MACHINE LEARNING BASED RESOURCE ALLOCATION FOR MASSIVE MIMO SYSTEMS

A Thesis Submitted to
the Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

in Electroincs and Communication Engineering

by
Hüseyin Can SEVGİ

June 2023
İZMİR

We approve the thesis of **Hüseyin Can SEVGİ**

Examining Committee Members:

Prof. Dr. Berna ÖZBEK
Department of Electrical and Electronics Engineering,
Izmir Institute of Technology

Prof. Dr. Barış ATAKAN
Department of Electrical and Electronics Engineering,
Izmir Institute of Technology

Assoc. Prof. Dr. Eylem ERDOĞAN
Department of Electrical and Electronics Engineering,
Istanbul Medeniyet University

14 June 2023

Prof. Dr. Berna ÖZBEK
Supervisor, Department of Electrical and Electronics Engineering,
Izmir Institute of Technology

Prof. Dr. Mustafa A. ALTINKAYA
Head of the Department of Electrical
and Electronics Engineering,
Izmir Institute of Technology

Prof. Dr. Mehtap EANES
Dean of the Graduate School of
Engineering and Sciences,
Izmir Institute of Technology

ACKNOWLEDGMENTS

First and foremost, I would like to express my gratitude and thankfulness to my supervisor Prof. Dr. Berna ÖZBEK for her attentive guidance, valuable suggestions and contributions throughout my master education and thesis research.

In addition, I present my respect and love to my dear mother Nermin SEVGİ, my dear father İdris SEVGİ and my dear sister Elif Naz SEVGİ. They were always stand behind me and made numerous sacrifices especially on the most difficult days of the undergraduate and graduate studies. Also, I would like to give special thanks to my fiance Buse ALTUNKAYA for her endless support, encouragement and tolerance. Her moral support during the writing of this thesis was very valuable.

Finally, I would like to thank to my managers and teammates at AirTies for their support and understanding during my master education.

ABSTRACT

MACHINE LEARNING BASED RESOURCE ALLOCATION FOR MASSIVE MIMO SYSTEMS

Cell-free massive MIMO communication systems is a promising technology that uses access-points(APs) deployed throughout the coverage area instead of usual cellular systems with centralized BS to serve multiple users simultaneously. By exploiting the large number of antennas and adopting advanced signal processing techniques, cell-free massive MIMO can mitigate inter-user interference and enhance the overall system performance. Optimal power allocation plays a crucial role in maximizing the spectral and energy efficiency of wireless networks. By intelligently allocating transmit power to different users, a balance between maximizing the system throughput and minimizing the total energy consumption can be achieved. In addition, user-centric clustering(UCC) is also a key technique to improve the performance of cell-free massive MIMO systems. This technique aims to pair user equipments (UEs) with appropriate APs to facilitate efficient resource allocation and interference management.

In this thesis, cell-free mMIMO communication system is investigated through user-centric clustering and power allocation. The power allocation optimization problem is formulated to maximize energy efficiency of cell-free mMIMO systems and solved by using interior-point algorithm. User-centric clustering algorithm is proposed by disabling the non-master APs that are serving only one user. This additional feature aims to reduce total power consumption of the system without sacrificing the advantages of the cell-free mMIMO communication systems. Additionally, we propose a machine learning(ML) approach to reduce the computation time required for power allocation optimization. Through extensive simulations, we demonstrate the effectiveness of the proposed algorithms in achieving significant gains in spectral and energy efficiency in cell-free massive MIMO systems. The results highlight the importance of optimal power allocation and user-centric clustering to design an efficient cell-free mMIMO systems through machine learning approach.

ÖZET

MASSIVE MIMO SİSTEMLERİ İÇİN MAKİNE ÖĞRENMESİ TABANLI KAYNAK TAHSİSİ

Hücresiz masif çok-girişli çok-çıkışlı haberleşme sistemleri günümüzde kullanılan baz istasyonu merkezli hücresel haberleşme sistemlerinin aksine, kapsama alanına dağıtılmış erişim noktaları kullanarak çok sayıda kullanıcıya eş zamanlı olarak hizmet verebilen gelecek vaat eden bir teknolojidir. Bu sistemler çok sayıda anten ve ileri düzey sinyal işleme tekniklerini kullanarak kullanıcılar arası girişimi azaltabilir ve sistem performansını arttırabilir. Optimal güç tahsisi enerji ve spektral verimlilik maksimizasyonunda çok önemli rol oynamaktadır. Her kullanıcı için uygun iletim gücü tahsisi yapılarak sistem veri hızı maksimizasyonu ile toplam enerji kullanımı arasındaki denge sağlanabilir. Sistem performansını arttırmanın bir diğer anahtar tekniği de kullanıcı-merkezli kümelemedir. Kullanıcı-merkezli kümelemenin amacı kullanıcıları uygun erişim noktaları ile eşleştirerek verimli kaynak tahsisi ve girişim yönetimi sağlamaktır.

Bu tezde, hücresiz masif çok-girişli çok-çıkışlı haberleşme sistemlerinin performansı kullanıcı-merkezli kümeleme ve güç tahsisi üzerinden incelenmektedir. Enerji vermimliliğini maksimize etmek için bir güç tahsisi optimizasyon problemi formüle edilmekte ve bu problem iç-nokta algoritması ile çözülmektedir. Ayrıca, ana erişim noktası olmayan ve az sayıda kullanıcıya hizmet veren erişim noktalarını devre dışı bırakmayı öneren yeni bir kullanıcı-merkezli kümeleme algoritması önerilmektedir. Bu ek özellik, hücresiz masif çok-girişli çok-çıkışlı sistemlerin avantajını kaybetmeden toplam kullanılan enerjiyi azaltmayı amaçlamaktadır. Ayrıca, güç tahsisi optimizasyonu için gereken hesaplama sürelerini azaltmak amacıyla makine öğrenmesi tabanlı güç tahsisi yaklaşımı önerilmektedir. Geniş simülasyon ve analizler ile, bu tezde hücresiz masif çok-girişli çok-çıkışlı haberleşme sistemlerinde önerilen metodolojilerin spektral ve enerji verimliliğinde önemli kazanımlar sağladığını göstermektedir. Sonuçlar optimal güç tahsisi, kullanıcı-merkezli kümeleme ve makine öğrenmesi yaklaşımının hücresiz masif çok-girişli çok-çıkışlı haberleşme sistemlerinde kullanılmasının önemini vurgulamaktadır.

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

LIST OF ABBREVIATIONS

APG	Accelerated Proximal Gradient
AP	Access Point
AP	Access Point
ASIC	Application Specific Integrated Circuit
AWGN	Additive White Gaussian Noise
BS	Base Station
CDF	Cumulative Distribution Function
CDF	Cumulative Distribution Function
CPU	Central Processing Unit
CSI	Channel State Information
CSI	Channel State Information
DNN	Deep Neural Network
DL	Downlink
EE	Energy Efficiency
ELU	Exponential Linear Unit
FPGA	Field Programmable Gate Array
MAE	Mean Absolute Error
MIMO	Multiple Input Multiple Output
mMIMO	Massive Multiple Input Multiple Output
MINLP	Mixed Integer Non-Linear Programming
ML	Machine Learning
MRT	Maximum Ratio Transmission
MSE	Mean Square Error
PA	Power Allocation
RF	Radio Frequency
RMSE	Root Mean Square Error
RZF	Regularized Zero Forcing
ReLU	Rectified Linear Unit
SE	Spectral Efficiency
SGD	Stochastic Gradient Descent
SINR	Signal to Interference and Noise Ratio
SOCP	Second Order Cone Programming
UCC	User-Centric Clustering
UE	User Equipment

ZF	Zero Forcing
2G	Second Generation
3GPP	3rd Generation Partnership Project
3G	Third Generation
4G	Fourth Generation
5G	Fifth Generation

CHAPTER 1

INTRODUCTION

In order to meet the increasing demands for higher data rate and higher number of connected devices across large geographical areas, one of the most promising methods is to employ massive multiple-input-multiple-output (MIMO) communication systems, where each base station (BS) is equipped with multiple antennas to simultaneously serve many users in the same time-frequency resource block. Massive MIMO offers high throughput, reliability and increased energy efficiency. Traditionally, large areas are covered by dividing them into cells and performing communication within each cell. However, cellular systems have several drawbacks as high computational complexity, limited spatial diversity and most commonly, boundary effects which can be simply explained as UEs located at cell edges experience high path loss and interference from other BSs or APs, resulting in a low signal-to-interference-and-noise ratio (SINR) and poor performance.

To overcome these issues, cell-free massive MIMO is one of a promising technique; in which, the transmitting antennas are distributed over the area as APs where each is capable of serving multiple users simultaneously and UEs are served by group of selected APs, called cooperation clusters. According to (Zhang et al., 2019), cell-free massive MIMO has lower deployment costs, better uniform coverage capability which eliminates the boundary effects, higher energy efficiency, and more spatial diversity compared to centralized massive MIMO. There are two types of cell-free massive MIMO as centralized and distributed manner. In centralized systems, the precoding and power allocation(PA) are performed in a central processing unit (CPU), while in distributed systems, these processes are performed in each AP locally. Centralized systems have better performance since the channel state information (CSI) between all UEs and all APs are available at the CPU. However, the backhaul load to connect each AP to the CPU and transfer CSI , precoding and power allocation coefficients is higher than the distributed systems. In distributed systems, each AP has only local CSI and performs precoding and power allocation locally. Therefore, the backhaul load is lower than the centralized systems. However, the performance of distributed systems is poorer than the centralized systems due to the lack of global CSI at each AP.

In this thesis, we focus on centralized cell-free massive MIMO systems with the objective of maximizing energy efficiency while taking into account users' experience and limitations of APs so that the thesis' outcomes could be applicable in real-world scenarios. The maximization of energy efficiency will be achieved by finding optimal UE-AP pairs and optimal power allocation coefficients for each pair. The UE-AP pairs are determined based on user-centric clustering algorithm and the power allocation coefficients are determined by solving the optimization problem. The results are provided in terms of energy efficiency. The thesis proposes a machine learning approach with the objective of reducing the computational time of user-centric clustering and power allocation. The proposed machine learning approach based on deep neural networks (DNN) determines UE-AP pairs and power allocation coefficients. This approach is compared with the analytical approach in terms of energy efficiency and computational time.

Throughout the thesis, the centralized cell-free massive MIMO communication system is assumed to have perfect CSI at the CPU. The users are randomly located over outdoor environment and their positions are assumed to be stationary.

The thesis is organized as follows:

- Chapter 2 gives literature review and provides a detailed information of the system model and channel characteristics of cell-free massive MIMO networks in the context of downlink systems. Furthermore, this chapter presents the proposed UE-AP pairing algorithm with extensive performance evaluations of the various precoding schemes along with UE-AP pairing schemes, from the perspective of energy efficiency. In addition, we define energy efficiency optimization problem and constraints for downlink cell-free massive MIMO communication system. Finally, the performance evaluations of the UE-AP pairing algorithms are presented and compared in terms of energy efficiency maximization performance and computational complexity.

- Chapter 3 introduces a machine learning approach for maximizing energy efficiency. The chapter begins with a brief background information of ML and DNNs. Then, the proposed machine learning approach along with the network architecture is presented for user-centric clustering and power allocation. Finally, the chapter presents the performance evaluation of the proposed machine learning approach and compares it with the analytical approach in terms of energy efficiency performance and computational time requirements.

- Chapter 4 concludes the thesis and provides a summary of the interpretation of the research outcomes.

CHAPTER 2

ENERGY EFFICIENCY MAXIMIZATION IN CELL-FREE MASSIVE MIMO

This chapter provides an in-depth analysis of the concept of energy efficiency maximization in centralized cell-free massive MIMO communication through a downlink system. The chapter gives the literature review about cell-free massive MIMO , the system model and the channel model, which includes path loss, shadowing, and multi-path effects. Additionally, the linear precoding schemes and user-centric clustering algorithms that have been examined in the literature for centralized cell-free massive MIMO are reviewed. Then, the proposed user-centric clustering algorithm is provided and a power allocation problem for energy efficiency optimization is formulated based on power consumption and minimum acceptable data rate constraints. Then, the performance evaluations of linear precoding schemes and user-centric clustering algorithms are provided through simulations. The goal of this evaluations is to reveal the impact of precoding schemes on the overall performance of the system and to provide comparison of user-centric clustering algorithms in cell-free massive MIMO systems.

2.1 Literature review for downlink cell-free massive MIMO

In the literature, various spectral efficiency and energy efficiency maximization approaches on cell-free massive MIMO have been examined for different scenarios as summarized in Table 2.1. In (Mai et al., 2022) and (Ngo et al., 2018), energy efficiency maximization was performed by analytical optimization approach for centralized and distributed cell-free systems respectively. Similarly, authors of (Ngo et al., 2017) worked on energy efficiency optimization with second order cone problems approach under the assumption of each AP serves all UEs. In (Björnson and Sanguinetti, 2020), the authors performed downlink spectral efficiency maximization by adjusting power coefficients and optimal UE-AP pairing, while (Chen et al., 2022) targeted energy efficiency maximization instead of spectral efficiency maximization. In (Zhao et al., 2020), spectral efficiency maximization was achieved in centralized cell-free massive MIMO systems using centralized deep neural network (DNN) approach under assumption of each UE served by all APs. Similar assumptions are used in (Zaher et al., 2021), where the spectral efficiency maximization performed for distributed cell-free systems using distributed deep neural networks. The article (Chakraborty et al., 2019) expanded the optimization parameters by adding UE-AP pairing for centralized cell-free network approach, suggesting the determination of precoding vectors locally at each AP using locally trained DNNs. In (Biswas and Vijayakumar, 2021), the authors used a machine learning approach for spectral efficiency maximization with UE-AP pairing in centralized architecture in DL. They set the power allocation to be a function instead of optimization problem and allowed ML to determine which UEs would be served by which APs. The comparison of different power allocation algorithms for downlink cell-free massive MIMO systems was given in (Chakraborty et al., 2021).

Even though there are several works focused on spectral efficiency, energy efficiency, and rate maximization for cell-free massive MIMO systems, as given in the Table 2.1, there is no work in the literature that consider energy efficiency maximization under the power and spectral efficiency constraints by using machine learning approach for UE-AP pairing and power allocation jointly.

Table 2.1. Literature Review

Reference	**Objective**	**Parameters**	**Approach**	**Architecture**
(Mai et al., 2022)	EE	PA	APG	Centralized
(Ngo et al., 2018)	EE	PA	SOCP	Distributed
(Ngo et al., 2017)	EE	PA	SOCP	Distributed
(Björnson and Sanguinetti, 2020)	SE	PA & UCC	Analytical	Distributed
(Chen et al., 2022)	EE	PA & UCC	MINLP	Distributed
(Zhao et al., 2020)	SE	PA	ML	Centralized
(Zaher et al., 2021)	SE	PA	ML	Distributed
(Chakraborty et al., 2019)	SE	PA & UCC	ML	Cent. & Dist.
(Biswas and Vijayakumar, 2021)	SE	UCC	ML	Centralized
(Chakraborty et al., 2021)	SE	PA	Analytical	Distributed

2.2 System Model

In this thesis, we consider cell-free Massive-MIMO system with L APs, each equipped with N antennas and K single-antenna UEs in an urban area where $LN \gg K$. The UEs are distributed randomly over the area while the APs are distributed over the area with equal spacing in wrap-around topology. The channel is assumed to be block fading channel where the time-varying wide-band channels are divided into time-frequency coherence blocks such that channels are static and frequency-flat within coherence blocks.

The signal transmitted from AP l will be exposed to path loss, shadowing and multipath effects. According to (3GPP, 2017), for the case where the distance v_{kl} between AP l and UE k satisfies $10m < v_{kl} < 2000m$, AP antenna height of 10m and UE antenna height between 1-2.5m the path loss(PL) between AP l and UE k in non line-of-sight(NLOS) urban area is modeled as

$$\text{PL(dB)} = 22.7\text{dB} + \alpha 10\log_{10}(v_{kl}) + 26\log_{10}(f_c) \tag{2.1}$$

where α is the path loss exponent, v_{kl} is the distance between UE and AP and f_c is the carrier frequency in GHz. For 2GHz carrier frequency and path loss exponent 3.67, the path loss can be further simplified as

$$\begin{aligned}\text{PL(dB)} &= 22.7 + 36.7\log_{10}(v_{kl}) + 7.827\\ &\approx 30.5 + 36.7\log_{10}(v_{kl})\end{aligned} \tag{2.2}$$

The propagation channel between UE k and AP l is denoted as $\boldsymbol{h}_{kl} \in \mathbb{C}^{Nx1}$ and modeled by correlated Rayleigh fading such that $\boldsymbol{h}_{kl} \sim \mathcal{N}_{\mathbb{C}}(\boldsymbol{0}, \boldsymbol{R}_{kl})$ where $\boldsymbol{R}_{kl} \in \mathbb{C}^{NxN}$ is the spatial correlation matrix. Here, the small-scale fading is modeled by the complex Gaussian distribution, while the large-scale fading, which incorporates factors such as geometric path loss, shadowing, antenna gains, and spatial channel correlation, is described by the positive semi-definite correlation matrix $\boldsymbol{R}_{kl}$. The normalized trace $\beta_{kl} = \frac{1}{N}\,\text{tr}(\boldsymbol{R}_{kl})$ accounts for the average channel gain from an antenna at AP l to UE k.

