

Received April 17, 2020, accepted May 2, 2020, date of publication May 6, 2020, date of current version May 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2992629

Power Allocation in Cell-Free Massive MIMO: A Deep Learning Method

YU ZHAO^{ID}, IGNAS G. NIEMEGERERS^{ID}, AND SONIA HEEMSTRA DE GROOT^{ID}

Department of Electrical Engineering, Eindhoven University of Technology, 5600MB Eindhoven, The Netherlands

Corresponding author: Yu Zhao (y.zhao3@tue.nl)

ABSTRACT Massive multiple-input multiple-output (MIMO) is a key technology in 5G. It enables multiple users to be served in the same time-frequency block through precoding or beamforming techniques, thus increasing capacity, reliability and energy efficiency. A key issue in massive MIMO is the allocation of power to the individual antennas, in order to achieve a specific objective, e.g., the maximization of the minimum capacity guaranteed to each user. This is a nondeterministic polynomial (NP)-hard problem that needs to be solved in a timely manner since the state of the channels evolves in time and the power allocation should stay in tune with this state. Although several heuristics have been proposed to solve this problem, these entail a considerable time-complexity. As a result, with the present methods, it cannot be guaranteed that power allocation happens in time. To solve this problem, we propose a deep neural network (DNN). A DNN has a low time complexity, but requires an extensive, offline, training process before it becomes operational. The DNN we propose is the combination of two convolutional layers and four fully connected layers. It takes as input the long-term fading information and it outputs the power for each antenna element to each user. We limit ourselves to the case of time-division duplex (TDD) based sub-6GHz networks. Numerical results show that, our DNN-based method approximates very closely the results of a commonly used heuristic based on the bisection algorithm.

INDEX TERMS Cell-free massive MIMO, deep learning, power allocation.

I. INTRODUCTION

Multiple-input multiple-output (MIMO) systems, where several antennas are deployed in both transmitter and receiver, have been studied during the last two decades and applied to many wireless standards [1]. While the initial work was focused on point-to-point links, multi-user MIMO (MU-MIMO) attracted more attention in recent years and is a technique used in 5G. In MU-MIMO a base station (BS) simultaneously serves several users in the same time-frequency resource block. Benefiting from the spatial diversity and multiplexing, MU-MIMO increases capacity, enhances reliability, improves energy efficiency and reduces interference [2]. In order to achieve further capacity gains, massive MIMO has been introduced, where the number of antennas in a BS has been increased to hundreds, being at least one order of magnitude larger than the number of user equipment (UEs) served simultaneously. As the number of antennas in an antenna array grows, the channels become asymptotically deterministic; this is known as channel hardening. As a result, the capacity only depends on the long-term

fading coefficients, which are stable over a time period of typically several seconds [3].

Recently, cell-free (CF) massive MIMO has been proposed [4], where several remote antenna arrays (RAA) are spread over a coverage area and are connected to a central controller (CC) via a fronthaul. The CC function, which in 5G is part of the BS (gNB), determines the power to be allocated for each UE at each individual antenna.

In a massive MIMO system, a UE in the coverage area can, in principle, be served by each and every antenna. In CF massive-MIMO, this is still true, but the antennas are now geographically distributed over the coverage area in clusters (sub-arrays). The difference with a cellular or small cell system, is that in those, a UE can only be served by antennas belonging to a particular cluster, i.e., a particular BS. Hence, in CF massive-MIMO, the concept of cell has disappeared [5] and all antennas coherently serve all UEs [6].

Compared to centralized massive MIMO (Fig.1), where all antennas are co-located in one BS, the merits of CF massive MIMO are:

(1) Much smaller physical size of the RAAs compared to a centralized antenna array, which facilitates deployment, in particular in indoor scenarios.

The associate editor coordinating the review of this manuscript and approving it for publication was Xingwang Li^{ID}.

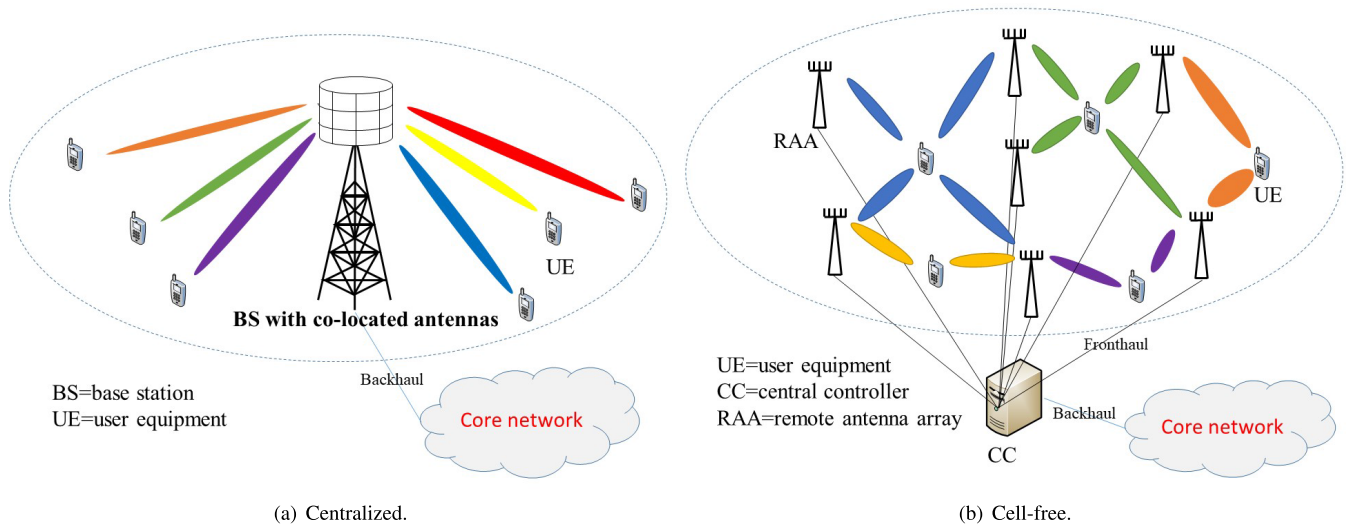


FIGURE 1. Centralized massive MIMO vs CF massive MIMO.

(2) Better performance for the same total number of antennas; e.g., in [7] it is shown that CF massive MIMO can have a significantly higher sum-rate than a centralized system, for the same precoding and transmission power.

(3) Higher robustness against shadow fading due to the spatial diversity of the RAA arrays; this is particularly important for mmWave.

There are also some drawbacks of CF massive MIMO with respect to centralized massive MIMO:

(1) Higher physical deployment costs due to the geographical spread of the RAAs and the need for a fronthaul.

(2) Non-negligible propagation delay in the fronthaul, because the RAAs are spread over a potentially large area.

Since in massive MIMO, and in particular CF massive MIMO, different UEs are served simultaneously in the same time-frequency domain, controlling the inter-user interference is important. The power allocation plays a crucial role in controlling this interference and in optimizing the performance [8], [9]. In this paper we address this problem for CF massive MIMO. To ensure quality of service fairness for the UEs, we intend to do this by max-min power control optimization. We assume that the UEs have a single active antenna.

A key benefit of massive MIMO is that the short-term fading can be neglected due to channel hardening [10]. The time scale of the long-term fading determines the power controlling time, which puts a constraint on the validity of the power allocation decisions. Hence, it limits the time budget available in the CC for the collection of the channel state information (CSI), the computation of the transmission power for each antenna to UE, and, the sending of the power settings to each RAA via the fronthaul.

The problem we are facing is that the time complexity of an exact solution of the max-min power optimization is excessive for the given time budget. Even the heuristics that

are proposed in the literature, e.g., the bisection algorithm in [4], are too complex to meet the time constraints.

This led us to propose a deep neural network (DNN) to perform the task. The time-complexity of a DNN is low. The price one pays is the off-line training that needs to be performed. This can be lengthy and requires the generation of a large training data set. However, this is not a real problem since it is an activity which is performed off-line, before the DNN becomes operational.

Deep learning has become a popular method to solve complex problems [11]. Universal approximation theory [12], proves that a DNN can approximate any function that has continuous values. It has shown competitive performance, compared to non-machine learning solutions, in communication networks such as multiple access scheme [13], routing optimization and congestion reduction [14].

A. RELATED WORK

Many papers have studied massive MIMO, e.g., [15], [16] and [17], however, most of these consider centralized massive MIMO. A few papers, in particular [7], [18] and [19], have addressed distributed massive MIMO. They found that a distributed system offers higher sum-rates than centralized systems, assuming perfect CSI and no interference. A realistic analysis however must account for imperfect CSI as well as interference between UEs. In [4] the max-min power allocation, based on imperfect CSI, is studied for the extreme case where each RAA has only one antenna. In [20] the same problem is studied for different RAA sizes, assuming fixed UE positions. However, none of these papers ([4] and [20]) consider the real-time constraints of the power-allocation, which is our main concern.

Recently the power of applying deep learning methods in the control of wireless communication systems has been shown in [21], [22], and, in particular for massive MIMO,

in [23], [24], and [25]. In [21] a fully connected neural network mimics the processing of the weighted minimum mean square error (WMMSE) algorithm to manage interference in multi-cell networks. The numerical results show that with a neural network one can closely approximate the performance of WMMSE but, in much less time. In [23] and [24] it is shown that a DNN achieves near-optimal accuracy of signal detection with much less computations compared to the classical method. DNN-based channel estimations were proposed in [25], [26] and [27], outperforming traditional estimators in terms of computational cost. Specifically, a fast and flexible denoising convolutional neural network (FFDNet) is proposed to estimate the channel for cell-free mmWave massive MIMO in [27], the authors claim that the proposed FFDNet achieves faster training speed than the existing DNN methods without sacrificing normalized mean square error performance.

Similarly, in [28] and [29] deep learning was used to solve power allocation in massive MIMO. A fully connected neural network and a recurrent neural network were proposed to maximize the spectrum efficiency (SE) and implement the max-min power policy respectively in [28]. In [29] a two-layer DNN was used for power allocation to combat inter-cell interference. The state-of-the-art residual dense block (ResDense) method was applied in [30] to allocate power in multi-cell massive MIMO. Numerical results show encouraging performance of DNN-based power allocation; hence, it is expected that DNNs will replace traditional heuristics, because of their lower time complexity.

B. CONTRIBUTIONS

In this work we consider the real-time power allocation problem in CF massive MIMO using time-division duplex operation (TDD). Our contributions are the following:

(1) We formulate the general max-min power allocation problem, then we propose a heuristic, consisting of a non-convex iteration algorithm combined with the bisection method, to solve it.

(2) We construct a DNN to approximate our proposed heuristic.

(3) We analyze the performance of the DNN in terms of the accuracy with which it approximates the heuristic and its time complexity. Our results show that the DNN provides a very good approximation, while requiring significantly less computation time.

The paper is organized as follows: in Section II, we formalize the CF massive MIMO system and the max-min optimization problem. In Section III we propose a heuristic to optimize the max-min power allocation. Its purpose is to generate offline a data set to be used for training the DNN and to assess the results of our DNN approach. The proposed DNN is discussed in Section IV. Section V shows numerical results, which provide evidence that our DNN solution approximates the heuristic quite closely. Section VI summarizes our conclusions and indicates further research that is needed.

