text{ in TTI } t, \\ 0, & \text{otherwise.} \end{cases}$$

Denote $\mathcal{L}_k^b(t)$ as the set of RRHs that are transmitting data to user k on RB b in TTI t, i.e., $\mathcal{L}_k^b(t) = \{l \in \mathcal{L} | x_{l,k}^b(t) = 1\}$. Denote $\hat{x}_k^b(t) \in \{0,1\}$ as a binary number indicating whether or not user k is receiving data from at least one RRH on RB b in TTI t, i.e.,

$$\hat{x}_k^b(t) = \begin{cases} 1, & \text{if } \left|\mathcal{L}_k^b(t)\right| > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{6.1}$$

Also note that under MU-MIMO, an RRH can transmit to multiple users on the same RB. As the maximum number of users scheduled on an RB cannot exceed the number of antennas at the RRH, we have the following MU-MIMO constraint for $x_{l,k}^b(t)$:

$$\sum_{k \in \mathcal{K}} x_{l,k}^b(t) \leq N_{\mathrm{T}}. \qquad (b \in \mathcal{B}, l \in \mathcal{L}) \tag{6.2}$$

Number of Data Streams A user may have multiple data streams on each RB that

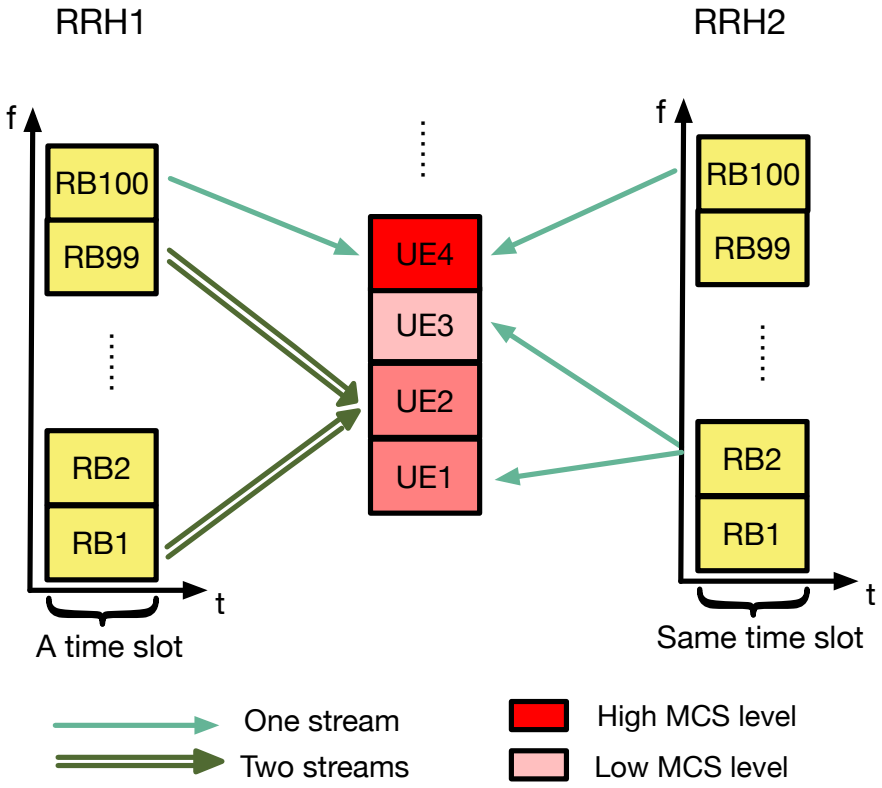


Figure 6.2: Within each time slot, the virtual BS jointly determines RB allocation, number of data streams, and MCS assignment for all users under all RRHs.

is allocated to her. But when a user receives its data streams on multiple RBs, then the number of data streams must be identical across all these RBs [128].

Denote $y_k(t)$ as the number of data streams for user k in TTI t (which is the same across all allocated RBs). As $y_k(t)$ cannot be greater than the number of receive antennas, we have

$$y_k(t) \leq N_{\mathrm{R}}. \qquad (k \in \mathcal{K}) \tag{6.3}$$

Also, at each RRH, the total number of data streams on each RB for MU-MIMO transmission cannot exceed the number of its antennas. We have

$$\sum_{k \in \mathcal{K}} x_{l,k}^b(t) y_k(t) \leq N_{\mathrm{T}}. \qquad (b \in \mathcal{B}, l \in \mathcal{L}) \tag{6.4}$$

Achieved SINR at Users Each RRH applies precoders to support joint transmission and/or MU-MIMO transmission. Let $\mathbf{F}_{l,k}^b(t)$ be an $N_{\mathrm{T}} \times x_{l,k}^b(t) y_k(t)$ precoding matrix for user k used by RRH l on RB b. Under the power constraint at an RRH, we have $\sum_{k \in \mathcal{K}} ||\mathbf{F}_{l,k}^b(t)||_F^2 \leq P_{\mathrm{T}}$ for all $l \in \mathcal{L}$, where P_{T} is the total power (per RB) at the RRH and $||\cdot||_F$ denotes the Frobenius norm. Then the received signal of user k on RB b is given by

$$\boldsymbol{c}_k^b = \sum_{l \in \mathcal{L}_k^b} \mathbf{H}_{l,k}^b \mathbf{F}_{l,k}^b \boldsymbol{s}_k^b + \sum_{l \in \mathcal{L}} \sum_{i \in \mathcal{K}}^{i \neq k} \mathbf{H}_{l,k}^b \mathbf{F}_{l,i}^b \boldsymbol{s}_i^b + \boldsymbol{n}_k^b,$$

where $\mathbf{H}_{l,k}^b \in \mathbb{C}^{N_{\mathrm{R}} \times N_{\mathrm{T}}}$ is the channel matrix from RRH l to user k on RB b, $\boldsymbol{n}_k^b$ is the $N_{\mathrm{R}} \times 1$ vector of i.i.d $\mathcal{CN}(0, n_0^2)$ additive complex Gaussian noise, $\boldsymbol{s}_k^b$ is the signal vector, and we omit the time-dependent notation (t) for matrices for brevity.

Then each user applies an $N_{\mathrm{R}} \times y_k(t)$ combiner $\mathbf{W}_k^b$ for the received signals. After this

combiner, we have the following signal for user k on RB b:

$$\begin{aligned}\tilde{\boldsymbol{c}}_k^b &= \mathbf{W}_k^{b\dagger}\boldsymbol{c}_k^b \\ &= \underbrace{\sum_{l\in\mathcal{L}_k^b}\mathbf{W}_k^{b\dagger}\mathbf{H}_{l,k}^b\mathbf{F}_{l,k}^b\boldsymbol{s}_k^b}_{\text{desired signal}} + \underbrace{\sum_{l\in\mathcal{L}}\sum_{i\in\mathcal{K}}^{i\neq k}\mathbf{W}_k^{b\dagger}\mathbf{H}_{l,k}^b\mathbf{F}_{l,i}^b\boldsymbol{s}_i^b}_{\text{interference}} + \mathbf{W}_k^{b\dagger}\boldsymbol{n}_k^b,\end{aligned} \tag{6.5}$$

where $(\cdot)^\dagger$ denotes the conjugate transpose of a matrix. Different beamforming schemes can be applied based on $\mathbf{H}_{l,k}^b$. In this chapter, we apply MMSE precoding scheme at the RRH side (based on $\mathbf{H}_{l,k}^b$) with equal power allocation for each data stream and MMSE combiner at the user side (based on $\mathbf{H}_{l,k}^b\mathbf{F}_{l,k}^b$).

For each $k \in \mathcal{K}$, the signal-to-interference-plus-noise ratio (SINR) of the f-th stream on RB b is then given by

$$\mathrm{SINR}_k^{b,f} = \frac{E_k^{b,f}}{\mathbf{Q}_k^{b,f} + n_0^2\mathbf{W}_k^{b,f\dagger}\mathbf{W}_k^{b,f}}, \tag{6.6}$$

where

$$\begin{aligned}E_k^{b,f} &= \left|\sum_{l\in\mathcal{L}_k^b}\mathbf{W}_k^{b,f\dagger}\mathbf{H}_{l,k}^b\mathbf{F}_{l,k}^{b,f}\right|^2 \\ \mathbf{Q}_k^b &= \sum_{i\in\mathcal{K}}\sum_{f'=1}^{y_i(t)}\left|\sum_{l\in\mathcal{L}}\mathbf{W}_k^{b,f\dagger}\mathbf{H}_{l,k}^b\mathbf{F}_{l,i}^{b,f'}\right|^2 - E_k^{b,f},\end{aligned}$$

and $(\cdot)_k^{b,f}$ is the f-th column of $(\cdot)_k^b$.

MCS Assignment In each TTI t, a set $\mathcal{M}$ of MCSs is available for data transmission for each user $k \in \mathcal{K}$. However, if a user is scheduled to receive data streams on multiple RBs, 3GPP requires that the user employs the same MCS across all data streams on all scheduled RBs [4]. Denote $z_k^m(t) \in \{0, 1\}$ as a binary variable indicating whether or not MCS $m \in \mathcal{M}$

is used by the virtual BS for user $k \in \mathcal{K}$ in TTI t, i.e.,

$$z_k^m(t) = \begin{cases} 1, & \text{if MCS } m \text{ is used for user } k \text{ in TTI } t, \\ 0, & \text{otherwise.} \end{cases} \tag{6.7}$$

To guarantee only one MCS is used across all scheduled RBs for user k, we have

$$\sum_{m \in \mathcal{M}} z_k^m = 1. \qquad (k \in \mathcal{K}) \tag{6.8}$$

Instantaneous Data Rate The instantaneous achievable data rate depends on the SINR of each stream and the selected MCS level. Specifically, with a higher MCS level m, the user data is encoded with a higher data rate r^m. However, to successfully decode the user data at a higher MCS level m, a higher level of SINR is required, or the data cannot be successfully decoded (i.e., a data rate of zero). Denote θ^m as the SINR threshold for successfully decoding the data with MCS m, and $r_k^{b,f,m}(t)$ as the instantaneous achievable data rate of user k's f-th data stream on RB b with MCS m in TTI t. Then we have

$$r_k^{b,f,m}(t) = \begin{cases} r^m, & \text{if } \mathrm{SINR}_k^{b,f} \geq \theta^m, \\ 0, & \text{otherwise.} \end{cases} \tag{6.9}$$
$$(f = 1, \cdots, y_k(t), k \in \mathcal{K}, b \in \mathcal{B}, m \in \mathcal{M})$$

where $\mathrm{SINR}_k^{b,f}$ is defined in Eq. (6.6).

The aggregate achievable data rate of user k in TTI t can be given by

$$R_k(t) = \sum_{b \in \mathcal{B}} \hat{x}_k^b(t) \sum_{f=1}^{y_k(t)} \sum_{m \in \mathcal{M}} z_k^m(t) r_k^{b,f,m}(t), \tag{6.10}$$

where $\hat{x}_k^b(t)$ is defined in Eq. (6.1), and we define $\sum_{f=1}^{y_k(t)}(\cdot) = 0$ if $y_k(t) = 0$.

Proportional Fair Metric Users that are far away from their RRH(s) may experience low SINR for a long period. Therefore, the consideration of fairness is essential for a C-RAN scheduler. A common scheduling objective is to maximize the PF metric. Specifically, a PF-oriented scheduler aims at maximizing the utility function $\sum_{k\in\mathcal{K}} \log\tilde{R}_k$, where $\tilde{R}_k$ is the long-term average data rate of user k.

For a time-slotted system, a widely used approach is to maximize the sum of normalized data rates in each time slot [95, 148, 169],

$$\sum_{k\in\mathcal{K}} \frac{R_k(t)}{\tilde{R}_k(t-1)}, \tag{6.11}$$

where the long-term average data rates $\tilde{R}_k(t-1)$ are updated using an exponentially weighted filter:

$$\tilde{R}_k(t-1) = \frac{T_c - 1}{T_c}\tilde{R}_k(t-2) + \frac{1}{T_c}R_k(t-1).$$

Problem Statement Our objectives are 1) to determine the users' RRH association and allocate RBs ($x_{l,k}^b(t)$'s), assign the number of data streams ($y_k(t)$'s) and MCSs ($z_k^m(t)$'s), as well as compute precoding matrices ($\mathbf{F}_{l,k}^b(t)$'s) for all users, such that the PF metric (6.11) is maximized; and 2) to ensure the scheduling solution can be found within each TTI (i.e., at most 1 ms) to meet 5G's timing requirement. This C-RAN scheduling problem can be written as follows.

OPT

$$\max \sum_{b\in\mathcal{B}}\sum_{k\in\mathcal{K}}\sum_{m\in\mathcal{M}}\sum_{f=1}^{y_k(t)}\frac{r_k^{b,f,m}(t)}{\tilde{R}_k(t-1)}\hat{x}_k^b(t)z_k^m(t),$$

s.t. User association and RB allocation constraint: (6.2);

Data stream allocation constraints: (6.4);

MCS assignemnt constraint: (6.8);

SINR and instantaneous data rate: (6.6), (6.9);

$x_{l,k}^b(t)\in\{0,1\}, y_k(t)\in\{0,1,\cdots,N_{\mathrm{R}}\}, z_k^m(t)\in\{0,1\}.$

In problem OPT, $x_{l,k}^b(t)$, $y_k(t)$ and $z_k^m(t)$ are decision variables, $\mathbf{F}_{l,k}^b(t)$, $\mathbf{W}_k^b(t)$, $\hat{x}_k^b(t)$, $r_k^{b,f,m}(t)$ and $\tilde{R}_k(t-1)$ are intermediate variables which can be determined with given $x_{l,k}^b(t)$, $y_k(t)$ and $z_k^m(t)$. The others are given constants.

Problem OPT is a nonlinear integer problem. The decision variables are tightly coupled together with extremely large search space. Further, the stringent timing requirement adds another level of challenge to design a solution for OPT.

6.3 M^3: Key Ideas and Road Map

Before we present the design blueprint of **M**^{3}, let's first offer some insights on the C-RAN scheduling problem. Different from traditional single-cell resource scheduling, C-RAN architecture has the potential to further improve PF objective through cooperative scheduling among neighboring cells. For cooperative scheduling, we first need to address the critical question of how to allocate RBs at each RRH for potential joint transmission. We have the

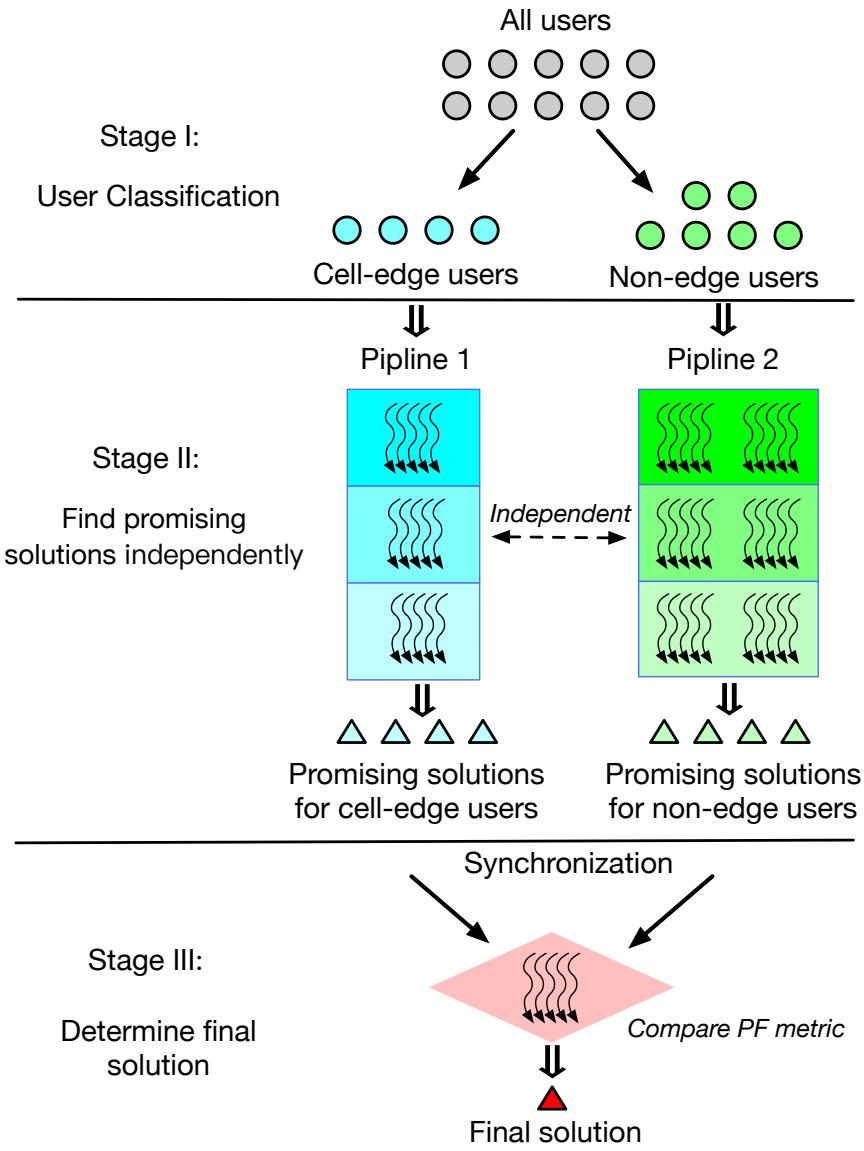


Figure 6.3: A flow chart for $\mathbf{M}^3$.

following two options:

- (i) Multiple RRHs use this RB to form joint transmission to a user;
- (ii) Each RRH uses this RB separately for different users.

There is a trade-off betwen the above two options. On the one hand, if an RB is used for joint transmission at multiple RRHs, potential interfering signals are effectively transformed into desired signals, which can significantly improve a receiver's data rate. On the other hand, joint transmission consumes more RB resources—a user that is supported by joint transmission requires all cooperative RRHs to use this RB for her. Without joint transmission, these RRHs may use this RB separately (and independently) to support more users.

