REDUCED-COMPLEXITY RADIO RESOURCE MANAGEMENT ALGORITHMS FOR HETEROGENEOUS MIMO CELLULAR NETWORKS

by

Hakimeh Purmehdi

A thesis submitted in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy
in
Communications

Department of Electrical and Computer Engineering
University of Alberta

Abstract

Multiuser multiple-input multiple-output (MU-MIMO) antenna techniques and heterogeneous network (HetNet) layouts are two promising approaches to dramatically increase throughput in future cellular networks to meet the exploding demand for ever higher data rates. HetNets increase capacity of cellular systems by employing dense layouts of various types of base stations (BSs) in the coverage area. MU-MIMO techniques increase spectral efficiency by enabling spatial multiplexing of data streams transmitted to/from different users. However, effective MIMO spatial multiplexing is achievable only at relatively high signal-to-interference-plus-noise ratios (SINRs). To maximize their capacity, conventional cellular networks are designed to operate at low SINRs. Hence, mitigation of inter-cell interference is required to obtain the benefits of MIMO spatial multiplexing. The most promising approach to achieve it is network coordination.

In this thesis, downlink transmission in a coordinated MU-MIMO HetNet is considered. We investigate the application of simulated annealing (SA) and particle swarm (PS) algorithms to perform user scheduling in a cluster of coordinated network nodes. Our proposed SA and PS algorithms are able to perform in discrete value search space to select users and determine their encoding order for various precoding methods. Moreover, a hybrid algorithm combining the traits of PS and a greedy scheduler is also proposed. We demonstrate that performance of the proposed algorithms, in terms of the achievable sum rate, is close to that of the optimal search at much lower complexity.

To mitigate inter-cluster interference, we develop a rotating clustering scheme, which increases average achievable throughput to cluster-edge users. Considering two different cellular layouts, different rotating patterns of clusters are introduced

and the performance of the network with the proposed clustering patterns is investigated. Our simulations demonstrate the effectiveness of the proposed cluster rotation approach and determine the speed of rotation, beyond which any further performance gains become negligible.

Lastly, we investigate the behaviour of our proposed user scheduling algorithms with temporally correlated channel gains. We further develop our user scheduling algorithms to take advantage of temporal correlation of channel gains to improve their convergence rate and achievable throughput. Our user scheduling algorithms designed specifically for temporally correlated channels perform very close to the optimal exhaustive search at significantly reduced complexity.

Preface

Chapter 3 of the thesis has been published as

- H. Purmehdi, R.C. Elliott, and W.A. Krzymień, "Reduced-complexity user scheduling algorithms for coordinated heterogeneous MIMO networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 8, Aug. 2016, pp. 6184–6203.

- H. Purmehdi, R.C. Elliott, and W.A. Krzymień, "Simulated annealing user scheduling for coordinated heterogeneous MIMO networks," in *Proc. 2012 IEEE Asilomar Conf. on Signals, Systems, and Comput.*, Pacific Grove, CA, USA, Nov. 2012, pp. 1157–1161.

Chapter 4 of the thesis has been published as

- H. Purmehdi, R.C. Elliott, W.A. Krzymień, and J. Melzer "Effect of cluster rotation speed in coordinated heterogeneous MIMO cellular networks with proportionally fair user scheduling," in *Proc. 2014 IEEE Asilomar Conf. on Signals, Systems and Comput.*, Pacific Grove, CA, USA, Nov. 2014, pp. 1816–1820.

- H. Purmehdi, R.C. Elliott, W.A. Krzymień, and J. Melzer "Rotating clustering with simulated annealing user scheduling for coordinated heterogeneous MIMO cellular networks," in *Proc. 2014 IEEE Int. Conf. Commun. (ICC'14)*, Sydney, Australia, Jun. 2014, pp. 5293–5298.

and its journal submission is in the final stages of preparation

- H. Purmehdi, R.C. Elliott, W.A. Krzymień, and J. Melzer "Rotating cluster mechanism for coordinated heterogeneous MIMO cellular networks," *Trans. Emerging Telecommun. Technol. (ETT)*, 14 two-column IEEE-format pages.

Chapter 5 of the thesis to be submitted as

- H. Purmehdi, R.C. Elliott, W.A. Krzymień, and J. Melzer "User scheduling for coordinated heterogeneous MIMO cellular networks with temporally correlated channel gains," in preparation, to be submitted as a correspondence to *IEEE Trans. Veh. Technol.*, 5 two-column IEEE-format pages.

For each publication listed above, I carried out the system model development, the mathematical analysis and computer simulations under the supervision of Prof. Witold Krzymień and cooperation of Dr. Robert C. Elliott. I was responsible for the manuscript composition, Dr. Elliott and Prof. Krzymień contributed to manuscript edits.

To my lovely parents and wonderful brothers.

Acknowledgements

I would like to express my special appreciation and thanks to my supervisor Professor Witold Krzymień, who has been a tremendous mentor for me. I would like to thank you for encouraging my research and for allowing me to grow as a research scientist. I also would like to express my sincere gratitude to Dr. Robert C. Elliott for his excellent cooperation and advice throughout my PhD research work by providing constructive and effective feedback.

I would also like to thank the members of my Ph.D. candidacy examination committee, Professors Chintha Tellambura, Hai Jiang, Yindi Jing and Pedram Mousavi, as well as the members of my final Ph.D. defence committee, Professors Abraham Fapojuwo, Chintha Tellambura, Hai Jiang, Yindi Jing and Pedram Mousavi for their most valuable comments and suggestions. My humble acknowledgement goes to Ms. Pinder Bains, the Graduate Student Advisor for her kind support and advice. All of you have been there to support me throughout my journey to the PhD degree.

I am grateful to Telecommunications Research Laboratories, the Rohit Sharma Professorship, and Natural Sciences and Engineering Research Council (NSERC) of Canada for their generous funding of my PhD research. I would also like to express my special thanks to TELUS Communications and Mitacs-Accelerate Graduate Research Internship Program for enabling for me the rewarding experience of working as an intern with TELUS. My research work has been facilitated by the computing resources of IST (Information Service Technologies) at the University of Alberta, WestGrid and Compute/Calcul Canada. Thanks to all of you.

A special thanks to my family. Words cannot express how grateful I am to my mother and father for all the sacrifices that they have made to enable my

pursuit of knowledge. Your prayer for me was what sustained me thus far. I would also like to thank my brothers and all of my friends and my group mates who supported me throughout my PhD program, and motivated me to strive towards my goal.

Contents

List of Tables

List of Figures

Notation

Symbol	Definition
a or A	A scalar
$\mathbf{a}$	A vector
$\mathbf{A}$	A matrix
$\mathcal{A}$	A calligraphic variable, denotes a set
$\mathbf{I}_M$	The $M \times M$ identity matrix
$\mathbf{A}^T$	The transpose of $\mathbf{A}$
$\mathbf{A}^H$	The conjugate (Hermitian) transpose of $\mathbf{A}$
$Tr(\mathbf{A})$	The trace of a square matrix $\mathbf{A}$
$\mathbf{A}^{-1}$	The inverse of a square matrix $\mathbf{A}$
$\mathbf{A}^{-1/2}$	The inverse square root of a square matrix $\mathbf{A}$
$\|a\|$	The absolute value of the variable
$\|\mathbf{A}\|$	The determinant of the (square) matrix
$\|\mathcal{A}\|$	The cardinality of the set
$\|\mathbf{A}\|_F$	The Frobenius norm of $\mathbf{A}$
$\mathbf{A} \succeq \mathbf{0}$	Denotes $\mathbf{A}$ is positive semi-definite
$diag[\cdot, \cdot, \cdots]$	A block-diagonal matrix formed from the given components
$sgn(\cdot)$	The sign function
$\mathbb{E}(\cdot)$	The expectation operator
$\lceil \cdot \rceil$	The ceiling function
$rand(\cdot)$	Generates a random scalar value according to the specified distribution
$\odot$	The element-wise multiplication of a vector or matrix.

List of Abbreviations

1G	first generation of cellular systems
2G	second generation of cellular systems
3G	third generation of cellular systems
4G	forth generation of cellular systems
5G	fifth generation of cellular systems
3GPP	third generation partnership project
BC	broadcast channel
BD	block diagonalization
BS	base station
CAGR	compound annual growth rate
CDF	cumulative distribution functions
CGP	conjugate gradient projection
CoMP	coordinated multi-point
CSI	channel state information
CSIR	channel state information at receiver
CSIT	channel state information at transmitter
DPC	dirty paper coding
F-norm	Frobenius norm
FDD	frequency division duplexing
GPS	Global Positioning System
GSM	Global System for Mobile communication
HetNets	heterogeneous networks
HGPS	hybrid greedy-particle swarm
HGRPS	hybrid greedy with the reflective border particle swarm
HGMPS	hybrid greedy with the modulo border particle swarm
ICI	inter-cluster interference

IDFT	inverse discrete Fourier transformation
i.i.d.	independent and identically distributed
IMT-Advanced	International Mobile Telecommunications-Advanced
IS-95	Interim Standard-95
ISD	inter-site distance
ITU	International Telecommunication Union
ITU-R	ITU-Radiocommunication Sector
JT	joint transmission
LoS	Line-of-sight
LTE	Long-Term Evolution (of 3G) standard
M2M	machine-to-machine
MAC	multiple-access channel
MIMO	multiple-input multiple-output
MPS$_x$	modulo PS of type x, for $x = 1,2,3$
MT	maximum throughput
MU-MIMO	multi-user MIMO
MUI	multi-user interference
NLoS	non-line-of-sight
NTT	Nippon Telegraph and Telephone corporation
PF	proportionally fair
PS	particle swarm
RPS	reflective PS
RRH	remote radio head
RRM	radio resource management
SA	simulated annealing
SA-ml	memoryless SA algorithm
SA-m	SA with memory algorithm
SISO	single-input single-output

SINR	signal-to-interference-plus-noise-ratio
SNR	signal-to-noise-ratio
SU-MIMO	single-user MIMO
SVD	singular value decomposition
SZF	successive zero-forcing
SZF-DPC	successive zero-forcing dirty paper coding
SZF(CGP)	SZF covariance matrices based on CGP
TC	temporally correlated
UMa	urban Macro
UMi	urban Micro

Chapter 1

Introduction

Basic ideas of cellular radiotelephony go back to 1950s and 1960s by considering the concept of cellular that is by breaking the coverage area into non-overlapping cells (i.e. subareas) to enable frequency reuse to overcome the problem of insufficient availability of the useful radio frequency spectrum [1, 2]. The first commercial cellular telephone system, the NTT system, was introduced in Japan in 1979 [1, 3], as the first one in the first generation (1G) of cellular systems. This system was analog and provided only voice service. Digital cellular system introduced in second generation (2G) with improved capacity and more reliable services [1,3]. The first 2G system was GSM (a European standard first deployed commercially in Finland 1991), which became dominant globally. Integrating multimedia services in third generation (3G) standards and beyond including a combination of text, audio, images, data, and video streams into the cellular system provides adequate services to the users. But multimedia needs high data rates and more bandwidth should be specified to a specific communication link. The forth generation (4G) has been developed to increase the capacity and decrease the network latency by introducing multiple-input multiple-output (MIMO) techniques, heterogeneous networks (HetNets), coordination techniques, etc. Long-term evolution (LTE) of 3G standard [4] (introduced in late 2009) and

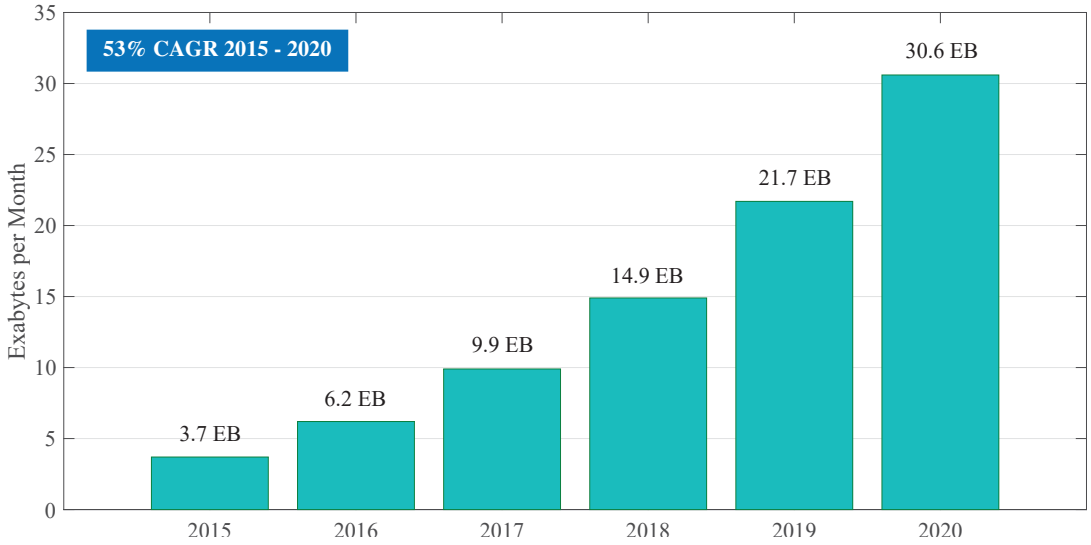

Figure 1.1: Global mobile data traffic [6]

its later enhancement, LTE-Advanced, were introduced and completed by 3GPP by 2011 [5]. LTE-Advanced standard meets or exceeds the requirements specified by ITU in the IMT-Advanced set of requirements for 4G cellular system.

The wireless networks experience ever-increasing demands for data services. In 2015, global mobile data traffic grew an estimated 74% and global mobile devices and connections grew to 7.9 billion. The mobile data traffic is expected to grow at a compound annual growth rate (CAGR) of 53% from 2015 to 2020 [6] (see Fig. 1.1), approaching 30.6 exabytes per month by 2020 (an eightfold increase over 2015). It is also expected that the global mobile devices and connections grow to 11.6 billion by 2020; there will be 8.2 billion personal mobile devices (which mostly will be smart phones) and 3.2 billion machine-to-machine (M2M) connections (e.g., GPS systems in cars, or medical applications making patient records, etc.) [6]. Smart phones will share nearly 50% of total devices and connections, and M2M mobile connections will reach 26% by 2020 [6]. More than proliferation of smart phones, which enables various high data traffic load applications, and M2M connections, the other reasons for this huge increase in

data traffic are such as the increase in traffic load of video applications, internet of things, and so many unforeseen expected applications. Exploding demand for ever-higher data throughputs in cellular networks is one of the main drivers behind their current evolution toward the 5th generation (5G) [7]. All these incremental improvements and possibly the use of new spectrum require advanced radio resource management (RRM). The objective of RRM is to utilize the limited radio resources as efficiently as possible, which is a major challenge in the design of modern cellular networks.

1.1 Heterogeneous MIMO Cellular Networks

Multiple antenna transmitters and receivers introduce spatial degrees of freedom in addition to frequency and time resources in data transmission, and have been an essential part of current cellular network standards such as LTE [4] and LTE-Advanced [5]. MIMO techniques provide diversity (improve link reliability) and spatial multiplexing (increase spectral efficiency) gains by exploiting rich scattering in radio channel [8, 9]. In a single-user MIMO (SU-MIMO) with N transmitting and M receiving antennas, the capacity is scaled linearly by the $\min(N, M)$ compared to the capacity of a single antenna transmission (known as single-input single-output or SISO) [9]. The high spectral efficiency gains achievable with MIMO spatial multiplexing are primarily achievable in the high signal-to-interference-plus-noise-ratio (SINR) regime and rich scattering propagation environment.

Multi-user MIMO (MU-MIMO) systems are able to simultaneously serve multiple users each equipped with a single or multiple antennas (it increases capacity by exploiting multi-user diversity) [10]. In a cellular system with multiple users, using MU-MIMO can significantly improve the spectral efficiency of cellular network. However, some requirements should be considered. First, users cannot

cooperate with each other to receive their signals in downlink. Thus, precoding at the transmitters is required to minimize the multi-user interference (MUI) so that the downlinks can operate in high SINR regime. In MU-MIMO cellular system, there is a large pool of users requesting service, and the system cannot support simultaneous and interference-free transmissions to unlimited number of users. Hence, scheduling of transmissions is necessary, as the second requirement.

By nature of the design of cellular networks, the system performance is inherently interference-limited, meaning that inter-cell interference is a very serious problem in the design of these systems. Multi-antenna base stations (BSs) are particularly adversely affected by this problem, thus this inter-cell interference must be mitigated. As a solution to mitigate the inter-cell interference, coordinated multi-point (CoMP) transmission/reception, also known as network MIMO, has been investigated in the literature, in which several BSs of a cellular system can transmit in a coordinated manner to reduce the inter-cell interference [11–14]. CoMP techniques are classified into coordinated scheduling/beamforming, joint transmission (JP), and dynamic cell selection; they are described in detail in Section 2.4. The joint transmission can achieve the highest throughput performance among all these three [13]. Any cooperating subset of BSs among all BSs of the network is defined as a cluster. To achieve the most possible throughput, antennas of multiple BSs of this cluster are able to transmit jointly as a single antenna array. Each mobile may receive useful signals from several nearby BSs that employ joint precoding and user (or data stream) scheduling [13–16].

Traditional cellular networks are homogeneous networks in which all BSs are identical with similar transmission features like transmitted signal power, approximately similar cell sizes, etc. Traditionally, macro BSs provide network coverage to a geographic area. With ever increasing demands for higher rates of data services, the existing cellular designs are approaching their limits rapidly. Further improvement in spectral efficiency gain is possible with increasing the density of

BSs and imposing various changes on the design of their deployments. Increasing the number of macro BSs, specifically in already dense networks, is very difficult due to expensive costs and lack of available sites [17]. Besides, large number of users demanding very high levels of data traffic is another challenge for operators to face with [17].

To solve these challenges, the utilization of low-power BSs, such as pico and/or femto BSs, is suggested in the literature [17–22], as a component of dense heterogenuous networks (HetNets) [17, 19–21]. HetNets include cells of widely varying sizes with different transmitted powers and possibly overlapping, from macrocells to pico and femto BSs, relays, and/or remote radio heads (RRH), operating within the same frequency band [17–22]. Each different network tier (or type of cell) may also experience different conditions, such as path loss exponents, various geographical distributions, different transmitting powers, etc. As an example, the macro BS transmission power is usually in ranges from 5 W to 40 W, for the low-power low-cost pico BS for outdoor applications it ranges from 250 mW to about 2 W, and indoor femto BSs are transmitting at less than 100 mW [17, 19]. Femto BSs are usually installed by users in their own premises and have no installation/maintenance fees for service providers.

Utilizing different transmitting powers, BSs create challenges like imbalance between uplink and downlink [19]. Also, a mixture of open and closed subscriber access nodes requires to establish new strategies to use time and/or frequency resources shared with macro BSs to avoid of creating lacking coverage; the strategies can assign same or different frequency bands for macros and other low-power elements as well as manage the time-frequency resource block assignment [19]. The deployment of a HetNet, in which all various types of BSs are working in the same frequency band, is known as co-channel deployment [18, 19]. This is the favoured deployment option, because it does not require additional radio frequency spectrum, requires low-cost low-power BSs as well as lower cost user devices to be

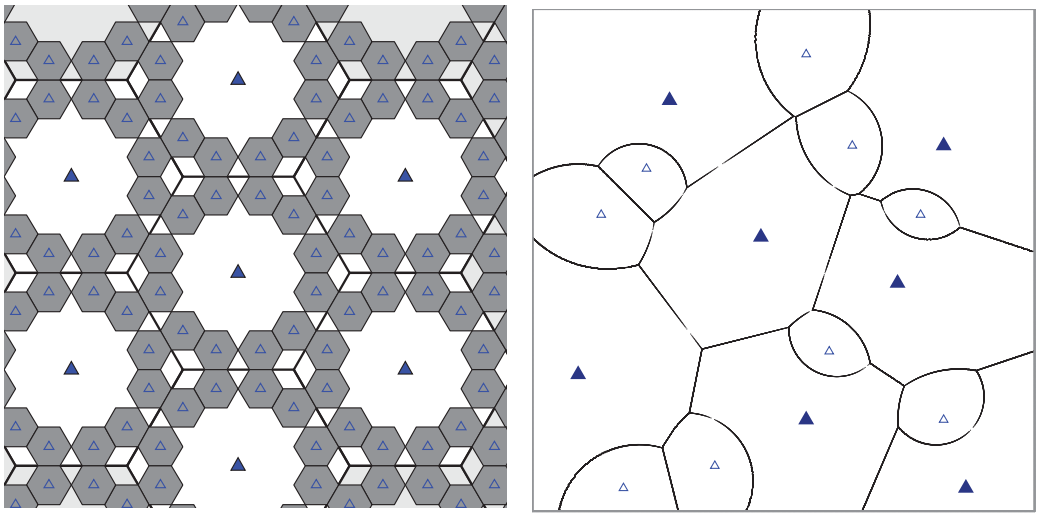

Figure 1.2: Illustration of two different layout deployments for cellular HetNet (a) deterministic hexagonal grid model (b) random spatial model. (cf. [18])

used [19]. Thus, the inter-cell interference coordination techniques are especially important for HetNet, because of their potential ability to reduce interference.

To model the deployment of BSs in a HetNet, two approaches usually used: deterministic (e.g. hexagonal grid model) and spatially random models, as depicted in Fig.1.2. In deterministic approach, the location of BSs and their corresponding distances, the number of each type of BSs in the network and other parameters are known and fixed. The hexagonal cell shape design is a common method to model the coverage area of BSs to tile whole network area. While the network is getting more dense, the cell design is becoming more irregular specially with low power nodes overlaid on the macro-cell area possibly with opportunistic distribution [18]. Thus, stochastic geometry tools have been introduced to model multi-tier HetNets.

MIMO spatial multiplexing is very desirable to increase the area spectral efficiency of HetNets. However, implementation of MIMO techniques in HetNets requires mitigation of interference, which is a serious problem in HetNets. Network coordination is a promising approach to achieve that; however, effective

implementation of it in MIMO HetNets is very challenging. A combination of CoMP and/or MIMO in HetNets, named as coordinated MIMO HetNets, is a research topic of interest [22–25], and these types of networks are inevitable solutions for future networks.

1.2 Radio Resource Management

Radio resources; e.g. bandwidth and power, are very limited and need to be suitably allocated and managed to maximize throughput and quality of service. The strategies and algorithms to control intra-cell interference, such as user sceduling and precoding, the inter-cell interference, such as clustering, in addition to the other radio transmission characteristics like transmit power, data rates, etc. in wireless communication is named as radio resource management (RRM). The objective of RRM is to utilize the limited frequency spectrum as efficiently as possible and it is a very important topic in wireless communication research [16,26–34]. The RRM techniques in terms of reduced-complexity user scheduling algorithms and clustering approaches to increase the throughput of the network MIMO HetNet cellular system are also of interest.

1.2.1 Precoding

In downlink of MU-MIMO system, sum rate capacity is achieved by using dirty paper coding (DPC) precoding [35,36] and transmitting to multiple users and removing interference on each user it encodes, successively. Hence, an encoded user will not experience any interference from other encoded users. This method requires perfect channel state information (CSI) of users and non-causal knowledge of each of the encoded users' signals at the transmitters. Using DPC, transmitter can determine the optimal power covariance matrices for each user to reduce the MUI. Thus, DPC is an optimal precoding method; however, it is extremely

complex and difficult to implement, and lower complexity precoding methods are desired.

Linear precoding methods are significantly less complex, though suboptimal, compared to DPC method. In linear precoding, also known as beamforming, the signal of each user is independently multiplied by a certain beamforming weight vector(s) across the multiple antenna transmitter(s). We discuss the technical aspects of various linear precoding methods in more detail in Chapter 2.

1.2.2 User Scheduling

As we already mentioned, multiple users can be served simultaneously in MU-MIMO system; however, the upper limit to the number of users that the transmitter can transmit simultaneously is relatively low, and it is usually related to the number of transmit antennas; i.e. N (when using DPC). Since in most practical multi-user cellular systems the pool of users requesting service is usually too large, that limit is very likely to be reached. Moreover, using multiple antenna to eliminate MUI decreases the number of simultaneously served users. Hence, the scheduling of users' transmissions at BSs is further necessitates for practical MU-MIMO cellular systems to guarantee that all users can receive the data service in optimized fashion.

An exhaustive search over all possible subsets of users is the optimal user scheduling method, but it is extraordinarily complex and rapidly becomes infeasible as the number of users increases. Furthermore, some precoding methods are sensitive to the order in which users are encoded, and complexity of exhaustive search will further increase, since all the possible encoding orders must also be searched. Thus, reduced complexity user scheduling algorithms, capable of providing performance close to that of an exhaustive search, are required. Design of reduced complexity user scheduling algorithm is a complex precess and requires to consider various factors, which are discussed in Chapter 2, and more specifically

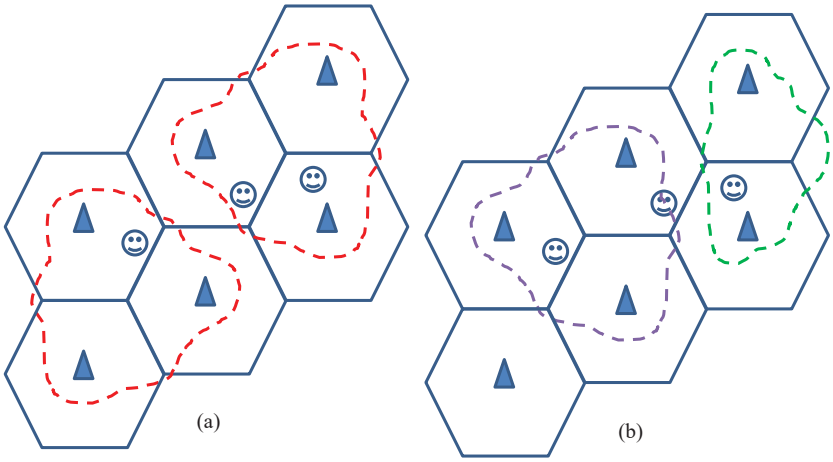

Figure 1.3: Clustering methods (a) fixed cluster, (b) dynamic cluster at sample time t.

in Chapter 3 and Chapter 5.

1.2.3 Clustering Approaches

In practical cellular system, the number of BSs in the system is very large and they are usually distributed in a large wide area (like a city); thus, it is unnecessary and impractical to coordinate transmissions of all BSs of the system altogether. Thus, the clusters of BSs are formed to employ CoMP transmissions. Various clustering patterns can be created using different methods.

In a fixed pattern, the sets of BSs that form clusters are static [11]. This method has comparatively low complexity and overhead, which is presented in Fig.1.3(a). While clustering BSs generally improves the SINR, poor conditions previously experienced by cell-edge users are not completely removed, but instead translated to cluster-edge users, and these users are still suffer from inter-cluster interference (ICI) and achieve low throughputs. Beyond this, in dynamic clustering [37–41] the size and/or shape of the clusters and their constituent BSs change over time (Fig.1.3(b) depicts this method for a sample time t). Of course, this method has high complexity of processing and signalling between the BSs as the

clusters are formed, but it can vastly improve the performance of the system by introducing variability of clustering patterns in time.

1.3 Thesis Objectives and Organization

1.3.1 Motivation and Objectives

Network MIMO HetNets are a promising approach to meet ever-increasing demands and performance expectations of the next generation of cellular networks, as mentioned. Considering the heavier traffic load typically seen in cellular networks on the downlink compared to the uplink, and also the more challenging features of the downlink like the need to have precoding and user scheduling, this work focuses on the downlink of network MIMO HetNets. In this thesis, joint transmission coordination on the downlink of multi-cell MU-MIMO cellular HetNets is considered. However, the combination of joint transmission and MIMO technique in cellular network with various types of BSs becomes even more challenging. Interference is a very important factor which affects the performance of these systems severely. The design and development of practical techniques to mitigate interference is one important aspect of RRM techniques. Some of these challenges are briefly discussed in previous sections.

User scheduling in MU-MIMO systems has been investigated to some extent in literature, but generally developing reduced-complexity and high-performance algorithms, which are able to perform close to exhaustive search with faster search speed, is of high interest. More specifically, developing efficient RRM techniques for coordinated multi-cell multi-user MIMO HetNets attracted interest of the research community recently; however, many limitations and research gaps in design, analysis, and other issues are yet remained and required to be addressed.

A cluster of HetNet BSs coordinates user scheduling, precoding, and transmissions from their antennas as a single virtual array, reducing the intra-cell in-

terference for users within the cluster. However, interference of transmissions in different clusters degrade the performance throughput of these networks, and become very challenging. Existing clustering methods, which are static or dynamic, have mostly been investigated for homogeneous cellular networks, in which all BSs have similar transmission characteristics. The clustering methods for coordinated HetNet, which can reduce ICI with reduced complexity, are also of interest, but their design and analysis are required to be addressed.

This thesis targets these problems, and attempts to exploit different RRM techniques such as introducing proper user scheduling algorithm or clustering technique to achieve closer to the full potential of multi-user multi-cell MIMO HetNets under variety of conditions. A goal of this research is to further investigate scheduling methods. Another goal of this research is targeting the clustering methods in multi-user multi-cell MIMO HetNets. The objective of this thesis is three-fold:

- Proposing reduced complexity user scheduling algorithms for coordinated HetNets designed to operate in the discrete-valued search space of user indices, which are capable of scheduling different encoding orders for the selected users.

- Proposing new cluster formation approach, cluster rotation, and its investigation for different network layouts, analysing its performance, and evaluating the possibility of adapting the cluster rotation to various network layouts.

- Investigation of the proposed user scheduling algorithms for temporally correlated channel; investigation of whether the correlation can be exploited to help the user scheduler make its decisions, improve its performance in terms of achievable sum rates, reduce complexity of the algorithm by decreasing the required number of iterations, etc.

1.3.2 Thesis Outline and Contributions

In this thesis, we focus on different RRM methods to improve the performance of the downlink of a multi-cell multi-user MIMO HetNet in terms of the achievable throughput. The background on MIMO techniques, precoding methods, user scheduling algorithms, and other related topics is covered in Chapter 2. The main contributions of this research are presented in Chapter 3 – Chapter 5. In Chapter 6, we present the concluding remarks and future research directions.

- In Chapter 3, we propose different reduced complexity user scheduling algorithm based on simulated annealing (SA) and particle swarm (PS) for downlink of coordinated heterogeneous MIMO network. We investigate how the algorithms behave using different parameters to control the algorithm, such as convergence parameters or having memory in the algorithm. Similarly, we propose the hybrid greedy-PS (HGPS) user scheduling method, combining the traits of greedy and PS algorithms to benefit from the properties of both methods. These algorithms are evaluated for successive zero-forcing (SZF), successive zero-forcing dirty paper coding (SZF-DPC), and block diagonalizatin (BD) precoding in terms of the maximum achievable sum rate and their performance compared with that of exhaustive search and with each other to investigate the best match between a user scheduling method and a precoding method in the examined system model. All algorithms are designed to operate in the discrete-valued search space of user indices when selecting multiple users for simultaneous transmission (whereas a typical PS or SA algorithm more commonly operates in a continuous-valued space). The algorithms are also capable of evaluating different encoding orders for the selected users, an important aspect when used in systems employing precoding methods which are sensitive to the order in which users are encoded (i.e. successive precoding methods).

- Using coordinated transmissions in a cluster, the users located near the border of each cluster suffer poorer performance compared to those in the middle of it. This problem can be alleviated by using the cluster rotation schemes. Therefore, users have the opportunity to be located in the interior of the cluster at least for a portion of time. In Chapter 4, we propose a rotating clustering scheme for the downlink of a coordinated multi-cell multi-user multiple-input multiple-output system. Two layouts are investigated in this chapter: regular hexagonal cell layout with 6 sectors, and the clover-leaf-shaped cell pattern with 3 sectors. Because of the closer resemblance between the cellular contour and the actual coverage pattern in clover-leaf-shaped layout, it is more often used in modeling practical cellular systems than the simple hexagonal layout pattern, and it is recommended by ITU-R for IMT-Advanced as described in [42, 43]. With both layouts, we attempt to find a suitable set of rotating patterns, which can improve the performance of coordinated MIMO HetNets. As a general result, we try to extend this rotating cluster idea to any possible form of cell patterns and investigate their performance under this method. The performance improvement of this scheme, in terms of the maximum achievable throughput, is analysed with both maximum throughput (MT) and proportionally fair (PF) metrics.

- In many theoretical analyses, the block fading[1] (or quasi-static) channels are modeled as independent and identically distributed (i.i.d.) [44,45]; however, in physical reality they are temporally correlated [46–51]. Finally in Chapter 5, considering temporally correlated (TC) channel gains, we are interested to extend our analysis of proposed user scheduling algorithms to

[1]For simplicity in analysis, the fading channels are usually assumed as quasi static in each transmission interval and named as block fading channels. In block fading channel model, it is assumed that the channel is constant in each transmission interval, and the channels vary independently within different blocks of transmission.

account for the effect of the temporal correlations on their performance. The performance of the proposed user scheduling algorithms are analiyzed with both MT and PF metrics, in this chapter. Investigating the effect of TC channels on the output of the user scheduler, we are interested to see how this correlation can help the scheduler to improve its achievable sum rates and/or to reduce its complexity. In TC channels, the channel samples of two consecutive scheduling intervals are relatively similar to each other, depending on the severity of the correlation of the channel. In this chapter, we proposed techniques and some updates into our proposed user scheduling algorithms to get the most advantage of TC channels in their search process to achieve throughput even closer to exhaustive search with more reduced-complexity. The proposed upgrades attempt to exploit the TC characteristics of the channel gains into the user scheduler's search process to improve its performance.

Chapter 2

Background

2.1 Wireless Propagation Channel Modeling

Radio signals in wireless channels between transmitter and receiver propagate through various mechanisms including reflection, diffraction and scattering. As a result, the wireless channels can be modeled based on various phenomena such as path loss, shadowing, and multipath fading [3].

Path Loss: The attenuation in the power of the signal is path loss, which is usually a function of the distance between transmitter and receiver. Path loss is commonly modeled as being proportional to $d^{-\alpha}$, where d is the physical distance between user and BS, and α is the path loss exponent in the range of 1.6 to 6 [1, 3]. Empirical path loss models also include additional attenuation effects depending on the propagation environment, e.g. the carrier frequency, the heights of buildings and antennas, the spacing between buildings, building penetration losses, etc. [3].

Shadowing: There are usually large objects like buildings and hills in the cellular mobile propagation environment, which cause deviations in the average channel power gain that one would anticipate from the path loss alone. These deviations are named as shadowing or shadow fading, which is a random variable

(e.g. ρ) most typically modeled by a log-normal distribution [3]. Depending on the model, the shadowing between nearby locations in the environment may simply be considered independent, or it may have spatial correlation.

Multipath Fading: The transmitted signal propagates along multiple paths caused by reflection, diffraction and scattering, and various copies of this signal arrive at the receiver from many different directions with various amplitudes and phases. The constructive or destructive combination of these copies causes rapid fluctuations in the amplitude and the phase of received signal, which is known as multipath fading or small-scale fading. If all frequency components of the received signal experience the same amplitude and phase changes, the fading is called flat fading; otherwise, it is known as frequency selective fading (various frequency components of the signal encounter different changes in amplitude and phase). The multipath flat fading is usually represented by a complex random variable (e.g. z). In a rich scattering environment with no significant line-of-sight (LoS) components, the fading amplitude is modeled by a Rayleigh distribution [3] and thus known as Rayleigh fading. Other fading models include Rician fading (which includes a LoS component), and Nakagami fading (which can model varying levels of severity of the small-scale fading) [3].

Thus, the radio link between BS and user in a multipath flat fading channel considering path loss and shadowing can be expressed as

$$h \propto z\sqrt{d^{-\alpha}\rho} \tag{2.1}$$

In each transmission interval, the complex downlink channel signal strength coeffcient h is a random variable, wherein the small-scale fading can be considered as i.i.d. between transmission intervals. However, in reality, the channel may exhibit temporal correlation. The characteristics of the wireless radio channels depend on various factors such as operating frequency, propagation environment,

mobility of the users, etc. In a typical cellular system, the macro BSs are usually installed on high towers to cover large cells (the radius of a macrocell is typically about 500 m to 1 km) with their antennas usually above the local scatterers. However, users are typically surrounded by local scatterers. Since the distance between user and macro BS is generally large, the radio propagation is usually assumed as a two-dimensional plane wave [3]. Due to the mobility of the users or surrounding scatterers, the channel gains may be correlated with each other.

2.2 MIMO Techniques and Challenges

A point-to-point link between one transmitter and one receiver, each with multiple antennas, is modeled by an SU-MIMO channel. The quasi-static MIMO channel is modeled by the channel matrix $\mathbf{H} \in \mathcal{C}^{M \times N}$, where N and M are, respectively, the number of transmitting and receiving antennas. Each element of $\mathbf{H}$ is given by $h_{m,n}$, where it characterizes the channel between nth transmitting antenna and mth receiving antenna. Transmissions through this channel experience multi-path fading caused by movements of transmitter, receiver, or their surrounding scatterers (known as small-scale fading). Considering the central limit theorem, the combination of multi-path signals can be modeled with a complex Gaussian random variable with rich scattering environment. If the antennas of the transmitter (similarly, for antennas of the receiver) are spaced sufficiently far apart (at least half of the wavelength), the elements of $\mathbf{H}$ can be modeled as uncorrelated (for complex Gaussian random variables, it is also considered as independent variables). Thus, the elements of $\mathbf{H}$ are modeled by i.i.d. complex Gaussian random variables with zero mean and unit variance. Assuming a total transmitting power constraint P, a lower bound of SU-MIMO capacity is achieved by sending independent data streams with the same power from each of the transmit antennas, sometimes known as isotropic input [52]. When CSI is

not available at the transmitter (but the receiver has perfect CSI), the isotropic input is the good choice, and the achievable rate of this transmission is given as [52]

$$\log_2 \left| \mathbf{I}_M + \frac{P}{N\sigma^2} \mathbf{H}\mathbf{H}^H \right| \tag{2.2}$$

where σ^2 is the variance of the additive circularly symmetric complex Gaussian noise.

According to information theory, the maximum spectral efficiency can be obtained by having perfect CSI at both the transmitter and receiver [8, 9], i.e., CSIT and CSIR, respectively. The data streams are spatially multiplexed and sent on r independent subchannels, where r is the rank of the MIMO channel[1] $\mathbf{H}$. The maximum SU-MIMO capacity is achieved by maximizing the sum rate of r streams subject to the sum power allocation constraint and is given as [52]

$$\max_{P_i : \sum_i P_i = P} \sum_{i=1}^{r} \log_2 \left(1 + \frac{P_i \lambda_i^2}{\sigma^2} \right) \tag{2.3}$$

where λ_i, $i = 1, \cdots, r$ are singular values of $\mathbf{H}$ and P_i is the power allocated to the ith eigenmode of the channel. The optimum values of P_i $(i = 1, 2, \cdots, r)$ are obtained through waterfiling power allocation over the singular values of the channel matrix, i.e., λ_i, when CSIT is available. At low signal-to-noise ratio (SNR), all the transmit power is allocated to the strongest eigenmode ($P_j^{opt} = P$, where $j = \arg\max_i \lambda_i$) [52]. By increasing the SNR, other eigenmodes will be some portion of the transmit power and thus become activated. When the SNR is sufficiently high, all r available data streams will be allocated some non-zero power P_i subject to the sum-power constraint that appears in (2.3), where P_i is constraint to the $\sum_i P_i = P$. When the system is at a very high level of SNR, the power is allocated more or less equally among all r streams. Thus, MIMO spatial multiplexing is most effective at sufficiently high SNR (or sufficiently high levels

[1] Assuming $\mathbf{H}$ is full rank, the rank of the MIMO channel is $\min(M, N)$.

of SINR, when the interference from other transmissions is also considered).

At low SNR and using isotropic input, the average capacity scales linearly with M; thus, multiple transmitting antennas do not improve the capacity in this case [52]. When CSI is available at both the transmitter and receiver, the average capacity at low SNR scales with $\mathbb{E}(\lambda^2_{max}(\mathbf{H}))$ [52]. In i.i.d. Rayleigh channels, $\mathbb{E}(\lambda^2_{max}(\mathbf{H}))$ is larger than or equal to $\max(N, M)$. This analysis clearly reveals the importance of perfect CSIT to achieve high promising gains of MIMO. An important metric characterizing MIMO spatial multiplexing is the multiplexing gain, which is defined as the asymptotic MIMO capacity, when the SNR approaches infinity [52]

$$\lim_{P/\sigma^2 \longrightarrow \infty} \frac{C_{MIMO}(N, M, P/\sigma^2)}{\log_2 P/\sigma^2} \tag{2.4}$$

This formula describes the number of simultaneous data streams (or interference-free subchannels in which the streams for single user do not interfere with each other) that can be employed to improve the capacity, by using MIMO spatial multiplexing with no additional power or bandwidth. At high SNR, the multiplexing gain is given by $\min(N, M)$ regardless of CSIT availability (despite the advantage of CSIT, the multiplexing gain at high SNR is the same with or without it) [52]. This result reveals that employing multiple antennas at both the transmitter and receiver is necessary to achieve multiplexing gain at high SNR. It is necessary to mention that the improvement in SU-MIMO capacity due to the availability of CSIT is usually small, whereas it becomes much more significant in MU-MIMO [52].

The spatial multiplexing gain of SU-MIMO is limited to data streams of a single user. In high-rank SU-MIMO transmission, this gain is limited by the number of antennas on the user's device [53]. In MU-MIMO systems, data streams from or to multiple users are multiplexed and transmitted simultaneously, and the achiev-

able rates of all users are limited by the K-dimensional capacity region, where K is the number of users. Thus, higher throughputs are achievable with MU-MIMO compared to SU-MIMO. The uplink and downlink channels of MU-MIMO are modeled as the Gaussian MIMO multiple-access channel (MAC) and broadcast channel (BC), respectively. Since in this work our focus is on the downlink, we describe the analysis of the downlink in this part. The K-dimensional BC rate region is given as [52]

$$\mathbf{R}_{\pi} = \left\{ (R_{\pi_1}, \cdots, R_{\pi_K}) : R_{\pi_k} = \log_2 \frac{|\sigma^2 \mathbf{I}_M + \mathbf{H}_{\pi_k} \sum_{i=k}^{K} \mathbf{Q}_{\pi_i} \mathbf{H}_{\pi_k}^H|}{|\sigma^2 \mathbf{I}_M + \mathbf{H}_{\pi_k} \sum_{i=k+1}^{K} \mathbf{Q}_{\pi_i} \mathbf{H}_{\pi_k}^H|} \right\} \tag{2.5}$$

where π_k is the user encoded kth in the ordered permutation π of the K users, $\mathbf{H}_{\pi_k}$ is the channel matrix between user π_k and the transmitter, $\mathbf{Q}_{\pi_k}$ is that user's corresponding transmit covariance matrix and is defined as $\mathbf{Q}_{\pi_k} := \mathbb{E}(\mathbf{s}_{\pi_k} \mathbf{s}_{\pi_k}^H)$, $\mathbf{s}_{\pi_k}$ is the transmitted signal vector for the kth encoded user, and the system is subject to the sum-power constraint $\sum_{j=1}^{K} Tr(\mathbf{Q}_j) \leq P$. It is noteworthy that this rate region is achievable using DPC precoding [35, 36, 54]. The sum-rate capacity is the maximum possible total achievable throughput obtained by all K users and it is given as

$$C_{MIMO}(N, (K, M), P/\sigma^2) = \max_{\pi \in \mathcal{S}} \max_{\mathbf{Q}_{\pi_k} : \mathbf{Q}_{\pi_k} \succeq \mathbf{0}, \sum_{j=1}^{K} Tr(\mathbf{Q}_j) \leq P} \sum_{k=1}^{K} R_{\pi_k} \tag{2.6}$$

where $\mathcal{S}$ is the set of all possible ordered subsets among K users. The rate R_{π_k} in (2.6) is obtained by using (2.5). At low SNR, the average sum rate is expressed as $\mathcal{O}(\mathbb{E}_{\mathbf{H}_k} \max_k [\lambda_{max}^2(\mathbf{H}_k)])$, which means that on the BC, the MIMO system transmits only to the user with the largest λ among all K users [52]. This implies that the more users, the more improvement in average sum rate due to multi-user diversity. At high SNR, the multiplexing gain achieves asymptotically the value of $\min(N, KM)$. If CSIT is not available, then the BS cannot multiplex

the multiple users' signals effectively.

In an MU-MIMO network, the MUI is handled in part by the multiple antennas, which in addition to providing per-link spatial diversity, also give the degrees of freedom necessary for spatial separation of the users. According to information theory, the optimum transmit strategy for the downlink of MU-MIMO includes MUI cancellation combined with optimized user scheduling and power allocation [10]. MU-MIMO systems face two important challenges. First, due to the lack of cooperation among users on the downlink, the transmitted signals need to be precoded at the transmitter(s) to mitigate multi-user interference and enable spatial multiplexing gains of MIMO. By exploiting CSIT, precoding matrices are generated, which mitigate the MUI within the cell. Second, with a very large number of users requesting data services, user scheduling is a necessity for any multi-user MIMO system to guarantee that all users can receive data in an optimized fashion with less MUI. After satisfying these requirements and assuming K served users in the system, the spatial multiplexing gain of $\min(N, KM)$ becomes achievable at high SINR.

All the above analyses indicate that the downlink of MU-MIMO can achieve its promising gains, if it has perfect CSIT and high levels of SINR. However, in practical systems, these requirements are not available very readily.

2.3 Precoding on Downlink of MU-MIMO

As mentioned already, the transmissions on the downlink of an MU-MIMO system should be precoded to help to reduce the interference between users. The multi-user MIMO BC capacity can be achieved by using DPC [35, 36, 54]. Stemming from Costa's research [54], if the transmitter knows about the interference caused by previously encoded users, channel capacity would be the same as the case that there were no interference from those users. This is an optimal and non-

linear encoding technique to achieve the MU-MIMO broadcast channel capacity (it requires non-causal knowledge of each user's signal at the transmitter). However, DPC is a very complex technique which is not practically implementable. Hence, suboptimal precoding methods with reduced complexity are of interest to alleviate MUI. Precoding is divided into linear and non-linear techniques. In applications with multiple-antenna users, several suboptimal precoding techniques have been introduced in the literature such as BD [55,56], SZF and SZF-DPC [30]. Suboptimal precoding techniques for single-antenna user applications have also been investigated like zero-forcing or minimum mean-square error [10, 30].