Notation: Boldface characters denote a matrix or a vector. $()^*$ and $()^H$ stand for conjugate and conjugate-transpose, respectively. $\|\cdot\|$ represents the Euclidean norm, and $\mathbb{E}\{\cdot\}$ is the expectation operator. We use UE and user interchangeably.

II. CF MASSIVE MIMO SYSTEM

Consider a CF massive MIMO system comprising N RAAs, each equipped with M antennas. K single-antenna UEs are served by these RAAs in a given coverage area. Furthermore, all RAAs are connected to a CC, where the power allocation is performed. We base our system model on the following common assumptions:

- (1) The system operates in the sub-6GHz frequency bands.
- (2) The standard block-fading model ([31], [32]) is used.
- (3) The time-frequency resources are divided into coherence intervals, during which the channel can be regarded as constant [33].
- (4) The channels between antenna m in RAA n and user k , for all m, n , and k , are i.i.d. with a Rayleigh fading distribution [9]:

$$g_{n,m}^k \sim \mathcal{CN}(0, \beta_{n,m}^k) \quad (1)$$

where $\beta_{n,m}^k$ is the long-term fading coefficient. [34] proposed that the channel model should be the combination of Rayleigh and Rician fading, because the LOS probabilities depend on the distance between the transmitters and receivers. In this work, we assume Rayleigh fading to simplify the model.

- (5) The propagation in the fronthaul is negligible. However, as argued in [4], the effects of the fronthaul should be quantified in future work, as is done in [35] and [36].

A. UPLINK CHANNEL ESTIMATION

Let τ_c be the coherence time expressed in number of modulation symbols. We assume pilot sequences of length K where $K < \tau_c$. UE k is preassigned the pilot $\boldsymbol{\psi}_k$ with $\mathbb{E}\{|\boldsymbol{\psi}_k|^2\} = 1$ and $\boldsymbol{\psi}_k \boldsymbol{\psi}_{k'}^H = 0$ for $k \neq k'$. This guarantees that there is no intra-cell pilot contamination [37]. In general, pilot contamination is a very important issue in MIMO systems. It degrades the quality of the CSI, which in turn reduces the system capacity. References [38] and [39] examine how intra-cell pilot contamination can be decreased for multi-cell massive MIMO, [40] checks how to reduce the pilot contamination for cell-free massive MIMO.

During the channel estimation phase, all users send their pilots with full transmission power. The received signal at antenna m in RAA n is the superposition of these pilots:

$$\mathbf{y}_{n,m} = \sum_{k=1}^K p_p^{1/2} g_{n,m}^k \boldsymbol{\psi}_k + \mathbf{w} + \boldsymbol{\iota} \quad (2)$$

where p_p is the pilot power, $\mathbf{w} \in \mathbb{C}^K$ is an additive Gaussian white noise vector with element power σ^2 and $\boldsymbol{\iota} \in \mathbb{C}^K$ is the pilot contamination vector from other cells with element power η^2 . Based on the received signal, the RAA performs conjugate operations to decode the desired signal from

user k :

$$y_{n,m}^k = \mathbf{y}_{n,m} \boldsymbol{\psi}_k^H = \sum_{k'=1}^K p_p^{1/2} g_{n,m}^k \boldsymbol{\psi}_{k'} \boldsymbol{\psi}_k^H + \boldsymbol{\psi}_k^H \mathbf{w} + \boldsymbol{\psi}_k^H \boldsymbol{\iota} \quad (3)$$

The Bayes estimator [41] will produce the estimated channel by prior probability $g_{n,m}^k \sim \mathcal{CN}(0, \beta_{n,m}^k)$:

$$\hat{g}_{n,m}^k = \mathbb{E}\{g_{n,m}^k | y_{n,m}^k\} = \frac{p_p^{1/2} \beta_{n,m}^k}{p_p \beta_{n,m}^k + \sigma^2 + \eta^2} y_{n,m}^k \quad (4)$$

See the details in *Appendix A*.

B. DOWNLINK TRANSMISSION

Based on the estimated channels, i.e., the CSI values, the CC determines the power for each antenna in each RAA to transmit data to UEs. Let q_k with $\mathbb{E}\{|q_k|^2\} = 1$ be the intended signal for user k , then the transmitted signal of antenna m in RAA n is :

$$x_{n,m} = \sum_{k=1}^K (p_{n,m}^k)^{1/2} f_{n,m}^k q_k \quad (5)$$

where $p_{n,m}^k$ is the downlink transmission power from antenna m in RAA n to user k and $f_{n,m}^k$ is an element of the precoding vector. Since the max-ratio precoding can be performed locally at each RAA, and hence is very suitable for a CF massive MIMO architecture [42], we use max-ratio precoding as was also done in [9]:

$$f_{n,m}^k = \frac{\hat{g}_{n,m}^k}{\sqrt{\mathbb{E}\{|\hat{\mathbf{g}}_{n,m}|^2\}}} \quad (6)$$

The max-ratio precoding has lower complexity than zero-forcing and approximates the performance of the optimal dirty-paper precoding [43] when increasing the number of antennas [44]. To find a low-complexity precoding algorithm with good performance is still a challenge. In [45] a promising deep-learning based algorithm is proposed, which, however, requires further research and analysis.

UE k will receive the superposition of the signals of all RAAs in the whole system:

$$y_k = \sum_{n=1}^N \sum_{m=1}^M (g_{n,m}^k)^* \sum_{k'=1}^K (p_{n,m}^{k'})^{1/2} f_{n,m}^{k'} q_{k'} + w_k \quad (7)$$

where w_k is the thermal noise with element power σ^2 at UE k .

C. ACHIEVABLE DOWNLINK DATA CAPACITY

We can decompose the right side of equation (7) into a sum of four terms: the desired signal, the fluctuation caused by the uncertain channel gains, the interference from other UEs, and the noise, as shown in (8). According to [9], the achievable downlink capacity for user k , C_k is given by the use-and-then-forget (UatF) bound:

$$C_k = (1 - \frac{K}{\tau_c}) \log_2(1 + SINR_k) \quad (9)$$

with the achievable signal-to-interference-plus-noise ratio (SINR) given by (10). Details are shown in *Appendix B*.

III. MAX-MIN POWER ALLOCATION

As we see in (10), the achievable SINR of user k is a function of the following variables: long-term fading, pilot power, white noise and downlink transmission power. In this section, we use a heuristic consisting of a combined bisection-plus-iteration algorithm for max-min power allocation. The max-min policy ensures quality of service fairness among all UEs. Other power allocation policies such as new fairness power allocation [46], max product SINR [30], given target SINR [31] could also be considered, which would lead to a different optimization objective function.

The max-min power allocation policy can be formulated as:

$$\begin{aligned} & \max_{p_{n,m}^k} \min_k SINR_k \\ & s.t. \sum_{k=1}^K p_{n,m}^k \leq p_l \end{aligned} \quad (11)$$

where p_l is the downlink transmission power limitation of each antenna, $n = 1, 2, \dots, N$; $m = 1, 2, \dots, M$ and $k = 1, 2, \dots, K$. An exact solution for this optimization problem is not feasible, since the time complexity increases exponentially as M, N and K increase linearly, i.e., it is nondeterministic polynomial (NP)-hard. Therefore, one has to resort to a heuristic to solve it. A widely adopted heuristic (see, e.g., [4]) is the bisection algorithm [47], which divides the problem into two sub-problems, namely, the candidate value problem and the feasibility problem. In each loop of the bisection algorithm, a candidate value is chosen to determine the constraints of the subsequent feasibility problem. The challenge of this method is that we need to solve a non-convex feasibility problem in each loop, which is a nonlinear inequalities problem with $K + NM$ constraints and NMK variables. There is no analytical solution, therefore we propose a numerical solution given by *Algorithm 1*. This, of course, will give us sub-optimal solutions.

The key idea of *Algorithm 1* is that, after an initial power allocation, the user with the highest capacity will always give its allocated power to the user with the lowest capacity, provided that its capacity does not go below the candidate capacity that corresponds to $SINR_{candidate}$, which is $SINR_{candidate} = (SINR_{max} + SINR_{min})/2$ in our bisection algorithm. At the stopping point of the algorithm, the difference in SINR between the user with the highest and the one with the lowest capacity, will be no more than ϵ_1 . If we set ϵ_1 small enough, the capacity of all users will be regarded as sufficiently equal, achieving fairness. The tolerances ϵ_2 and ϵ_3 , here, are used to avoid endless loops.

The time complexity to solve (11) by our heuristic is $O(\log_2(MN)M^3N^3K^2)$ which is shown in *Appendix C*. Even such a polynomial complexity can be too high when the

Algorithm 1

Input: $SINR_{candidate}$
Output: feasibility, $p_{n,m}^k$

- 1: Set initial power allocation and tolerance $\epsilon_1, \epsilon_2, \epsilon_3 > 0$
- 2: while $SINR_{max} - SINR_{min} > \epsilon_1$
- 3: if $SINR_{max} - SINR_{candidate} < \epsilon_2$
- 4: break while
- 5: end if
- 6: $i = \arg \max_k SINR_k$
- 7: $j = \arg \min_k SINR_k$
- 8: for $t = 1 : MN$
- 9: set: $sup = 1$
- 10: while $p_t^i / 2^{sup} > \epsilon_3$
- 11: $(p_t^i)' = p_t^i - p_t^i / 2^{sup}$
- 12: $(p_t^j)' = p_t^j + p_t^j / 2^{sup}$
- 13: if $SINR_i' \geq SINR_{candidate}$
- 14: $p_t^i = (p_t^i)', p_t^j = (p_t^j)'$
- 15: if $SINR_j' \geq SINR_{candidate}$
- 16: break for
- 17: end if
- 18: else
- 19: $sup = sup + 1$
- 20: end if
- 21: end while
- 22: end for
- 23: if $SINR_{min} \geq SINR_{candidate}$
- 24: break while
- 25: end if
- 26: end while
- 27: if $SINR_{min} < SINR_{candidate}$
- 28: unfeasible
- 29: else
- 30: feasible
- 31: end if

solution has to be obtained within the time constraints. This was the motivation to try DNNs to solve the problem.

IV. DEEP LEARNING BASED POWER ALLOCATION

In this section we propose a DNN to perform the max-min power allocation. This method has low complexity: it only requires a number of layers of simple operations such as matrix or vector multiplications [21]. In addition, a DNN is expected to run on neural processing units (NPU) [48], specifically designed to support machine learning, making it possible to parallelize parts of the computation. Hence it should be easy to meet the time constraints. Unlike traditional iterative methods of solving (11), the DNN-based power allocation operates as a nonlinear regression: given the input information (the long-term fading coefficients $\beta_{n,m}^k$), the DNN will output the transmission power for each antenna. Since we use a supervised-learning method, a training data set is required. For a given CF massive MIMO network, the training should be based on the particular network configuration and the usage scenarios to be expected.