$\mathbf{M}^3$ tackles the RB allocation problem with the following ideas. First, it is clear that not every user can benefit from joint transmission. Under a virtual BS, we have multiple adjacent RRHs, forming multiple adjacent cells in the system. Then only cell-edge users may benefit from receiving signals from multiple RRHs, while other (non-edge) users are better served solely by their own RRHs. Therefore, $\mathbf{M}^3$ reduces the search space by dividing all users into two groups: cell-edge users $\mathcal{K}^{\mathrm{E}}$ and non-edge users $\mathcal{K}^{\not{\mathrm{E}}}$. Only users in $\mathcal{K}^{\mathrm{E}}$ have the opportunity for joint transmission. This division should be based on large-scale path loss and can be determined based on long term measurements.

Second, based on above user division, $\mathbf{M}^3$ exploits parallelism to determine RB allocation, along with MCS and number of data streams, to problem OPT. Specifically, $\mathbf{M}^3$ employs two independent *parallel pipelines*, where one pipeline is a sequence of operations focusing on cell-edge users (to explore joint transmission) and the other pipeline is for non-edge users (to explore MU-MIMO transmission). To determine whether an RB b is better used for $\mathcal{K}^{\mathrm{E}}$ or $\mathcal{K}^{\not{\mathrm{E}}}$ in terms maximizing the PF metric, $\mathbf{M}^3$ will check both cases in parallel by

using those two pipelines. That is, $\mathbf{M}^3$ allows both pipelines to use each RB, determines the scheduling solutions by each pipeline (for $\mathcal{K}^{\mathrm{E}}$ or $\mathcal{K}^{\not{\mathrm{E}}}$), and then obtains the PF values achieved on each RB in both cases by these pipelines. Then, by comparing the PF values obtained by each pipeline, $\mathbf{M}^3$ makes the final decision on how each RB is allocated, as well as corresponding assignment of MCS and number of data streams. Note that in addition to the multi-pipeline structure, parallelism is carried out throughout the design within each pipeline. For example, many operations are purposely designed to be independent among RRHs and RBs, and therefore they can be implemented in parallel.

Fig. 6.3 shows a flow chart of $\mathbf{M}^3$, which consists of the following key steps.

- Stage I: User classification. $\mathbf{M}^3$ divides all users into two groups: cell-edge users $\mathcal{K}^{\mathrm{E}}$ and non-edge users $\mathcal{K}^{\not{\mathrm{E}}}$, based on large-scale path loss. For each user k, $\mathbf{M}^3$ identifies its potential serving set $\mathcal{L}_k$.

- Stage II: Find promising solutions of RB allocation, as well as MCS assignment and number of data streams, for both cell-edge and non-edge users, by performing two independent parallel pipelines:

  - P1: Cell-Edge Pipeline. P1 finds promising solutions for $k \in \mathcal{K}^{\mathrm{E}}$, assuming each RB is used for cell-edge users.

  - P2: Non-Edge Pipeline. P2 finds promising solutions for $k \in \mathcal{K}^{\not{\mathrm{E}}}$, assuming each RB is used for non-edge users.

- Stage III: Compare the PF metrics provided by each pipeline and determine the final solution.

6.4 M^3: Design Details

In this section, we described the operations of $\mathbf{M}^3$ in detail at each stage.

6.4.1 Stage I: User classification

In this stage, we divide all users into two groups: cell-edge users $\mathcal{K}^{\mathrm{E}}$ and non-edge users $\mathcal{K}^{\not{\mathrm{E}}}$. For non-edge users, they are likely to experience much better channel qualities from their closest RRH than other RRHs. Therefore, users identified in $\mathcal{K}^{\not{\mathrm{E}}}$ will only be served by their closest RRH. Also, an RRH may perform MU-MIMO transmission to multiple non-edge users under this RRH to achieve higher throughput. On the other hand, for cell-edge users, they are likely to experience similar signal strength from at least two RRHs. To reinforce the desired signal strength, users identified in $\mathcal{K}^{\mathrm{E}}$ will be served by multiple RRHs simultaneously. $\mathbf{M}^3$ only applies SU-MIMO transmission for cell-edge users, as MU-MIMO transmission is not beneficial for improving each individual user's SINR, which is what a cell-edge user needs.

$\mathbf{M}^3$ determines whether a user is a non-edge user or a cell-edge user, as well as its RRH(s), based on the relative large-scale path loss from different RRHs to this user. Fast fading and beamforming gain are not considered in this stage. Let $g_{l,k}$ be the path loss from the RRH l to user k. Then the set of user k's RRH(s) is given by

$$\mathcal{L}_k = \left\{ l \in \mathcal{L} \,\middle|\, \frac{g_{l,k}}{\min_{n \in \mathcal{L}} \; g_{n,k}} \leq \delta \right\}, \quad (k \in \mathcal{K})$$

where δ ($\delta \geq 1$) is a pre-defined threshold to determine the subset of RRHs.

Next, if $\mathcal{L}_k$ for user k has more than one RRH, then user k is classified as a cell-edge user. Denote $\mathcal{K}^{\mathrm{E}}$ as the set of all cell-edge users. Then $\mathcal{K}^{\mathrm{E}} = \{k || \mathcal{L}_k | > 1, k \in \mathcal{K}\}$. Otherwise,

if $\mathcal{L}_k$ for user k has only one RRH, then user k is classified as a non-edge user. Denote $\mathcal{K}^{\not\mathrm{E}}$ as the set of all non-edge users. Then $\mathcal{K}^{\not\mathrm{E}} = \{k||\mathcal{L}_k| = 1, k \in \mathcal{K}\}$. Further, denote $\mathcal{K}_l^{\not\mathrm{E}}$ as the subset of non-edge users in $\mathcal{K}^{\not\mathrm{E}}$ that are receiving service from RRH l. Then $\mathcal{K}_l^{\not\mathrm{E}} = \{k|\mathcal{L}_k = \{l\}, k \in \mathcal{K}^{\not\mathrm{E}}\}$.

Since the decision of user-RRH association largely depends on the user's location, $\mathcal{L}_k$ can be updated based on long-term measurement. Therefore, Stage I will not be counted toward $\mathbf{M}^3$'s total execution time.

Upon the completion of Stage I, the search space for $x_{l,k}^b(t)$ variables will be narrowed down. That is, we have $x_{l,k}^b(t) = 0$ if $l \notin \mathcal{L}_k$. Stage I lays the foundation for employing independent parallel pipelines, which we will describe in Stage II.

6.4.2 Stage II: Find promising solutions for cell-edge and non-edge users

Stage II consists of two independent pipelines that can be implemented in parallel. Each pipeline is a sequence of operations to find promising solutions for users in $\mathcal{K}^{\mathrm{E}}$ and $\mathcal{K}^{\not\mathrm{E}}$, respectively. Within each pipeline, we have three key steps. Each step focuses on one type of variable (i.e., $x_{l,k}^b(t)$'s, $y_k(t)$'s or $z_k^m(t)$'s) for $\mathcal{K}^{\mathrm{E}}$ or $\mathcal{K}^{\not\mathrm{E}}$. That is, in each step, $\mathbf{M}^3$ will restrict the search space for that type of variable into a small but promising subspace. As RB allocation is the key problem for joint transmission under C-RAN, we start with $x_{l,k}^b(t)$ variables first under both pipelines. Then it is followed by steps for $y_k(t)$ and $z_k^m(t)$ variables.

Pipeline 1: Edge-Users It consists of the following two steps.

Step 1-A: Determine RB Allocation for Cell-Edge Users. In this step, we focus on the $x_{l,k}^b(t)$ variable for cell-edge users. That is, $\mathbf{M}^3$ identifies a subset $\mathcal{B}_k^{\mathrm{E}}$ of promising RBs from

$\mathcal{B}$ for each user $k \in \mathcal{K}^{\mathrm{E}}$. Note that all RRHs for user k (i.e., all $l \in \mathcal{L}_k$) must use the same RBs to perform joint transmission to user k.

To maximize the PF objective function (6.11), it is equivalent to maximize each user's instantaneous data rate normalized by its long-term data rate (i.e., $\frac{R_k(t)}{\tilde{R}_k(t-1)}$). For a cell-edge user under joint transmission, the instantaneous data rate is tightly related to the aggregated channel quality $(\sum_{l\in\mathcal{L}_k}\sqrt{P_{\mathrm{T}}}||\mathbf{H}_{l,k}^{b}||_F)^2$. Intuitively, when user k is experiencing high channel qualities from all its RRHs on an RB b, scheduling this RB jointly by $\mathcal{L}_k$ is likely to achieve a high data rate for this user. The achievable data rate can be approximated based on the channel capacity formula, i.e., $\log_2\left(1+\left(\sum_{l\in\mathcal{L}_k}\sqrt{P_{\mathrm{T}}}||\mathbf{H}_{l,k}^{b}||_F\right)^2\frac{1}{n_0^2}\right)$. Further, the data rate should be normalized by the user's long-term average data rate $\tilde{R}_k(t-1)$ in order to maximize the PF objective. Therefore, we consider the following metric to determine $\mathcal{B}_k^{\mathrm{E}}$:

$$q_k^{b,\mathrm{E}} = \frac{\log_2\left(1+\left(\sum_{l\in\mathcal{L}_k}\sqrt{P_{\mathrm{T}}}||\mathbf{H}_{l,k}^{b}||_F\right)^2\frac{1}{n_0^2}\right)}{\tilde{R}_k(t-1)},$$

where $||\mathbf{H}_{l,k}^{b}||_F$ is the Frobenius norm of $\mathbf{H}_{l,k}^{b,\mathrm{E}}$.

Fig. 6.4 illustrates how $\mathbf{M}^3$ finds $\mathcal{B}_k^{\mathrm{E}}$. First, $\mathbf{M}^3$ creates $|\mathcal{B}||\mathcal{K}^{\mathrm{E}}|$ independent parallel flows, each of which calculates one $q_k^{b,\mathrm{E}}$ for a user k and RB b. Second, $\mathbf{M}^3$ generates $|\mathcal{B}|$ independent flows, where each flow sorts $\{q_1^{b,\mathrm{E}}, q_2^{b,\mathrm{E}}, \cdots, q_{|\mathcal{K}^{\mathrm{E}}|}^{b,\mathrm{E}}\}$ in descending order. The sorting result indicates each user's priority to use an RB b. Third, with $|\mathcal{B}|$ independent flows, $\mathbf{M}^3$ allocates each RB based on the priority list and resolves the potential conflicts among RRHs. We describe the conflict resolution process through a simple example (see the bottom half in Fig. 6.4). For RB 4 in Fig. 6.4, RRHs 1 and 3 allocates RB 4 to user 3 because user 3 has the highest priority on this RB. Subsequently, user 1 cannot have RB 4, as RRH 1—one of its RRHs, has reserved this RB for user 3. But user 2 can have RB 4 because none of its RRHs have allocated RB 4 to a user with a priority higher than user 2.

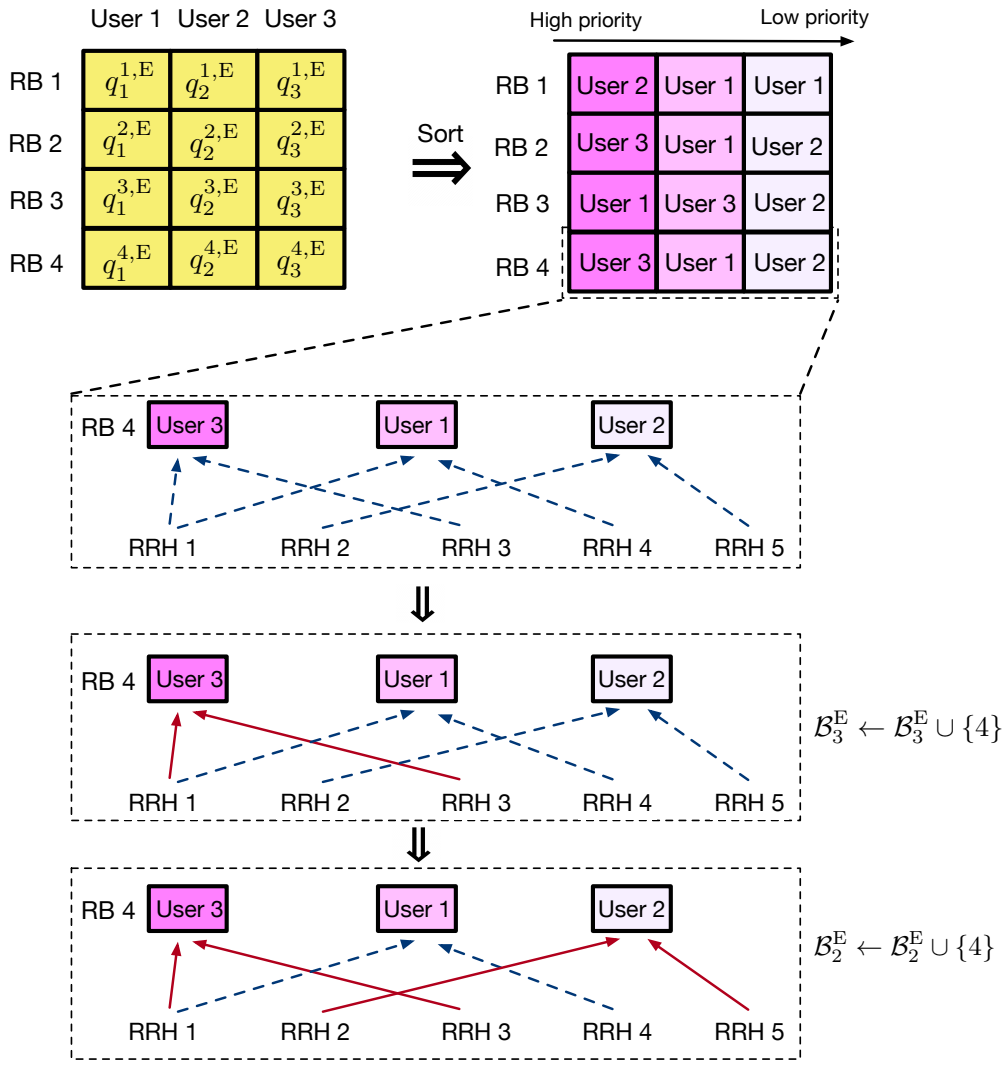


Figure 6.4: An illustration of Step 1-A in Pipeline 1.

Now we have the RB allocation results $\mathcal{B}_k^{\mathrm{E}}$ for each cell-edge user, which restricts the decision of $x_{l,k}^b(t)$ variables. That is, if $k \in \mathcal{K}^{\mathrm{E}}$ and $b \notin \mathcal{B}_k^{\mathrm{E}}$, then we have $x_{l,k}^b(t) = 0$ for all $l \in \mathcal{L}_k$. For $k \in \mathcal{K}^{\mathrm{E}}$ and $b \in \mathcal{B}_k^{\mathrm{E}}$, a potential (promising) solution could be to let $x_{l,k}^b(t) = 1$. But the final decision will be made later (after we compare the solution from non-edge pipeline).

Step 1-B: Determine Number of Data Streams for Cell-Edge Users. In this step, we focus on the $y_k(t)$ variables for cell-edge users. There exists a trade-off between the achievable data rate and number of data streams for a user. On one hand, more data streams have the potential to increase a user's total data rate, leveraging the spatial multiplexing for MIMO channels. On the other hand, transmitting too many data streams may cause performance loss, due to power splitting and interference among the streams.

To gain some insights on how we should decide the value of $y_k(t)$, let's first consider a single-link MIMO channel $\mathbf{H}$. Consider the eigenmode beamforming and equal power allocation. The achievable data rate can be given as a function of the number of data streams [9], i.e.,

$$r(y) = \sum_{i=1}^{y} \log\left(1 + \frac{\frac{P_{\mathrm{T}}}{y}[\sigma(i)]^2}{n_0^2}\right), \tag{6.12}$$

where y is the number of data streams on channel $\mathbf{H}$ ($y \leq \mathrm{rank}(\mathbf{H})$) and $\sigma(i)$ is the i-th largest eigenvalue of $\mathbf{H}$. Eq. (6.12) shows the fundamental relationship between achievable data rate and the number of streams on a single-link MIMO channel, which is tightly related to per-stream SNR ($\frac{P_{\mathrm{T}}}{y\ n_0^2}$) and singular values $\sigma(i)$.

Inspired by Eq. (6.12), we propose the following approach to determine the value of $y_k(t)$ in the context of joint transmission. First, recall that a user must use the same MCS across all streams. This suggests that we should consider the lowest SINR among user k's all streams. If user k is receiving f streams, then we evaluate an estimated SINR of the stream

with the f-th largest eigenvalue, which is defined as

$$\gamma_k^{\mathrm{E}}(f) = \frac{\left(\sum_{l\in\mathcal{L}_k}\sqrt{\frac{P_{\mathrm{T}}}{f}}\ E_b[\sigma_{l,k}^b(f)]\right)^2}{n_0^2}.$$

In the above expression, $\sigma_{l,k}^b(f)$ is the f-th largest eigenvalue of $\mathbf{H}_{l,k}^b$ and $E_b[\sigma_{l,k}^b(f)]$ is the average eigenvalue over all $b \in \mathcal{B}_k^{\mathrm{E}}$. Obviously, we have $\gamma_k^{\mathrm{E}}(1) > \gamma_k^{\mathrm{E}}(2) > \cdots \geq \gamma_k^{\mathrm{E}}(N_{\mathrm{R}})$ for any user k.

Next, let $\bar{m}_k^{\mathrm{E}}(f)$ be the largest MCS level in $\mathcal{M}$ that can be used to satisfy user k's estimated SINR $\gamma_k^{\mathrm{E}}(f)$, i.e.,

$$\begin{aligned}\bar{m}_k^{\mathrm{E}}(f) = \max_{m\in\mathcal{M}} \quad & m \\ \text{s.t. } & \gamma_k^{\mathrm{E}}(f) \geq \theta^m.\end{aligned}$$

We must have $\bar{m}_k^{\mathrm{E}}(1) \geq \bar{m}_k^{\mathrm{E}}(2) \geq \cdots \geq \bar{m}_k^{\mathrm{E}}(N_{\mathrm{R}})$ for any user $k \in \mathcal{K}^{\mathrm{E}}$. As the data rate corresponding to $\bar{m}_k^{\mathrm{E}}(f)$ is $r^{\bar{m}_k^{\mathrm{E}}(f)}$, the sum rate of f streams can be given by $f \cdot r^{\bar{m}_k^{\mathrm{E}}(f)}$, if all streams are successfully received.