In an MU-MIMO system with K users, the received signal $\mathbf{y}_k$ for user k is given by

$$\mathbf{y}_k = \mathbf{H}_k \sum_{j=1}^{K} \mathbf{W}_j \mathbf{s}_j + \mathbf{n}_k. \tag{2.7}$$

where $\mathbf{W}_k \in \mathcal{C}^{N \times M}$ is the precoding matrix and $\mathbf{n}_k$ is the complex additive white Gaussian noise vector with zero mean and $\sigma^2 \mathbf{I}_M$ covariance matrix. The summation in the first term in (2.7), i.e. $\mathbf{x} = \sum_{j=1}^{K} \mathbf{W}_j \mathbf{s}_j$, is the transmitted signal. The precoding matrices are designed to remove some or all interference between users. In zero-forcing precoding [36], the transmitted signal is $\mathbf{x} = \mathbf{H}^{\dagger}\mathbf{s}$, where $\mathbf{s}$ is the data vector of all K users, and the aggregate precoding matrix is defined by $\mathbf{H}^{\dagger}$, the Moore-Penrose pseudoinverse of $\mathbf{H}$, given by $\mathbf{H}^{\dagger} = \mathbf{H}^H \left(\mathbf{H}\mathbf{H}^H\right)^{-1}$, where $\mathbf{H} = \left[\mathbf{H}_1^T, \mathbf{H}_2^T, \cdots, \mathbf{H}_K^T\right]^T$ is the aggregate channel matrix of all users. The precoding matrix $\mathbf{W}_k$ is the kth column of $\mathbf{H}^{\dagger}$. The aggregate precoding matrix of the minimum mean-square error precoding method is defined by $\mathbf{H}^H \left(\mathbf{H}\mathbf{H}^H + \frac{K}{SNR}\mathbf{I}_M\right)^{-1}$ [57]. These two methods are sensitive to the channel estimation errors, and with increasing SNR, the capacity gap between these methods and DPC increases [58].

Under BD precoding, MUI is completely canceled at the transmitter side. The precoding matrix of each user is designed to fall in the nullspace of (i.e.,

be orthogonal to) the channels of all other served users [55, 56]; i.e., $\mathbf{H}_k\mathbf{W}_j = \mathbf{0}$ for all $k \neq j$, $1 \leq (k, j) \leq K$. This nullspace constraint results in a limitation on the number of users that can be served simultaneously. Since the MUI is fully cancelled for each user with BD, the order that the users are encoded in is irrelevant; every possible ordering has the identical sum rate. In contrast, for SZF the MUI is canceled only partially by the precoding; the signals of previously encoded users interfere with the signals for users later in the encoding order (i.e., $\mathbf{H}_k\mathbf{W}_j = \mathbf{0}$ only for $j > k$). Because of this, the order that the users are encoded in will affect the rates they can achieve, as well as the sum rate. The encoding ordering of the scheduled users is thus very important for sum rate maximization, resulting in more complexity for the scheduling and throughput maximization problems (the latter being non-convex) [30]. Compared to BD, SZF can achieve higher throughputs, and in some cases is also capable of serving more users simultaneously [30].

Both BD and SZF, being linear, are suboptimal precoding methods, so their performance in terms of the sum rate capacity is inferior to that of DPC. SZF-DPC is also proposed in [30], which fully eliminates MUI by using both nulling in one direction of the encoding order and DPC in the opposite direction. In SZF-DPC it is assumed that for each user there is non-causal knowledge of the interfering signals intended for users encoded earlier [30, 59]. Therefore, as per the principle of DPC, the channel capacity in the presence of this interference is the same as if said interference were not present [59]. The interference from the remaining users (those encoded later) is nulled. Although SZF-DPC's performance is much better than BD and SZF, it still is inferior to DPC in terms of its achievable capacity. Its chief advantage is in its lower complexity (compared to that for DPC) of allocating power to the scheduled users to maximize the sum rate.

2.4 Coordinated Multi-Point Transmission Techniques

A very promising approach to increase SINR on the downlink of cellular networks without reducing the frequency reuse factor[2] or traffic load is network MIMO. As mentioned in Section 1.1, coordinated transmission can be categorized into three different approaches. In joint transmission, antennas of multiple transmitters of multiple base stations act together as a single antenna array, and each mobile may receive useful signals from several nearby BSs (known as cluster) [11–14]. The signals are jointly precoded in this cluster. This approach requires availability at all involved BSs (transmitting nodes) of data and CSI for all simultaneously scheduled users [13] (see Fig. 2.1(a)).

Another category of coordinated transmission is dynamic cell selection (also known as transmission point selection) in which a single BS, which is selected based on the maximum averaged SINR[3] dynamically among all coordinating BSs, transmits a signal towards the user while the other coordinating BSs are mute [13]; this approach is depicted in Fig. 2.1(b). The performance of this approach is somewhat inferior to joint transmission.

Another approach to mitigating inter-cell interference, known as coordinated scheduling/beamforming (see Fig. 2.1(c)), is simpler in that it does not require availability of all users' data at all BSs involved in coordination. By using CSI, the coordinating BSs generate the beamforming vectors for their corresponding users in order to reduce the interference within the coordinated cells. The performance of this method is inferior to joint transmission and dynamic cell selection [13].

[2]This factor is a measure of how often the same frequency can be used in different areas of the network. Using F as the number of cells which cannot use the same frequencies for transmission, the frequency reuse factor can be defined as $1/F$.

[3]The SINR is usually averaged over a slot (radio resource block) or subframe.

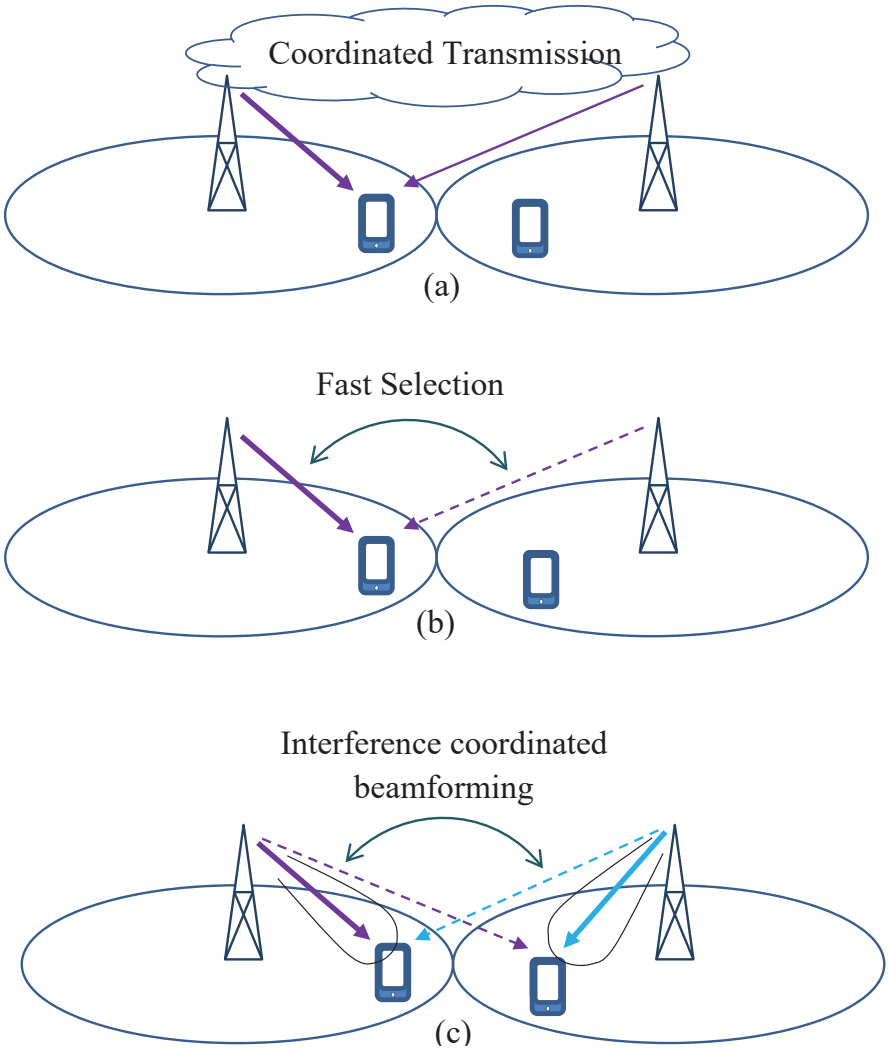

Figure 2.1: CoMP transmission in downlink (a) joint transmission, (b) dynamic cell selection, (c) coordinated beamforming (cf. [13]).

2.5 User Scheduling Algorithms

For the downlink of an MU-MIMO system, the upper limit on the number of users that can be served simultaneously is equal to the number of transmitting antennas N. This constraint is a soft limit when DPC is employed [35, 36, 54, 60] and the system can transmit to more than N users; however, it is not required. In contrast, with linear precoding methods such as SZF or BD, this constraint is a hard limit on the number of scheduled users and the system cannot support

more than N users simultaneously [55, 61]. Therefore, on the downlink of an MU-MIMO system, and more specifically on the downlink of a coordinated MU-MIMO cellular HetNet with a very large number of users requesting service, user scheduling is necessary. A user scheduler is primarily responsible to decide which users should be scheduled for the next transmission. But, user scheduling algorithms are not necessarily limited to this purpose. They can also manage the order in which users are encoded, the amount of data (or equivalently the data rate) that users are assigned, the power that is allocated for each scheduled user, etc.

The optimal user scheduling method, an exhaustive search over all possible subsets of users with all possible encoding orders, is extraordinarily complex and rapidly becomes infeasible as the number of users increases. To determine the optimal vector of scheduled users, exhaustive search must search over $\sum_{j=1}^{N} j!\binom{K}{j}$ ordered combinations. Thus, reduced-complexity user scheduling algorithms that are capable of achieving throughputs (sum rates) as close as possible to optimal are of interest.

Various heuristic scheduling methods try to find a reduced-complexity near-optimal solution. In a greedy algorithm, as an example, when each user is selected, the incremental gain of the scheduling metric is maximized. The metrics considered by some greedy algorithms (e.g. [26, 27, 56]) include the channel capacity or sum rate, channel gains or norms, and/or the amount of interference caused to the other scheduled users. There are other scheduling metrics, which try to maximize the system throughput under different constraints such as delay [62] or the minimum achievable throughput per user [63]. Metaheuristic stochastic optimization is another category of techniques, which includes genetic algorithms, simulated annealing, and particle swarm optimization methods. The first of the three, the genetic algorithm, is inspired by biological evolution, with the solution evolving over time towards the optimal one. In [31, 32], schedulers based on

genetic algorithms are proposed.

Simulated annealing is another stochastic optimization technique to find a solution among a wide range of candidates, and is inspired by the metallurgical process of annealing [64]. This process involves heating and controlled cooling of a material to improve its crystal structure. The SA algorithm searches randomly among neighbouring candidate solutions. It is capable of escaping from local optima in favour of searching more solutions for the global optimum (or at least a better suboptimal solution) by using an acceptance criterion that defines the probability of accepting a worse solution than the current candidate (better solutions are always accepted). During the algorithm iterations, this probability is decreased gradually by a temperature parameter that is analogous to the cooling in annealing. In [65], a related SA algorithm for data stream scheduling in a single-cell MIMO system is described. Earlier works [29, 66] on SA-based user scheduling (also jointly optimizing adaptive modulation and coding for the users) only considered a single cell with a single user being scheduled per resource block, with no capability of spatial multiplexing for simultaneous transmission to more than one user per resource block; the algorithms also could not account for a user precoding order.

In 1995, Kennedy and Eberhart proposed the particle swarm optimization technique by simulating the collective intelligence and social behavior that exists among some species of animals and combining it with evolutionary computational methods [67–69]. When solving a problem with PS, initial swarms (made up of candidate solutions referred to as particles) explore the search space of the problem iteratively to find solutions, by considering a velocity and location in the search space for each particle. The particles' movement is influenced by both the best solution found by each individual particle, and the best overall solution of the swarm [67–69]. In [70, 71], PS-based user schedulers are proposed. Their solution structure assigns a continuous-valued "fitness" variable (between 0 and

1) to each user to indicate how good of a choice that user is to be scheduled. However, that value of fitness for each user seems largely unrelated to the value of the utility function being optimized, and there appears to be nothing preventing the fitness value from departing from the valid range of 0 to 1. Moreover, the algorithms do not directly adapt the number of scheduled users (instead always scheduling the maximum number possible), nor again are they capable of considering an encoding order in their described design[4]. (Zero-forcing precoding is applied in [70, 71] and thus the users' order of encoding is not important therein. However, consequently the proposed PS algorithms in [70, 71] cannot be applied to successive precoding methods.)

2.6 Scheduling Criteria

Generally, in any system throughput analysis, and specifically in MU-MIMO system analysis, two scheduling metrics are most commonly considered: the maximum throughput (MT) and the proportionally fair (PF) metrics. For the MT metric, the scheduler attempts to select and allocate radio resources to those users with the highest supportable data rates, in which their achievable sum rate is the maximum among all other permutations of users [72]. In each scheduling interval, the MT metric maximizes the sum rate, which is given as $\sum_{j=1}^{K} R_k$, where R_k is the kth user's achievable rate; this metric achieves the highest possible capacity for the system.

In any wireless network, some users experience very poor channel conditions. Thus, their achieved rates are small and consequently they might be selected by an MT scheduler less often (or potentially never be selected) unless their channel conditions improve, the channel conditions of the served users deteriorates, etc. MT scheduling focuses solely on aggregate throughput and does not take into

[4]In general, user scheduling algorithms often do not include this ordering aspect, whether they are based on metaheuristic stochastic optimization techniques or otherwise.

account the fairness among users. Thus, service starvation is a possible scenario. Hence, some compromise between fairness and throughput is required.

The PF metric, in contrast, is based upon maintaining a balance between the maximum system throughput and user fairness [73–75]. A PF scheduler maximizes the utility function given as $\sum_{j=1}^{K} \log_2 \bar{R}_k$, where $\bar{R}_k$ is the average long-term throughput of the kth user [73]. This is accomplished by assigning each data flow (i.e. each user in our context) a scheduling priority that is inversely proportional to its anticipated average resource consumption. In the PF metric, each user has a weight related to its priority for being chosen by the scheduler, and the scheduler adjusts each weight based on the average achieved rates over the user's history. A PF scheduler chooses those users whose instantaneous rates relative to their average rates are better than the others, and uses a weighted sum rate as its utility function given as [73, 74]

$$\sum_{k=1}^{K} \mu_k R_k \tag{2.8}$$

where μ_k is the priority weight of the kth user and is defined as $\mu_k = 1/\bar{R}_k$, and $\bar{R}_k$ is the average throughput approximated by a moving average window over the past t_c scheduling intervals. The value of $\bar{R}_k$ is updated for each upcoming scheduling interval, i.e. the $(t+1)$th scheduling interval, when the kth user's rate at scheduling interval t has been decided, and it is calculated by [73–75]

$$\bar{R}_k(t+1) = (1 - \frac{1}{t_c})\bar{R}_k(t) + \frac{1}{t_c} R_k(t) \tag{2.9}$$

If user k is not scheduled at time t, then the rate $R_k(t)$ is 0 for that user in (2.9). Using the PF metric, if a user has been selected by the scheduler often, its weight for the next interval will be decreased (as its average rate increases), i.e. its chance to be chosen in the next scheduling interval diminishes. Meanwhile, another user

with worse channel conditions may have more opportunity to be scheduled in the next interval simply by having higher weight. This method provides more fairness in the network among all users. It is noteworthy to mention that MT can be considered as a special case of weighted sum rate maximization, which is defined by setting μ_k to a constant value of 1, for all users.

Chapter 3

Reduced-Complexity User Scheduling Algorithms for MIMO HetNets

3.1 Introduction

With the constraint on the maximum number of scheduled users in a MU-MIMO system, researchers and designers of coordinated multiuser multi-cell MIMO HetNets with very large number of users requesting data are enthusiastically seeking reduced-complexity high performance user scheduling algorithms. As mentioned earlier, the optimal user scheduling algorithm exhaustively searches all possible choices, which is very complex especially for a large pool of users. Various heuristic scheduling methods have been introduced in the literature that try to find a reduced-complexity algorithm capable of providing performance close to that of an exhaustive search [26, 27, 31, 32, 56, 65, 76]. One example of such methods is the category of metaheuristic stochastic optimization techniques. These methods involve random elements to guide their movement through the search space of the problem, and are able to find a good solution to the optimization problem. In

practical systems, we also require finding a good solution as often as each transmission interval; thus, the generally high speed of these techniques in arriving at a solution makes them attractive for use in the context of scheduling. Simulated annealing (SA) and particle swarm (PS) are two examples of reduced-complexity metaheuristic optimization approaches for user scheduling. However, implementing these methods for the multi-user multi-cell MIMO HetNet is difficult and requires research.

3.1.1 Motivation and Contributions

In this chapter, reduced-complexity user scheduling algorithms based on PS and SA are proposed and their performance for a multi-user network MIMO HetNet is evaluated. A typical PS or SA algorithm more commonly operates in a continuous-valued space; however, our proposed algorithms based on PS or SA are designed to operate in the discrete-valued search space of user indices when selecting multiple users for simultaneous transmission. Moreover, both proposed algorithms are capable of evaluating different encoding orders for the selected users, a vital aspect when used in systems employing successive precoding methods (typical SA and/or PS algorithms are not able to recognize the encoding order of users in the scheduling vector).

The initialization and selection/tuning of parameter values are important factors in the operation of the algorithms. We have devoted a significant portion of the discussion of the simulation results to this aspect. It is noteworthy that the precoding method used in the system has an impact on the selection of the parameters in the scheduling algorithms. Thus, we also investigate the performance of our proposed algorithms in combination with various precoding methods.

For the PS algorithm, we propose various border methods for the search space used in conjunction with the particle position updates to ensure the solution elements remain integers in the range of valid user indices. Furthermore, the

proposed SA algorithm is equipped with a neighbourhood function in the form of a Markov chain to search among possible candidate solutions by switching users and/or their encoding order, and increasing/decreasing the number of users scheduled for transmission. Additionally, we propose a hybrid greedy-PS user scheduling method, combining the traits of greedy and PS algorithms. In this algorithm, the user with the strongest channel is always considered in all particles (inspired by how some greedy algorithms would choose that user first). To achieve its highest possible throughput, that user is also encoded as the first user in each particle; thus, there is no interference from previously encoded users to decrease the achievable rate of this user (this feature is mostly important with successive precoding methods). Users for the rest of the particle are chosen as per the process done in the base PS algorithm. The analyses of the algorithms consider a throughput maximization metric for each of the precoding methods.

3.2 System Model and Sum Rate Maximization

We consider the downlink of a coordinated multi-user multi-cell MIMO HetNet cellular network. In the assumed HetNet model, each macrocell has a sectorized high-power BS surrounded by 12 low-power pico BSs, which form picocells overlaying the macro coverage area as shown in Fig. 3.1. Without loss of generality, we consider the target area of the network as depicted by the red triangle[1] in Fig. 3.1. This area includes a set of 3 macro BS sectors and 6 omnidirectional pico BSs each equipped with N transmitting antennas. Any subset of these 9 transmitting nodes can form a cluster. There are K users uniformly distributed over target area, each with M receiving antennas. K_c of these users are assigned to cluster c, from which U_c users are served. Each cluster transmits coordinated signals from all its nodes to its scheduled users.

[1]The examination herein applies equally to any arbitrary triangular-shaped area of cooperating macro sectors and picocells as illustrated in Fig. 3.1.

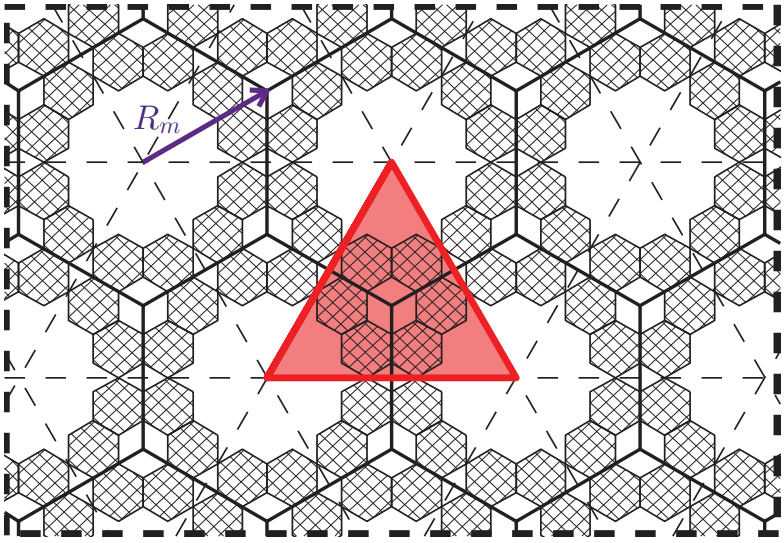

Figure 3.1: Schematic deployment of HetNet. Larger hexagons represent macrocells, while smaller shaded hexagons denote picocells.

The strategy of cluster forming is a user-centric strategy according to the average received power from each node, without coordination with any other nodes. Prior to any other sort of signal processing occurring, the users measure the average received power from each BS (e.g. from a carrier wave). Each user then reports the received power from all 9 BSs within the target area back to the BSs. There is assumed to be a central unit processor that analyzes this data (according to LTE network architecture [77], this central processor can be located in either the radio access network portion, to manage a subset of BSs locally, or in the core network portion, to manage all BSs through the network, as presented in Fig. 3.2). The processor chooses the BSs that should be members of the cluster for a given user as those from which their received power at the user's location is within 20 dB of the strongest received by that user. Based on this comparison, there will be a subset of BSs that can serve each user to satisfy a minimum acceptable SINR. If two or more users are to be served simultaneously and they share at least one BS in common in their serving subsets, one larger serving cluster for those users is formed as the union of these subsets. That larger

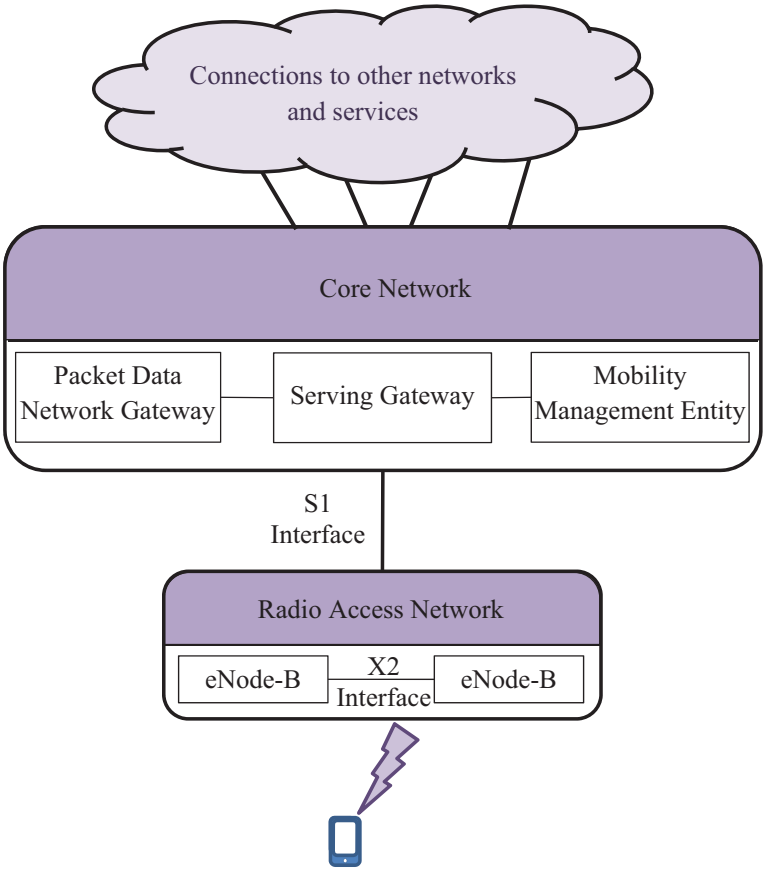

Figure 3.2: Schematic of LTE network architecture (cf. [77]). The central processor controlling the coordination between eNode-Bs (i.e. BSs) may be in either the core network or the radio access network.

cluster is of potential use for all of the users whose subsets have been combined into the larger set. Otherwise, if the subsets are disjoint, and hence there is no overlap in the serving BSs, multiple clusters will be formed in the target area. Multiple clusters can also result, if some subsets have been combined, but the resulting larger sets remain disjoint. Note that with larger numbers of users within the target area, it becomes increasingly likely that only a single cluster will be formed from all 9 BSs in the target area; however, this is not necessarily the case in general. After clusters are formed, the central processor communicates

its decision on the clusters to the users[2].

Let the aggregate downlink channel of the kth user from all B_c BSs in the cth cluster be denoted by $\mathbf{H}_{c,k} = [\mathbf{H}_{c,k}(1), \cdots, \mathbf{H}_{c,k}(B_c)]$, where $\mathbf{H}_{c,k}(b) \in \mathcal{C}^{M \times N}$ denotes the downlink channel matrix between the kth user and bth BS of the cluster, which accounts for path loss, log-normal shadowing and Rayleigh fading. Each element of $\mathbf{H}_{c,k}(b)$ is denoted by $h_{c,k}(b, m, n)$, which is the complex downlink channel signal strength coefficient between the mth receiving antenna of the kth user and the nth transmitting antenna of the bth BS in the cth cluster, as given by

$$h_{c,k}(b, m, n) = z_{c,k}(b, m, n) \sqrt{\Gamma_0 P_t(b) \left(\frac{R_m}{d_{c,k}(b)} \right)^{\alpha} \rho_{c,k}(b) A(\theta, b)}, \tag{3.1}$$

where $z_{c,k}(b, m, n)$ is an i.i.d. complex Gaussian random variable with zero mean and unity variance representing small-scale frequency-flat Rayleigh fading. R_m is the macrocell radius (see Fig. 3.1), which is also the reference distance, and equal to 5 times the picocell radius. Γ_0 is a scaling factor controlling the reference SNR at the distance R_m along the macrocell antenna boresight, $d_{c,k}(b)$ is the distance between user k and BS b in cluster c, and α is the path loss exponent, which may or may not be different for the different tiers of the HetNet. $P_t(b)$ is the transmit power of BS b, $\rho_{c,k}(b)$ denotes the log-normal shadow fading coefficient with standard deviation σ_ρ, and $A(\theta, b)$ is the antenna pattern of a macro BS sector, where θ is the angle between the direction of interest and the boresight of the antenna [78]. For the pico BSs with omnidirectional antennas, $A(\theta, b)$ is equal to unity.

Defining $\mathbf{s}_{c,k} \in \mathcal{C}^{M \times 1}$ as the data vector of user k in cluster c, it is cooperatively

[2]Although we have not done so, it would be trivial for the central processor to apply a weight or bias to certain transmission nodes when forming clusters, to favour certain parts of the HetNet. This could, for instance, allow a specific tier to be emphasized, to make it more likely that users receive data from those nodes, such as perhaps in the case the system operator wished to prioritize traffic being delivered from smaller cells, or perhaps to avoid clustering on macrocells if they periodically employed transmission blanking [20] as a further method to manage interference.

precoded from all B_c BSs using the precoding aggregate matrix $\mathbf{W}_{c,k} \in \mathcal{C}^{B_c N \times M}$. The received signal $\mathbf{y}_k \in \mathcal{C}^{M \times 1}$ for user k is given by

$$\mathbf{y}_k = \mathbf{H}_{c,k} \sum_{j=1}^{U_c} \mathbf{W}_{c,j} \mathbf{s}_{c,j} + \underbrace{\sum_{\hat{c} \in \mathcal{A}_0, \hat{c} \neq c} \mathbf{H}_{\hat{c},k} \sum_{j=1}^{U_{\hat{c}}} \mathbf{W}_{\hat{c},j} \mathbf{s}_{\hat{c},j} + \sum_{\check{c} \notin \mathcal{A}_0} \mathbf{H}_{\check{c},k} \sum_{j=1}^{U_{\check{c}}} \mathbf{W}_{\check{c},j} \mathbf{s}_{\check{c},j} + \mathbf{n}_k}_{\mathbf{Z}_{c,k}} . \tag{3.2}$$

where the first term is the received signal from cluster c, to which the user belongs, while the second and third terms describe the interference from other clusters within or outside of the target area, respectively. $\mathcal{A}_0$ refers to the set of clusters formed in the target area. Applying the central limit theorem, the total interference signal from outside the target area is approximated by an $M \times 1$ complex Gaussian random variable with zero mean and covariance matrix $\sigma_I^2 \mathbf{I}_M$. It is assumed that all BSs outside the target area are transmitting with full power, which results in the worst case scenario for ICI. The interference from these BSs experienced at the center of the target area is determined via Monte Carlo simulation over many channel realizations and the standard deviation of this interference is used as the value of σ_I. Our simulations have shown that the interference and its standard deviation at the center of the area are quite representative of the interference over the entire target area; hence, this value is applied for all locations within the area. For interfering nodes inside the target area, the interference is calculated using the known channel matrices. The last term $\mathbf{n}_k \in \mathcal{C}^{M \times 1}$ is a complex additive white Gaussian noise vector with each element having zero mean and unity variance. The summation of the last three terms is denoted by $\mathbf{Z}_{c,k}$, which indicates the noise plus interference. To suppress the noise and the interference from any transmissions outside of cluster c, a whitening filter is applied at the receiver [27, 79], which is denoted by $\mathbf{Q}_{\mathbf{Z}_{c,k}}^{-1/2}$,

where $\mathbf{Q}_{\mathbf{Z}_{c,k}}$ is the covariance matrix of $\mathbf{Z}_{c,k}$ and calculated as

$$\mathbf{Q}_{\mathbf{Z}_{c,k}} = \mathbf{I}_M + \sum_{\forall \tilde{c} \neq c} \mathbf{H}_{\tilde{c},k} \left(\sum_{j=1}^{U_{\tilde{c}}} \mathbf{W}_{\tilde{c},j} \mathbf{Q}_{\tilde{c},j} \mathbf{W}_{\tilde{c},j}^H \right) \mathbf{H}_{\tilde{c},k}^H, \tag{3.3}$$

where $\mathbf{Q}_{\tilde{c},j} = \mathbb{E}(\mathbf{s}_{\tilde{c},j} \mathbf{s}_{\tilde{c},j}^H)$, and $\mathbf{W}_{\tilde{c},j}$ is the aggregate precoding matrix of user j belonging to cluster $\tilde{c}$. Hence, by defining $\tilde{\mathbf{H}}_{c,k} = \mathbf{Q}_{\mathbf{Z}_{c,k}}^{-1/2} \mathbf{H}_{c,k}$ as the post-processed equivalent channel matrix and $\tilde{\mathbf{Z}}_{c,k} = \mathbf{Q}_{\mathbf{Z}_{c,k}}^{-1/2} \mathbf{Z}_{c,k}$ as the whitened interference plus noise, (3.2) is revised as

$$\tilde{\mathbf{y}}_k = \tilde{\mathbf{H}}_{c,k} \sum_{j=1}^{U_c} \mathbf{W}_{c,j} \mathbf{s}_{c,j} + \tilde{\mathbf{Z}}_{c,k}. \tag{3.4}$$

Given a set of users with encoding order π_c^j, $j \in \{1, 2, \cdots, |\mathcal{D}_s|\}$ where $\mathcal{D}_s$ is a set of all possible combinations of the ordered users belonging to cluster c and defining the user encoded at position k as $\pi_{c,k}^j$, the post-processed received signal in (3.4) can be modified and expanded as

$$\tilde{\mathbf{y}}_{\pi_{c,k}^j} = \tilde{\mathbf{H}}_{c,\pi_{c,k}^j} \mathbf{W}_{c,\pi_{c,k}^j} \mathbf{s}_{c,\pi_{c,k}^j} + \tilde{\mathbf{H}}_{c,\pi_{c,k}^j} \sum_{i<k} \mathbf{W}_{c,\pi_{c,i}^j} \mathbf{s}_{c,\pi_{c,i}^j} + \tilde{\mathbf{H}}_{c,\pi_{c,k}^j} \sum_{i>k} \mathbf{W}_{c,\pi_{c,i}^j} \mathbf{s}_{c,\pi_{c,i}^j} + \tilde{\mathbf{Z}}_{c,\pi_{c,k}^j}. \tag{3.5}$$

The transmission for user $\pi_{c,i}^j$, $1 \leq i \leq U_c$, $i \neq k$, is intra-cluster interference for user $\pi_{c,k}^j$ as given by the two summations in (3.5), which model the multi-user interference. If the intra-cluster interference and $\tilde{\mathbf{Z}}_{c,\pi_{c,k}^j}$ are independent and the intra-cluster interference is known non-causally at the transmitter, according to the dirty paper coding theorem the capacity of the channel is equal to the case with no interference [54]. Assuming the proper design of the precoding matrices $\mathbf{W}_{c,\pi_{c,k}^j}$, $k \in \{1, \cdots, U_c\}$, the maximum achievable sum rate R in the target area

is given by

$$R = \sum_{c=1}^{|\mathcal{A}_0|} \max_{\pi_c^j : 1 \le j \le U_c!} \left\{ \mathbf{Q}_{c,\pi_{c,k}^j} \right\}_{k \in \{1,\cdots,U_c\}} \max_{: \mathbf{Q}_{c,\pi_{c,k}^j} \succeq \mathbf{0}, \sum_{\forall k} Tr\left(\mathbf{Q}_{c,\pi_{c,k}^j}\right) \le 1} \sum_{k=1}^{U_c} R_{c,\pi_{c,k}^j}, \tag{3.6}$$

where $R_{c,\pi_{c,k}^j}$ is the achievable rate of each user and $\mathbf{Q}_{c,\pi_{c,k}^j}$ is the transmit covariance matrix[3] for user $\pi_{c,k}^j$. In the following discussion on precoding methods, the methodology is the same within each cluster. Without loss of generality, we may consider an arbitrary cluster in the target area for the rest of our analysis, i.e. one specific cluster c in this target area. However, an identical approach is applied to all clusters that are formed in the target area. Therefore, we drop the c subscript for ease of notation where appropriate.

In BD [55], the MUI is canceled by designing the precoding matrix of each user such that it is constrained to lie in the nullspace of the aggregate channel matrix $\bar{\mathbf{H}}_k = \left[\tilde{\mathbf{H}}_{\pi_1^j}^T, \cdots, \tilde{\mathbf{H}}_{\pi_{k-1}^j}^T, \tilde{\mathbf{H}}_{\pi_{k+1}^j}^T, \cdots, \tilde{\mathbf{H}}_{\pi_{U_c}^j}^T\right]^T$ of all scheduled users excluding π_k^j. Given $\mathbf{W}_{\pi_k^j}$ as the precoding matrix of user π_k^j, the multiplication of the channels of the other users by $\mathbf{W}_{\pi_k^j}$ results in zero, i.e. $\tilde{\mathbf{H}}_{\pi_i^j}\mathbf{W}_{\pi_k^j} = \mathbf{0}$ for all $i \neq k$, $1 \le (i,k) \le U_c$. Therefore, the rank of the nullspace of $\bar{\mathbf{H}}_k$ should be greater than zero, which (assuming full-rank channels) implies that the number of served users in each cluster, U_c, is equal or less than the total transmitting antennas per cluster divided by the number of receiving antennas per user, rounded up (i.e. $U_c \le \lceil B_c N/M \rceil$). Denoting the singular value decomposition (SVD) of $\bar{\mathbf{H}}_k$ as $\bar{\mathbf{H}}_k = \bar{\mathbf{U}}_k \bar{\mathbf{\Lambda}}_k \left[\bar{\mathbf{V}}_k^1 \bar{\mathbf{V}}_k^0\right]^H$, for a given ordered user π_k^j, its achievable rate, $R_{\pi_k^j}$, is

[3]The sum-trace (transmit power) constraint of 1 on $\mathbf{Q}_{c,\pi_k^j}$ in (3.6) assumes the transmit power $P_t(b)$ for BS b is embedded in the channel matrix, as we have done in (3.1); this method is equivalent to not embedding the power in the channel matrix, but instead using a transmit power constraint of $P_t(b)$ for BS b. Embedding the power in the channel matrix also allows the rates to be calculated independently of the type of transmission nodes (and their specific transmitted power levels) used in the network. For strict accuracy, a per-BS power constraint should be considered, but for simplicity of calculation, we have used a sum-power constraint over all coordinated BS antennas instead.

given by [55]

$$R_{\pi_k^j} = \log_2 \left| \mathbf{I}_M + \left(\tilde{\mathbf{H}}_{\pi_k^j} \bar{\mathbf{V}}_k^0 \right) \mathbf{Q}_{\pi_k^j} \left(\tilde{\mathbf{H}}_{\pi_k^j} \bar{\mathbf{V}}_k^0 \right)^H \right|, \tag{3.7}$$

where $\bar{\mathbf{V}}_k^0 \in \mathcal{C}^{B_c N \times (B_c N - \bar{r}_k)}$, in which $\bar{r}_k$ is the rank of $\bar{\mathbf{H}}_k$, denote the matrix of the rightmost right-singular vectors of $\bar{\mathbf{H}}_k$, which is a set of vectors comprising an orthonormal basis for the nullspace of $\bar{\mathbf{H}}_k$. Defining the effective channel matrix for user π_k^j as $\mathbf{G}_{\pi_k^j} = \tilde{\mathbf{H}}_{\pi_k^j} \bar{\mathbf{V}}_k^0$, the solution of (3.6) using (3.7) as $R_{\pi_k^j}$ is obtained by waterfilling over $\mathbf{G} = diag \left[\mathbf{G}_{\pi_1^j}, \cdots, \mathbf{G}_{\pi_{U_c}^j} \right]$ considering the sum power constraint of 1 [55, 76].

In SZF and SZF-DPC [30], the user encoding order is important, i.e. different orders of a subset of users will result in different sum rates. Assuming full-rank channels, U_c users can be served simultaneously with SZF or SZF-DPC if $U_c \leq \lceil B_c N / M \rceil$. In SZF-DPC, the effect of the first summation in (3.5) is removed by the use of dirty paper coding. The second summation in (3.5) will be canceled if the precoding matrix $\mathbf{W}_{\pi_k^j}$ is constrained to lie in the nullspace of all users encoded before π_k^j. Let $\bar{\bar{\mathbf{H}}}_{k-1} = \left[\tilde{\mathbf{H}}_{\pi_1^j}^T, \cdots, \tilde{\mathbf{H}}_{\pi_{k-1}^j}^T \right]^T$. Using the SVD of $\bar{\bar{\mathbf{H}}}_{k-1} = \bar{\bar{\mathbf{U}}}_{k-1} \bar{\bar{\mathbf{\Lambda}}}_{k-1} \left[\bar{\bar{\mathbf{V}}}_{k-1}^1 \bar{\bar{\mathbf{V}}}_{k-1}^0 \right]^H$, for a given ordered user π_k^j, its achievable rate $R_{\pi_k^j}$ is given by [30]

$$R_{\pi_k^j} = \log_2 \left| \mathbf{I}_M + \left(\tilde{\mathbf{H}}_{\pi_k^j} \bar{\bar{\mathbf{V}}}_{k-1}^0 \right) \mathbf{Q}_{\pi_k^j} \left(\tilde{\mathbf{H}}_{\pi_k^j} \bar{\bar{\mathbf{V}}}_{k-1}^0 \right)^H \right|, \tag{3.8}$$

where $\bar{\bar{\mathbf{V}}}_{k-1}^0 \in \mathcal{C}^{B_c N \times (B_c N - \bar{\bar{r}}_{k-1})}$, in which $\bar{\bar{r}}_{k-1}$ is the rank of $\bar{\bar{\mathbf{H}}}_{k-1}$, denotes the matrix of rightmost right-singular vectors of $\bar{\bar{\mathbf{H}}}_{k-1}$; $\bar{\bar{\mathbf{V}}}_0^0 \triangleq \mathbf{I}_{B_c N}$. Assuming $\bar{\mathbf{G}}_{\pi_k^j} = \tilde{\mathbf{H}}_{\pi_k^j} \bar{\bar{\mathbf{V}}}_{k-1}^0$, and similar to BD, the solution of (3.6) by using $R_{\pi_k^j}$ as presented in (3.8) is obtained by waterfilling over $\bar{\mathbf{G}} = diag \left[\bar{\mathbf{G}}_{\pi_1^j}, \cdots, \bar{\mathbf{G}}_{\pi_{U_c}^j} \right]$ considering the sum power constraint of 1 [30, 76].

Similarly to SZF-DPC, successive precoding (specifically, successive nulling) is used in SZF, but the signals for previously encoded users now interfere since DPC is no longer applied to them. Therefore, the achievable rate $R_{\pi_k^j}$ for user π_k^j

is given by [30]

$$R_{\pi_k^j} = \log_2 \frac{\left| \mathbf{I}_M + \tilde{\mathbf{H}}_{\pi_k^j} \left(\sum_{i=1}^{k} \bar{\bar{\mathbf{V}}}_{i-1}^0 \mathbf{Q}_{\pi_i^j} \left(\bar{\bar{\mathbf{V}}}_{i-1}^0 \right)^H \right) \tilde{\mathbf{H}}_{\pi_k^j}^H \right|}{\left| \mathbf{I}_M + \tilde{\mathbf{H}}_{\pi_k^j} \left(\sum_{i=1}^{k-1} \bar{\bar{\mathbf{V}}}_{i-1}^0 \mathbf{Q}_{\pi_i^j} \left(\bar{\bar{\mathbf{V}}}_{i-1}^0 \right)^H \right) \tilde{\mathbf{H}}_{\pi_k^j}^H \right|}. \tag{3.9}$$

Since the optimization problem in (3.6) using (3.9) as $R_{\pi_k^j}$ is not a convex problem[4], a suboptimal method of calculating the covariance matrices was proposed in [30], which we have applied herein.

Since the SZF sum-rate optimization problem is less constrained than that for BD, it is expected that the achievable sum rates using SZF should exceed (or at least be no worse than) those obtainable when using BD. However, in [33], it was observed that at medium to high SNRs and especially as the number of transmit antennas grows, the performance of SZF becomes inferior compared to BD, even if an optimal scheduler is used for both. This is because the method to calculate SZF covariance matrices from [30] is increasingly suboptimal under those conditions. To correct for this method, [33] proposed an improved method to calculate the SZF covariance matrices based on conjugate gradient projection (CGP). This method guarantees that the performance of SZF is no worse than that of BD, and in general is much better. We also directly apply this improved method herein, using the CGP algorithm to calculate transmit filter and covariance matrices for the users in each cluster. This method is named as SZF(CGP) in this thesis. We refer the reader to [33] for details on the algorithm.

[4]While $\log x$ is concave for $x > 0$ and $log|\mathbf{X}|$ is concave for a positive definite matrix $\mathbf{X}$, $\log(|\mathbf{A}|/|\mathbf{B}|) = \log|\mathbf{A}| - \log|\mathbf{B}|$ is a difference of two concave functions, and therefore is in general neither convex nor concave [80]. Note also that $\mathbf{Y} = \mathbf{A}\mathbf{A}^H$ is always a positive semi-definite matrix, and if $\mathbf{Z} = \mathbf{I} + \mathbf{Y}$, where $\mathbf{Y}$ is positive semi-definite, then $\mathbf{Z}$ is strictly positive definite.

3.3 Scheduling Algorithms

The basis of our proposed user scheduling algorithms to maximize the sum rate is their heuristic nature of exploring the search space to find possible solutions. The sum rate maximization problem for a given subset of users and precoding method is actually an optimization problem of finding beamforming vectors and allocating power in order to maximize the utility function. While the precoding optimization can be solved relatively straightforwardly (although sometimes sub-optimally) in conjunction with the use of fairly low-complexity precoding methods such as SZF or SZF-DPC, the user selection problem is far more difficult. For user scheduling, the optimal solution would be an exhaustive search over all possible combinations and subsets of users, which requires a prohibitively large number of calculations, especially when the number of users requesting service becomes large. Furthermore, in methods like SZF and SZF-DPC where the encoding order affects the sum rate, an additional layer of complexity is added by necessitating a search over all possible orders for each subset of users. Hence, it is necessary to find a suboptimal scheduling method with reduced complexity. In this section, three different reduced-complexity user scheduling algorithms are proposed and discussed.

3.3.1 Particle Swarm (PS) User Scheduling Algorithm

Particle swarm optimization is inspired by the natural behavior of herds of animals, such as birds. One may picture a group of birds flying over a landscape and searching for a specific target; PS as a heuristic optimization technique similarly explores the search space of a problem by using a set of particles. The initial set of particles is randomly generated, after which they can migrate over the solution space searching for the optimal solution according to the design criteria.

In the PS algorithm, denoting $\mathcal{D}_s$ as the search space, there are P particles

representing scheduling solutions, each particle an L-dimensional vector, where $L = \min\left(\lceil B_c N/M \rceil, K_c\right)$ is the maximum number of users out of all available K_c users ($L \leq K_c$) that can be served by cluster c. Each particle has a position in the search space denoted as $\mathbf{x}_i = [x_{i,1}, \cdots, x_{i,L}] \in \mathcal{D}_s$, where $i \in \{1, \cdots, P\}$. Each element in $\mathbf{x}_i$, defined as $x_{i,l}$, where $l \in \{1, \cdots, L\}$, can take on any integer value in the range of 0 to K_c, which indicates what user in the pool of users in the cluster can be placed in the lth position in the scheduled vector $\mathbf{x}_i$. $x_{i,l} = 0$ means that the corresponding spot in the encoding order is not assigned in the scheduling vector[5]. For each $\mathbf{x}_i$, there is a velocity vector $\mathbf{v}_i = [v_{i,1}, \cdots, v_{i,L}]$ where $|v_{i,l}| \leq V_{max}$. The particle's velocity should not become an arbitrary value, since higher velocities would overshoot the optimum solution, and lower values might cause the particle to get trapped in local optima. Therefore, each particle's velocity is limited per element to the interval of $[-V_{max}, +V_{max}]$ to guarantee the convergence of the algorithm in addition to helping ensure that particles stay within the search region.