Let there be N RAAs in the coverage area. For each realization, we assume K UEs are uniformly and randomly distributed in the coverage area. The max-min power allocation is computed by the heuristic described in section III, to generate a sufficiently large number of data points as training samples for the DNN. We record three elements of each realization to compose a training data point: the long-term fading coefficients $K \times MN$ matrix β , the achievable signal to noise ratio S , i.e., the final $SINR_{candidate}$, and the corresponding computed power allocation vector p . Multiple realizations are produced to generate the training dataset.

$$\begin{aligned}
 y_k = & \underbrace{\mathbb{E}\left\{\sum_{n=1}^N \sum_{m=1}^M (g_{n,m}^k)^* (p_{n,m}^k)^{1/2} f_{n,m}^k q_k\right\}}_{\text{Desired}} + \underbrace{\sum_{n=1}^N \sum_{m=1}^M (g_{n,m}^k)^* (p_{n,m}^k)^{1/2} f_{n,m}^k q_k - \mathbb{E}\left\{\sum_{n=1}^N \sum_{m=1}^M (g_{n,m}^k)^* (p_{n,m}^k)^{1/2} f_{n,m}^k q_k\right\}}_{\text{Fluctuation}} \\
 & + \underbrace{\sum_{n=1}^N \sum_{m=1}^M (g_{n,m}^k)^* \sum_{k' \neq k} (p_{n,m}^{k'})^{1/2} f_{n,m}^{k'} q_{k'}}_{\text{Interference}} + \underbrace{w_k}_{\text{Noise}}
 \end{aligned} \tag{8}$$

$$SINR_k = \frac{\left| \sum_{n=1}^N \sum_{m=1}^M (p_{n,m}^k)^{1/2} \frac{1}{\sqrt{\sum_{i=1}^K \frac{(p_p^{1/2} \beta_{n,m}^i)^2}{p_p \beta_{n,m}^{k'} + \sigma^2 + \eta^2}}} \frac{p_p (\beta_{n,m}^k)^2}{p_p \beta_{n,m}^{k'} + \sigma^2 + \eta^2} \right|^2}{\sum_{k'=1}^K \sum_{n=1}^N \sum_{m=1}^M p_{n,m}^{k'} \frac{1}{\sum_{i=1}^K \frac{(p_p^{1/2} \beta_{n,m}^i)^2}{p_p \beta_{n,m}^{k'} + \sigma^2 + \eta^2}}} \frac{p_p (\beta_{n,m}^{k'})^2}{p_p \beta_{n,m}^{k'} + \sigma^2 + \eta^2} \beta_{n,m}^{k'} + \sigma^2 + \eta^2} \tag{10}$$

$$MSE = \frac{1}{i} \sum_{i=1}^I \left(\underbrace{\frac{1}{NMK} \sum_{n=1}^N \sum_{m=1}^M \sum_{k=1}^K \left(\frac{(p_{n,m}^k)^{pre,i} - (p_{n,m}^k)^{target,i}}{p_l} \right)^2}_{\text{average power coefficient error}} + \underbrace{\left(\frac{(S)^{pre,i} - (S)^{target,i}}{(S)^{target,i}} \right)^2}_{\text{regression rate error}} \right) \tag{12}$$

A. DESIGN OF THE DNN

Several DNNs have been proposed in the literature to solve the power allocation problem. In [21], a fully connected DNN was proposed to control interference. Its purpose was to approximate the WMMSE algorithm. Although it can be proved that an arbitrary accuracy can be achieved by several layers, overfitting, i.e., good performance for training data while bad performance for testing data, is inherently a shortcoming of a fully connected network. In [30] the residual dense block (ResDense), which consists of several convolutional layers is used for power allocation in massive MIMO. Good performance was achieved by this powerful DNN; however, the objective function is different from ours: the sum spectral efficiency is maximized. The solution for this maximization problem is based on the greedy strategy, i.e., a user will always be served by its closest BS. This makes the problem easier to solve as the solution is confined to a small feasible region. Moreover, since [30] studied multi-cell MIMO, based on a centralized MIMO setup, the system model is entirely different from ours. Since neither of these DNNs was suitable for solving our problem, we needed a different design.

As mentioned in Section III our proposed algorithm for power allocation involves two parts: the candidate value problem and the feasibility problem. We note that the candidate value $SINR_{candidate}$ is determined by a feasible solution, while the power allocation is based on $SINR_{candidate}$. This makes the problem more complicated since we need to solve two sub-optimization problems. If one is known or obtained easily, e.g., the final $SINR_{candidate}$ (S) is known, we can decrease the loop in Algorithm 1 or, if we can easily determine whether $SINR_{candidate}$ is feasible, in each loop of Algorithm 1, a simple bisection computation will get the final $SINR_{candidate}$ by an iteration involving multiple steps, thereby simplifying the optimization significantly.

Based on the above analysis, and according to the universal approximation theory [12], we expect to get S by using a DNN with several layers. We use a convolutional neural network for the regression of S , because compared with the fully connected neural network, it can significantly decrease the number of parameters. After we get S , we need an iterative algorithm to solve the feasibility problem, i.e., the power allocation. Referring to [21], we can use several fully connected layers to approximate this iterative algorithm. So, we design the structure of our DNN as consisting of two stages: regression processing and allocation processing.

While we propose a specific DNN structure, it is worth to point out that finding the best DNN structure and the values of the hyper-parameters can also be seen as optimization problems in their own right, requiring further research. Referring to the literature ([21], [49]), we tried several structures and hyper-parameters, including fully connected networks (from one layer to six layers) and traditional convolutional neural networks (two convolutional layers with a number of fully connected layers varying from one to four), to choose the

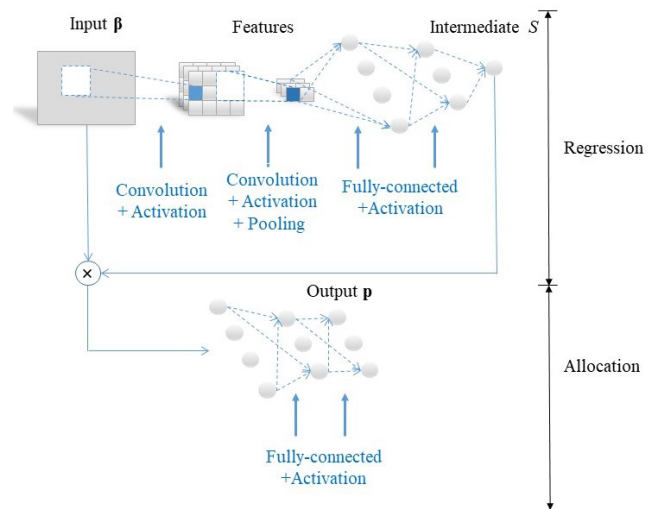


FIGURE 2. Structure of the DNN.

best configuration, i.e., the one that gives us the lowest mean square error (MSE) on the training dataset.

Let us now discuss in detail the two stages of the DNN. For an explanation of the general concepts used in convolutional neural networks, we refer to [11] and [49]. Rectified linear units (ReLu) are used as our activation functions in all layers.

Regression: The objective of the regression part is to get S , the achievable SINR, from the input, i.e., from the long-term fading coefficients matrix β . We use two convolutional layers and two fully connected layers for this process (see Fig.2). Specifically, for the first convolutional layer, we use $5 \times M \times Q$ filters with stride $[1, M]$ to operate on the input matrix. The result of this convolutional operation yields Q feature matrices with $K \times N$ elements by zero padding 2. Note that we do not use a pooling operation after feature extraction in this layer. In the second layer, we use $5 \times 5 \times Q$ filters with stride $[1, 1]$ and zero padding 2 to guarantee the same number of inputs and outputs in this layer. Then a max-pooling operation is used to decrease the number of parameters. Here we use a 2×2 kernel size with stride $[2, 2]$. After that we adopt a two-layer fully connected network to get the output S . The numbers of neurons in these two fully connected layers are $\lceil K/2 \rceil \times \lceil N/2 \rceil \times Q$ and $\lceil K/2 \rceil \times \lceil N/2 \rceil \times \lceil Q/2 \rceil$, respectively. $\lceil \cdot \rceil$ represents the ceiling operation.

Allocation: When the intermediate S is obtained, the next step is to perform the process of power allocation. We derive from (10) that there is a multiplication operation of β and S to calculate the transmission power; so $S\beta$ is the input in this phase. Finally, two fully connected layers with $2KNM$ and KNM neurons are employed to describe the nonlinear relationship between the input $S\beta$ and the output p .

B. TRAINING OF THE DNN

The DNN is trained in two phases, i.e., the input-output pair (β, S) is used to adjust the filters in the first two fully connected layers (regression processing), while $(S\beta, p)$ is used

to train the following two fully connected layers (allocation processing). MSE is used as loss function, see (12). I is the number of training data, and $(p_{n,m}^k)^{pre,i}$, $(p_{n,m}^k)^{target,i}$, $(S)^{pre,i}$, $(S)^{target,i}$ represent the output of the power allocation by the DNN, the target output of the power allocation by the training data set, the intermediate value by the DNN, and the intermediate value by the training data set for sample i , respectively. The first term in (12) represents the average power coefficient error while the second one is the regression rate error of the intermediate S .

The DNN is trained using the adapted gradient descent method described in [50], which means that the learning rate is updated in each epoch: in initial epochs the learning rates are large to guarantee that the loss function will converge, while in the later epochs the learning rates are getting smaller to achieve sufficient searching. The training process contains updating of weights between neurons and updating of the bias in each layer by partial derivatives. More details of how the training process works are described in [11].

C. TIME COMPLEXITY OF THE DNN

The time complexity of the DNN solution mainly lies in the training phase, which is not a problem for the operational phase of the system, since it can be done *offline*. The complexity of the DNN power allocation operation lies in the online forward propagation part. Assuming a well-trained DNN, the time complexity is $O(K^2N^2M^2)$; see Appendix D for details. Compared to the complexity of the heuristic algorithm $O(\log_2(MN)M^3N^3K^2)$ in Section III, the DNN power allocation eliminates multiple iteration loops. The computation times of the two methods are compared in the next section.

V. COMPARISON OF DNN-BASED AND HEURISTIC-BASED POWER ALLOCATION

In this section, we show by simulations that the DNN-based power allocation can closely approximate the performance of the bisection heuristic with a much lower time complexity.

A. CONFIGURATION AND SCENARIO FOR SIMULATION

We consider a $200 \times 200 m^2$ square coverage area with a total of 100 antennas to serve $K = 5$ users. There are 9 RAAs ($N = 9$) placed in a regular grid as shown in Fig.3. Each RAA has 11 antennas ($M = 11$), except for the central one which has 12. This has no particular significance, except that the simulations were originally done for studying the effects of the degree of distribution of the antennas on the network capacity. Note that determining the optimal degree of distribution and geographical deployment of RAAs in CF massive MIMO is an open issue. Finding the optimal configuration, given a particular number of antennas requires further research. We have used a sample configuration of a CF massive MIMO system to demonstrate the potential of the DNN-based solution. However, for a new configuration (e.g., different deployment of RAAs), the DNN might need to be retrained.