Finally, we determine the number of data streams $y_k(t)$ for a user $k \in \mathcal{K}^{\mathrm{E}}$ by choosing an f that maximizes $f \cdot r^{\bar{m}_k^{\mathrm{E}}(f)}$, i.e.,

$$y_k(t) = \arg\max_{f\leq N_{\mathrm{R}}} \quad f \cdot r^{\bar{m}_k^{\mathrm{E}}(f)}.$$

Note that determination of $y_k(t)$ is independent among the cell-edge users and therefore can be implemented in parallel.

Step 1-C: Determine Candidate MCS for Cell-Edge Users. In this step, we focus on $z_k^m(t)$ variables. In Step 1-B, we identified the largest MCS level $\bar{m}_k(y_k(t))$ based on the estimated SINR $\gamma_k^{\mathrm{E}}(f)$. However, simply applying MCS $\bar{m}_k(y_k(t))$ to user k can be overly optimistic

for a number of reasons. First, $\gamma_k^{\mathrm{E}}(f)$ is approximated based on channels' eigenvalues, while non-SVD based beamforming techniques, such as MMSE and ZF, cannot process signals in the eigenspace and will result in an inferior performance. Second, $\gamma_k^{\mathrm{E}}(f)$ does not consider inter-cell interference from RRHs $l \notin \mathcal{L}_k$. Third, $\gamma_k^{\mathrm{E}}(f)$ is based on averaged eigenvalues over all $b \in \mathcal{B}_k^{\mathrm{E}}$. Applying a lower MCS for user k has the potential to facilitate successful transmissions on more RBs and thus achieve a higher sum rate.

Therefore, instead of simply applying MCS $\bar{m}_k^{\mathrm{E}}(y_k(t))$ to user k, we propose to picking up multiple MCS candidates that are lower than $\bar{m}_k^{\mathrm{E}}(y_k(t))$ simultaneously. Then $\mathbf{M}^3$ can process multiple MCS choices for a user in parallel. How to determine its final MCS jointly with all other users will be discussed in Stage III. Now let's focus on determining the MCS candidates for a particular cell-edge user k.

Note that any MCS that is much less than $\bar{m}_k^{\mathrm{E}}(y_k(t))$ is not a promising candidate, as it can only support a low data rate. Therefore, $\mathbf{M}^3$ only chooses a candidate MCS that is lower than $\bar{m}_k^{\mathrm{E}}(y_k(t))$ but the difference between $\bar{m}_k^{\mathrm{E}}(y_k(t))$ and this MCS is within a pre-defined value M_Δ. Formally, $\mathbf{M}^3$ determines the candidate MCS set for user k through the following expression:

$$\mathcal{M}_k^{\mathrm{E}} = \{m \in \mathcal{M} | 0 \leq \bar{m}_k^{\mathrm{E}}(y_k(t)) - m < M_\Delta\}.$$

After this step, we have $z_k^m(t) = 0$ if $k \in \mathcal{K}^{\mathrm{E}}$ and $m \notin \mathcal{M}_k^{\mathrm{E}}$. As described in problem OPT, the choice of a user's MCS is coupled with other (both cell-edge and non-edg) users. How to finalize MCS selection in $\mathcal{M}_k^{\mathrm{E}}$ for all users will be discussed in Stage III (after we discuss MCS selection for non-edge users).

Pipeline 2: Non-Edge Users This pipeline is designed for non-edge users $\mathcal{K}^{\not{\mathrm{E}}}$, assuming that each RB is available only for non-edge users. Note that for a user $k \in \mathcal{K}^{\not{\mathrm{E}}}$, it receives much higher signal strength from its own RRH than that from any other RRHs. Thus,

when finding promising scheduling solutions at an RRH l, we can reply solely on the channel information within RRH l, while treating the inter-cell interference as noise. This allows $\mathbf{M}^3$'s decision for non-edge users to be independent among different RRHs and therefore achieves a higher level of parallelism. Pipeline 2 consists of the following three steps.

Step 2-A: Select Promising MU-MIMO Users on Each RB under Each RRH. In this step, we deal with $x_{l,k}^b(t)$ variables for non-edge users $\mathcal{K}_l^{\not\Vdash}$. Compared to cell-edge users, non-edge users experience better SINR on average. To exploit this property for a higher throughput, we employ MU-MIMO transmission for non-edge users, which is not used for cell-edge user. Under MU-MIMO transmission, more than one user will be selected on each RB under each RRH.

$\mathbf{M}^3$ selects users on each RB for MU-MIMO transmission through two operations. First, $\mathbf{M}^3$ finds a subset $\widetilde{\mathcal{K}}_l^{b,\not\Vdash}$ of promising users from $\mathcal{K}_l^{\not\Vdash}$ based on channel quality. Second, $\mathbf{M}^3$ further selects a subset $\mathcal{K}_l^{b,\not\Vdash}$ of promising users from $\widetilde{\mathcal{K}}_l^{b,\not\Vdash}$ based on the channel correlation among users to form MU-MIMO transmission. We now describe each operation in detail.

First, $\mathbf{M}^3$ intensifies promising MU-MIMO users on each RB based on channel quality. Similar to Pipeline 1, we introduce a metric $q_{l,k}^{b,\not\Vdash}$ to approximate user k's data rate normalized by its long-term data rate, except that we now consider only one RRH for each non-edge user. Specifically, $\mathbf{M}^3$ creates $\sum_{l\in\mathcal{L}} |\mathcal{B}||\mathcal{K}_l^{\not\Vdash}|$ independent parallel flows, each of which calculates one $q_{l,k}^{b,\not\Vdash}$ for $l \in \mathcal{L}, k \in \mathcal{K}_l^{\not\Vdash}$ and $b \in \mathcal{B}$ and $q_{l,k}^{b,\not\Vdash}$ is given by:

$$q_{l,k}^{b,\not\Vdash} = \frac{\log_2\left(1 + \frac{P_{\mathrm{T}}}{n_0^2}||\mathbf{H}_{l,k}^b||_F^2\right)}{\tilde{R}_k(t-1)}.$$

Then, $\mathbf{M}^3$ generates $|\mathcal{L}||\mathcal{B}|$ independent parallel flows, each of which sorts $\{q_{l,1}^{b,\not\Vdash}, q_{l,2}^{b,\not\Vdash}, \cdots, q_{l,|\mathcal{K}_l^{\not\Vdash}|}^{b,\not\Vdash}\}$ in descending order for a RRH $l \in \mathcal{L}$ and a RB $b \in \mathcal{B}$. Let $\pi_{l,k}^b$ be the order of $q_{l,k}^{b,\not\Vdash}$ in

$\{q_{l,1}^{b,\not\mathbb{E}}, q_{l,2}^{b,\not\mathbb{E}}, \cdots, q_{l,|\mathcal{K}_l^{\not\mathbb{E}}|}^{b,\not\mathbb{E}}\}$. Suppose $\mathbf{M}^3$ selects K_{Q} ($< |\mathcal{K}_l^{\not\mathbb{E}}|$) candidate users based on channel qualities, then the subset $\widetilde{\mathcal{K}}_l^{b,\not\mathbb{E}}$ of promising users is determined by

$$\widetilde{\mathcal{K}}_l^{b,\not\mathbb{E}} = \{k \in \mathcal{K}_l^{\not\mathbb{E}} | \pi_{l,k}^b \leq K_{\mathrm{Q}}\}. \qquad (b \in \mathcal{B}, l \in \mathcal{L})$$

Second, we identify promising users from $\widetilde{\mathcal{K}}_l^{b,\not\mathbb{E}}$ on each RB to form MU-MIMO transmission based on channel correlations. That is, among the users in $\widetilde{\mathcal{K}}_l^{b,\not\mathbb{E}}$ for each $b \in \mathcal{B}, l \in \mathcal{L}$, $\mathbf{M}^3$ selects a subset $\mathcal{K}_l^{b,\not\mathbb{E}}$ ($\subset \widetilde{\mathcal{K}}_l^{b,\not\mathbb{E}}$) of users, such that the users in $\mathcal{K}_l^{b,\not\mathbb{E}}$ have low channel correlations among themselves. The rationale behind this operation is that, in general, the lower correlations among the co-scheduled users, the higher sum of data rate can be achieved. This is because mutually orthogonal channels can better preserve the desired signal strength after applying beamforming matrices [23, 51].

We evaluate channel correlations based on *chordal distance* [150], which measures the angle between two multi-dimensional subspace. A larger value of chordal distance means more orthogonality between these two subspaces. Let $\mathring{\mathbf{H}}_{l,k}^b$ be the orthonormal base of $\mathbf{H}_{l,k}^b$. Then the chordal distance between $\mathbf{H}_{l,k_1}^b$ and $\mathbf{H}_{l,k_2}^b$ is given by:

$$d_l^b(k_1, k_2) = \frac{1}{\sqrt{2}} ||\mathring{\mathbf{H}}_{l,k_1}^{b\dagger} \mathring{\mathbf{H}}_{l,k_1}^b - \mathring{\mathbf{H}}_{l,k_2}^{b\dagger} \mathring{\mathbf{H}}_{l,k_2}^b||_F.$$

$\mathbf{M}^3$ computes $d_l^b(k_1, k_2)$ for all $k_1 \in \mathcal{K}_l^{\not\mathbb{E}}, k_2 \in \mathcal{K}_l^{\not\mathbb{E}}$, $k_1 \neq k_2$, which can be executed in parallel on each RB $b \in \mathcal{B}$ and $l \in \mathcal{L}$. Suppose we are going to select K_{MU} ($< |\mathcal{K}_l^{\not\mathbb{E}}|$) users for MU-MIMO transmission. Then the subset $\mathcal{K}_l^{b,\not\mathbb{E}}$ of candidate users is determined by the following. $\mathbf{M}^3$ adds the first user to $\mathcal{K}_l^{b,\not\mathbb{E}}$ that has the highest $q_{l,k}^{b,\not\mathbb{E}}$, i.e., $\mathcal{K}_l^{b,\not\mathbb{E}} = \{\arg\max_{k \in \mathcal{K}_l^{\not\mathbb{E}}} q_{l,k}^{b,\not\mathbb{E}}\}$. Subsequently, we add users one at a time to $\mathcal{K}_l^{b,\not\mathbb{E}}$, by picking the user with the largest average chordal distance to existing users in $\mathcal{K}_l^{b,\not\mathbb{E}}$, until we have K_{MU} users

in $\mathcal{K}_l^{b,\mathbb{E}}$.

After Step 2-A, $\mathbf{M}^3$ restricts the users that can be scheduled on RB b under RRH l. That is, for any $k \in \mathcal{K}_l^{\mathbb{E}}$ and $k \notin \mathcal{K}_l^{b,\mathbb{E}}$, we have $x_{l,k}^b(t) = 0$. On the other word, under each RRH and each RB, the total number of possibilities to allocate RBs is reduced to $2^{K_{\mathrm{MU}}}$, as $x_{l,k}^b(t)$ can be either 1 or 0 and $|\mathcal{K}_l^{b,\mathbb{E}}| = K_{\mathrm{MU}}$. The final decision will be made in Stage III (after we have MCS solutions and evaluate the corresponding objective values).

Step 2-B: Determine Number of Data Streams for Non-edge Users. In this step, we work on $y_k(t)$ variables for non-edge users. Similar to cell-edge users, we consider a metric of estimated SINR as a function of the number of streams. But for a non-edge user, we only need to consider its own RRH. That is, if RRH l's non-edge user k is receiving f streams, the estimated SINR per stream is given by

$$\gamma_{l,k}^{\mathbb{E}}(f) = \frac{P_{\mathrm{T}}}{f} \frac{E_b[\sigma_{l,k}^b(f)]^2}{n_0^2},$$

where $E_b[\sigma_{l,k}^b(f)]$ is the average eigenvalue over all $b \in \mathcal{B}_k^{\mathbb{E}}$, and $\mathcal{B}_{l,k}^{\mathbb{E}}$ is the set of candidate RBs for non-edge user k under RRH l, i.e., $\mathcal{B}_{l,k}^{\mathbb{E}} = \{b \in \mathcal{B} | k \in \mathcal{K}_l^{b,\mathbb{E}}\}$.

Next, following the same rationale as cell-edge users, let $\bar{m}_k^{\mathbb{E}}(f)$ be the largest MCS that satisfies $\gamma_{l,k}^{\mathbb{E}} \geq \theta^{\bar{m}_k^{\mathbb{E}}(f)}$. Then the number of data streams $y_k(t)$ for a non-edge user k is determined by choosing the f that maximizes $f \cdot r^{\bar{m}_k^{\mathbb{E}}(f)}$, i.e.,

$$y_k(t) = \arg\max_{f \leq N_{\mathrm{R}}} \quad f \cdot r^{\bar{m}_k^{\mathbb{E}}(f)}.$$

Step 2-C: Determine Candidate MCS for Non-edge Users. The MCS $\bar{m}_k^{\mathbb{E}}(y_k(t))$ found in Step 2-B for a non-edge user is also an optimistic choice, because the transmit power P_{T} shall be shared with other non-edge users due to the MU-MIMO transmission. Similar to the

approach for cell-edge users, $\mathbf{M}^3$ only chooses a candidate MCS that is lower than $\bar{m}_k^{\not\mathrm{E}}(y_k(t))$ and the difference between $\bar{m}_k^{\not\mathrm{E}}(y_k(t))$ and this MCS is within a pre-defined value M_Δ. That is, the candidate MCS set for non-edge user k is given by

$$\mathcal{M}_k^{\not\mathrm{E}} = \{m \in \mathcal{M} | 0 \leq \bar{m}_k^{\not\mathrm{E}}(y_k(t)) - m < M_\Delta\}.$$

After this step, we have $z_k^m(t) = 0$ if $m \notin \mathcal{M}_k^{\not\mathrm{E}}$ for a non-edge user k.

6.4.3 Stage III: Determine final solution

In this stage, we show how $\mathbf{M}^3$ performs comparison among different candidate solutions and selects the final solution to problem OPT. The key steps of Stage III are illustrated in Fig. 6.5.

$\mathbf{M}^3$ will enumerate all possible $x_{l,k}^b(t)$ and $y_k(t)$ assignments from each pipeline in Stage II. However, for $z_k^m(t)$ variables, the search space is too large for enumeration. Fortunately, the MCS candidates $\mathcal{M}_k^{\mathrm{E}}$ and $\mathcal{M}_k^{\not\mathrm{E}}$ for cell-edge and non-edge users (identified by Stage II) are already promising MCS solutions. Therefore, $\mathbf{M}^3$ can further identify a smaller subset of MCS solution space based on $\mathcal{M}_k^{\mathrm{E}}$'s and $\mathcal{M}_k^{\not\mathrm{E}}$'s. Specifically, let $\widetilde{\mathcal{M}}$ be the MCS solution space, i.e.,

$$\widetilde{\mathcal{M}} = \mathcal{M}_1^{\mathrm{E}} \times \cdots \times \mathcal{M}_{|\mathcal{K}^{\mathrm{E}}|}^{\mathrm{E}} \times \mathcal{M}_1^{\not\mathrm{E}} \times \cdots \times \mathcal{M}_{|\mathcal{K}^{\not\mathrm{E}}|}^{\not\mathrm{E}} \subseteq \mathcal{M}^{|\mathcal{K}|},$$

where $\times$ denotes the Cartesian product. Then each element from $\widetilde{\mathcal{M}}$ (a $1 \times |\mathcal{K}|$ vector) is a feasible MCS solution for all users. Supposing we are going to try M_{S} different MCS solutions in parallel, $\mathbf{M}^3$ will randomly select M_{S} elements from $\widetilde{\mathcal{M}}$ in parallel. As each $\mathcal{M}_k$ is a set of promising MCS candidates for user k, we can apply a simple yet effective approach to determine MCS solutions—by randomly (uniform) picking M_{S} elements from $\widetilde{\mathcal{M}}$.

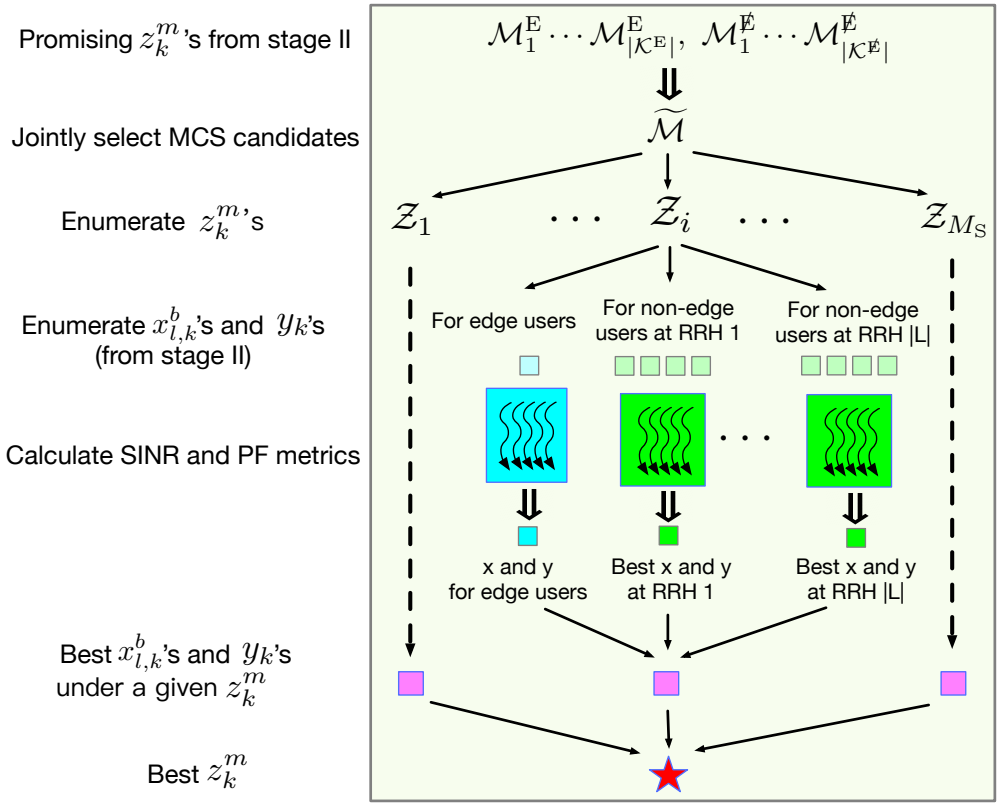


Figure 6.5: Stage III determines the final scheduling solutions for all users under all RRHs.