In the nth iteration of the PS algorithm, the best position visited (i.e. the best solution found) by the ith particle so far is denoted by $\check{\mathbf{x}}_i(n)$, while the overall best position visited by all the particles is named as $\hat{\mathbf{x}}(n)$. Using $\mathbf{v}_i(n)$ as the current value of the velocity of the ith particle in the nth iteration, the velocity is updated (see Fig. 3.3 for a graphical representation) as [67–69]

$$\mathbf{v}_i(n+1) = \underbrace{w(n)\mathbf{v}_i(n)}_{\mathbf{y}_1} + \underbrace{c_1\boldsymbol{\varphi}_1 \odot \left(\check{\mathbf{x}}_i(n) - \mathbf{x}_i(n)\right)}_{\mathbf{y}_2} + \underbrace{c_2\boldsymbol{\varphi}_2 \odot \left(\hat{\mathbf{x}}(n) - \mathbf{x}_i(n)\right)}_{\mathbf{y}_3}, \quad (3.10)$$

where c_1 and c_2 are stochastic acceleration weighting constants, which influence

[5]This means in practice that all spots after that spot in the encoding order will also be zero. One cannot, for example, have no user encoded third in the order, yet still have users encoded fourth or beyond.

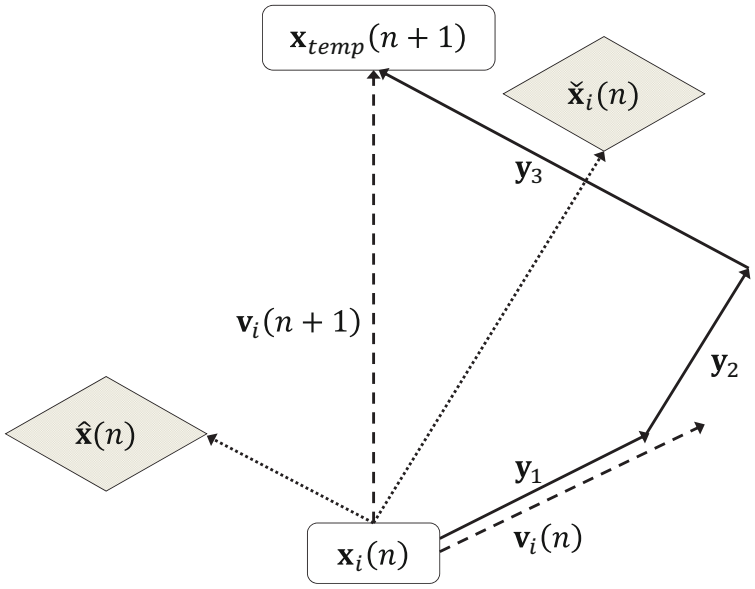

Figure 3.3: Velocity and position update diagram for particle i (cf. [69])

the particles' random movement around the solution region[6]. $\boldsymbol{\varphi}_1$ and $\boldsymbol{\varphi}_2$ are two random length-L vectors whose elements are uniformly distributed in the range of $(0, 1)$, where each represents the acceleration of the particle towards $\check{\mathbf{x}}_i(n)$ and $\hat{\mathbf{x}}(n)$, respectively. The inertia weight $w(n)$ is defined as

$$w(n) = w_s - n \times \left(\frac{w_s - w_e}{n_{max}} \right), \tag{3.11}$$

where w_s and w_e are, respectively, the starting and ending values for the inertia weight, which are obtained experimentally [67, 68], and n_{max} is the maximum number of iterations of the PS algorithm. In the initial stage of execution, $w(n)$ is close to w_s and the algorithm explores more of the entire search area, while for n close to n_{max}, $w(n)$ approaches w_e, where it helps the algorithm to converge to a suboptimal solution [67, 68]. All P particles have the same inertia weight during each iteration n.

[6]Their values are determined experimentally by choosing different values for them and comparing the sum rate results with optimal results, which in our case can be obtained by exhaustive search.

Using the updated velocity $\mathbf{v}_i(n+1)$, the intermediate temporary position $\mathbf{x}_{temp}$ is defined as

$$\mathbf{x}_{temp} = \mathbf{x}_i(n) + \mathbf{v}_i(n+1). \tag{3.12}$$

Since the elements $x_{i,l}$ are integers in $[0, K_c]$, the elements of $\mathbf{x}_{temp}$ may be any real value in the interval $[-V_{max}, K_c+V_{max}]$. Thus, they must be rounded or otherwise adjusted to obtain valid integer values from 0 to K_c (i.e. so the particles don't leave the valid search region); see also Sections 3.3.1 and 3.3.1 for more on this. Then, the result of this modification will become the updated position[7] $\mathbf{x}_i(n+1)$. The value of V_{max} is set as K_c. We have also considered and simulated other values including $V_{max} = 2K_c$ and $V_{max} = K_c/2$, which yielded no significant difference in the performance of the algorithm. Let the utility function for particle $\mathbf{x}_i$ at iteration n be defined as $s_{\mathbf{x}_i(n)} = \sum_k R_{\pi_k^j}$ as in (3.6), where $\mathbf{x}_i(n)$ is used for[8] π^j. $\check{\mathbf{x}}_i(n)$ is replaced by $\mathbf{x}_i(n)$ if the value of $s_{\mathbf{x}_i(n)}$ is larger than $s_{\check{\mathbf{x}}_i(n)}$. Furthermore, if $s_{\mathbf{x}_i(n)}$ is larger than the sum rate of the best solution found over all the particles so far, then $\hat{\mathbf{x}}(n)$ will be updated by $\mathbf{x}_i(n)$. The algorithm is iterated until some desired stop criteria are satisfied. In our algorithm, two factors are considered to halt the execution. The algorithm will run for a maximum of n_{max} iterations. However, the algorithm also monitors for changes in the best sum rate $s_{\check{\mathbf{x}}_i(n)}$ seen by each particle. If no particle has $s_{\check{\mathbf{x}}_i(n)}$ change in the past $n_{stalled}$ iterations, the algorithm is declared stalled and execution terminates. With this latter condition, the algorithm will stop when it converges, even before getting to n_{max}. Hence, the computation time may be reduced considerably. The pseudocode for the particle swarm user scheduling algorithm is described in Algorithm 1.

Considering the performance of the PS algorithm, the population size (number of particles in the swarm) is an important factor for both the convergence of the

[7] If not for the rounding and border concerns, $\mathbf{x}_i(n+1)$ could be directly updated by (3.12), which is more typical in a PS algorithm (cf. [67,68]).

[8] $\mathbf{x}_i(n)$ might be padded with zeros if the number of served users $U_c < L$; these zeros would be removed from π^j.

Algorithm 1 Particle Swarm (PS) User Scheduling Algorithm for HetNet MIMO

for all clusters $c \in \mathcal{A}_0$ in parallel **do**

Initialize: $stalled = \mathsf{false}$, $n = 1$, n_{max}, $n_{stalled}$, w_s, w_e, c_1, c_2, P, V_{max}, $w = w_s$, $s_{\check{\mathbf{x}}_i(n)} = 0$, $s_{\hat{\mathbf{x}}(n)} = 0$;

$\mathbf{x}_i(1)$ = a vector of up to L ordered users randomly chosen from $\mathcal{D}_s$, $\forall i \in \{1, \cdots, P\}$;

$\mathbf{v}_i(1)$ = a vector of L velocities for the elements of $\mathbf{x}_i(1)$, uniformly randomly chosen from $[-V_{max}, +V_{max}]$, $\forall i \in \{1, \cdots, P\}$;

while $n \leq n_{max}$ AND $stalled = \mathsf{false}$ **do**

Compute $\mathbf{s}(n) = [s_{\mathbf{x}_1(n)}, \cdots, s_{\mathbf{x}_P(n)}]$.

for $i = 1$ to P **do**

if $s_{\mathbf{x}_i(n)} > s_{\check{\mathbf{x}}_i(n)}$ **then**

$\check{\mathbf{x}}_i(n) = \mathbf{x}_i(n)$; $s_{\check{\mathbf{x}}_i(n)} = s_{\mathbf{x}_i(n)}$;

end if

if $s_{\mathbf{x}_i(n)} > s_{\hat{\mathbf{x}}(n)}$ **then**

$\hat{\mathbf{x}}(n) = \mathbf{x}_i(n)$; $s_{\hat{\mathbf{x}}(n)} = s_{\mathbf{x}_i(n)}$;

end if

end for

for $i = 1$ to P **do**

$\varphi_{1,j} = rand(\sim \mathcal{U}(0,1))$ and $\varphi_{2,j} = rand(\sim \mathcal{U}(0,1))$, $\forall j \in \{1, \cdots, L\}$;

$\mathbf{v}_i(n+1) = w(n)\mathbf{v}_i(n) + c_1\boldsymbol{\varphi}_1 \odot \Big(\check{\mathbf{x}}_i(n) - \mathbf{x}_i(n)\Big) + c_2\boldsymbol{\varphi}_2 \odot \Big(\hat{\mathbf{x}}(n) - \mathbf{x}_i(n)\Big)$;

for each element $v_{i,j}$ of $\mathbf{v}_i(n+1)$ with $|v_{i,j}| > V_{max}$ **do**

Set $v_{i,j} = sgn(v_{i,j}) \times V_{max}$.

end for

$\mathbf{x}_{temp} = \mathbf{x}_i(n) + \mathbf{v}_i(n+1)$;

$\mathbf{x}_i(n+1)$ is the rounded and revised version of $\mathbf{x}_{temp}$ based on the defined border behavior of the valid region of user indices.

end for

$w(n) = w_s - n \times \left(\frac{w_s - w_e}{n_{max}}\right)$; $n = n + 1$;

if for all particles in the last $n_{stalled}$ iterations $s_{\check{\mathbf{x}}_i(n)}$ has not changed, **then** $stalled = \mathsf{true}$; **end if**

end while

end for

algorithm and its complexity. In general, the product of the iterations and the population size should be large enough for the algorithm to sufficiently converge to a good solution, but there is no exact rule for this. However, the appropriate population size will depend on the specific problem being considered. For small search spaces (in terms of the total number of possible solutions), the population can usually be considered as a small proportion of the size of the search space, e.g. 20%, while for large search spaces where this proportion is an unfeasibly large number, 20 to 50 particles can often be considered good enough [67, 68]. The behaviour of the particles when they approach the borders of the search space can be different. In this thesis, we consider two different approaches.

Reflective Borders

With reflective borders, the particles reflect or bounce off the borders of the search region, much like light does off a mirror or a ball off a wall, i.e. the borders reflect the particle back into the search region (see Fig. 3.4 for more details). For large values of V_{max}, the reflection may repeat until the particle falls within the valid region of user indices. As an example, assume that we can choose at most $L = 7$ users out of a pool of $K_c = 10$ users, we set $V_{max} = 10$. Let us furthermore assume that one particle's position in the nth iteration is $\mathbf{x}(n) = [2, 5, 1, 7, 0, 0, 0]$ and its corresponding (updated) velocity is $\mathbf{v}(n+1) = [-8.9, 2.0, 10, 0.7, -0.2, -3.3, 0.4]$. (The third element in $\mathbf{v}(n+1)$ was larger than V_{max}, and thus was replaced by V_{max}). Using (3.12) to update the particle's position (and rounding it to integer values), we will have $\mathbf{x}_{temp} = [-7, 7, 11, 8, 0, -3, 0]$. Elements of this particle that would cross the borders are reflected instead. For this example, the resulting reflected position will be $\mathbf{x}_b = [7, 7, 9, 8, 0, 3, 0]$. If any digits are repeated within the position vector, the algorithm will keep the first replica in its position and replace the others with zeros (since a given user cannot be scheduled more than

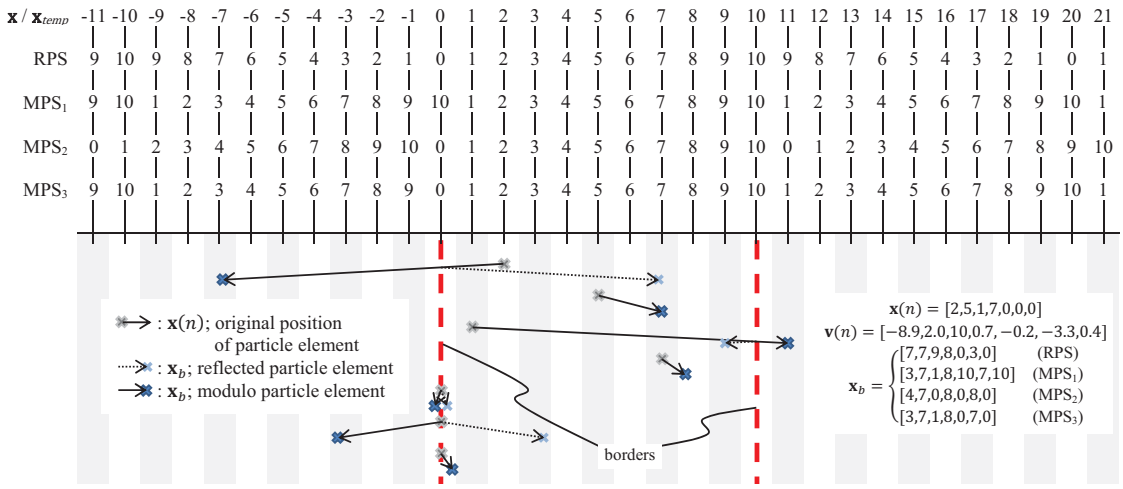

Figure 3.4: Depiction of effect of border operations on particle elements; $L = 7$, $K_c = 10$.

once at the same time). All zeros are then shifted to the end of the vector[9]. The final version of the updated position for the example is $\mathbf{x}(n+1) = [7, 9, 8, 3, 0, 0, 0]$. We name this method as reflective PS (RPS).

Modulo Borders

In the following variations, the border constraints are handled by a modulo operation using the temporary updated position (i.e. $\mathbf{x}_{temp}$ from (3.12)) and the number of users in cluster c (i.e. the value of K_c).

Type 1: The position vector is mapped according to $\{[\mathbf{x}_{temp} - 1] \bmod K_c\} + 1$. In this case, any zeros in the vector are mapped to the value of K_c. Again considering the above example, the updated position vector $\mathbf{x}_{temp} = [-7, 7, 11, 8, 0, -3, 0]$ considering $K_c = 10$ will be mapped to $\mathbf{x}_b = [3, 7, 1, 8, 10, 7, 10]$. After, similarly to the RPS approach above, replicated digits are again replaced by zeros, and zeros are shifted to the end of the vector. Hence, the final updated position under

[9]The elements of the velocity could also potentially be shifted or otherwise updated along with mapping $\mathbf{x}_{temp}$ to a "valid" position with non-repeated non-zero digits. We have not done so in our work. Doing so would for the most part just affect how long it takes for the algorithm to converge to a solution, although the effect would by no means be straightforward or simple to predict.

this approach will be $\mathbf{x}(n+1) = [3, 7, 1, 8, 10, 0, 0]$. This approach is named as modulo PS of type 1 (MPS_1).

Type 2: Assume that the vector of $\mathbf{x}_{temp}$ has a zero, which means that the corresponding position should not be allocated to any user, or alternatively, in consideration of the upcoming border operation, that fewer than the maximum number of users should be scheduled. MPS_1 would replace this zero with K_c. Thus, especially in cases where a solution would schedule fewer than L users, the MPS_1 variation can be biased towards scheduling user K_c. If the modulo mapping instead is set to $\mathbf{x}_{temp} \bmod (K_c + 1)$, then values of zero and K_c will both be kept by the algorithm, e.g. for $\mathbf{x}_{temp} = [-7, 7, 11, 8, 0, -3, 0]$, the result is $\mathbf{x}_b = [4, 7, 0, 8, 0, 8, 0]$ and $\mathbf{x}(n+1) = [4, 7, 8, 0, 0, 0, 0]$. This approach is named MPS_2.

Type 3: In type 2, the value of $\pm(K_c + 1)$ is set to zero after the modulo operation, as seen above for the "11" in position 3 of $\mathbf{x}_{temp}$. In general, removing a user that was previously set to be scheduled, thereby reducing the number of scheduled users, can be quite detrimental to the system performance, especially at large K_c. This for the most part should be avoided. Therefore, to deal with this, the zeros of $\mathbf{x}_{temp}$ are kept as-is, but the non-zero elements of $\mathbf{x}_{temp}$ are mapped similar to type 1. Thus, the zeros and K_c are treated differently. By using type 3, the next position vector is obtained as $\mathbf{x}_b = [3, 7, 1, 8, 0, 7, 0]$ and $\mathbf{x}(n+1) = [3, 7, 1, 8, 0, 0, 0]$. This method is named MPS_3.

The performance of all three types of modulo approaches is investigated in this work in Section 3.4.

3.3.2 Hybrid of Greedy and Particle Swarm (HGPS) User Scheduling Algorithm

In the literature, scheduling algorithms employing greedy methods have been widely used because of their relative simplicity and low complexity [26, 27, 31, 56].

A greedy algorithm maximizes the incremental utility in each step as it makes its decisions. In this context of scheduling, this means iteratively selecting users that maximize the increase in the scheduling metric. For example, the user with the best channel gain may be selected in the scheduling vector of users as the first user in the encoding order [26, 27]. We have not proposed a purely greedy algorithm within this work, since such algorithms are ubiquitous and have already been studied extensively.

Within this work, greedy-style selection is combined with the PS algorithm to improve its performance; we call this hybrid methodology the HGPS algorithm. Consequently we will have HGRPS and HGMPS variants, respectively, for the reflective HGPS and modulo HGPS border approaches. Rather than operating almost solely at random, the PS uses this greedy information as part of its structure. Since the rate of the ith user is affected by any previously encoded user k, where $k < i$, and trivially following that better channel gains result in higher (single-user) achievable rates, it is reasonable to put the user with the best channel gain in the first position of the scheduling vector. Thus, the achievable rate of this user will be at its highest [27, 31]. According to this idea, for all particles i, the first user in the ordered user vector is fixed as the one with the largest squared Frobenius norm (F-norm) of its channel matrix, that is $x_{i,1} = \arg\max_j \|\tilde{\mathbf{H}}_{c,j}\|_F^2$, $j \in \{1, \cdots, K_c\}$. For the remaining users in the vector, the algorithm operates similarly to the PS algorithm described in the previous subsection. In other words, after choosing the first user as a common user in all particles based on the greedy criterion, this particular user will be omitted from the pool of the users to choose from. Then, the rest of the scheduling vector will be chosen by PS optimization from the remaining pool. Since the first location in all scheduling vectors is fixed, the updating procedure for this location need not occur any more. This is described in Algorithm 2.

Algorithm 2 Hybrid Greedy - Particle Swarm (HGPS) User Scheduling Algorithm for HetNet MIMO

for all clusters $c \in \mathcal{A}_0$ in parallel **do**
  Initialize: $stalled = \mathsf{false}$, $n = 1$, n_{max}, $n_{stalled}$, w_s, w_e, c_1, c_2, P, V_{max}, $w = w_s$,
  $s_{\check{\mathbf{x}}_i(n)} = 0$, $s_{\hat{\mathbf{x}}(n)} = 0$;
  $x_{i,1} = \arg\max_{j \in \{1,\cdots,K_c\}} \|\tilde{\mathbf{H}}_{c,j}\|_F^2$;
  $\mathbf{x}_i(1)$ = a vector of up to L ordered users randomly chosen from $\mathcal{D}_s$, given that the first user is set to $x_{i,1}$ from above, $\forall i \in \{1, \cdots, P\}$;
  $\mathbf{v}_i(1)$ = a vector of L velocities for the elements of $\mathbf{x}_i(1)$, uniformly randomly chosen from $[-V_{max}, +V_{max}]$, $\forall i \in \{1, \cdots, P\}$;
  set $v_{i,1} = 0$, $\forall i$;
  while $n \leq n_{max}$ AND $stalled = \mathsf{false}$ **do**
    Compute $\mathbf{s}(n) = [s_{\mathbf{x}_1(n)}, \cdots, s_{\mathbf{x}_P(n)}]$.
    for $i = 1$ to P **do**
      if $s_{\mathbf{x}_i(n)} > s_{\check{\mathbf{x}}_i(n)}$ **then**
        $\check{\mathbf{x}}_i(n) = \mathbf{x}_i(n)$; $s_{\check{\mathbf{x}}_i(n)} = s_{\mathbf{x}_i(n)}$;
      end if
      if $s_{\mathbf{x}_i(n)} > s_{\hat{\mathbf{x}}(n)}$ **then**
        $\hat{\mathbf{x}}(n) = \mathbf{x}_i(n)$; $s_{\hat{\mathbf{x}}(n)} = s_{\mathbf{x}_i(n)}$;
      end if
    end for
    for $i = 1$ to P **do**
      $\varphi_{1,j} = rand(\sim \mathcal{U}(0,1))$ and $\varphi_{2,j} = rand(\sim \mathcal{U}(0,1))$, $\forall j \in \{2, \cdots, L\}$;
      $\varphi_{1,1} = \varphi_{2,1} = 0$;
      $\mathbf{v}_i(n+1) = w(n)\mathbf{v}_i(n) + c_1\boldsymbol{\varphi}_1 \odot \left(\check{\mathbf{x}}_i(n) - \mathbf{x}_i(n)\right) + c_2\boldsymbol{\varphi}_2 \odot \left(\hat{\mathbf{x}}(n) - \mathbf{x}_i(n)\right)$;
      for each element $v_{i,j}$ of $\mathbf{v}_i(n+1)$ with $|v_{i,j}| > V_{max}$ **do**
        Set $v_{i,j} = sgn(v_{i,j}) \times V_{max}$.
      end for
      $\mathbf{x}_{temp} = \mathbf{x}_i(n) + \mathbf{v}_i(n+1)$;
      $\mathbf{x}_i(n+1)$ is the rounded and revised version of $\mathbf{x}_{temp}$ based on the defined border behavior of the valid region of user indices (in all cases $x_{i,1}$ retains its original value).
    end for
    $w(n) = w_s - n \times \left(\frac{w_s - w_e}{n_{max}}\right)$; $n = n + 1$;
    if for all particles in the last $n_{stalled}$ iterations $s_{\check{\mathbf{x}}_i(n)}$ has not changed,
    then $stalled = \mathsf{true}$; **end if**
  end while
end for

3.3.3 Simulated Annealing (SA) User Scheduling Algorithm

In physics and metallurgy, annealing is a process in which a material first is heated to a sufficiently high temperature and then its temperature is gradually decreased to allow its molecules to rearrange to an improved crystalline structure with reduced energy. In this thesis, we propose a simulated annealing user scheduling method for the maximization of the sum rate. Without memory, the memoryless SA (SA-ml) algorithm could potentially move away from and forget a better solution previously seen in earlier iterations and never return there. Having memory forces the SA with memory (SA-m) algorithm to keep track of the best solution (and its sum rate) seen over all iterations, regardless of if it has since moved on to explore another area of the search space. It can then report that the overall best solution when the algorithm finishes.

The SA algorithm consists of a search space $\mathcal{D}_s$, a utility function s, a parameter τ which is analogously called the temperature, and a cooling function. Given a candidate solution vector $\mathbf{x}$, there is a neighbourhood set, defined by $\mathcal{D}_s(\mathbf{x})$, which is a subset of all candidate solutions except $\mathbf{x}$, i.e. $\mathcal{D}_s(\mathbf{x}) \subseteq \mathcal{D}_s \backslash \{\mathbf{x}\}$, and consists of those solutions which are in some manner "nearby" to $\mathbf{x}$. In general, the nth state (i.e. the solution being considered at iteration n) is the output of the neighbourhood function, $\mathrm{N}(\mathbf{x}(n))$, which operates on the neighbour set of $\mathbf{x}(n)$ (i.e. $\mathcal{D}_s(\mathbf{x}(n))$). The transition from $\mathbf{x}(n)$ to $\mathbf{x}(n+1) \in \mathcal{D}_s(\mathbf{x}(n))$ is a Markov chain with transition probabilities p_j, $j \in \{1, \cdots, |\mathcal{D}_s(\mathbf{x}(n))|\}$ where the summation of all p_js over every possible $\mathbf{x}(n+1)$ equals to one; i.e. $\sum_j p_j = 1$. The neighbourhood function and several example elements of $\mathcal{D}_s(\mathbf{x}(n))$ obtained by the Markov chain state diagram are depicted in Fig. 3.5. The vector of users has up to $L = \min(\lceil B_c N/M \rceil, K_c)$ ordered users randomly chosen from $\mathcal{D}_s$. A zero in this vector means no user is scheduled at that position in the order. In

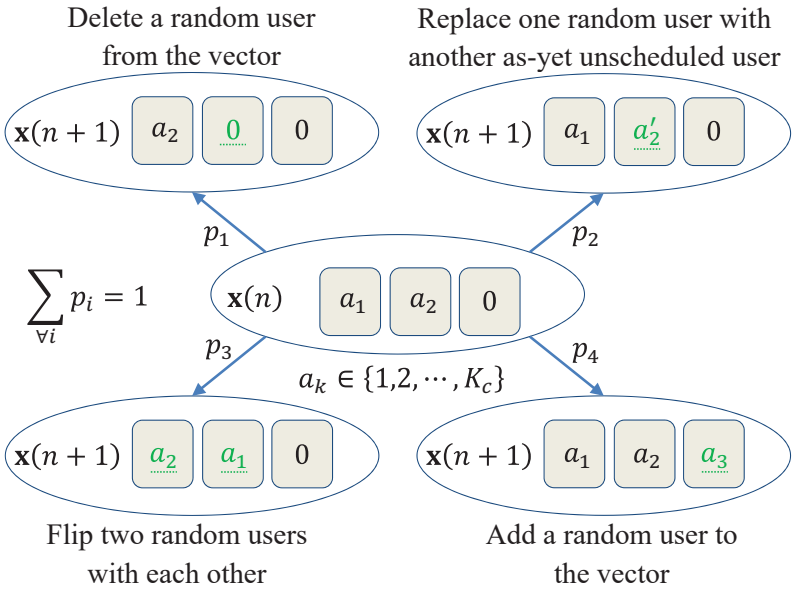

Figure 3.5: Neighbourhood function for a sample vector $\mathbf{x}(n)$ with length 3.

iteration n, the neighbourhood function will randomly select one of four possible actions with probability of p_i, $i = 1, \cdots, 4$, namely:

- Deleting a random user from the vector,
- Replacing one random user with another user not yet scheduled,
- Flipping the order of two random users with each other, or,
- Add a random unscheduled user to the vector (if possible to do so).

Depending on the precoding method and for a given set of candidate users with order π^j, the maximum achievable sum rate of the candidate solution (or the utility function) is calculated by (3.6) and one of the equations (3.7)–(3.9). Defining $\mathbf{x}(n) = [a_1, \cdots, a_L]$ and $\hat{\mathbf{x}}(n) = [u_1, \cdots, u_L]$, respectively, as the vectors of the solution at iteration n and the selected solution found up to the nth iteration, the respective sum rates $s_{\mathbf{x}(n)}$ and $s_{\hat{\mathbf{x}}(n)}$ are calculated as $\sum_k R_{\pi_k^j}$ as in (3.6), where $\mathbf{x}(n)$ and $\hat{\mathbf{x}}(n)$ are used for π^j (see footnote 7). If the solution represented

by $\mathbf{x}(n+1)$ has a better sum rate compared to $s_{\hat{\mathbf{x}}(n)}$, then it will be selected as the solution for the next iteration, i.e., $\hat{\mathbf{x}}(n+1) = \mathbf{x}(n+1)$. If not, it still may be selected as a suboptimal solution according a probability function, given as

$$\mathrm{P}(n) = \exp\left(\frac{-\left(s_{\hat{\mathbf{x}}(n)} - s_{\mathbf{x}(n+1)}\right)}{\tau}\right), \tag{3.13}$$

where the parameter τ is the temperature. This temperature monotonically decreases (cools) and results in decreasing probabilities of accepting suboptimal solutions as the execution of the algorithm progresses. The temperature should be initialized to a value (named as τ_{hot}) such that the algorithm initially accepts suboptimal solutions with higher probabilities, and cooled down gradually until eventually mostly solutions with better utility functions are accepted (i.e., SA reverts to a hill-climbing algorithm in the later stages of execution). The cooling function is given as

$$\tau = \tau_{hot}\phi^{p_1}, \tag{3.14}$$

where ϕ can be chosen randomly or deterministically from the interval of $(0, 1)$, and p_1 is the iteration counter of a loop which gradually reduces the temperature. Algorithm 3 describes the proposed suboptimal SA scheduling algorithm. First, the algorithm runs for an initial training period to find the proper value of τ_{hot} such that the algorithm initially has at least a 90% probability of accepting suboptimal solutions. The duration of this training period is determined by practical experiments[10]. The algorithm has two loops that control how many iterations the algorithm runs before it terminates, i.e. they trade off how closely the algorithm approaches to the optimal solution with the cost of complexity. For the outer

[10]To get the best training interval duration to set τ_{hot}, which helps the main algorithm converge faster with acceptable accuracy compared to the results obtained from exhaustive search, we tested different ranges of iterations for the training interval. Correspondingly, the SA algorithm was run based on their resulting value of τ_{hot}. The shortest training interval which can consequently provide SA scheduling solutions with a performance sufficiently close to exhaustive search was chosen as the training time (i.e. the number of iterations for the training loop).

Algorithm 3 Simulated Annealing (SA) User Scheduling Algorithm for HetNet MIMO, With (SA-m) and Without (SA-ml) Memory

for all clusters $c \in \mathcal{A}_0$ in parallel **do**
  Initialize: τ_{hot}; $n = 1$; $p_1 = 0$; $\tau = \tau_{hot}$; ϕ; B_1; B_2; τ_f;
  $\mathbf{x}(1)$ = a vector of U_c ordered users randomly chosen from $\mathcal{D}_s$;
  $\hat{\mathbf{x}}(1) = \mathbf{x}(1)$; Compute $s_{\hat{\mathbf{x}}(1)}$.
  (SA-m only:) $\mathbf{x}_{best} = \hat{\mathbf{x}}(1)$; $s_{\mathbf{x}_{best}} = s_{\hat{\mathbf{x}}(1)}$.
  while $\tau > \tau_f$ **do**
    $b = 1$;
    while $b < B_2$ **do**
      Find $\mathbf{x}(n+1)$ from the neighbourhood function $\mathrm{N}(\mathbf{x}(n))$.
      Compute $s_{\mathbf{x}(n+1)}$.
      The best solution for the next step is decided as
      $[\hat{\mathbf{x}}(n+1), s_{\hat{\mathbf{x}}(n+1)}] =$

$$\begin{cases} [\mathbf{x}(n+1), s_{\mathbf{x}(n+1)}], & \text{if } s_{\mathbf{x}(n+1)} \geq s_{\hat{\mathbf{x}}(n)}; \\ [\mathbf{x}(n+1), s_{\mathbf{x}(n+1)}], & \text{if } r < \exp\left(\frac{-(s_{\hat{\mathbf{x}}(n)} - s_{\mathbf{x}(n+1)})}{\tau}\right), \\ & \quad \text{where } r = rand\,(\sim \mathcal{U}(0,1)); \\ [\hat{\mathbf{x}}(n), s_{\hat{\mathbf{x}}(n)}], & \text{otherwise.} \end{cases}$$

      (SA-m only:) **if** $s_{\hat{\mathbf{x}}(n+1)} > s_{\mathbf{x}_{best}}$ **then**
        $\mathbf{x}_{best} = \hat{\mathbf{x}}(n+1)$; $s_{\mathbf{x}_{best}} = s_{\hat{\mathbf{x}}(n+1)}$;
      end if
      $b = b + 1$; $n = n + 1$.
    end while
    $p_1 = p_1 + 1$; $\tau = \tau_{hot}\phi^{p_1}$.
  end while
  (SA-ml only:) return $\hat{\mathbf{x}}(n)$, $s_{\hat{\mathbf{x}}(n)}$
  (SA-m only:) return $\mathbf{x}_{best}$, $s_{\mathbf{x}_{best}}$
end for

loop, we assume that there are B_1 steps to cool down the temperature, or equivalently, a final "frozen" temperature $\tau_f = \tau_{hot}\phi^{B_1}$, at which the loop halts. For each temperature τ, we have an inner loop whose length is controlled by a fixed number of iterations[11], B_2.

3.3.4 Complexity Analysis of PS, HGPS and SA

We compare the complexity per cluster of all the proposed user scheduling algorithms in terms of the number of flops[12] required. In prior work [28, 31, 33], the complexity of calculating the sum rate and transmit covariance matrices for a single ordered selection of U_c users under SZF, SZF(CGP), and BD precoding was calculated to be $\mathcal{O}(U_c T^3)$ for all three methods, where T is the total number of transmit antennas. In this work, $T \triangleq B_c N$. We perform a similar calculation for SZF-DPC precoding as follows. The system is assumed to transmit to the maximum possible number of users U_c simultaneously, with $U_c = \lceil T/M \rceil = T/M + \zeta$, or $T = (U_c - \zeta)M$, where $0 \leq \zeta < 1$. For the purposes of order of complexity, the "$-\zeta$" portion can be neglected, and so the product $U_c M$ grows on the same order as T. For SZF-DPC, the following steps are taken, where we assume without loss of generality that the users are ordered by their increasing indices, i.e. user 1 is first, user 2 is second, etc.:

1. For users $k = 2$ to U_c, the aggregate channel matrix $\bar{\bar{\mathbf{H}}}_{k-1} \in \mathcal{C}^{(k-1)M \times T}$ is formed and its nullspace basis vectors $\bar{\bar{\mathbf{V}}}_{k-1}^0$ are found. (This is not required for user 1, as $\bar{\bar{\mathbf{V}}}_0^0 = \mathbf{I}_T$.) The nullspace vectors can be found with an SVD; a QR decomposition may also be done with lower total complexity, but both methods have the same order of complexity. For a complex matrix $\mathbf{A}$ of

[11] We have also tested controlling the inner loop based on the standard deviation of the selected solutions during the loop, halting the loop when it drops below a threshold. However, we found this method performed consistently worse than simply using a fixed inner loop length. Therefore, we do not go into any further detail on this alternative method.

[12] A flop is a real-valued floating point operation [81] which can be used as a basis to describe the complexity of an algorithm.

size $m \times n$ or $n \times m$, where $m \geq n$, the number of flops required for an SVD is $16m^2n + 32mn^2 + 36n^3$ [31, 81]. Thus, for an SVD of all the aggregate matrices, the total flops required is given as

$$\sum_{k=2}^{U_c} \Big(16T^2(k-1)M + 32T((k-1)M)^2 + 36((k-1)M)^3 \Big). \tag{3.15}$$

2. Effective channel matrices are formed as $\mathbf{G}_{\pi_k^j} = \tilde{\mathbf{H}}_{\pi_k^j} \bar{\bar{\mathbf{V}}}_{k-1}^0$. As described in Section 3.2, the matrix $\bar{\bar{\mathbf{V}}}_{k-1}^0$ is of size $T \times (T - \bar{\bar{r}}_{k-1})$, where $\bar{\bar{r}}_{k-1} = (k-1)M$ assuming full-rank channels for every user. The multiplication of a complex $m \times n$ matrix with an $n \times p$ matrix requires $8mnp$ flops. Thus, to form all of the effective channel matrices $\mathbf{G}_{\pi_k^j}$, the total flops required is $\sum_{k=2}^{U_c} 8MT[T - (k-1)M]$.

3. Form the block-diagonal matrix $\mathbf{G} = diag\left[\mathbf{G}_{\pi_1^j}, \cdots, \mathbf{G}_{\pi_{U_c}^j}\right]$ and waterfill for the power allocation. To waterfill, one could find the singular values of $\mathbf{G}$, but it is more efficient to find the singular values of each of the separate $\mathbf{G}_{\pi_k^j}$ matrices instead. For a complex matrix $\mathbf{A}$ of size $m \times n$ or $n \times m$, where $m \geq n$, when only the singular values $\mathbf{\Lambda}$ are needed from the SVD $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^H$, the full SVD does not need to be performed, and only $16mn^2 - \frac{16}{3}n^3$ flops are required [28, 81]. The dimensions of $\mathbf{G}_{\pi_k^j}$ are $M \times [T - (k-1)M]$. The complexity of finding the singular values for all the $\mathbf{G}_{\pi_k^j}$ matrices is thus $\sum_{k=1}^{U_c} \left[16(T - (k-1)M)M^2 - \frac{16}{3}M^3 \right]$.

   In general, for all users, it will hold that $M < T - (k-1)M$. The single exception may be for the final user when $k = U_c$, where it will be that $M \geq T - (U_c - 1)M$. Since $U_c = \lceil T/M \rceil$, equality will occur if T is a multiple of M; strict inequality occurs otherwise. In the worst case, the total number of (real-valued, non-zero) singular values η is $U_c M$. These singular values must first be squared to become eigenvalues, using η flops. Then,

waterfilling over the eigenvalues requires (at most) $2\eta^2 + 6\eta$ flops [28, 56].

4. The sum-throughput from the waterfilling of the block-diagonal matrix in the previous step can be found by $\log_2\left[\prod_{i=1}^{\eta}(1+\lambda_i \mathfrak{p}_i/\sigma^2)\right]$, where λ_i are the eigenvalues from the previous step and $\mathfrak{p}_i$ are their associated waterfilled powers. σ^2 is the AWGN variance of the system. The sum-rate calculation thus requires 1 addition, multiplication, and division for each term (the division potentially may not be required if σ^2 is normalized to 1), and η multiplications to multiply all the terms together in the worst case. (Fewer may be required in practice if, for instance, some of the powers $\mathfrak{p}_i$ allocated to the eigenvalues are zero.) Thus, a total of 4η flops is required, wherein the complexity of the single $\log_2$ operation at the end may be neglected.

Hence, the total complexity for a single SZF-DPC sum-rate calculation is the summation of all these 4 steps. After expanding the sums, collecting terms, and substituting $U_c M = T$, we find the complexity is $\mathcal{O}(U_c T^3)$, resulting from the SVDs in step 1. Hence, for all precoding methods of interest, the order of complexity to calculate the sum rate of the scheduled user vector π^j is the same, i.e. $\mathcal{O}(\Omega)$, where $\Omega = U_c T^3$ for compactness of notation.

In the PS algorithm, in each loop, the algorithm first computes $\mathbf{s}(n) = [s_{\mathbf{x}_1(n)}, \cdots, s_{\mathbf{x}_P(n)}]$, which consists of P sum rate calculations; hence, its complexity is of order $\mathcal{O}(P\Omega)$ flops. The complexity of the comparison with the previous best sum rates can be neglected here, as can the complexity of generating the random $\boldsymbol{\varphi}$ vectors, since both operations are much less complex. For each particle, the velocity update of $\mathbf{v}_i$ has $5L$ multiplications and $4L$ additions; we neglect the complexity of comparing the magnitude of the updated velocities with V_{max}. Similarly, the complexity of updating the position $\mathbf{x}_i$ of each particle requires L flops. We can furthermore neglect the complexity of the rounding and border operations; these will not affect the order of complexity. Thus, the particle velocity and position updates in total require about $10PU_c$ flops, which is $\mathcal{O}(PU_c)$. The

complexity of updating $w(n)$ can be neglected, as it is an $\mathcal{O}(1)$ operation. The worst case complexity (in terms of calculations required) occurs when $U_c = L$. Since the algorithm repeats at most n_{max} iterations, the total complexity of the PS algorithm is thus no higher than $\mathcal{O}(Pn_{max}(\Omega + L)) \cong \mathcal{O}(Pn_{max}\Omega)$. If P is set as $0.2K_c$ (or otherwise proportional to K_c), the complexity of the PS algorithm can be restated as $\mathcal{O}(n_{max}K_cU_cT^3)$.

The complexity of the HGPS algorithm is similar to that of the basic PS. Finding the user among K_c with the largest squared F-norm of its $M \times B_cN$ channel matrix requires $4B_cNMK_c$ flops (for order $\mathcal{O}(TMK_c)$) [31,56]. The order of complexity of calculating the sum rate does not change with one scheduled user being fixed. When updating $\mathbf{v}_i$ and $\mathbf{x}_i$, L would be replaced with $L - 1$, as the first element of the position, being fixed in place, need not be updated. The order of complexity is therefore $\mathcal{O}(TMK_c + n_{max}K_cU_cT^3) \cong \mathcal{O}(n_{max}K_cU_cT^3)$. Thus, the complexity of PS and HGPS are of the same order.

For the SA algorithm, the most complex operation is the calculation of the sum rate. The neighbourhood function in comparison simply makes small changes to the vector of scheduled users, and thus its complexity is much smaller. With a fixed number of iterations B_1 and B_2 in the outer and inner loops, respectively, its total order of complexity will therefore be $\mathcal{O}(B_1B_2U_cT^3)$. In general, at least one of B_1 or B_2 should increase to some degree along with K_c, although for small changes to K_c, they can be kept constant. If we assume that B_1 is set equal to n_{max} from the PS algorithm, then the difference in complexity between the SA and PS algorithms will just depend on the difference between B_2 and P.

As mentioned in Chapter 2, an exhaustive search examines all $\sum_{k=1}^{U_c} k!\binom{K_c}{k}$ possible ordered subsets of users. In each case, it calculates a sum rate for that ordered subset. In some cases, the results of certain operations (e.g. maybe SVDs of channel matrices) could be reused between subsets, but as a general case, each sum rate calculation would be $\mathcal{O}(U_cT^3)$ flops as found earlier. The

Table 3.1: Comparison of Complexity for Different User Scheduling Algorithms For Order-Dependent Precoding Methods

	Complexity in general-case network	Complexity in a set network layout, in which T and U_c are fixed and setting $B_1 = n_{max}$
PS	$\mathcal{O}(n_{max}PU_cT^3)$	$\mathcal{O}(n_{max}P)$
HGPS	$\mathcal{O}(n_{max}PU_cT^3)$	$\mathcal{O}(n_{max}P)$
SA	$\mathcal{O}(B_1B_2U_cT^3)$	$\mathcal{O}(n_{max}B_2)$
Exh. Search	$\mathcal{O}(K_c^{U_c}U_cT^3)$	$\mathcal{O}(K_c^{U_c})$

highest order term occurs when the the exhaustive search considers subsets of cardinality U_c (i.e. when $k = U_c$ in the sum above). Overall, the complexity of the exhaustive search would be $\mathcal{O}\big(U_c!\binom{K_c}{U_c}U_cT^3\big) \cong \mathcal{O}(K_c^{U_c}U_cT^3)$. In the case of BD precoding, where different encoding orders are not relevant, the complexity reduces to $\mathcal{O}\big(\binom{K_c}{U_c}U_cT^3\big)$.

The complexity of the algorithms is summarized and compared in Table 3.1. The last column of the table assumes a set network layout, where the number of transmit antennas T is fixed (and consequently U_c is also fixed); it is also assumed that $B_1 = n_{max}$ and that $n_{max} \gg M$. In Fig. 3.6(a), the order of complexity of exhaustive search, SA and PS algorithms are compared in terms of the number of required flops for $K_c = 7$. In this analysis, we assumed that PS algorithm has 26 particles, and B_2 in SA algorithm is set to 50. The PS and SA algorithms are compared with each other based on their parameters n_{max} and B_1. In Fig. 3.6(b), the order of complexity of the algorithms are compared in terms of the number of required flops when K_c is increased for $B_1 = B_2 = 50$, or $P = 26$. In this figure, we assume that $U_c = 7$, i.e. the maximum number of users that can be served simultaneously. The difference between the order of the complexity between exhaustive search and our proposed algorithms grows rapidly when the number of users is increased.

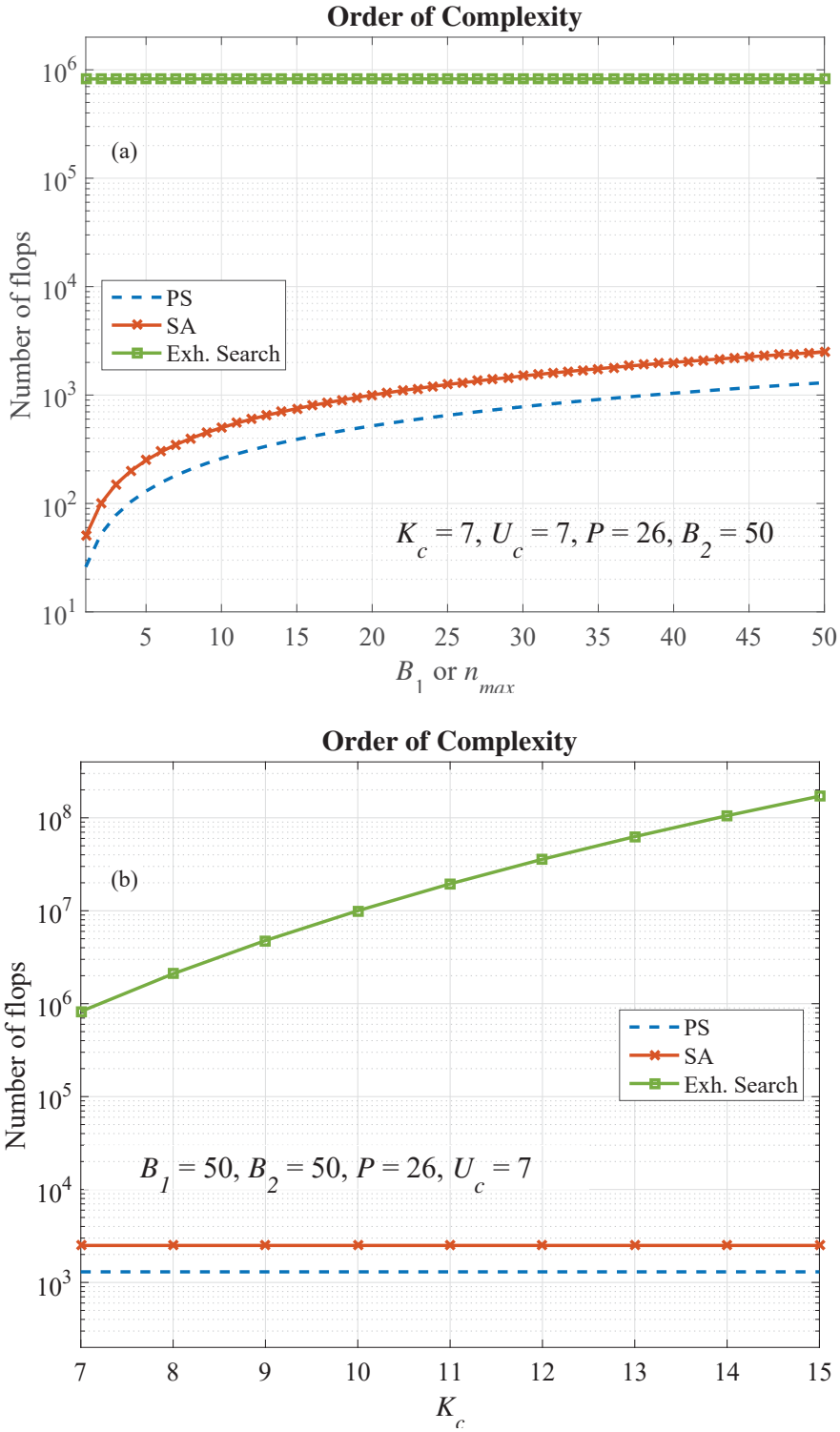

Figure 3.6: Order of complexity comparison for exhaustive search, SA and PS algorithms for order-dependent precoding methods; (a) with various B_1 or n_{max} when $K_c = 7$, (b) with different K_c when $B_1 = 50$ and $n_{max} = 50$; for $U_c = 7$, $B_2 = 50$, $P = 26$.