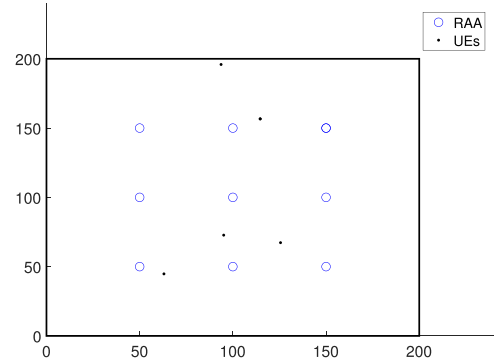


FIGURE 3. CF massive MIMO configuration used in simulations.

As in [4], the maximum power levels for each RAA antenna and each UE are 200 mW and 100 mW respectively. The carrier frequency is 1.9 GHz and the available bandwidth is 20 MHz. We set the height of the RAAs to 15 m and for the UEs to 1.65 m. We assume that the noise power is -94 dBm, and the standard deviation of shadow fading is 8 dB. The number of modulation samples in each coherence interval is assumed to be 200. The pilot contamination from other cells is -80 dBm.

The mini batch size, i.e., the number of samples to process before the parameters are updated in the DNN, is 500. The initial learning rate is 0.002 and the maximum number of iterations is 800. The initial weights and bias are Gaussian random variables that have an $\mathcal{N}(0, 0.01)$ distribution. The number of filters in each convolutional layer is 60. For the first two fully connected layers, we set the number of neurons to 900 and 450 respectively; while for the latter fully connected layers, we select 1000 and 500 respectively. The simulation parameters are listed in Table 1.

We assume that a correlated shadowing path loss model can be used, where the long-term fading coefficients $\beta_{n,m}^k$ are given by:

$$\beta_{n,m}^k = PL_{n,m}^k \times 10^{\frac{\sigma_{sh} z_{n,m}^k}{10}} \quad (13)$$

where $PL_{n,m}^k$ is the path loss in dB, the second factor represents the shadow fading with standard deviation σ_{sh} and $z_{n,m}^k$ is the shadow fading coefficient defined as in [4], [51]:

$$z_{n,m}^k = \sqrt{\kappa} a_{n,m} + \sqrt{1 - \kappa} b_k \quad (14)$$

where $a_{n,m} \sim \mathcal{N}(0, 1)$ and $b_k \sim \mathcal{N}(0, 1)$ are independent random variables, and κ , where $0 \leq \kappa \leq 1$, is a parameter. When $\kappa = 0$, the shadowing from a given user is the same to all RAAs, which means that the obstacle is near to the UE; while for $\kappa = 1$, the shadowing from a given RAA is the same to all users, means that the obstacle is near to the RAA. In our simulation, we set $\kappa = 0.5$ and adopt the covariance functions of $a_{n,m}$ and b_k in [51]. The three-slope path loss model [52] is formulated as in (15). $d_{n,m}^k$ is the distance between user k and antenna m of RAA n . L is defined in [53] as (16), where f is the carrier frequency, h_{RAA} is the height of the RAAs and h_u is the UE antenna height.

TABLE 1. Parameters.

Parameters for MIMO		Parameters for DNN	
Parameter	value	Parameter	value
Carrier frequency	1.9GHz	Batch size	500
Bandwidth	20MHz	Initial learning rate	0.002
Noise/contamination power	-94dBm/-80dBm	Maximum number of iterations	800
Height of RAAs	15m	Number of filters	60
Height of UEs	1.65m	Neurons in fully connected layers	[900,450;1000,500]
Power of pilots, downlink transmission	100mW		
Standard deviation of shadowing fading	8dB		

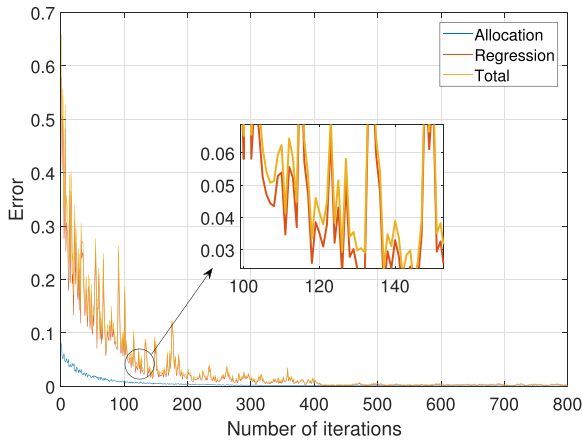


FIGURE 4. Power coefficient error, regression rate error and loss function as a function of the number of iterations.

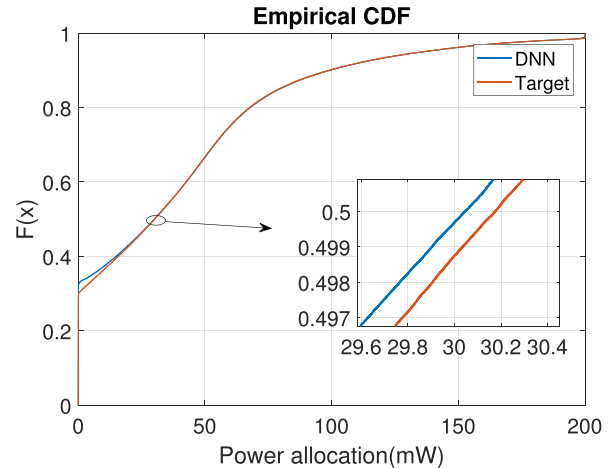


FIGURE 5. CDF of power allocation with $N = 9, M = 11/12$, fixed UE scenario.

B. PERFORMANCE OF DNN BASED POWER ALLOCATION

We evaluate the DNN-based power allocation in two scenarios, namely fixed-position UEs and moving UEs.

We should first train the DNN before testing and using it. Therefore, we generated a dataset, containing 432000 data points. The DNN is trained using 430000 data points. Fig.4 shows how the power coefficient error, the regression rate error and the loss function, evolve with the number of iterations.

Two hyper-parameters influence the loss function. The first one is the learning rate, i.e., the evolution rate of the DNN. A higher rate results in a faster convergence but runs the risk of ending up in a local optimum, which means insufficient searching has been done. A low rate, on the other hand, may lead to a loss function that does not converge. The second influential hyper-parameter is the number of filters (Q) in each convolutional layer. More filters improve the prediction accuracy, i.e., the loss becomes smaller, however, it also implies that the training time becomes longer.

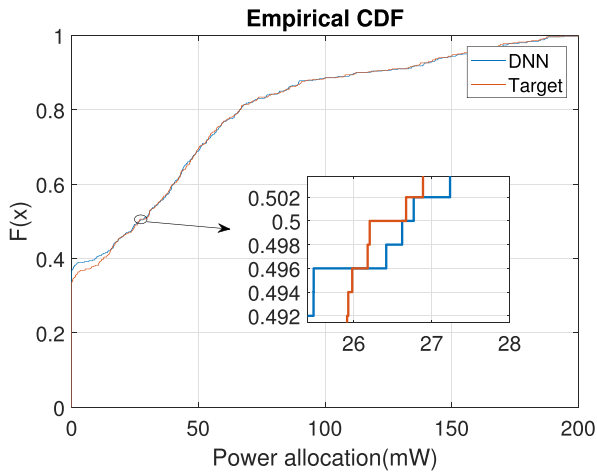
It is worth to point out that the number of filters dominates the loss function of the regression while the learning rate has more influence on the loss function of the allocation. We tried different hyper-parameters, considering the trade-offs between the number of filters and the implementation complexity and, between the learning rate and the training time. We see from Fig.4 that after 800 iterations, the loss function is close to 0; its value is actually 10^{-5} . This value is still expected to decrease as the number of iterations grows.

1) FIXED-POSITION UEs

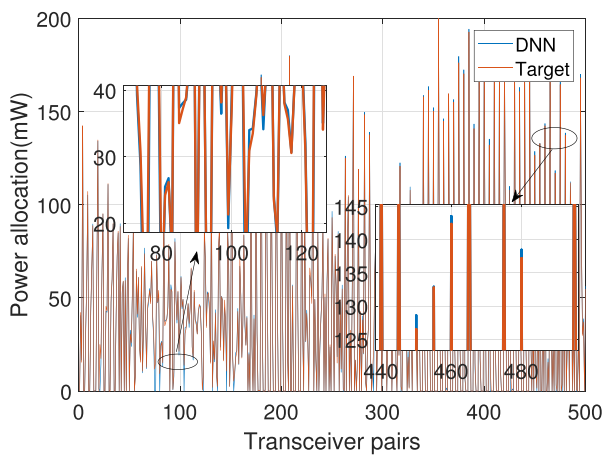
For the UE fixed-position scenario, we use the remaining 2000 data points for testing. Fig.5 compares the power allocation obtained by the DNN and the target, determined by the bisection heuristic. It does this by comparing the CDF of the power allocation from 2,000 testing data points. Each data point consists of 500 transmit-receive pairs (MNK), so the data in Fig.5 is the result of 10^6 pairs. A specific transmit-

$$PL_{n,m}^k = \begin{cases} -L - 35 \log_{10}(d_{n,m}^k), & d_{n,m}^k > d_1 \\ -L - 15 \log_{10}(d_1) - 20 \log_{10}(d_{n,m}^k), & d_0 < d_{n,m}^k \leq d_1 \\ -L - 15 \log_{10}(d_1) - 20 \log_{10}(d_0), & d_{n,m}^k \leq d_0 \end{cases} \quad (15)$$

$$L = 46.3 + 33.9 \log_{10}(f) - 13.82 \log_{10}(h_{RAA}) - (1.1 \log_{10}(f) - 0.7)h_u + 1.56 \log_{10}(f) - 0.8 \quad (16)$$



(a) CDF of power allocation.



(b) Line chart of transmit-receive pairs.

FIGURE 6. One random data from 2,000 data points with $N = 9$, $M = 11/12$, fixed UE scenario.

receive pair corresponds to a particular antenna of an RAA and the receiver of a UE. We found that the power allocated to around 30% of the channels is zero. This means that some antennas are not providing power to some users. We also observe that the power distribution generated by the DNN and the target almost overlap, which means that the two methods achieve similar results.

However, the CDF only describes the results statistically, different transmit-receive pairs may get very close transmission power, as a result, a similar CDF is obtained. To make it clear, we chose the results of one data point, randomly taken from the test set, which contains 2000 data points, see Fig.6.