According to (Björnson et al., 2017) spatial correlation matrix elements modeled as

$$[\boldsymbol{R}]_{l,m} = \beta_{kl} \int_{-\infty}^{\infty} e^{j2\pi d_{ant}(l-m)\sin(\theta+\delta)} \frac{1}{\sqrt{2\pi}\sigma_\theta}\, e^{\frac{-\delta^2}{2\sigma_\theta^2}} d\delta \tag{2.3}$$

where σ_θ is angular standard deviation in radians, θ is angle from AP l to UE k, δ is the integration variable and d_{ant} is antenna spacing between adjacent antennas on same AP in terms of wavelength.

The average channel gain β_{kl} is defined in terms of path loss and shadowing as

$$\beta_{kl} = 10^{(\frac{-(\text{PL(dB)}+\text{SF(dB)})}{10})} \tag{2.4}$$

where shadow fading is represented as

$$\text{SF} \sim \mathcal{N}(0, \sigma_{sf}^2) \tag{2.5}$$

and σ_{sf}^2 is the variance of shadow fading.

Let $\boldsymbol{x}_l$ denote the signal transmitted by AP l. The received signal at UE k can be expressed as

$$y_k = \sum_{l=1}^{L} \boldsymbol{h}_{kl}^T \boldsymbol{x}_l + \text{n}_k \tag{2.6}$$

Let s_k denote the independent unit-power data signal intended for UE k where $\mathbb{E}\{|s_k|^2\} = 1, k = 1, ..., K$. Then, the transmitted signal $\boldsymbol{x}_l \in \mathbb{C}^{N\times 1}$ at AP l can be expressed as

$$\boldsymbol{x}_l = \sqrt{P_t}\sum_{i=1}^{K}\sqrt{\eta_{il}}d_{il}\boldsymbol{w}_{il}s_i \tag{2.7}$$

where P_t is the maximum transmit power of AP in Watts, η_{kl} is the power allocation coefficient for AP l and UE k, d_{kl} indicates if the UE k is served by the AP l and $\boldsymbol{w}_{kl} \in \mathbb{C}^{N\times 1}$ is the precoding vector for UE k and AP l.

$$d_{kl} = \begin{cases} 1 & \text{UE } k \text{ is served by AP } l \\ 0 & \text{UE } k \text{ is not served by AP } l \end{cases} \tag{2.8}$$

By substituting (2.7) into (2.6) we obtain general form of the received signal at UE k as

$$y_k = \sqrt{P_t}\sum_{l=1}^{L}\boldsymbol{h}_{kl}^T\sqrt{\eta_{kl}}d_{kl}\boldsymbol{w}_{kl}s_k + \sqrt{P_t}\sum_{\substack{i=1\\ i\neq k}}^{K}\sum_{l=1}^{L}\boldsymbol{h}_{kl}^T\sqrt{\eta_{il}}d_{il}\boldsymbol{w}_{il}s_i + \mathrm{n}_k \tag{2.9}$$

The term n_k is the additive white Gaussian noise (AWGN) at UE k with zero mean and variance σ^2. We define the channel matrix $\boldsymbol{H} \in \mathbb{C}^{K\times LN}$ as

$$\boldsymbol{H} = [\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_K] \tag{2.10}$$

where $\boldsymbol{h}_k = [\boldsymbol{h}_{k1}\ \boldsymbol{h}_{k2}\ \ldots\ \boldsymbol{h}_{kL}] \in \mathbb{C}^{1\times LN}$ and $\boldsymbol{h}_{kl} = [\boldsymbol{h}_{kl1}\ \boldsymbol{h}_{kl2}\ \ldots\ \boldsymbol{h}_{kLN}] \in \mathbb{C}^{1\times N}$ represents the channel between UE k and the antennas of the AP l.
The precoding matrix $\boldsymbol{W} \in \mathbb{C}^{LN\times K}$ is defined as

$$\boldsymbol{W} = [\boldsymbol{w}_1\ \boldsymbol{w}_2\ \ldots\ \boldsymbol{w}_K] \tag{2.11}$$

where $\boldsymbol{w}_k = [\boldsymbol{w}_{k1}, \boldsymbol{w}_{k2}, \ldots, \boldsymbol{w}_{kL}] \in \mathbb{C}^{LN\times 1}$ is the precoding vector for UE k.

Spectral Efficiency

Spectral efficiency is a measure of the amount of information that can be transmitted per unit of bandwidth or frequency spectrum in a communication system. It is defined as the ratio of the information rate to the bandwidth and expressed in bits per second per Hertz (bps/Hz). The rate of the UE k can be expressed as

$$R_k = B\log_2(1+\gamma_k) \tag{2.12}$$

where B is the bandwidth of the system and γ_k is the signal to interference and noise ratio (SINR) at UE k is defined as the ratio of the power of the desired signal to the power of the interference and noise

$$\gamma_k = \frac{|\sum\limits_{l=1}^{L}\sqrt{P_t}\sqrt{\eta_{kl}}\boldsymbol{h}_{kl}^{T}d_{kl}\boldsymbol{w}_{kl}|^2}{\sum\limits_{i=1}^{K}|\sum\limits_{l=1}^{L}\sqrt{P_t}\sqrt{\eta_{il}}\boldsymbol{h}_{kl}^{T}d_{il}\boldsymbol{w}_{il}|^2 - |\sum\limits_{l=1}^{L}\sqrt{P_t}\sqrt{\eta_{kl}}\boldsymbol{h}_{kl}^{T}d_{kl}\boldsymbol{w}_{kl}|^2 + P_n} \tag{2.13}$$

where, P_n is the noise power at UE k expressed as

$$P_n = \text{ThermalNoise} + 10\log_{10}(B) + \text{NF} \quad (\text{dBm}) \tag{2.14}$$

where NF is the noise figure of the receiver. Then, the sum data rate is determined as

$$R_{sum} = \sum_{k=1}^{K} R_k \tag{2.15}$$

and the spectral efficiency of UE k can be expressed as

$$\text{SE}_k = \log_2(1 + \gamma_k) \tag{2.16}$$

Energy Efficiency

Economic and environmental concerns have led to the development of energy efficient wireless communication systems. Energy efficiency is a measure of the trade-off between the energy required to transmit information over a communication channel and the amount of information that is transmitted. Therefore, it is one of the most critical concerns in the design and deployment of large scale wireless communication systems. The energy efficiency of the system can be expressed as

$$\text{EE} = \frac{\text{R}_{sum}}{\text{P}_{total}} \tag{2.17}$$

where R_{sum} is the sum data rate of the system in Equation (2.15) and P_{total} is the total power consumption of the system.

The total power consumption can be broken down several components that can be summarized in three main parts as transmit power (P_{tx}), AP internal power consumption P_{int} and backhaul power consumption P_{bh}(Björnson et al., 2015),(Zuo et al., 2017). Transmit power consumption is the power consumed to transmit the intended signal including the signal power as well as power consumed at power amplifier to transmit the signal with required power. AP internal power consumption covers digital signal processing units such as FPGAs and ASICs, RF chains that includes mixers, filters and oscillators, cooling systems and power management units. Backhaul power consumption is the power consumption related to the structure between APs and CPU. This structure includes network processing components such as the switches and routers that are responsible for routing and forwarding data packets, signal boosters for long backhaul links, protocol converters, traffic managers, security and fault detection, recovery components in general.

$$P_{total} = \mathrm{P}_{tx} + \mathrm{P}_{int} + \mathrm{P}_{bh} \tag{2.18}$$

The total transmit power P_{tx} can be expressed as

$$P_{tx} = \sum_{l=1}^{L} \frac{1}{\alpha_{dl}} \left(\sum_{k=1}^{K} P_t \eta_{kl} d_{kl} ||\boldsymbol{w}_{kl}||^2 \right) \tag{2.19}$$

where α_{dl} represents power amplifier efficiency. This power consumption is higher than the total transmitted signal power due to the power amplifier efficiency.

Total internal power consumption of APs P_{int} can be expressed as

$$P_{int} = LNP_{ac} + \sum_{l=1}^{L} R_l P_{tc} \tag{2.20}$$

where P_{ac} represents the fixed circuit power consumption at each antenna such as converters, mixers and filters attached to each antenna at APs. P_{tc} stands for load dependent power consumption of APs such as coding.

Lastly, total power consumption at backhaul links between CPU and APs is denoted as P_{bh}. As P_{int}, backhaul power consumption P_{bh} has traffic dependent and fixed terms that can be expressed as

$$P_{bh} = LP_0 + \sum_{l=1}^{L} R_l P_{bt} \tag{2.21}$$

where P_0 is the fixed power consumption of each backhaul due to infrastructural components, P_{bt} is the traffic-dependent power consumption of backhaul and

$$R_l = \sum_{k=1}^{K} R_k d_{kl} \tag{2.22}$$

is the data transfer rate of the backhaul l. The number of APs that serve UE k is determined by

$$d_k = \sum_{l=1}^{L} d_{kl} \tag{2.23}$$

In real systems, AP and backhauls consume small amount of power even when they are on sleep mode. Since they are smaller than hundredth of the power consumptions in active state, the consumptions on sleep mode will be neglected for simplicity.

2.3 Linear Precoding Schemes

Linear precoding involves the manipulation of signals at the transmitters to allow multiple users to share the same frequency resources without causing significant interference by exploiting the spatial domain. This technique is particularly useful in cell-free massive MIMO, where the base stations are not confined to a specific area and can be deployed anywhere in the network.

In this thesis, we refer to three linear precoding schemes such as maximum ratio transmission (MRT), zero forcing precoding (ZF) and regularized zero forcing precoding (RZF). We will examine the performance of these schemes considering the number of UEs and APs and thresholds of user-centric clustering algorithm schemes in terms of energy efficiency.

2.3.1 Maximum Ratio Transmission Precoding

The idea behind maximum ratio transmission (MRT) precoding is to transmit user signals in such a way that it is received with the maximum power possible at the user equipments. This goal is achieved by multiplying the signal with a precoding vector that are obtained from complex inverse of the channel matrix.

One advantage of MRT is that it is simple to implement and does not require any additional information, such as the interference covariance matrix, to calculate the precoding vector. However, MRT may suffer from inter-user interference, as it does not take into account the interference from other users. Therefore, MRT is typically preferred over ZF and RZF in scenarios where the SINR is high and the system requires a simple and computationally efficient precoding scheme. The MRT precoding matrix $\boldsymbol{W}^{'}_{MRT}$ is defined as complex conjugate of the channel matrix and expressed as

$$\boldsymbol{W}^{'}_{MRT} = \boldsymbol{H}^{H} \tag{2.24}$$

where $\boldsymbol{W}^{'}_{MRT} = [\boldsymbol{w}^{'}_{1}, \boldsymbol{w}^{'}_{2}, \ldots, \boldsymbol{w}^{'}_{K}] \in \mathbb{C}^{LN \times K}$

To eliminate precoding effects on transmit power, the precoding matrix is normalized such that

$$\boldsymbol{w}_k = \frac{\boldsymbol{w}^{'}_{k}}{||\boldsymbol{w}^{'}_{k}||} \tag{2.25}$$

2.3.2 Zero Forcing Precoding

The main goal of zero forcing (ZF) precoding is to eliminate inter-user interference, which is one of the main challenges in cell-free massive MIMO, by designing such that the signal intended for one user does not interfere with the signals intended for other users. This is achieved by orthogonalization of the signals at the receive terminals, effectively eliminating the interference.

The significant advantage of ZF precoding is that it can effectively eliminate inter-user interference, leading to improved system performance. However, ZF precoding also amplifies the noise, therefore it may result reduced SINR compared to MRT. Therefore, ZF precoding is typically preferred over MRT in scenarios where the channel matrix is well conditioned and the potential inter-user interference is higher than the noise power.

The ZF precoding matrix $\boldsymbol{W}'_{ZF}$ is defined as

$$\boldsymbol{W}'_{ZF} = \boldsymbol{H}^H(\boldsymbol{H}\boldsymbol{H}^H)^{-1} \tag{2.26}$$

where $\boldsymbol{W}'_{ZF} = [\boldsymbol{w}'_1, \boldsymbol{w}'_2, \ldots, \boldsymbol{w}'_K] \in \mathbb{C}^{LN \times K}$

The resulting precoding vectors are normalized as in Equation (2.25).

2.3.3 Regularized Zero Forcing Precoding

Regularized zero forcing (RZF) precoding is an extension of the ZF precoding technique. As mentioned above, ZF has a disadvantage in the low SNR cases compared to MRT. The main goal of regularized zero forcing (RZF) precoding is balancing the trade-off between inter-user interference elimination and noise amplification. This is achieved by adding a small amount of noise to the channel matrix, which is known as regularization. The regularization parameter λ is a positive real number that controls the trade-off between the SNR and the inter-user interference. If λ is set to zero, the precoding matrix become identical to the ZF precoding matrix and if λ is set to infinity, the precoding matrix become identical to the MRT precoding matrix (Bobrov et al., 2022).

The RZF is typically preferred over ZF and MRT in low SNR cases, since it can effectively balance the trade-off between inter-user interference elimination and noise amplification. Also when there is a high correlation between the antennas or the channel matrix is close to singular, the inverse matrix used in ZF may be ill-conditioned, leading to numerical instabilities. In this case, RZF can be used to provide a more stable solution (Bobrov et al., 2022),(Krishnamoorthy and Schober, 2023).

The RZF precoding matrix $\boldsymbol{W}'_{RZF}$ is calculated as

$$\boldsymbol{W}^{'}_{RZF} = \boldsymbol{H}^{H}(\boldsymbol{H}\boldsymbol{H}^{H} + \lambda \boldsymbol{I}_{K})^{-1} \tag{2.27}$$

where $\boldsymbol{W}^{'}_{RZF} = [\boldsymbol{w}^{'}_{1}, \boldsymbol{w}^{'}_{2}, \ldots, \boldsymbol{w}^{'}_{K}] \in \mathbb{C}^{LN \times K}$, λ is the regularization parameter and $\boldsymbol{I}_K$ is the identity matrix with dimension $K \times K$. The resulting precoding vectors are normalized as in Equation (2.25).

2.4 User Equipment & Access Point Pairment

Centralized cell-free massive MIMO systems require more backhaul connections (the link between AP and the central processing unit) to transfer data between APs and CPU compared to colocated massive MIMO systems. These backhaul connections increases with the number of APs L. It can easily be seen from (2.18) and (2.21) that backhaul power consumption is directly related to the number of backhaul links and data transfer rate of each link. As given in (2.17), to improve overall energy efficiency, we need to reduce the total power consumption. One way to reduce power consumption is reducing the number of backhaul links and data transfer rates per link. Without using any UE-AP pairing schemes, all UEs in the network would be served by all APs. This means that each AP's backhaul would need to transfer all of the UEs' data, which results a high amount of data transfer on the backhaul links. This, in turn, leads to excessive power consumption on backhaul links due to load dependent power consumptions. To address this issue, there are several UE-AP pairing schemes in literature that can be used to reduce the amount of data on the backhaul links. One of the pairing schemes in (Björnson and Sanguinetti, 2020) is user-centric clustering, will be considered in this thesis.