Each user scheduling algorithm requires the same information to make its decisions for every scheduling interval, i.e. the (whitened) channel matrices for the K_c users from the B_c base stations in cluster c. This requires $K_c M B_c N$ complex values to be sent on the backhaul. Once the scheduling decision is made, the information on that decision (again, of the same form regardless of the algorithm used) is communicated to the BSs in the cluster over the backhaul. This information includes, in part, an ordered vector of U_c user indices. For the larger and more significant portion of the information sent on the backhaul, in theory, the BSs could just be informed of the overall transmitted signal vector $\boldsymbol{\chi}_c \in \mathcal{C}^{B_c N \times 1}$, where $\boldsymbol{\chi}_c = \sum_{k=1}^{U_c} \mathbf{W}_{c,k} \mathbf{s}_{c,k}$, using $B_c N$ complex values. However, in practice, the individual precoding matrices $\mathbf{W}_{c,k}$ and data vectors $\mathbf{s}_{c,k}$ could be sent for every scheduled user k. If U_c users are each assigned M data streams, then a total of $U_c M(B_c N+1)$ complex values are sent over the backhaul for cluster c in this latter case. Either way, each BS in the cluster only needs the portion of the distributed information corresponding to its own transmit antennas.

3.4 Simulation Evaluation and Results

In this section, simulation results of the proposed algorithms are presented and their performance compared with optimal scheduling using several precoding techniques. The average sum rate and average number of scheduled users are determined for the target area using the Monte Carlo simulation method. Channel matrices for the users are generated according to (3.1). The numbers of transmit antennas per BS and receive antennas per user are assumed to be, respectively, $N = M = 2$. It is assumed that the radius of each macrocell is $R_m = 1$ km (the radius of the picocells is 20% of that of the macrocells). We also assume this distance as the reference distance as described in (3.1). The transmitting power of each macro BS in each sector is 10 dBW, and Γ_0 from (3.1) is set to result

in an (interference-free) SNR of 14.45 dB at the reference distance of R_m from a macro BS along the antenna boresight (see Fig. 3.1). The transmitting power of each pico BS is assumed to be 1/40 of the macro BS power. The path loss exponent is assumed to be 4 for all BSs and the standard deviation of shadow fading is $\sigma_\rho = 6$ dB and 4 dB, respectively, for macro and pico BSs [78]. The total interference from the other cells outside the target area is approximated by a complex Gaussian random vector with zero mean and variance[13] $\sigma_I^2 = 81$.

Prior work [82] has indicated that controlling the SA cooling rate with a parameter value in the interval $0.85 \leq \phi \leq 0.99$ is acceptable. In this thesis, the value of ϕ is chosen in two different ways. First, we set ϕ as a constant of 0.99. Secondly, we allow ϕ to be a random variable uniformly distributed in the interval of $(0, 1)$. A new value is used each time the algorithm runs (i.e. for each scheduling interval). By choosing $B_1 = \{10, 25, 50\}$, we can control the duration of cooling. We set the inner loop length with the parameter $B_2 = \{10, 25, 50\}$. The transition probabilities for the Markov chain in Fig. 3.5 are all equal.

In the PS algorithm, for c_1 and c_2, based on experimental trials in many applications [67, 68], values of 2 for each are reported to be good. We therefore set $c_1 = c_2 = 2$ in this work. For w_s and w_e in (3.11), again based on experimental trials [67, 68], suitable values for w have a range starting around $w_s = 0.9$ and ending around $w_e = 0.4$; we use these same values. w is decreased over the maximum number of iterations n_{max} which is set to 100. Also, the population size P is set to 20% of K_c, rounded up. We have run several trial simulations with different $n_{stalled}$, and have observed that for $n_{stalled} = 20$, the results are sufficiently close to optimal when the algorithm indicates that it has stalled. Finally, V_{max} is set to K_c in the simulations. For HGPS, the same parameter

[13]To obtain σ_I^2, we assume that all BSs outside the target area transmit at full power. Assuming the worst location in the target area (the place that receives the weakest signal power and/or lowest SINR from all BSs in the area) is at its center, the interference at this point is measured for millions of channel realizations. The variance of these measurements is σ_I^2.

Table 3.2: Definition of Case Numbers for Different User Scheduling Algorithms and Their Parameter Settings

			$\phi = 0.99$			$\phi = rand(\sim \mathcal{U}(0,1))$		
			B_2			B_2		
			10	25	50	10	25	50
SA	B_1	10	#1	#2	#3	#10	#11	#12
		25	#4	#5	#6	#13	#14	#15
		50	#7	#8	#9	#16	#17	#18

	Reflective borders	Modulo borders		
		Type 1	Type 2	Type 3
PS	#19	#20	#21	#22
HGPS	#23	#24	#25	#26
Greedy	#27			

values as for PS are applied.

We have also simulated a greedy user scheduling algorithm based on the method introduced in [27, 28], in order to compare its results with our proposed scheduling algorithms. The correlation threshold that maximizes the sum rate for the greedy algorithm is determined as in [27] (see therein for more details) and via simulations for $K = 7$. It has been observed that the optimum correlation threshold is 0.6 (which provides the best performance compared to that obtained by exhaustive search).

In Fig. 3.7, we compare the difference between the achievable sum rates obtained by different user scheduling methods considering different parameters under various precoding methods with the corresponding exhaustive search. Likewise, Fig. 3.8 compares the number of scheduled users in the target area under the same conditions. Different scheduling algorithms and their parameter settings are denoted by case numbers as described in Table 3.2. For example, case #1 represents the SA algorithm with ϕ set to the fixed value of 0.99, and B_1 and B_2 both set to 10. Case #19 represents RPS, while case #25 represents HGPS with modulo borders of type 2. The goal of this comparison is to find the best

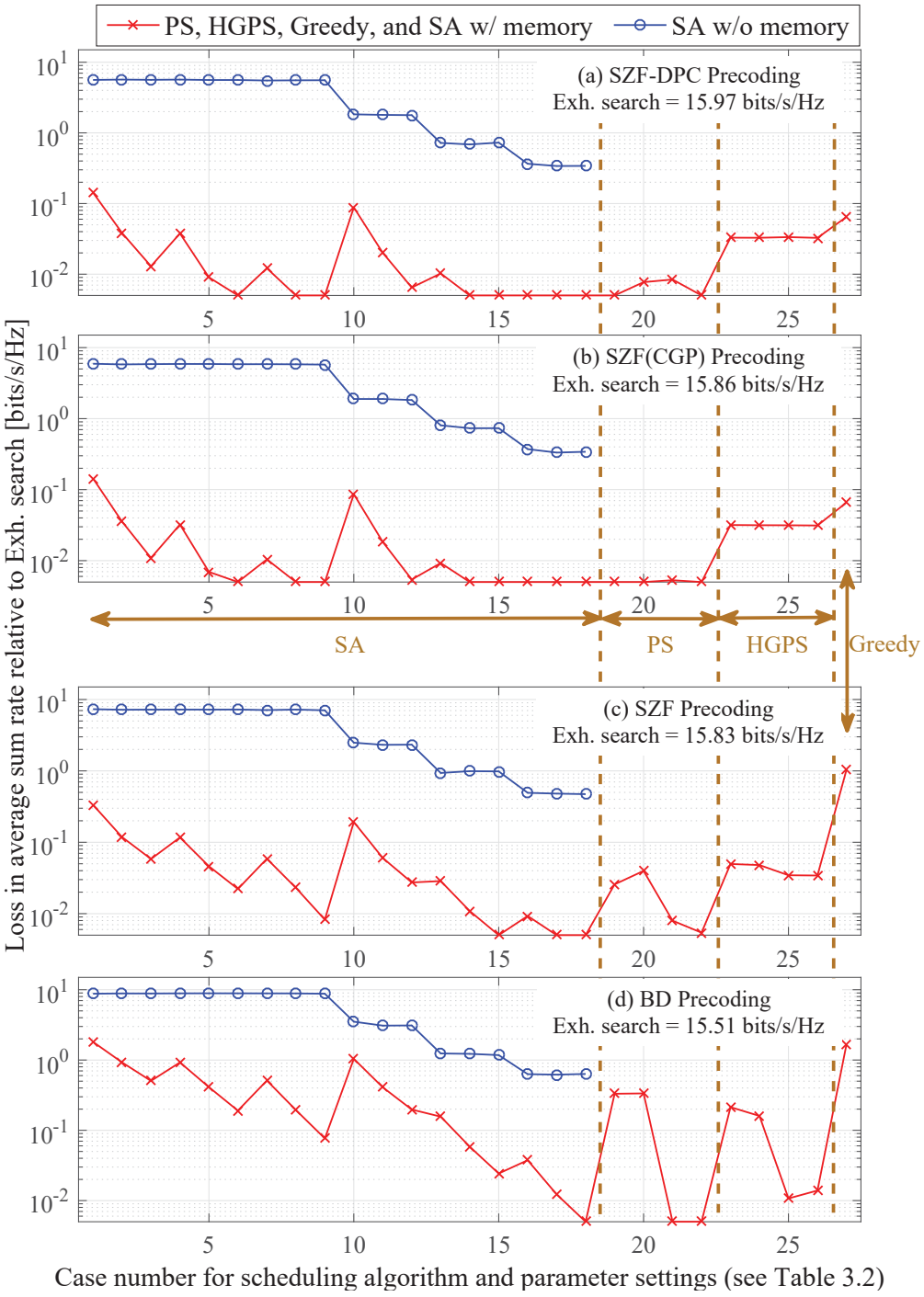

Figure 3.7: Comparison of loss in target area sum rate between different proposed PS, HGPS, and SA (with and without memory) user scheduling algorithms, and existing greedy algorithm relative to exhaustive search, using (a) SZF-DPC, (b) SZF(CGP), (c) SZF, and (d) BD precoding; $M = N = 2$, $K = 7$, $P_t = 10$ dBW.

user scheduling algorithm to obtain a solution close to optimum with reduced computational complexity.

As previously discussed, an exhaustive search will find the optimal ordered selection of users and the maximum sum rate. Naturally, the sum rate will depend on the precoding method used. In our system with $K = 7$ users, in Fig. 3.7, exhaustive search yields sum rates of 15.97, 15.86, 15.83, and 15.51 bits/s/Hz for SZF-DPC, SZF(CGP), SZF, and BD, respectively. (This ordering of precoding from highest to lowest throughput is also as expected.) It is interesting to note, though, that in Fig. 6, exhaustive search also generally schedules the fewest users when achieving those maximal rates.

We now discuss the SA algorithm. In Fig. 3.7(a), the differences in the sum rates of SA-ml and SA-m from those of exhaustive search are compared for SZF-DPC precoding in cases #1 to #18. The SA-m algorithm outperforms SA-ml by a considerable margin for most of the cases. As one example, case #9 for SA with memory achieves about 55% better performance (15.96 bits/s/Hz, near-identically to optimal) than the memoryless version SA-ml (10.29 bits/s/Hz, about 5.68 bits/s/Hz from optimal). In comparison, for the memoryless best-case #18, the improvement of SA-m over SA-ml is about 2.1% (15.96 bits/s/Hz vs. 15.63 bits/s/Hz, respectively). This demonstrates the benefit of the negligible extra complexity to keep track of the overall best solution the algorithm has observed during the search, even though it may have later left the vicinity of that solution. Considering the number of served users (Fig. 3.8(a)), SA-m and SA-ml serve almost the same number of users, and both serve more users than exhaustive search.

Using SZF-DPC precoding, the best result is obtained for the SA-m algorithm with $\phi = rand(\sim \mathcal{U}(0,1))$ and B_1 and B_2 both set to 50, i.e. case #18. However, it should be noted from Fig. 3.7, especially for SZF-DPC and SZF(CGP) precoding, several of the SA-m cases (as well as the PS cases) appear to have essentially

the same indistinguishable performance. This means that the difference between the results of these cases and the exhaustive search is smaller than what can reasonably be reported according to the precision available from the simulations (about 0.005 bits/s/Hz). That said, similar overall performance trends can be seen for the other precoding methods and with different values of K. For all precoding methods with any K, the case #18 has the best sum rate performance with both SA-ml and SA-m, resulting from that case having the most iterations.

Increasing either B_1 or B_2 while keeping the other constant results in a higher sum rate (or equivalently, a smaller loss in sum rate relative to optimal), but at the trade off of higher complexity. For an example, in Fig. 3.7(d) with $B_1 = 10$ and for SA-m, increasing B_2 improves the performance regardless of the type of ϕ that is chosen. Similar results can be seen for the reverse; assuming constant $B_2 = 25$, again an increase in B_1 improves the sum rate towards the optimal solution. For SA-m with fixed ϕ, increasing either B_1 or B_2 is equivalent; the result on the performance is nearly identical. That is, in all cases for ϕ fixed, setting $B_1 = x$, $B_2 = y$ yields the same sum rate as $B_1 = y$, $B_2 = x$. For example, comparing the cases between #2 and #4 or between #6 and #8, the achievable sum rates are almost identical for all precoding methods. With ϕ random, it appears that increasing B_1 is slightly better than increasing B_2. Hence, it seems that it is better in this instance to drop to a lower temperature than spending more time iterating at each given temperature. Comparison of the results for the cases #12 and #16 in Fig. 3.7(c) and (d) best illustrate this result; case #16 ($B_1 = 50$, $B_2 = 10$) consistently outperforms #12 ($B_1 = 10$, $B_2 = 50$).

Notably, the parameter ϕ affects the temperature changes. With fixed ϕ, every single decrease in temperature will have a fixed ratio, while random ϕ means that those ratios during the cooling are random each time the algorithm runs (i.e. sometimes (or indeed, usually) the algorithm can be cooled down faster). Since the SA-ml algorithm is memoryless, there is a higher possibility of locating and,

in particular, *staying* nearby a better solution by using a random ϕ than a fixed value of 0.99. Therefore, using a random ϕ achieves better sum rates for SA-ml than keeping ϕ fixed. Note also that the SA-ml performance is approximately equal with increasing B_2 at a fixed value of B_1. When ϕ is fixed at 0.99, even increasing B_1 has no positive effect on the performance. Larger values of B_2 increase the opportunity of SA-ml to find a better solution at each temperature, but also to move away from and, without memory, forget a previously better solution. (In comparison, with memory, SA-m strictly benefits from larger B_2.) Both events occur with decreasing probability as the temperature cools and the algorithm gets trapped at a local optimum. Larger B_2 does not help at this point, as the algorithm remains trapped in the vicinity of the local optimum, whose location is more determined by the more random movement at higher temperatures. Similar improved performance can be seen for SA-m with random ϕ over fixed ϕ, and for similar reasons. However, with the addition of memory, leaving a good solution is not nearly as detrimental, and so the effects of random ϕ are considerably smaller. Since the average value of the random ϕ is 0.5, we can say that the temperature is reduced on average by a ratio of 0.5. Thus, a smaller value than 0.99 for fixed ϕ may potentially result in better performance, especially for SA-ml, which would appear to highly benefit from much more rapid cooling.

As Fig. 3.8 presents, SA-ml serves about the same number or more users relative to SA-m with all precoding; SA-m notably serves fewer users than SA-ml with SZF and BD. Iterating longer in the inner loop of SA-m (with the temperature fixed) might result in a higher sum rate, but the number of scheduled users decreases under SZF and BD. The number is closer to constant with SZF-DPC and SZF(CGP). For SA-ml with random ϕ, the number of users decreases with increasing values of B_1, corresponding with the increase in sum rate; like the sum rate, the number of users only changes with B_1, but not with B_2, for similar

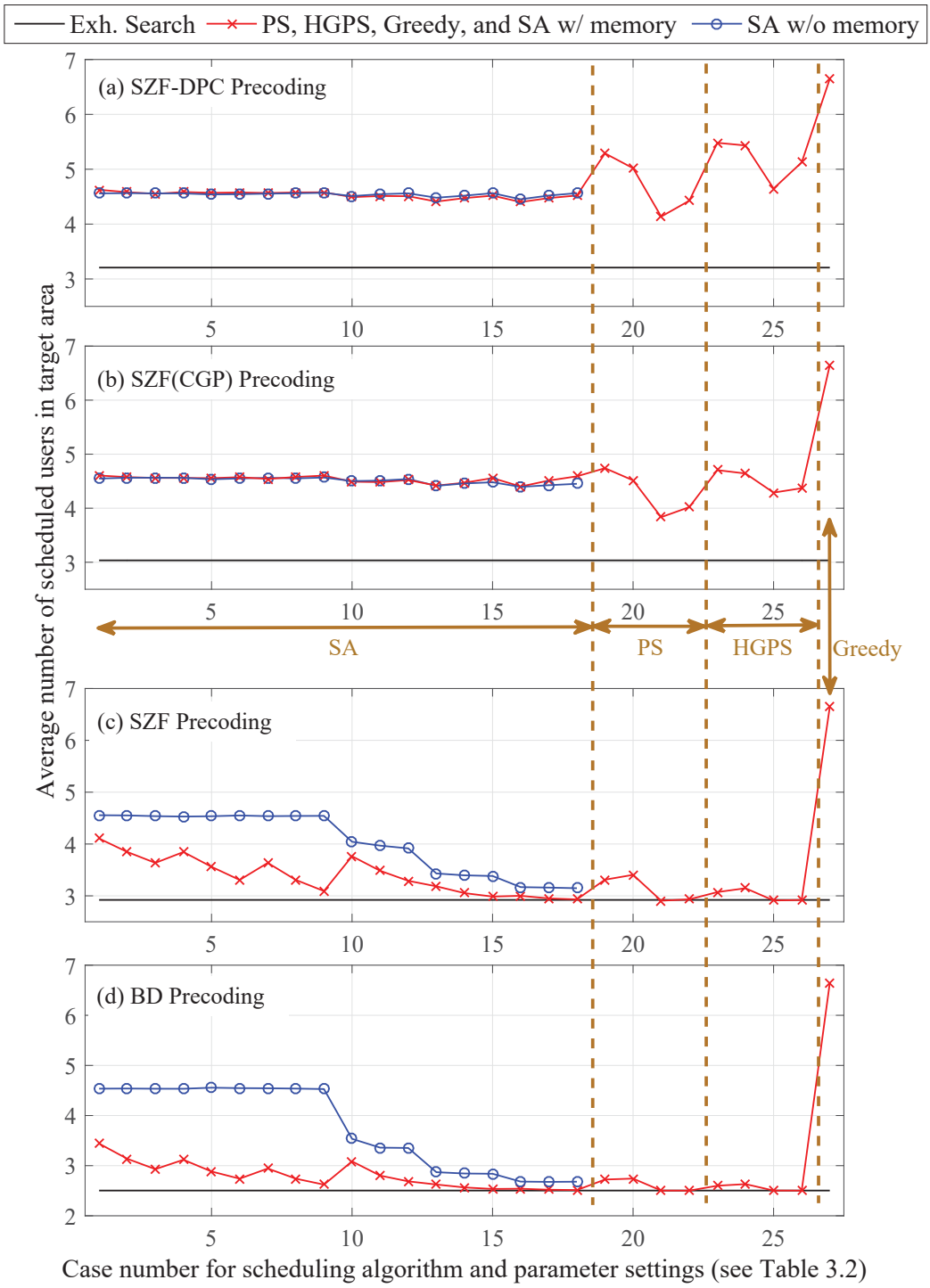

Figure 3.8: Comparison of target area average number of scheduled users between different proposed PS, HGPS, and SA (with and without memory) user scheduling algorithms, and existing greedy algorithm to exhaustive search, using (a) SZF-DPC, (b) SZF(CGP), (c) SZF, and (d) BD precoding; $M = N = 2$, $K = 7$, $P_t = 10$ dBW.

reasons for the constant performance as given in the previous paragraph.

Finally, the best SA method and parameters that can be advised to be used with all precoding methods is the SA with memory denoted by case #18 (i.e. $B_1 = B_2 = 50$), which achieves the highest sum rate of the examined SA cases. We use case #18 for the SA algorithm in the remainder of the chapter.

Similar examinations are presented in Figs. 3.7 and 3.8 to compare the border methods of PS and HGPS in case numbers 19 to 26. In Fig. 3.7(a), the sum rate of RPS, HGRPS, and the various MPS and HGMPS methods is compared for SZF-DPC precoding; a similar comparison is depicted in Fig. 3.7(b)-(d) for SZF(CGP), SZF and BD precoding, respectively. As Fig. 3.7(a) depicts, the PS and HGPS algorithms both perform very well. The RPS and MPS_3 methods (i.e. cases #19 and #22) provide sum rates the closest to an exhaustive search. While the different border methods affect the PS performance, there is no effect on the HGPS performance when the precoding method is SZF-DPC.

Considering Fig. 3.8, among all approaches, RPS and HGRPS generally serve more users than the rest. Since we are interested in serving a larger number of users as often as possible, while still keeping close to the optimal sum throughput, it seems that the RPS variant can satisfy this aspect the best under SZF-DPC precoding. The reflective border also does so without trading off the sum rate performance in return (unlike the SA-ml algorithm, for instance). Similar results are shown for SZF(CGP) in Fig. 3.8(b). For the remaining precoding methods, MPS_3 (case #22) provides higher sum rates for SZF and BD, respectively, in Figs. 3.7(c) and (d), though MPS_2 (case #21) yields indistinguishable performance from MPS_3 for BD. However, the average number of scheduled users is smaller than with RPS and MPS_1 as shown in Figs. 3.8(c) and (d). Of the MPS methods, MPS_1 serves the largest number of users. This is not surprising; as discussed in Section 3.3.1, MPS_1 has a bias towards ensuring the user number K_c is scheduled, if the number of users scheduled is less than the maximum possible.

Hence, that additional user is selected more often, leading to the larger average number overall. Furthermore, as also discussed earlier, MPS_2 has a slight bias towards unscheduling a user; this manifests itself in MPS_2 generally scheduling the fewest users. MPS_3 represents an intermediate point between MPS_1 and MPS_2. Since the numbers under the RPS method are also relatively larger, this indicates the reflective border method also tends somewhat to avoid particle element positions near the "0" border, at which users are not scheduled.

While the HGPS performance is very good, it does not surpass PS in terms of sum rate maximization. The act of keeping the first scheduled user fixed in its position limits the search area; i.e. it is narrowed to a subset of all possible vectors of users in which the first element of all these vectors is the same. This constraint forces the algorithm to search among only a subset of all the choices; however, the global optimum might not belong to this subset. The different approaches for the behaviour of the borders of the search area for the most part appear to have little effect on the resulting sum rate of the scheduled user vector (specifically for the HGPS methods); the exception is BD, which shows larger changes using different type of borders. As is evident, selecting the user with the highest channel gain as the first in the encoding order is the most dominant factor in determining the sum rate. This is reasonable, since the first user in SZF-DPC and SZF usually achieves the highest rate; users encoded later experience stronger interference, and thus achieve lower rates, hence contributing less to the sum rate. It is only these latter users that may differ between the different border methods in the HGPS algorithm, leading to the small to insignificant difference in sum rate between the methods. Since BD has no encoding order to speak of, the user chosen "first" is less dominant, and there is consequently more variation in the HGPS sum rate between the methods.

The border method does, however, affect the number of users scheduled by HGPS in the target area more significantly, especially when the precoding method

is SZF-DPC or SZF(CGP). Using SZF-DPC precoding, HGRPS and HGMPS_2 schedule 5.5 and 4.7 users on average for the same sum rate, respectively (a decrease of 14.5%). A similar observation can be made for the other precoding methods. While the first selected user may dominate the sum rate, it does not have nearly as much an effect on the number of users scheduled and encoded after it. Thus, there are larger variations in the HGPS number of scheduled users between the border methods, similar to those displayed for the base PS algorithm. However, we note the difference in the sum rates between all the methods is very small and generally negligible. Thus, we can overall remark that RPS can achieve very good sum rate performance compared to the exhaustive search, while serving the largest number of scheduled users. RPS is used for the PS algorithm in the remainder of the chapter.

The greedy algorithm from [27,28] (case #27) provides sum rates on par with the worst cases (with the fewest iterations) of our SA-m algorithm, as seen in Fig. 3.7. However, Fig. 3.8 shows that the greedy algorithm also schedules the largest number of users out of any of the algorithms we have examined. If, for instance, the number of users scheduled is an important factor, then the greedy algorithm might be a more attractive option. The order of complexity of the greedy algorithm may also be different (potentially lower), depending on how the number of iterations in the SA and PS algorithms are defined (e.g. a fixed constant, proportional to K_c, etc.)

Fig. 3.9 depicts a comparison of the different user scheduling algorithms while changing K from 5 to 15 under various precoding methods in terms of average sum rate and average number of scheduled users in the target area. Since PS performs about the same or better than HGPS (as observed in Figs. 3.7 and 3.8), we just show PS and SA in Fig. 3.9. The results are compared with exhaustive search as well[14]. The maximum possible number of simultaneously served users

[14]As an exhaustive search is highly complex and time-consuming, we have only simulated it up to $K = 9$. Simulation times for larger K are prohibitively large.

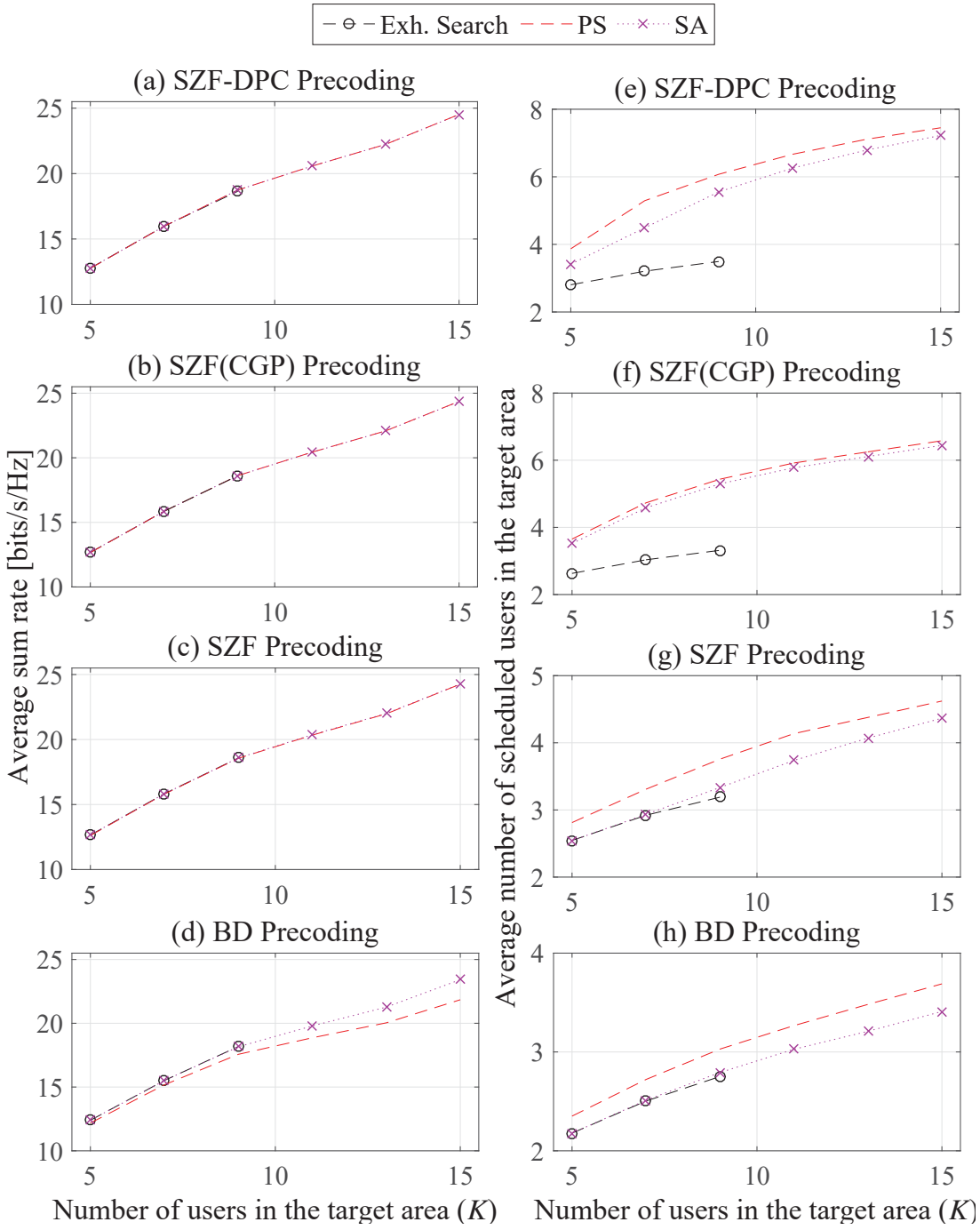

Figure 3.9: Comparison of performance vs. K in terms of (a)–(d) sum rate and (e)–(h) average number of scheduled users in target area between proposed SA and PS user scheduling algorithms, using SZF-DPC, SZF(CGP), SZF, and BD precoding methods; $N = M = 2$, $P_t = 10$ dBW.

is 9 according to our network design and parameters. Thus, for $K > 9$, there is a decline in how fast the sum rate grows with K, and for very large values of K, the number of scheduled users in the target area should asymptotically approach 9 due to multi-user diversity. In Fig. 3.9(a), SA and PS perform almost the same under SZF-DPC, providing sum rates nearly identical to what an exhaustive search can obtain. PS also schedules more users on average than SA for the entire examined range of K (see Fig. 3.9(e)). Serving more users simultaneously generally translates to more fairness to the pool of users requesting service, as it typically results in smaller overall delays in service. Thus, since the PS algorithm provides the same sum rate as SA, while being a bit fairer, the pairing of PS with SZF-DPC is recommended. The SZF and SZF(CGP) results behave similarly to those for SZF-DPC, thus the same recommendation of PS is made for those two precoding methods. As Fig. 3.9(d) depicts, SA consistently provides a higher sum rate than PS for BD. However, PS serves more users with this type of precoding. For example, in Fig. 3.9(h), at $K = 13$, the gain of PS over SA is 8.5% for BD precoding in terms of the number of scheduled users, while the performance gain is 6.2% for the sum rate of SA over PS (Fig. 3.9(d)). Since the aim of the algorithms is to maximize the system sum rate, the SA algorithm can in general be recommended for BD. However, in many circumstances, the system operator may be willing to trade off a bit a performance in favor of fairness to the users; hence, the PS algorithm can still be a very viable option.

We have thus far investigated the performance of our proposed algorithms at a fixed transmitting power of $P_t = 10$ dBW for the macro BSs. In Fig. 3.10 we consider K fixed at 9 and the SA and PS scheduling algorithms are evaluated at different transmitted powers. The pico BS transmitted power continues to be 1/40 of the macro power. Increasing P_t enlarges the variance of the interference[15] on the target area as well as the received SNR at the user's end. The average sum rate is

[15]We have set $P_t = \{-5, 0, 5, 10, 15, 20\}$ dBW, which results in corresponding variances for the surrounding interference of $\sigma_I^2 = \{2.6, 8, 25.7, 81, 256, 784\}$, respectively.

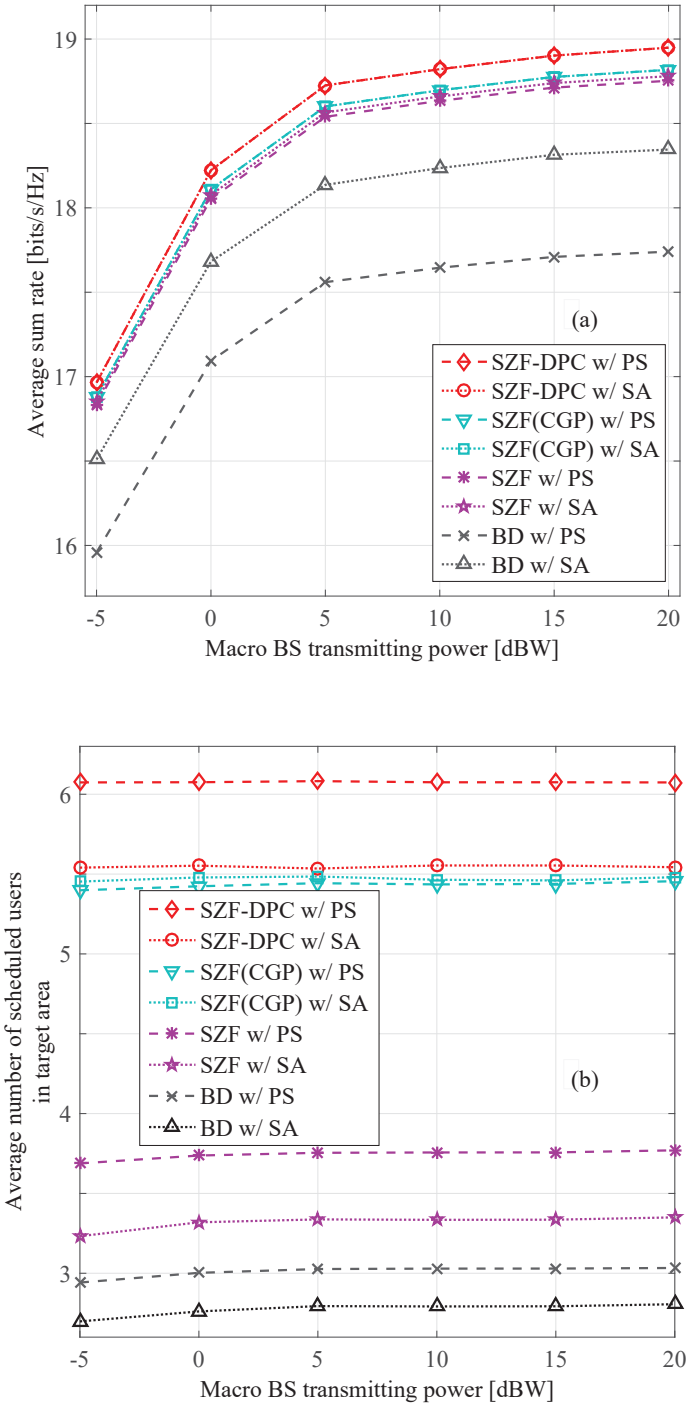

Figure 3.10: Comparison of performance in terms of (a) sum rate and (b) average number of scheduled users in target area between different proposed SA and PS user scheduling algorithms vs. macro BS transmitting power, using SZF-DPC, SZF(CGP), SZF, and BD precoding methods; $N = M = 2$, $K = 9$.

increased with larger SNRs, but with diminishing gains. Eventually, more power does not improve the sum rate that much, since the system is interference-limited and hence the SINR stays approximately constant. In Fig. 3.10(a), for SZF-DPC with PS user scheduling, the sum rate improvement from $P_t = -5$ dBW to $P_t = 5$ dBW is about 10.4%, while this improvement from $P_t = 5$ dBW to $P_t = 15$ dBW is about 0.9%. For SZF-DPC and SZF(CGP), PS and SA perform the same for the entire range of P_t, however, SA outperforms PS when the precoding method is either SZF or BD (the difference is far more significant for BD than for SZF). Moreover, the number of scheduled users in the target area is not affected by increasing the transmitted power; i.e. the number is approximately constant for the whole range of powers (see Fig. 3.10(b)). Increasing the transmitted power also proportionally increases the interference experienced by users in the cluster. Again, since the system is interference-limited, this means that the overall situation experienced by the users changes very little. Those users that are in a better situation to be selected by the scheduler (e.g. closer to a BS) are still in relatively the same situation to each other. Thus, with an increase in P_t, although there is an increase in the resulting rates per user and thus sum rates, as seen in Fig. 3.10(a), the selections of and numbers of users chosen by the scheduler remain almost the same.

3.5 Conclusion

In this chapter, the downlink of a coordinated heterogeneous MIMO network including macro and pico BSs has been considered. We have proposed various simplified user scheduling algorithms for this system, using reduced-complexity simulated annealing, particle swarm and hybrid greedy-particle swarm algorithms. The proposed algorithms are flexible to the number and type of transmission nodes in the network. Each scheduling algorithm has been evaluated for the

different parameters and options that can affect its performance, with the best choices determined by simulations. The performance of the proposed algorithms has been evaluated under various precoding methods, namely SZF-DPC, SZF, SZF(CGP), and BD precoding. Moreover, the complexity of the proposed algorithms has been calculated and compared. SA algorithms with and without memory were considered. According to the simulation results, SA with memory outperforms the memoryless version in terms of average sum rate. Of the two PS border methods examined, reflective PS (where the particles "bounce off" the search area boundaries if they were to cross them) performs marginally better than modulo borders for all precoding techniques, except for BD precoding. The HGPS algorithm performs better than the examined greedy algorithm and very close to PS, but the PS algorithm still achieves better results considering the sum rate maximization criterion. All of the proposed algorithms perform quite close to an optimal exhaustive search in terms of the sum rate they provide, but at much lower complexity. The simulations demonstrate that PS can serve more users, while performing close to an exhaustive search for almost all of the examined types of precoding. Otherwise, overall the difference in performance between PS and the other two proposed methods is mostly negligible. The results have been tested for different ranges of the number of users and BS transmitted powers. As the number of users in the target area increases, the achievable sum rate grows. Increasing the transmitted power at the macro BSs can improve the sum rate somewhat, but has no notable effect on the number of scheduled users.

The sum rate maximization criterion used in our scheduling algorithms is well known to not account for any aspects of user fairness as part of the scheduling decisions. However, such issues of user fairness are quite important in a practical network. Factors like fairness and delay of service can be handled by incorporating them into the scheduling metric. A proportionally fair scheduler [73–75] is one such example. In our proposed scheme, this would simply require a corre-

sponding change to the utility function calculated by the scheduling algorithms. The remainder of the operation would essentially be unchanged.

Chapter 4

Rotating Cluster Mechanism

4.1 Introduction

Clusters of the HetNet BSs coordinate user scheduling, precoding, and transmissions from their constituent antennas as a single virtual array, reducing the interference for users within each cluster. Users located near the border of each cluster suffer more inter-cluster interference compared to those near its center. Although using a PF metric [73–75] instead of MT will improve the throughput and fairness to these cluster-edge users, dynamic clustering, in which differing clusters can be formed and change over time [37–40], will improve their performance even more. For example, [83] introduced a type of dynamic clustering algorithm based on weighted sum rate maximization for the downlink of a MU-MIMO CoMP system. The clusters were formed specifically for the set of scheduled users such that they experience minimum ICI. It was shown that the algorithm improved the system sum rate in comparison to that of static clustering methods. Although dynamic clustering improves the performance of the cellular system, it is a complex method with high signalling overhead on the processor and backhaul.

Representing a middle ground between fixed and dynamic clustering, rotating clustering is a simplified method of dynamic clustering, and can be employed

to ameliorate the cluster-edge effects. In this method, the processor has several predefined patterns of clustering, which are interchanged periodically. Thus, users will periodically be near either the edge or the center of a cluster. In fact, rotating clustering can strike a balance between performance and the complexity of the cluster forming approach.

4.1.1 Motivation and Contributions

Different cellular layouts require different rotating patterns based on the BSs' physical arrangement to achieve better performance; the design of these patterns and investigation of their performance for the downlink of MIMO CoMP HetNets is the objective of this chapter. Considering cellular layouts with 3 or 6 macrocells per site, different rotating patterns of clusters are proposed. Assuming a 6-cell layout, which creates a hexagonal-shaped cooperating area, we propose a set of 2 rotating cluster patterns. Similarly, for a 3-cell layout, which forms a clover-leaf-shaped cooperating area, we propose a set of 5 rotating cluster patterns. The performance of the systems with the proposed sets of clustering patterns is investigated using a simulated annealing algorithm as the user scheduler and successive zero-forcing dirty paper coding as the precoding method; the systems' performance is compared with that of static clusters.

The rotating clustering scheme is primarily designed to improve the throughput of users with poor channel gains, which are usually located at the cluster edge. We analyse the effect of cluster rotation on those users. In this chapter, the effectiveness of the proposed methods with two different scheduling metrics, MT and PF, is investigated. We also analyse the capability of this scheme to improve the average achievable sum rate in a cluster, which can be considered as an extra benefit of this approach, if it is possible. Moreover, we investigate the effect of the speed of rotation on the performance of the system to determine the maximum speed that clusters may rotate, while still improving the system's

throughput non-negligibly compared to the lower speeds. The proposed cluster rotation schemes are less complex than a fully dynamic clustering method and hence they may be more easily implemented in a practical system. We also discuss this matter in this chapter.

We wish to emphasize, though, that the focus of this work is on the cluster rotation and its performance, not on the scheduling algorithm, nor on the precoding method. The cluster rotation methods employed in this work can be generalized and applied both to different user scheduling algorithms and different precoding methods, including (but not limited to) those discussed in Chapter 3, and the outcomes of this chapter are expected to be independent from the user scheduling and/or precoding approaches. Please recall that the cluster rotation scheme addresses the impact of *inter*-cluster interference, while precoding and user scheduling attempt to resolve the *intra*-cluster interference.

4.2 System Model, Design, and Achievable Weighted Sum Rate

We consider the downlink of a coordinated multicell MU-MIMO HetNet. Several macro BSs are co-located at each macro site, which is partitioned into different cells each covered by a directional antenna installed on a macro BS. We assume two different network layouts, the first with 6 macrocells per site, and the second with 3 macrocells per site. (For shorthand, we refer to these respectively as "6-cell" and "3-cell" layouts in this chapter.) Different system model characteristics are assumed, which are described in Sections 4.2.1 and 4.2.2, respectively, for the 6-cell and 3-cell layouts. In both system models, omnidirectional pico BSs surround each macro site and overlay the macro coverage area. The macro BSs transmit with power P_t and the inter-site distance (ISD) between macro sites is fixed and denoted by D. Each macro BS is equipped with N_{macro} transmit

antennas, while each pico BS has N_{pico} antennas. We assume for simplicity that N_{macro} and N_{pico} are both equal to N. Since all macro and pico BSs in a cluster are jointly transmitting their signals, this assumption is a valid assumption in our analysis and would not impact the proposed methods or the analysis in this chapter. Having N_{macro} be larger than N_{pico} would of course increase the system sum rates and number of users that can be scheduled simultaneously, but also thereby significantly (and likely needlessly) increase the complexity and length of time for simulations.

4.2.1 Layout 1: 6-Cell Layout (Hexagonal-Shaped Cooperating Area)

In our first HetNet model, the coverage area of the 6 cells per site overall forms a hexagonal-shaped region, with each macro BS covering a 60° angle of the area with a directional antenna. The macro site is surrounded by 12 low-powered pico BSs that form picocells overlaying the macro coverage area (see Fig. 4.1). Of the 6 cells per site, two adjacent ones are coordinated at any given time to form an effectively larger cell area. The picocells also coordinate within whatever cluster that the macrocell they overlay is part of. The location of pico BSs and their borders are determined according to the effective coverage area analysis by pico BSs compared to the macro ones; details of this analyse is described in Appendix A. Without loss of generality, we may consider any arbitrary macro site (with coverage area shown in green) and the clusters it participates in (shown by the red dashed lines). Therefore, the BSs of any macro site contribute to three different clusters.

As depicted in Fig. 4.1, two different patterns of clustering are possible, in which different adjacent cells cooperate with each other. All cells within each thick red dashed hexagon coordinate signals from their BSs to form a cluster; one example cluster in each pattern is emphasized in the figure for clarity. As

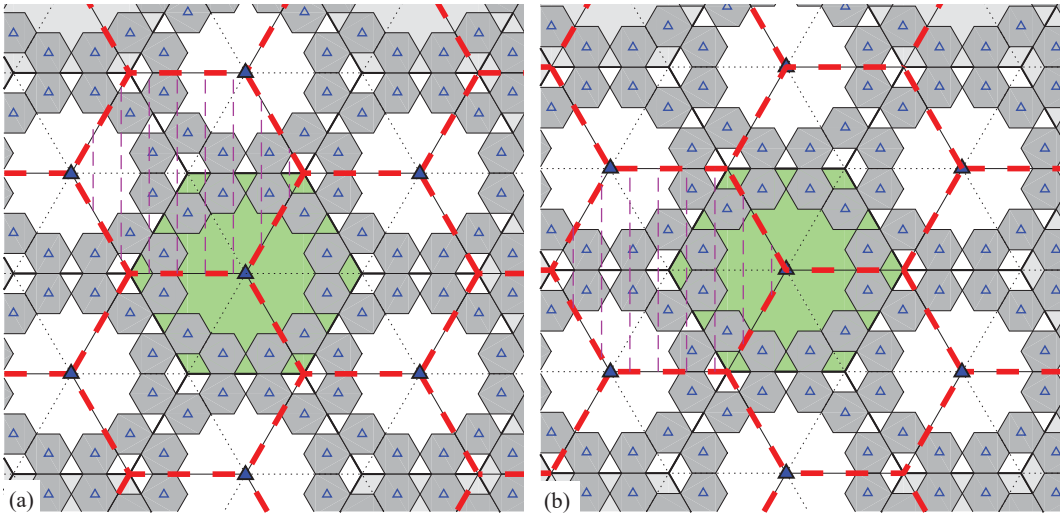

Figure 4.1: Network layout with 6 cells per macro site for HetNet with cluster rotation; (a) and (b) depict two alternating clustering patterns of BS coordination; solid and open triangles represent, respectively, macro BS sites and pico BSs, and the thick red dashed hexagons denote clusters.