After this step, we have M_{S} candidate MCS solutions (i.e., M_{S} sets of feasible $z_k^m(t)$'s), denoted by $\mathcal{Z}_1, \mathcal{Z}_2, \cdots, \mathcal{Z}_{M_{\mathrm{S}}}$, respectively. Now we can enumerate $z_k^m(t)$ variables, calculate PF metrics, and perform comparisons.

After obtaining $\mathcal{Z}_i$'s, $\mathbf{M}^3$ calculates SINR of each data stream and PF metrics under a given $z_k^m(t)$. Specifically, $\mathbf{M}^3$ enumerates all possible $x_{l,k}^b(t)$ and $y_k(t)$ assignments for cell-edge and non-edge users from Stage II. With the given $x_{l,k}^b(t)$'s and $y_k(t)$'s, we derive the corresponding beamforming matrices (based on MMSE beamforming) at each RRH and each user under a given scheduling solution, and then calculate SINR for every stream. When calculating user k's SINR, we exploit independency and reduce the computational burden by only considering the beamforming gain from user k's RRHs $\mathcal{L}_k$. That is, for interference from RRHs other than $\mathcal{L}_k$, we only consider their transmit power P_{T} and pathloss attenuation, without using their beamforming matrices to obtain the inter-cell interference.[2] Although such a simplification has little impact on SINR accuracy, it can decouple SINR calculations among the RRHs and make it possible for parallel implementation.

Next, with given SINRs and $z_k^m(t)$'s, we can obtain PF metric $\sum_{f=1}^{y_k(t)} \frac{r_k^{b,f,m}(t)}{\tilde{R}_k(t-1)}$ for each user on each RB. We conduct two-level comparisons to determine the final decision. At the first level, we focus on a given $\mathcal{Z}_i$ and determine whether an RB b should be allocated to a cell-edge user or a non-edge user (and all RBs are evaluated in parallel). If RB b is used for a cell-edge user k, then the PF value can be computed by

$$V_k^{b,\mathrm{E}} = \sum_{f=1}^{y_k(t)} \frac{r_k^{b,f,m_k^*}(t)}{\tilde{R}_k(t-1)}, \tag{6.13}$$

where m_k^* is user k's MCS level under the given $\mathcal{Z}_i$. On the other hand, if this RB is used

[2] All beamforming matrices will be considered for throughput evaluation in Sec. 6.6.

for non-edge users separately by user k's RRHs $\mathcal{L}_k$, then the PF value can be computed by

$$V_{\mathcal{L}_k}^{b,\not\mathrm{E}} = \sum_{l\in\mathcal{L}_k} \max_{x_{l,k'}^b} \sum_{k'\in\mathcal{K}_l^{b,\not\mathrm{E}}} \sum_{f=1}^{y_{k'}(t)} \frac{r_{k'}^{b,f,m_{k'}^*}(t)}{\tilde{R}_{k'}(t-1)} x_{l,k'}^b(t), \tag{6.14}$$

where $V_{\mathcal{L}_k}^{b,\not\mathrm{E}}$ is maximized over $2^{K_{\mathrm{MU}}}$ cases (for MU-MIMO user selection) under each RRH (see Step 2-A).

In $M_{\mathrm{S}}|\mathcal{K}^{\mathrm{E}}||\mathcal{B}|$ parallel flows, $\mathbf{M}^3$ compares $V_k^{b,\mathrm{E}}$ with $V_{\mathcal{L}_k}^{b,\not\mathrm{E}}$ (under a given $\mathcal{Z}_i$). If $V_k^{b,\mathrm{E}} \geq V_{\mathcal{L}_k}^{b,\not\mathrm{E}}$, then we allocate RB b to cell-edge user k at RRHs $\mathcal{L}_k$, i.e., $x_{l,k}^b(t) = 1$ for all $l \in \mathcal{L}_k$, and $x_{l,k'}^b(t) = 0$ for all $l \in \mathcal{L}_k, k' \in \mathcal{K}_l^{\not\mathrm{E}}$. Otherwise, we allocate RB b to non-edge users at each RRH in $\mathcal{L}_k$, corresponding to the MU-MIMO user selection that maximizes $V_{\mathcal{L}_k}^{b,\not\mathrm{E}}$. Note that if an RRH is not associated with any cell-edge users by Stage I, then $x_{l,k}^b(t)$ can be directly determined based on MU-MIMO user selection, without any comparison with cell-edge users. After the first level of comparison, we complete the $x_{l,k}^b(t)$ assignment, along with corresponding $y_k(t)$'s, under a given $\mathcal{Z}_i$. Thus the objective value under each $\mathcal{Z}_i$ is obtained.

At the second level of comparison, among M_{S} intermediate best solutions under each $\mathcal{Z}_i$, we choose the $\mathcal{Z}_i$ that offers the highest objective value to problem OPT, along with corresponding $x_{l,k}^b(t)$'s and $y_k(t)$'s. This gives the final solution to problem OPT.

In this section, we described in detail the three stages of $\mathbf{M}^3$. Throughout our design, the exploration of independent operations is carried out, which makes it possible for a parallel implementation. In the following section, we move forward to implement $\mathbf{M}^3$ on an actual hardware for our ultimate goal—offering a solution to problem OPT in real-time.

6.5 A Real-Time GPU-based Implementation

In this section, we present our implementation of $\mathbf{M}^3$. We choose a COTS GPU platform as our hardware for implementation. For optimal implementation, one must have a thorough knowledge of a given GPU's capability as well as its limitations. In the rest of this section, we document our implementation efforts.

Platform We implement $\mathbf{M}^3$ on a NVIDIA DGX station, which comes with 4 COTS V100 GPU cards. We will only use two of them. Each V100 card has 5120 CUDA cores. Data communication between CPU and GPU is based on a PCIe V3.0 architecture, and the GPU-to-GPU data communication is based on NVIDIA NVLink architecture [151]. The CPU of the DGX Station is Intel Xeon E5-2698 v4 2.2 GHz (20-core). The programming platform is CUDA v10.2 [117].

Independent Pipelines Traditional parallel designs have only one pipeline, within which all parallel threads have the same computation procedures (see e.g., [95, 119, 120]). In contrast, Our $\mathbf{M}^3$ has two independent pipelines, as described in Sec. 6.4. The computation procedures for different threads are identical within the same pipeline while they are different from those in the other pipeline. This is possible on GPU by employing a programming method called *streams* that is offered by CUDA. A stream can execute a sequence of operations (e.g., Pipeline 1) successively on a GPU, while the operations executed by another stream (e.g., Pipeline 2) may run concurrently. By employing the streams, GPU cores can be utilized more efficiently. That is, whenever the computing operations in one pipeline do not fully occupy all the GPU cores, the GPU's streaming multiprocessor (SM) can schedule the remaining cores for the other pipeline.

Using Multiple GPU Cards As most of $\mathbf{M}^3$'s operations on RBs are designed to be independent, we can distribute computation tasks to multiple GPU cards based on RBs.

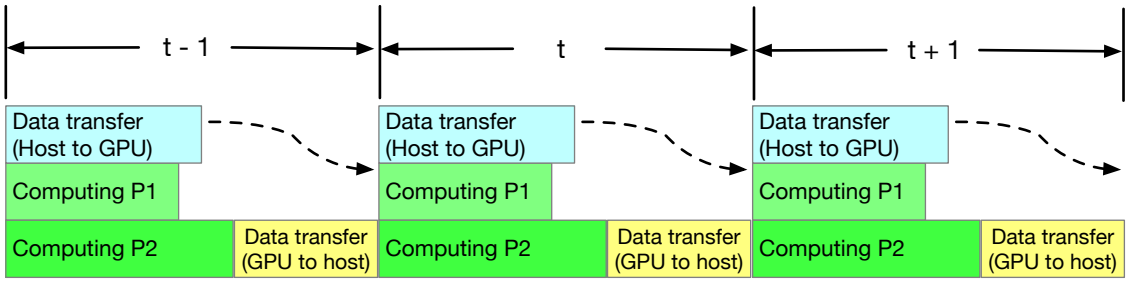


Figure 6.6: An illustration of parallel operations of pipelines and data transfer.

Specifically, with two GPU cards, the first card can handle operations for the first $|\mathcal{B}|/2$ RBs for both pipelines, and the second card will handle the remaining RBs. A small amount of operations requires the information on all RBs, such as computing the candidate MCS set $\mathcal{M}_k$. These operations will be performed on only one GPU card after a data transfer from the other GPU card.

Data Exchange between Pipelines / GPU Cards Although different pipelines may run concurrently, their completion time is unpredictable. $\mathbf{M}^3$ requires data exchange between pipelines occasionally, e.g., when we perform PF metric comparisons. Thus, it is important to do a synchronization before data exchange. Likewise, the computation progress in one GPU card may differ from the other. When data exchange between GPU cards is required, a device-level synchronization is needed. As a synchronization will pause part of the program and also introduce CPU's scheduling overhead, we must keep such an operation to a minimum and only use it when it is absolutely necessary.

Large-Scale Parallelism In addition to multi-pipeline and multi-GPU, parallelism is carried out throughout our implementation. For example, the operations for different RRHs under non-edge users' pipeline are independent and implemented in parallel. The operations of matrix inversions for computing the beamforming matrices are implemented in parallel and the fast on-chip shared memory is employed to reduce the memory access time.

Data Transfer between Host and GPU Data transfer time between host memory and

GPU memory also needs to be considered. Fortunately, data transfer and GPU computing can be done in parallel if the computation does not rely on the data that is being transferred. To take advantage of this property, we propose the following structure for data transfer (see Fig. 6.6). In TTI t, GPU computes the solution to problem OPT based on the channel information transferred in the previous TTI (i.e., TTI $t-1$), and the computation in TTI $t+1$ will be based on information transferred in TTI t, and so forth. This method is valid as long as channel coherence time is at least 3 TTIs (which is the case for most communication scenarios [170]) and can effectively mask out the transfer time of channel information from host to GPU.

6.6 Experimental Evaluation

6.6.1 Settings

We consider a C-RAN architecture, where a centralized BBU pool is serving multiple small cells. We randomly generate $|\mathcal{L}| = 7$ RRHs in a circle within a radius of 700 m. The minimum distance between every two RRHs is 350 m. $|\mathcal{K}|$ users are randomly deployed in the circle, and $|\mathcal{K}| = 100$ unless stated otherwise. Fig. 6.7 shows an instance of the network topology and the results of user classification done by Stage I. In Fig. 6.7, the red circles are the cell-edge users and the blue circles are non-edge users. Fig. 6.7 shows that we have 17 cell-edge users (out of 100 users) in this case ($\delta = 3$ dB).

The number of antennas at each RRH is chosen from $\{8, 12\}$ and the number of antennas at each user is 2. The number of MU-MIMO users in a cell K_{MU} is chosen from $\{2, 4\}$. The number of RBs $|\mathcal{B}|$ is 100 unless indicated otherwise. For the wireless channels, $\mathbf{H}_{l,k}^b(t)$ includes both large-scale fading $g_{l,k}$ and small-scale fading $\bar{\mathbf{H}}_{l,k}^b(t)$, i.e., $\mathbf{H}_{l,k}^b(t) = g_{l,k}^{-1}\bar{\mathbf{H}}_{l,k}^b(t)$.

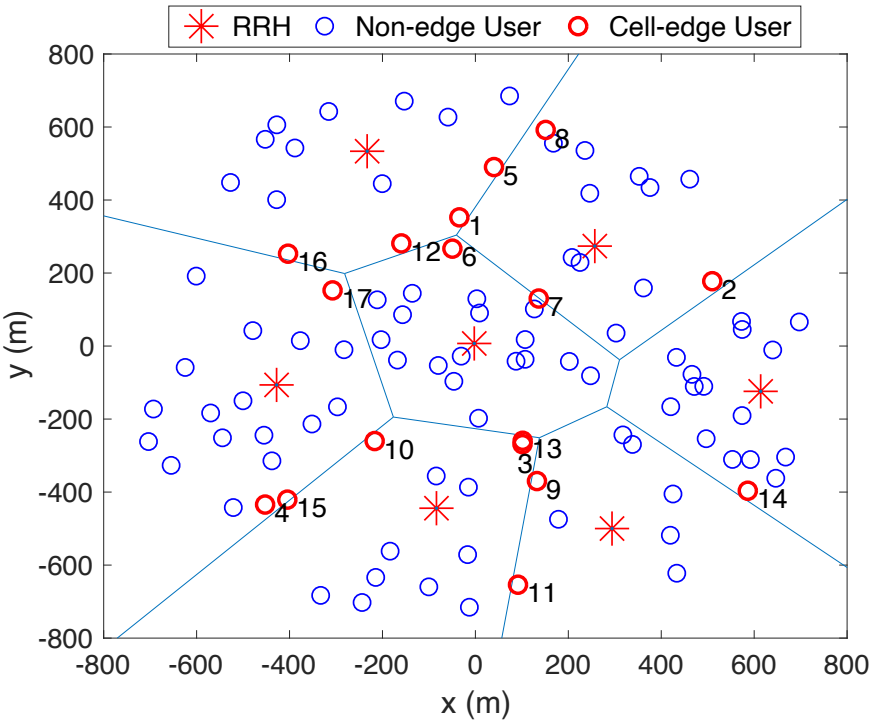


Figure 6.7: An instance of network topology with 7 RRHs and 100 users. Classification of cell-edge and non-edge users is done by Stage I with $\delta = 3$ dB.

Large-scale fading $g_{l,k}$ is given by $140.7 + 36.7 \log_{10}(d_{l,k})$ (in dB), where $d_{l,k}$ is the distance between RRH l and user k (in km). The small-scale fading $\bar{\mathbf{H}}_{l,k}^{b}(t)$ is modeled by Rayleigh channel model. We set transmit power P_{T} to be 36 dBm. The background noise power is set to -169 dBm/Hz and the channel bandwidth is 20 MHz. For parameters δ, K_{Q}, M_{Δ}, and M_{S} in our algorithm (see Sec. 6.4), we set $\delta = 3$ dB, $K_{\mathrm{Q}} = 10, M_{\Delta} = 6, M_{\mathrm{S}} = 300$.

6.6.2 Timing Performance

We first verify that $\mathbf{M}^3$ can meet the 1 ms real-time requirement, which is a major criterion for it to be useful for 5G C-RAN. We conduct experiments for 300 consecutive TTIs under two different settings: (a) $|\mathcal{K}| = 50$, $N_{\mathrm{T}} \in \{8, 12\}$, and (b) $|\mathcal{K}| = 100$, $N_{\mathrm{T}} \in \{8, 12\}$. The experimental results are shown in Fig. 6.8. We find that that $\mathbf{M}^3$ is able to offer a

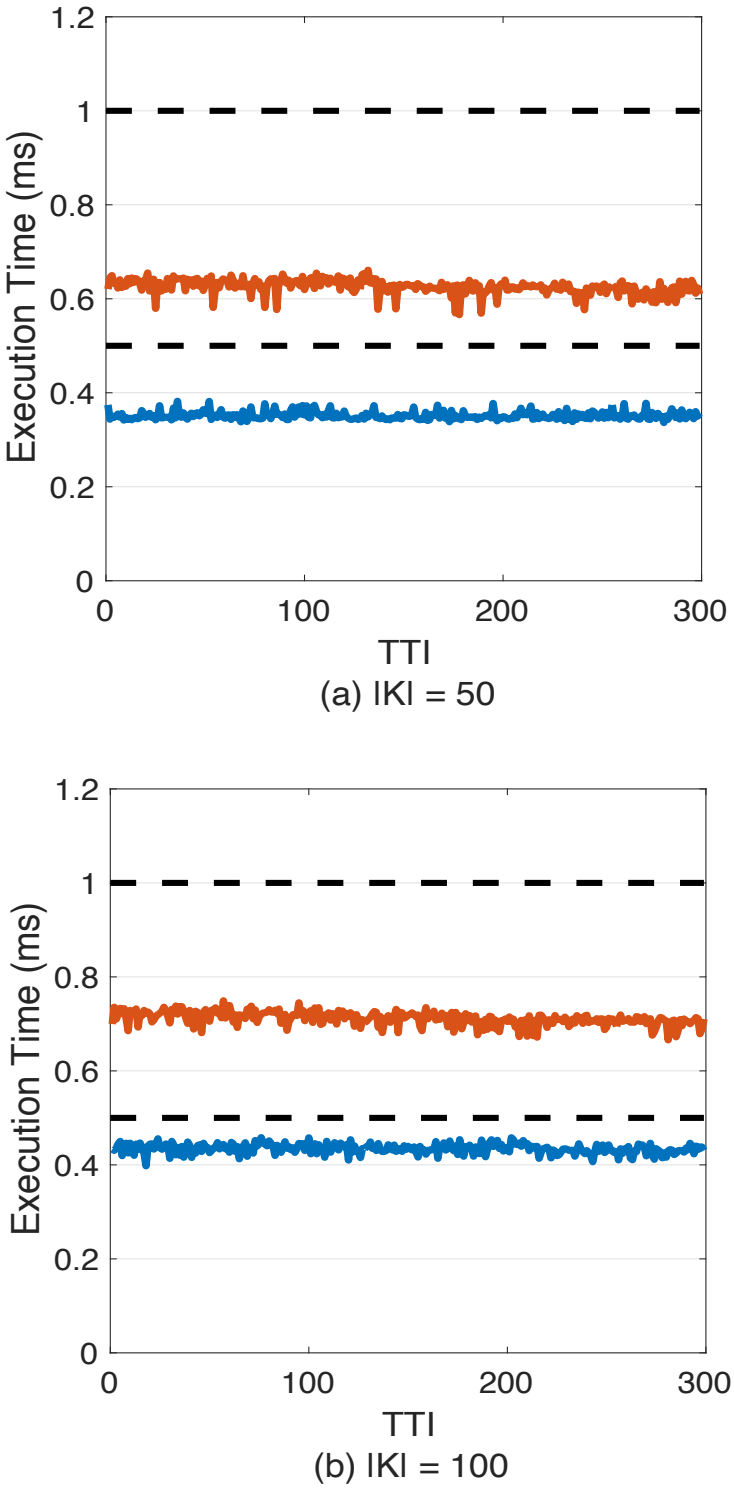


Figure 6.8: $\mathbf{M}^3$'s execution time. (a) $|\mathcal{K}| = 50$, $N_{\mathrm{T}} \in \{8, 12\}$, and (b) $|\mathcal{K}| = 100$, $N_{\mathrm{T}} \in \{8, 12\}$.

scheduling solution within 1 ms under all cases and TTIs in our experiments. Specifically, when $N_{\mathrm{T}} = 12$, $\mathbf{M}^3$'s average execution time is 626 μs and 712 μs for $|\mathcal{K}| = 50$ and $|\mathcal{K}| = 100$, respectively, which can meet the timing requirement for 5G NR numerology 0 (1 ms). When $N_{\mathrm{T}} = 8$, the average execution time is 351 μs and 435 μs for $|\mathcal{K}| = 50$ and $|\mathcal{K}| = 100$, respectively, which can meet 5G NR numerology 1 (500 μs).