2.4.1 User-centric Clustering Algorithm

User-centric clustering algorithm is based on how does additional APs contribute in communication performance and how much power is consumed in exchange. APs that located far from UEs generally has low channel gains and does not contribute communication performance significantly in terms of spectral efficiency improvement but they still transfer data for those UEs over their backhauls. Therefore overall power consumption will be increased even though there is no significant improvement on spectral efficiency. In stationary UE case, the UE-AP matching process is performed every time when a new UE is joined to or an existing UE left from the cell-free network. To overcome this problem and improve overall energy efficiency, UEs should not be served by APs that has no significant contributions. In the user-centric clustering algorithm in (Björnson and Sanguinetti, 2020),

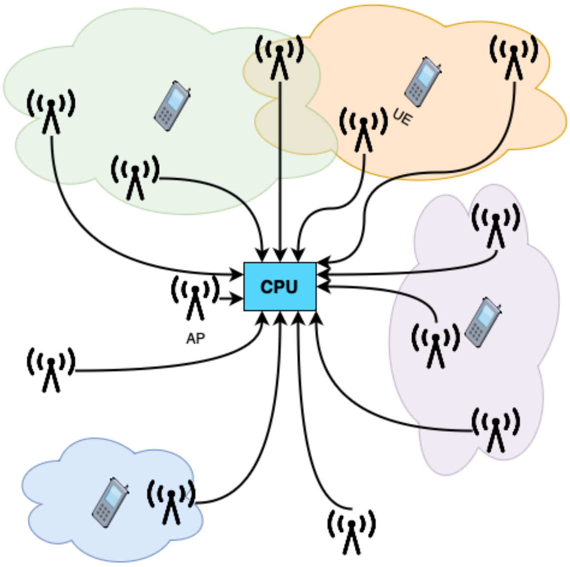


Figure 2.1. Representation of User-Centric Clustering.

APs will serve UEs only if the channel gain between UE-AP pair is higher than the channel gain between the UE and its master AP with a given threshold. This algorithm is given in Algorithm 1.

Algorithm 1: User-centric clustering algorithm in (Björnson and Sanguinetti, 2020)

```
Input: Channel gains between all UEs and all APs as $\beta_{kl}$, UE-AP pairing
       threshold $\lambda_1 \geq 0$
Output: Cooperation clusters formed by APs to serve UEs
Initialization: $d_{kl} = 0;\ \forall k = 1, 2, \ldots, K$ and $\forall l = 1, 2, \ldots, L$
foreach UE $k$ do
    Find AP with highest channel gain and assign it as master AP
    $l \leftarrow \arg\max_l \beta_{kl}$
    $d_{kl} \leftarrow 1$
    $\beta^*_{kl} \leftarrow \beta_{kl}$
end
foreach AP $l$ do
    foreach UE $k$ do
        if $\beta_{kl} \geq \beta^*_{kl} + \lambda_1$ then
            $d_{kl} \leftarrow 1$
        end
    end
end
```

As a first step, each UE chooses a master AP that has the highest channel gain. Then, APs need to form cooperation clusters by deciding which UEs they will serve. Each AP will check each UE's channel gain. If the channel gain for the selected UE is higher than the channel gain between selected UE and its master AP by threshold λ_1 or more, the AP will serve that UE. If the channel gain for the selected UE is less than the channel gain between selected UE and it's master AP by this threshold, the AP will not serve that UE.

As discussed before, non-master APs should serve UEs only if it contributes to the system performance of UEs by reasonable amount. However, these contributions depends on many different parameters such as channel gains, the number of UEs served by the AP, the number of APs in the system and additional interference caused by those APs. Therefore, channel gain alone may not be enough to justify additional power consumptions of those APs and decide whether to serve a UE or not. However, considering all those parameters on the decision is a complex task. Therefore, a simple and efficient algorithm is needed to decide whether to serve a UE or not.

In this context, we propose a novel energy efficiency focused user-centric clustering algorithm. The proposed algorithm is based on the idea of minimizing the number of APs that are on active mode by refusing to serve only a few UEs if the AP is not the master AP of those UEs by introducing a threshold value λ_2. The proposed algorithm is given in Algorithm 2.

Algorithm 2: The proposed user-centric clustering algorithm

```
Input: Channel gains between all UEs and all APs as $\beta_{kl}$, UE-AP pairing
       threshold $\lambda_1 \geq 0$, AP disabling threshold $\lambda_2 \geq 0$
Output: Cooperation clusters formed by APs to serve UEs
Initialization: $d_{kl} = 0;\ \forall k = 1, 2, \ldots, K$ and $\forall l = 1, 2, \ldots, L$

foreach UE $k$ do
    Find AP with highest channel gain and assign it as master AP
    $l \leftarrow \arg\max_l \beta_{kl}$
    $d_{kl} \leftarrow 1$
    $\beta^*_{kl} \leftarrow \beta_{kl}$
end
foreach AP $l$ do
    foreach UE $k$ do
        if $\beta_{kl} \geq \beta^*_{kl} + \lambda_1$ then
            $d_{kl} \leftarrow 1$
        end
    end
    $d_l \leftarrow \sum_{k=1}^{K} d_{kl}$
    if $d_l < \lambda_2$ and AP $l$ is not master AP of any UE then
        Set $d_{kl} = 0;\ \forall\ k = 1, 2, \ldots, K$
        Disable AP $l$
    end
end
```

The threshold λ_2 is a empirical value that determines the minimum number of UEs that non-master APs should serve. If the number of UEs that non-master APs serve is less than λ_2, the AP will be disabled and will not serve any UE. Therefore, the total power consumption and the computational complexity of the power allocation will be reduced.

2.5 Energy Efficiency Maximization

As mentioned in Section 2.2, energy efficiency is a crucial aspect in the design and operation of communication systems, particularly in the context of cell-free massive MIMO networks. According to (Andrae and Edler, 2015), even though 22% annual energy efficiency improvement is included, by 2030, energy consumption of 5G data networks is expected to reach nearly the total energy consumption of all 2G/3G voice and 2G/3G/4G data networks combined at 2020. This highlights the increasing demand for energy in communication systems and the need to develop more energy-efficient technologies to sustain the growth of data networks. Minimizing energy consumption in cell-free massive MIMO systems can reduce the operating costs of the network, moreover, it also may reduce the carbon footprint of the network which can help to mitigate the effects of climate change and make the world wide communications more sustainable. The aim of this research is maximizing energy efficiency of the cell free massive MIMO systems while satisfying the users' data rate requirements.

2.5.1 The Optimization Problem

In general, energy efficiency maximization in wireless communication systems can be formulated as an optimization problem which takes into account different constraints such as rate, maximum transmit power, fairness etc. The convexity of the optimization problem depends on the specific formulation of the objective function and the constraints.

In the context of this thesis, objective of the optimization problem is maximizing the energy efficiency by adjusting power allocation coefficients in a way that satisfies certain constraints. The constraints are guaranteed that all UEs are connected to at least one AP, the data rate should be above a certain threshold, and the total transmit power per AP should be below the maximum available power. These constraints are critical to ensure that the system operates reliably and does not exceed the power limits.

Non-convex optimization problems are generally more difficult to solve than convex optimization problems since they can have multiple local optima, which can make it challenging to find the global optimum. Solving this problem requires the use of advanced optimization techniques, such as interior point methods, semi-definite programming, or alternating optimization. These are powerful optimization methods that can handle non-convex optimization problems, but they can also be computationally intensive and require expertise to implement.

The optimization problem in this thesis is non-convex and we formulate it as follows:

$$(\mathcal{P}): \begin{cases} \max\limits_{\{\eta_{kl}\}} & \text{EE} \\ \text{s.t.} & R_k \geq R_{th} \qquad \forall k = 1, 2, ..., K \\ & \sum\limits_{k=1}^{K} \eta_{kl} \leq 1 \quad \forall l = 1, 2, ..., \; L \\ & d_k > 0 \qquad\quad \forall k = 1, 2, ..., K \end{cases} \tag{2.28}$$

Where, R_{th} is the minimum data rate required for each UE to provide a satisfactory level of service to the user.

In this thesis, MATLAB's `fmincon` solver is used with the interior point method to solve the optimization problem. The interior point method is a type of optimization algorithm used to solve non-convex optimization problems. The interior point method approach is based on the idea of transforming the original optimization problem into a sequence of convex optimization problems, each of which has a strictly feasible interior point. It works by constructing a sequence of points that are strictly inside the feasible region of the optimization problem, and gradually moving towards the optimal solution while satisfying a set of constraints. The method can be computationally intensive, especially for large-scale problems and it may be sensitive to the choice of the initial point used in the algorithm.

Power Constraints

Due to regulatory and physical limitations, transmit power of the access points cannot be higher than a certain amount. In order to obtain real-world applicable solutions, the optimization problem must obey these power constraints. The transmit power of AP l is

$$\mathrm{P}_l = P_t \sum_{k=1}^{K} \eta_{kl} d_{kl} ||\boldsymbol{w}_{kl}||^2 \tag{2.29}$$

Since $\mathbb{E}\{||\boldsymbol{w}_{kl}||^2\} = 1; \; \forall k = 1, .., K$, the equation (2.29) can be further simplified as

$$\mathrm{P}_l = P_t \sum_{k=1}^{K} \eta_{kl} d_{kl} \tag{2.30}$$

Since $P_{ln} \leq P_t$ by definition, the power constraint can be expressed as

$$\sum_{k=1}^{K} \eta_{kl} \leq 1 \tag{2.31}$$

Data Rate and Availability Constraints

In a wireless network, the goal of maximizing energy efficiency may cause trade-off with maintaining high levels of data rates or availability. For example, reducing the number of active APs in the network can save energy, but it may also reduce the coverage and increase the congestion in the network, resulting in a lower data rate and coverage for the end users. To ensure a reliable and stable wireless communication system, it is essential to satisfy both data rate and availability constraints.

The availability constraint ensures that each UE in the network will always be served by at least one AP. This guarantees that every UE has access to the network. The UE-AP paring is managed by d_{kl} values and if d_k given in Equation (2.23) is greater than zero, it means that there exists at least one AP that connected and serving to the UE k.

The data rate constraint, on the other hand, guarantees that the downlink data rate of each UE will never fall below a predefined threshold R_{th}. This threshold represents the minimum data rate required for each UE to provide a satisfactory level of service and should be set based on the specific requirements and applications. By ensuring that the downlink data rate remains above this threshold, the data rate constraint helps to guarantee a high level of performance and a reliable user experience.

2.6 Performance Evaluations

In the simulation environment, we consider a downlink cell-free massive MIMO system with L APs and K single-antenna UEs. Each AP has N antennas and uniformly distributed over the $1500\text{m} \times 1500\text{m}$ area and each UE is located at a random position in the area. The required data rate is arbitrarily chosen to be 40Mbps which corresponds to 2 bps/Hz spectral efficiency for 20MHz system bandwidth. The disabling threshold λ_2 in Algorithm 2 is set to 2.

In (Zaher et al., 2021) and (3GPP, 2017), the maximum downlink transmit power per AP is set to 30dBm. In power allocation optimization performance evaluations, equal power allocation is used as $\eta_{kl} = 0.25;\ \forall l = 1, 2, ..., L$ and $\forall k = 1, 2, ..., K$ as a baseline solution.

By using Equation (2.14), noise power can be calculated as

$$\begin{aligned}\mathrm{P_n} &= \mathrm{ThermalNoise} + 10\log_{10}(B) + \mathrm{NF} \quad (\mathrm{dBm}) \\ &= -174 + 10\log_{10}(20 \times 10^6) + 7 \quad (\mathrm{dBm}) \\ &\approx -94\mathrm{dBm}\end{aligned} \tag{2.32}$$

where the thermal noise at the receiver under room temperature can be approximated as -174 dBm/Hz (Papazafeiropoulos et al., 2020) and NF is the noise figure of the receiver in dB. The noise figure of the receiver is the ratio of the noise power at the input of the receiver to the noise power at the output of the receiver and taken as 7dB according to (Demir et al., 2021).

The simulation parameters are summarized in Table 2.2 (Ngo et al., 2018), (Zuo et al., 2017), (Ngo et al., 2017). The results are obtained through Monte Carlo simulations over 2500 randomly generated setups unless otherwise stated.

Table 2.2. Simulation Parameters

Parameter	Value
Area	$1500\mathrm{m} \times 1500\mathrm{m}$
Number of antennas per AP, N	4
System bandwidth, B	20MHz
Noise figure, NF	7dB
Standard deviation of shadow fading , σ_{sf}	4dB
UE-AP pairing threshold, λ_1	-10dB
AP disabling threshold, λ_2	2
Height of APs	10m
Height of UEs	1.5m
Path loss exponent , α	3.67
Angular standard deviation of the UE, σ_θ	20°
Power amplifier efficiency, α_{dl}	0.4
Maximum transmit power per AP, P_t	1W
Internal power consumption per antenna, P_{ac}	0.2W
Fixed power consumption at each backhaul, P_0	0.825W
Traffic-dependent power consumption at each backhaul, P_{bt}	0.25 W/Gbps
Traffic-dependent power consumption at each AP, P_{tc}	0.1 W/Gbps

For performance comparison, we consider the case of equal power allocation for all UEs as a base value. The simulations discussed in this paper were implemented on an Apple M1 Pro 3.2GHz 10-core CPU with 16 GB RAM and computational complexity of the optimization problem is measured using MATLAB's `time` command in terms of time spent for the optimization process.

The simulations discussed in this paper were implemented on an Apple M1 Pro 3.2GHz 10-core CPU with 16 GB RAM.

Spectral Efficiency

To investigate the effect of the number of access points and user equipments on spectral efficiency the simulations included both fixed number of UEs and varying number of APs, as well as fixed number of APs and varying number of UEs under the assumption of each UE being served by all APs are performed.

The first set of simulations are involved a fixed number of UEs ($K = 7$) and different numbers of APs as $L = 3, 5, 7, 9, 11, 13$. The results are given in Figure 2.2, which represents the cumulative density function(CDF) of the resulting spectral efficiency. The average spectral efficiency values for $L = 3, 5, 7, 9, 11, 13$ are obtained as $1.61, 3.34, 4.76, 5.71, 6.64, 7.32$ bps/Hz respectively. It can be clearly seen from the Figure 2.2 and the given average spectral efficiency values that, as the number of AP is increased, the spectral efficiency is also increased. Since each UE is being served by more APs, it results in a higher level of spatial diversity, an improved precoding capability.

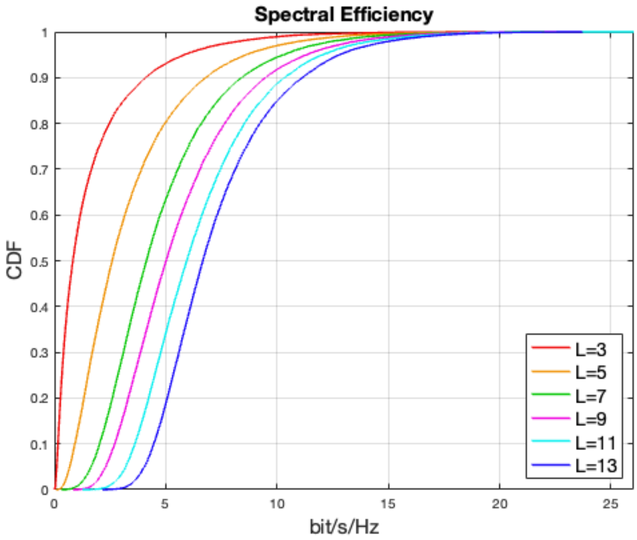


Figure 2.2. CDF of spectral efficiency for different number of APs, for K=7 and N = 4.

The second set of simulations are involved a fixed number of APs ($L = 9$) with different number of UEs $K = 3, 5, 7, 9, 11, 13$ and the average spectral efficiencies are obtained as $7.73, 6.88, 6.29, 5.81, 5.38, 5.02$ bps/Hz respectively. The results are given in Figure 2.3 shows that, the spectral efficiency is decreased as the number of UEs is increased. This decrease in spectral efficiency is due to the increased interference and the reduced transmit power per UE as a result of adding more UEs to the network.

The results of the simulations on spectral efficiency in a cell-free massive MIMO system shows that increasing the number of APs led to an improvement in spectral efficiency, while increasing the number of UEs led to a decrease in spectral efficiency.

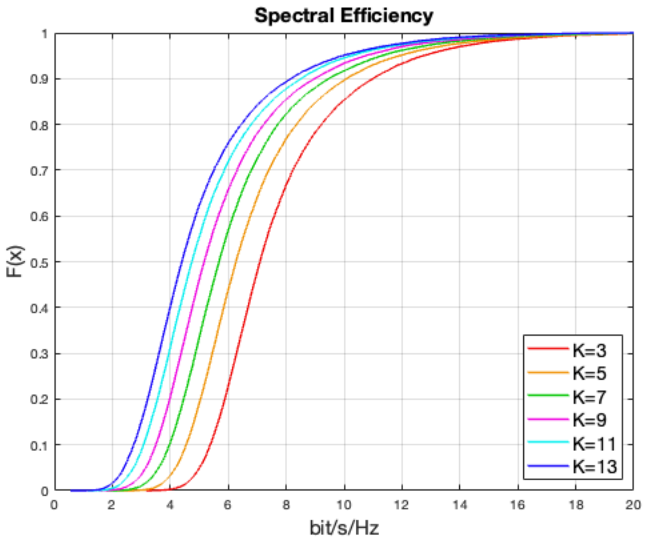


Figure 2.3. CDF of spectral efficiency for different number of UEs, for L=9 and N = 4.