Fig. 4.1(a) depicts, those users in a cluster that are located near the border of the cluster experience the poorest channel conditions from the BSs in the cooperating set. By rotating the clustering pattern by 60° around any macro site (see Fig. 4.1(b)), those previously poor-coverage users are now in the middle of the cluster (i.e. they will have better channel gains or higher achievable rates). Therefore, most users will have the opportunity to have a higher chance of being scheduled and to achieve reasonably good data rates for a fraction of the overall transmission time. Averaging the throughput over all transmission periods and clustering patterns, the overall achievable transmission rate of the users will be improved.

There are K users uniformly distributed over the coverage area of each macro site, each user equipped with M receive antennas. $K_{c(i)}$ is the number of users assigned to cluster $c(i)$, from which $U_{c(i)}$ users are served, where i refers to the ith pattern of clustering. Each cluster transmits coordinated data signals from all its BSs to its scheduled users.

4.2.2 Layout 2: 3-Cell Layout (Clover-Leaf-Shaped Cooperating Area)

For the second HetNet model, which is more commonly used in LTE-Advanced design [43] and is called a clover-leaf model, each cell in a macro site is covered by a high-powered BS, which is located at a corner of the cell. The directional antenna at a macro BS covers a hexagonal-shaped cell within the angle of 120°. Each macrocell is overlaid by 4 low-powered omnidirectional pico BSs, which are located near the 4 edges of the macrocell that are the most distant from the macro BS, as depicted in Fig. 4.2 and also described in Appendix A. Any three adjacent macrocells and their constituent picocells may form a cluster, if the macrocells share a corner that is not a site. Therefore, considering an arbitrary macro site and its corresponding 3 macrocells (shown in green in Fig. 4.2), the macro BSs may belong to two or three independent clusters (shown by the red dashed lines).

As depicted in Fig. 4.2, five different patterns of clustering are possible. We again highlight one example cluster in each pattern for clarity. Those users that are located near the edge of the cluster experience poor channel conditions from the BSs in cooperating set, and consequently their achievable rates will be smaller compared to the users in the middle of the cluster. By rotating the clustering pattern (see Fig. 4.2(b)), a portion of those previously poor-coverage users are now in the middle of the cluster, and some of the users, previously located at the middle of cluster, are now near the edge of the cluster. To put the remainder of the cluster-edge users in Fig. 4.2(a) near the middle of a cluster, more rotations are required, which are depicted in Fig. 4.2(c)-4.2(e). Therefore, in this layout after 5 intervals of rotation, all users have the opportunity of being at least once in the middle of the cooperating area, and thus having a higher chance of being scheduled and achieving reasonably good data rates.

There are K users, each equipped with M receive antennas, uniformly dis-

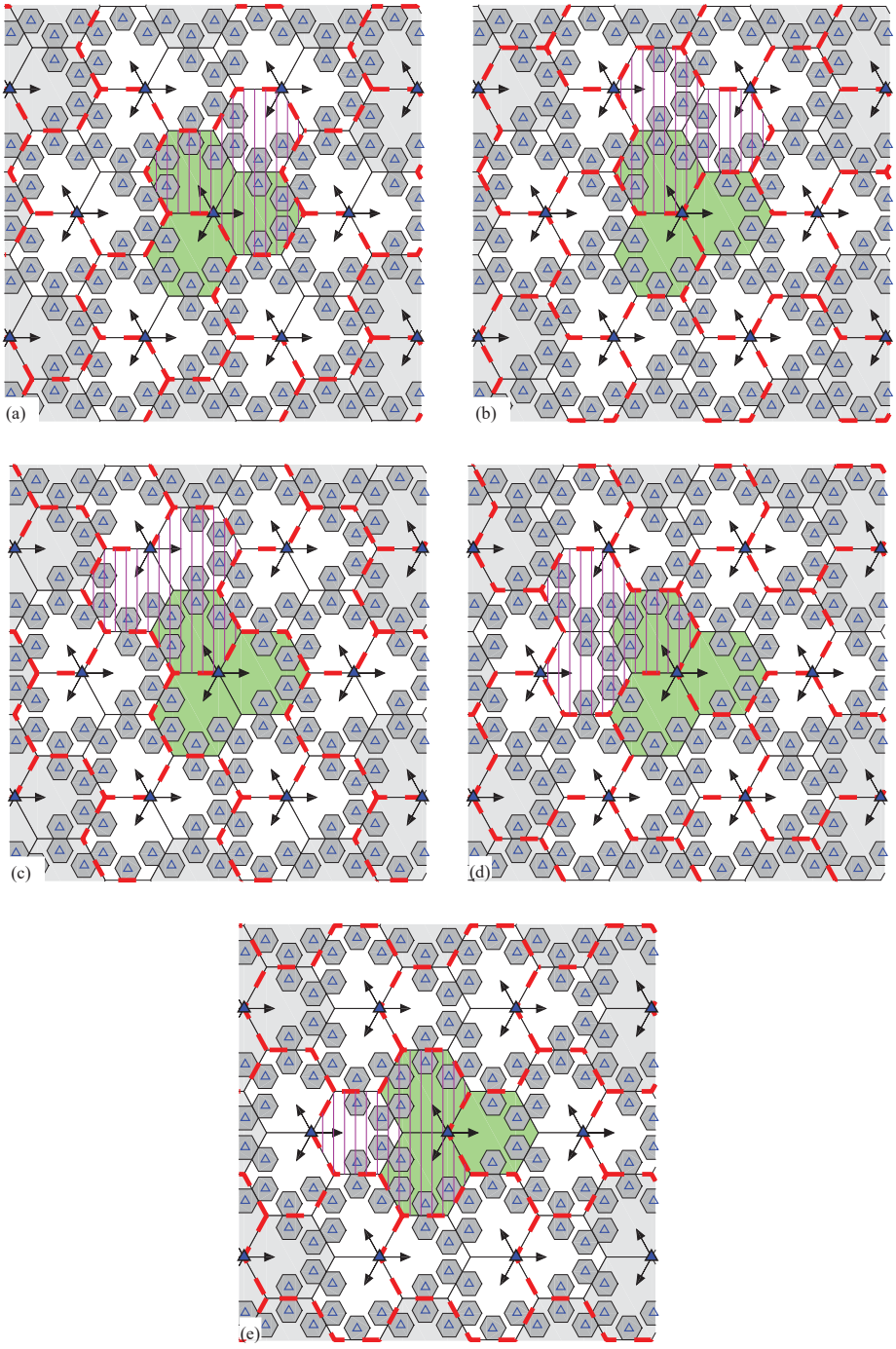

Figure 4.2: Network layout with 3 cells per macro site for HetNet with cluster rotation; (a)-(e) depict five different clustering patterns of BS coordination; solid and open triangles represent, respectively, macro BS sites and pico BSs, and thick red dashed lines denote cluster borders.

tributed over each cell. This is in contrast to the 6-cell layout, which has K users distributed over the coverage area of the macro site. Thus, for the 3-cell layout, there are $3K$ users per site; i.e. $K_{c(i)} = 3K$.

For both layouts, other patterns of clustering could in theory be used, e.g. by coordinating more macrocells together in a cluster. However, please consider the corners of each macrocell that do not contain a macro BS site. These locations have the worst SINR when no coordination occurs. The patterns that we use cluster the smallest possible number of macro BSs such that it allows each of those corners to be in the center of the cluster in one of the patterns. At the same time, the duration between any given corner being in the center (as the scheme rotates through the patterns) is also the smallest possible for the number of macrocells per cluster being used. (Note there are 2 of these corners per macrocell in the 6-cell layout, and 5 such corners in the 3-cell layout, hence leading to 2 and 5 patterns respectively for the layouts.)

4.2.3 Achievable Weighted Sum Rate and User Scheduling

For both layouts, averaging the throughput over all transmission periods and clustering patterns will improve the overall achievable transmission rate of the users. Defining T_{cl} as a specific clustering pattern duration in units of scheduling intervals, rotation to the next pattern will occur every T_{cl} scheduling intervals. Denoting $B_{c(i)}$ as the number of BSs in the $c(i)$th cluster of the ith pattern, the aggregate downlink channel of the kth user from all these $B_{c(i)}$ BSs is defined by $\mathbf{H}_{c(i),k} = [\mathbf{H}_{c(i),k}(1), \cdots, \mathbf{H}_{c(i),k}(B_{c(i)})]$, where $\mathbf{H}_{c(i),k}(b) \in \mathcal{C}^{M \times N}$ denotes the downlink channel matrix between the kth user and bth BS of the cluster. Each element of $\mathbf{H}_{c(i),k}(b)$, denoted by $h_{c(i),k}(b, m, n)$, is the complex downlink channel signal strength coefficient between the mth receiving antenna of the kth user and the nth transmitting antenna of the bth BS in the $c(i)$th cluster, which includes

path loss, log-normal shadowing, and Rayleigh fading, and is modeled by

$$h_{c(i),k}(b,m,n) = z_{c(i),k}(b,m,n) \times \sqrt{\Gamma_0 P_t(b) \Big(\frac{R_m}{d_{c(i),k}(b)} \Big)^{\alpha(b)} \rho_{c(i),k}(b) A(\theta, b)}. \quad (4.1)$$

$z_{c(i),k}(b,m,n)$ represents small-scale frequency-flat Rayleigh fading with an i.i.d. complex Gaussian random variable distributed as $\mathcal{CN}(0,1)$. R_m is the reference distance[1] and Γ_0 is a scaling factor controlling the reference SNR at a distance of R_m in the direction of the boresight of the directional antenna. The distance between user k and BS b in cluster $c(i)$ is represented by $d_{c(i),k}(b)$, and $\alpha(b)$ is the path loss exponent for BS b. $P_t(b)$ is the transmit power of BS b, and $\rho_{c,k}(b)$ denotes the log-normal shadow fading coefficient with standard deviation σ_ρ. The antenna pattern $A(\theta, b)$ of a macro BS, where θ is the angle between the direction of interest and the boresight of the antenna at BS b, and is defined as described in [43,78], while $A(\theta, b)$ is equal to unity for pico BSs with omnidirectional antennas.

All $B_{c(i)}$ BSs of cluster $c(i)$ cooperatively transmit the data vector $\mathbf{s}_{c(i),k} \in \mathcal{C}^{M\times 1}$ for user k using the aggregate precoding matrix $\mathbf{W}_{c(i),k} \in \mathcal{C}^{B_{c(i)}N\times M}$. The received signal $\mathbf{y}_k \in \mathcal{C}^{M\times 1}$ for user k is given by

$$\mathbf{y}_k = \mathbf{H}_{c(i),k} \sum_{j=1}^{U_{c(i)}} \mathbf{W}_{c(i),j} \mathbf{s}_{c(i),j} + \underbrace{\sum_{\check{c}(i)\neq c(i)} \mathbf{H}_{\check{c}(i),k} \sum_{\forall j} \mathbf{W}_{\check{c}(i),j} \mathbf{s}_{\check{c}(i),j} + \mathbf{n}_k}_{\mathbf{z}_{c(i),k}}. \quad (4.2)$$

The first term in (4.2) is the received signal from cluster $c(i)$, to which the user belongs, while the second term describes the interference from other clusters. Applying the central limit theorem, the total interference signal from all clusters not including $c(i)$, denoted by $\check{c}(i) \neq c(i)$, is approximated by an $M \times 1$ complex

[1] R_m is defined as the farthest distance from the macro BS to the edge of its cell in the direction of its boresight. Since we fix the value of the ISD for both layouts in our simulations, R_m in the 6-cell layout is $D/2$ while it is $2D/3$ for the 3-cell layout.

Gaussian random vector with zero mean and standard deviation σ_I. To estimate the standard deviation of this interference, it is assumed that all BSs outside the cluster $c(i)$ are transmitting with full power. The interference from these BSs experienced at different locations within the cluster $c(i)$ is determined and averaged via Monte Carlo simulation over many channel realizations. The standard deviation of these realizations is used as the value of σ_I. The last term $\mathbf{n}_k \in \mathcal{C}^{M\times 1}$ is a complex additive white Gaussian noise vector with each element having zero mean and unity variance. The summation of interference and noise is denoted by $\mathbf{Z}_{c(i),k}$, which with the Gaussian interference approximation ends up as a complex Gaussian random vector with zero mean and variance $\sigma_I^2 + 1$. For convenience of calculation, the interference-plus-noise power is normalized at the receiver. This is equivalent to applying a filter at the receiver of $\mathbf{Q}_r = (\sigma_I^2 + 1)^{-1/2}\,\mathbf{I}_M$. Hence, by defining $\tilde{\mathbf{H}}_{c(i),k} = \mathbf{Q}_r\mathbf{H}_{c(i),k}$ as the post-processed equivalent channel matrix and $\tilde{\mathbf{Z}}_{c(i),k} = \mathbf{Q}_r\mathbf{Z}_{c(i),k}$ as the normalized interference plus noise, (4.2) is revised as

$$\tilde{\mathbf{y}}_k = \tilde{\mathbf{H}}_{c(i),k}\sum_{j=1}^{U_{c(i)}}\mathbf{W}_{c(i),j}\mathbf{s}_{c(i),j} + \tilde{\mathbf{Z}}_{c(i),k}. \tag{4.3}$$

We choose to use the SZF-DPC precoding technique, where the encoding order of the users is very important for maximization of the achievable weighted sum rate. Given a set of users with order $\pi^j_{c(i)}$ and defining the user encoded at position k as $\pi^j_{c(i),k}$, the post-processed received signal can be modified and expanded as

$$
\begin{aligned}
\tilde{\mathbf{y}}_{\pi^j_{c(i),k}} &= \tilde{\mathbf{H}}_{c(i),\pi^j_{c(i),k}} \mathbf{W}_{c(i),\pi^j_{c(i),k}} \mathbf{s}_{c(i),\pi^j_{c(i),k}} \\
&+ \tilde{\mathbf{H}}_{c(i),\pi^j_{c(i),k}} \sum_{l<k} \mathbf{W}_{c(i),\pi^j_{c(i),l}} \mathbf{s}_{c(i),\pi^j_{c(i),l}} \\
&+ \tilde{\mathbf{H}}_{c(i),\pi^j_{c(i),k}} \sum_{l>k} \mathbf{W}_{c(i),\pi^j_{c(i),l}} \mathbf{s}_{c(i),\pi^j_{c(i),l}} \\
&+ \tilde{\mathbf{Z}}_{c(i),\pi^j_{c(i),k}}.
\end{aligned} \tag{4.4}
$$

The two summations in the second and third line of (4.4) represent the intra-cluster interference for user k. In SZF-DPC, the precoding matrix $\mathbf{W}_{c(i),\pi^j_k}$ is constrained to lie in the null space of the channel matrices of all users encoded before $\pi^j_{c(i),k}$; the aggregate channel matrix of previously encoded users is defined as $\mathbf{H}_{k-1} = \left[\tilde{\mathbf{H}}^T_{c(i),\pi^j_{c(i),1}}, ..., \tilde{\mathbf{H}}^T_{c(i),\pi^j_{c(i),k-1}}\right]^T$. The precoding matrix cancels the intra-cell interference from the summation in the third line of (4.4), while the effect of the remaining intra-cell interference represented by the summation in the second line of (4.4) is removed by using DPC. Using singular value decomposition of $\mathbf{H}_{k-1}$, for a given ordered user $\pi^j_{c(i),k}$, its achievable rate $R_{c(i),\pi^j_{c(i),k}}$ is given by

$$
R_{c(i),\pi^j_{c(i),k}} = \log_2 \left| \mathbf{I}_M + (\tilde{\mathbf{H}}_{c(i),\pi^j_{c(i),k}} \mathbf{V}^0_{k-1}) \mathbf{Q}_{c(i),\pi^j_{c(i),k}} (\tilde{\mathbf{H}}_{c(i),\pi^j_{c(i),k}} \mathbf{V}^0_{k-1})^H \right|, \tag{4.5}
$$

where $\mathbf{Q}_{c(i),\pi^j_{c(i),k}}$ is the transmit covariance matrix for user $\pi^j_{c(i),k}$ in cluster $c(i)$ and $\mathbf{V}^0_{k-1}$ are orthonormal basis vectors for the joint null space of $\mathbf{H}_{k-1}$ for the users before $\pi^j_{c(i),k}$ in the encoding order; $\mathbf{V}^0_0 \triangleq \mathbf{I}_{B_{c(i)}N}$.

The throughput maximization criterion results in the selection of a scheduled vector of users that achieves the largest sum rate among all possible vectors of users. Those users who have better channel gains have a higher likelihood to be selected by a MT scheduler. Thus, users with poorer channel gains may be very infrequently (and potentially never) selected by the scheduler, which is not fair. In PF scheduling, each user has a weight related to its priority for being chosen

by the scheduler, and the scheduler adjusts each weight based on the average achievable rates in the user's history. A PF scheduler chooses those users whose instantaneous rates relative to their average rates are better than the others, and uses a weighted sum rate as its scheduling metric; i.e. the combination of those users with maximum weighted sum rate will be chosen to be scheduled. If a user has been selected by the scheduler often, its weight for the next interval will be decreased (as its average rate increases), i.e. its chance to be chosen in the next scheduling interval degrades. Meanwhile, another user with a worse channel matrix may have more opportunity to be scheduled in the next interval simply by having higher weight. Using this method provides more fairness in the network among all users.

In each cluster, the achievable weighted sum rate $WSR_{c(i)}$ is given by[2]

$$WSR_{c(i)} = \max_{\pi^j_{c(i)}: j\in\{1,2,\cdots,U_{c(i)}!\}} \quad \max_{\substack{\left\{\mathbf{Q}_{c(i),\pi^j_{c(i),k}}\right\}_{k\in\{1,\cdots,U_{c(i)}\}}: \\ \mathbf{Q}_{c(i),\pi^j_{c(i),k}}\succeq\mathbf{0},\ \sum_{\forall k} Tr(\mathbf{Q}_{c(i),\pi^j_{c(i),k}})\le 1}} \sum_{k=1}^{U_{c(i)}} \mu_{c(i),\pi^j_{c(i),k}}(t) R_{c(i),\pi^j_{c(i),k}}(t) \tag{4.6}$$

where $\mu_{c(i),\pi^j_{c(i),k}}(t)$ is the priority weight of the kth user during the tth scheduling interval in cluster $c(i)$. In PF scheduling, for the lth user out of $K_{c(i)}$, $\mu_{c(i),l}(t) = 1/\bar{R}_{c(i),l}(t)$, where $\bar{R}_{c(i),l}(t)$ is the average achievable data rate of the lth user at time t, averaged over a window of the past t_c intervals. In each time interval, $\bar{R}_{c(i),l}(t)$ (and thus $\mu_{c(i),l}(t)$) is updated by an exponential filter as

[2]In (4.6), the sum-trace (transmit power) constraint of 1 on $\mathbf{Q}_{c(i),\pi^j_{c(i),k}}$ assumes the transmit power $P_t(b)$ for BS b is embedded in the channel matrix, as we have done in (4.1); this method is equivalent to not embedding the power in the channel matrix, but instead using a transmit power constraint of $P_t(b)$ for BS b. Embedding the power in the channel matrix also allows the rates to be calculated independently of the type of transmission nodes (and their specific transmitted power levels) used in the network. For simplicity of calculation, we have used a sum-power constraint over all coordinated BS antennas instead of per-BS constraints.

$$\bar{R}_{c(i),l}(t+1) = \begin{cases} (1-\frac{1}{t_c})\bar{R}_{c(i),l}(t) + \frac{R_{c(i),l}(t)}{t_c} & \text{if the } l\text{th user is scheduled in interval } t \\ (1-\frac{1}{t_c})\bar{R}_{c(i),l}(t) & \text{otherwise} \end{cases} \tag{4.7}$$

where $R_{c(i),l}(t)$ is the instantaneous rate of the lth user, and is obtained from (4.5), assuming the lth user is scheduled in position k of the ordered scheduling vector $\pi^j_{c(i)}$. One important special case of achievable weighted sum rate maximization is MT, which is defined by setting $\mu_{c(i),l}$ to a constant of 1 for all users. Let us define the best ordered user vector as π^*, then in any clustering pattern i, the maximum average achievable weighted sum rate over the area of an arbitrary macrocell, averaged over time t when using pattern i, is given as

$$\mathbb{E}_t\big(WSR(t,i)\big) = \mathbb{E}_t\Big(\big(\sum_{k=1}^{U_{c(i)}} \mu_{c(i),\pi^*_{c(i),k}}(t) R_{c(i),\pi^*_{c(i),k}}(t)\big)\Big)/w_{c(i)} \tag{4.8}$$

where $w_{c(i)}$ is the number of macrocells in cluster $c(i)$. To solve the optimization problem in (4.6) using (4.5) as $R_{c(i),\pi^j_{c(i),k}}$, we need to consider the $\mu_{c(i),\pi^j_{c(i),k}}(t)$ weights in calculating the power allocation during the water-filling algorithm over the eigenmodes of the block-diagonal matrix formed using the effective channel matrices[3] $\mathbf{G}_{c(i),\pi^j_{c(i),k}} = \mathbf{H}_{c(i),\pi^j_{c(i),k}} \mathbf{V}^0_{k-1}$. The user selection within a cluster is performed by using an SAS algorithm similar to what we proposed in [85], and described in Algorithm 3 therein. The main difference is the solution values are now achievable weighted sum rates as per (6); the rest of the operation of the algorithm is unchanged. We refer the reader to [85] for more details on our SA algorithm. Note that no change to the scheduling algorithm is required for rotating clustering; better cluster patterns for users are automatically detected by the algorithm through the corresponding more favourable channel gains and/or

[3]This is similar to [84, eq. (32)], except replacing the $|\mathbf{h}_k\mathbf{w}_k|^2$ terms there with squared singular values of $\mathbf{G}_{c(i),\pi^j_{c(i),k}}$.

achievable rates during that pattern, making the users more likely to be scheduled during those better patterns.

4.2.4 Complexity Comparison of Dynamic Clustering and Rotating Clustering

Fully dynamic clustering (whether the scheme in [83] or otherwise) results in significantly higher overhead in calculation and signalling. The system must determine and exchange possible choices of BSs for each user, run some sort of optimization or other routine to determine the choice of which BSs to serve which user, and finally communicate those choices across the network and to the users. This could occur potentially as often as every scheduling interval, though the system could also perform these operations less frequently. In comparison, almost none of those calculations are required with rotating clustering, since the sets of clusters are predetermined beforehand, and known at all transmitting nodes. The additional overhead beyond that of static clustering is simply the same as the last part of fully dynamic clustering, i.e. to periodically inform the users what their new cluster will be. Furthermore, there are additional savings in complexity in regards to cell association. With rotating clustering, the association of a user to a specific anchor BS has much less impact on the network's operation (disregarding the context of high user mobility and/or hand-off, which are outside the scope of this work). Note that a user receives data from a macrocell and all picocells overlaying that macrocell. Borders between macrocells (where the received power from the BSs of those cells are equal) are statistically identical; at times that cell border may be also be a cluster border, while at other times it won't. Thus, a complicated cell association scheme is not required. Whether a user chooses an anchor BS by closest distance, highest average received power, adding on a tier-dependent association bias factor, etc., the performance of the scheme is by and large unchanged. Essentially, users can be considered more to be associated

with a cluster rather than with an individual cell; in terms of performance, it is largely equivalent to associate with any one of the cells in the cluster. There may still be, for example, considerations of offloading traffic, but these would now be between clusters rather than between cells; in any event, such factors are beyond the scope of this paper.

4.3 Simulation Setup and Results

In this section, the simulations of the proposed rotating clustering mechanisms are presented and compared with fixed clustering of the otherwise identical cooperative HetNet, employing the SZF-DPC precoding technique and SAS algorithm. Both MT and PF scheduling are considered. The average achievable sum rate and the average achievable rate per user are determined using the Monte Carlo simulation method. The numbers of transmitter and receiver antennas are assumed to be $N_{macro} = N_{pico} = 2$ and $M = 2$, respectively. It is assumed that the ISD in both layouts has the same value of[4] $D = 1732$ m. In the 6-cell layout, there are 12 pico BSs spaced evenly on the imaginary circle with radius 693 m around each macro site. For the 3-cell layout, 4 picos are located on a circle of radius 356 m centered at the center point of each macrocell coverage area[5]. The scaling factor for the SNR, i.e. Γ_0 in (4.1), is set to result in an SNR of 9.6 dB at the distance $R_m = 866$ m in the 6-cell layout, and an SNR of 4.7 dB at $R_m = 1155$ m for the 3-cell layout, where in both cases R_m is measured from the

[4]Using this ISD, the circle circumscribing the macro site coverage area in the 6-cell layout has a radius of $D/\sqrt{3}$, which equals 1 km. The radius of the circle circumscribing a single macrocell in the in 3-cell layout is $D/3$. Using simple geometry, the area covered by a macrocell in the 3-cell layout can be calculated as twice that of a macrocell in the 6-cell layout. Since the 6-cell layout has twice as many macrocells, the total coverage area of a macro site is the same in both layouts.

[5]In both layouts, the location of pico BS is selected such that the boarder of its covered area touches the border of overlaid macro cell area. This coverage area is decided according to the comparison between received SINRs from the pico BS and the corresponding macro BS, which pico BS belongs to.

macro site in the direction of the BS antenna boresight. The transmitting power $P_t(b)$ of each macro BS is 40 times greater than that of each pico BS. The path loss exponent $\alpha(b)$ is assumed to be 3.91 and 3.67 and the standard deviation of the log-normal shadow fading is $\sigma_\rho = 6$ dB and 4 dB, respectively, for macro and pico BSs[6].

The total interference from the other clusters outside the target cluster is approximated by the complex Gaussian random vector with zero mean and standard deviation σ_I. This variance is measured and averaged over the area of an entire cluster by employing similar methodology for both layouts as was used in [85]. The mean value of σ_I in the 6-cell layout is measured as 13.8. In the 3-cell layout, all 5 cluster patterns are not exactly symmetric nor statistically identical to each other. As depicted in Fig. 4.2, the cluster patterns in Fig. 4.2(a,c,e) are similar in that for all three patterns, two sites contribute to a given cluster. Of the three macro BSs covering the cluster, two of them are co-located at the same site. However, in the cluster patterns of Fig. 4.2(b,d) there are three equidistant macro BSs per cluster, with each BS belonging to a different site. This difference creates asymmetry in some features of these patterns. Most notably, the mean standard deviation of the interference across the clusters of Fig. 4.2(a,c,e) is equal, but different from that of Fig. 4.2(b,d), with values of 21.3 and 26.3, respectively, for the two cases. We account for these differences in our simulations. In the SA algorithm, we use parameter values corresponding to SA-m case #18 in [85]; we refer readers to that reference for details. A summary of the simulation parameters and their values is provided in Table 4.1.

[6]The values of the path loss exponents and the shadow fading standard deviations used here for macrocells and picocells correspond to the Urban Macro (UMa) non-line-of-sight (NLoS) scenario and the Urban Micro (UMi) NLoS scenario, respectively, as found in [42].

Table 4.1: Simulation Setup Parameters and Values for 6-cell and 3-cell Layouts

	6-cell Layout	3-cell Layout
Number of Tx antennas per macro BS, N_{macro}	2	
Number of Tx antennas per pico BS, N_{pico}	2	
Number of Rx antennas per user, M	2	
Macro antenna pattern (including 3-dB beamwidth θ_{3dB})	Directional [78]; $\theta_{3dB} = 35°$	Directional [78]; $\theta_{3dB} = 70°$
Pico antenna pattern	Omnidirectional	
Path loss exponent, α	3.91 (macro); 3.67 (pico) [43]	
Log-normal shadow fading standard deviation, σ_ρ	6 (macro); 4 (pico) [43]	
Inter-site distance, D	1732 m	
Distance from pico BS to macro site	693 m	–
Distance from pico BS to center of macrocell coverage area	–	356 m
Reference distance, R_m	866 m	1155 m
Macro SNR at distance of R_m along antenna boresight	9.6 dB	4.7 dB
Interference standard deviation, σ_I	13.8	21.3 (Patterns (a,c,e)); 26.3 (Patterns (b,d))
PF averaging window size, t_c	100	40
SAS parameters	Case #18 in [85]	

4.3.1 Simulation Results for 6-Cell Layout

The clustering pattern is changed every T_{cl} scheduling intervals, while the PF metric averages the achievable rates over t_c scheduling intervals. The ratio of $\frac{t_c}{T_{cl}}$ is defined as the speed of cluster rotation. We set $t_c = 100$ for the size of the averaging window in the PF metric. As an example, if the clustering pattern is changed every $T_{cl} = 50$ intervals, the cluster rotation speed is 2, which means that over the duration of t_c intervals, the clustering pattern will change twice, with each change occurring 50 scheduling intervals apart (see Fig. 4.3). We considered a minimum of four full sets of rotations through both patterns for our simulations (with that minimum occurring at the slowest rotation speed).

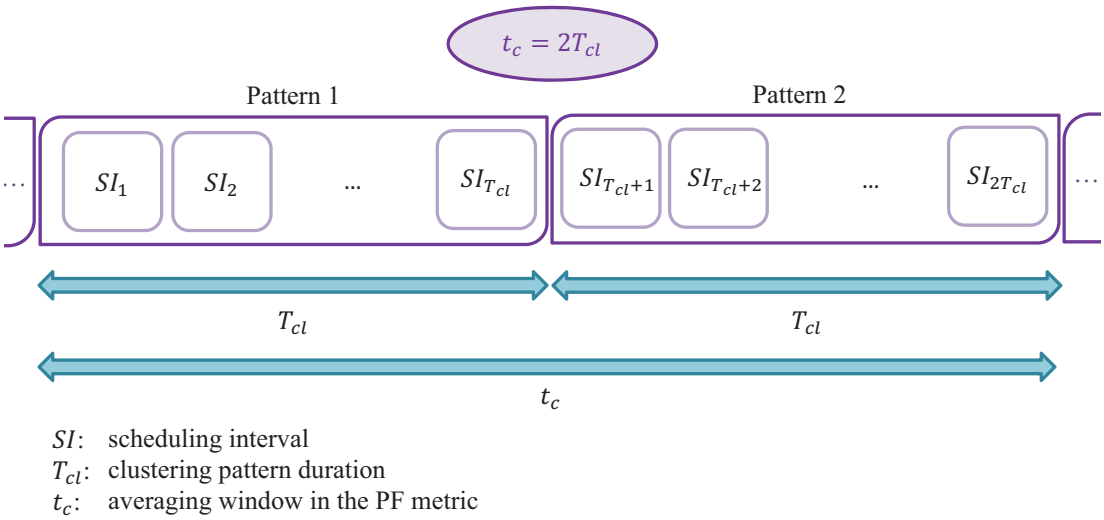

Figure 4.3: Comparison of averaging interval of proportionally fair metric, t_c, with clustering pattern duration T_{cl}, for an example with $t_c = 2T_{cl}$.

The cumulative distribution functions (CDFs) of the average achievable rates per user for $K_c = 12$ and various T_{cl} values are depicted in Fig. 4.4. As this figure shows, rotating clustering has the most benefit in particular for those users with poorer channel gains (e.g. those with an average achievable rate around the 5th percentile). In comparison, users with better channels (who achieve higher average rates, such as those around the 90th percentile) see their average achievable rate drop with rotating clustering. The poorer 5th percentile users are those having low SINRs in a cluster in one of the patterns. Their achievable rates significantly benefit from rotation because they are located in a position in the cluster where they can have a better SINR after the rotation. The better users trade off their average achievable rate to provide more overall fairness to the system. Considering different rotating speeds, Fig. 4.4 depicts that the rotation rate has an impact on each user's average rate, the most easily seen for lower-rate users like those around the 5th percentile. The 5th percentile per-user rates also generally increase with decreasing T_{cl}, but eventually reach an upper limit, where even faster rotation yields no further significant gains.

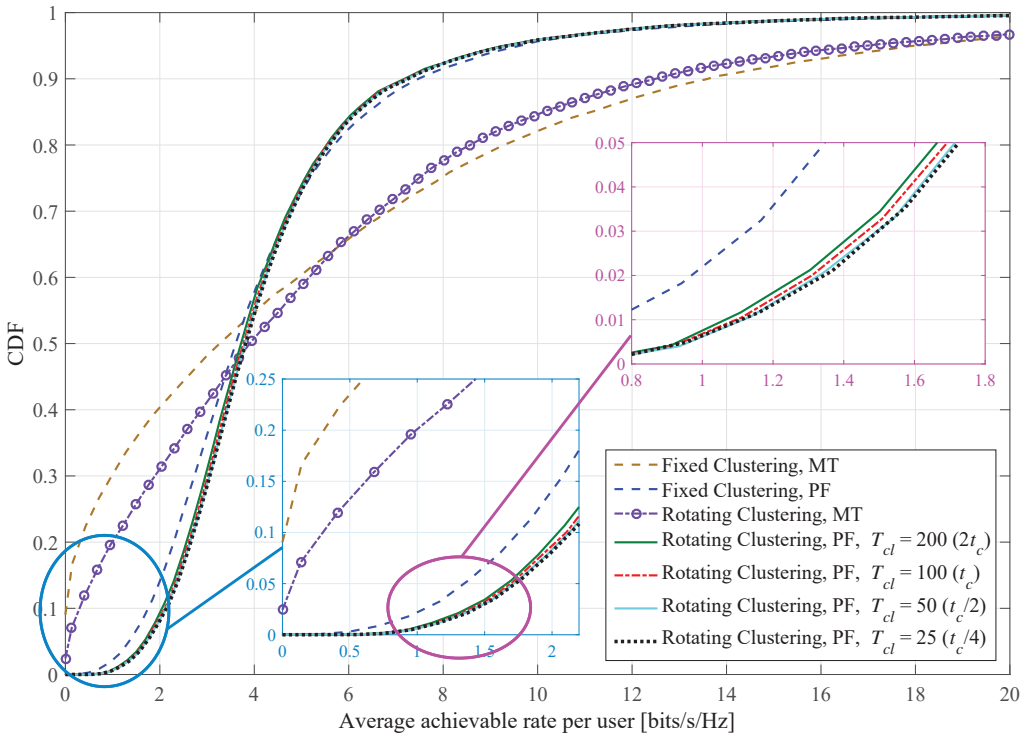

Figure 4.4: CDF of average achievable per-user rate with both proportionally fair (PF) and maximum throughput (MT) scheduling metrics, comparing proposed rotating clustering scheme and fixed clustering for 6-cell layout, using simulated annealing scheduling and SZF-DPC precoding. $K_c = 12$, $N_{macro} = N_{pico} = 2$, $M = 2$.

In each cluster, there are some users with particularly poor channel gains and even cluster rotation cannot help improve them much (such as those users located near the cell borders). These users have a very low chance to be selected by the MT scheduler and their average achieved rate is very close to (and sometimes equal to) zero, especially for fixed clustering. Cluster rotation helps these users by occasionally given them better channels; however, there still remain some users that suffer from starvation. Nonetheless, although the probability of starvation is still non-zero for MT scheduling, it is reduced significantly (by a factor of over half for $K_c = 12$), by using cluster rotation, as seen in Fig. 4.4.

With rotation, the best users' average achievable rates are reduced because

they may be scheduled less often; however, they achieve higher instantaneous rates when they are selected. As an example, consider a theoretical extreme case of maximum throughput scheduling, where with fixed clustering a few users would be scheduled most of the time with high throughput. With rotating clustering, instead about twice as many users would be scheduled overall, half of this total in each clustering pattern. As they would be scheduled about half as often (mostly only during their favourable pattern), their average achievable rate would also be about halved, but they would achieve somewhat better instantaneous rates when scheduled, leading to an increased (albeit slightly) achievable sum rate. Similar and larger effects on the instantaneous and average achievable rates are seen with PF scheduling. As expected, while the average sum rate of PF is less than that of the maximum throughput scheduling, its 5th percentile average achievable per-user rates are higher than for maximum throughput, and there is less overall variation in the average per-user rates achieved.

Fig. 4.5 shows the average achievable sum rate for maximum throughput and proportionally fair scheduling vs. K_c over the area of a macrocell in an arbitrary cluster. In Fig. 4.5, $T_{cl} = 100$, which is equal to t_c. As seen, rotating clustering outperforms fixed clustering for both scheduling metrics. However, rotating clustering increases the average achievable sum rate with PF relatively more than with maximum throughput. For instance, for MT scheduling, rotating clustering provides slightly higher throughput, increasing about 0.8% and 0.7% for $K_c = 4$ and $K_c = 12$, respectively, while for PF scheduling the throughput is more significantly higher (about 1.4% and 3.2%, respectively, for $K_c = 4$ and 12). This is expected because the users that take the most advantage of the rotation are the users either near the border of the cluster or in poor coverage areas, who are scheduled more often with PF than with maximum throughput.

In Fig. 4.6, the average achievable sum rate vs. K_c for proportionally fair scheduling considering different rotation speeds is presented. As seen, while ro-

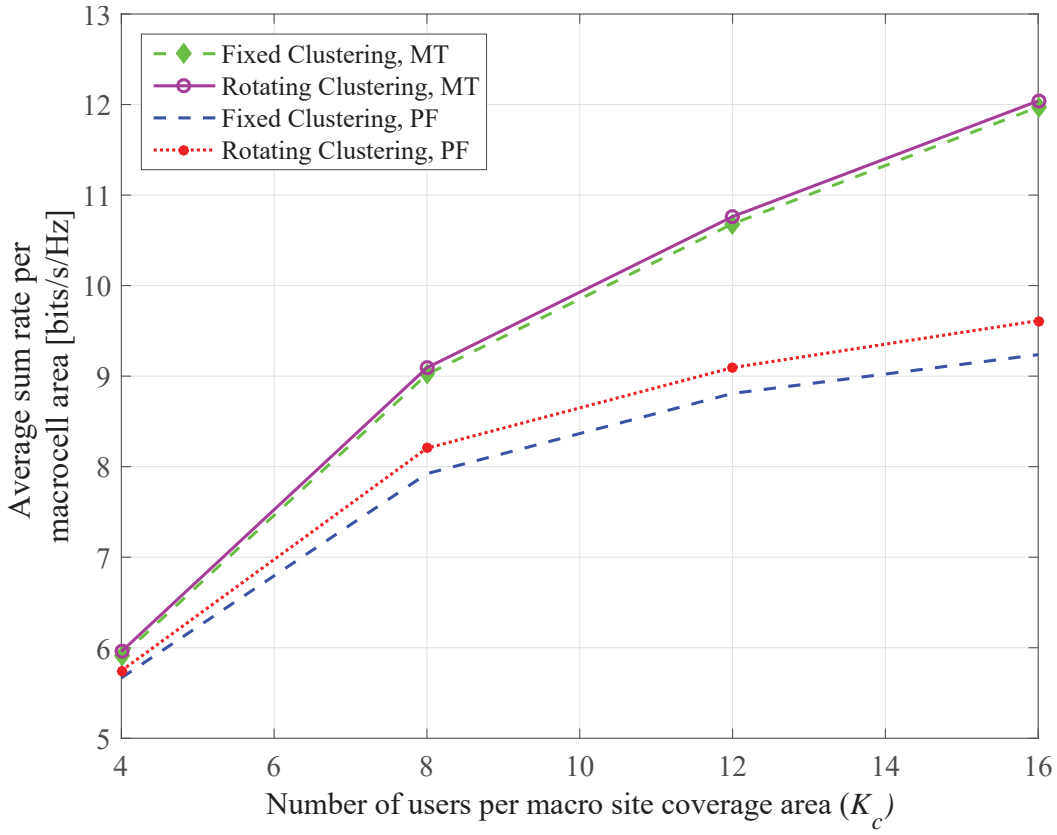

Figure 4.5: Average achievable sum rate vs. K_c with maximum throughput (MT) and proportionally fair (PF) scheduling metrics over the area of a macrocell in an arbitrary cluster, comparing proposed rotating clustering scheme and fixed clustering for 6-cell layout, using simulated annealing scheduling and SZF-DPC precoding. $N_{macro} = N_{pico} = 2$, $M = 2$, $T_{cl} = t_c = 100$.

tating clustering still outperforms fixed clustering in terms of sum rate, faster rotation (i.e. smaller T_{cl}) yields diminishing gains. For instance, rotating clustering provides higher throughput with respect to fixed clustering with faster rotation[7], increasing by about 2.1%, 3.2%, 4.1% and 4.5% with $K_c = 12$ for $T_{cl} = 200$, 100, 50, and 25, respectively. Considering an upper limit seen in the sum rate, rotating faster would not provide any further significant increase, but would increase complexity in signalling overhead for cluster setup. For our sys-

[7]A similar evaluation has been performed for MT; the results show no notable improvements with higher rotation speeds. This is also expected because MT overall gains very little in sum rate by using cluster rotation in contrast to PF.

tem, the limit is reached at about $T_{cl} = 50$ (half of t_c). Smaller values of T_{cl} (e.g. $T_{cl} = 25$) yield almost no additional sum rate; the additional gain in sum rate relative to fixed clustering is only a few tenths of a percentage.

Assuming the scheduler has selected a user with poor channel gains in a cluster, the chance of this user being chosen again by the scheduler is comparatively low, since its priority weight will have likely dropped in accordance with the update of its average rate by the exponential filter. The user's priority will gradually increase as time passes if the user is not scheduled. If the cluster pattern change interval is smaller than t_c, the possibility of that user being in a better position in another cluster pattern, and consequently dramatically improving its priority weight by virtue of its higher SINR and thus achievable rate, is increased. Similarly, a user with high SINR in a cluster that finds itself near a cluster edge after the pattern rotates will have its priority weight suddenly drop. It will be less likely to be scheduled until either the PF scheduling window passes or the cluster pattern rotates back, whichever comes first. Hence, faster rotations relative to t_c can potentially result in higher priority weights and higher sum rates. However, smaller T_{cl} does not necessarily mean a linear increase in priority weights with T_{cl}.

With larger K_c, cluster rotation is more effective at increasing the sum rate; for example, the achievable gains with $T_{cl} = 100$ are 1.4% and 3.2% respectively for $K_c = 4$ and 12 using PF scheduling, while faster rotation with larger K_c is more beneficial. For example, with $K_c = 8$ and comparing the rotation speed of $T_{cl} = 200$ to $T_{cl} = 100$ and 50, the sum rate gains are 0.9% and 1.4%, respectively, whereas with $K_c = 16$, the gain increments are 1.4% and 2.4%. The probability of having users that are located in the cluster with favourable channel gains is increased with larger K_c, simply as a result of multiuser diversity. Thus, being able to schedule more of these users improves the sum rate. Recall, though, that users are uniformly distributed over the entire coverage area. This statistically

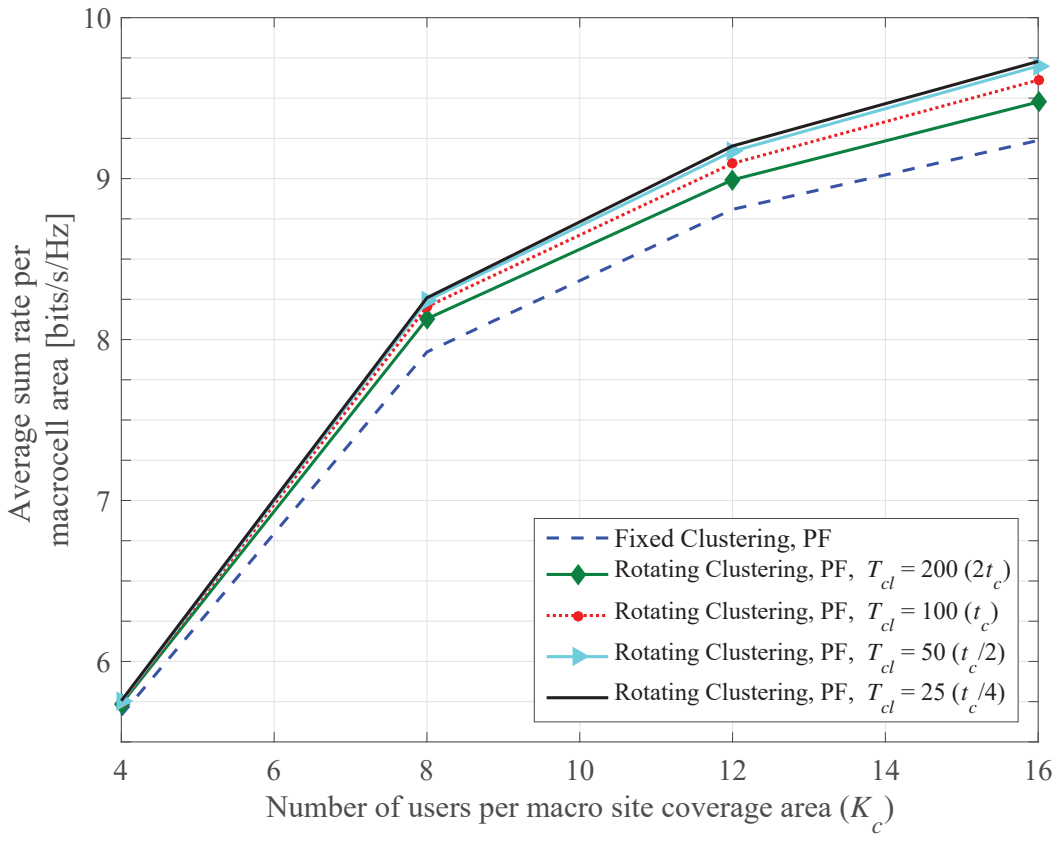

Figure 4.6: Average achievable sum rate vs. K_c over the area of a macrocell in an arbitrary cluster, comparing various T_{cl} values for proposed rotating clustering scheme for 6-cell layout, using simulated annealing scheduling with proportionally fair (PF) scheduling metric ($t_c = 100$) and SZF-DPC precoding; $N_{macro} = N_{pico} = 2$, $M = 2$.

results in a higher proportion of users who are farther from a BS (with poorer channel gains) than those who are nearer. An increase in K_c thus also means an increase in the total number of "poorer" users. As was seen, it is those users who benefit the most from rotation, explaining further why the gains with rotation are better with higher K_c.