Fig.6 (a) shows the CDF of the power allocation for one randomly chosen data from 2,000 test data, Fig.6 (b) shows, for this specific data, the target and the DNN power allocation for each of the 500 transmit-receive pairs. The errors between DNN and target (bisection) power allocation are mostly less than 1mW. However, compared to the right side in Fig.6 (a), the errors on the left side are much larger. This is because a max-pooling operation is used after the second convolutional layer in the DNN, causing the information of small values

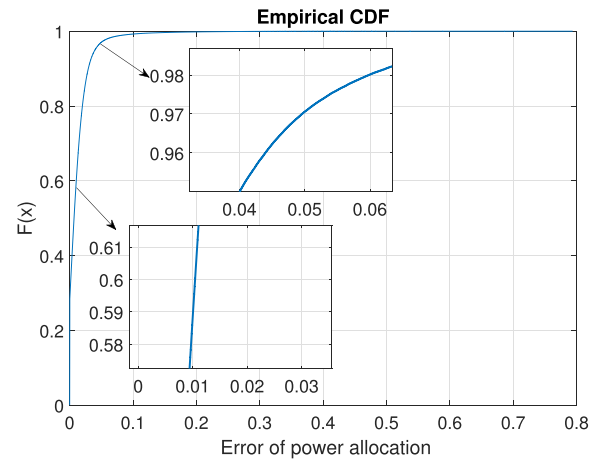


FIGURE 7. CDF of errors of power allocation with $N = 9$, $M = 11/12$, fixed UE scenario.

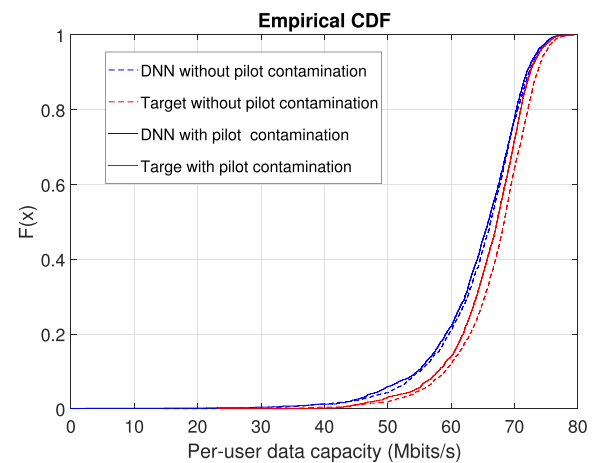


FIGURE 8. CDF of per-user data capacity with $N = 9$, $M = 11/12$, fixed UE scenario.

to disappear. Fig.6 also demonstrates that our DNN-based power allocation does not perform as well as Fig.5 shows: different transmit-receive pairs may get close or even same allocated power. This result leads us to make a statistical analysis of errors for the test data. We recorded the CDF of errors for the 2000 test data (i.e., 10^6 transmit-receive pairs) in Fig.7.

We observe that the relative error ($|Target-DNN|/p_l$) between the DNN allocation and the target is less than 0.01 for around 59% transmit-receive pairs and 0.05 for 97%. We calculate that the average error of power allocation for the whole test data set, is 0.0123.

Even if the DNN achieves a very similar power allocation as the bisection algorithm, a question arises: what is the effect of the allocation error on the per-user data capacity? Fig.8 gives some insight. It compares the CDF of the per-user data capacity for the DNN allocation and the target bisection heuristic allocation, based on the total test set containing 2,000 data points.

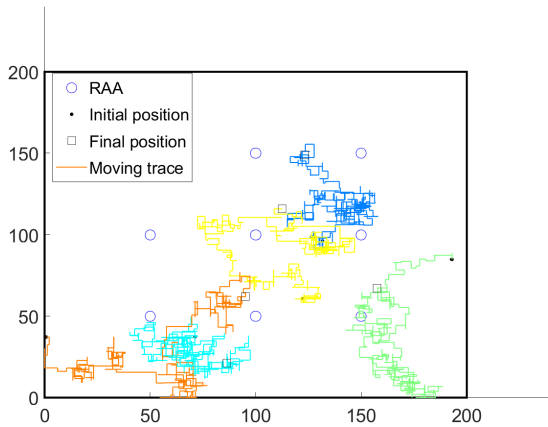


FIGURE 9. Moving-UE scenario traces.

We use the same trained DNN to perform power allocation for both the case with pilot contamination and without pilot contamination. We observe that the difference of DNN and target heuristic for both cases are almost the same, which means that the performance of our DNN is not noticeably affected by pilot contamination. This is because the power allocation depends on the long-term fading β . The pilot contamination only affects the performance of the channel estimation, causing a degradation of the per-user data capacity. Fig.8 also shows that the gap for both cases is no more than 4Mbits/s, which maps to at least 94% accuracy of approximation, measured as $(1-|Target-DNN|/target)$. So, we can conclude that the DNN approximates the behavior of the bisection heuristic very closely in the fixed-position UE scenario.

2) MOVING UEs

For the moving-UE scenario, each UE has an initial position that is random and uniformly distributed over the coverage area. Each UE moves in a random direction (up, down, left and right) with a randomly chosen velocity distributed uniformly between 0 and 5m/s. It maintains its speed and direction for 1s, before selecting a new speed and direction. When the UE reaches the boundary of the coverage area it reverses its direction of movement to stay within the coverage area. A trace of a realization of such a scenario, over the simulation period, for five UEs, is shown in Fig.9. We simulate a duration of 200 seconds at a random initial position of UEs and assume that the UEs will not get out of the coverage area. The DNN-based power allocation is performed every second, after the CC has estimated the CSI from the uplink pilots. We compare again the DNN-based method and the bisection-algorithm based heuristic. Fig.10 shows the CDF of the power allocation for both methods, over a 200s period, thus the results come from 10^5 transceiver pairs.

From Fig.10 we can see that DNN-based power allocation works worse than in fixed-position scenario. Much more errors occurred in $1 \sim 70$ mW. We also record the results of one random second out of the 200 seconds in Fig.11.

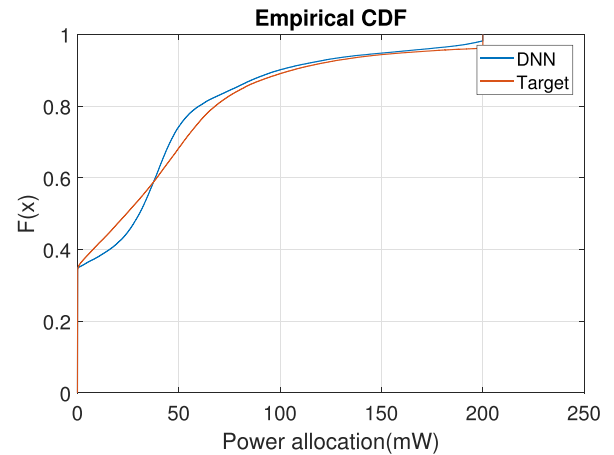
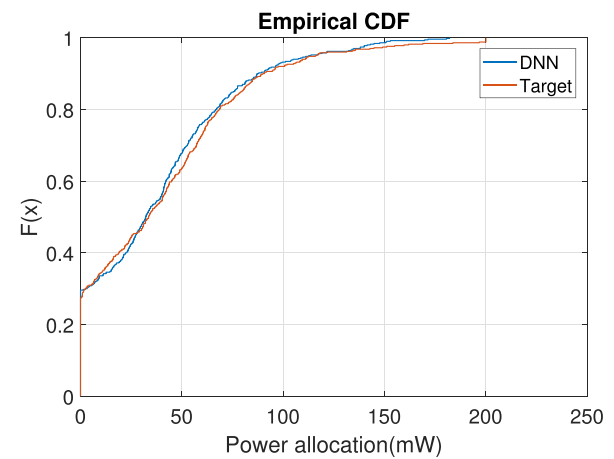
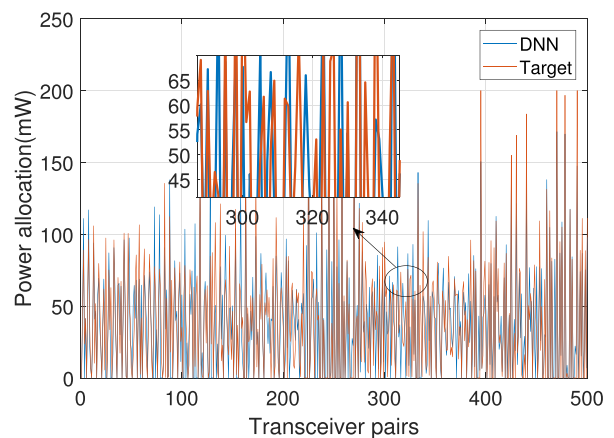


FIGURE 10. CDF of power allocation with $N = 9, M = 11/12$, moving UE scenario.



(a) CDF of power allocation.



(b) Line chart of transmit-receive pairs.

FIGURE 11. One random data from 200 seconds with $N = 9, M = 11/12$, moving UE scenario.

Compared to Fig.6, Fig.11 shows that the DNN-based power allocation approximates the trend of the target bisection method, but has larger errors with respect to the target

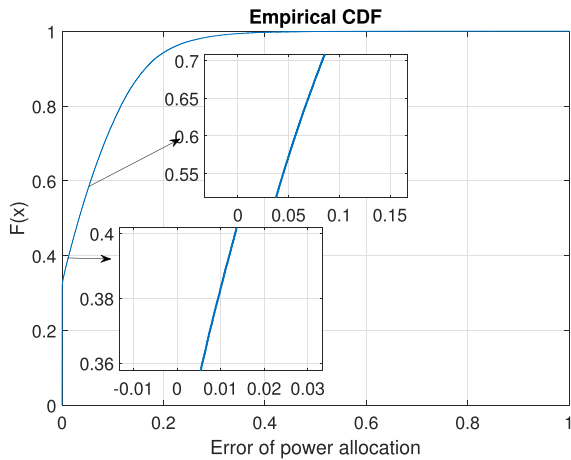


FIGURE 12. CDF of errors of power allocation with $N = 9, M = 11/12$, moving UE scenario.

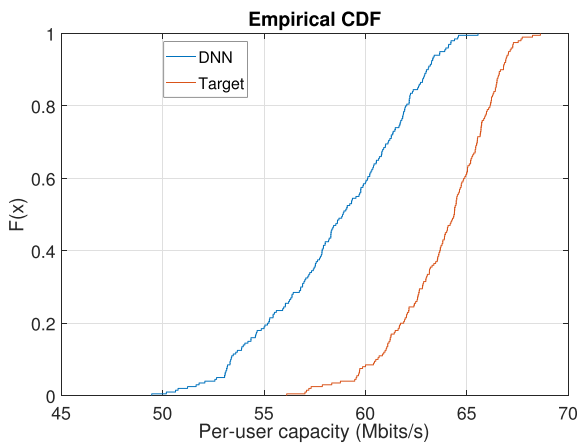


FIGURE 13. CDF of per-user data capacity with $N = 9, M = 11/12$, moving UE scenario.

than in the fixed-position scenario. The input to the DNN, consists of the long-term fading coefficients. However in a moving UE scenario, with random moving directions, random velocity of the UEs and shadowing, more noise is created for the input data. As in the fixed-UE scenario, we also record the statistical errors in Fig.12. We note that 38.1% of the errors are less than 0.01 and 57.5% of the errors are less than 0.05. The average error is 0.0609. This large error may be caused by underfitting, i.e., our DNN is insufficiently sensitive to the noise of input data. This could likely be solved by using a recurrent neural network. However this needs to be investigated in future research.

Finally, we calculate the final per-user capacity of moving UEs in Fig.13, the results come from a 200 second operation period.