Energy Efficiency

As in previous section, to investigate the effect of the changes in the number of access points and user equipments on energy efficiency, the same set of simulations are conducted with both fixed numbers of UEs and varying numbers of APs, as well as fixed numbers of APs and varying numbers of UEs under the assumption of each UE being served by all APs.

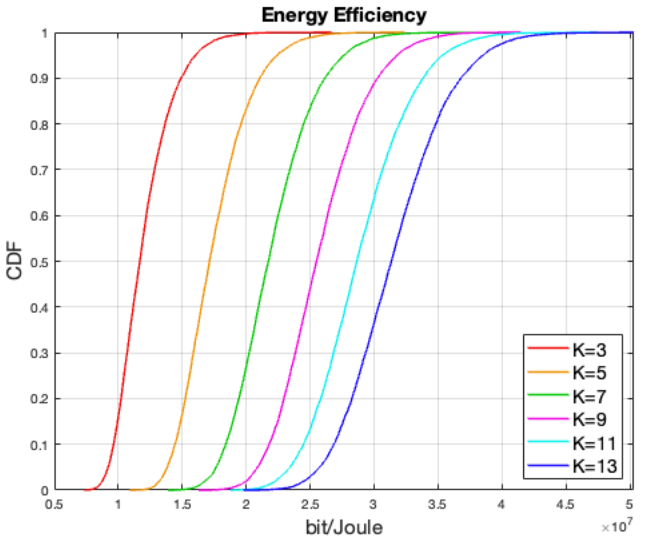


Figure 2.4. CDF of energy efficiency for different number of UEs, for L=9 and N=4.

The first set of simulations is involved a fixed number of APs (L = 9) and different numbers of UEs ($K = 3, 5, 7, 9, 11, 13$). The results are given in Figure 2.4 as CDF of the energy efficiency. The corresponding average energy efficiency values are calculated as $12.0, 17.5, 22.0, 25.8, 28.9, 31.6$ Mb/J respectively.

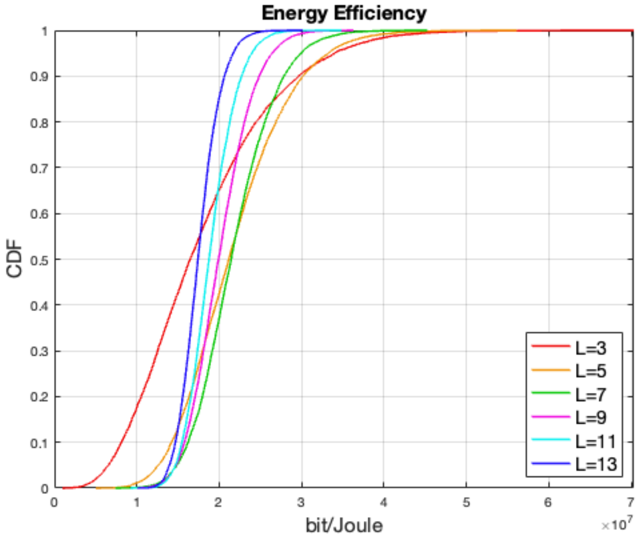


Figure 2.5. CDF of energy efficiency for different number of APs, for K=7 and N=4.

The second set of simulations is involved a fixed number of UEs (K = 7) and different numbers of APs ($L = 3, 5, 7, 9, 11, 13$). The results are given in Figure 2.5 as CDF of the energy efficiency. The corresponding average energy efficiency values are calculated as $17.8, 21.8, 21.8, 20.1, 18.9, 17.6$ Mb/J respectively. Unlike the spectral efficiency results, the average energy efficiency values are not monotonically increasing or decreasing by the number of APs.

The reason is that, both the fixed and load dependent power consumptions are increased approximately by $\frac{1}{L}$ when an additional AP is joined to the system with L active APs. However, as mentioned in Section 2.2, the data rate does not increase at the same rate.

To examine this behavior, the changes on spectral efficiency, power consumption and energy efficiency relative to previous case for fixed number of UEs (K = 7) and different numbers of APs ($L = 3, 5, 7, 9, 11, 13$) are calculated. Increasing L from 3 to 5 is increased the spectral efficiency by 106.9% while the power consumption is increased by 70.7%. This results in an increase in energy efficiency by 21.8%. Similarly, increasing L from 5 to 7 is increased the spectral efficiency by 42.3% and power consumption by 42.3% resulting negligible increase on energy efficiency. Contrarily, increasing L from 7 to 9 is increased the spectral efficiency by 20.1% while power consumption is increased by 30.3% which results in a decrease in energy efficiency by 7.6%.

Precoding Performance

In this section, the energy efficiency of each precoding scheme is inspected for different scenarios as shown in Figure 2.6. First of all, the simulation results for different AP and UE ratios are evaluated with equal power allocation and allocating all APs to all UEs. After that, we evaluate the performance of the power allocation optimization process in terms of energy efficiency and computational complexity for three different precoding schemes including MRT, ZF and RZF. In order to evaluate the performance of the power allocation optimization, the simulations are also performed for three different user-centric clustering scenarios such as Algorithm 1, Algorithm 2 and the baseline scheme where all UEs are served by all APs.

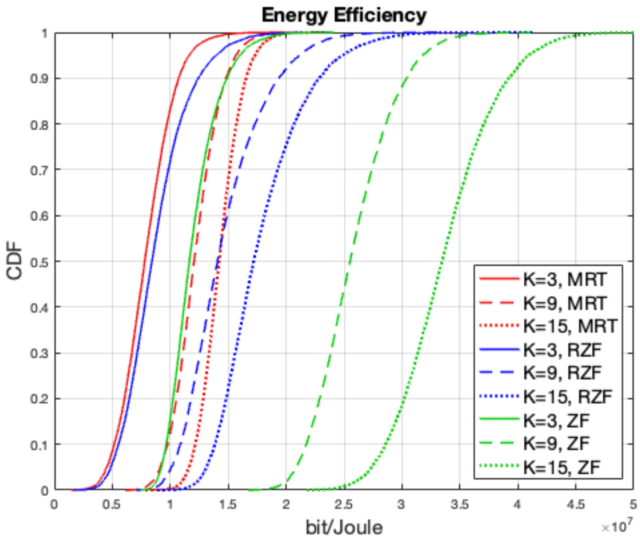


Figure 2.6. Precoding performance comparison for different number of UEs, for L=9 and N=4.

As seen in the Figure 2.6, ZF precoding scheme provides the best performance in terms of energy efficiency. Then, to further investigate the performance of ZF precoding scheme, we provide results on user-centric clustering and power allocation as given in Figure 2.7. It can clearly be seen that the energy efficiency of the system increases with both user-centric clustering and power allocation optimization. The user-centric clustering has more significant effect on energy efficiency than power allocation optimization. The reason is that the user-centric clustering reduces the number of active APs and therefore reduces the total power consumption. However, since the transmit power corresponds to a small portion of the total power consumption, power allocation optimization does not have significant impact on total power consumption. Also, since the ZF precoding eliminates the interference, power allocation optimization has limited effect on SINR and consequently on the energy efficiency.

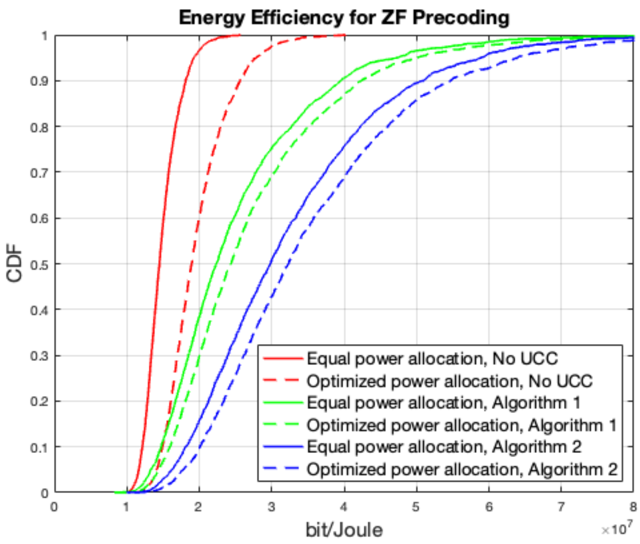


Figure 2.7. Energy efficiency of ZF precoding for equal and optimized power allocations, for K=4, L=9, N=4.

For the case of no user-centric clustering, the power allocation optimization on ZF precoding is increased the energy efficiency of the system by 32.54% relative to the equal power allocation case. However, for the user-centric clustering, the energy efficiency of the system increased by 8.26% and 9.07% by using Algorithm 1 and Algorithm 2, respectively. This infers that, the proposed user-centric algorithm is more effective than the Algorithm 1 in terms of energy efficiency improvement by power allocation optimization. The energy efficiency improvement of the system with joint user-centric clustering and power allocation optimization is 84.06% and 138.52% for Algorithm 1 and Algorithm 2, respectively. One of the key outcomes of this result is that, the proposed UCC algorithm even without power allocation optimization achieves higher energy efficiency performance than the Algorithm 1 with power allocation optimization. This is because, the proposed user-centric clustering algorithm is designed to minimize the active APs and reduce the power consumption while compromising some robustness to the different setup topologies.

The Figure 2.8 represents the effect of data rate constraint on the power allocation optimization performance for the systems with no user-centric clustering, Algorithm 1 and Algorithm 2. The data rate constraint is defined as the minimum required data rate for each UE. It can be seen from the Figure 2.8 that as the minimum required data rate increases, the energy efficiency gain from the power allocation optimization slightly reduces and the most significant reduction is observed with the Algorithm 2. The reason is that, the proposed algorithm has lowest number of connected APs per UE and therefore, there are lower amount of optimizable parameters to satisfy data rate requirements.

To further investigate the performance of RZF precoding scheme, we provide results on user-centric clustering and power allocation as given in Figure 2.9. As seen in

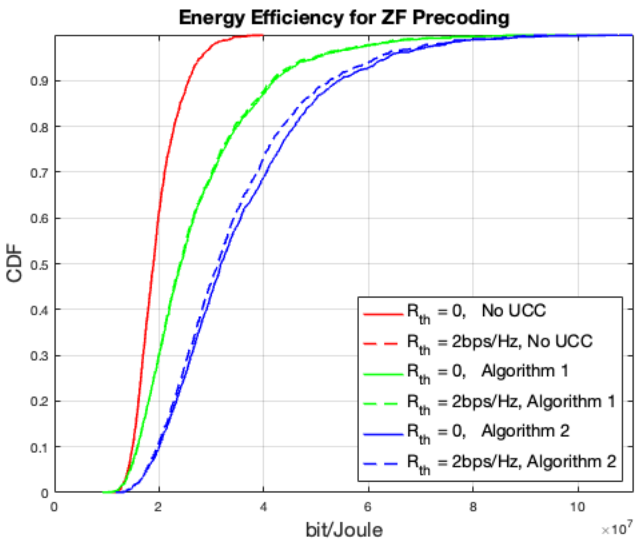


Figure 2.8. Effect of data rate constraints on the performance using ZF precoding, for K=4, L=9, N=4.

the Figure 2.9, the user-centric clustering and power allocation with RZF precoding has higher energy efficiency gains compared to the ZF precoding. The gap between user-centric clustering and power allocation gains become narrower with RZF precoding. Also, the gap between equal power allocation and power allocation optimization become wider. The reason is that, RZF precoding does not eliminate the interference completely and therefore, power allocation optimization has more impact on the interference and consequently on the energy efficiency.

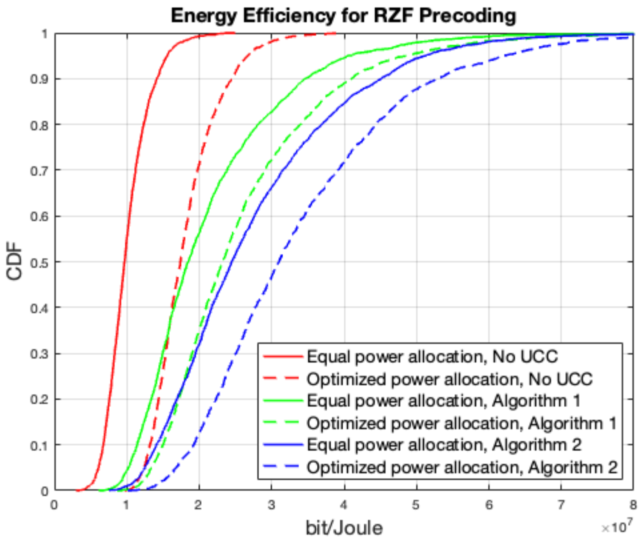


Figure 2.9. Energy efficiency of RZF precoding for equal and optimized power allocations, for K=4, L=9, N=4.

For the case of no user-centric clustering, power allocation optimization on RZF precoding increased the energy efficiency of the system by 79.89% relative to the equal power allocation case. However, for the user-centric clustering, the energy efficiency of the system increased by 22.63% and 24.27% for the setups by using Algorithm 1 and Algorithm 2, respectively. The energy efficiency improvement of the system with joint user-centric clustering and power allocation optimization is 159.01% and 237.33% for Algorithm 1 and Algorithm 2, respectively.

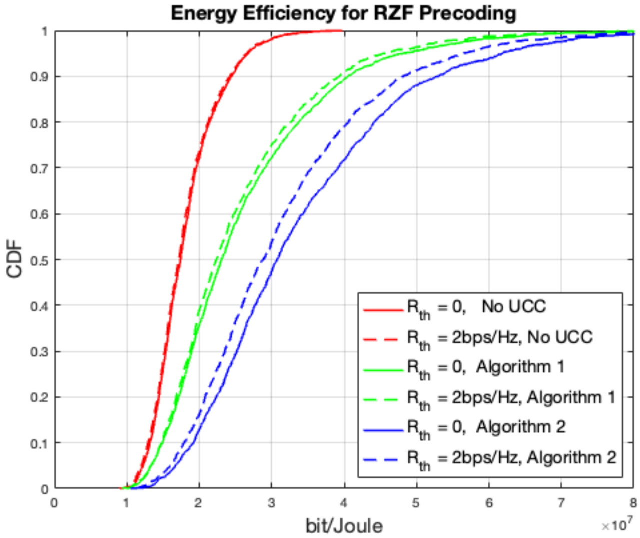


Figure 2.10. Effect of data rate constraints on the performance using RZF precoding, for K=4, L=9, N=4.

As seen in the Figure 2.10 the effect of data rate constraint on the energy efficiency performance of the system with RZF precoding is higher than the ZF precoding case.

As the last of precoding performance evaluations, to further investigate the performance of MRT precoding scheme, we provide results on user-centric clustering and power allocation as given in Figure 2.11. As seen in the Figure 2.11, the user-centric clustering and power allocation with MRT precoding has higher energy efficiency gains compared to the ZF and RZF precoding schemes. The MRT precoding results narrowest gap between user-centric clustering and power allocation gains. Also, the gap between equal power allocation and power allocation optimization is wider than the ZF and RZF precoding. The reason is that, MRT precoding does not eliminate the interference and therefore, power allocation optimization has more impact on the interference than ZF and RZF precoding schemes.