An increase in K_c also means more delays in scheduling users, as a larger pool competes for the same limited resources. Faster rotation also means less of a wait for any given user's cluster pattern to be at its best, and thus potentially for the user to be scheduled, improving their average rate. In other words, faster

rotation somewhat compensates for the increased scheduling delays as K_c grows, thus leading to the higher gains in rate observed with faster rotation at larger K_c.

Figs. 4.7 and 4.8 further demonstrate the effects of rotating clustering (with $T_{cl} = 100$) by depicting the average achievable rate per user based on the user's position within the cellular network; Fig. 4.7 depicts MT scheduling, whereas Fig. 4.8 depicts PF. Figs. 4.7(a) and 4.8(a) show the average achievable rates with fixed clustering. The highest rates are unsurprisingly achieved by users nearest to a BS, especially macro BSs, but also to a lesser extent pico BSs. The lowest achievable rates (darkest red) are seen near any macrocell corner where three clusters meet. We will henceforth refer to these areas as the "corner" areas for shorthand. Lower rates are also seen in the area around the border of each cell, whereas somewhat higher achievable rates (light red to yellow) are seen near areas that correspond to the directions of the macro BS boresights. For any macro BS, users located near the left or right borders of the cell (i.e. at a 30° angle relative to the antenna boresight) receive weaker signals on account of the antenna directivity pattern. This situation is at its worst near the corners of the macrocells, at the furthest distance from the site. Furthermore, although the corners are surrounded by pico BSs, the received signals from those pico BSs are also weak due to being beyond the picocell borders. Hence, even with clustering (either fixed or rotating) and coordination, users in the corners achieve consistently smaller user rates, compared to the users in other locations of the cluster.

Figs. 4.7(b) and 4.8(b) show the achievable rates with rotating clustering. In comparison, the average rates (or the colours) are more evenly distributed over the entire area, indicating higher overall fairness. As expected, the largest increase in average achievable rate is experienced by users that were closest to the cluster borders in the fixed scheme. Those users previously received the worst

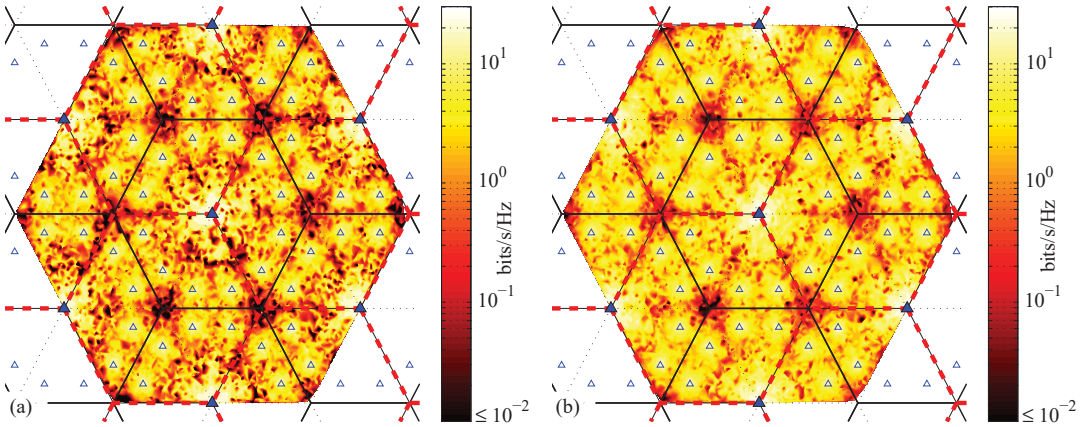

Figure 4.7: Average achievable rate per user based on user position within cellular network, with MT scheduling metric for 6-cell layout, using simulated annealing scheduling and SZF-DPC precoding; $N_{macro} = N_{pico} = 2$, $M = 2$, $K_c = 12$. (a) Fixed clustering. (b) Rotating clustering ($T_{cl} = 100$).

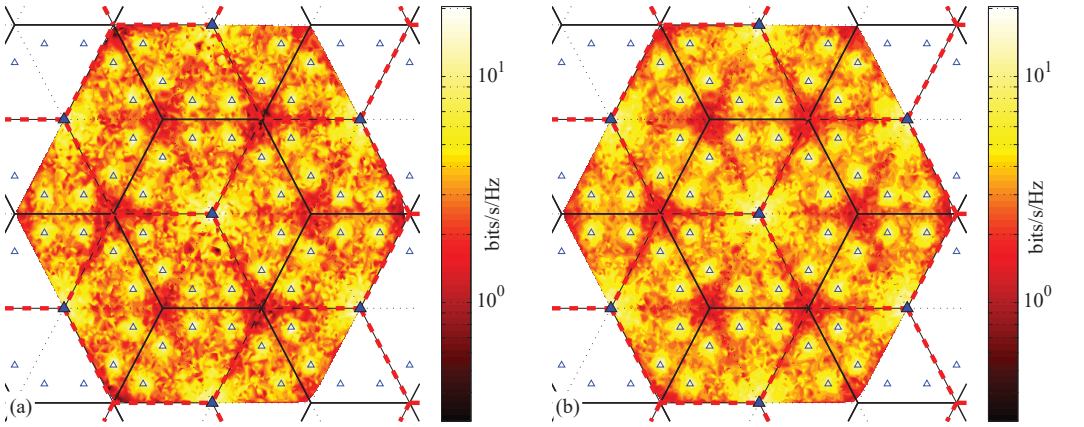

Figure 4.8: Average achievable rate per user based on user position within cellular network, with PF scheduling metric for 6-cell layout, using simulated annealing scheduling and SZF-DPC precoding; $N_{macro} = N_{pico} = 2$, $M = 2$, $K_c = 12$. (a) Fixed clustering. (b) Rotating clustering ($T_{cl} = 100$).

SINRs and/or the most ICI. In contrast, most users who are located close to the macro BSs or near the directions of the macro boresights receive very good signal power. The largest decrease in average achievable rates relative to the fixed scheme is by that latter group of users, as well as to a lesser extent users nearby pico BSs. It is these latter users who trade off some of their average rates to

provide more fairness and uniformity of throughput across the coverage area. It is interesting to note that there are fairly large areas where the average achievable rate of a user does not change much with rotating clustering. These areas and users are those located in the interior of the cluster. Their conditions (in terms of both useful signals and interference) are more or less the same under either cluster pattern. Hence, rotation does not change their situation much, and so their achievable rates do not change appreciably from the fixed scheme either. There is also little difference in the rates seen in the cell "corner" areas with rotation. The rather small changes that exist there are not readily visible in Figs. 4.7 and 4.8. However, where 3 clusters meet in the fixed scheme, those corners do experience a small boost in rates with rotation due to periodically increased coordination. Likewise, the corners at the cluster centers in the fixed scheme see a small decrease in rates with rotation, as signals received there are no longer always coordinated from all nearby BSs.

4.3.2 Simulation Results for 3-Cell Layout

Similar to the previous section, the clustering pattern changes (rotates) every T_{cl} scheduling intervals. For the 6-cell layout, as described in Section 4.3.1, we simulated for a minimum of 4 complete sets of rotations. Assuming $T_{cl} = 200$, which corresponds to the slowest (non-zero) rotating speed, the total number of simulated channel realizations (or scheduling intervals) for every drop of users is thus 1600. To compare the 3-cell layout with the 6-cell layout, we considered two different scenarios. First, we keep the same total number of 1600 channel realizations per drop with 4 complete sets of rotations between the 5 clustering patterns. We also maintain the longest pattern duration as $2t_c$. Thus, t_c for the PF window becomes 1600 realizations/drop $\div$4 sets/drop $\div$5 patterns/set $\div$2 windows/pattern = 40 (realizations per window). This gives us different considered rotating speeds corresponding to pattern durations of $T_{cl} = [10, 20, 40, 80]$.

Second, we keep the value of t_c equal to what we have used in Section 4.3.1, i.e. 100, and the same pattern durations, i.e. $T_{cl} = [25, 50, 100, 200]$. With still 4 full sets of rotations at minimum, this results in 4000 channel realizations per drop. The simulations with these two scenarios were run for $K_c = 12$ and the results were compared with each other. We observed that the results of both scenarios were very close to each other[8]. Thus, we present only the results for where the number of channel realizations per drop equals[9] to 1600 and $t_c = 40$.

In Fig. 4.9, the CDF of the average achievable rates per user for $K_c = 12$ is illustrated. Similar to the results obtained from Fig. 4.4, rotating clustering again yields higher per-user rates for those who are located near cluster edges (or in otherwise poor coverage areas) in the 3-cell layout (e.g. the users with rates around the 5th percentile). Comparison of these two figures is also interesting. As was seen in Fig. 4.4 with MT scheduling, while rotating clustering improved the per-user rates in the 6-cell layout, there were still some users whose achievable rates' were very low, even to the point of starvation. However, cluster rotation in the 3-cell layout improves the average throughput of users considerably such that the probability of very low rates and/or starvation becomes almost zero. There is a larger tradeoff in the high-percentile users' rates to achieve this, though.

In the 3-cell layout and with PF scheduling, the rotation speed corresponding to $T_{cl} = t_c$ yields a per-user rate increase of about 0.5 bits/s/Hz at the 5th percentile, while the improvement with the 6-cell layout is around two-thirds of that. In relative terms, the 5th-percentile rate improvements rotating clustering (with $T_{cl} = t_c$) over fixed in the 6- and 3-cell layouts are, respectively, about 22.8%

[8]This is likely as a result of the assumption that there is no temporal correlation in the small-scale fading component of the users' channel gains, i.e. each realization of $z_{c(i),k}(b, m, n)$ in (4.1) is independent from any other. If there was temporal correlation, the results of the two scenarios would likely differ.

[9]For certainty, we simulated all cases for both scenarios and in each case the achieved results were similar. Aside from the fact that fewer realizations means less computation time for the simulations, using 1600 realizations for both the 6-cell and 3-cell layouts means a comparison of the results between the two layouts is somewhat more statistically equitable.

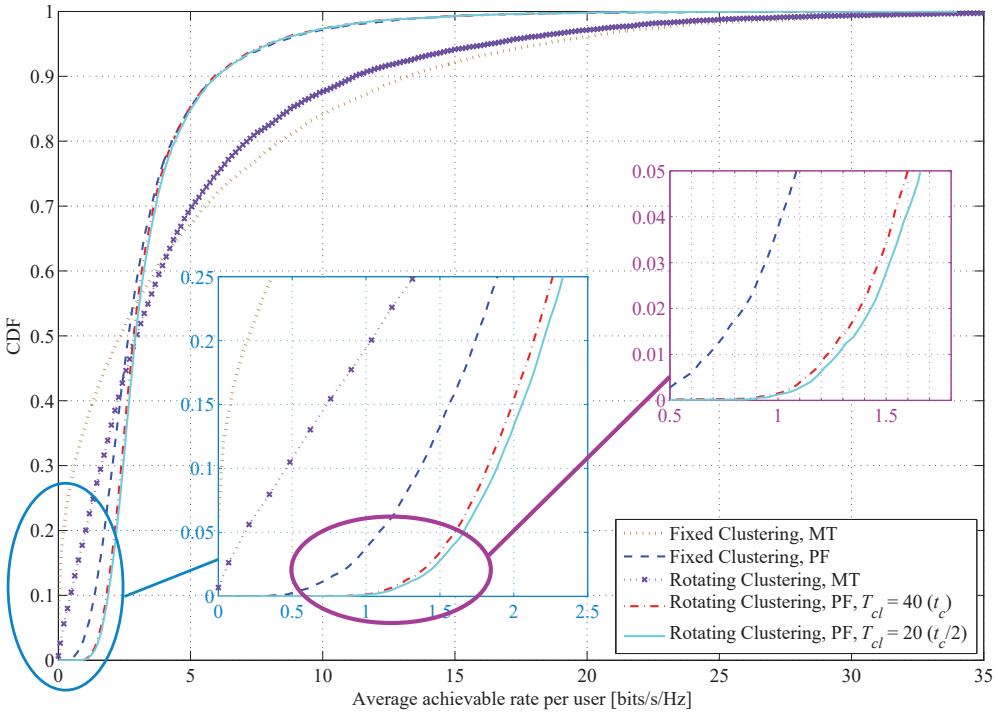

Figure 4.9: CDF of average achievable per-user rate with both proportionally fair (PF) and maximum throughput (MT) scheduling metrics, comparing proposed rotating clustering scheme and fixed clustering for 3-cell layout, using simulated annealing scheduling and SZF-DPC precoding. $K_c = 12$, $N_{macro} = N_{pico} = 2$, $M = 2$.

and 48.4%. This indicates cluster rotation is even more effective at improving the cluster-edge user rates in the 3-cell layout. Considering the faster rotation speed corresponding to $T_{cl} = t_c/2$, the additional 5th-percentile users' rate increase (vs. $T_{cl} = t_c$) is about 2.3% in the 6-cell layout, while this increase is about 4.4% using the 3-cell layout. This shows that faster rotation in the 3-cell layout is more beneficial than in the 6-cell layout to help cluster-edge users to gain further higher throughput, which is not surprising, considering the former rotates through a larger set of patterns.

We also investigate the user rates vs. their position within the network in Figs. 4.10 (MT scheduling) and 4.11 (PF scheduling). Basically, the influence of cluster rotation on achievable per-user rates with MT and PF scheduling is similar

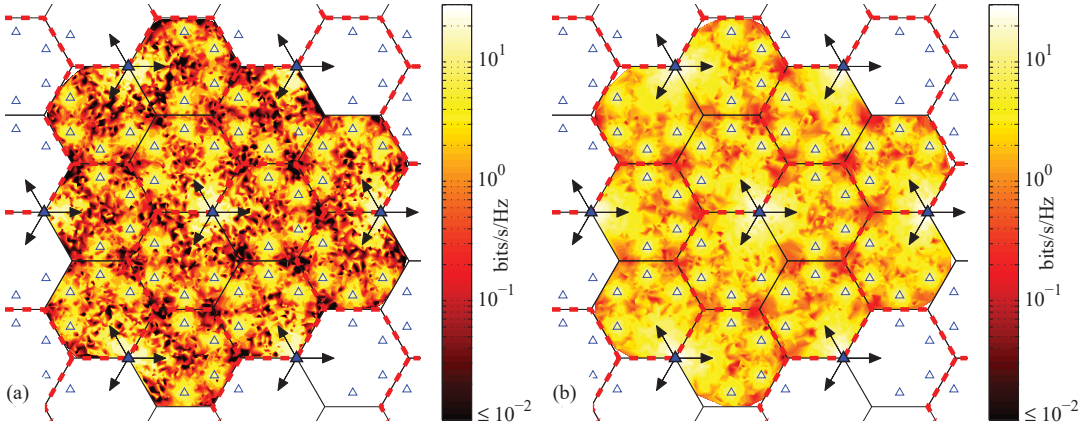

Figure 4.10: Average achievable rate per user based on user position within cellular network, with MT scheduling for 3-cell layout, using simulated annealing scheduling and SZF-DPC precoding; $N_{macro} = N_{pico} = 2$, $M = 2$, $K_c = 12$. (a) Fixed clustering. (b) Rotating clustering ($T_{cl} = 40$).

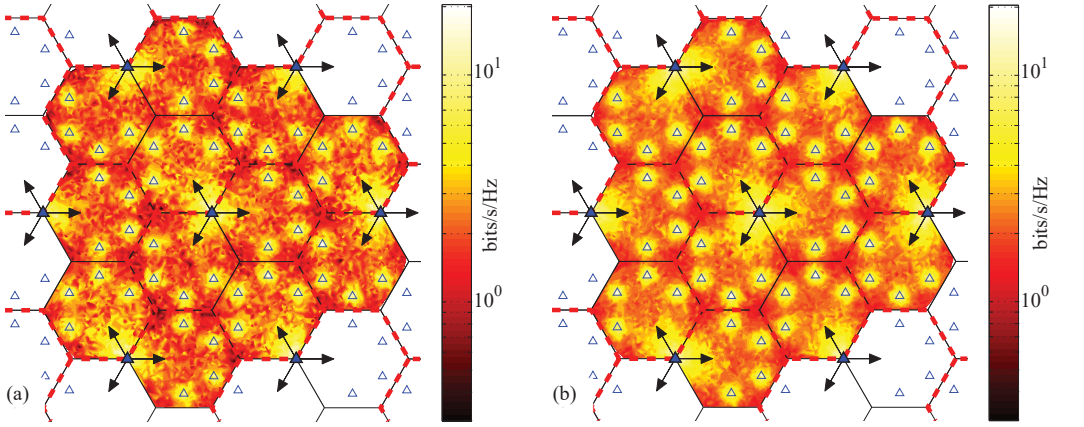

Figure 4.11: Average achievable rate per user based on user position within cellular network, with PF scheduling for 3-cell layout, using simulated annealing scheduling and SZF-DPC precoding; $N_{macro} = N_{pico} = 2$, $M = 2$, $K_c = 12$. (a) Fixed clustering. (b) Rotating clustering ($T_{cl} = 40$).

to what we have discussed in Section 4.3.1. As before, it is users located at the cell corners away from the sites that achieve the lowest rates. To a lesser extent, users near the middle of cells, not especially near a macro BS nor a pico BS, also achieve somewhat lower rates. As is obvious from Figs. 4.10 and 4.11, the achieved users' rates are distributed more uniformly with rotating clustering than

with fixed clustering for both MT and PF scheduling. Furthermore, the overall rates are increased. The achievable throughput of those users who are located in the corners of a cell is improved, and the size of the areas experiencing lower rates is diminished. This is particularly noticeable in Fig. 9 for MT scheduling; the colors in the cell corners increase from dark red to orange.

Cell-center users are also significantly affected by cluster rotation. Note that in fixed clustering, and in the patterns in Fig. 4.2(a,c,e), two parts of the cluster come from the same site. Because of the macro BS antenna pattern, the coverage of the beams from those two BSs has little overlap. Hence, those two cells are essentially impacted by, at best, two coordinated signals: one from their own cell, and one from the third BS at the other site that contributes to the cluster. However, with rotation through the patterns in Fig. 4.2(b,d), receiving a significant coordinated signal from 3 BSs is more common. Thus, users in those two cells in the cluster see the most improvement from rotation[10]. The tradeoff now comes from users in the third cell of the fixed scheme, who with cluster rotation now are forced to experience less advantageous cluster patterns for the sake of overall network fairness.

Fig. 4.12 demonstrates the achievable sum rates vs. K_c for the 3-cell layout. Much like the 6-cell layout, MT scheduling again doesn't display a gain in sum rate from cluster rotation, for similar reasons as described in Section 4.3.1. However, the throughput with PF scheduling does again increase considerably. For example, with $K_c = 12$, comparing the achievable throughput of rotating vs. fixed clustering shows an increase of 4.5% and 6.7%, for T_{cl} equal to t_c and $t_c/2$, respectively. Comparing the 6- and 3-cell layouts with the same number of users in each cluster (such as $K_c = 12$), and considering that the ISD between macro sites is the same in both layouts, the 6-cell layout generally yields higher values for area spectral efficiency on average. For example, the average total sum rates

[10]This is despite the clusters of the patterns in Fig. 4.2(b,d) experiencing more surrounding interference, as seen by their larger value of σ_I in the simulation setup.

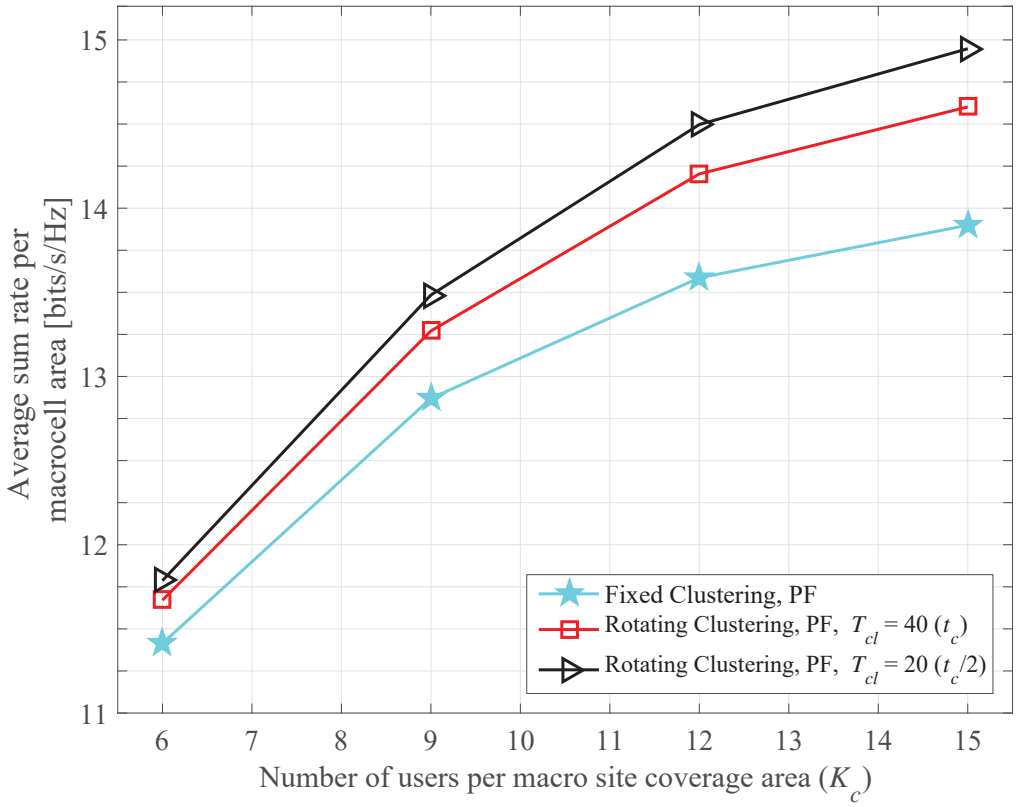

Figure 4.12: Average achievable sum rate vs. K_c with proportionally fair (PF) scheduling ($t_c = 40$) over the area of a macrocell in an arbitrary cluster, comparing various T_{cl} for proposed rotating clustering scheme and fixed clustering for 3-cell layout, using simulated annealing scheduling and SZF-DPC precoding. $N_{macro} = N_{pico} = 2$, $M = 2$.

per macrocell for $K_c = 12$ are 8.8 and 13.6 bits/s/Hz, respectively, for the 6- and 3-cell layouts with PF using fixed clustering, but also recall that the macrocell area in the 6-cell layout is half of that in the 3-cell layout. Hence, the sum rate per unit area (e.g. per km^2) is larger in the 6-cell layout. However, the per-user rate improvement achieved by using cluster rotation obtained in the 3-cell layout is much more significant. Faster rotation yields higher gains in this layout than the 6-cell layout. For instance, the additional sum rate improvement achieved by setting $T_{cl} = t_c/2$ compared to $T_{cl} = t_c$ is 2.1% in the 3-cell layout, while this gain in the 6-cell layout is only 0.8%. Although faster rotation improves the average

sum rate, the upper limit for higher speeds of rotation is still similar to what we described in Section 4.3.1; thus, we have only presented the results for T_{cl} equal to t_c and $t_c/2$. All these comparisons demonstrate that the rotating cluster method overall helps the system performance in the clover-leaf-shaped network layout more than the hexagonal-shaped network layout.

4.3.3 Comparison of Rotating Cluster Method with Dynamic Cluster Method

In [83], a dynamic clustering method was proposed and compared with static clustering. There are several differences in the simulation methodology in that paper compared to our own (e.g. a single-tier homogeneous network, differing numbers of users and antennas, the precoding method, etc.). However, it can still serve as a rough guide for the performance benefits of a fully dynamic scheme over static clustering, and that can be used for comparison with our less complex rotating clustering method. In [83], with 3 macrocells per site, 4 transmit antennas per macrocell, 2 antennas per user, and PF scheduling, the dynamic scheme was shown to have approximately an 18% gain in average sum rate per cell [83, Fig. 5], and about a 22% gain in 5th percentile user rate [83, Fig. 6], versus a static clustering scheme of 3 cells per cluster with one macrocell contributing from each site (similar to the macrocells in our 3-cell layout). The 5th percentile user rate gains approximately doubled when the maximum cluster size of the dynamic scheme increased from 3 to 6, though at the same time also enforcing that just a single data stream be sent to each user [83, Fig. 9]. In comparison, for our 6-cell layout, the 5th percentile user rates for $K_c = 12$ and PF scheduling increase by about 23% to 28% over static clustering for the minimum and maximum examined rotation speeds. In the 3-cell layout, the gains were even higher, i.e. 47-53% for $K_c = 12$. The gain in sum rate per cell was lower, about 2.0-4.5% for $K_c = 12$ in the 6-cell layout, and about 4.5-6.7% for $K_c = 12$ in the 3-cell layout. However,

we remind the reader that our rotating clustering scheme was primarily designed to help cluster-edge users (e.g. those around the 5th percentile), while any gains in cell sum rate are an added secondary bonus. Hence, we can see that, compared to the fully dynamic scheme of [83], our rotating scheme provides gains for cluster-edge users that are on par with or better than those in [83], but at the tradeoff of lower gains in the sum rate per cell.

4.4 Conclusion

In this paper, the downlink of a coordinated heterogeneous MIMO cellular network including macro and pico BSs in two different network layouts has been considered. The layout designs that have been employed here are a "traditional" hexagonal-shaped network layout and a clover-leaf-shaped network layout. We have proposed a different rotating set of clustering patterns for each cellular layout. The performance of the proposed schemes under SZF-DPC precoding and a reduced-complexity simulated annealing user scheduling algorithm, considering both throughput maximization and proportionally fair scheduling metrics, has been evaluated by simulation. The results demonstrate that this rotating clustering scheme performs better than fixed clustering for both metrics and in both layouts. The average achievable rate per user has been improved for those users achieving amongst the lowest rates in the system, as well as the average achievable sum rate. The first and foremost goal of rotating clustering is to improve the performance of cluster-edge users with less complexity than fully dynamic clustering; the enhancement of the overall sum rate throughput can be considered as a secondary benefit. Overall, users with the highest average achievable rates are scheduled less often with rotating clustering than with fixed clustering, thereby sacrificing some of their average performance for higher fairness to others. However, the instantaneous achievable rates of all users when they are scheduled tend

to be better with cluster rotation than without. User rates also become more evenly distributed over the network coverage area.

We have also evaluated the effect of different cluster rotation speeds on the system performance in both cellular layouts. The results demonstrate that faster rotation performs better than slower rotation, but there is an upper limit on the rotation speed, beyond which further increases do not result in any notable additional gains in sum rate or per-user rate. The clover-leaf-shaped network layout gains more in performance by using cluster rotation than the hexagonal-shaped layout. This indicates in general that some network layouts (and their associated cluster patterns) may benefit more by cluster rotation than others.

While we have investigated two regular grid-like network layouts in this work, the concept of rotating clustering can also be applied to more general irregular layouts. To consider cluster rotation within irregular layouts, it would first be necessary to determine sets of BSs in the network that should be coordinated, and then assign different patterns of clustering to them. This may not be as simple as with a regular layout, but it is certainly feasible, given a set of BS locations and the areas they are expected to cover and/or where they would cause interference without coordination. (Voronoi diagrams of order n [86] could be of use to locate regions where BSs should coordinate, by identifying the n nearest BSs at any given location; the distances should also be weighted based on the type/tier of each transmitting node.) The system can then rotate through those patterns just as in this work.

Chapter 5

User Scheduling Algorithms with Temporally Correlated Channel Gains

5.1 Introduction

In the analysis of wireless systems, fading channels are usually considered to have independent and identically distributed small-scale channel gains for different transmission intervals of the block fading channel. These uncorrelated time-varying channel gains provide the highest capacity [50, 51]. However, in most practical systems, channel gains demonstrate time-correlated small-scale fading [46–51], because of mobility in the system (either of the users or the environment around them)[1]. Temporally correlated (TC) fading channels provide less capacity than i.i.d. channels [50, 51]; the achievable capacity increases while the channel correlation decreases [50]. In spite of having temporal correlation of channel gains, the capacity improves linearly when the spatial multiplexing gain increases.

[1]There are also spatial correlations in the shadowing and path-loss components of the channel gains, as well as sometimes between antennas for the small-scale component in the case of MIMO systems. However, these are outside the scope of our work.

However, this correlation causes a downward shift of the capacity [50] (i.e., the difference between the capacity of uncorrelated and TC MIMO channels increases as the spatial multiplexing gain increases).

In general, when perfect CSI is available at both the transmitter and receiver, the transmitter is capable of optimally adapting its power, data rate, etc. to achieve the capacity of fading channel. Thus, the throughput on a TC fading channel can be improved by employing CSIT and adaptive transmission [51], which is also a requirement in MU-MIMO systems, as we have mentioned before. In frequency division duplexing (FDD) system, the BS transmits a training signal to the users, and each user measures and estimates its downlink channel (i.e., CSIR). For CSIT, the estimated CSIR is sent back to the transmitter by users over separate feedback link. Obtaining accurate CSI at the transmitter is a serious challenge in MU-MIMO systems, and has been addressed in the literature [44, 45, 87–90]. For the downlink of a MU-MIMO system, this CSIT is used to generate appropriate precoding matrices to reduce MUI and/or to find an appropriate set of users to be scheduled.

If the wireless channel between users and BSs experiences time-correlated fading, any two consecutive channel gains are more or less similar to each other depending on the value of their correlation coefficient [46–48, 51, 91] (i.e., their similarity to each other depends on how highly they are correlated). TC channels can be beneficial to increase the data transmission efficiency of the wireless network by exploiting the TC characteristic of the channel gains [50,91]. In fact, the characteristics of temporal correlation causes redundancy in CSI. The redundant CSI can be removed to reduce the amount of CSI required to be fed back. As a result, exploiting the time correlation of channels can help those practical systems with limitations on the feedback path. Moreover, the similarities in two consecutive channel samples can be exploited in the search process of user scheduling algorithms to improve their performance. Investigations of techniques that are

able to harness this characteristic of correlated channel gains and improve the performance of the downlink of MU-MIMO system are of interest.

5.1.1 Motivation and Contributions

In this chapter, we consider the downlink of a coordinated multi-cell MU-MIMO HetNet with the assumption of correlated channel gains. To evaluate the performance of the throughput, we consider two different metrics here: MT (as we used in Chapter 3) and PF. We first are interested to explore the performance of our already proposed user schedulers when fairness is considered. Similar to the approach used in Chapter 3, the parameter setup should also be chosen for the PF user scheduler. Then, we investigate the effect of temporal correlation of the channel on the output of the user scheduler to see if this correlation affects the performance of our proposed algorithms or their parameters, as well as if it can help the scheduler to achieve better performance or to explore the search space faster. If so, the achieved throughput of the system can be improved and/or the complexity of the user scheduling algorithms can be reduced notably, which is of most interest to us in design of the large wireless networks. In this chapter, we suggest to use the behaviour of correlated channels within the search process of the user scheduling algorithm in order to reduce its complexity and to improve the achievable sum rate.

We consider SZF-DPC as the precoding method. As with the results in Chapter 3, the outcomes of this chapter can be relatively trivially extended to different precoding methods. In Chapter 3, we proposed the HGPS algorithm; the performance of this algorithm was investigated when the MT metric was used. Using the PF metric, users are chosen in part according to their priority weights, which means that the users with higher priority will be chosen more likely by the scheduler. The user with the largest channel gain may be selected by the scheduler for a few transmission intervals, then its priority may be decreased dramatically

relative to the other users and it might not be selected again for some time. This is contradictory to the base HGPS algorithm, which always schedules the user with the largest channel gain. In this chapter, we investigate the performance of PS and SA algorithms under various parameter setups, using TC channel gains.

5.2 System Model

The coverage area of a macro BS site is partitioned into six cells each covered by a 60° directional macro antenna installed on a macro BS. In this chapter, each cluster includes two high power macro BSs and four low power pico BSs which are overlaid on the coverage area of the macros. A sample cluster is depicted in Fig. 5.1. Each macro directional antenna and pico omnidirectional antenna is equipped with N transmitting antennas. In each macro cell, there are K users uniformly distributed across the cell; as a result, the number of users in each cluster is twice K, i.e. $K_c = 2K$. The received signal by user k is modeled as

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x} + \mathbf{Z}_k \tag{5.1}$$

where $\mathbf{x}$ is the transmitted signal from the cluster and $\mathbf{Z}_k$ is the interference from other clusters plus noise, which is described in (4.2). $\mathbf{H}_k$ is the aggregate downlink channel of the kth user from all BSs in the cluster similar to Chapter 3, which accounts for path loss, log-normal shadowing and Rayleigh fading. Each element of $\mathbf{H}_k$ is denoted similar to (3.1), where for simplicity the suffix c is now omitted. In this work, we consider temporal correlation of small scale fading, which is caused by mobility of at least one of the communicating transceivers and/or mobility of scatterers in the environment. With mobility on a small scale (a couple of wavelengths), the shadow fading and path loss components of the channel gains can be assumed approximately constant, and for simplicity we can consider that they do not change over our simulation period. Thus, the correlation

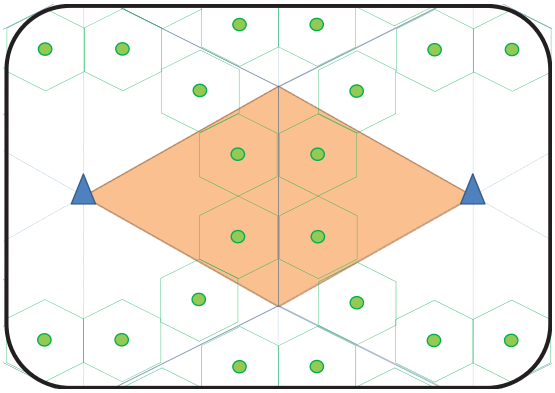

Figure 5.1: Schematic deployment of HetNet. Large triangle-shaped areas represent macro cells, while small hexagons denote pico coverage areas. The shaded diamond-shaped area is the cooperating cluster.

of the channel gains occur in the short-term fading, modeled as Rayleigh fading. According to (3.1), the Rayleigh fading element of $h_k^{(b,m,n)}[t]$ is modeled by the parameter $z_k^{(b,m,n)}[t]$, which is a complex Gaussian random variable. Assuming the correlation coefficient of a TC channel, one commonly used method to model the TC channel is known as Clarke's two dimensional isotropic scattering model [3, 92] , which assumes that the received signal from all scatterers around the receiver come from all azimuth directions with equal probability. One approach to generate these channel samples, for simulation purpose, is described in [93,94] and is well-known as filtered white Gaussian noise, and one method of this approach is IDFT (inverse discrete Fourier transformation) method.

In IDFT method, the correlated Rayleigh complex random variations are generated by passing two generated uncorrelated i.i.d. sequences of random variables through low pass filter to shape their power spectrum to the desired random process via computer simulations, details of this method are provided in Appendix B. The autocorrelation function of the output signal can be translated as the correlation coefficient of the signal, i.e. $\rho = J_0(2\pi f_d \tau)$, where f_d is the maximum Doppler frequency and τ is the time delay between two samples; the $f_d\tau$ is the normalized time delay [3, 92]. It is assumed that the transmitters have

perfect knowledge of CSI in every transmission interval, and try to create the corresponding precoding matrices. In (5.1), $\mathbf{x}$ is the transmitted signal and $\mathbf{Z}_k$ is the summation of additive white Gaussian noise and the interference from all other clusters, which is distributed as $\mathcal{CN}(0, (\sigma^2 + \sigma_I^2)\mathbf{I}_M)$, where $\sigma^2\mathbf{I}_M$ is the covariance matrix of AWGN noise and $\sigma_I^2\mathbf{I}_M$ is the covariance matrix of inter-cluster interference. The features of these two terms are similar to what we considered in Chapter 3 for the same equation.

In this chapter and without loss of generality, we only consider SZF-DPC as the precoding method, which is a sensitive precoding method to the order of the users for the maximization of the achievable weighted sum rate, and described in details in Chapter 3 and in [30]. Using similar procedure, for a given ordered user π_k^j, its achievable rate $R_{\pi_k^j}$ at time t is given by (3.5), by replacing $\tilde{\mathbf{H}}_{c(i),\pi_{c(i),k}^j}$ with $\mathbf{H}_k$ from this chapter. In each cluster, the maximum achievable weighted sum rate is given by

$$\max_{\pi^j : j \in \{1,2,\cdots,U!\}} \quad \max_{\substack{\left\{\mathbf{Q}_{\pi_k^j}\right\}_{k\in\{1,\cdots,U\}} : \\ \mathbf{Q}_{\pi_k^j} \succeq \mathbf{0},\ \sum_{\forall k} Tr(\mathbf{Q}_{\pi_k^j}) \leq 1}} \quad \sum_{k=1}^{U} \mu_{\pi_k^j}[t] R_{\pi_k^j}[t] \tag{5.2}$$

where $\mu_{\pi_k^j}[t]$ is the priority weight of the kth user during the tth scheduling interval, and U is the maximum number of the users which can be served simultaneously in the cluster (it is equal to the minimum of either total number of transmitting or receiving antennas). In PF scheduling, for the lth user in the cluster, $\mu_l[t]$ is defined similar to the procedure in Chapters 2 and 4. The rest of all calculations to generate the maximum achievable weighted sum rate at time t summed over the cluster is given as described in previous chapters, defining the best ordered user vector as π^*.

5.3 User Scheduling Algorithms with Temporally Correlated Channel Gains

In this section, we investigate the performance of already proposed user scheduling algorithms in Chapter 3 under the assumption of TC channel gains. In that chapter, SA and PS algorithms are analysed under the assumption of MT performance metric and uncorrelated channel gains. In this chapter, simulation results of PS and SA algorithms under the assumption of TC channels are presented and their performance is compared with optimal user scheduling using SZF-DPC precoding technique, for both MT and PF metrics. The average sum rate and average number of scheduled users are determined for the cluster using the Monte Carlo simulation method.

The numbers of transmit antennas per BS and receive antennas per user are assumed to be 2. It is assumed that the distance between macro BSs is 1732 m. The transmitting power of each macro BS in each sector is 16 dBW. The transmitting power of each pico BS is assumed to be 1/40 of the macro BS power. The path loss exponent is assumed to be 3.91 and 3.67 and the standard deviation of the log-normal shadow fading is 6 dB and 4 dB, respectively, for macro and pico BSs[2]. The total interference from the other clusters outside the target cluster is approximated by a complex Gaussian random vector with zero mean and variance $\sigma_I^2 = 141$. For simplicity, we assume that the channel temporal correlation for all users has the same correlation coefficient, i.e. the channels of all users in the network experience the same correlation; this assumption reduces the complexity of analysis and simulations. We consider 4 different correlation coefficients for our TC channels: a) $\rho = 0$, for uncorrelated channel gains; b) $\rho = 0.9998$, for pedestrian users with an average speed of 3 km/h, assuming that the

[2]The values of the path loss exponents and the shadow fading standard deviations used here for macrocells and picocells correspond to the Urban Macro (UMa) non-line-of-sight (NLoS) scenario and the Urban Micro (UMi) NLoS scenario, respectively, as found in [42].

transmission carrier frequency is 2 GHz and the scheduling interval is 1 ms; c) $\rho = 0.92$, for vehicular users with an average speed of 50 km/h; d) $\rho = 0.75$, for even higher speeds. For SA and PS algorithms, the simulation setup parameters are similar to what we assumed in Chapter 3. Different scheduling algorithms and their parameter settings are denoted by case numbers similar to Table. 3.2 in Chapter 3. The goal of this comparison is to find the parameter setup for SA and PS algorithms as well as to select the best user scheduling algorithm to obtain a solution close to exhaustive search when temporal correlation of channel gains are considered, using both MT and PF.

In Fig. 5.2(a) and 5.3(a), we compare the difference between the achievable sum rates obtained by different user scheduling methods considering different setup parameters with the exhaustive search, under different correlation coefficients, respectively, for both PF and MT metric. The Rayleigh fading component of channel gains are assumed TC samples with correlation coefficients of $\rho = 0$ and 0.75 in time, using $K_c = 6$. The comparison of the number of scheduled users in the cluster under the same conditions are presented in Fig. 5.2(b) and 5.3(b) for both metrics. The optimal ordered selection of users are found by exhaustive search in which achieves the maximum sum rate. Using i.i.d. and TC channel gains with $K_c = 6$, exhaustive search yields sum rates of 135.7 and 135.5 bits/s/Hz, respectively, for $\rho = 0$ and 0.75 with MT metric (similarly, with PF metric, the achieved rates respectively are given as 134.1 and 133.9; there is a drop off in sum rate using PF metric which is expected). These results depicts that the achievable sum rate of the system is reduced with channel correlation.

In PS algorithm with MT metric, RPS performs very close to exhaustive search with the maximum number of scheduled users; i.e. 5.98 users are served in average. Using PF metric, RPS algorithm also performs better than the other PS approaches, and it serves approximately the same number of users that MT serves. Thus, RPS approach is chosen to represent the proposed PS user schedul-

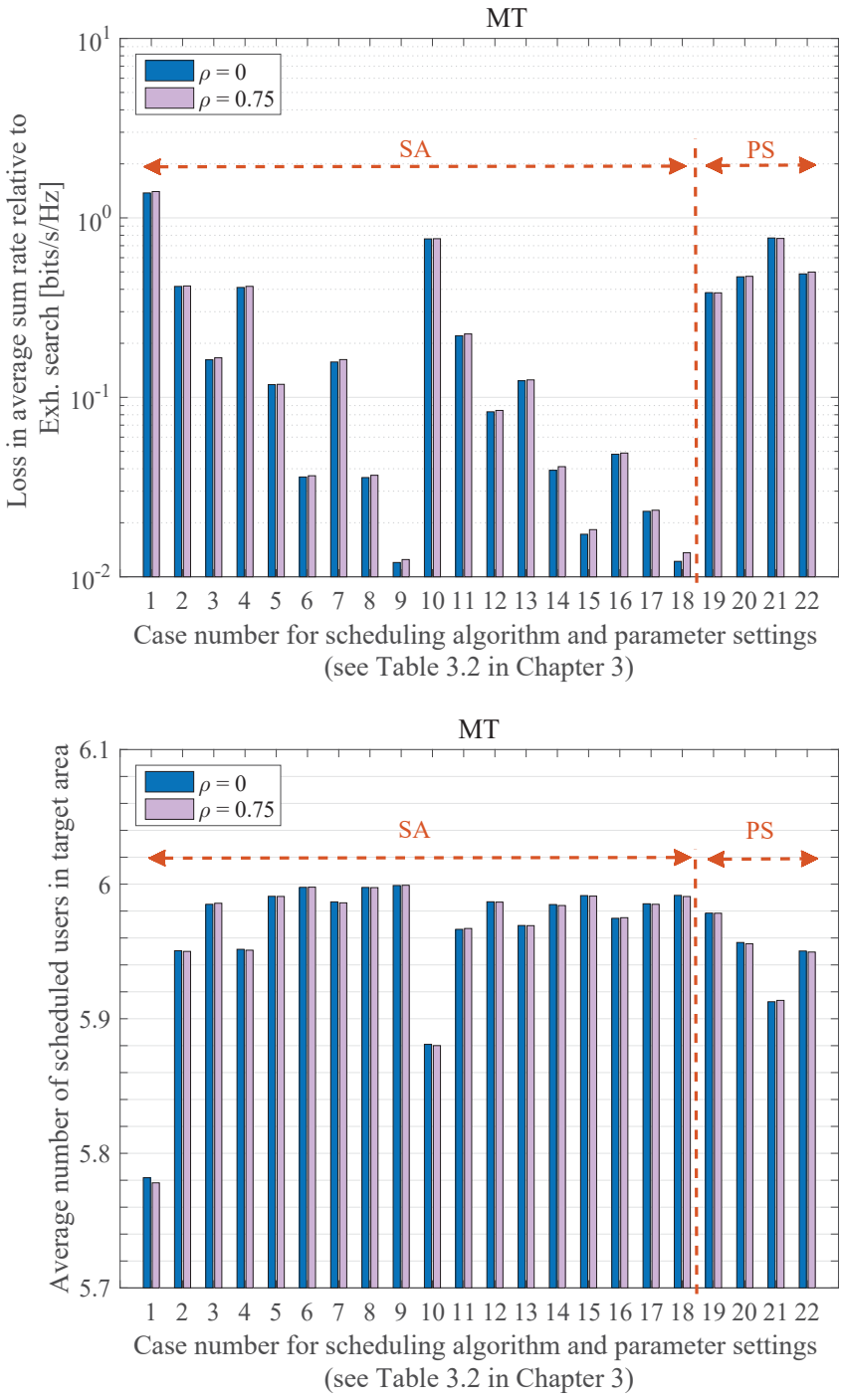

Figure 5.2: Comparison of (a) loss in target area sum rate, (b) target area average number of scheduled users, between different proposed PS and SA user scheduling algorithms, relative to exhaustive search, using SZF-DPC and different channel correlation coefficients, with MT metric; $M = N = 2$, $K_c = 6$, $P_t = 16$ dBW.