Next, we vary the number of users $|\mathcal{K}|$ from 50 to 150 to show its impact on $\mathbf{M}^3$'s execution time. We consider the following settings: (a) $N_{\mathrm{T}} = 8, K_{\mathrm{MU}} = 2$, and (b) $N_{\mathrm{T}} = 12, K_{\mathrm{MU}} = 4$. $|\mathcal{B}|$ is 100. Fig. 6.9 shows $\mathbf{M}^3$'s average execution time with the maximum and minimum values over 100 consecutive TTIs. The results indicate that $\mathbf{M}^3$ finds the solution within 500 μs and 800 μs for up to 150 users under settings (a) and (b), respectively. Although the execution time increases with the number of users, Fig. 6.9 suggests that $\mathbf{M}^3$ can still meet 5G NR timing requirement (1 ms). Further, the rate of increase is much slower than that of the number of users. This is because that by the design of Step 1-A and Step 2-A in Stage II, $\mathbf{M}^3$ identifies a subset of the most promising users based on channel quality. Thus, only for a fixed and small number of users we need to perform those time-consuming calculations, such as beamforming matrices and SINR.

Now we study the timing performance as a function of the number of available RBs $|\mathcal{B}|$. We vary $|\mathcal{B}|$ from 20 to 100. The number of users is $|\mathcal{K}| = 100$. The results in Fig. 6.10(a) demonstrate that when $N_{\mathrm{T}} = 8$, the total execution time is well below 500 μs under different numbers of RBs. The increase of computation time is much slower than that of the number of RBs, because GPU has sufficient computing resources to accommodate parallel operations among different RBs. When $N_{\mathrm{T}} = 12$, Fig. 6.10(b) shows that the execution time is within 800 μs for all cases, and it is lower than 500 μs for up to $\sim$60 RBs. The increase in computation time w.r.t. the number of RBs is slightly faster than the case for $N_{\mathrm{T}} = 8$. This is because matrix operations are much more intensive on each RB when $N_{\mathrm{T}} = 12$, leading

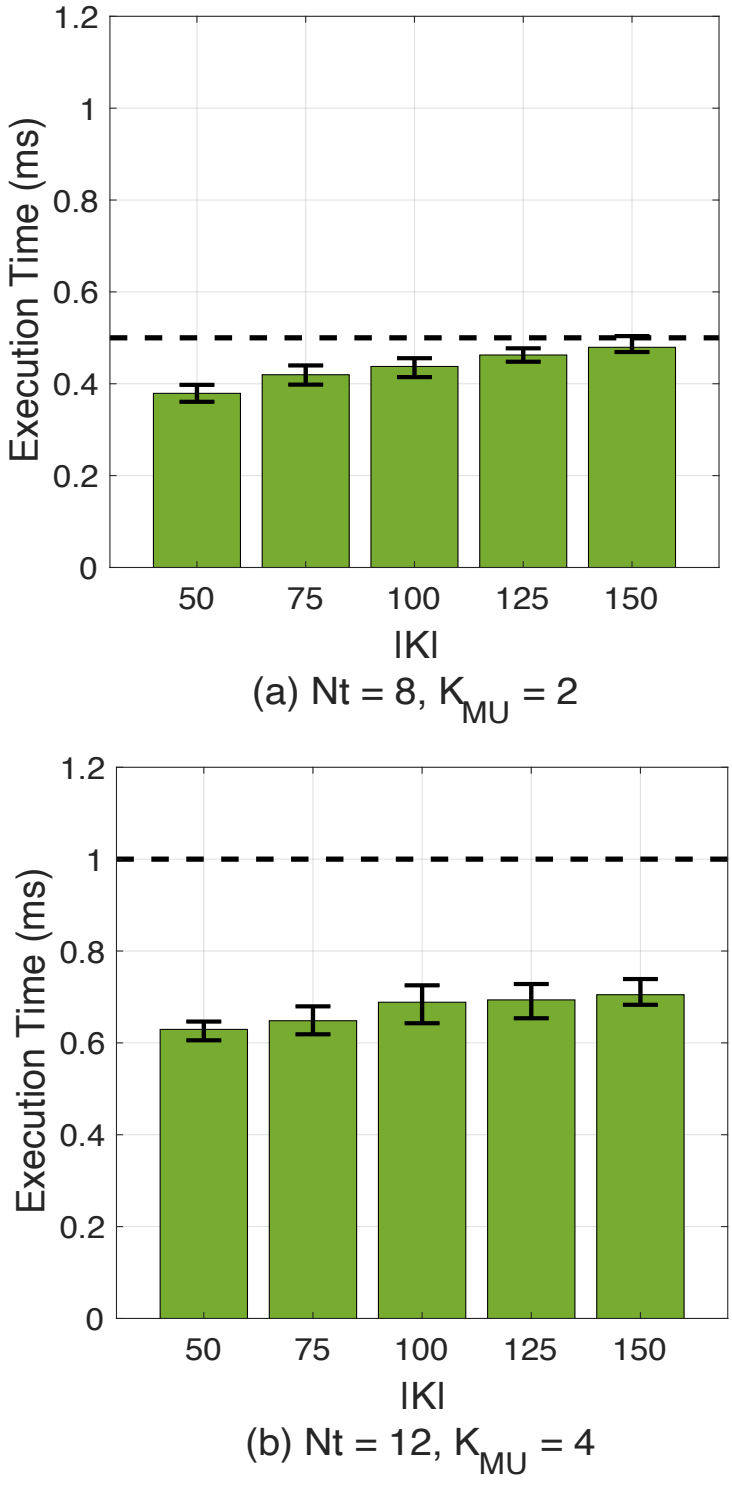


Figure 6.9: $\mathbf{M}^3$'s execution time (mean, max and min values over 100 consecutive TTIs) vs. the number users. (a) $N_{\mathrm{T}} = 8, K_{\mathrm{MU}} = 2$, and (b) $N_{\mathrm{T}} = 12, K_{\mathrm{MU}} = 4$.

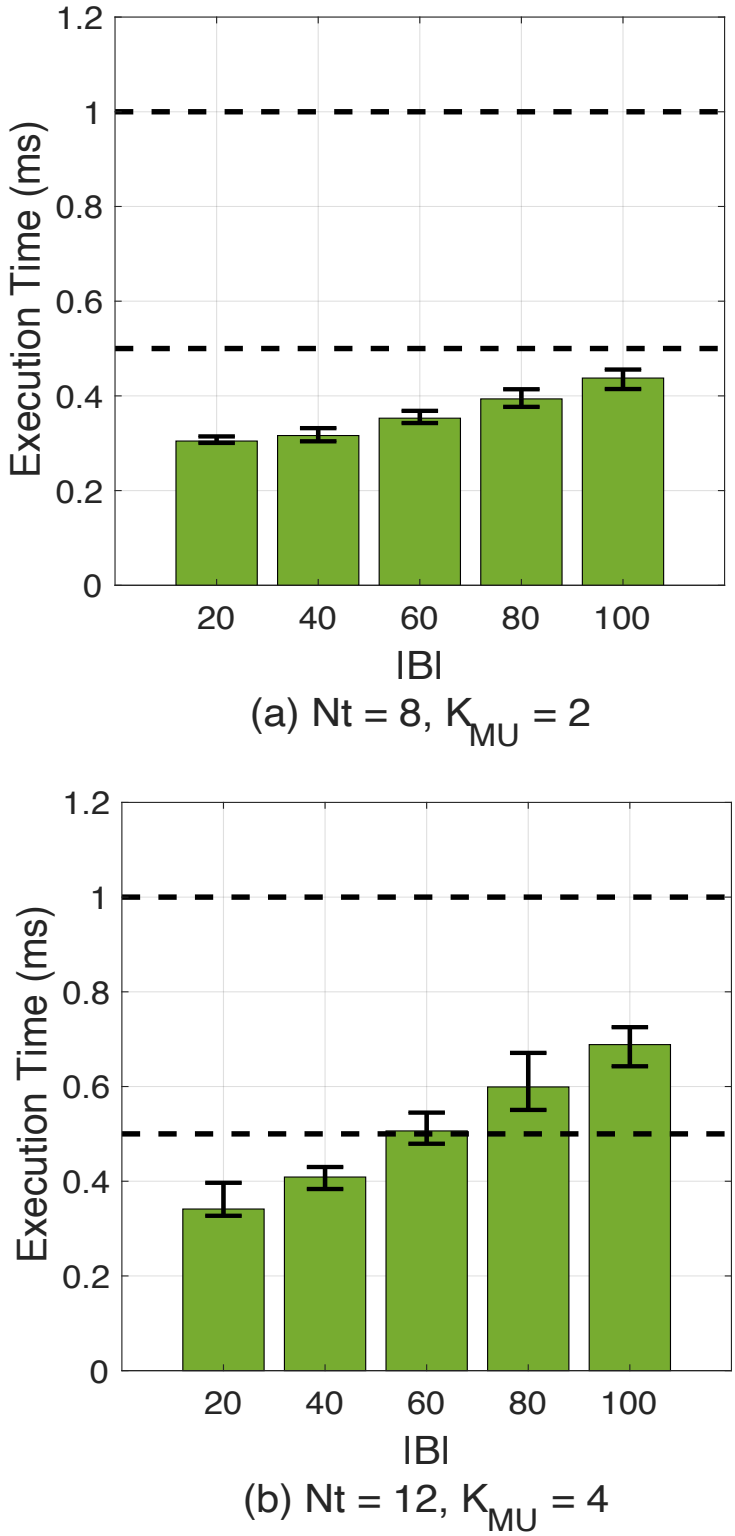


Figure 6.10: $\mathbf{M}^3$'s execution time (mean, max and min values over 100 consecutive TTIs) vs. the number of available RBs. (a) $N_{\mathrm{T}} = 8, K_{\mathrm{MU}} = 2$, and (b) $N_{\mathrm{T}} = 12, K_{\mathrm{MU}} = 4$.

Table 6.2: Comparison of user throughput at different percentiles when $N_{\mathrm{T}} = 8, K_{\mathrm{MU}} = 2$.

Percentile	Without JT (Mbps)	With JT (Mbps)	Gain
5th	0.52	0.64	23%
10th	0.56	0.72	28%
15th	0.72	0.97	34%
20th	0.98	2.57	161%
25th	1.37	3.05	122%
30th	2.42	3.71	53%
35th	2.74	3.82	40%
40th	3.09	3.95	28%
45th	3.18	4.03	27%
50th	3.54	4.30	22%

to more computation time. However, $\mathbf{M}^3$ is able to complete the computation in real-time (within 1 ms) for all cases by taking advantage of the large-scale parallelism.

6.6.3 Throughput Performance

We now evaluate $\mathbf{M}^3$'s throughput performance. Under the topology in Fig. 6.7, we compare the throughput performance achieved under joint transmission (with $\delta = 3$ dB) with the case when joint transmission is not used (i.e., $\delta = 0$ dB). We consider two different settings: (a) $N_{\mathrm{T}} = 8, K_{\mathrm{MU}} = 2$, and (b) $N_{\mathrm{T}} = 12, K_{\mathrm{MU}} = 4$.

Fig. 6.11 shows the cumulative distribution functions (CDF) of users' long-term average throughput. For example, in Fig. 6.11(a), the point (3.09 Mbps, 0.4) on the blue curve indicates that the 40th lowest user throughput (among 100 users) is 3.09 Mbps. The results in Fig. 6.11 suggest that the design of $\mathbf{M}^3$ is able to offer a better throughput performance when joint transmission (JT) is employed. To have a clear picture of the performance improvement over non-joint transmission, we use Tables 6.2, 6.3 and 6.4 to offer more details.

Table 6.2 shows the comparison of user throughput at different percentiles (ranging from

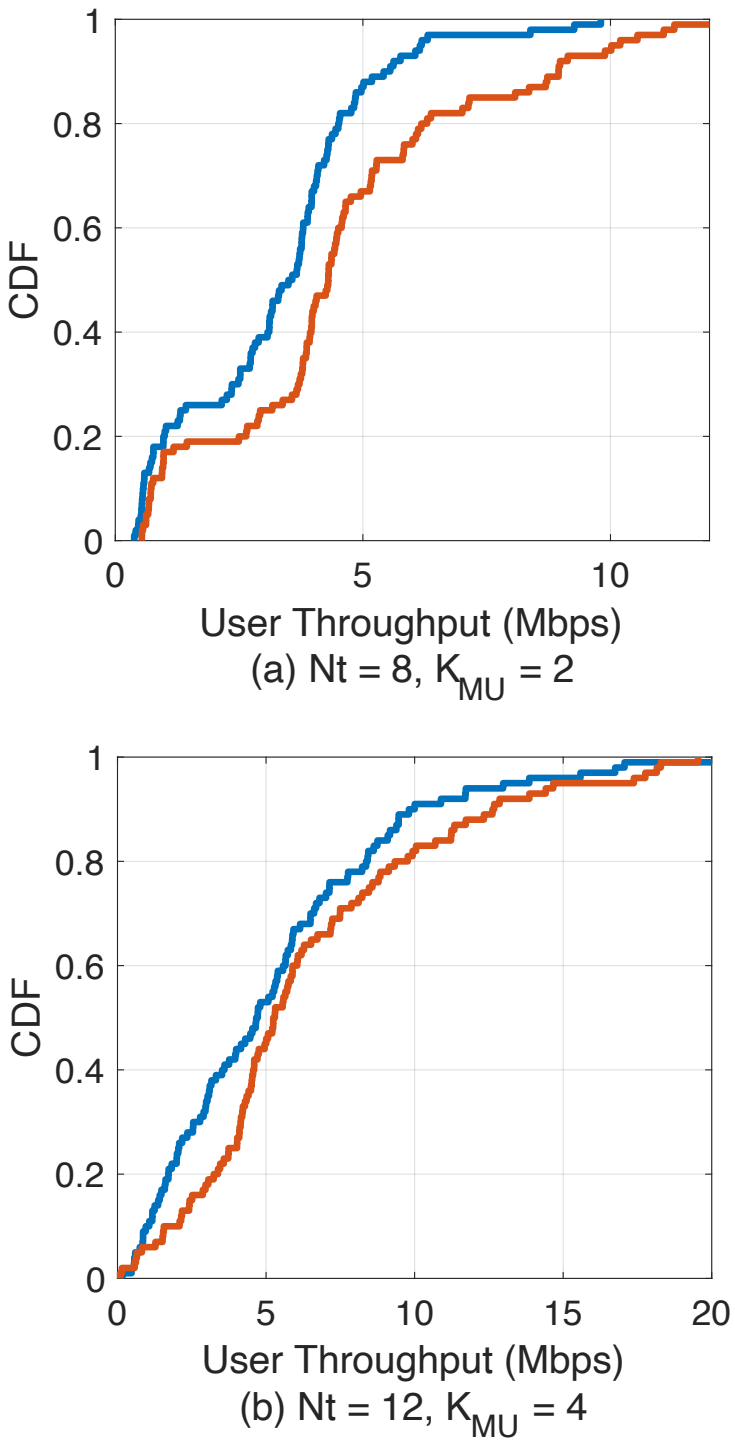


Figure 6.11: Comparison of CDFs of users' long-term average throughput.

Table 6.3: Comparison of each cell-edge users' throughput when $N_{\mathrm{T}} = 8, K_{\mathrm{MU}} = 2$.

Cell-edge user	Without JT (Mbps)	With JT (Mbps)	Gain
1	2.90	4.07	40%
2	4.50	4.37	-3%
3	1.26	4.61	266%
4	2.15	3.78	76%
5	3.11	3.97	28%
6	0.38	4.31	1030%
7	0.41	4.76	1054%
8	3.88	3.57	-8%
9	0.97	6.11	530%
10	1.42	4.50	216%
11	0.97	6.07	525%
12	3.07	3.97	29%
13	1.30	4.57	250%
14	3.12	8.95	187%
15	2.25	3.92	74%
16	2.77	3.69	33%
17	2.52	3.85	53%
Average	2.18	4.65	113%

5th to 50th percentile) with and without joint transmission for the setting (a) $N_{\mathrm{T}} = 8, K_{\mathrm{MU}} = 2$. The results suggest that the user throughput can be significantly improved by employing joint transmission. Specifically, the user throughput is increased by at least 20% at all examined percentiles and can be up to 160% for some cases. We have also examined the performance for the setting (b) $N_{\mathrm{T}} = 12, K_{\mathrm{MU}} = 4$ and have a similar observation.