For the case of no user-centric clustering, power allocation optimization on MRT precoding increased the energy efficiency of the system by 98.58% relative to the equal power allocation case. For the user-centric clustering, the energy efficiency of the system increased by 30.38% and 32.97% for the setups with Algorithm 1 and Algorithm 2,

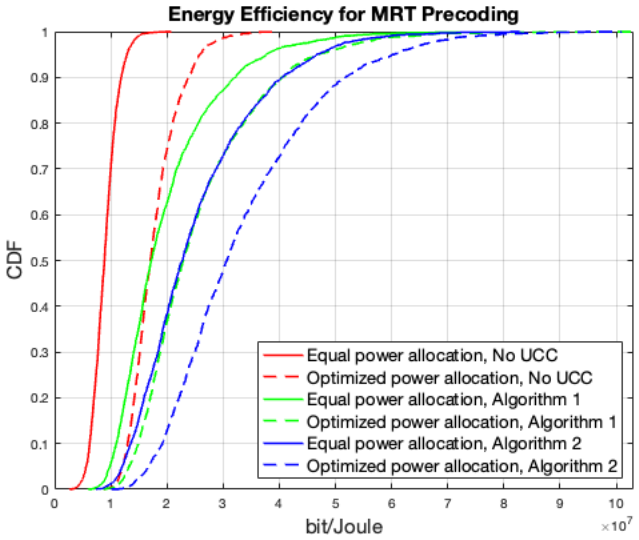


Figure 2.11. Energy efficiency of MRT precoding for equal and optimized power allocations, for K=4, L=9, N=4.

respectively. The energy efficiency improvement of the system with joint user-centric clustering and power allocation optimization is 187.23% and 273.86% for Algorithm 1 and Algorithm 2, respectively.

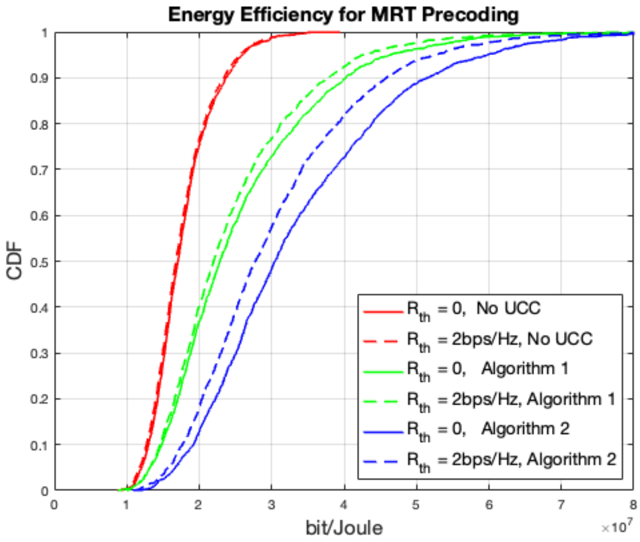


Figure 2.12. Effect of data rate constraints on the performance using MRT precoding, for K=4, L=9, N=4.

As seen in the Figure 2.12 the effect of data rate constraint on the energy efficiency performance of the system with MRT precoding is higher than the ZF and RZF precoding cases.

2.6.1 User-centric Clustering Performance Results

Monte Carlo simulations are performed to evaluate the impact of different pairing thresholds on system performance. The first set of simulations with equal power allocation carried out on 100,000 randomly generated setups using Algorithm 1 to provide a comprehensive assessment of the network's behavior under different threshold values. The results of the simulations are presented in Table 2.3 in terms of the percentage change relative to the baseline case where all UEs are served by all APs. The results indicated that, as the threshold value increases, the average number of connected UEs per AP decreases. This reduction in the number of connected APs leads to a decrease in the overall data transfer on the backhaul links, which reduces the load on the network and increases the probability of having non-serving APs and their corresponding backhaul links to be put into sleep mode. This, in turn, helps in saving power. However, serving UEs with less APs also reduces the spatial diversity which leads to a decrease in spectral efficiency. Therefore, there is a need to strike a balance between power consumption and spectral efficiency. To achieve this balance, the threshold must be set at an appropriate value that maximizes spectral efficiency while minimizing power consumption.

Table 2.3. Threshold effects on user-centric clustering relative to non-clustered case

Threshold λ_1	EE Gain	Power Cons. Reduction	SE Loss	Active AP Reduction
-5dB	104.11%	70.96%	44.61%	48.92%
-10dB	70.42%	59.94%	37.42%	32.64%
-15dB	38.86%	44.86%	28.83%	14.76%
-20dB	18.74%	28.97%	19.22%	3.51%
-25dB	9.47%	16.61%	11.03%	0.42%
-30dB	5.12%	8.87%	5.81%	0.03%
-35dB	2.87%	4.62%	2.99%	0.00%
-40dB	1.61%	2.40%	1.54%	0.00%
No Clustering	Baseline	Baseline	Baseline	Baseline

The results in Table 2.3 show that the threshold value has a significant impact on the performance of the network. For $\lambda_1 \leq -15\mathrm{dB}$, user-centric clustering becomes less efficient as it tends to serve all UEs with all APs, resulting increased power consumption without significant contribution to spectral efficiency. On the other hand, for $\lambda_1 \geq -15\mathrm{dB}$, only few APs cooperate, leading to reduced spatial diversity.

The performance of the proposed user-centric clustering algorithm under equal power allocation is evaluated in terms of energy efficiency, power consumption, spectral efficiency, and the number of active APs for different threshold λ_1 values and constant λ_2 that is set to 2. Simulation results are given in Table 2.4 compared to the Algorithm 1.

Table 2.4. Performance of the proposed algorithm relative to Algorithm 1

Threshold λ_1	EE Gain	Power Cons. Reduction	SE Loss	Active AP Reduction
-5dB	14.15%	22.40%	7.92%	24.13%
-10dB	28.27%	30.29%	9.62%	34.09%
-15dB	31.35%	24.61%	6.62%	30.04%
-20dB	17.73%	10.98%	2.47%	15.21%
-25dB	5.33%	2.65%	0.53%	4.17%
-30dB	1.11%	0.44%	0.08%	0.75%
-35dB	0.18%	0.06%	0.01%	0.11%
-40dB	0.03%	0.01%	0.00%	0.01%

The results in Table 2.4 shows that, by the proposed algorithm, spectral efficiency, number of active APs and total power consumption is further reduced compared to the Algorithm 1. Consequently, energy efficiency is also increased.

The energy efficiency improvement of the proposed algorithm is higher for the threshold values above -20dB and the highest energy efficiency improvement is obtained by -15dB threshold. However, highest active AP reduction is achieved by -10dB threshold. Threshold values higher than -10dB results lower UE-AP connections, which reduces the number of active APs that serves less than λ_2 UEs. Therefore, active AP reduction performance of the proposed algorithm relative to Algorithm 1 is reduced. Threshold values lower than -10dB results higher UE-AP connections, which increases the number of active APs that serves more than λ_2 UEs. Therefore, active AP reduction performance of the proposed algorithm is reduced.

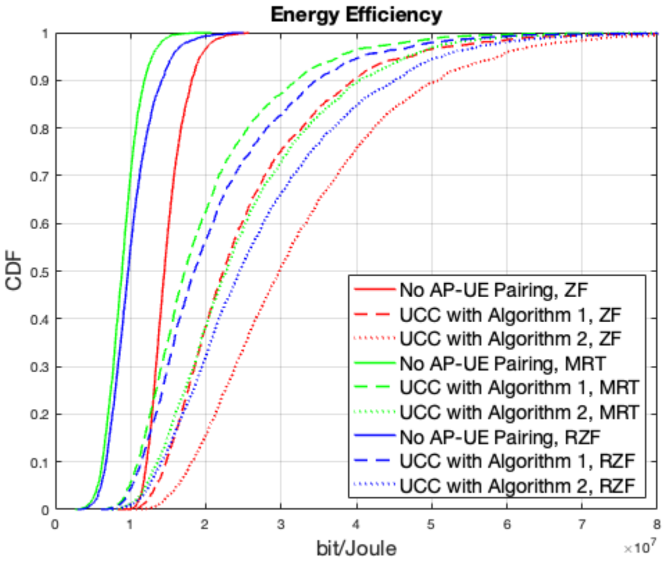


Figure 2.13. Energy efficiency comparison for different user-centric clustering algorithms and precoding schemes, for L=9, K=4, N=4.

The performance results for both Algorithm 1 and Algorithm 2 under different precoding schemes along with the no UE-AP pairing case are shown in Figure 2.13. The results indicates that the proposed algorithm outperforms the Algorithm 1 in terms of energy efficiency for all precoding schemes and ZF precoding has the best overall energy efficiency results among the three precoding schemes.

CHAPTER 3

MACHINE LEARNING BASED POWER ALLOCATION OPTIMIZATION

Machine learning is a rapidly growing technology that allows the systems to learn from provided data and make predictions. Machine learning has vast range of applications and it revolutionized various fields from health to military and communication to logistics. The fundamental concept behind the machine learning is that deep feedforward neural networks are proved to be universal function approximators, which means they can approximate functions as relations between input and output data without being explicitly programmed (Hornik et al., 1989). The main motivation of this chapter is to take advantage of both the prediction capability and low computational complexity of machine learning while performing user-centric clustering and finding the optimal power allocation coefficients for a centralized cell-free massive MIMO system.

The chapter begins with explaining the concept and the key components of deep neural networks. Then, an application of DNN to perform user-centric clustering and power allocation in cell-free massive MIMO systems is proposed. At the end, the performance evaluations of the proposed DNN based user-centric clustering and power allocation are provided through simulations.

3.1 Deep Neural Networks

Neural networks are a class of machine learning algorithms that are inspired by the human brain and they are well-known for the capability of learning complex patterns in data which makes them suitable for a wide range of applications. The neural networks that consists of multiple hidden layers are classified as deep neural networks. Each layer of a neural network consists of multiple neurons, which are the basic computational units of the neural networks. The representation of a neuron is shown in Figure 3.1. Each neuron has at least one connection to the neurons of the previous layer with a weight that determines the strength of the connection. The output of each neuron is computed by applying the activation function of the layer to the weighted sum of the inputs and the bias. The resulting output of each layer is then used as the input of the next layer.

Each layer in the neural network has its own activation function which can be determined according to the learning targets and data type. The first layer of the neural network is called as input layer, as it receives raw input data and the last layer is called

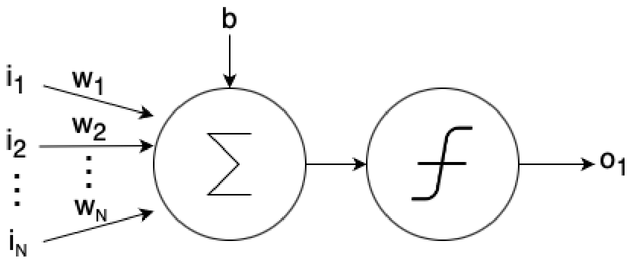


Figure 3.1. Visualization of a neuron.

the output layer, as it gives the resulting prediction. The layers in between are called hidden layers, as their outputs are not directly visible. The number of hidden layers and the number of neurons in each layer are hyperparameters that can be tuned to improve the performance of the model. The representation of a deep neural network is shown in Figure 3.2.

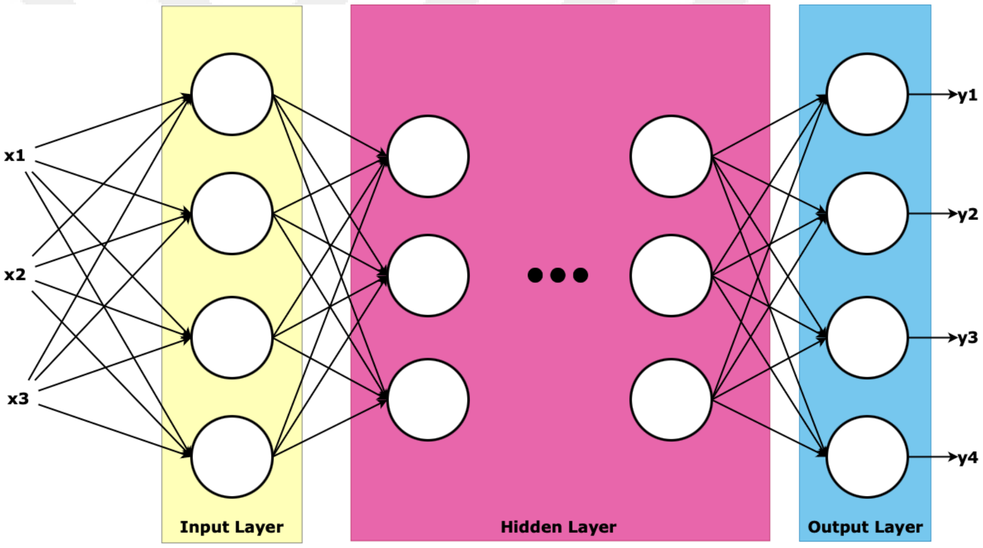


Figure 3.2. Visualization of a Deep Neural Network.

There are several types of deep neural networks, each with its own field of application due to their strengths and weaknesses. The most common types are:

1. Feedforward Neural Networks: These are the simplest type of DNN, where the data flows in one direction through the layers, without any loops or feedback connections. If outputs of each neuron in the previous layer is connected to the input of each neuron of the next layer, it is called fully connected neural network. They are commonly used for classification and regression tasks, where the goal is to predict the class or value of an input based on its features.

2. Convolutional Neural Networks: These are designed specifically for image recognition tasks, where the goal is to classify images based on their content. They use

a special type of layer called a convolutional layer. This layer uses a small filter and computes the dot product between the filter and the input pixels by sliding the filter over the input image. This process results a feature map that captures the presence of certain patterns in the image.

3. Recurrent Neural Networks: These are designed for sequential data, where the order of the inputs matters. They have a feedback connection that allows them to use the information from previous inputs while making the predictions about the current input. These systems are well-suited for the tasks with a sequence like speech recognition and natural language processing.

3.1.1 Batch Size & Epoch Number

The size of each batch is a hyperparameter that is set before the beginning of training process. It determines how many samples from the dataset will be used in each iteration of training process. The resulting loss values of each sample in the batch is averaged and the backpropagation is performed once for each batch with the resulting average. A larger batch size can speed up the training process, as more samples are processed in parallel, but it requires more memory and may result slower convergence since it averages many samples and update weights less often. On the other hand, a smaller batch size may result in slower training but can lead to better generalization since it updates the parameters more frequently with smaller and more diverse samples.

Epoch number is another hyperparameter that determines the number of times the DNN repeats the entire training dataset. Each epoch consists of one pass through the entire dataset. Increasing the number of epochs can improve the accuracy of the DNN, but it can also lead to overfitting. Overfitting is a common problem in machine learning where the trained model memorize the training data instead of learning the underlying mapping and consequently have poor generalization to unseen data. Easiest way to determine overfitting is to check training and validation accuracy. If the model has high training accuracy but low validation accuracy, it means that the model is memorizing the training data rather than learning the underlying patterns that generalize to new data. In this thesis, batch randomization is used so that in each epoch, the batches will have different training samples. Therefore, averaging on error will performed over different set of samples and may reduce the overfitting probability.

In practice, the optimal batch size and epoch number depends on the dataset features and architecture of the neural network, and is often chosen by trial and error.

3.1.2 Learning Rate

The learning rate is a hyperparameter that controls the step size of the DNN while updating the weights in response to the loss during training. The effect of the learning rate is given in Figure 3.3.

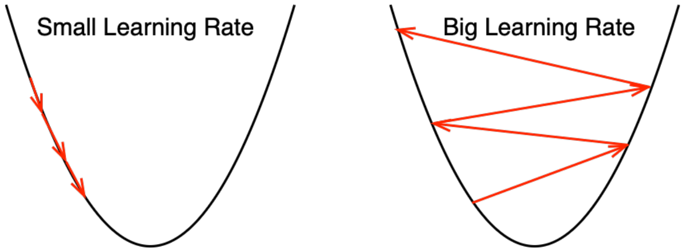


Figure 3.3. Effect of the learning rate.

A high learning rate may cause the DNN to overshoot the optimal weights and bounce around, while a low learning rate may cause the DNN to take a long time to converge to the optimal weights. In most DNN applications, variable learning rate technique is used where the learning rate is set to a higher value at the beginning of training and reduced gradually over epochs. As in batch size and epoch number, the optimal learning rate depends on the specific DNN architecture and the dataset being used, and is typically determined through trial and error.

3.1.3 Training Process

The main feature of deep neural networks is their ability to learn complex patterns from data. The training process of a DNN can be simply explained as minimizing the difference between the predicted output and the actual output given in the training data by iteratively adjusting the parameters of the model.

The training process starts with randomly initializing the weights and biases of the neural network. The input data is then passed through the network, and the predicted output is compared to the actual output. The difference between the predicted output and the actual output is measured by a loss function, which quantifies the error of the model. Then, the weights and biases of the neural network is adjusted according to the resulting loss.