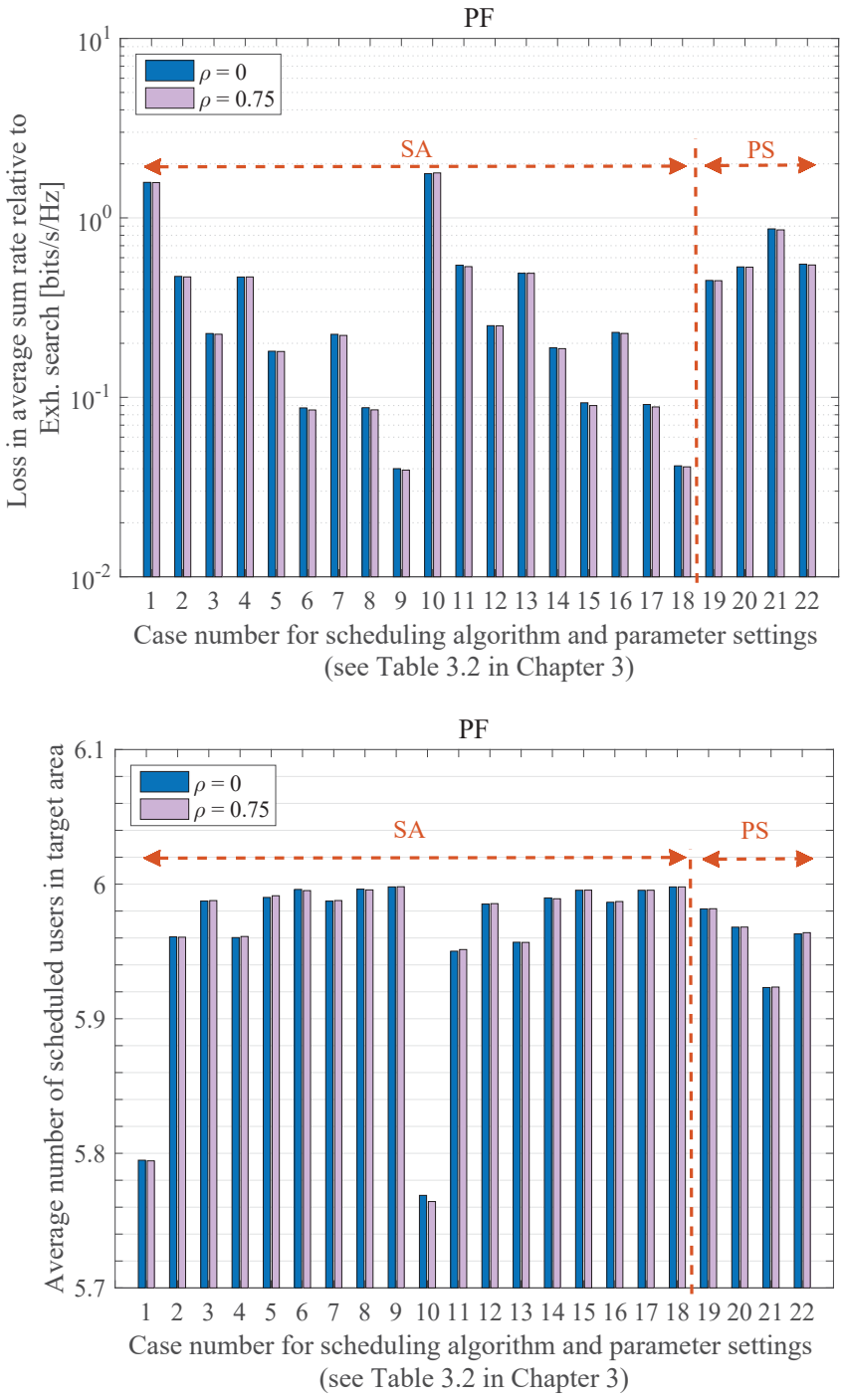

Figure 5.3: Comparison of (a) loss in target area sum rate, (b) target area average number of scheduled users, between different proposed PS and SA user scheduling algorithms, relative to exhaustive search, using SZF-DPC and different channel correlation coefficients, with PF metric; $M = N = 2$, $K_c = 6$, $P_t = 16$ dBW.

ing algorithm. The difference between achieved sum rates of different types of PS algorithm and exhaustive search are approximately the same with different channel correlation coefficient ρ, in both metrics. It means that our proposed user scheduling algorithms behave the same under uncorrelated and correlated channel assumptions; the proposed PS algorithm is resistant to the time correlation of the channel. However, the performance loss of PS algorithm is more sever when PF metric is used. For an example, performance loss of RPS (compared to exhaustive search), using PF metric, is 18.4% more than the loss of performance when MT is used, which means that RPS performs more close to the exhaustive search when MT scheduling metric is employed.

Assuming uncorrelated channel gains, SA algorithm with case numbers #9 and #18 (See setup methods in Chapter 3) perform better than other SA approaches with approximately similar performance loss compared to exhaustive search, using both metrics. These two cases can serve almost the same number of users (see Fig. 5.2(b) and 5.3(b)). Correlated time-varying channel samples cannot cause changes on performance loss on different SA approaches, using either MT or PF metrics; except for SA algorithm case #18 using MT metric. TC channel gains are able to slightly affect this approach; however, the performance loss of this circumstance is negligible (about 0.002 bits/s/Hz). Since cases #9 and #18 are performing very close to each other in different circumstances, and in line with results obtained in Chapter 3, we prefer to choose SA case #18 as our proposed SA user scheduling algorithm for all channel gain circumstances and/or performance metrics.

As it is clearly presented in Fig. 5.2 and 5.3, the correlation of the channel gains cannot affect the output of our algorithms very much and the results, which are confirmed in Chapter 3, are still valid with both metrics and in any correlation circumstances. Thus, RPS (case #19) as the selected PS algorithm and SA cases #18 achieve the highest sum rate in their own categories with the largest

number of served users. They, thus, can be chosen by any system with throughput maximization or fairness concerns.

In Fig. 5.4, the achievable sum rates of PS and SA user scheduling algorithms are compared with exhaustive search for different correlation coefficients of TC channel gains and various number of users in the cluster, with both MT and PF metrics. Using MT, both algorithms perform very close to optimal scheduling method, while SA performs closer to exhaustive search than PS. The performance loss of SA algorithm is quite negligible for all values of K_c and/or different temporal correlations of the channel. The loss of average achievable throughput between PS and exhaustive search are approximately the same when ρ is increasing; however, larger pool of users in a cluster reduces the performance loss (multi-user diversity gain). As an example, for $\rho = 0.92$ and $K_c = 4$, the performance loss of PS compared to exhaustive search is about 1.4%. When $K_c = 6$ with the same ρ, the performance loss is decreased as 0.3%, which means that PS user scheduling algorithm performs better when K_c is large, using MT metric.

When PF is considered, the performance loss of SA and PS algorithms compared to exhaustive search are, respectively, as 0.04% and 0.35% (for $K_c = 6$ and $\rho = 0.92$). This performance loss is increased by the increase in either K_c or ρ. Using PF, the performance loss of SA and PS algorithms when $K_c = 8$ and $\rho = 0.9998$ are, respectively, obtained as 0.4% and 1.3%. This analysis reveals that performance of both PS and SA algorithms, using PF schedulin metric, are affected by TC channels. The average achievable rates are decreased when channel temporal correlation is increased (also mentioned in [50]). When K_c increases, the achievable sum rate is increased because of employing multi-user diversity of MU-MIMO, and the SA algorithm achieves higher sum rates (closer to that exhaustive search obtains) compared to PS. The PS and SA algorithms are performing approximately close to the exhaustive search and increase in the temporal correlation of channel gains generally cannot change their performance

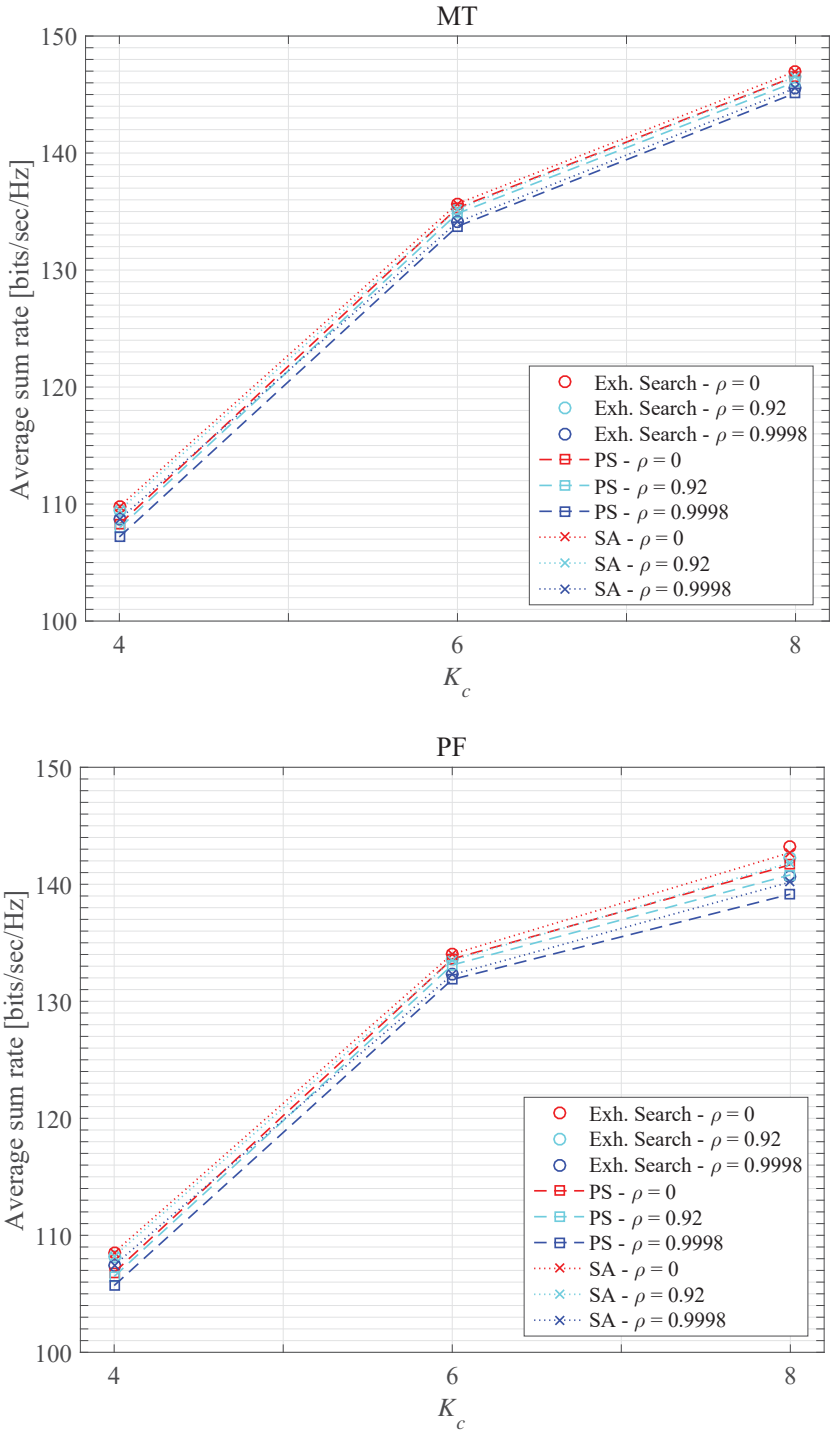

Figure 5.4: Comparison of average achievable sum rate with different correlation coefficients ρ and number of users in cluster K_c, for (a) MT (b) PF metrics, between PS and SA user scheduling algorithms and exhaustive search, using SZF-DPC.

compared to the optimal solution.

5.4 Developments on SA and PS User Scheduling Algorithms

For TC channels, we introduce updates to our proposed algorithms from [85] to take advantage of channel correlation during the search process in order to speed their convergence and to improve their achieved sum rates. In this paper, the scheduling metric s for all algorithms in [85] is now a WSR.

In Chapter 3, the PS user scheduling algorithm is described, and its pseudocode is presented in Algorithm 1. Temporal correlation in the users' channels can be exploited in the initialization part of the algorithm. In the original version of this algorithm, the initial set of particles is generated randomly from all possible vectors of ordered users. The particles then explore the search space to find a near-optimal solution. The best overall solution that has been found by all particles, named as $\hat{\mathbf{x}}^{[t]}$, is the output of the algorithm and becomes the scheduled vector of users. Having TC channels, it is quite possible that near-optimal solutions of two consecutive fading channel realizations would have some similarities with respect to each other; when two consecutive channel realizations are correlated, it would be logical that some of the same users should be scheduled at both of those times. Hence, the two solutions might be located near each other in the multi-dimensional search space. Under such circumstances, it is reasonable to initialize our next search such that the set of particles includes the outcome of the previous exploration; i.e. $\hat{\mathbf{x}}^{[t-1]}$ is used as one of the initial particle positions in the tth scheduling interval. The velocity of this particle is also imported from the previous interval. The rest of the PS algorithm operates the same as described in Chapter 3.

Using $\hat{\mathbf{x}}^{[t-1]}$ as an initial particle position at time t allows the algorithm to in-

Algorithm 4 Updated Particle Swarm User Scheduling Algorithm

for all time realizations starting from $t = 1$ **do**

Initialize: $stalled = \mathsf{false}$, $n = 1$, n_{max}, $n_{stalled}$, w_s, w_e, c_1, c_2, P, V_{max}, $w = w_s$, $s_{\hat{\mathbf{x}}_i(n)} = 0$, $s_{\hat{\mathbf{x}}(n)} = 0$; ρ;

if $\rho \neq 0$ and $t > 1$ **then**

$\mathbf{x}_1(1) = \hat{\mathbf{x}}^{[t-1]}$, $\mathbf{v}_1(1) = \hat{\mathbf{v}}^{[t-1]}$;

$\mathbf{x}_i(1) =$ a vector of up to L ordered users randomly chosen from $\mathcal{D}_s$, $\forall i \in \{2, \cdots, P\}$;

$\mathbf{v}_i(1) =$ a vector of L velocities for the elements of $\mathbf{x}_i(1)$, uniformly randomly chosen from $[-V_{max}, +V_{max}]$, $\forall i \in \{2, \cdots, P\}$;

else

$\mathbf{x}_i(1) =$ a vector of up to L ordered users randomly chosen from $\mathcal{D}_s$, $\forall i \in \{1, \cdots, P\}$;

$\mathbf{v}_i(1) =$ a vector of L velocities for the elements of $\mathbf{x}_i(1)$, uniformly randomly chosen from $[-V_{max}, +V_{max}]$, $\forall i \in \{1, \cdots, P\}$;

end if

{Similar to the Algorithm 1 in Chapter 3 ... }

return $\hat{\mathbf{x}}^{[t]}$, $\hat{\mathbf{v}}^{[t]}$

$t = t + 1$

end for

Algorithm 5 Updated Simulated Annealing User Scheduling Algorithm

for all time realizations starting from $t = 1$ **do**

{Similar to the Algorithm 3 in Chapter 3 ... }

if $\rho \neq 0$ and $t > 1$ **then**

$\hat{\mathbf{x}}_{Temp} = \mathbf{x}_{best}^{[t-1]}$; Compute $s_{\hat{\mathbf{x}}_{Temp}}$.

if $s_{\hat{\mathbf{x}}_{Temp}} > s_{\mathbf{x}_{best}^{[t]}}$ **then**

$\mathbf{x}_{best}^{[t]} = \mathbf{x}_{best}^{[t-1]}$;

$s_{\mathbf{x}_{best}^{[t]}} = s_{\hat{\mathbf{x}}_{Temp}}$;

end if

end if

return $\mathbf{x}_{best}^{[t]}$, $s_{\mathbf{x}_{best}^{[t]}}$

$t = t + 1$

end for

vestigate the neighbourhood of this solution; the other (still randomly initialized) particles still enable exploration of other parts of search space. If a near-optimal solution is located somewhere nearby $\hat{\mathbf{x}}^{[t-1]}$, the algorithm can more quickly focus its exploration on that area of the search space to find the best candidate solution. Thus, it will converge to this solution in fewer iterations on average. The pseudocode of this modification is presented in Algorithm 4 herein and named as the updated PS algorithm.

In Chapter 3, the SA algorithm with and without memory is proposed, and Algorithm 3 presents the corresponding pseudocodes. In this paper, we only consider the version with memory, SA-m, because of its higher achievable throughput. In each execution of the SA algorithm, it initializes a random vector of users chosen from the K_c users requesting service. Then, the algorithm proceeds according to Algorithm 3 until it converges. The ordered vector of users that gives the highest WSR found during the algorithm's iterations is named as $\mathbf{x}_{best}^{[t]}$. In the modification to the original SA algorithm, the algorithm then compares the achievable WSR $s_{\mathbf{x}_{best}}^{[t]}$ of $\mathbf{x}_{best}^{[t]}$ with that of the solution from the previous scheduling interval, but using the corresponding channel gains belonging to the current interval t. Let $\hat{\mathbf{x}}_{prev}$ be the solution of the previous interval, and $s_{\hat{\mathbf{x}}_{prev}}^{[t]}$ be the achievable WSR of $\hat{\mathbf{x}}_{prev}$ at interval t. If $s_{\hat{\mathbf{x}}_{prev}}^{[t]} > s_{\mathbf{x}_{best}}^{[t]}$, $\mathbf{x}_{prev}$ replaces $\mathbf{x}_{best}^{[t]}$ as the best solution. The updated SA algorithm then outputs its final best solution. Algorithm 3 from Chapter 3 is updated accordingly and the updated sections of pseudocode are provided here as Algorithm 5. The modifications let the algorithm incorporate the solution of the previous scheduling interval, after the channels are updated. If the previous solution is still a better candidate than the one found by the normal iterations, the algorithm will keep that prior solution as its output[3]. This process in essence improves the memory of SA algorithm, and

[3]We have also considered updating the SA algorithm wherein the previous solution is used in the initialization rather than near the end, similar to the process used in the updated PS algorithm. That attempt resulted in very negligible improvements in the achieved sum rates.

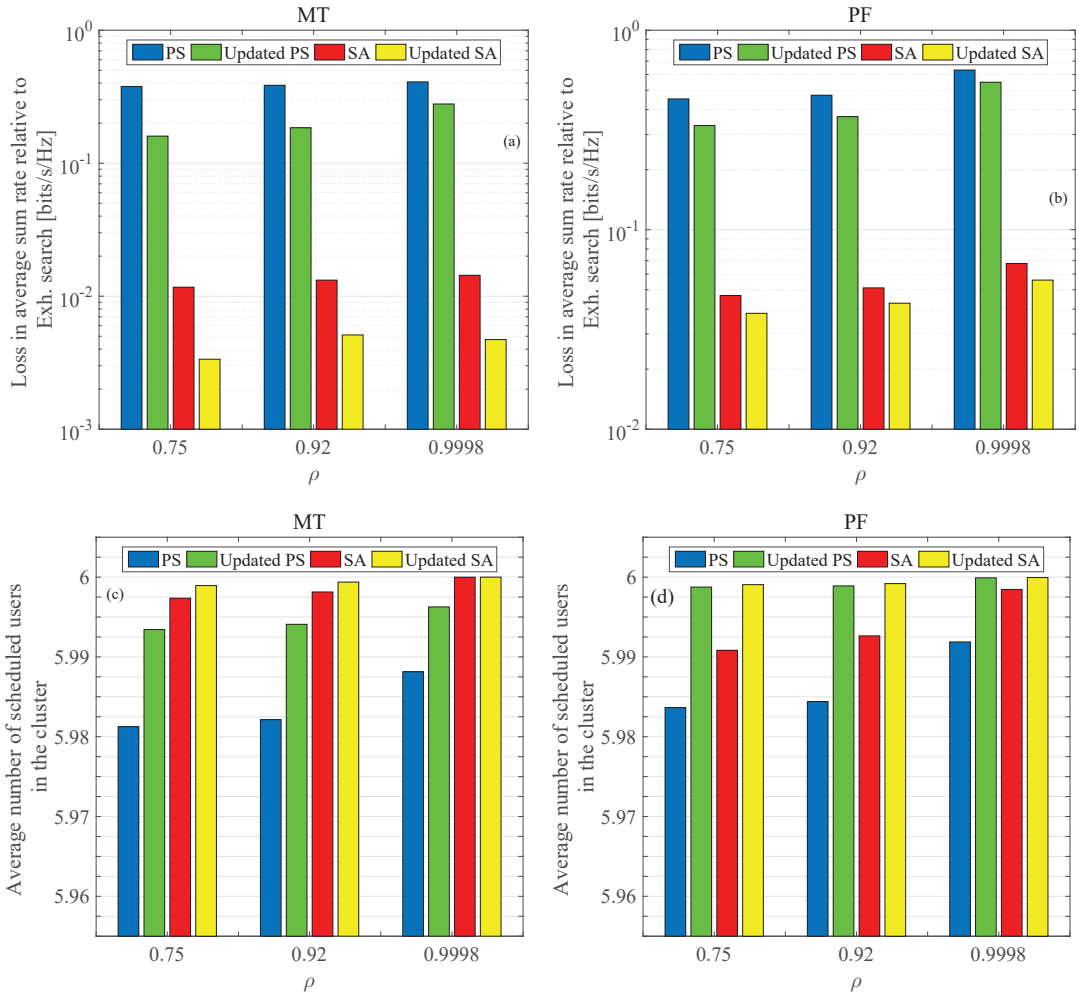

Figure 5.5: Comparison of loss in target area sum rate for (a) MT and (b) PF metrics, and average number of scheduled users for (c) MT and (d) PF, between different updated PS and SA user scheduling algorithms, relative to exhaustive search, using SZF-DPC, for correlated channel gains with various correlation coefficients ρs; $M = N = 2$, $K_c = 6$, $P_t = 16$ dBW.

consequently, the achievable throughput is improved, while the complexity has changed very little. The proposed method is named as the updated SA algorithm.

In Fig. 5.5, we compare the results of the updated PS and updated SA

This is because the algorithm gets stuck in the neighbourhood of the previous solution and can't sufficiently escape to search elsewhere. Note, there is only a single initialization point in the search space that can be modified for the SA algorithm , vs. 1 modified point out of P for PS.

algorithms with the original PS and SA algorithms from [85] when the correlation coefficient[4] $\rho = 0.75$, 0.92, and 0.9997 between scheduling intervals, for $K_c = 6$. The performance of both updated algorithms is improved, especially when the MT metric is used. Considering $\rho = 0.92$, the updated PS algorithm yields a loss in sum rate, relative to that provided by an exhaustive search, that is 52% and 22% smaller than the loss of the original PS algorithm, respectively for the MT and PF scheduling metrics. The improvements of the updated SA algorithm over the original are 61% and 16% for MT and PF, respectively. Thus, both updated algorithms under both metrics take advantage of the temporal correlation in the fading channels to enhance their performance significantly. The gain is smaller for PF because its near-optimal choice of users changes faster, due to the changing user weights each scheduling interval. We have also investigated the number of users served by all the algorithms. The number of served users with this developed PS and SA are compared in Fig. 5.5(c,d). Both algorithms can also serve more users with their developed version compared to their previous version introduced in Chapter 3; however, this improvement with updated PS is more considerable than others. With MT metric, the number of served users increases with updated PS and SA, respectively as 0.2% and 0.02%, when $\rho = 0.92$. Since the number of users are approximately 6 in all assumptions, we conclude that the system usually serves the maximum possible number of users that can be served simultaneously. In all four cases, they ended up almost always serving all 6 users requesting service. (The worst case, the original PS algorithm, served fewer users less than 2% of the time.)

In Fig. 5.6, we compare the updated PS and SA algorithms for different numbers of users K_c in the cluster, with the channel correlation coefficient assumed

[4]These values respectively correspond to a normalized time delay of {0.165, 0.093, 0.0056}. If we assume $\tau = 1$ ms and a carrier frequency of 2 GHz, then the mobile speed would be {89, 50, 3} km/h. In the case of $\rho = 0.75$, it starts to stretch our assumption of constant path loss and shadowing, so the value is mostly just for the purposes of comparing with a smaller ρ.

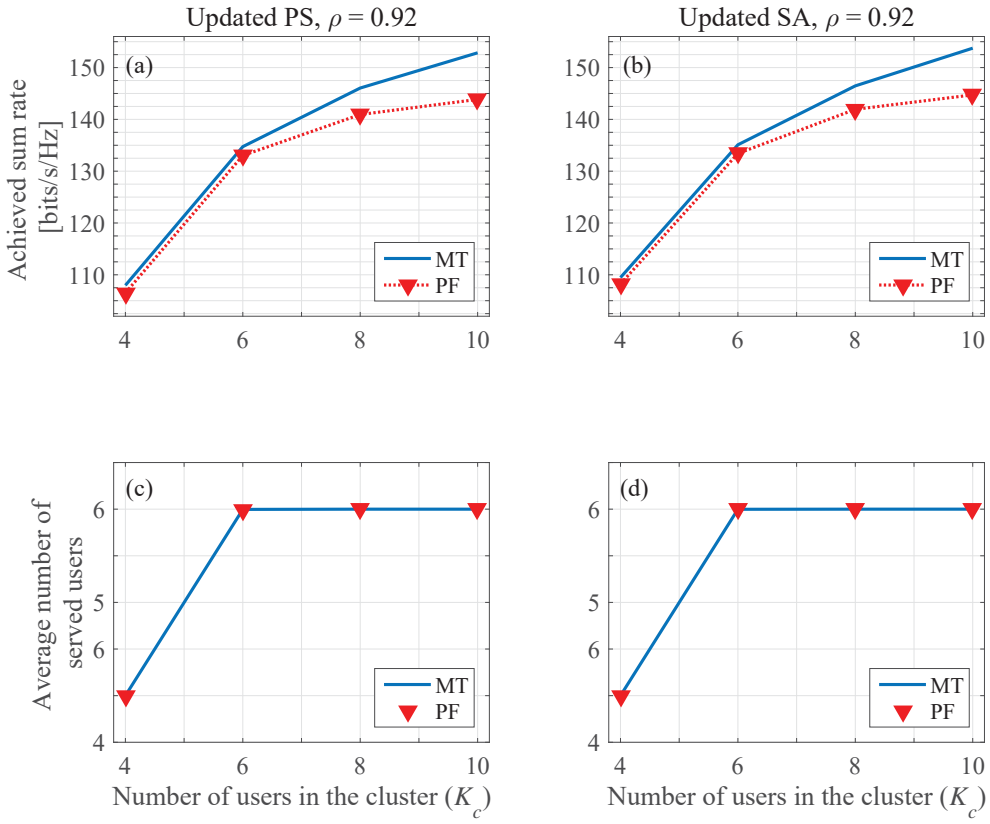

Figure 5.6: Comparison of achievable sum rate in the cluster with (a) updated PS and (b) updated SA, and average number of scheduled users with (c) updated PS and (d) updated SA, for MT and PF metrics and using SZF-DPC, for correlated channel gains with $\rho = 0.92$; $M = N = 2$, $P_t = 16$ dBW.

to be 0.92. As K_c increases, the average sum rate also increases, which is the result of multiuser diversity. The maximum possible number of simultaneously served users in our system model is $U = \lceil BN/M \rceil = 6$. The system consistently serves $\min(K_c, U)$ users in all cases. With K_c less than or equal to the maximum number of served users in a cluster, our proposed updated SA and PS behave similarly with both metrics (There is a small difference between the PF and MT in this range of K). The results of Fig. 5.6 have the same trends for other values of the correlation coefficient ρ, with increasing sum rates as ρ decreases, as expected.

We also consider the change in complexity of the updated PS and SA algorithms in terms of the number of iterations required to converge on a solution. Our investigation found that the updated PS algorithm converges faster than the original, with the convergence speed increasing the higher the value of ρ. For example, when $K_c = 10$ and $\rho = 0.75$, 0.92, and 0.9997, the updated PS algorithm on average finishes, respectively, 5.6%, 6.1%, and 8% faster than the unmodified PS algorithm from Chapter 3. However, the updated SA algorithm, despite the improved sum-rate results, did not yield a significant change in the speed of convergence vs. the original.

5.5 Conclusion

In this chapter, the downlink of a coordinated heterogeneous MIMO network with temporally correlated channel gains has been considered. We have investigated our already proposed reduced-complexity user scheduling algorithms, simulated annealing and particle swarm, under MT and PF metrics accounting for temporal correlation of channel gains, using SZF-DPC precoding. Each algorithm, again, is examined for the different parameters and options that can affect its performance (the best choices determined by simulations), using TC channels. Both algorithms perform quite close to an optimal exhaustive search in terms of achievable sum rate they provide. The simulations demonstrate that temporal correlation of fading channels cannot change the behaviour of our algorithms dramatically, and both PS and SA algorithms perform close to the optimal solution in different correlations.

We also proposed some updates to our PS and SA algorithms to improve their performance in terms of achievable sum rate and number of served users, by exploiting the characteristics of TC channels. Using the similarities between two channel samples in TC fading channels, the updated PS employs the solution of

previous transmission as one of the particles in its current search, and tries to find the best solution based on using the solution of previous scheduling interval. For SA, the algorithm explores the search space in each scheduling interval independently. Then, it compares the achieved sum rate of the ordered users obtained from the current search process with the sum rate of the scheduled users in the previous transmission interval, using the current channel gains. The solution with larger sum rate is returned as the output of the algorithm.

Our proposed updated PS and SA user scheduling algorithms are able to get advantage of channel correlation and perform even more closer to exhaustive search than their previous versions; updated PS is also able to find the solution even faster than before (lower complexity). The results have been examined for different ranges of the number of users and various correlation coefficients. As the number of users in the cluster increases, the achievable sum rate grows in different correlation coefficients. The outcomes of this chapter is independent of the choice of the precoding, and they can easily be extended to different precoding methods.

Chapter 6

Conclusion and Future Work

6.1 Summary of Contributions

In Chapter 3, we have proposed various simplified user scheduling algorithms for the downlink of a coordinated heterogeneous MIMO network, using reduced-complexity simulated annealing, particle swarm and hybrid greedy-particle swarm algorithms. The performance of the proposed algorithms has been evaluated under SZF-DPC, SZF, SZF(CGP), and BD precoding methods, and using maximum throughput scheduling metric. The algorithms were designed to be flexible with the number of users, type of BSs, etc. For each proposed algorithm, different parameters have been evaluated and determined by using simulations to find the best options to set up the algorithm. Thus, the algorithm achieves result very close to the exhaustive search. Computing the complexity for each of the proposed algorithms, we demonstrated how much its complexity has been reduced compared to the exhaustive search, while it performs quite close to an optimal exhaustive search in terms of the sum rate it provides. We have analysed SA algorithm with and without memory; we also proposed two methods for PS algorithm when particles are approaching borders of search space, reflective PS and modulo PS. SA with memory and reflective PS have been selected as the best solutions,

respectively, for SA and PS algorithms. Although HGPS outperforms greedy algorithms, PS algorithm outperforms HGPS algorithm in terms of achievable sum rate. A scheduled user vector obtained from PS algorithm usually includes more users with higher sum rate rather than the other two proposed methods. Increasing SNR and/or the number of users improve the achieved results of our proposed user scheduling algorithms.

We have proposed a rotating clustering scheme for two different 6-cell hexagonal-shaped and 3-cell clover-leaf-shaped layouts in Chapter 4. The performance of the proposed scheme under SZF-DPC precoding and the simulated annealing user scheduling algorithms considering both throughput maximization and proportionally fair scheduling metrics has been evaluated by simulation. For both metrics, the rotating clustering scheme outperforms fixed clustering and the achieved throughput of most users is improved. Using PF metric, the sum rate of the system is improved by using the cluster rotation scheme. The cluster rotation scheme is basically suggested to improve the performance of the cluster-edge users. By using rotating clustering scheme, the cluster-edge users are located in a position in the cluster where they can have a better SINR after the rotation. This scheme distributes the user rates more evenly through the network and the combination of PF, it gains better fairness and higher throughputs. We have demonstrated that the cluster rotation scheme improves the performance of 5th percentile users considerably with both layouts and saves those users from very low throughput (using PF) or service starvation (using MT). Also, the optimum speed of this rotation has been evaluated in this chapter. The results demonstrate that faster rotation performs better than slow rotation. However, there is an upper limit on increasing the rotation speed, and further increases (beyond that point) do not result in any notable gains in sum rate or per-user rate. The concept of cluster rotation can be extended to different network layouts from grid-like network to randomly deployed models. However, some of the network layouts gain more from

cluster rotation than other ones; comparison of the throughput improvements for our two assumed layouts demonstrates this.

In Chapter 5, we have, first, investigated our proposed user scheduling algorithms when the fading channel is temporally correlated. Temporally correlated channel models are important to model more practical scenarios in which the users' channel gains are not changing very quickly. Thus, the two consecutive channel samples may be strongly correlated. The performance of simulated annealing and particle swarm user scheduling algorithms have been evaluated under TC channel model, using SZF-DPC precoding method and MT and PF performance metrics. Our investigations demonstrate that the performance of our proposed SA and PS algorithms are resistant to channel correlation; both algorithms perform close to exhaustive search with uncorrelated and correlated channel gains. We have also intended to take advantage of the characteristic of TC channels and improve our SA and PS algorithms to perform better and faster. Thus, we proposed some updates to our already proposed simulated annealing and particle swarm user scheduling algorithms. In the updated version of each algorithm, the solution of the previous scheduling interval is employed in the process of exploration to find the suboptimal solution. Our simulations demonstrate that the proposed updates improve the performance of both PS and SA algorithms dramatically with respect to the optimal solution and they perform very close to exhaustive search: higher correlation, throughput closer to exhaustive search. The PS algorithm is also capable to perform even faster by employing the proposed updates. The modifications in SA and PS user scheduling algorithms improve our algorithms in order to employ better in real systems. The results of this chapter can be easily extended to any system model with any kind of precoding method.

6.2 Future Research Directions

- When investigating our proposed user scheduling algorithms, we assumed perfect channel state information and joint transmission network coordination scheme. In CoMP, perfect CSI of BSs to users channels needs to be available at all BSs at any transmission interval to achieve maximum throughput. In a practical system, however, the perfect CSI is not available at BSs, and acquiring accurate CSI is a big challenge in wireless systems [88]. Downlink CSI is imperfect for two main reasons. First, the estimation of the downlink channel, which is done by the user, usually involves some errors [89]. Second, user quantizes the estimated CSI and sends it back to the BSs using some limited CSI feedback technique, which results in furthur distortion of the CSI [89]. Hence, the CSI at BSs in a cellular system is an imperfect (inaccurate) version of its actual value. Another example of imperfect CSI could include a delay in the availability of channel state information at BSs. The imperfect CSI affects the achievable throughput of the MU-MIMO system, by affecting the precoding and user scheduling at BSs. Our proposed user scheduling algorithms employs perfect CSI to select the appropriate set of users to be served, with the performance very close to the optimal search. If this CSI is an imperfect version of the channel, the output of the algorithm is expected to be affected adversely and the algorithm might not perform that much close to exhaustive search. An important future direction of this work would be the consideration of this imperfect CSI in design and development of user scheduling algorithms. Also, the analysis of the overhead of centralized scheduling could be considered in future work.

- HetNets are considered to employ different cell sizes from macro cell to femto cells. The femto BS is usually user deployed BS with indoor applications

and its access can be varied from open access to all users in the network, closed access to just its own users, or a combination of both [95]. This restricted access also challenges all proposed RRM methods (user scheduling algorithms and/or clustering methods) even more, and requires to be considered as another future direction of this work.

- In this work, we have analysed all proposed user scheduling and cluster rotation methods under the assumption of joint processing network coordination. However, it is essential to investigate how these proposed algorithms and clustering methods will perform under different CoMP approaches. A combination of different coordination methods with various HetNet deployments challenges our proposed RRM techniques and is one more interesting future direction.

- Dynamic clustering is the optimal clustering approach to achieve high throughput in wireless network [37–40]. As we mentioned earlier, the processing and signaling overhead of this method is very complex. There are different dynamic clustering methods introduced in literature [37–40]. An open area of interest for future work would be the consideration of our proposed user scheduling algorithms combined with dynamic clustering method in HetNets. Joint scheduling and dynamic clustering should be considered, specifically when a BS belongs to two different clusters at the same time [41,96,97]; this imposes more challenges to our proposed RRM strategies.

Bibliography

[1] T. S. Rappaport, *Wireless Communications: Principles and Practice*. Prentice Hall PTR New Jersey, 2002.

[2] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.

[3] G. L. Stüber, *Principles of Mobile Communication*. Springer Science & Business Media, 2011.

[4] "Universal mobile telecommunications system (UMTS); technical specifications and technical reports for a UTRAN-based 3GPP system (release 8)," Tech. Rep. 3GPP TS 21.101 v8.4.0 (2012-03), 3rd Generation Partnership Project, Mar. 2012.

[5] "Universal mobile telecommunications system (UMTS); technical specifications and technical reports for a UTRAN-based 3GPP system (release 10)," Tech. Rep. 3GPP TS 21.101 v10.4.0 (2014-07), 3rd Generation Partnership Project, Jul. 2014.

[6] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update, 2015-2020." `http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html`. Accessed Feb. 3, 2016.

[7] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. Soong, and J. CharlieZhang, "What will 5G be?," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, 2014.

[8] İ. E. Telatar, "Capacity of multi-antenna Gaussian channels," *Eur. Trans. Telecommun.*, vol. 10, pp. 585–595, Nov.-Dec. 1999.

[9] G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless Pers. Commun.*, vol. 6, pp. 311–335, Mar. 1998.

[10] D. Gesbert, M. Kountouris, R. W. Heath, Jr., C.-B. Chae, and T. Sälzer, "Shifting the MIMO paradigm," *IEEE Signal Process. Mag.*, vol. 24, pp. 36–46, Sep. 2007.

[11] M. K. Karakayali, G. J. Foschini, and R. A. Valenzuela, "Network coordination for spectrally efficient communications in cellular systems," *IEEE Wireless Commun.*, vol. 13, pp. 56–61, Aug. 2006.

[12] H. Zhang and H. Dai, "Cochannel interference mitigation and cooperative processing in downlink multicell multiuser MIMO networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2004, pp. 222–235, Dec. 2004.

[13] M. Sawahashi, Y. Kishiyama, A. Morimoto, D. Nishikawa, and M. Tanno, "Coordinated multipoint transmission/reception techniques for LTE-advanced," *IEEE Wireless Commun.*, vol. 17, pp. 26–34, Jun. 2010.

[14] D. Lee, H. Seo, B. Clerckx, E. Hardouin, D. Mazzarese, S. Nagata, and K. Sayana, "Coordinated multipoint transmission and reception in LTE-advanced: deployment scenarios and operational challenges," *IEEE Commun. Mag.*, vol. 50, pp. 148–155, Feb. 2012.

[15] D. Gesbert, S. Hanly, H. Huang, S. Shamai Shitz, O. Simeone, and W. Yu, "Multi-cell MIMO cooperative networks: a new look at interference," *IEEE J. Sel. Areas Commun.*, vol. 28, pp. 1380–1408, Dec. 2010.

[16] M. Eslami, R. C. Elliott, W. A. Krzymień, and M. Al-Shalash, "Location-assisted clustering and scheduling for coordinated homogeneous and heterogeneous cellular networks," *Trans. Emerging Telecommun. Technol. (ETT)*, vol. 24, pp. 84–101, Jan. 2013.

[17] J. G. Andrews, "Seven ways that hetnets are a cellular paradigm shift," *IEEE Commun. Mag.*, vol. 51, pp. 136–144, March 2013.

[18] H. S. Dhillon, M. Kountouris, and J. G. Andrews, "Downlink MIMO HetNets: modeling, ordering results and performance analysis," *IEEE Trans. Wireless Commun.*, vol. 12, pp. 5208–5222, Oct. 2013.

[19] A. Damnjanovic, J. Montojo, Y. Wei, T. Ji, T. Luo, M. Vajapeyam, T. Yoo, O. Song, and D. Malladi, "A survey on 3GPP heterogeneous networks," *IEEE Wireless Commun.*, vol. 18, pp. 10–21, Jun. 2011.

[20] A. Ghosh, N. Mangalvedhe, R. Ratasuk, B. Mondal, M. Cudak, E. Visotsky, T. A. Thomas, J. G. Andrews, P. Xia, H. S. Jo, H. S. Dhillon, and T. D. Novlan, "Heterogeneous cellular networks: from theory to practice," *IEEE Commun. Mag.*, vol. 50, pp. 54–64, Jun. 2012.

[21] H.-S. Jo, Y. J. Sang, P. Xia, and J. G. Andrews, "Heterogeneous cellular networks with flexible cell association: a comprehensive downlink SINR analysis," *IEEE Trans. Wireless Commun.*, vol. 11, pp. 3484–3495, Oct. 2012.

[22] P. Xia, C.-H. Liu, and J. G. Andrews, "Downlink coordinated multi-point with overhead modeling in heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 12, pp. 4025–4037, Aug. 2013.

[23] G. Nigam, P. Minero, and M. Haenggi, "Coordinated multipoint hoint transmission in heterogeneous networks," *IEEE Trans. Commun.*, vol. 62, pp. 4134–4146, Nov. 2014.

[24] Y. Wu, Y. Cui, and B. Clerckx, "Analysis and optimization of inter-tier interference coordination in downlink multi-antenna HetNets with offloading," *IEEE Trans. Wireless Commun.*, vol. 14, pp. 6550–6564, Dec, 2015.

[25] W. Nie, F.-C. Zheng, X. Wang, W. Zhang, and S. Jin, "User-centric cross-tier base station clustering and cooperation in heterogeneous networks: rate improvement and energy saving," *IEEE J. Sel. Areas Commun.*, vol. 34, pp. 1192–1206, May 2016.

[26] S. Sigdel and W. A. Krzymień, "Simplified fair scheduling and antenna selection algorithms for multiuser MIMO orthogonal space-division multiplexing downlink," *IEEE Trans. Veh. Technol.*, vol. 58, pp. 1329–1344, Mar. 2009.

[27] S. Sigdel and W. A. Krzymień, "User scheduling for network MIMO systems with successive zero-forcing precoding," in *Proc. IEEE 72nd Veh. Technol. Conf. (VTC 2010-Fall)*, pp. 1–6, Ottawa, Canada, Sep. 2010.

[28] S. Sigdel, R. C. Elliott, W. A. Krzymień, and M. Al-Shalash, "Greedy and genetic user scheduling algorithms for multiuser MIMO systems with block diagonalization," in *Proc. IEEE 70th Veh. Technol. Conf. (VTC 2009-Fall)*, pp. 1–6, Anchorage, AK, USA, Sep. 2009.

[29] M. E. Aydin, R. Kwan, J. Wu, and J. Zhang, "Multiuser scheduling on the LTE downlink with simulated annealing," in *Proc. IEEE 73rd Veh. Technol. Conf. (VTC 2011-Spring)*, (Budapest, Hungary), pp. 1–5, May 2011.

[30] A. D. Dabbagh and D. J. Love, "Precoding for multiple antenna Gaussian broadcast channels with successive zero-forcing," *IEEE Trans. Signal Process.*, vol. 55, pp. 3837–3850, Jul. 2007.

[31] R. C. Elliott, S. Sigdel, and W. A. Krzymień, "Low complexity greedy, genetic and hybrid user scheduling algorithms for multiuser MIMO systems with successive zero-forcing," *Trans. Emerging Telecommun. Technol. (ETT)*, vol. 23, pp. 604–617, Nov. 2012.

[32] R. C. Elliott and W. A. Krzymień, "Downlink scheduling via genetic algorithms for multiuser single-carrier and multicarrier MIMO systems with dirty paper coding," *IEEE Trans. Veh. Technol.*, vol. 58, pp. 3247–3262, Sep. 2009.

[33] R. C. Elliott and W. A. Krzymień, "Improved and weighted sum rate maximization for successive zero-forcing in multiuser MIMO systems," *EURASIP J. Wireless Commun. Netw.*, vol. 2011, pp. 1–16, Oct. 2011.

[34] J. Gong, S. Zhou, Z. Niu, L. Geng, and M. Zheng, "Joint scheduling and dynamic clustering in downlink cellular networks," in *Proc. IEEE Glob. Telecom. Conf. (GLOBECOM 2011)*, (Houston, TX, USA), pp. 1–5, Dec. 2011.

[35] G. Caire and S. Shamai, "On achievable rates in a multi-antenna broadcast downlink," in *Proc. 38th Annu. Allerton Conf. Commun., Control and Comput.*, (Monticello, IL, USA), pp. 1188–1193, Oct. 2000.

[36] G. Caire and S. Shamai (Shitz), "On the achievable throughput of a multiantenna Gaussian broadcast channel," *IEEE Trans. Inf. Theory*, vol. 49, pp. 1691–1706, Jul. 2003.

[37] A. Papadogiannis, D. Gesbert, and E. Hardouin, "A dynamic clustering approach in wireless networks with multi-cell cooperative processing," in *Proc. IEEE Intl. Conf. Commun. (ICC'08)*, (Beijing, China), pp. 4033–4037, May 2008.

[38] J.-Y. Hwang, J. Kim, T. Kim, and Y. Han, "A periodic frequency band rotation scheme for multi-cell coordination clustering," *IEEE Commun. Lett.*, vol. 15, pp. 956–958, Sep. 2011.

[39] "Dynamic cell clustering for CoMP," Tech. Rep. 3GPP TSG RAN WG1 Meeting No.56 R1-090657, LG Electronics, Athens, Greece, Feb. 2009.

[40] N. Lee, R. W. Heath Jr., D. Morales-Jimenez, and A. Lozano, "Base station cooperation with dynamic clustering in super-dense cloud-RAN," in *Proc. 2013 IEEE Glob. Telecom. Conf. (GLOBECOM'13)- 1st Intl. Workshop Cloud-Process. Heterogeneous Mobile Commun. Netw.*, (Austin, TX, USA), pp. 784–788, Dec. 2013.

[41] H. S. Kang and D. K. Kim, "User-centric overlapped clustering based on anchor-based precoding in cellular networks," *IEEE Commun. Lett.*, vol. 20, no. 3, pp. 542–545, 2016.

[42] "Guidelines for evaluation of radio interface technologies for IMT-advanced," Tech. Rep. ITU-R M.2135-1, Int. Telecommun. Union, Dec. 2009.

[43] "Coordinated multi-point operation for LTE physical layer aspects (release 11)," Tech. Rep. 3GPP TR 36.819 v11.1.0 (2011-12), 3rd Generation Partnership Project, Dec. 2011.

[44] K. K. Mukkavilli, A. Sabharwal, E. Erkip, and B. Aazhang, "On beamforming with finite rate feedback in multiple-antenna systems," *IEEE Trans. Inf. Theory*, vol. 49, pp. 2562–2579, Oct. 2003.