The largest difference between the two methods occurs around 60 Mbits/s. The corresponding accuracy is around 85%. We also observe that the difference is the largest in the low capacity part, which implies that for the cases where

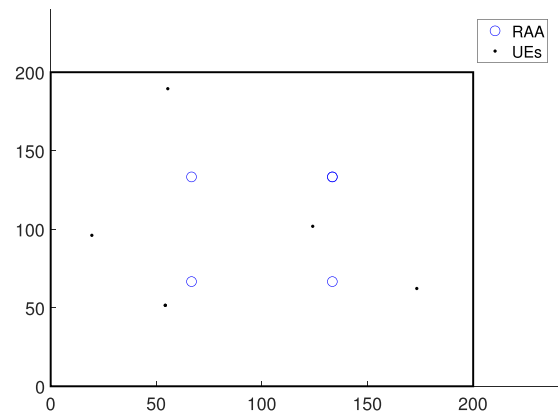


FIGURE 14. 4 RAA deployment.

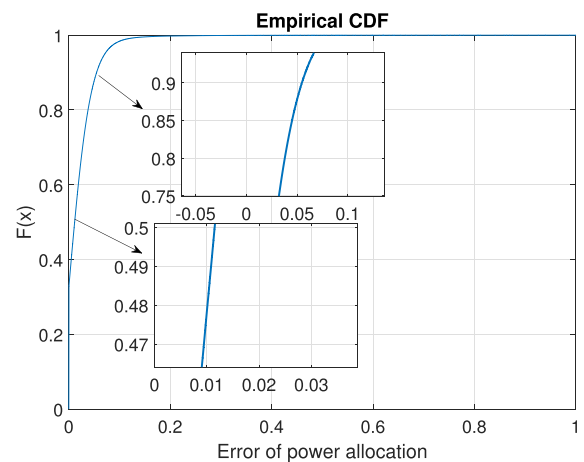


FIGURE 15. CDF of errors of power allocation with $N = 4, M = 25$ for the 4-RAA and fixed-UE scenario.

UEs are in an unfavorable location, e.g., due to shadowing, the DNN does not perform as well.

3) DIFFERENT DEPLOYMENTS OF RAAs

To see what impact a different deployment of the RAAs has on the effectiveness of the DNN method compared to the bisection heuristic, we consider the RAA configuration shown in Fig.14. 4 RAAs are placed in the coverage area and each RAA is equipped with 25 antennas, 5 single-antenna UEs are served. The UEs are stationary.

We generate 173800 data points using the bisection heuristic. Then 171800 data points are used for continuous training, i.e., totally 601800 (171800+430000) data points have been used to train our DNN. The rest, 2000 data points are for testing. We show the CDF of the power allocation error (with respect to the bisection heuristic) in Fig.15. We observe that 48% of the errors are less than 0.01 and 87.5% are less than 0.05. The average error is 0.0209.

Finally, in Fig.16 we show the CDF of per-user data capacity. The largest errors occur in the low capacity part (from 0 to 60 Mbits/s), where the poor channels add more difficulties for the DNN. While for the other part, the difference between

TABLE 2. Executing time of two methods to solve the power allocation problem.

Running time (count in seconds)										
DNN	0.0521	0.0723	0.0561	0.0492	0.0618	0.0632	0.0565	0.0682	0.0497	0.0572
Bisection	12.6715	11.8417	17.1962	22.5681	27.3842	26.3927	15.6314	22.5837	25.8864	30.9875

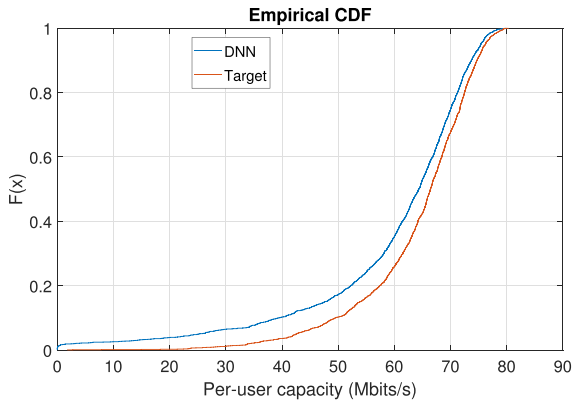


FIGURE 16. CDF of per-user data capacity with $N = 4$, $M = 25$ for the 4-RAA and fixed-UE scenario.

DNN and the bisection heuristic is no more than 6 Mbits/s, mapping to around 88% accuracy.

4) COMPARISON OF EXECUTION TIME

The time complexity of the DNN method is dramatically lower than the one of the bisection heuristic. Nevertheless, it is difficult to do a fair comparison in terms of processing time on a real implementation. One important reason is that the implementations will be based on different hardware architectures. Where the bisection heuristic is well suited to be executed on a classical (multicore) CPU architecture, the DNN will likely run on a hardware architecture optimized for machine leaning, e.g., an NPU. Nevertheless, it is revealing to see the huge difference, when both are executed on the same hardware. In Table 2 we recorded 10 random samples from 2,000 data points. We use the same platform, a 4 core Intel® Core i5-7300 CPU with 2.6 GHz frequency. The programs are both written in Python 3.7.2.

From table 2 it is obvious that the DNN-based power allocation requires much less processing time than the bisection heuristic and has less variation. For the DNN, the number of calculations is constant. The fluctuation of processing time comes from the calculation of different floating-point numbers and the inaccuracy of reading the system time. The CPU load also plays a role in this fluctuation. For the bisection heuristic, the time fluctuation mainly comes from different initializations, i.e., a different starting point of the search can make a large difference in the time needed to find the optimum.

VI. CONCLUSION

In this paper, we proposed a DNN to perform the power allocation in a CF massive MIMO system. The max-min power policy, which provides a fair quality of service for all

users, was considered. We showed that this NP-hard problem, for which a time-consuming heuristic is required to meet the time constraints imposed by the coherence time, can be better solved by a well-trained DNN. The DNN approach has a low time complexity while exhibiting a perform very close to the commonly used heuristic based on the bisection algorithm. The cost of using a DNN is the lengthy training required. But this should not be a problem in practice since it is done offline, before the network becomes operational.

We demonstrated the qualities of the DNN solution using a particular network configuration and scenario. These qualities, i.e., the close approximation of the behavior of a heuristic, should in principle hold for any network and scenario, since DNNs have been proven, given enough training, to be capable of approximating any function arbitrarily close.

However, there are some concerns that we did not address in this paper and that require further research. In particular finding the most suitable DNN structure, possibly related to the configuration of the CF massive MIMO system, and the determination of the hyper-parameters, which is important for the (prediction) capacity of the DNN. Another issue is that we considered a microwave network. To extend this to the mmWave domain, might be a more complex problem since the UEs are likely to be equipped with multiple antenna arrays, instead of a single antenna like we assumed in this paper. Furthermore, more complex channel models, e.g., Rician or the combination of Rician and Rayleigh, should be considered.

APPENDIX

A. BAYES ESTIMATION FOR THE CHANNEL

As $|\psi_k|^2 = 1$ and $\psi_k \psi_{k'}^H = 0$ for $k \neq k'$, we can rewrite (3) as:

$$y_{n,m}^k = p_p^{1/2} g_{n,m}^k + \psi_k^H \mathbf{w} + \psi_k^H \mathbf{t} \tag{A.1}$$

From the prior probability $g_{n,m}^k \sim \mathcal{CN}(0, \beta_{n,m}^k)$ and (A.1) we can easily obtain the prior probability $y_{n,m}^k \sim \mathcal{CN}(0, p_p \beta_{n,m}^k + \sigma^2 + \eta^2)$ as well as the posterior probability $y_{n,m}^k | g_{n,m}^k \sim \mathcal{CN}(p_p^{1/2} g_{n,m}^k, \sigma^2 + \eta^2)$, then see (A.2). Similar processing for the imaginary part yields (4).

B. ACHIEVABLE SINR OF UE

Reference [33] shows that the achievable data capacity of reliable data transmission [54] will not exceed the Shannon limit [55] in (9). The pre-log factor $(1 - K/\tau_c)$ refers to payload transmission, as K modulation symbols are used during channel estimation. $SINR_k$ is formulated as (B.1).

1) $|Desired|^2$

See (B.2). From (3) and (4) we have:

$$\begin{aligned} \mathbb{E}\{|\hat{\mathbf{g}}_{n,m}^k|^2\} &= \sum_{k=1}^K \frac{(p_p^{1/2} \beta_{n,m}^k)^2}{(p_p \beta_{n,m}^k + \sigma^2 + \eta^2)^2} \mathbb{E}\{|y_{n,m}^k|^2\} \\ &= \sum_{k=1}^K \frac{(p_p^{1/2} \beta_{n,m}^k)^2}{p_p \beta_{n,m}^k + \sigma^2 + \eta^2} \end{aligned} \quad (\text{B.3})$$

and:

$$\begin{aligned} &\mathbb{E}\{(g_{n,m}^k)^* \hat{g}_{n,m}^k\} \\ &= \mathbb{E}\{(g_{n,m}^k)^* \frac{p_p^{1/2} \beta_{n,m}^k}{p_p \beta_{n,m}^k + \sigma^2 + \eta^2} (p_p^{1/2} g_{n,m}^k + \boldsymbol{\psi}_k^H \mathbf{w} + \boldsymbol{\psi}_k^H \boldsymbol{\iota})\} \\ &= \frac{p_p (\beta_{n,m}^k)^2}{p_p \beta_{n,m}^k + \sigma^2 + \eta^2} \end{aligned} \quad (\text{B.4})$$

2) $\mathbb{E}|Fluctuation|^2$

See(B.6), then since $g_{n,m}^k \sim \mathcal{CN}(0, \beta_{n,m}^k)$, we have:

$$\begin{aligned} \mathbb{D}\{|g_{n,m}^k|^2\} &= \mathbb{D}\{(\text{Re}(g_{n,m}^k))^2 + (\text{Im}(g_{n,m}^k))^2\} \\ &= \mathbb{D}\{(\text{Re}(g_{n,m}^k))^2\} + \mathbb{D}\{(\text{Im}(g_{n,m}^k))^2\} \end{aligned} \quad (\text{B.7})$$

where $\text{Re}(g_{n,m}^k), \text{Im}(g_{n,m}^k) \sim \mathcal{N}(0, \beta_{n,m}^k/2)$, then:

$$\left(\frac{\text{Re}(g_{n,m}^k)}{\sqrt{\beta_{n,m}^k/2}}\right)^2, \left(\frac{\text{Im}(g_{n,m}^k)}{\sqrt{\beta_{n,m}^k/2}}\right)^2 \sim \chi(1) \quad (\text{B.8})$$

So we can get:

$$\mathbb{D}\left\{\left(\frac{\text{Re}(g_{n,m}^k)}{\sqrt{\beta_{n,m}^k/2}}\right)^2\right\} = \mathbb{D}\left\{\left(\frac{\text{Im}(g_{n,m}^k)}{\sqrt{\beta_{n,m}^k/2}}\right)^2\right\} = 2 \quad (\text{B.9})$$

$$\mathbb{D}\{(\text{Re}(g_{n,m}^k))^2\} = \mathbb{D}\{(\text{Im}(g_{n,m}^k))^2\} = \frac{(\beta_{n,m}^k)^2}{2} \quad (\text{B.10})$$

Plugging (B.6), (B.7) and (B.10):

$$\begin{aligned} &\mathbb{E}\{|Fluctuation|^2\} \\ &= \sum_{n=1}^N \sum_{m=1}^M p_{n,m}^k \frac{1}{\sum_{k'=1}^K \frac{(p_p^{1/2} \beta_{n,m}^{k'})^2}{p_p \beta_{n,m}^{k'} + \sigma^2 + \eta^2}} \frac{p_p (\beta_{n,m}^k)^3}{p_p \beta_{n,m}^k + \sigma^2 + \eta^2} \end{aligned} \quad (\text{B.11})$$

3) $\mathbb{E}|Interference|^2$

See (B.12). Finally, we get (10) by combining (B.5), (B.11), (B.12).