Forward Pass

The forward pass is also referred as the forward-propagation is the process of calculating the output of the neural network for given input data. The input data is fed through the input layer and the output of the layer is calculated by applying the activation function the layer to the weighted sum of the inputs. The output of layer is then used as the input of the next layer. This process is applied layer by layer until the output layer is reached. Output of each neuron can be represented as

$$o_j^l = f\left(\sum_{j=0}^{J} w_{i,j}^l \, o_j^{l-1} + b_j^l\right) \tag{3.1}$$

where f is the activation function, o_j^l is the output of the j^{th} neuron of l^{th} layer, $w_{i,j}^l$ is the weight between i^{th} neuron of previous layer and j^{th} neuron of l^{th} layer and b_j^l is the bias of the j^{th} neuron of l^{th} layer.

Backpropagation

The backpropagation is an iterative process that enables the learning of deep neural networks during training phase. The process starts from the last layer, computes the gradients of the network parameters such as weights and biases with respect to a given loss function and propagates the gradients backward through the neural network. The gradients are then used by an gradient-based optimizer to update the parameters of the network in order to minimize the loss function. This process is repeated at the end of each batch by using the gradients of the average loss with respect to each parameter.

3.2 Loss Functions

One of the key component of training a deep neural network is defining a loss function, which measures the performance of the neural network on a given task. The choice of loss function depends on the specific task and the nature of the data. The loss function is used to calculate the error between the predicted power allocation coefficients and the actual power allocation coefficients.

In literature, there are many different loss functions that can be used in deep neural networks. However, not all loss functions are appropriate for all types of problems. Different problems may require different loss functions to accurately capture the nature of the data and optimize the model's performance. For example, some loss functions may be more suitable for classification problems where the goal is to assign input data to discrete categories, while others may be better suited for regression problems where the goal is to predict continuous values. In this section, loss functions that are commonly used for regression problems will be presented.

Mean Squared Error (MSE)

Mean squared error (MSE) is a commonly used loss function for regression problems. It is calculated by taking the average of the squared difference between the predicted value and the actual value for each predicted output. The MSE loss function is defined as follows:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{o}_i - o_i)^2 \tag{3.2}$$

where n is the number of predicted values at output, o_i is the actual value and $\hat{o}_i$ is the predicted value.

The MSE loss function is convex and continuously differentiable function, which makes it easy to optimize using gradient-based methods such as gradient descent. However, MSE is highly sensitive to outliers, if predicted value is significantly different from the target value, the squared loss will increase significantly and can lead to overfitting if the model is too complex. Throughout this thesis, MSE loss function is used to train the neural network since the dataset does not contain any outliers and MSE is well-suited to use with gradient-based methods.

Mean Absolute Error (MAE)

Mean absolute error (MAE) is another commonly used loss function for regression problems. It is calculated by taking the average of the absolute difference between the predicted value and the actual value for each predicted output. The MAE loss function is defined as follows:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|o_i - \hat{o}_i| \tag{3.3}$$

The MAE loss function is used as an alternative to MSE when the dataset contains large number of outliers. It is less sensitive to outliers than MSE since it does not square the difference between the predicted value and the actual value. However, MAE is not differentiable at zero and does not perform properly as the average distance approaches to zero. Therefore, it becomes more difficult to optimize using gradient-based methods. Also the gradient of MAE stay same throughout the losses, this means the gradient will be large even for small loss values and may result in poor learning performance due to overshooting. To fix this problem, variable learning rate can be used.

Huber Loss

Huber loss is a loss function that combines MSE and MAE. It is less sensitive to outliers than MSE and less sensitive to small errors than MAE. Huber loss is defined as follows:

$$\mathrm{HL} = \begin{cases} \frac{1}{n}\sum\limits_{i=1}^{n}(o_i - \hat{o}_i)^2 & \text{for } |o_i - \hat{o}_i| \leq \delta \\ \frac{1}{n}\sum\limits_{i=1}^{n}\delta(|o_i - \hat{o}_i| - \frac{1}{2}\delta) & \text{otherwise} \end{cases} \tag{3.4}$$

where δ is hyperparameter that controls the threshold at which the loss function changes from quadratic to linear. Huber loss is quadratic for small values of $(o_i - \hat{o}_i)$ and linear for large values of $(o_i - \hat{o}_i)$. The hyperparameter δ is usually set to 1, which means that the loss function is quadratic for $|o_i - \hat{o}_i| \leq 1$ and linear for $|o_i - \hat{o}_i| > 1$.

Root Mean Squared Error (RMSE)

RMSE is a commonly used loss function for regression problems. It is very similar to MSE loss function and is calculated by taking the square root of the average of the squared difference between the predicted value and the actual value for each predicted output. Like MSE, the RMSE loss function is continuously differentiable and convex, which makes it easy to optimize using gradient-based methods. The RMSE loss function is defined as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(o_i - \hat{o}_i)^2} \tag{3.5}$$

The main difference between MSE and RMSE is square root term. Taking the square root ensure that the loss is in the same unit as the data, which makes it easier to interpret. However, square root term in RMSE reduces the impact of regular prediction errors relative to outliers on loss and makes it less robust to outliers than MSE.

Log-Cosh Loss

Log-cosh loss is another loss function that is commonly used for regression problems. It is calculated by taking the average of the logarithm of the hyperbolic cosine of the difference between the predicted value and the actual value for each predicted output. The log-cosh loss function is defined as follows:

$$\text{LCL} = \frac{1}{n}\sum_{i=1}^{n}\log(\cosh(\hat{o}_i - o_i)) \tag{3.6}$$

Even though it is still sensitive to outliers, the log-cosh loss function is more robust than MSE and MAE. Furthermore, unlike the Huber loss function, log-cosh is also continuously differentiable.

3.3 Activation Functions

Activation functions are one of the most critical components of a neural network as they introduce non-linearity into the model that allows the neural network to learn complex patterns and relationships in the data. In addition to introducing non-linearity, activation functions also help to smooth the gradient during backpropagation. This is particularly important in deep neural networks with many layers, where the gradient can quickly become unstable and cause the model to fail to converge.

Each activation function has its own set of advantages and disadvantages which may impact the performance of the neural network significantly. Therefore, it is important to choose the right activation function for the specific problem at hand. In this section, we will discuss some of the most commonly used activation functions and their characteristics for our problem.

Linear Activation Function

The linear activation function is one of the simplest activation functions, defined as

$$f(x) = x \tag{3.7}$$

and the derivative of the linear activation function is

$$f'(x) = 1 \tag{3.8}$$

It is a linear function that does not introduce any non-linearity into the model, and its output is proportional to its input. The linear activation function is primarily used in output layers of regression problems, where the goal is to predict a continuous value. In these problems, the output of the neural network needs to be a linear combination of the input features, so a linear activation function is suitable for this task. In most cases, the linear activation function is not used in deep neural networks except the output layer because it does not allow the model to learn complex patterns and relationships in the data.

Sigmoid Activation Function

The sigmoid activation function is one of the most commonly used activation function in neural networks. It is a non-linear function that maps any input value to a value between 0 and 1, and is defined as follows:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{3.9}$$

The derivative of the sigmoid function is

$$f'(x) = f(x)(1 - f(x)) \tag{3.10}$$

where x is the input to the neuron. The output of the sigmoid function is always in the range of [0, 1]. This makes the sigmoid function well-suited for applications where the predicted value represents a probability or a binary decision. Therefore, it is often used in the output layer of binary or multi-class classification problems where the output needs to be a probability value for each class. However, the sigmoid function has two major drawbacks. First one is that its output saturates at either 0 or 1 for large positive or negative input values, which may cause the gradients during training to become very small, leading to slow convergence or vanishing gradients. This phenomenon is known as the vanishing gradient problem. Second issue with the sigmoid function is that it is not zero-centered, which can lead to slower convergence during training. To address these issues, other activation functions such as the Rectified Linear Unit (ReLU) and its variants have been invented and widely used in practice.

ReLU Activation Function

The Rectified Linear Unit (ReLU) function is one of the most popular activation functions used in deep neural networks today and defined as

$$f(x) = max(0, x) \tag{3.11}$$

and the derivative of the ReLU function is

$$f'(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases} \tag{3.12}$$

where x is the input to the neuron.

As seen in Equation (3.11), the ReLU function is a piecewise linear function that gives the same value for the positive inputs, and outputs zero if it is negative. Since it is a non-linear function, it allows the neural network to learn complex patterns and relationships between the input and output. ReLU is computationally efficient and helps to alleviate the vanishing gradient problem. However, it can suffer from the problem of "dead neurons", where a neuron has large negative bias and therefore, always outputs zero. This constant zero output leads to a zero gradient and no learning. This problem can be mitigated by using variants of the ReLU function, such as the leaky ReLU or the parametric ReLU, which allow some output for negative input values.

Leaky ReLU Activation Function

The Leaky ReLU function is a modified version of the ReLU function that avoids the problem of dead neurons. It introduces a small positive slope for negative input values, allowing neurons to activate even when the input is negative and defined as follows:

$$f(x) = max(\alpha x, x) \tag{3.13}$$

and the derivative of the Leaky ReLU function is

$$f'(x) = \begin{cases} 1 & \text{if } x > 0 \\ \alpha & \text{if } x \leq 0 \end{cases} \tag{3.14}$$

where alpha is a hyperparameter that controls the slope of the negative part of the function. The value of α is typically set to a small value such as 0.01.

Tanh Activation Function

The Tanh function is a non-linear function that maps the input values to output values between -1 and 1. The Tanh function is defined as:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{3.15}$$

and the derivative of the Tanh function is

$$f'(x) = 1 - f^2(x) \tag{3.16}$$

Unlike the sigmoid function, the Tanh function is symmetric around the origin, which means that it can output both positive and negative values, making it suitable for tasks such as classification where the output may be either positive or negative. Furthermore, the tanh function also has a steeper gradient compared to the sigmoid function, which can help the neural network learn more quickly during the initial stages of training. However, like the sigmoid function, it is also susceptible to the vanishing gradient problem.

ELU Activation Function

The Exponential Linear Unit (ELU) is an activation function that was introduced as an alternative to the Rectified Linear Unit (ReLU) activation function. The ELU function is similar to the ReLU function in that it is also piecewise linear, with a linear slope for positive values of x. However, it differs from ReLU in that it has a smooth and non-zero slope for negative values of x, which can help alleviate the dead neurons problem. The ELU function is defined as follows:

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha(e^x - 1), & \text{if } x \leq 0 \end{cases} \tag{3.17}$$

and the derivative of the ELU function is

$$f'(x) = \begin{cases} 1, & \text{if } x > 0 \\ f(x) + \alpha, & \text{if } x \leq 0 \end{cases} \tag{3.18}$$

where α is a hyperparameter that controls the slope of the negative part of the function. The ELU function has been shown to improve the performance of neural networks on a variety of tasks, including image classification and speech recognition. It is also computationally efficient, with similar or better performance compared to other activation functions such as ReLU and its variants. Overall, the ELU activation function is a promising alternative to the widely used ReLU function, providing a smooth and non-zero slope for negative inputs and helping to alleviate the "dying ReLU" problem during training.

Softmax Activation Function

The Softmax activation function is a commonly used activation function in neural networks, especially for classification tasks with multiple classes. It is a generalization of the sigmoid function and maps the input values to a probability distribution over the output classes that sums to one.Each output represents the probability of the input belonging to a particular class. The Softmax function is defined as:

$$f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{k} e^{x_j}}, \text{for all } j = 1, 2, \ldots, k \tag{3.19}$$

and the derivative of the Softmax function is

$$f'(x_i) = f(x_i)(1 - f(x_i)), \text{for all } j = 1, 2, \ldots, k \tag{3.20}$$

where k is the total number of classes, and x_i is the input value corresponding to the i^{th} class.

The main disadvantage of the Softmax function is that it can be sensitive to outliers in the input values, which can cause numerical stability issues during computation. To address this, techniques such as scaling or shifting the input values are often used.

Swish Activation Function

The Swish function is a relatively new activation function in literature that has shown promising results in deep neural networks. It was introduced by researchers at Google in 2017 and has been shown to improve the performance of neural networks on certain tasks (Ramachandran et al., 2017). The Swish function is defined as:

$$f(x) = x \text{ sigmoid}(\beta x) = \frac{x}{1 + e^{-\beta x}} \tag{3.21}$$

and the derivative of the Swish function is

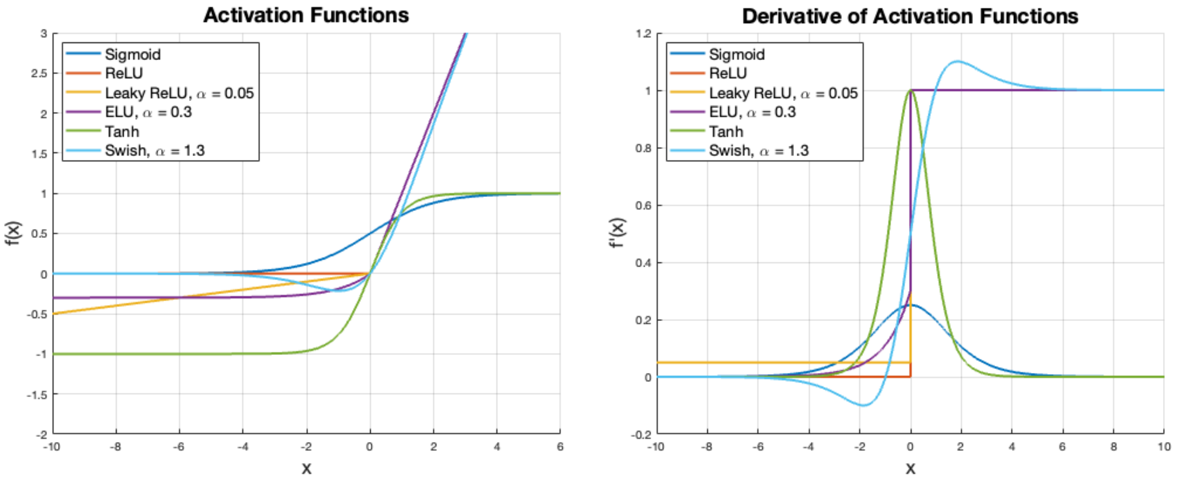
$$f'(x) = \beta f(x) + \frac{1}{1 + e^{-\beta x}}(1 - \beta f(x)) \tag{3.22}$$

where beta is a scalar parameter that controls the shape of the function. When beta = 1, the Swish function reduces to a smoothed version of the ReLU function.

The Swish function has been shown to perform well on a variety of tasks, including image classification, language modeling, and speech recognition. It has been suggested that the Swish function is particularly effective when used in the intermediate layers of a neural network, where it can help to introduce non-linearity and improve the expressive power of the network.

One potential disadvantage of the Swish function is that it can be computationally expensive compared to other activation functions such as the ReLU. However, recent advances in hardware and software optimizations have made it possible to use the Swish function in practical applications.

The activation functions in deep neural networks, including the sigmoid, tanh, ReLU, Leaky ReLU, ELU and Swish function and their derivatives are shown in Figure 3.4 where the hyperparameters are chosen to obtain distinguishable lines on graph.



(a) Activation function graph.

(b) Derivative of activation functions graph.

Figure 3.4. Activation Functions.

Each of these functions has its own advantages and disadvantages, and there is no single activation function that is best for all tasks. The choice of activation function depends on the task and the type of neural network being used. The reason of the mentioned vanishing gradient problem can be seen in Figure 3.4b where the derivative of the sigmoid function is close to zero for most of the input values, which can cause the gradient to vanish during backpropagation.

3.4 Optimizers

Optimizers play a crucial role in training deep neural networks. The goal of an optimizer is to minimize the loss function of the DNN by updating the parameters of the network. The optimization problem is typically non-convex, high-dimensional, and noisy, making it challenging to find the global minimum of the loss function.

There are several optimization algorithms used in DNNs, and the choice of optimizer can have a significant impact on the training process and the performance of the model. In this section, we will discuss some of the most commonly used optimizers in DNNs.

Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent (SGD) is a widely used optimization algorithm in DNNs. It updates the weights of the model based on the gradient of the loss function with respect to the weights. The update rule involves multiplying the gradient by a learning rate and subtracting it from the current weight. The learning rate determines the step size of the update and is typically set to a small value. SGD is a simple algorithm that is easy to implement and work well for small datasets. However, it can be slow to converge and may get stuck in local minima for larger datasets or more complex models and it requires careful tuning of the learning rate to prevent oscillations or divergence.

Adagrad

Adagrad is classified as adaptive optimization algorithm since it adjusts the learning rate based on the historical gradient information. It accumulates the squared gradient over time and scales the learning rate accordingly. Adagrad is well-suited for sparse data and can handle different scales of gradients. However, Adagrad can be sensitive to the initial learning rate and can suffer from diminishing learning rates over time, which can slow down the convergence of the model.