In Table 6.3, we study the throughput performance for each cell-edge user in $\mathcal{K}^{\mathrm{E}}$ under setting (a) $N_{\mathrm{T}} = 8, K_{\mathrm{MU}} = 2$. Among the 17 cell-edge users, 15 users achieved a much higher throughput after joint transmission is employed, while only two users experienced a marginal decrease. By examining those two users in detail, we find that they can already achieve high throughput without joint transmission. Therefore, they would be better off not being classified as cell-edge users in Stage I. Even under such a "mis-classification", $\mathbf{M}^3$ is

Table 6.4: Comparison of average user throughput under different settings.

Setting	Without JT (Mbps)	With JT (Mbps)	Gain
(a) $N_{\mathrm{T}} = 8$	3.25	4.58	40.5%
(b) $N_{\mathrm{T}} = 12$	5.33	6.45	20.9%

able to offer comparable high throughput for these users. The most significant throughput improvement ($\sim 10\times$) is observed at user 7. This is because user 7 is closely located to both of its RRHs, and the distances between user 7 and each RRH are almost identical (see, Fig. 6.7). Therefore, user 7 receives strong and similar signal strength from its RRHs and can benefit much from joint transmission. On average, the throughput performance for cell-edge users $\mathcal{K}^{\mathrm{E}}$ is increase by 113% in our case study.

In Table 6.4, we show the average throughput of all 100 users. Table 6.4 shows that the average user throughput is improved by 40.5% and 20.9% through joint transmission under the two settings. The detailed experimental results in Tables 6.2, 6.3 and 6.4 demonstrate that $\mathbf{M}^3$ can deliver the desired throughput improvement under C-RAN.

6.7 Related Work

C-RAN Schedulers C-RAN's ability to better manage inter-cell interference has attracted much attention in the research community. For example, the designs in [162, 163, 164, 165, 166, 167, 168, 169] developed coordinated scheduling/beamforming schemes for multi-cell systems. But none of them has considered *actual* running time of their algorithm (in "wall-clock" time), which is the ultimate benchmark in practice. In particular, many algorithms in the literature (see, e.g., [162, 163, 164, 165]) are based on an iterative optimization (each iteration includes an optimization problem to solve). These designs cannot be applied to

practical cellular systems due to their poor real-time performance. In addition, prior works did not jointly optimize the RB allocation, MCS assignment and beamforming matrices for a multi-cell system as we did in this chapter. For instance, the designs in [162, 163, 164, 165, 166] developed cooperative scheduling or beamforming schemes for multi-cells without the consideration of RB allocation or MCS assignment. In [167], RB allocation was not considered in their models, and MCS selection was not considered in [168, 169].

Single-cell Schedulers In the literature, there have been active research works on the design of 5G schedulers for a single cell [95, 119, 131, 138]. These designs can offer (real-time or non-real-time) scheduling solutions at a traditional BS (serving a single cell). However, none of them can take advantage of the potential cooperation in C-RAN, such as joint transmission by multiple cells.

GPU-based Real-time Designs Applying GPU to solve complex optimization problems is not new. Indeed, recent years have witnessed a number of successful research works that leveraged GPU's large-scale parallel computation capability (see, e.g., [95, 119, 120, 146]). For example, the authors in [95, 119] designed real-time schedulers for a single cell based on GPU platform. In [146], the authors employ GPU to accelerate LDPC decoding. The work in [120] studied a MIMO detection problem based on a parallel design. Our GPU-based design and implementation are inspired by these prior arts. However, the problem that we studied in this chapter is new and has never been studied in these previous efforts.

6.8 Chapter Summary

This chapter presents $\mathbf{M}^3$—the first real-time scheduler for a multi-cell MIMO system under C-RAN architecture. $\mathbf{M}^3$ jointly optimizes RB allocation, MCS assignment, and beamforming matrices for all users under all RRHs and is able to offer a solution within 1 ms. To

address the stringent real-time requirement, we developed a novel multi-pipeline design that exploits large-scale parallelism. For validation, we implemented $\mathbf{M}^3$ on a COTS Nvidia DGX Station. Through extensive experiments, we showed that $\mathbf{M}^3$ can find a scheduling solution within 1 ms for all tested cases, while it can significantly increase user throughput by leveraging joint transmission among neighboring cells.

Chapter 7

Summary and Future Work

7.1 Summary

In this dissertation, we studied many-antenna MIMO techniques from a networking perspective. As new knowledge and understanding of many-antenna MIMO at the PHY layer begin to emerge, there is a critical need to explore many fundamental problems in terms of throughput, latency, reliability, among others. The objective of this dissertation is to address the many-antenna MIMO networking research in two critical areas: (i) DoF-based modeling and (ii) real-time optimization.

This dissertation consists of two parts. In the first part (Chapters 2 and 3), we studied DoF-based modeling for MIMO networks and developed a new general model for DoF-based interference cancellation under general channel rank conditions. Based on our new DoF model, we explored how to efficiently allocate DoFs to improve network throughput. We summarize the main contributions and findings of this part as follows.

- In Chapter 2, we developed novel DoF models and theories under general channel rank conditions, with the rank of a MIMO channel given *a priori*. We showed that the existing works claiming unilateral DoF consumption is optimal no longer hold when channel rank is deficient (not full). We found that for IC, shared DoF consumption at both Tx and Rx nodes is the most efficient for DoF allocation. Further, we showed

that DoF consumption under the existing full-rank assumption is a special case of our generalized DoF model. Based on this theory, we explored DoF allocation in a general multi-link MIMO network by formulating a set of constraints to characterize a feasible DoF scheduling. Through extensive case studies, we showed that the general IC model can achieve larger feasible DoF regions or improved objective values than existing unilateral IC models.

- In Chapter 3, we studied how to set channel ranks and exploited efficient DoF utilization. We observed that, in addition to the fact that channel is not full-rank, the strength of signals on different directions in the eigenspace is extremely uneven. This offers a much more general approach to define rank-deficiency, comparing to deficiency being defined in a strictly zero-signal sense. We introduced a novel concept called "effective rank threshold." Based on this threshold, we proposed efficient DoF utilization on an interference link. Specifically, DoFs are consumed only to cancel strong interferences in the eigenspace while weak interferences are treated as noise in throughput calculation. To better understand the benefits of this approach, we studied the fundamental trade-off between network throughput and the effective rank threshold for an MU-MIMO network. Our simulation results showed that network throughput under the optimal rank threshold is significantly higher than that under existing DoF IC models.

In the second part (Chapters 4, 5 and 6), we offered real-time designs and implementations to solve many-antenna MIMO problems for 5G cellular systems. We studied three critical MIMO problems for 5G—hybrid beamforming, MU-MIMO scheduling, and joint transmission under C-RAN architecture. All our solutions offered in this part were validated on COTS GPU and examined by wall-clock time. A brief summary of these three chapters is given below:

- In Chapter 4, we studied the beamforming problem under the HB architecture. The objective was to offer a beamforming solution in real-time (sub-ms) with desired throughput performance. To address this problem, we presented Turbo-HB, an ultra-fast beamforming design under HB architecture. To reduce the computation time, we developed low-complexity SVD by exploiting the randomized SVD technique and leveraging channel sparsity at mmWave frequencies. Further, we developed fully functioning parallelism for Turbo-HB, with optimized matrix operations and minimized memory accesses. We validated Turbo-HB by implementing it on a COTS Nvidia GPU. Extensive experiments were performed to examine both the timing performance and throughput performance. Our experimental results showed that Turbo-HB is able to find beamforming matrices successfully in $\sim$500 μs. Turbo-HB also offers competitive or higher throughput performance compared with state-of-the-art algorithms.

- In Chapter 5, we investigated a scheduling problem in 5G MU-MIMO system. The scheduler needs to determine RB allocation, number of data streams and MCS assignment for each user in each TTI. The real-time requirement for determining a scheduling solution is at most 1 ms. To address this challenge, we presented mCore+—the first 5G MU-MIMO scheduler that achieves 500-μs scheduling. To accelerate computation, mCore+ consists of a multi-phase optimization, leveraging large-scale parallel computation. In each phase, mCore+ either decomposes the optimization problem into a number of independent sub-problems, or reduces the search space into a smaller but most promising subspace, or both. We implemented mCore+ on a COTS GPU platform. Experimental results showed that mCore+ can obtain a scheduling solution in $\sim$500 μs. At the same time, mCore+ is able to offer a better or comparable throughput performance compared with other state-of-the-art algorithms.

- In Chapter 6, we studied the scheduling problem for a multi-cell MIMO system un-

der C-RAN architecture. Our objective was to jointly optimize RB allocation, MCS assignment, and beamforming matrices for all users under all RRHs so that the PF objective is maximized. In addition, we aimed to find a scheduling solution within each TTI (i.e., at most 1 ms) to conform to the frame structure defined by 5G NR. We proposed $\mathbf{M}^3$—a novel multi-pipeline design that exploits large-scale parallelism. Under $\mathbf{M}^3$, one pipeline performs a sequence of operations for cell-edge users to explore joint transmission, and in parallel, the other pipeline is for cell-center users to explore MU-MIMO transmission. We implemented $\mathbf{M}^3$ on a COTS GPU. Experimental results showed that $\mathbf{M}^3$ is capable of offering a scheduling solution within 1 ms for a C-RAN system. Meanwhile, $\mathbf{M}^3$ offers $\sim$40% throughput gain on average by employing joint transmission among multiple cells.

7.2 Future Work

MIMO technology remains to be the core of modern wireless communications and continues to evolve at a fast pace. Our work in this dissertation advances many-antenna MIMO techniques for networking research. Research in this area is still limited and there are many open problems that need to be explored. We outline some open problems from this dissertation as follows.

- **Open problems from Chapter 3 (efficient DoF utilization).** In Chapter 3, we introduced the concept of "effective rank threshold". Based on this concept, we proposed efficient DoF utilization on an interference link, aiming at conserving DoF and maximizing throughput. One limitation of the proposed approach in Chapter 3 is that we only focused on using conserved DoFs for SM (i.e., supporting more data streams), but did not consider to DoFs for diversity. That is, we did not explore the

SM-diversity trade-off in this chapter. We expect there exists an optimal trade-off on SM-diversity beyond IC based on effective channel rank. Given that DoFs can also be used for diversity (instead of SM) to increase throughput, a future research direction is how to allocate DoFs for diversity, in addition to SM and IC. The intricate dependency of these variables (effective rank setting, DoF allocation for diversity, SM, and IC) and their unique impacts on throughput make the overall problem both challenging and intriguing.

- **Open problems from Chapter 4 (ultra-fast hybrid beamforming).** Our work in Chapter 4 is the first effort that achieves real-time beamforming with high throughput performance under the hybrid architecture. Some open problems are listed as follows. First, our study focused on digital beamforming with given analog beamforming. But in some applications, such as tracing fast-moving mobile devices, analog beamforming also has a very stringent timing requirement. Therefore, a real-time solution of analog beamforming (or joint analog and digital beamforming) for these applications deserves future research. Second, designing a beamforming scheme with limited feedback/CSI is an important issue. Due to a large number of antennas for mmWave systems, the estimation and feedback of the full channel require a prohibitively large amount of CSI that is difficult to obtain in practice. The consideration of limited CSI can help with a more accurate modeling and a more robust solution. As expected, it will also add more complexity to the beamforming design. How to design a real-time beamforming solution with limited CSI remains an open problem.

- **Open problems from Chapters 5 and 6 (real-time 5G schedulers).** The results in Chapters 5 and 6 offered real-time schedulers for 5G systems, with a focus on scheduling RB resources, MCS, and MIMO users. Although beamforming matrices are calculated and applied at BSs and users, we employed simple linear beamforming

techniques, such as ZF and MMSE. There are opportunities to improve the throughput performance by employing a more advanced beamforming technique. For example, the weighted minimum mean square error (WMMSE) algorithm [101, 162] can offer near-optimal beamforming solutions to a weighted sum-rate maximization problem. However, WMMSE algorithm is a non-linear beamforming scheme based on iterative optimization, which is very challenging to be implemented in real-time. How to offer a near-optimal beamforming solution (in terms of maximizing throughput) and address its real-time challenge remains an open problem.

Bibliography

[1] IEEE 802.11n, "IEEE standard for information technology-telecommunications and information exchange between systems-Local and metropolitan area networks–Specific requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications-Amendment 5: Enhancements for Higher Throughput," *IEEE Standards 802.11n*, Oct. 2009.

[2] IEEE 802.11ac, "IEEE standard for information technology-telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications-Amendment 4: Enhancements for Very High Throughput for Operation in Bands below 6 GHz," *IEEE Standards 802.11ac*, Dec. 2013.

[3] 3GPP TS 36.213 version 10.1.0 Release 10, "Evolved universal terrestrial radio access (E-UTRA); Physical layer procedures," *3GPP Standards*, Apr. 2011.

[4] 3GPP TS 38.214 version 16.0.0 Release 16, "NR; Physical layer procedures for data," *3GPP Standards*, Jan. 2020.

[5] P.A. Eliasi, S. Rangan, and T.S. Rappaport, "Low-rank spatial channel estimation for millimeter wave cellular systems," *IEEE Transactions on Wireless Communications*, vol. 16, no. 5, pp. 2748–2759, Apr. 2017.

[6] H. Yin, D. Gesbert, M. Filippou, and Y. Liu, "A coordinated approach to channel estimation in large-scale multiple-antenna systems," *IEEE Journal on Selected Areas in Communications*, vol. 31, no, 2, pp. 264–273. Feb. 2013.

[7] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Chapter 1, New York, NY, USA: Freeman, 1990.

[8] Y. Chen, Y. Huang, Y. Shi, Y.T. Hou, W. Lou and S. Kompella, "A general model for DoF-based interference cancellation in MIMO Networks with rank-deficient channels," in *Proc. of IEEE INFOCOM*, pp. 900–907, Honolulu, HI, USA, April 2018.

[9] D. Tse and P. Viswanath, *Fundamentals of wireless communication,* Chapter 7, Cambridge University Press, 2005, ISBN: 9780521845274

[10] L. Zheng and D.N.C. Tse, "Diversity and multiplexing: A fundamental tradeoff in multiple-antenna channels," *IEEE Transactions on Information Theory*, vol. 49, no. 5, pp. 1073–1096, May 2003.

[11] A. Host-Madsen and A. Nosratinia, "The multiplexing gain of wireless networks," in *Proc. of IEEE ISIT*, pp. 2065–2069, Adelaide, Australia, Sept. 2005.

[12] S.A. Jafar and M.J. Fakhereddin, "Degrees of freedom for the MIMO interference channel," *IEEE Transactions on Information Theory*, vol. 53 no. 7, pp. 2637–2642, July 2007.

[13] L. Zheng and D.N.C. Tse, "Communication on the Grassmann manifold: A geometric approach to the noncoherent multiple-antenna channel," *IEEE Transactions on Information Theory*, vol. 48, no. 2, pp.359–383, Feb. 2002.

[14] R. Bhatia and L. Li, "Throughput optimization of wireless mesh networks with MIMO links," in *Proc. of IEEE INFOCOM*, pp. 2326–2330, Barcelona, Spain, May 2007.

[15] D.M. Blough, G. Resta, P. Santi, R. Srinivasan, and L. M. Cortés-Pena, "Optimal one-shot scheduling for MIMO networks," in *Proc. of IEEE SECON*, pp. 404–412, Salt Lake City, UT, USA, June 2011.

[16] H. Zeng, Y. Shi, Y.T. Hou, R. Zhu, and W. Lou, "A novel MIMO DoF model for multi-hop networks," *IEEE Network*, vol. 28, no. 5, pp. 81–85, Sept. 2014.

[17] H. Yu, O. Bejarano and L. Zhong, "Combating inter-cell interference in 802.11ac-based multi-user MIMO networks," in *Proc. of ACM MobiCom*, pp. 141–152, Maui, Hawaii, USA, Sept. 2014.

[18] J.-S. Park, A. Nandan, M. Gerla, and H. Lee, "SPACE-MAC: Enabling spatial reuse using MIMO channel-aware MAC," in *Proc. of IEEE ICC*, pp. 3642–3646, Seoul, South Korea, May 2005.

[19] J.C. Mundarath, P. Ramanathan, and B.D. Van Veen, "Exploiting spatial multiplexing and reuse in multi-antenna wireless ad hoc networks," *Elsevier Ad Hoc Networks*, vol. 7, no. 2, pp. 281–293, Mar. 2009.

[20] B. Hamdaoui and K.G. Shin, "Characterization and analysis of multi-hop wireless MIMO network throughput," in *Proc. of ACM MobiHoc*, pp. 120–129, Montreal, Quebec, Canada, Sept. 2007.

[21] Y. Shi, J. Liu, C. Jiang, C. Gao, and Y.T. Hou, "A DoF-based link layer model for multi-hop MIMO networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 7, pp. 1395–1408, July 2014.

[22] D.M. Blough, P. Santi, and R. Srinivasan, "On the feasibility of unilateral interference cancellation in MIMO networks," *IEEE/ACM Transactions on Networking*, vol. 22, no. 6, pp. 1831–1844, Dec. 2014.

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

[24] A. Wiesel, Y.C. Eldar and S. Shamai, "Zero-forcing precoding and generalized inverses," *IEEE Transactions on Signal Processing*, vol. 56, no. 9, pp. 4409–4418, Sept. 2008.

[25] L. Yang and W. Zhang, "On degrees of freedom region of three-user MIMO interference channels," *IEEE Transactions on Signal Processing*, vol. 63, no. 3, pp. 590–603, Feb. 2015.

[26] S.A. Jafar and S. Shamai, "Degrees of freedom region of the MIMO X channel," *IEEE Transactions on Information Theory*, vol. 54, no. 1, pp. 151–170. Jan. 2008.