C. TIME COMPLEXITY OF THE PROPOSED HEURISTIC

To calculate the computational complexity of the bisection-based heuristic we should consider the worst case of the power allocation. The time complexity of the bisection algorithm is $O(\log_2(MN))$ according to [47].

We calculate the complexity of *Algorithm 1*. $SINR_{max}$ is approximately linearly increasing with MN while $SINR_{min}$ is 0 for the worst case. In each iteration of this loop, $SINR_{max}$ and $SINR_{min}$ are updated as:

$$SINR_{max} = SINR_{max} - \delta_1 \quad (\text{C.1})$$

$$SINR_{min} = SINR_{min} + \delta_2 \quad (\text{C.2})$$

where δ_1 and δ_2 are updating steps for $SINR_{max}$ and $SINR_{min}$, respectively. The calculation of δ_1 and δ_2 are very complicated because they are dependent on variable factors such as the actual deployment of the RAAs, the position of the users etc. Moreover, they are always changing in each loop of the iteration. However, we can derive from the convergence of our proposed algorithm that the smallest values of δ_1 and δ_2 (the worst case), are unrelated to the scale of the problem. This means that we can use the smallest value of δ_1 and δ_2 to address the worst case of the iteration to meet the termination condition:

$$\begin{aligned} &(SINR_{max} - iter \times \min(\delta'_1, \delta'_2)) \\ &\quad - (SINR_{min} + iter \times \min(\delta'_1, \delta'_2)) \leq \epsilon_1 \end{aligned} \quad (\text{C.3})$$

$$\begin{aligned} \mathbb{E}\{\text{Re}(g_{n,m}^k | y_{n,m}^k)\} &= \int_{-\infty}^{+\infty} x \frac{\text{P}\{\text{Re}(y_{n,m}^k | g_{n,m}^k)\} \times \text{P}\{\text{Re}(g_{n,m}^k)\}}{\text{P}\{\text{Re}(y_{n,m}^k)\}} dx \\ &= \int_{-\infty}^{+\infty} x \frac{\frac{1}{\sqrt{2\pi} \sqrt{\frac{\sigma^2 + \eta^2}{2}}} e^{-\frac{\text{Re}(y_{n,m}^k - \sqrt{p_p} x)^2}{2 \times \frac{\sigma^2 + \eta^2}{2}}} \times \frac{1}{\sqrt{2\pi} \sqrt{\frac{\beta_{n,m}^k}{2}}} e^{-\frac{\text{Re}(x^2)}{2 \times \frac{\beta_{n,m}^k}{2}}}}{\frac{1}{\sqrt{2\pi} \sqrt{\frac{p_p \beta_{n,m}^k + \sigma^2 + \eta^2}{2}}} e^{-\frac{\text{Re}(y_{n,m}^k)^2}{2 \times \frac{p_p \beta_{n,m}^k + \sigma^2 + \eta^2}{2}}}} dx = \frac{p_p^{1/2} \beta_{n,m}^k}{p_p \beta_{n,m}^k + \sigma^2 + \eta^2} \text{Re}(y_{n,m}^k) \end{aligned} \quad (\text{A.2})$$

$$SNR_k = \frac{|Desired|^2}{\mathbb{E}\{|Fluctuation|^2\} + \mathbb{E}\{|Interference|^2\} + \mathbb{E}\{|Noise|^2\}} \quad (\text{B.1})$$

$$|Desired|^2 = \left| \mathbb{E}\left\{ \sum_{n=1}^N \sum_{m=1}^M (g_{n,m}^k)^* (p_{n,m}^k)^{1/2} f_{n,m}^k q_k \right\} \right|^2 = \left| \mathbb{E}\left\{ \sum_{n=1}^N \sum_{m=1}^M (g_{n,m}^k)^* (p_{n,m}^k)^{1/2} \frac{\hat{g}_{n,m}^k}{\sqrt{\mathbb{E}\{|\hat{\mathbf{g}}_{n,m}^k|^2\}}} \right\} \right|^2 \quad (\text{B.2})$$

$$|Desired|^2 = \mathbb{E}\left\{\left|\sum_{n=1}^N \sum_{m=1}^M (g_{n,m}^k)^* (p_{n,m}^k)^{1/2} f_{n,m}^k q_k\right|^2\right\} = \left|\sum_{n=1}^N \sum_{m=1}^M (p_{n,m}^k)^{1/2} \frac{1}{\sqrt{\sum_{k=1}^K \frac{(p_p^{1/2} \beta_{n,m}^k)^2}{p_p \beta_{n,m}^k + \sigma^2 + \eta^2}}} \frac{p_p (\beta_{n,m}^k)^2}{p_p \beta_{n,m}^k + \sigma^2 + \eta^2}\right|^2 \quad (B.5)$$

$$\begin{aligned} \mathbb{E}\{|Fluctuation|^2\} &= \mathbb{E}\left\{\left|\sum_{n=1}^N \sum_{m=1}^M (g_{n,m}^k)^* (p_{n,m}^k)^{1/2} f_{n,m}^k q_k - \mathbb{E}\left\{\sum_{n=1}^N \sum_{m=1}^M (g_{n,m}^k)^* (p_{n,m}^k)^{1/2} f_{n,m}^k q_k\right\}\right|^2\right\} \\ &= \mathbb{D}\left\{\sum_{n=1}^N \sum_{m=1}^M (g_{n,m}^k)^* (p_{n,m}^k)^{1/2} f_{n,m}^k q_k\right\} \\ &= \sum_{n=1}^N \sum_{m=1}^M p_{n,m}^k \frac{1}{\sum_{k'=1}^K \frac{(p_p^{1/2} \beta_{n,m}^{k'})^2}{p_p \beta_{n,m}^{k'} + \sigma^2 + \eta^2}} \frac{p_p (\beta_{n,m}^k)^2}{(p_p \beta_{n,m}^k + \sigma^2 + \eta^2)^2} [p_p \mathbb{D}\{|g_{n,m}^k|^2\} + \beta_{n,m}^k \sigma^2 + \beta_{n,m}^k \eta^2] \quad (B.6) \end{aligned}$$

$$\begin{aligned} \mathbb{E}\{|Interference|^2\} &= \mathbb{E}\left\{\left|\sum_{n=1}^N \sum_{m=1}^M (g_{n,m}^k)^* \sum_{k' \neq k} (p_{n,m}^{k'})^{1/2} f_{n,m}^{k'}\right|^2\right\} = \mathbb{D}\left\{\sum_{n=1}^N \sum_{m=1}^M (g_{n,m}^k)^* \sum_{k' \neq k} (p_{n,m}^{k'})^{1/2} f_{n,m}^{k'}\right\} \\ &= \sum_{k' \neq k} \sum_{n=1}^N \sum_{m=1}^M p_{n,m}^{k'} \frac{1}{\sum_{i=1}^K \frac{(p_p^{1/2} \beta_{n,m}^i)^2}{p_p \beta_{n,m}^i + \sigma^2 + \eta^2}} \frac{p_p (\beta_{n,m}^{k'})^2}{p_p \beta_{n,m}^{k'} + \sigma^2 + \eta^2} \beta_{n,m}^k \quad (B.12) \end{aligned}$$

where δ'_1 and δ'_2 are the smallest values of δ_1 and δ_2 , respectively, $iter$ is the number of iterations of Step 2 in *Algorithm 1*. So we can calculate $iter = \theta MN$, where θ is a linear coefficient. The complexity of step 8-22 in *Algorithm 1* is:

$$O(2 \log_2(p_l/\epsilon_3) MN (MNK^2)) = O(M^2 N^2 K^2) \quad (C.4)$$

D. TIME COMPLEXITY OF THE PROPOSED DNN

The computational complexity of the DNN lies in the forward propagation. In the first convolutional layer, we use $5 \times M \times Q$ filters with stride $[1, M]$. The input is one $K \times MN$ matrix so the output of the first convolutional layer is a $K \times N \times Q$ matrix, which implies $K \times N \times Q$ convolutional computations. So, the complexity of the first convolutional layer is $O(5KMNQ)$. Similarly, the complexity of the second convolutional layer is $O(25KMNQ^2)$. The first two fully connected layers in the regression processing have a complexity:

$$\begin{aligned} &O(\lceil K/2 \rceil \times \lceil N/2 \rceil \times Q \times \lceil K/2 \rceil \times \lceil N/2 \rceil \times \lceil Q/2 \rceil \\ &\quad + \lceil K/2 \rceil \times \lceil N/2 \rceil \times \lceil Q/2 \rceil) \\ &\approx O(K^2 N^2 Q^2 / 32 + KNQ/8) \quad (D.1) \end{aligned}$$

The last fully connected layer in the optimization processing has a complexity of $O(2K^2 N^2 M^2 + KNM)$. So, we can get the total time complexity as:

$$\begin{aligned} &O(5KNMQ + 25KNMQ^2 \\ &\quad + K^2 N^2 Q^2 / 32 + KNQ + 2K^2 N^2 M^2 + KNM) \\ &= O(K^2 N^2 M^2) \quad (D.2) \end{aligned}$$

Finally we get the total time complexity of our proposed heuristic:

$$\begin{aligned} &O(\log_2(MN) \times \theta MN \times M^2 N^2 K^2) \\ &= O(\log_2(MN) M^3 N^3 K^2) \quad (C.5) \end{aligned}$$

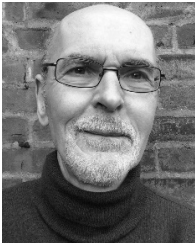
REFERENCES

- [1] L. Lu, G. Y. Li, A. L. Swindlehurst, A. Ashikhmin, and R. Zhang, "An overview of massive MIMO: Benefits and challenges," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 742–758, Oct. 2014.
- [2] H. Quoc Ngo, E. G. Larsson, and T. L. Marzetta, "Energy and spectral efficiency of very large multiuser MIMO systems," *IEEE Trans. Commun.*, vol. 61, no. 4, pp. 1436–1449, Apr. 2013.
- [3] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 186–195, Feb. 2014.
- [4] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, "Cell-free massive MIMO versus small cells," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1834–1850, Mar. 2017.
- [5] J. Zhang, S. Chen, Y. Lin, J. Zheng, B. Ai, and L. Hanzo, "Cell-free massive MIMO: A new next-generation paradigm," *IEEE Access*, vol. 7, pp. 99878–99888, 2019.
- [6] G. Interdonato, E. Björnson, H. Quoc Ngo, P. Frenger, and E. G. Larsson, "Ubiquitous cell-free massive MIMO communications," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, p. 197, Dec. 2019.
- [7] Q. Wang, D. Debbarma, A. Lo, I. Niemegeers, and S. H. de Groot, "Sum-rate performance of large centralized and distributed MU-MIMO systems in indoor WLAN," *Infocommunications J.*, vol. 7, no. 3, pp. 17–24, 2015.
- [8] R. Hamdi, E. Driouch, and W. Ajib, "Resource allocation in downlink large-scale MIMO systems," *IEEE Access*, vol. 4, pp. 8303–8316, 2016.
- [9] E. Björnson, J. Hoydis, and L. Sanguinetti, "Massive MIMO networks: Spectral, energy, and hardware efficiency," *Found. Trends Signal Process.*, vol. 11, nos. 3–4, pp. 154–655, 2017.
- [10] T. L. Marzetta, "Massive MIMO: An introduction," *Bell Labs Tech. J.*, vol. 20, pp. 11–22, Mar. 2015.
- [11] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [12] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, Jan. 1989.
- [13] G. Gui, H. Huang, Y. Song, and H. Sari, "Deep learning for an effective nonorthogonal multiple access scheme," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8440–8450, Sep. 2018.
- [14] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2224–2287, 3rd Quart., 2019.