Adadelta

Adadelta is a variant of Adagrad that seeks to reduce the aggressive and monotonically decreasing learning rate of Adagrad. It uses a sliding window to keep a running average of the gradients and updates the learning rate based on this average. Adadelta does not require an initial learning rate and can handle different scales of gradients.

RMSprop

RMSprop is another adaptive optimization algorithm that scales the learning rate by a running average of the squared gradient. It is similar to Adagrad but uses an exponentially decaying average of the gradient squared instead of accumulating all past squared gradients. RMSprop is less sensitive to the initial learning rate than Adagrad and can be more robust to noisy gradients. However, as in SGD, it can also suffer from oscillations or divergence if the learning rate is not properly tuned.

Adam

Adam is one of the most popular optimization algorithms that computes adaptive learning rates for each parameter. It combines the benefits of both AdaGrad and RMSProp by adapting the learning rate based on the first and second moments of the gradients. Furthermore, it maintains a moving average of the gradient and the squared gradient, which are then used to compute the adaptive learning rates. Adam is an efficient and robust optimization algorithm that can handle noisy or sparse gradients. It is also computationally efficient and has been shown to converge faster than other optimization algorithms on many datasets. However, sometimes it may overfit the training data, especially for small datasets and it requires more memory than SGD.

The Adam algorithm is given in Algorithm 3 (Kingma and Ba, 2014) where the $\boldsymbol{g}_t$ is the gradient of the loss function with respect to the parameters $\boldsymbol{\theta}$ at time step t, ϵ is a small constant to prevent division by zero, and λ is the learning rate. The first moment vector $\boldsymbol{m}_t$ is an exponentially decaying average of the gradients, and the second moment vector $\boldsymbol{v}_t$ is an exponentially decaying average of the squared gradients. The first and second moment vectors are then used to compute the adaptive learning rates $\hat{\boldsymbol{m}}_t$ and $\hat{\boldsymbol{v}}_t$, which are then used to update the parameters $\boldsymbol{\theta}_t$ at time step t.

In this thesis, we used Adam optimizer to train the neural network since it combines the benefits of both AdaGrad and RMSProp and has been shown to perform well on a variety of tasks. Adam optimizer is used with default parameters, which are $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$.

```
Algorithm 3: Adam Optimizer (Kingma and Ba, 2014)
  Input: Learning rate $\lambda$, decay rate for the first moment estimates $\beta_1$, decay rate for
         the second moment estimates $\beta_2$
  Initialization: Set first moment vector $\boldsymbol{m}_0 = \mathbf{0}$, second moment vector $\boldsymbol{v}_0 = \mathbf{0}$,
   time step t = 0
  foreach batch do
      $t \leftarrow t + 1$
      Compute gradients $\boldsymbol{g}_t$
      $\boldsymbol{m}_t \leftarrow \beta_1 \boldsymbol{m}_{t-1} + (1 - \beta_1)\boldsymbol{g}_t$
      $\boldsymbol{v}_t \leftarrow \beta_2 \boldsymbol{v}_{t-1} + (1 - \beta_2)\boldsymbol{g}_t^2$
      $\hat{\boldsymbol{m}}_t \leftarrow \frac{\boldsymbol{m}}{1-\beta_1^t}$
      $\hat{\boldsymbol{v}}_t \leftarrow \frac{\boldsymbol{v}}{1-\beta_2^t}$
      $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta} - \lambda \frac{\hat{\boldsymbol{m}}}{\sqrt{\hat{\boldsymbol{v}}+\epsilon}}$
  end
```

3.5 Data Scaling

Data scaling is a data transformation technique that is often used to improve the performance of machine learning algorithms by fitting the data in a consistent scale without distorting the differences in the range of values. Scaling can help to stabilize the training process by preventing the gradients from becoming too large or too small. This can help to prevent the model from overfitting or underfitting the training data. Additionally, scaling also helps to ensure that all of the samples are on a similar scale and making the model more robust to noise and outliers in the data. In this section, we will discuss different normalization techniques and their effects on the performance of DNNs.

Mean Normalization

First and simplest scaling technique given in Equation (3.23) is called mean normalization by subtracting the mean of the data from each data point.

$$\boldsymbol{\beta}_1 = \boldsymbol{\beta} - \mu \tag{3.23}$$

where μ is the mean of $\boldsymbol{\beta}$. Since all the data points are shifted by the same amount, the mean of the normalized data is zero and the variance of the normalized data is equal to the variance of the original data.

Min-Max Normalization

Second scaling technique given in Equation (3.24) is called min-max normalization by scaling the data to a specific range. By subtracting the minimum value of the data, it ensures the data samples will have values equal or greater than zero. And dividing by the range of the data, the samples will have values equal or less than one.

$$\boldsymbol{\beta}_2 = \frac{\boldsymbol{\beta} - \min(\boldsymbol{\beta})}{\max(\boldsymbol{\beta}) - \min(\boldsymbol{\beta})} \tag{3.24}$$

By this normalization, the data is scaled to the range of $0 \leq \boldsymbol{\beta}_2 \leq 1$. However, the mean and variance of the normalized data are also scaled and different from the original data.

Standardization

Last scaling technique considered in the thesis given in Equation (3.25) is called standardization by scaling the data to have zero mean and unit variance. By subtracting the mean of the data and dividing by the standard deviation of the data, it ensures the data samples will have zero mean and unit variance.

$$\boldsymbol{\beta}_3 = \frac{\boldsymbol{\beta} - \mu}{\sigma} \tag{3.25}$$

where μ is the mean of $\boldsymbol{\beta}$ and σ is the standard deviation of $\boldsymbol{\beta}$.

The mentioned scaling techniques are applied to the channel gain β given in Section 2.2. In each case, the normalized data has the same distribution type as the original data. However, the mean and variance of the normalized data are different from the original data.

3.6 Machine Learning Based Resource Allocation Approach

In this section, the proposed DNN architectures for user-centric clustering and power allocation are introduced and the dataset preparation process is explained.

The block diagram of the proposed machine learning based approach is given in Figure 3.5. The input data is the channel gains between the APs and UEs. The output of the user-centric clustering part is passed to the power allocation part along with channel gains. The output of the power allocation part is the power allocation coefficients for the active AP-UE pairs.

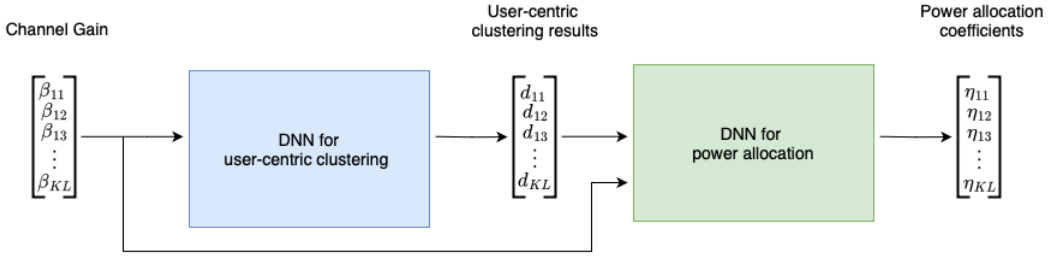

Channel Gain
β_{11} β_{12} β_{13} $\vdots$ β_{KL}
DNN for user-centric clustering
User-centric clustering results
d_{11} d_{12} d_{13} $\vdots$ d_{KL}
DNN for power allocation
Power allocation coefficients
η_{11} η_{12} η_{13} $\vdots$ η_{KL}

Figure 3.5. Block diagram of the proposed machine learning based resource allocation approach for cell-free massive MIMO systems.

3.6.1 Model Architecture

The DNN architecture is designed in two parts as user-centric clustering and power allocation. The user-centric clustering part is a binary classifier that determines the state of UE-AP pairs as active or inactive. The power allocation part is a regression model that predicts the power allocation coefficients for the active UE-AP pairs. These parts are trained separately and then cascaded to each other. The first training process will be more time consuming since it will be done in two separate DNNs. However, in this architecture, user-centric clustering algorithm can be changed by training a new user-centric clustering DNN in parallel and replacing only that part instead of re-training whole network. Therefore, it is more modular and time efficient in long run. Also, the loss function and optimizer can be assigned independently for each part, which makes the architecture more controllable.

The proposed architecture also reduces the complexity of the problem since user-centric clustering part performs only binary predictions and the power allocation part considers only active AP-UE pairs while making regression predictions. The most significant advantage of this architecture is that it reduces the number of layers by dividing the network into two parts. This helps avoiding vanishing gradient problem since each network will be more shallow. For both parts, the number of hidden layers are determined empirically by trial and error. The final number of hidden layers are chosen heuristically.

DNN Approach for User-Centric Clustering

The DNN for user-centric clustering is a binary classifier that determines the state of UE-AP pairs as active or inactive. The DNN is designed to be fully connected DNN that has an input layer with 360 neurons, 7 hidden layers and an output layer with 36 neurons. The input data consists of the channel gains for each AP-UE pairs. Each output of the

360 neurons at the input layer corresponds to the linear combinations of the channel gains between the AP-UE pairs with trainable weights. Each output of the 36 neurons at the output layer corresponds to the user-centric clustering decision as a binary state for the AP-UE pairs.

The activation function of input layer is chosen as linear activation function to expand the data with linear combinations and passing these combinations to hidden layers. The activation functions of the hidden layers are determined as ReLU, Tanh, and ELU by trial and error.

The output of the user-centric clustering is either 0 or 1. Since the Sigmoid activation function maps the input values between 0 and 1, it is chosen as the activation function of the output layer. Since the Sigmoid function is used as the loss function of output, the network is designed to have low amount of hidden layers such as 7 to prevent vanishing gradient problem and overfitting.

The DNN architecture for user-centric clustering is given in Table 3.1.

Table 3.1. DNN architecture for user-centric clustering

Layer	Size	Activation Function
Input	360	Linear
Layer 1	360	ReLU
Layer 2	360	Tanh
Layer 3	360	ELU
Layer 4	180	Tanh
Layer 5	180	ELU
Layer 6	180	ReLU
Layer 7	180	Tanh
Output	36	Sigmoid

DNN Approach for Power Allocation

The DNN for power allocation is a regression model that predicts the power allocation coefficients for the active UE-AP pairs. The DNN is designed to be fully connected DNN that has an input layer with 360 neurons, 5 hidden layers and an output layer with 36 neurons. The input data consists of the channel gains for the active AP-UE pairs and zeros for deactivated AP-UE pairs. Each output of the 360 neurons at the input layer corresponds to the combinations of the channel gains between the AP-UE pairs with trainable weights and each output of the 36 neurons at the output layer corresponds to the predicted power allocation coefficients for the AP-UE pairs.

The activation function of input layer is chosen as linear activation function as in

DNN of user-centric clustering to expand the data with linear combinations and passing these combinations to hidden layers. The activation functions of the hidden layers are chosen as ELU to prevent dead neurons problem and the vanishing gradient problem. The activation function of the output layer is chosen as Sigmoid since it maps the input values between 0 and 1 and the power allocation coefficients are defined to have values between the same interval. As in user-centric clustering part, the network is designed to have low amount of hidden layers such as 5 to prevent vanishing gradient problem and overfitting. The resulting DNN architecture for power allocation is given in Table 3.2.

Table 3.2. DNN Architecture for Power Allocation

Layer	Size	Activation Function
Input	360	Linear
Layer 1	360	ELU
Layer 2	360	ELU
Layer 3	180	ELU
Layer 4	180	ELU
Layer 5	180	ELU
Output	36	Sigmoid

3.6.2 Dataset Preparation

Training of the DNN for user-centric clustering and power allocation with high accuracy requires a large dataset. To generate dataset, simulations are performed over 10^6 randomly generated setups using the same system model and procedure as in Section 2.6 for Algorithm 1 and the proposed algorithm. In the cases where the optimization is failed to reach local minima, the power allocation coefficients are set as equal power allocation. The channel gains of the generated setups, the user-centric clustering results and the corresponding power allocation coefficients are combined to form a dataset. Since the power allocation coefficients and user-centric clustering decisions are valued between 0 and 1, we use min-max normalization on average channel gains to scale them into the same range as discussed in Section 3.5.

The resulting datasets are split into training, validation and test sets as 90%, 5% and 5% respectively. The training set is used to train the DNN, the validation set is used to tune the hyperparameters of the DNN such as learning rate and batch size and the test set is used to evaluate the performance of the DNN based approach for user selection and power allocation.

3.7 Performance Evaluations

In this section we provide the performance of the proposed DNN approach for user-centric clustering and power allocation in terms of the energy efficiency and required computation time.

3.7.1 DNN Results for User-Centric Clustering

The performance of the proposed DNN approach for user-centric clustering is evaluated in terms of the energy efficiency and the decision error that is defined as the percentage of the AP-UE pairs that are misclassified by the DNN. Given that the user-centric clustering results are restricted to have values 0 or 1 and the input data is normalized to eliminate outliers as discussed in Section 3.5, the robustness to outliers is not considered in this thesis. The appropriate loss function is chosen as MSE. The training set is randomly split into mini-batches of size 500 at each epoch to speed up the training process which is performed using the Adam optimizer over 250 epochs with a learning rate of 0.001. In post-processing, the predicted values are set as 0 or 1 to obtain the binary state of the AP-UE pairs by using threshold 0.5.

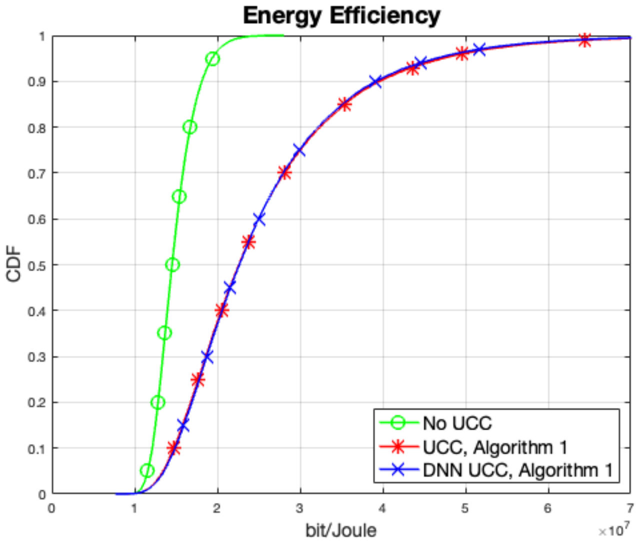


Figure 3.6. Energy efficiency of DNN based user-centric clustering using Algorithm 1, for K=4, L=9, N=4.

The proposed DNN architecture for user-centric clustering is trained using the training set and the resulting decision error for the user-centric clustering algorithm in Algorithm 1 is calculated as 1.58% while the resulting decision error for the proposed user-centric clustering algorithm is calculated as 1.46%. These error percentages indicates that the DNN to misclassifies an AP-UE pair approximately in every two setups. The difference between the decision error percentages of the Algorithm 1 and the proposed user-centric clustering algorithm is that the proposed user-centric clustering algorithm results less active AP-UE pairs and therefore the DNN needs to learn less relationships on top of predicting the pairs as inactive.

The CDF graphs of the energy efficiency under equal power allocation for Algorithm 1 and DNN trained by Algorithm 1 are given along with the case that all UEs are served by all APs in Figure 3.6. It can be seen that, the proposed DNN approach for user-centric clustering achieves almost same energy efficiency performance as the Algorithm 1 and the proposed algorithm. The average energy efficiency for the proposed DNN trained by Algorithm 1 is 0.326% lower than the Algorithm 1 which is negligible in most use cases.

The CDF graphs of the energy efficiency under equal power allocation for the proposed user-centric clustering algorithm using the proposed algorithm and DNN approaches are given along with the case that all UEs are served by all APs in Figure 3.7.

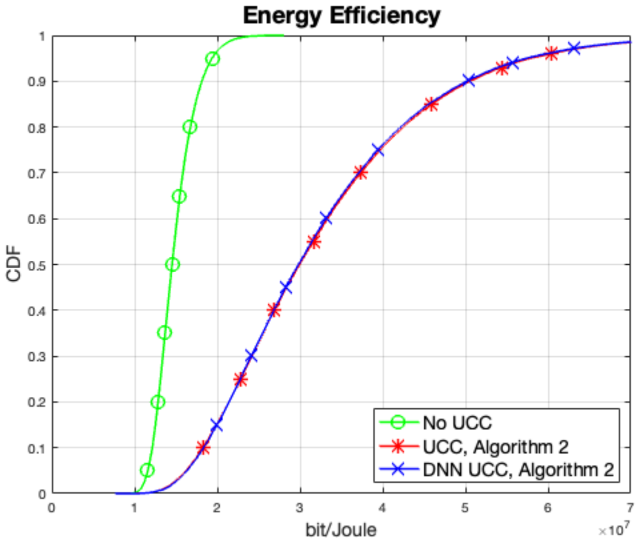


Figure 3.7. Energy efficiency performance of DNN based user-centric clustering using the Algorithm 2, for K=4, L=9, N=4.