[27] S.R. Krishnamurthy, A. Ramakrishnan, and S.A. Jafar, "Degrees of freedom of rank-deficient MIMO interference channels," *IEEE Transactions on Information Theory*, vol. 61, no. 1, pp. 341–365, Jan. 2015.

[28] A.G. Burr, "Capacity bounds and estimates for the finite scatterers MIMO wireless channel," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 5, pp. 812–818, June 2003.

[29] M. Nicoli, "Multiuser reduced rank receivers for TD/CDMA systems," *Ph.D. dissertation*, Dipartimento di Elettronica e Informazione, Politecnico di Milano, Milano, Italy, 2001.

[30] H. Shin and J.H. Lee, "Capacity of multiple-antenna fading channels: Spatial fading correlation, double scattering, and keyhole," *IEEE Transactions on Information Theory*, vol. 49, no. 10, pp. 2636–2647, Oct. 2003.

[31] D. Gesbert, H. Bolcskei, D.A. Gore, and A.J. Paulraj, "Outdoor MIMO wireless channels: models and performance prediction," *IEEE Transactions on Communications*, vol. 50, no. 12, pp. 1926–1934, Dec. 2002.

[32] FCC Eyes Midband Spectrum Between 3.7 and 24 GHz. [Online]. Available: http://www.fiercewireless.com/wireless/fcc-eyes-mid-band-spectrum-between-3-7-and-24-ghz?

[33] FCC Opens Inquiry into New Opportunities in Mid-Band Spectrum. [Online]. Available: https://www.fcc.gov/document/fcc-opens-inquiry-new-opportunities-mid-band-spectrum-0

[34] Y. Zeng, X. Xu, Y.L. Guan, E. Gunawan, and C. Wang, "Degrees of freedom of the three-user rank-deficient MIMO interference channel," *IEEE Transactions on Wireless Communications*, vol. 13, no. 8, pp. 4179–4192, Aug. 2014.

[35] B. Yuan, H. Sun and S. A. Jafar, "Replication-based outer bounds: on the optimality of "half the cake" for rank-deficient MIMO interference networks," *IEEE Transactions on Information Theory*, vol. 63, no. 10, pp. 6607–6621, Oct. 2017.

[36] H. Sun, S.R. Krishnamurthy, and S.A. Jafar, "Rank matching for multihop multiflow," *IEEE Transactions on Information Theory*, vol. 61, no. 9, pp. 4751–4764, June 2015.

[37] H. Sun, C. Geng, T. Gou and S.A. Jafar, "Degrees of freedom of MIMO X networks: Spatial scale invariance, one-sided decomposability and linear feasibility," in *Proc. of IEEE ISIT*, pp. 2082–2086, Cambridge, MA, Aug. 2012.

[38] S.H. Chae, S.W. Jeon and S.Y. Chung, "Cooperative relaying for the rank-deficient MIMO relay interference channel," *IEEE Communications Letters*, vol. 16, no. 1, pp. 9–11, Nov. 2012.

[39] J. Fanjul, Ó. González, I. Santamaria and C. Beltrán, "Homotopy continuation for spatial interference alignment in arbitrary MIMO X Networks," *IEEE Transactions on Signal Processing*, vol. 65, no. 7, pp. 1752–1764, April, 2017.

[40] V. Cadambe and S. Jafar, "Interference alignment and degrees of freedom of the K-user interference channel," *IEEE Transactions on Information Theory*, vol. 54, no. 8, pp. 3425–3441, August 2008.

[41] C.M. Yetis, T. Gou, S. A. Jafar and A.H. Kayran, "On feasibility of interference alignment in MIMO interference networks," *IEEE Transactions on Signal Processing*, vol. 58, no. 9, pp. 4771–4782, Sept. 2010.

[42] H. Zeng, F. Tian, Y.T. Hou, W. Lou, and S.F. Midkiff, "Interference alignment for multi-hop wireless networks: Challenges and research directions," *IEEE Network*, vol. 30, no. 2, pp. 74–80, March/April 2016.

[43] G. Matsaglia and G. PH Styan, "Equalities and inequalities for ranks of matrices," *Linear and Multilinear Algebra*, vol. 2, no. 3, pp. 269–292, 1974

[44] H. Zeng, Y. Shi, Y.T. Hou, W. Lou, S. Kompella and S.F. Midkiff, "An analytical model for interference alignment in multi-hop MIMO networks," *IEEE Transactions on Mobile Computing*, vol. 15, no. 1, pp. 17–31, Mar. 2016.

[45] H.D. Sherali and W.P. Adams, *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems,* Springer, Dec. 2010, ISBN-13: 978-1441948083.

[46] Y.T. Hou, Y. Shi and H.D. Sherali, *Applied Optimization Methods for Wireless Networks,* Chapter 6, Cambridge University Press, May 2014, ISBN-13: 978-1107018808.

[47] M. Herceg, M. Kvasnica, C.N. Jones, and M. Morari, "Multi-parametric toolbox 3.0," in *Proc. of European Control Conference*, pp. 502–510, Zurich, Switzerland, July, 2013. Available: http://control.ee.ethz.ch/~mpt

[48] IBM. "Branch and cut in CPLEX." Available: https://www.ibm.com/support/knowledgecenter/SSSA5P_12.5.1/ilog.odms.cplex.help/refcppcplex/html/branch.html

[49] Y. Chen, S. Li, C. Li, Y.T. Hou and B. Jalaian, "To cancel or not to cancel: Exploiting interference signal strength in the eigenspace for efficient MIMO DoF utilization," in *Proc. of IEEE INFOCOM*, pp. 1954–1962, Paris, France, April 29–May 2, 2019.

[50] K. Sundaresan, R. Sivakumar, M.A. Ingram and C. Tae-Young, "Medium access control in ad hoc networks with MIMO links: Optimization considerations and algorithms," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 350–365, Oct. 2004.

[51] X. Xie, X. Zhang and E. Chai, "Cross-cell DoF distribution: Combating channel hardening effect in multi-cell MU-MIMO networks," in *Proc. of ACM MobiHoc,* pp. 337–346, Hangzhou, China, June 2015.

[52] S. Kumar, D. Cifuentes, S. Gollakota and D. Katabi, "Bringing cross-layer MIMO to today's wireless LANs," in *Proc. of ACM SIGCOMM*, pp. 387–398, Hong Kong, China, Aug. 2013.

[53] O. Bejarano, E. Magistretti and O. Gurewitz, "Mute: Sounding inhibition for MU-MIMO WLANs," in *Proc. of IEEE SECON*, pp. 135–143, Singapore, June 30–July 3, 2014.

[54] X. Xie and X. Zhang, "Scalable user selection for MU-MIMO networks," in *Proc. of IEEE INFOCOM*, pp. 808–816, Toronto, Canada, April 27–May 2, 2014.

[55] C. Shepard, H. Yu, N. Anand, E. Li, T. Marzetta, R. Yang and L. Zhong, "Argos: Practical many-antenna base stations," in *Proc. of ACM MobiCom*, pp. 53–64, Istanbul, Turkey, Aug. 2012.

[56] G.S. Smith, "A direct derivation of a single-antenna reciprocity relation for the time domain," *IEEE Transactions on Antennas and Propagation*, vol. 52, no. 6, pp. 1568–1577, June 2004.

[57] Ettus Research, "Software-defined radio device: USRP N210," available: www.ettus.com/product/details/UN210-KIT

[58] Ettus Research, "OctoClock-G CDA-2990," available: www.ettus.com/product/details/OctoClock-G

[59] E. Blossom, "GNU Radio: Tools for exploring the radio frequency spectrum," *Linux Journal*, vol. 2004, no. 122, pp. 4, 2004.

[60] R.H. Etkin, N.C. David and H. Wang, "Gaussian interference channel capacity to within one bit," *IEEE Transactions on Information Theory*, vol. 54, no. 12, pp. 5534–5562, Dec. 2008.

[61] F. Negro, S.P. Shenoy, I. Ghauri and D.T. Slock, "On the MIMO interference channel," in *Proc. of IEEE Information Theory and Applications Workshop*, pp. 1–9, San Diego, CA, USA, Jan. 2010.

[62] J.P. Kermoal, L. Schumacher, K.I. Pedersen, P.E. Mogensen and F. Frederiksen, "A stochastic MIMO radio channel model with experimental validation," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 6, pp. 1211–1226, Aug. 2002.

[63] K. Yu, M. Bengtsson, B. Ottersten, D. McNamara, P. Karlsson, and M. Beach, "Modeling of wide-band MIMO radio channels based on NLoS indoor measurements," *IEEE Transactions on Vehicular Technology*, vol. 53, no. 3, pp. 655–665, May 2004.

[64] R.K. Mallik, "The exponential correlation matrix: Eigen-analysis and applications,"

IEEE Transactions on Wireless Communications, vol. 17, no. 7, pp. 4690–4705, July 2018.

[65] J. Choi and D.J. Love, "Bounds on eigenvalues of a spatial correlation matrix," *IEEE Communications Letters*, vol. 18 no. 8, pp. 1391–1394, Aug. 2014.

[66] C. Eckart, and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, 1936.

[67] Y. Chen, Y. Huang, Y. Shi, Y.T. Hou, W. Lou and S. Kompella, "On DoF-based interference cancellation under general channel rank conditions," *IEEE/ACM Transactions on Networking*, vol. 28, no. 3, pp. 1002–1016, June 2020.

[68] F.P. Kelly, A.K. Maulloo and D.K.H. Tan, "Rate control for communication networks: Shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, no. 3, pp. 237-252, April 1998.

[69] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, Chapter 3, SIAM, 2001. ISBN: 9780898714913.

[70] Gurobi Optimization, Inc. "Gurobi optimizer reference manual," 2018. Available: http://www.gurobi.com

[71] B. Hochwald and S. Vishwanath, "Space-time multiple access: Linear growth in the sum rate," in *Proc. 40th Annual Allerton Conf. Communications, Control and Computing*, Monticello, IL, USA, Oct. 2002.

[72] Q. Wang and Y. Jing, "New rank detection methods for reduced-rank MIMO systems," *EURASIP Journal on Wireless Communications and Networking*, vol. 2015, no. 1, pp. 230, Oct. 2015.

[73] M. Nicoli and U. Spagnolini, "Reduced-rank channel estimation for time-slotted mobile communication systems," *IEEE Transactions on Signal Processing*, vol. 53, no. 3, pp. 926-944, March 2005.

[74] C.R. Johnson and J. A. Link, "Solution theory for complete bilinear systems of equations," *Numerical Linear Algebra with Applications*, vol. 16, no. 11–12, pp. 929–934, Nov./Dec. 2009.

[75] K. Gomadam, V.R. Cadambe and S.A. Jafar, "Approaching the capacity of wireless networks through distributed interference alignment," in *Proc. of IEEE GLOBECOM*, pp. 1–6, New Orleans, LO, USA, Nov 30–Dec 4, 2008.

[76] S.W. Peters and R.W. Heath, "Cooperative algorithms for MIMO interference channels," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 1, pp. 206–218, Oct. 2011.

[77] M. Razaviyayn, G. Lyubeznik and Z.Q. Luo, "On the degrees of freedom achievable through interference alignment in a MIMO interference channel," *IEEE Transactions on Signal Processing*, vol. 60 no. 2, pp.812–821, Feb. 2012.

[78] W. Hong, K. Baek, Y. Lee, Y. Kim and S. Ko, "Study and prototyping of practically large-scale mmWave antenna systems for 5G cellular devices," *IEEE Communications Magazine*, vol. 52, no. 9, pp. 63–69, Sept. 2014.

[79] S. Han, C. I, Z. Xu and C. Rowell, "Large-scale antenna systems with hybrid analog and digital beamforming for millimeter wave 5G," *IEEE Communications Magazine*, vol. 53, no. 1, pp. 186–194, Jan. 2015.

[80] A.F. Molisch, V.V. Ratnam, S. Han, Z. Li, S.L.H. Nguyen, L. Li and K. Haneda, "Hy-

brid beamforming for massive MIMO: A survey," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 134–141, Dec. 2017.

[81] W. Ni, X. Dong and W.S. Lu, "Near-optimal hybrid processing for massive MIMO systems via matrix decomposition," *IEEE Transactions on Signal Processing*, vol. 65, no. 15, pp. 3922–3933, Aug. 2017.

[82] O. El Ayach, S. Rajagopal, S. Abu-Surra, Z. Pi and R.W. Heath Jr., "Spatially sparse precoding in millimeter wave MIMO systems," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 3, pp. 1499–1513, March 2014.

[83] T.E. Bogale and L.B. Le, "Beamforming for multiuser massive MIMO systems: Digital versus hybrid analog-digital," in *Proc. of IEEE GLOBECOM*, pp. 4066–4071, Austin, TX, Dec. 2014.

[84] C. Rusu, R. Mendez-Rial, N. Gonzalez-Prelcic and R.W. Heath, "Low complexity hybrid precoding strategies for millimeter wave communication systems," *IEEE Transactions on Wireless Communications*, vol. 15, no. 12, pp. 8380–8393, Dec. 2016.

[85] A. Alkhateeb, J. Mo, N. Gonzalez-Prelcic and R.W. Heath, "MIMO precoding and combining solutions for millimeter-wave systems.," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 122–131, Dec. 2014.

[86] W. Ni and X. Dong, "Hybrid block diagonalization for massive multiuser MIMO systems". *IEEE Transactions on Communications*, vol. 64, no. 1, pp. 201–211, Jan. 2016.

[87] L. Liang, W. Xu and X. Dong, "Low-complexity hybrid precoding in massive multiuser MIMO systems," *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 653–656, Dec. 2014.

[88] Y. Ghasempour, M.K. Haider, C. Cordeiro, D. Koutsonikolas and E. Knightly, "Multi-stream beam-training for mmWave MIMO networks," in *Proc. of ACM MobiCom*, pp. 225–239, New Delhi, India, Oct. 2018.

[89] Y. Ghasempour and E.W. Knightly, "Decoupling beam steering and user selection for scaling multi-user 60 GHz WLANs." in *Proc. of ACM MobiHoc*, pp. 1–10, Chennai, India, July 2017.

[90] S. Sur, I. Pefkianakis, X. Zhang, and K.H. Kim, "Towards scalable and ubiquitous millimeter-wave wireless networks," in *Proc. of ACM MobiCom*, pp. 257–271, New Delhi, India, Oct. 2018.

[91] 3GPP TS 38.211 version 16.2.0, "NR; physical channels and modulation." Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3213

[92] S. K. Mohammed and E. G. Larsson, "Improving the performance of the zero-forcing multiuser MISO downlink precoder through user grouping," *IEEE Transactions on Wireless Communications*, vol. 15, no. 2, pp. 811–826, Feb. 2016.

[93] V. Stankovic and M. Haardt, "Multi-user MIMO downlink precoding for users with multiple antennas," *Wireless World Research Forum*, pp. 12–14, Toronto, ON, Canada, Nov. 2004.

[94] D. Patil, "Block diagonalization based beamforming," *Master Thesis*, Department of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden, 2017.

[95] Y. Huang, S. Li, Y.T. Hou and W. Lou, "GPF: A GPU-based design to achieve ∼100 μs scheduling for 5G NR," in *Proc. of ACM MobiCom*, pp. 207–222, New Delhi, India, Oct. 2018.

[96] Y. Huang, S. Li, Y. Chen, Y.T. Hou, W. Lou, J. Delfeld, V. Ditya, "GPU: A new enabling platform for real-time optimization in wireless networks," *IEEE Network*, vol. 34, no. 6, pp. 77–83, Nov./Dec. 2020.

[97] 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 Transactions on Signal Processing*, vol. 54, no. 9, pp. 3658–3663, Sept. 2006.

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

[99] W. Yang, G. Durisi and E. Riegler, "On the capacity of large-MIMO block-fading channels," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 2, pp. 117–132, Feb. 2013.

[100] A. Alkhateeb, G. Leus and R.W. Heath, "Limited feedback hybrid precoding for multi-user millimeter wave systems," *IEEE Transactions on Wireless Communications*, vol. 14, no. 11, pp. 6481–6494, July 2015.

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

[102] A. Zhou, X. Zhang and H. Ma, "Beam-forecast: Facilitating mobile 60 GHz networks via model-driven beam steering," in *Proc. of IEEE INFOCOM*, Atlanta, GA, pp. 1–9, May 2017.

[103] W. Roh, J.Y. Seol, J. Park, B. Lee, J. Lee, Y. Kim, J. Cho, K. Cheun and F. Aryanfar, "Millimeter-wave beamforming as an enabling technology for 5G cellular communica-

tions: theoretical feasibility and prototype results," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 106–113, Feb. 2014.

[104] M.K. Haider, Y. Ghasempour, D. Koutsonikolas, and E.W. Knightly, "LiSteer: mmWave beam acquisition and steering by tracking indicator LEDs on wireless APs," in *Proc. of ACM MobiCom*, pp. 273–288, New Delhi, India, Nov. 2018.

[105] E. Bjornson, L. Van der Perre, S. Buzzi and E.G. Larsson, "Massive MIMO in sub-6 GHz and mmWave: Physical, practical, and use-case differences," *IEEE Wireless Communications*, vol. 26, no. 2, pp. 100–108, April 2019.

[106] V. Raghavan, A. Partyka, A. Sampath, S. Subramanian, O.H. Koymen, K. Ravid, J. Cezanne, K. Mukkavilli and J. Li, "Millimeter-wave MIMO prototype: Measurements and experimental results," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 202–209, Jan. 2018.

[107] T.S. Rappaport, *Wireless communications: Principles and Practice*. New Jersey: Prentice hall PTR, 1996. ISBN: 9780133755367.

[108] N. Halko, P.G. Martinsson and J.A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM review*, vol. 53, no. 2, pp. 217–288, May 2011.

[109] M. Gu and S.C. Eisenstat, "Efficient algorithms for computing a strong rank-revealing QR factorization," *SIAM Journal on Scientific Computing*, vol. 17, no. 4, pp. 848–869, July 1996.

[110] G.H. Golub and C.F. Van Loan, *Matrix Computations (4th Edition)*, Chapter 7, Johns Hopkins University Press, 2013. ISBN-13: 978-1421407944.