[45] A. Narula, M. J. Lopez, M. D. Trott, and G. W. Wornell, "Efficient use of side information in multiple-antenna data transmission over fading channels," *IEEE J. Sel. Areas Commun.*, vol. 16, pp. 1423–1436, Oct. 1998.

[46] L. Zhang, L. Song, M. Ma, and B. Jiao, "On the minimum differential feedback for time-correlated MIMO Rayleigh block-fading channels," *IEEE Trans. Commun.*, vol. 60, pp. 411–420, Feb. 2012.

[47] K. Kim, I. Kim, and D. J. Love, "Utilizing temporal correlation in multiuser MIMO feedback," in *Proc. IEEE 42nd Asilomar Conf. Signals, Systems and Comput.*, (Pacific Grove, CA, USA), pp. 121–125, IEEE, Oct. 2008.

[48] K. Kim, T. Kim, D. J. Love, and I. H. Kim, "Differential feedback in codebook-based multiuser MIMO systems in slowly varying channels," *IEEE Trans. Commun.*, vol. 60, pp. 578–588, Feb. 2012.

[49] J. W. Wallace and M. A. Jensen, "Time-varying MIMO channels: measurement, analysis, and modeling," *IEEE Trans. Antennas and Propagation*, vol. 54, pp. 3265–3273, Nov. 2006.

[50] G. J. Byers and F. Takawira, "Spatially and temporally correlated MIMO channels: modeling and capacity analysis," *IEEE Trans. Veh. Technol.*, vol. 53, pp. 634–643, May 2004.

[51] A. J. Goldsmith and P. P. Varaiya, "Capacity of fading channels with channel side information," *IEEE Trans. Info. Theory*, vol. 43, pp. 1986–1992, Nov 1997.

[52] H. Huang, C. B. Papadias, and S. Venkatesan, *MIMO Communication for Cellular Networks*. Springer Science & Business Media, 2011.

[53] L. Liu, R. Chen, S. Geirhofer, K. Sayana, Z. Shi, and Y. Zhou, "Downlink MIMO in LTE-advanced: SU-MIMO vs. MU-MIMO," *IEEE Commun. Mag.*, vol. 50, pp. 140–147, February 2012.

[54] M. H. M. Costa, "Writing on dirty paper," *IEEE Trans. Inf. Theory*, vol. 29, pp. 439–441, May 1983.

[55] Q. H. Spencer, A. L. Swindlehurst, and M. Haardt, "Zero-forcing methods for downlink spatial multiplexing in multiuser MIMO channels," *IEEE Trans. Signal Process.*, vol. 52, pp. 461–471, Feb. 2004.

[56] Z. Shen, R. Chen, J. G. Andrews, R. W. Heath, Jr., and B. L. Evans, "Low complexity user selection algorithms for multiuser MIMO systems with block diagonalization," *IEEE Trans. Signal Process.*, vol. 54, pp. 3658–3663, Sep. 2006.

[57] M. Joham, W. Utschick, and J. A. Nossek, "Linear transmit processing in MIMO communications systems," *IEEE Trans. Signal Process.*, vol. 53, pp. 2700–2712, Aug. 2005.

[58] Q. H. Spencer, C. B. Peel, A. L. Swindlehurst, and M. Haardt, "An introduction to the multi-user MIMO downlink," *IEEE Commun. Mag.*, vol. 42, pp. 60–67, Oct. 2004.

[59] W. Yu and J. M. Cioffi, "Sum capacity of Gaussian vector broadcast channels," *IEEE Trans. Inf. Theory*, vol. 50, pp. 1875–1892, Sep. 2004.

[60] D. J. Mazzarese and W. A. Krzymień, "Scheduling algorithms and throughput maximization for the downlink of packet-data cellular systems with mul-

tiple antennas at the base station," *Wireless Pers. Commun.*, vol. 43, no. 2, pp. 215–260, 2007.

[61] T. Yoo and A. Goldsmith, "On the optimality of multiantenna broadcast scheduling using zero-forcing beamforming," *IEEE J. Sel. Areas Commun.*, vol. 24, pp. 528–541, Mar. 2006.

[62] X. Liu, E. K. P. Chong, and N. B. Shroff, "Opportunistic transmission scheduling with resource-sharing constraints in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 19, pp. 2053–2064, Oct. 2001.

[63] X. Liu, E. K. P. Chong, and N. B. Shroff, "Transmission scheduling for efficient wireless resource utilization with minimum-performance guarantees," in *Proc. IEEE 54th Veh. Technol. Conf. (VTC 2001-Fall)*, vol. 2, (Atlantic City, NJ, USA), pp. 824–828, IEEE, Oct. 2001.

[64] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, May 1983.

[65] M. M. Misiewicz, R. C. Elliott, K. R. Jacobson, and W. A. Krzymień, "Eigenmode scheduling via simulated annealing for multiuser MIMO downlink with successive zero-forcing precoding," in *Proc. 10th Int. Symp. Wireless Commun. Systems (ISWCS 2013)*, (Ilmenau, Germany), pp. 552–556, Aug. 2013.

[66] R. Kwan, M. E. Aydin, C. Leung, and J. Zhang, "Multiuser scheduling in high speed downlink packet access," *IET Commun.*, vol. 3, pp. 1363–1370, Aug. 2009.

[67] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, vol. 4, pp. 1942–1948, Nov.-Dec. 1995.

[68] J. Kennedy, R. C. Eberhart, and Y. Shi, *Swarm Intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers, 2001.

[69] R. Hassan, B. Cohanim, O. de Weck, and G. Venter, "A comparison of particle swarm optimization and the genetic algorithm," in *Proc. 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Mater. Conf.*, (Austin, TX, USA), pp. 1–13, Apr. 2005.

[70] H. Yong-qiang, X.-H. Li, K.-C. Yi, and X. Li, "Multiuser scheduling in downlink MIMO systems using particle swarm optimization," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC 2009)*, (Budapest, Hungary), pp. 1–5, Apr. 2009.

[71] J. Liu, X. Li, H. Ji, and Y. Tang, "Traffic-pairing scheme based on particle swarm optimization in downlink CoMP-MU-MIMO system," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC 2014)*, (Istanbul, Turkey), pp. 1779–1784, Apr. 2014.

[72] A. Bedekar, S. Borst, K. Ramanan, P. Whiting, and E. Yeh, "Downlink scheduling in CDMA data networks," in *Proc. 1999 IEEE Glob. Telecom. Conf. (GLOBECOM'99)*, vol. 5, pp. 2653–2657, IEEE, 1999.

[73] F. P. Kelly, A. K. Maulloo, and D. K. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *J. Oper. Res. Soc.*, vol. 49, pp. 237–252, Mar. 1998.

[74] P. Viswanath, D. N. C. Tse, and R. Laroia, "Opportunistic beamforming using dumb antennas," *IEEE Trans. Inf. Theory*, vol. 48, pp. 1277–1294, Jun. 2002.

[75] A. Jalali, R. Padovani, and R. Pankaj, "Data throughput of CDMA-HDR a high efficiency-high data rate personal communication wireless system," in *Proc. IEEE 51st Veh. Technol. Conf. (VTC 2000-Spring)*, vol. 3, (Tokyo, Japan), pp. 1854–1858, May 2000.

[76] H. Purmehdi, R. C. Elliott, and W. A. Krzymień, "Simulated annealing user scheduling for coordinated heterogeneous MIMO networks," in *Proc. 2012 Asilomar Conf. Signals, Systems and Comput.*, (Pacific Grove, CA, USA), pp. 1157–1161, Nov. 2012.

[77] A. Ghosh, J. Zhang, J. G. Andrews, and R. Muhamed, *Fundamentals of LTE*. Prentice-Hall, 2010.

[78] "Spatial channel model for multiple input multiple output (MIMO) simulations (release 12)," Tech. Rep. 3GPP TR 25.996 v12.0.0 (2014-09), 3rd Generation Partnership Project, Sep. 2014.

[79] J. Zhang, R. Chen, J. G. Andrews, A. Ghosh, and R. W. Heath, Jr., "Networked MIMO with clustered linear precoding," *IEEE Trans. Wireless Commun.*, vol. 8, pp. 1910–1921, Apr. 2009.

[80] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[81] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD, USA: The John Hopkins Univ. Press, 4th ed., 2013.

[82] H. Sanvicente-Sánchez and J. Frausto-Solís, "A method to establish the cooling scheme in simulated annealing like algorithms," in *Proc. 2004 Int. Conf. Comput. Sci. and Its Applicat. (ICCSA 2004), Springer Berlin Heidelberg*, vol. 3045, (Assisi, Italy), pp. 755–763, May 2004.

[83] P. Baracca, F. Boccardi, and N. Benvenuto, "A dynamic clustering algorithm for downlink CoMP systems with multiple antenna UEs," *EURASIP J. Wireless Commun. Netw.*, vol. 2014, pp. 1–14, Jan. 2014.

[84] V. K. N. Lau, "Optimal downlink space-time scheduling design with convex utility functions — multiple-antenna systems with orthogonal spatial multiplexing," *IEEE Trans. Veh. Technol.*, vol. 54, pp. 1322–1333, Jul. 2005.

[85] H. Purmehdi, R. C. Elliott, and W. A. Krzymień, "Reduced-complexity user scheduling algorithms for coordinated heterogeneous MIMO networks," *IEEE Trans. Veh. Technol.*, vol. 65, pp. 6184–6203, Aug. 2016.

[86] D.-T. Lee, "On k-nearest neighbor voronoi diagrams in the plane," *IEEE Trans. Comput.*, vol. C-31, pp. 478–487, Jun. 1982.

[87] K. Huang, R. W. Heath Jr., and J. G. Andrews, "Limited feedback beamforming over temporally-correlated channels," *IEEE Trans. Signal Process.*, vol. 57, pp. 1959–1975, May 2009.

[88] G. Caire, N. Jindal, M. Kobayashi, and N. Ravindran, "Multiuser MIMO achievable rates with downlink training and channel state feedback," *IEEE Trans. Inf. Theory*, vol. 56, pp. 2845–2866, Jun. 2010.

[89] M. Kobayashi, N. Jindal, and G. Caire, "Optimized training and feedback for MIMO downlink channels," in *Proc. IEEE Inf. Theory Workshop Netw. Inf. Theory (ITW 2009)*, (Volos, Greece), pp. 226–230, Jun. 2009.

[90] M. Kobayashi, G. Caire, and N. Jindal, "How much training and feedback are needed in MIMO broadcast channels?," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT 2008)*, pp. 2663–2667, Toronto, ON, Canada, Jul. 2008.

[91] B. Makki, T. Svensson, and M. Debbah, "Reinforcement-based data transmission in temporally-correlated fading channels: Partial CSIT scenario," in *Proc. IEEE 11th Intern. Symp. Wireless Commun. Sys. (ISWCS)*, (Barcelona, Spain), pp. 176–181, IEEE, Aug. 2014.

[92] R. H. Clarke, "A statistical theory of mobile-radio reception," *Bell System Technical Journal*, vol. 47, no. 6, pp. 957–1000, 1968.

[93] D. J. Young and N. C. Beaulieu, "The generation of correlated Rayleigh random variates by inverse discrete Fourier transform," *IEEE Trans. Commun.*, vol. 48, no. 7, pp. 1114–1127, 2000.

[94] J. I. Smith, "A computer generated multipath fading simulation for mobile radio," *IEEE Trans. Veh. Technol.*, vol. 3, no. 24, pp. 39–40, 1975.

[95] G. de la Roche, A. Valcarce, D. López-Pérez, and J. Zhang, "Access control mechanisms for femtocells," *IEEE Commun. Magazine*, vol. 48, no. 1, pp. 33–39, 2010.

[96] J. Kim, H.-W. Lee, and S. Chong, "Virtual cell beamforming in cooperative networks," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1126–1138, 2014.

[97] M. Hong, R. Sun, H. Baligh, and Z.-Q. Luo, "Joint base station clustering and beamformer design for partial coordinated transmission in heterogeneous networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 2, pp. 226–240, 2013.

Appendix A

Location of Picocells in Coordinated HetNets

Assume a pico BS located in the coverage area of a macro BS, which is also named as target area. The equal signal to interference ratio (SIR) rule defines the border of pico vs. macro BSs (consequently the location of the pico BS inside the coverage area of the macro BS). Assuming coverage area of an arbitrary macro BS and for any location of this area, the SIR of a pico BS is defined as

$$SIR_p = \frac{P_{r_p}}{I + P_{r_m}}, \tag{A.1}$$

where P_{r_p} is the received power from the corresponding pico BS, P_{r_m} is the received power from the macro BS, and I is the interference from all transmitting pico and macro BSs outside the target area. Similarly, we can define the SIR from macro BS at the same point by switching P_{r_p} and P_{r_m}. The borders are defined at those points in which $SIR_p = SIR_m$. Using (A.1), the equality is obtained by $P_{r_m} + 1/2 = \pm(P_{r_p} + 1/2)$. This system of equations has two sets of results: $P_{r_m} = P_{r_p}$ or $P_{r_m} = -(I + P_{r_p})$. The latter is only possible if at least one of the P_{r_m} or P_{r_p} are negative. Since P_{r_m}, P_{r_p}, and I are all positive values, the only

valid solution is $P_{r_m} = P_{r_p}$. This means that the border of pico BS is defined at those locations in which the received signal from both pico and macro BSs are equal.

In general, the received power, in dB, is defined as [2]

$$P_r = P_t + PL + A + \chi + \zeta, \tag{A.2}$$

where P_t is the transmitting power in dBW, PL is the path loss, A is the antenna pattern of BS, χ is the log-normal shadow fading with zero mean and standard deviation σ_χ, and ζ is the small-scale Rayleigh fading component, where it is i.i.d complex Gaussian random variable with zero mean and unit standard deviation. The average received power conditioned to a specific location (x, y) is given as

$$\mathbb{E}\left(P_r|(x,y)\right) = P_t + PL(x,y) + A(x,y). \tag{A.3}$$

The $PL(x, y)$ and $A(x, y)$ are defined as constant values at any (x, y). The path loss is obtained as $PL(x, y) = -10\alpha \log_{10} d(x, y) + g$, where $d(x, y)$ is the distance between the transmitter and the location of point (x, y), g is a constant value representing parameters of the antenna and transmission, and α is the path loss exponent. The antenna pattern of the macro BS cell for a 120° directional antenna is defined as [78]

$$A = -\min\left(12\Big(\frac{\hat{\theta}}{\theta_{3dB}}\Big)^2, A_m\right), \tag{A.4}$$

where $\hat{\theta}$ is the angle between the location of (x, y) with macro BS's main beam, θ_{3dB} is 3dB beamwidth in degrees which equals to 70°, and A_m is the maximum

attenuation and equals to 20 dB[1]. The equation in (A.4) can be revised as

$$A = \begin{cases} -12\left(\frac{\hat{\theta}}{\theta_{3dB}}\right)^2, & \hat{\theta} < \theta_c, \\ -A_m, & \hat{\theta} > \theta_c, \end{cases} \tag{A.5}$$

where θ_c is the cut-off angle; i.e. the min function switches to either of its limits in this angle. Using (A.5), θ_c is given as $\theta_{3dB}\sqrt{\frac{A_m}{12}}$, and it is calculated for 120°–cell and 60°–cell, respectively, as 90.37° and 48.46°. Since any location in the coverage area of a cell has $\hat{\theta} < \theta_c$, the first row in (A.5) is just used in the calculations.

- Assume a cell of a 6-cell layout with its two pico BSs which are presented in Fig. A.1. The pico BSs are located on an imaginary circle with radius of $c = R(\sqrt{3}/2 - \gamma)$ around the center of hexagon. The macro BS is also located on the center of this hexagon site, and γ is the distance between the border of the site and the imaginary circle, in the direction of the antenna's main beam (see Fig. A.1). The macro BS is equipped with 60° directional antenna. The location of any pico BS can be defined by using c and β, where $\beta =$ is the angle[2] between the pico BS and the main beam of the macro BS antenna. In this layout, pico BSs are evenly spaced around the macro BS, such that the angle between them is 30°, and their angle from the macro boresight is fixed to 15°, i.e. $\beta = 15°$. So, they are defined accordingly as $L_1 = c\sin\beta$ and $L_2 = c\cos\beta$, where L_1 and L_2 are depicted in Fig. A.1.

  An arbitrary point (x, y) is laid inside the macro cell. Its distance from macro and pico BSs are, respectively, defined with d and r; the angle between this point and the main beam of the macro BS is defined by $\hat{\theta}$. By using polar coordinates instead of Cartesian, (x, y) can be replaced by (r, θ)

[1]Similarly, for the macro BS cell with a 60° directional antenna, the θ_{3dB} and A_m are, respectively, given as 35° and 23 dB.

[2]The direction of positive angles here is counter-clockwise and depicted by θ in Fig.A.1.

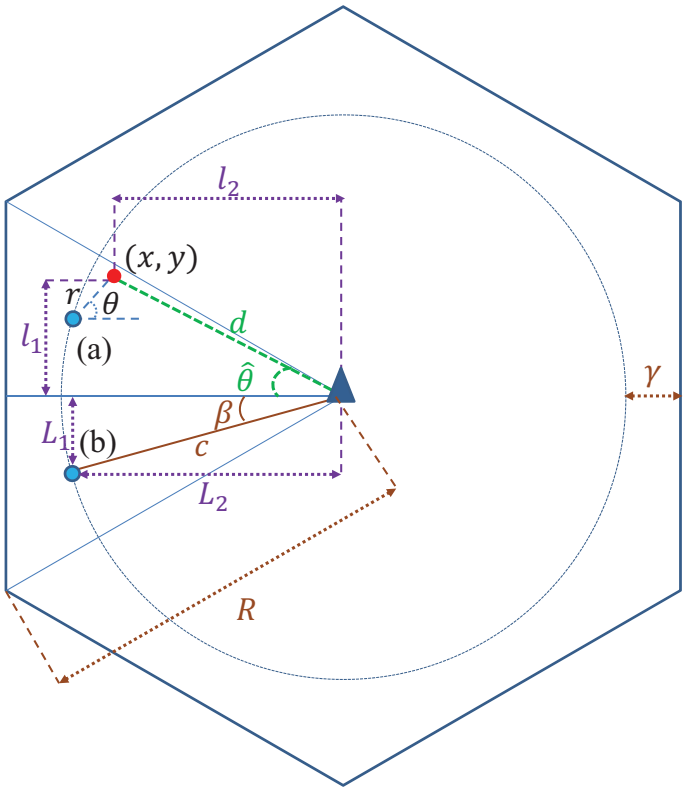

Figure A.1: Location of the point (x, y) on an arbitrary macro cell of a 6-cell layout.

(see the pico BS (a) in Fig. A.1); the distance of this location from macro BS is obtained as

$$d_a(r,\theta) = \sqrt{l_1^2 + l_2^2} = \sqrt{c^2 + r^2 + 2rc(\sin\beta\sin\theta + \cos\beta\cos\theta)} \qquad \text{(A.6)}$$

where $l_1 = L_1 + r\sin\theta$ and $l_2 = L_2 + r\cos\theta$. With similar calculations, the distance of arbitrary point from pico BS b is obtained as $d_b(r,\theta) = \sqrt{c^2 + r^2 + 2rc(\cos\beta\cos\theta - \sin\beta\sin\theta)}$. The angles $\hat{\theta}_a$ and $\hat{\theta}_b$ are also calculated as

$$\hat{\theta}_a(r,\theta) \;=\; \cos^{-1}\frac{c\cos\beta + r\cos\theta}{d_a(r,\theta)} \qquad \text{(A.7)}$$

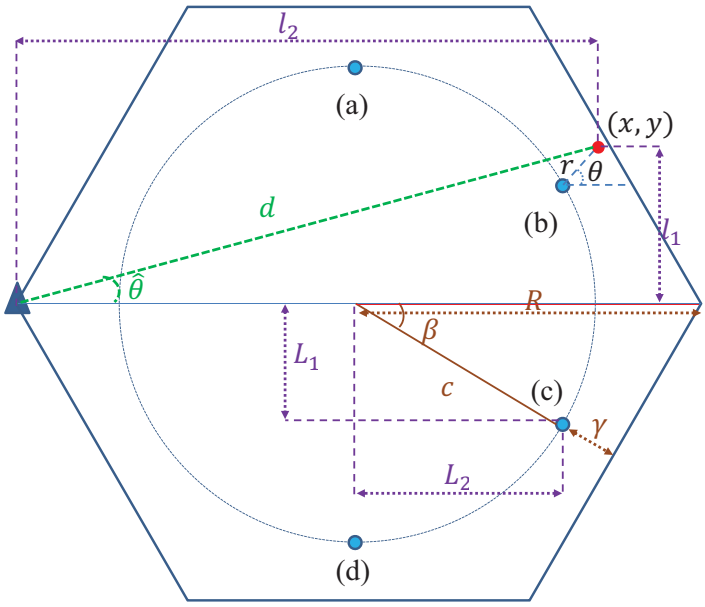

Figure A.2: Location of the point (x, y) on an arbitrary macro cell of a 3-cell layout.

$$\hat{\theta}_b(r, \theta) \quad = \quad \cos^{-1} \frac{c \cos \beta + r \cos \theta}{d_b(r, \theta)} \tag{A.8}$$

where $\hat{\theta}_a$ is the angle between the point (r, θ) and the direction of the main beam of macro BS, using the pico BS (a) as the origin of coordination. Using the symmetry in the macro site, similar formulas are achieved for other pico BSs in different cells.

- Assume a cell coverage area of a 3-cell layout, which the macro BS is located on the mid-right corner of the hexagon and it is equipped with a 120° directional antenna which its main beam is directed towards the mid-left corner of the hexagon. There are four pico BSs located in the middle of four sides of the hexagon with the distance of γR from the corresponding side, as depicted in Fig. A.2. In fact, the pico BSs are again located on an imaginary circle as depicted in Fig. A.2. An arbitrary point (x, y) is assumed inside the macro cell (as presented in Fig. A.2), which its distance

from a closest pico BS is r. The distance between this point and the macro BS is again defined by d and the angle between this point and the main beam of the macro BS is presented by $\hat{\theta}$. By translating Cartesian coordinates into polar coordinates, i.e. the (x, y) coordinates into (r, θ), all distances and angles can be described with the same variables.

After a few calculations, the distance between the point (x, y) and the macro BS and its corresponding angle, i.e. d and $\hat{\theta}$ respectively, are given as

$$\begin{aligned}
d_a(r,\theta) &= \sqrt{(L_1 + r\sin\theta)^2 + (R + r\cos\theta)^2} \\
\hat{\theta}_a(r,\theta) &= \cos^{-1}\frac{R + r\cos\theta}{d_a(r,\theta)} \\
d_b(r,\theta) &= \sqrt{(L_1 + r\sin\theta)^2 + (L_2 + R + r\cos\theta)^2} \\
\hat{\theta}_b(r,\theta) &= \cos^{-1}\frac{R + L_2 + r\cos\theta}{d_b(r,\theta)} \\
d_c(r,\theta) &= \sqrt{(L_1 - r\sin\theta)^2 + (L_2 + R + r\cos\theta)^2} \\
\hat{\theta}_c(r,\theta) &= \hat{\theta}_b(r,\theta) \\
d_d(r,\theta) &= \sqrt{(L_1 - r\sin\theta)^2 + (R + r\cos\theta)^2} \\
\hat{\theta}_d(r,\theta) &= \hat{\theta}_a(r,\theta)
\end{aligned}$$

Thus, the conditional average received power from macro and pico BSs at point ir, θ are, respectively, given as

$$\mathbb{E}_t(P_{r_m}|(r,\theta)) = W_m - 10\alpha_m \log_{10} d(r,\theta) - \min\left(12\Big(\frac{\hat{\theta}(r,\theta)}{\theta_{3dB}}\Big)^2, A_m\right) \quad \text{(A.9)}$$

$$\mathbb{E}_t(P_{r_p}|(r,\theta)) = W_p - 10\alpha_p \log_{10} r \quad \text{(A.10)}$$

where $W_m \triangleq P_{t_m} + g_m$ and $W_p \triangleq P_{t_p} + g_p$. The border at this points happens when $\mathbb{E}_t(P_{r_m}|(r,\theta)) = \mathbb{E}_t(P_{r_p}|(r,\theta))$. The equations in (A.9) and (A.10) should be examined over all (r, θ), where $r \in (0, R]$ and $\theta \in (0, 2\pi]$, and the origin of

coordinates is centred at pico BS's location.

To find the parameters like W_m, W_p, α_m, and α_p, the link budget formula is applied. The link budget at the distance of d from the transmitter is obtained as the summation of P_t, transmitting Antenna gain, connector loss, receiving antenna gain, minus the UMa path loss model, the noise floor, and receiving noise figure, where P_t is in dBW, transmitting Antenna gain and connector loss are given as 17 dBi, receiving antenna gain is 0 dBi, and the sum of noise floor and receiving noise figure equals -127 dBm/Hz. The UMa channel model for NLoS scenario, in dB, is given as [42, 78]

$$\begin{aligned} PL_{UMa} = {} & 161.04 - 7.1\log_{10} W + 7.5\log_{10} h - \left(24.37 - 3.7\big(\frac{h}{h_{BS}}\big)^2\right)\log_{10} h_{BS} \\ & + (43.42 - 3.1\log_{10} h_{BS})(\log_{10} d - 3) + 20\log_{10} f_c \\ & - \left(3.2\big(\log_{10}(11.75 h_{UT})\big)^2 - 4.97\right) \end{aligned}$$

where W is the street width, h is average building height, h_{BS} is height of BS, d is the distance of the user to the BS, f_c is the transmitting frequency, and h_{UT} is the average user's height. These values are set as $W = 20$ m, $h = 20$ m, $h_{BS} = 25$ m, $h_{UT} = 1.5$ m, $f_c = 2$ GHz. Thus, the above PL_{UMa} is simplified as $PL_{UMa} = 19.5653 - 39.0864\log_{10} d$, and the link budget is rewritten as

$$P_{t_m} + 17 - \big(19.5653 - 39.0864\log_{10} d\big) + 127 \tag{A.11}$$

Similarly, the link budget for pico BS is obtained by replacing the values of the transmitting antenna gain and the noise floor, and the receiving noise figure, respectively, with 5 dBi and -127 dBW. The UMi NLoS scenario is given as $PL_{UMi} = 36.7\log_{10} d + 22.7 + 26\log_{10} f_c = 36.7\log_{10} d + 30.5268$. Thus, the link

budget is given as

$$P_{t_p} + 5 - \big(30.5268 - 36.7 \log_{10} d\big) + 127 \tag{A.12}$$

Hence, according to the link budget formulas, W_m and W_p are calculated, respectively, as $P_{t_m} + 17 - 19.5653 + 127$ and $P_{t_p} + 5 - 30.5268 + 127$. The path loss coefficient α for macro and pico BSs are, respectively, as 3.90864 (approximated by 3.91) and 3.67. The macro BS transmitting power, P_{t_m}, is set 40 W, the pico transmits 1/40 of that[3]. In both system models, the ISD of two macro BSs is assumed as $\sqrt{3}$ km or 1732 m.

In a cell of a 6-cell layout, the contour of the borders of two sample pico coverage areas are illustrated in Fig. A.3. As Fig. A.3(a-d) depict, the γ is gradually decreased from 17% to 9.5% of radius R, thus, the contours are moved towards the borders. When the γ is $0.17R$, the borders of pico coverage area are quite laid inside the macro cell; however, a small part of the border area of the macro cell is still not covered properly by pico coverage area. As this figure presents, the $\gamma = 0.12R$ is very good choice to set the picos overlaid on macro cell, while their covered area is touching the macro cell border and not passing it (if the coverage area of the pico BS passes the border of the macro BS, it will interfere with the coverage area of other pico BS located in neighbouring macro cell).

The contours of the border of pico BSs for a macro cell of 3-cell layout is presented in Fig. A.4. In Fig. A.4(a), the distance of the picos from the borders of macro cell, γ, is assumed about $0.17R$. Since the coverage area of pico BSs are overlapping with other pico BSs from adjacent macro cell, we need to move the pico BSs a little inside the macro cell to avoid interfering with other pico BS from other macro cell. The pico BSs are gradually moved toward the center of

[3]It should be noticed that the transmitting powers should be translated into dBW and then be applied into equations (A.11) and (A.12); i.e. the P_{t_m} is changed to 16 dBW, etc.

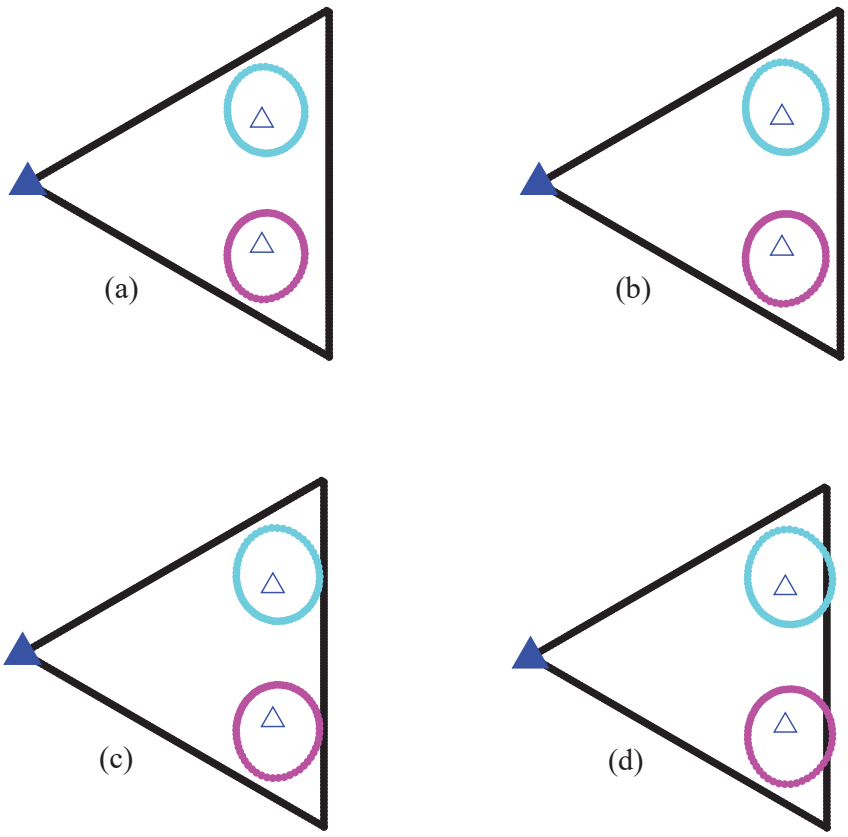

Figure A.3: Contours of borders of pico BSs in a cell of 6-cell layout with ISD = $\sqrt{3}$ km, $P_{t_m} = 40P_{t_p} = 40W$; distance of pico BSs from the border of macro cell, γ, is (a) $0.17R$, (b) $0.145R$, (c) $0.12R$, (d) $0.095R$.

the macro cell (the pico BSs are located in distances of $0.2R$, $0.25R$, and $0.3R$ of the borders of macro cell) and the border of pico coverage area is presented in Fig. A.4(b-c). As the results illustrate, the best option is in Fig. A.4(c) with $\gamma = 0.25R$, which covers the area close to the border of macro cell as much as possible.

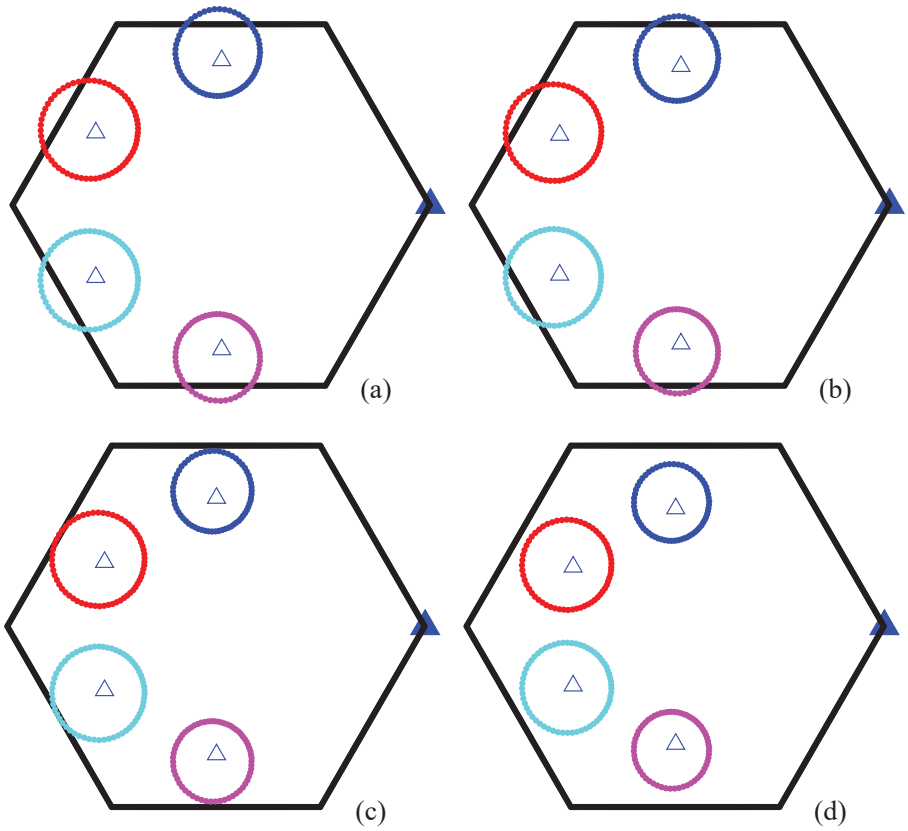

Figure A.4: Contours of borders of pico BSs in a cell of 3-cell layout with ISD $= \sqrt{3}$ km, $P_{t_m} = 40P_{t_p} = 40W$; distance of pico BSs from the borders of macro cell, γ, is (a) $0.17R$, (b) $0.2R$, (c) $0.25R$, (d) $0.3R$.

Appendix B

Channel Models in Simulation

In cellular systems, a user is usually surrounded by local scatters and the received signal arrives to the user's antenna from many different directions. It is assumed that the BS transmitted signal is a 2D plane waveform[1] in which the plane waves arrive at the user's antenna from all azimuth directions with equal probability. This propagation channel model is known as Clarke's 2D isotropic scattering model, which is simple and commonly used model in the literature [3,92–94]. One approach to generate the correlated channel samples, for simulation purpose, is described in [93, 94] and it is well known as IDFT method. In this method, the correlated Rayleigh random variations are generated by inverse discrete Fourier transformation via computer simulations. To generate the coloured random sequence, two independent sequences of Gaussian random variables, namely as $A_1[\hat{n}]$ and $A_2[\hat{n}]$, are generated; the elements of each sequences are i.i.d random variables distributed as $\mathcal{N}(0, 1/\sqrt{2})$, and both sequences are independent from each other $\forall k$. Then, these two are filtered with an appropriate filter named as $F[\hat{n}]$ and passed from an IDFT, then the output presents the time-domain sequence of correlated samples of complex faded envelope $x[\hat{n}]$. The coloured

[1]It is a valid assumption, if the distance between user and BS is sufficiently large.

sequence is given as

$$x[\hat{n}] = \text{IDFT}\{F[\hat{n}]A_1[\hat{n}] - jF[\hat{n}]A_2[\hat{n}]\}_{\hat{n}=0}^{\hat{N}-1}. \tag{B.1}$$

where $\hat{N}$ is the number of samples. The $F[\hat{n}]$ filter is chosen to approximate the spectrum of the desired signal and to shape the power spectrum of a white Gaussian noise. The choice of appropriate coefficients for $F[\hat{n}]$ corresponds to the approximation of the autocorrelation function of discrete signal $x[\hat{n}]$. The autocorrelation function is given as $J_0(2\pi\hat{f}_d|\hat{n}|)$, where $\hat{f}_d$ is the maximum normalized Doppler frequency and defined as $\hat{f}_d = f_d T_s$, where f_d is the maximum Doppler frequency and T_s is the sampling period [3, 92]. The autocorrelation can be translated as the correlation coefficient of the signal, i.e. $\rho = J_0(2\pi\hat{f}_d|\hat{n}|)$.

The process to find the coefficients of $F[\hat{n}]$ filter is described in [3, 93]. The normalized Doppler spectrum $S(f)$ is given as

$$S(f) = \begin{cases} \frac{a_0}{\sqrt{1-(f/f_d)^2}} & |f| \leq f_d \\ 0 & otherwise \end{cases} \tag{B.2}$$

where a_0 is defined as $\frac{1.5}{\pi f_d}$ [93]. To obtain the $F[\hat{n}]$, this continuous spectrum should be sampled at frequencies of $f_{\hat{n}} = \frac{\hat{n}}{\hat{N}T_s}$, $\hat{n} = 0, \cdots, \hat{N}-1$. The coefficients of $F[\hat{n}]$ filter should have the structure given as [3, 93]

$$F[\hat{n}] = \begin{cases} 0 & \hat{n} = 0 \\ \frac{G[\hat{n}]}{\sqrt{2}} & \hat{n} = 1, 2, \cdots, \frac{\hat{N}}{2} - 1 \\ G[\hat{n}] & \hat{n} = \frac{\hat{N}}{2} \\ \frac{G[\hat{N}-\hat{n}]}{\sqrt{2}} & \hat{n} = \frac{\hat{N}}{2} + 1, \cdots, \hat{N} - 1 \end{cases} \tag{B.3}$$

where $G[\hat{n}]$ is defined as [93, 94]

$$G[\hat{n}] = \begin{cases} 0 & \hat{n} = 0 \\ \sqrt{\frac{a_0}{\sqrt{1-(\frac{\hat{n}}{\hat{N}\hat{f}_d})^2}}} & \hat{n} = 1, 2, \cdots, k_m - 1 \\ \sqrt{a_0(\frac{\pi}{2} - \arctan\left(\frac{k_m - 1}{\sqrt{2k_m - 1}}\right))} & \hat{n} = k_m \\ 0 & otherwise \end{cases} \tag{B.4}$$

k_m is defined as $\lfloor \hat{f}_d \hat{N} \rfloor$ [3, 93]. It worth to mention that $G[0] = 0$ satisfies the zero-mean criterion [3, 93]. Thus, the filter $F[\hat{n}]$ in (B.3) is revised as

$$F[\hat{n}] = \begin{cases} 0 & \hat{n} = 0 \\ \sqrt{\frac{a_0/2}{\sqrt{1-(\frac{\hat{n}}{\hat{N}\hat{f}_d})^2}}} & \hat{n} = 1, 2, \cdots, k_m - 1 \\ \sqrt{\frac{a_0}{2}(\frac{\pi}{2} - \arctan\left(\frac{k_m - 1}{\sqrt{2k_m - 1}}\right))} & \hat{n} = k_m \\ 0 & \hat{n} = k_m + 1, \cdots, \hat{N} - k_m - 1 \\ \sqrt{\frac{a_0}{2}(\frac{\pi}{2} - \arctan\left(\frac{k_m - 1}{\sqrt{2k_m - 1}}\right))} & \hat{n} = \hat{N} - k_m \\ \sqrt{\frac{a_0/2}{\sqrt{1-(\frac{\hat{N}-\hat{n}}{\hat{N}\hat{f}_d})^2}}} & k = \hat{N} - k_m + 1, \cdots, \hat{N} - 1 \end{cases} \tag{B.5}$$

Using $F[\hat{n}]$ from (B.5) into (B.1), the complex-value samples of the coloured channel gains $x[\hat{n}]$ with correlation coefficient ρ are generated.

In this thesis, we generate signals according to the process mentioned above. The auto correlation of real and imaginary parts of the generated samples and the cross correlation of them are presented in Fig. B.1(a) for correlation coefficient zero and 0.75. In this simulation, we generate million of samples, i.e. $\hat{N} = 10^6$. We also compare the autocorrelation of the generated coloured signal with respect to the continuous correlation function, which are obtained via Bessel function of

first order (i.e. $J_0(\cdot)$). As this figure illustrates, there is very good compatibility between simulated signals and the analytic autocorrelation function.

Uncorrelated channel samples, taken from this generated wide-sense stationary random process, have the autocorrelation function value of zero with normalized sample separation $\hat{f}_d\hat{n} \approx 0.38$ (i.e. with normalized time separation $f_d\tau \approx 0.38$). According to this figure, each two consecutive samples with $\hat{n} = 1$ and maximum normalized Doppler frequency of $\hat{f}_d$ have correlation coefficient ρ. For instance, with $\hat{f}_d = 0.164$, two generated samples with sample distance of $\hat{n} = 1$ are correlated with $\rho = 0.75$, while two samples with $\hat{n} = 2.32$ are uncorrelated (i.e. $\hat{f}_d\hat{n} = 0.38$). Fig. B.1(b) illustrates the autocorrelation of generated samples with other correlation coefficients. We also examine two practical scenarios in a wireless network: first, we assume $\hat{f}_d = 0.005$, which is the maximum normalized Doppler frequency of pedestrian with walking speed less than 3 km/h; second, we assume $\hat{f}_d = 0.093$, which is assumed for a vehicle with speed of 50 km/h, on average[2]. For the sake of simplicity in our simulations, we assume that path loss and shadow fading do not change over the period of our simulations[3]. The correlation coefficients of two samples separated by $\hat{n} = 1$ for these two speeds are obtained, respectively, as $\rho = 0.9998$ and 0.92. Similarly, if two consecutive samples have the correlation coefficient of $\rho = 0.5$ (i.e. $\hat{f}_d = 0.24$), the sampling time for a pedestrian user should be 43.4 ms, while the channel of the vehicle is sampled every 2.6 ms.

[2]In this work, the carrier frequency and the scheduling interval are considered to be 2 GHz and 1 ms, respectively.

[3]For each drop of users, we simulate 2000 scheduling intervals and this assumption can still be valid at the above mentioned speeds. This period of simulation is the equivalent of about 2 seconds for an LTE-A system. The distance between a BS and a user with the maximum assumed speed, 50 km/h, would change by about 27 m, which does not affect the large-scale fading considerably. Thus, we can assume it is constant in our simulation for each drop of users.

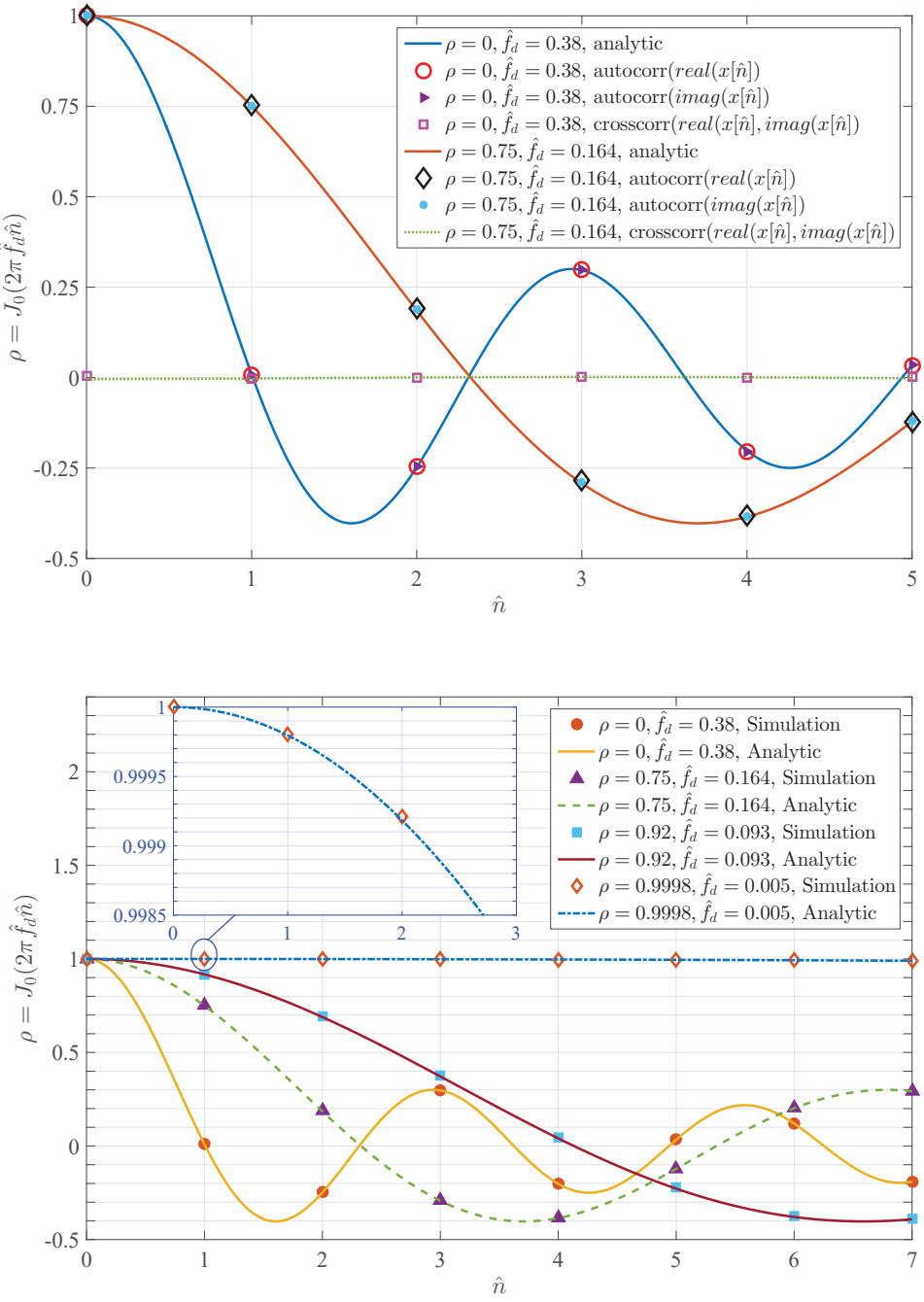

Figure B.1: Correlation coefficient of generated channel samples for simulations (a) autocorrelation and cross correlation of real and imaginary parts of generated samples for $\rho = 0$ and 0.75 (b) autocorrelation of generated samples for $\rho = 0$, 0.75, 0.92, and 0.9998.