The Figure 3.7 indicates that the proposed DNN approach for user-centric clustering achieves almost the same energy efficiency performance as the Algorithm 2. The average energy efficiency for the proposed DNN trained by Algorithm 2 is 0.523% lower than the Algorithm 2 which is also negligible in most use cases.

Even though the proposed user-centric clustering algorithm achieves better in terms of the decision error, the impact of the decision error on energy efficiency is higher than the Algorithm 1. This can be explained by the fact that the proposed user-centric clustering algorithm results less active UE-AP pairs and therefore, each misclassified UE-AP pair has higher chances to activate a deactivated AP and therefore has more impact on the energy efficiency performance.

3.7.2 DNN Results for Power Allocation

The performance of the proposed DNN approach for power allocation is evaluated in terms of the achieved energy efficiency and the computational time required to find power allocation coefficients. As in the user-centric clustering part, the power allocation part is also trained using the Adam optimizer over 250 epochs with a learning rate of 0.001. The training process is performed using the MSE loss function and the training set is randomly split into mini-batches of size 500 at each epoch to speed up the training process.

The time requirements are measured in terms of the time passed during the process in seconds and given in three parts. First is the time required for power allocation using MATLAB's fmincon solver with 5 parallel workers on a computer with Apple M1 Pro processor and 16GB RAM. Second is the time required for predicting the power allocation coefficients using the trained DNN on a computer with NVIDIA GTX 1060 GPU and 16GB RAM. The last one is the time required for training the DNN on the same computer setup.

The average time required for training the DNN using the dataset contains 900000 setups is measured as 1971 seconds. On average, the power allocation requires 0.54 seconds per setup for Algorithm 1 and 0.78 seconds per setup for the proposed user-centric clustering algorithm while the DNN approach requires 0.0001 seconds per setup for both algorithms. Equal time requirements for both user-centric clustering algorithms means that the proposed DNN approach eliminates the time requirement disadvantage of the proposed user-centric clustering algorithm. Also the results indicates that the DNN approach reduced the required computational time by 99.9%. However, the time required for training the DNN is not included in this comparison since it is a one-time process and the trained DNN can be used for multiple setups.

The energy efficiency performance of the proposed DNN approach for power allocation is evaluated seperately for Algorithm 1 and the proposed user-centric clustering algorithms. The simulation results for Algorithm 1 are given in Figure 3.8 as equal power allocation, power allocation using Algorithm 1 and DNN based power allocation. As it can be seen in the Figure 3.8, the proposed DNN power allocation achieves almost the same performance in terms of energy efficiency using Algorithm 1. The achieved average energy efficiency for the proposed DNN approach is 0.28% lower than the Algorithm 1 which is considered to be negligible in most use cases.

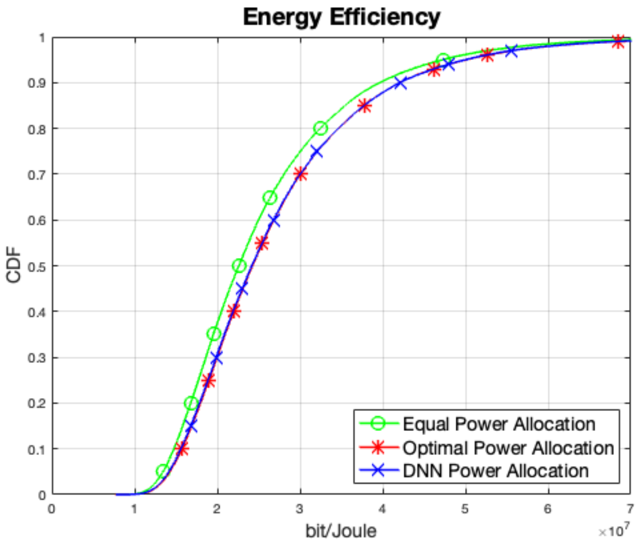


Figure 3.8. Energy efficiency of the DNN for power allocation using Algorithm 1, for K=4, L=9, N=4.

Furthermore, the simulations are shown for the proposed user-centric clustering algorithm and the equal power allocation, power allocation using Algorithm 2 and DNN power allocation in Figure 3.9. The power allocation performance for the DNN approach is almost the same as the Algorithm 2, with less than 0.01% reduction in energy efficiency. The energy efficiency difference between the Algorithm 2 and DNN approach using Algorithm 2 is considerably lower than the one using Algorithm 1, which indicates that the proposed user-centric clustering algorithm achieves better energy efficiency performance.

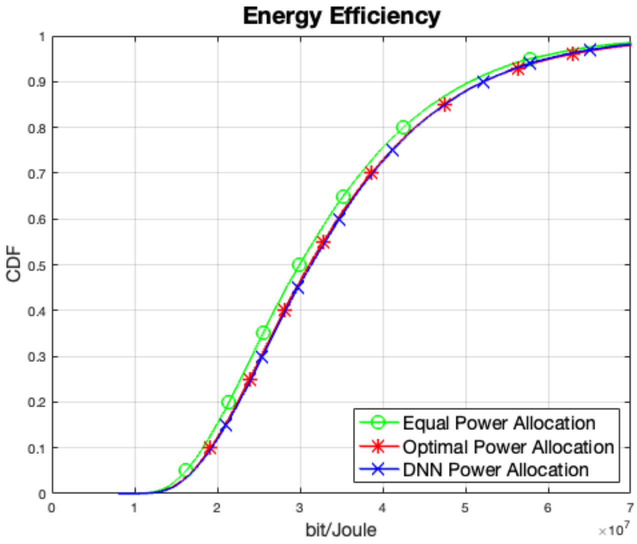


Figure 3.9. Energy efficiency of DNN for power allocation using Algorithm 2, for K=4, L=9, N=4.

3.7.3 DNN Results for Joint User-Centric Clustering and Power Allocation

The DNN performance of the joint user-centric clustering and PA is evaluated in terms of the energy efficiency. First, the user-centric clustering is performed using the trained DNN. Then, the output of the DNN is used as the input of the power allocation DNN along with the channel gain information. The resulting energy efficiency for Algorithm 1 and the proposed user-centric clustering algorithms are given in Figure 3.10.

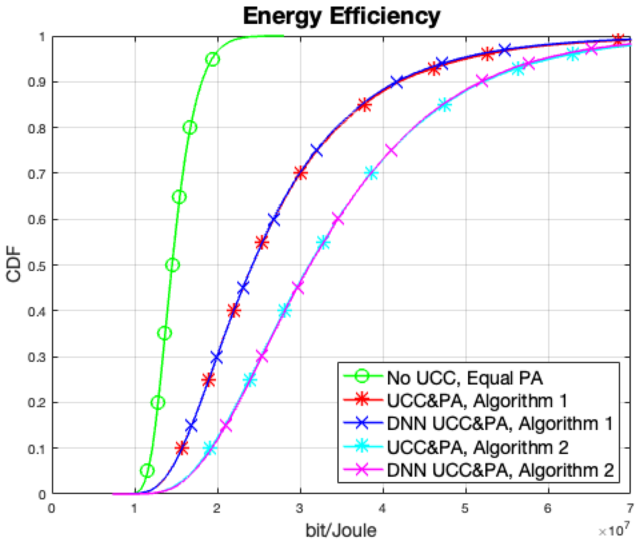


Figure 3.10. Energy efficiency of DNN for joint power allocation and user-centric clustering, for K=4, L=9, N=4.

The simulation results indicates that the proposed DNN approach for joint user-centric clustering and power allocation achieves the same performance as the Algorithm 1 and Algorithm 2. The energy efficiency performance of the proposed DNN trained by Algorithm 1 is 0.597% lower than the Algorithm 1 and DNN trained by Algorithm 2 0.212% is lower than the Algorithm 2. Even though the DNN approach achieves better user-centric clustering performance with Algorithm 1, the proposed user-centric clustering algorithm has better overall energy efficiency performance.

In this chapter, we proposed a DNN approach for user-centric clustering and power allocation in cell-free massive MIMO systems. The proposed DNN approach consists of two parts as user-centric clustering part and power allocation part to achieve modular and controllable network. The individual performance of the user-centric clustering part is evaluated in terms of the decision accuracy and the energy efficiency. The results indicates that the proposed DNN approach decides the binary state of the AP-UE pairs with 1.58% error and achieves 0.326% lower energy efficiency when trained by Algorithm 1 while the proposed user-centric clustering algorithm achieves 1.46% decision error and 0.523%

lower energy efficiency. The reason can be explained as low AP-UE pairing errors has smaller probabilities to activate a disabled APs and therefore, does not affect the total power consumption significantly. Also, the precoding scheme ZF has the ability of eliminating the inter-user interference which results that the AP-UE pairing errors does not cause inter-user interference. The proposed user-centric clustering algorithm results less active AP-UE pairs and therefore, each misclassified AP-UE pair has more impact on the energy efficiency performance than the Algorithm 1.

Then, the individual performance of the power allocation is evaluated in terms of the energy efficiency and the computational time required to find power allocation coefficients. The time measurements indicates that the proposed DNN approach requires 99.9% less computational time on average for both Algorithm 1 and Algorithm 2. The energy efficiency results indicates that the average energy efficiency for the proposed DNN approach is 0.28% lower for Algorithm 1 and 0.01% lower for the proposed user-centric clustering algorithm.

Finally, the performance of the joint user-centric clustering and power allocation using DNN approach is evaluated in terms of the energy efficiency. The results indicates that the proposed joint DNN approach achieves almost the same performance as the for Algorithm 1 and Algorithm 2. The energy efficiency performance of the proposed DNN approach is 0.597% lower for Algorithm 1 and 0.212% lower for the proposed user-centric clustering algorithm. Even though the DNN approach achieves better user-centric clustering performance using Algorithm 1, the proposed algorithm has better overall energy efficiency performance.

The results of the joint resource allocation for cell-free massive MIMO concludes that the proposed DNN approach achieves the accuracy and time requirements by providing almost the same performances.

CHAPTER 4

CONCLUSION

In this thesis, the resource allocation algorithms for cell-free massive MIMO communication system has been examined for maximizing the energy efficiency and reducing the computation time.

The novel user-centric clustering algorithm has been proposed that allows disabling the non-master APs that are serving few amount of users. This additional feature aims to reduce total power consumption of the system without sacrificing the advantages of the cell-free massive MIMO communication systems. The performance of user-centric clustering has been examined for different AP-UE pairing thresholds under equal power allocation by performing user-centric clustering instead of serving all UEs by all APs. The proposed user-centric clustering algorithm provided higher energy efficiencies than the Algorithm 1. This concludes that disabling the APs with non-significant contributions to the system performance is a promising approach to improve the energy efficiency of cell-free massive MIMO systems.

Then, the power allocation optimization problem has been formulated to maximize the energy efficiency in the cell-free massive MIMO systems and solved by using interior-point algorithm. The performance of the power allocation has been evaluated in terms of the achieved energy efficiency improvement compared to the equal power allocation.

The machine learning based power allocation approach has been proposed to reduce the computation time of user-centric clustering and power allocation. In this approach, the trained DNN model approximates the optimal solution with sufficient accuracy and significantly less computation time compared to the interior-point algorithm. The proposed DNN architecture has two parts as user centric clustering and power allocation. These parts are trained separately and then cascade-connected to each other. The performance results indicate that the DNN approach for joint user-centric clustering and power allocation achieves almost the same performance as the analytical approach. The achieved energy efficiency performance of the DNN approach is less than 1% lower compared to the interior-point algorithm while the required computational time is reduced by more than $\%99.9$. This concludes that the proposed DNN approach is suitable for cell-free massive MIMO systems having significantly less computation time.

As the future works, the proposed user-centric clustering algorithm and DNN architecture can be extended to the distributed cell-free massive MIMO systems.

BIBLIOGRAPHY

3GPP (2017). Further advancements for e-utra physical layer aspects. Technical Specification (TS) 36.814, 3rd Generation Partnership Project (3GPP).

Andrae, A. S. G. and T. Edler (2015). On global electricity usage of communication technology: Trends to 2030. *Challenges 6*(1), 117–157.

Biswas, S. and P. Vijayakumar (2021). Ap selection in cell-free massive mimo system using machine learning algorithm. In *2021 Sixth International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pp. 158–161.

Björnson, E., J. Hoydis, and L. Sanguinetti (2017). Massive mimo networks: Spectral, energy, and hardware efficiency. *Foundations and Trends® in Signal Processing 11*(3-4), 154–655.

Björnson, E. and L. Sanguinetti (2020). Scalable cell-free massive mimo systems. *IEEE Transactions on Communications 68*(7), 4247–4261.

Björnson, E., L. Sanguinetti, J. Hoydis, and M. Debbah (2015). Optimal design of energy-efficient multi-user mimo systems: Is massive mimo the answer? *IEEE Transactions on Wireless Communications 14*(6), 3059–3075.

Bobrov, E., B. Chinyaev, V. Kuznetsov, H. Lu, D. Minenkov, S. Troshin, D. Yudakov, and D. Zaev (2022). Adaptive regularized zero-forcing beamforming in massive mimo with multi-antenna users.

Chakraborty, S., E. Björnson, and L. Sanguinetti (2019). Centralized and distributed power allocation for max-min fairness in cell-free massive mimo. In *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pp. 576–580.

Chakraborty, S., O. T. Demir, E. Björnson, and P. Giselsson (2021). Efficient downlink power allocation algorithms for cell-free massive mimo systems. *IEEE Open Journal of the Communications Society 2*, 168–186.

Chen, X., T. Zhao, Q. Sun, Q. Hu, and M. Xu (2022). Cell-free massive mimo with energy-efficient downlink operation in industrial iot. *Mathematics 10*(10).

Demir, O. T., E. Björnson, and L. Sanguinetti (2021). Foundations of user-centric cell-free massive mimo.

Hornik, K., M. Stinchcombe, and H. White (1989). Multilayer feedforward networks are universal approximators. *Neural Networks 2*(5), 359–366.

Kingma, D. and J. Ba (2014, 12). Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Krishnamoorthy, A. and R. Schober (2023). Downlink massive mu-mimo with successively-regularized zero forcing precoding. Volume 12, pp. 114–118.

Mai, T. C., H. Q. Ngo, and L.-N. Tran (2022). Energy efficiency maximization in large-scale cell-free massive mimo: A projected gradient approach. *IEEE Transactions on Wireless Communications 21*(8), 6357–6371.

Ngo, H. Q., L.-N. Tran, T. Q. Duong, and M. Matthaiou (2017). Energy efficiency optimization for cell-free massive mimo. *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 1–5.

Ngo, H. Q., L.-N. Tran, T. Q. Duong, M. Matthaiou, and E. G. Larsson (2018). On the total energy efficiency of cell-free massive mimo. *IEEE Transactions on Green Communications and Networking 2*(1), 25–39.

Papazafeiropoulos, A. K., E. Björnson, P. Kourtessis, S. Chatzinotas, and J. M. Senior (2020). Scalable cell-free massive mimo systems with hardware impairments. pp. 1–7.

Ramachandran, P., B. Zoph, and Q. V. Le (2017). Searching for activation functions.

Zaher, M., O. Tuğfe Demir, E. Björnson, and M. Petrova (2021). Distributed dnn power allocation in cell-free massive mimo. In *2021 55th Asilomar Conference on Signals, Systems, and Computers*, pp. 722–726.

Zhang, J., S. Chen, Y. Lin, J. Zheng, B. Ai, and L. Hanzo (2019). Cell-free massive mimo: A new next-generation paradigm. *IEEE Access 7*, 99878–99888.

Zhao, Y., I. G. Niemegeers, and S. H. De Groot (2020). Power allocation in cell-free massive mimo: A deep learning method. *IEEE Access 8*, 87185–87200.

Zuo, J., J. Zhang, C. Yuen, W. Jiang, and W. Luo (2017). Energy-efficient downlink transmission for multicell massive das with pilot contamination. *IEEE Transactions on Vehicular Technology 66*(2), 1209–1221.