[111] N. Song, H. Sun and T. Yang, "Coordinated hybrid beamforming for millimeter wave multi-user massive MIMO systems," in *Proc. of IEEE GLOBECOM*, pp. 1–6, Washington, DC, Dec. 2016.

[112] T.S. Rappaport, E. Ben-Dor, J.N. Murdock and Y. Qiao, "38 GHz and 60 GHz angle-dependent propagation for cellular & peer-to-peer wireless communications," in *Proc. of IEEE ICC*, pp. 4568–4573, Ottawa, ON, June 2012.

[113] J. Blinn, "Consider the lowly 2 x 2 matrix," *IEEE Computer Graphics and Applications*, vol. 16, no. 2, pp. 82–88, March 1996.

[114] G.H. Golub and C.F. Van Loan, *Matrix Computations (4th Edition)*, Chapter 5, Johns Hopkins University Press, 2013. ISBN-13: 978-1421407944.

[115] Nvidia, "NVIDIA TESLA V100 GPU architecture." Available: https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf

[116] Nvidia, "Optimizing parallel reduction in CUDA." Available: https://developer.download.nvidia.com/assets/cuda/files/reduction.pdf

[117] Nvidia, "CUDA C programming guide v10.2.89." Available: https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html

[118] Y. Huang, Y. Chen, Y.T. Hou and W. Lou, "Achieving fair LTE/WiFi coexistence with real-time scheduling," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 1, pp. 366–380, March 2020.

[119] Y. Chen, Y. Wu, Y.T. Hou and W. Lou, "mCore: Achieving sub-millisecond scheduling for 5G MU-MIMO systems," in *Proc. of IEEE INFOCOM*, pp. 1–10, online conference, May 2021.

[120] C. Husmann, G. Georgis, K. Nikitopoulos and K. Jamieson, "FlexCore: Massively parallel and flexible processing for large MIMO access points," *USENIX Symposium on Networked Systems Design and Implementation,* pp. 197–211, Boston, MA, March 2017.

[121] G. Falcao, L. Sousa and V. Silva, "Massively LDPC decoding on multicore architectures," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 2, pp. 309–322, Feb. 2011.

[122] P. Hailes, L. Xu, R. G. Maunder, B. M. Al-Hashimi and L. Hanzo, "A survey of FPGA-based LDPC decoders," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1098–1122, 2016.

[123] E. Dahlman, S. Parkvall and J. Skold, J. *5G NR: The next generation wireless access technology.* Chapter 7, Academic Press, Aug. 2018. ISBN: 9780128143230

[124] A. Zaidi, F. Athley, J. Medbo, U. Gustavsson, G. Durisi, and X. Chen, *5G Physical Layer: Principles, Models and Technology Components.* Academic Press, Sept. 2018. ISBN: 9780128145784.

[125] Qualcomm, "Exploring 5G new radio: use cases, capabilities & Timeline." Available: https://www.qualcomm.com/media/documents/files/heavy-reading-whitepaper-exploring-5g-new-radio-use-cases-capabilities-timeline.pdf

[126] X. Lin, J. Li, R. Baldemair, J.F. Cheng, S. Parkvall, D.C. Larsson, H. Koorapaty, M. Frenne, S. Falahati, A. Grovlen and K. Werner, "5G new radio: Unveiling the essentials of the next generation wireless access technology," *IEEE Communications Standards Magazine*, vol. 3, no. 3, pp. 30–37, Sept. 2019.

[127] A. Ghosh, A. Maeder, M. Baker and D. Chandramouli, "5G Evolution: A View on

5G Cellular Technology Beyond 3GPP Release 15," *IEEE Access*, vol. 7, pp. 127639–127651, Sept. 2019.

[128] 3GPP TS 38.212 version 16.6.0, "NR; Multiplexing and channel coding." Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3214

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

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

[131] S. Lee, I. Pefkianakis, S. Choudhury, S. Xu and S. Lu, "Exploiting spatial, frequency, and multiuser diversity in 3GPP LTE cellular networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 11, pp. 1652–1665, Nov. 2012.

[132] H. Zhang, N. Prasad and S. Rangarajan, "MIMO downlink scheduling in LTE systems," in *Proc. of IEEE INFOCOM*, pp. 2936–2940, Orlando, FL, May 2012.

[133] H. Liao, P. Chen, and W. Chen, "An efficient downlink radio resource allocation with carrier aggregation in LTE-Advanced networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 10, pp. 2229–2239, Oct. 2014.

[134] R. Kwan, C. Leung, and J. Zhang, "Proportional fair multiuser scheduling in LTE," *IEEE Signal Processing Letters*, vol. 16, no. 6, pp. 461–464, June 2009.

[135] S. Lee, S. Choudhury, A. Khoshnevis, S. Xu and S. Lu, "Downlink MIMO with

frequency-domain packet scheduling for 3GPP LTE," in *Proc. of IEEE INFOCOM*, pp. 1269–1277, Rio de Janeiro, Brazil, Apr. 2009.

[136] A. Ragaleux, S. Baey and A. Fladenmuller, "An efficient and generic downlink resource allocation procedure for pre-5G networks," *Wireless Communications and Mobile Computing*, vol. 10, no. 17, pp. 3089–3103, Dec. 2016.

[137] R. Kwan, C. Leung, J. Zhang,. "Multiuser scheduling on the downlink of an LTE cellular system," *Research Letters in Communications*, vol. 2008, Jan. 2008.

[138] Y. Xu, H. Yang, F. Ren, C. Lin and X.S. Shen, "Frequency domain packet scheduling with MIMO for 3GPP LTE downlink," *IEEE Transactions on Wireless Communications*, vol. 12, no. 4, pp. 1752–1761, April 2013.

[139] G. Yue, N. Prasad and S. Rangarajan, "Downlink multiuser MIMO scheduling in LTE Advanced systems," in *Proc. of IEEE ICC*, pp. 4484–4488, Ottawa, ON, Canada, Jun 2012.

[140] J. Fan, G. Y. Li and X. Zhu, "Multiuser MIMO scheduling for LTE-A downlink cellular networks," in *Proc. of IEEE VTC Spring*, pp. 1–5, Seoul, South Korea, Jan. 2014.

[141] W. Guo, J. Fan, G.Y. Li, Q. Yin, X. Zhu, "Adaptive SU/MU-MIMO scheduling schemes for LTE-A downlink transmission," *Communications IET*, vol. 11, no. 6, pp. 783–792, April 2017.

[142] S. Roger, C. Ramiro, A. Gonzalez, V. Almenar and A. M. Vidal, "Fully parallel GPU implementation of a fixed-complexity soft-output MIMO detector," in *IEEE Transactions on Vehicular Technology*, vol. 61, no. 8, pp. 3796–3800, Oct. 2012.

[143] C. Li, Y. Huang, Y. Chen, B. Jalaian, Y.T. Hou, and W. Lou, "Kronos: A 5G sched-

uler for AoI minimization under dynamic channel conditions," in *Proc. IEEE ICDCS*, pp. 1466–1475, Dallas, TX, Jul. 2019.

[144] S. Li, Y. Huang, C. Li, B. Jalaian, S. Russell, Y.T. Hou, W. Lou and B. MacCall, "A real-time solution for underlay coexistence with channel uncertainty," in *Proc. of IEEE GLOBECOM*, Waikoloa, HI, Dec. 2019.

[145] Y. Chen, Y. Huang, C. Li, Y.T. Hou and W. Lou, "Turbo-HB: A novel design and implementation to achieve ultra-fast hybrid beamforming," in *Proc. IEEE INFOCOM 2020*, pp. 1489–1498, online conference, July 2020.

[146] G. Wang, M. Wu, Y. Sun and J.R. Cavallaro, "GPU accelerated scalable parallel decoding of LDPC dodes," in *Proc. of IEEE Asilomar Conference on Signals, Systems and Computers*, pp. 2053–2057, Pacific Grove, CA, Nov. 2011.

[147] Y. Huang, Y. Chen, Y.T. Hou and W. Lou, "Achieving fair LTE/WiFi coexistence with real-time scheduling," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 1, pp. 366–380, March 2020.

[148] Nortel Networks, "Nortel Networks' reference simulation methodology for the performance evaluation of OFDM/WCDMA in UTRAN," R1-03-0785, 3GPP TSG RAN WG1#33, New York, NY, USA, Aug. 2003.

[149] A. Stolyar, "On the asymptotic optimality of the gradient scheduling algorithm for multi-user throughput allocation," *Operations Research*, vol. 53, no. 1, pp. 12–25, 2005.

[150] K. Ko and J. Lee, "Multiuser MIMO user selection based on chordal distance," *IEEE Transactions on Communications*, vol. 60, no. 3, pp. 649–654, March 2012.

[151] Nvidia, "NVIDIA DGX Station." Available: https://images.nvidia.com/content/newsletters/email/pdf/DGX-Station-WP.pdf

[152] Nvidia, "Achieved Occupancy." Available: https://docs.nvidia.com/gameworks/content/developertools/desktop/analysis/report/cudaexperiments/kernellevel/achievedoccupancy.htm

[153] Nvidia, "CUDA toolkit documentation: cuRAND," Available: https://docs.nvidia.com/cuda/curand/index.html

[154] Fujitsu, "Evolving to an open C-RAN architecture for 5G." Available: https://www.fujitsu.com/us/Images/FNC-Fujitsu-Evolving-to-an-Open-C-RAN-Architecture-for-5G-White-Paper.pdf

[155] M. Peng, Y. Sun, X. Li, Z. Mao and C. Wang, "Recent advances in cloud radio access networks: System architectures, key techniques, and open issues," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2282–2308, 2016.

[156] A. Checko, H.L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M.S. Berger, and L. Dittmann, "Cloud RAN for mobile networks—A technology overview," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 405–426, First quarter 2015.

[157] C. I, J. Huang, R. Duan, C. Cui, J. Jiang and L. Li, "Recent progress on C-RAN centralization and cloudification," *IEEE Access*, vol. 2, pp. 1030–1039, Aug. 2014.

[158] Ericsson introduces Cloud RAN for 5G. Available: https://www.fiercewireless.com/operators/ericsson-introduces-cloud-ran-for-5g

[159] 3GPP TR 38.816 version 15.0.0, "Technical specification group radio access network; Study on CU-DU lower layer split for NR." Available: https://portal.3gpp.org/de

sktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3364

[160] 3GPP TR 38.801 version V14.0.0, "Technical specification group radio access network; Study on new radio access technology: Radio access architecture and interfaces." Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3056

[161] R. Fantini, W. Zirwas, L. Thiele, D. Aziz and P. Baracca, *5G Mobile and Wireless Communications Technology.* Chapter 9, Cambridge University Press, June 2016. ISBN: 9781107130098.

[162] Q. Shi, M. Razaviyayn, Z. Luo and C. He, "An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4331–4340, Sept. 2011.

[163] B. Dai and W. Yu, "Sparse beamforming and user-centric clustering for downlink cloud radio access network," *IEEE Access*, vol. 2, pp. 1326–1339, 2014.

[164] M. Hong, R. Sun, H. Baligh and Z. Luo, "Joint base station clustering and beamformer design for partial coordinated transmission in heterogeneous networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 2, pp. 226–240, Feb. 2013.

[165] C.T.K. Ng and H. Huang, "Linear precoding in cooperative MIMO cellular networks with limited coordination clusters," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 9, pp. 1446–1454, Dec. 2010.

[166] S. Luo, R. Zhang and T. J. Lim, "Downlink and uplink energy minimization through user association and beamforming in C-RAN," *IEEE Transactions on Wireless Communications*, vol. 14, no. 1, pp. 494–508, Jan. 2015.

[167] W. Liao, M.G. Kibria, G.P. Villardi, O. Zhao, K. Ishizu and F. Kojima, "Coordinated multi-point downlink transmission for dense small cell networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 431–441, Jan. 2019.

[168] M. Yassin, S. Lahoud, K. Khawam, M. Ibrahim, D. Mezher, B. Cousin, "Centralized versus decentralized multi-cell resource and power allocation for multiuser OFDMA networks," *Computer Communications*, vol. 107, pp. 112-124, July 2017.

[169] L. Liu, Young-Han Nam and J. Zhang, "Proportional fair scheduling for multi-cell multi-user MIMO systems," in *Proc. 2010 44th Annual Conference on Information Sciences and Systems*, pp. 1–6, May 2010.

[170] A. Goldsmith, *Wireless communications.* Chapter 3, Cambridge University Press, 2005. ISBN: 9780521837163.

Vita

Yongce Chen was born in Guiyang, Guizhou, China, in 1991. He received his B.S. and M.S. degrees in communication engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 2013 and 2016, respectively. From 2016 to 2021, Yongce was a Ph.D. student in the Bradley Department of Electrical and Computer Engineering at Virginia Tech, Blacksburg, VA, USA.

While a Ph.D. student, Yongce was a Graduate Research Assistant (GRA) in the Complex Networks and Security Research (CNSR) Lab. His current research focused on optimization, MIMO techniques, and real-time implementation of wireless networks. His work has appeared in top-tier international conferences and IEEE journals.

While a master student, Yongce was awarded a National Scholarship from Beijing University of Posts and Telecommunications in 2015. During his Ph.D. study at Virginia Tech, he was awarded a VT Wireless Fellowship in 2016 and a Pratt Fellowship in 2021, respectively. Yongce received the Best Paper Award in IEEE INFOCOM 2021. He was a recipient of Student Travel Grants of IEEE INFOCOM 2018 and IEEE INFOCOM 2021.

Journal Publications

1. **Y. Chen**, Y. Huang, C. Li, Y.T. Hou, W. Lou, "Turbo-HB: A sub-millisecond hybrid beamforming design for 5G mmWave systems," submitted to *IEEE Transactions on Mobile Computing*, under review.

2. **Y. Chen**, S. Li, C. Li, H. Zeng, B. Jalaian, Y.T. Hou and W. Lou, "On DoF conservation in MIMO interference cancellation based on signal strength in the eigenspace," submitted to *IEEE Transactions on Mobile Computing*, under a major revision.

3. **Y. Chen**, Y. Huang, Y. Shi, Y.T. Hou, W. Lou, S. Kompella, "On DoF-based interference cancellation under general channel rank conditions," *IEEE/ACM Transactions on Networking*, vol. 28, issue 3, pp. 1002–1016, June 2020.

4. C. Li, Y. Huang, S. Li, **Y. Chen**, J. Brian, Y.T. Hou, W. Lou, J. Reed, S. Kompella, "Minimizing AoI in a 5G-based IoT network under varying channel conditions," submitted to *IEEE/ACM Transactions on Networking*, under review.

5. Y. Huang, S. Li, **Y. Chen**, Y.T. Hou, W. Lou, J. Delfeld, V. Ditya, "GPU: A new enabling platform for real-time optimization in wireless networks," *IEEE Network*, vol. 34, no. 6, pp. 77–83, November/December 2020.

6. C. Li, S. Li, **Y. Chen**, Y.T. Hou, W. Lou, "Minimizing age of information under general models for IoT data collection," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2256–2270, Oct.–Dec. 2020.

7. Y. Huang, **Y. Chen**, Y.T. Hou, W. Lou, "Achieving fair LTE/WiFi coexistence with real-time scheduling," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, issue 1, pp. 366–380, March 2020.

8. Y. Huang, **Y. Chen**, Y.T. Hou, W. Lou, and J.H. Reed, "Recent advances of LTE/WiFi coexistence in unlicensed spectrum," *IEEE Network*, vol. 32, no. 2, pp. 107–113, March–April, 2018.

Conference Publications

1. **Y. Chen**, Y.T. Hou, W. Lou, "$\mathbf{M}^3$: A sub-millisecond scheduler for multi-cell MIMO networks under C-RAN architecture", submitted to *IEEE INFOCOM 2022*.

2. **Y. Chen**, Y. Wu, Y.T. Hou, W. Lou and S. Kompella, "mCore: Achieving sub-millisecond scheduling for 5G MU-MIMO systems," in *Proc. IEEE INFOCOM*, pp. 1–10, online conference, May 2021.

3. **Y. Chen**, Y. Huang, C. Li, Y.T. Hou, W. Lou, "Turbo-HB: A novel design and implementation to achieve ultra-fast hybrid beamforming," in *Proc. IEEE INFOCOM*, pp. 1489–1498, oinline conference, July 2020.

4. **Y. Chen**, S. Li, C. Li, Y. T. Hou, W. Lou and B. Jalaian, "To cancel or not to cancel: Exploiting interference signal strength in the eigenspace for efficient MIMO DoF utilization," in *Proc. IEEE INFOCOM*, pp. 1954–1962, Paris, France, Apr, 2019,

5. **Y. Chen**, Y. Huang, Y. Shi, Y.T. Hou, W. Lou and S. Kompella, "A general model for DoF-based interference cancellation in MIMO networks with rank-deficient channels," in *Proc. IEEE INFOCOM*, pp. 900–907, Honolulu, HI, USA, April 2018.

6. S. Li, Q. Liu, C. Li, **Y. Chen**, Y. T. Hou, W. Lou, "On scheduling with AoI violation tolerance," in *Proc. IEEE INFOCOM*, pp. 1–9, online conference, May 2021.

7. C. Li, S. Li, **Y. Chen**, Y.T. Hou, W. Lou, "AoI scheduling with maximum thresholds," in *Proc. IEEE INFOCOM*, pp. 436–445, online conference, July 2020.

8. C. Li, Y. Huang, **Y. Chen**, B. Jalaian, Y.T. Hou, and W. Lou, "Kronos: A 5G scheduler for AoI minimization under dynamic channel conditions," in *Proc. IEEE ICDCS*, pp. 1466–1475, Dallas, TX, USA, July 2019.

9. Y. Huang, **Y. Chen**, Y.T. Hou, and W. Lou, "CURT: A real-time scheduling algorithm for coexistence of LTE and Wi-Fi in unlicensed spectrum," in *Proc. IEEE DySPAN*, pp. 1–9, Seoul, Korea, Oct. 2018.