- [15] E. Bjornson, L. Sanguinetti, J. Hoydis, and M. Debbah, "Optimal design of energy-efficient multi-user MIMO systems: Is massive MIMO the answer?" *IEEE Trans. Wireless Commun.*, vol. 14, no. 6, pp. 3059–3075, Jun. 2015.
- [16] X. Gao, O. Edfors, F. Rusek, and F. Tufvesson, "Massive MIMO performance evaluation based on measured propagation data," *IEEE Trans. Wireless Commun.*, vol. 14, no. 7, pp. 3899–3911, Jul. 2015.
- [17] G. Gui, N. Liu, F. Adachi, and L. Xu, "Low-complexity large-scale multiple-input multiple-output channel estimation using affine combination of sparse least mean square filters," *IET Commun.*, vol. 9, no. 17, pp. 2168–2175, Nov. 2015.
- [18] Z. Liu and L. Dai, "A comparative study of downlink MIMO cellular networks with co-located and distributed base-station antennas," *IEEE Trans. Wireless Commun.*, vol. 13, no. 11, pp. 6259–6274, Nov. 2014.
- [19] J. Joung, Y. K. Chia, and S. Sun, "Energy-efficient, large-scale distributed-antenna system (L-DAS) for multiple users," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 954–965, Oct. 2014.
- [20] N. Akbar, E. Björnson, E. G. Larsson, and N. Yang, "Downlink power control in massive MIMO networks with distributed antenna arrays," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.
- [21] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for interference management," *IEEE Trans. Signal Process.*, vol. 66, no. 20, pp. 5438–5453, Oct. 2018.
- [22] H. Huang, S. Guo, G. Gui, Z. Yang, J. Zhang, H. Sari, and F. Adachi, "Deep learning for physical-layer 5G wireless techniques: Opportunities, challenges and solutions," *IEEE Wireless Commun.*, vol. 27, no. 1, pp. 214–222, Feb. 2020.
- [23] N. Samuel, T. Diskin, and A. Wiesel, "Deep MIMO detection," in *Proc. IEEE 18th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jul. 2017, pp. 1–5.
- [24] X. Yan, F. Long, J. Wang, N. Fu, W. Ou, and B. Liu, "Signal detection of MIMO-OFDM system based on auto encoder and extreme learning machine," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 1602–1606.
- [25] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 114–117, Feb. 2018.
- [26] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Deep learning-based channel estimation for beamspace mmWave massive MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 852–855, Oct. 2018.
- [27] Y. Jin, J. Zhang, S. Jin, and B. Ai, "Channel estimation for cell-free mmWave massive MIMO through deep learning," *IEEE Trans. Veh. Technol.*, vol. 68, no. 10, pp. 10325–10329, Oct. 2019.
- [28] L. Sanguinetti, A. Zappone, and M. Debbah, "Deep learning power allocation in massive MIMO," in *Proc. 52nd Asilomar Conf. Signals, Syst., Comput.*, Oct. 2018, pp. 1257–1261.
- [29] M. A. Wijaya, K. Fukawa, and H. Suzuki, "Intercell-interference cancellation and neural network transmit power optimization for MIMO channels," in *Proc. IEEE 82nd Veh. Technol. Conf. (VTC-Fall)*, Sep. 2015, pp. 1–5.
- [30] T. Van Chien, T. Nguyen Canh, E. Björnson, and E. G. Larsson, "Power control in cellular massive MIMO with varying user activity: A deep learning solution," 2019, *arXiv:1901.03620*. [Online]. Available: <http://arxiv.org/abs/1901.03620>
- [31] T. L. Marzetta, *Fundamentals of Massive MIMO*. Cambridge, U.K.: Cambridge Univ. Press, 2016.
- [32] H. Jiang and G. Gui, *Channel Modeling in 5G Wireless Communication Systems*. New York, NY, USA: Springer, 2020.
- [33] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge, U.K.: Cambridge Univ. Press, 2005.
- [34] H. Q. Ngo, H. Tataria, M. Matthaiou, S. Jin, and E. G. Larsson, "On the performance of cell-free massive MIMO in ricean fading," in *Proc. 52nd Asilomar Conf. Signals, Syst., Comput.*, Oct. 2018, pp. 980–984.
- [35] P. Marsch and G. Fettweis, "Uplink CoMP under a constrained backhaul and imperfect channel knowledge," *IEEE Trans. Wireless Commun.*, vol. 10, no. 6, pp. 1730–1742, Jun. 2011.
- [36] O. Simeone, O. Somekh, E. Erkip, H. V. Poor, and S. Shamai, "A broadcast approach to robust communications over unreliable multi-relay networks," in *Proc. Inf. Theory Appl. Workshop*, Feb. 2009, pp. 334–340.
- [37] J. Jose, A. Ashikhmin, T. L. Marzetta, and S. Vishwanath, "Pilot contamination and precoding in multi-cell TDD systems," *IEEE Trans. Wireless Commun.*, vol. 10, no. 8, pp. 2640–2651, Aug. 2011.
- [38] H. Q. Ngo, T. L. Marzetta, and E. G. Larsson, "Analysis of the pilot contamination effect in very large multicell multiuser MIMO systems for physical channel models," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2011, pp. 3464–3467.
- [39] K. Appaiah, A. Ashikhmin, and T. L. Marzetta, "Pilot contamination reduction in multi-user TDD systems," in *Proc. IEEE Int. Conf. Commun.*, May 2010, pp. 1–5.
- [40] H. Liu, J. Zhang, X. Zhang, A. Kurniawan, T. Juhana, and B. Ai, "Tabu-Search-Based pilot assignment for cell-free massive MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 2286–2290, Feb. 2020.
- [41] G. Casella, "An introduction to empirical bayes data analysis," *Amer. Statistician*, vol. 39, no. 2, pp. 83–87, May 1985.
- [42] Z. Chen and E. Bjornson, "Channel hardening and favorable propagation in cell-free massive MIMO with stochastic geometry," *IEEE Trans. Commun.*, vol. 66, no. 11, pp. 5205–5219, Nov. 2018.
- [43] J. Lee and N. Jindal, "High SNR analysis for MIMO broadcast channels: Dirty paper coding versus linear precoding," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4787–4792, Dec. 2007.
- [44] G. Caire and S. Shamai, "On the achievable throughput of a multiantenna Gaussian broadcast channel," *IEEE Trans. Inf. Theory*, vol. 49, no. 7, pp. 1691–1706, Jul. 2003.
- [45] H. Huang, Y. Peng, J. Yang, W. Xia, and G. Gui, "Fast beamforming design via deep learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 1065–1069, Jan. 2020.
- [46] G. Gui, H. Sari, and E. Biglieri, "A new definition of fairness for non-orthogonal multiple access," *IEEE Commun. Lett.*, vol. 23, no. 7, pp. 1267–1271, Jul. 2019.
- [47] K. Sikorski, "Bisection is optimal," *Numerische Math.*, vol. 40, no. 1, pp. 111–117, Feb. 1982.
- [48] D. A. Palmer and M. Florea, "Neural processing unit," U.S. Patent 8 655 815, Feb. 18, 2014.
- [49] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
- [50] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," 2012, *arXiv:1212.5701*. [Online]. Available: <http://arxiv.org/abs/1212.5701>
- [51] Z. Wang, E. K. Tameh, and A. R. Nix, "Joint shadowing process in urban peer-to-peer radio channels," *IEEE Trans. Veh. Technol.*, vol. 57, no. 1, pp. 52–64, Jan. 2008.
- [52] A. Tang, J. Sun, and K. Gong, "Mobile propagation loss with a low base station antenna for NLOS street microcells in urban area," in *Proc. IEEE VTS 53rd Veh. Technol. Conf., Spring*, vol. 1, May 2001, pp. 333–336.
- [53] Y. Singh, "Comparison of okumura, hata and COST-231 models on the basis of path loss and signal strength," *Int. J. Comput. Appl.*, vol. 59, no. 11, pp. 37–41, 2012.
- [54] D. Hammerschmidt, T. Dittfeld, and S. Brewerton, "Reliable data transmission with reduced bit error rate," U.S. Patent 8 887 022, Nov. 11, 2014.
- [55] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1," in *Proc. IEEE Int. Conf. Commun. (ICC)*, vol. 2, May 1993, pp. 1064–1070.



YU ZHAO received the B.S. and M.S. degrees in control science and engineering from Xi'an Jiaotong University, China, in 2014 and 2017, respectively. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, Eindhoven University of Technology, The Netherlands. His current research interests include massive multiple-input multiple-output, deep learning, and artificial intelligent algorithms.



IGNAS G. NIEMEGEERS received the M.Sc. degree in electrical engineering from the University of Gent, Belgium, in 1970, and the M.Sc. and Ph.D. degrees in computer engineering from Purdue University, USA, in 1972 and 1978, respectively. From 1978 to 1981, he was a Designer of one of the first commercial packet switching networks with Bell Telephone Manufacturing Company, Antwerp, Belgium. From 1981 to 2002, he was a Professor with the University of Twente, The

Netherlands, where he was a Scientific Director of the Centre for Telematics and Information Technology (CTIT), from 1995 to 2002. From 2002 to 2012, he chaired the Telecommunications Department, Delft University of Technology, where he has been an Emeritus Professor, since August 2012. He has been an Advisor with the Centre for Wireless Technology, Eindhoven University of Technology, The Netherlands, since August 2012. Since March 2016, he has been a Part-Time Guest Professor in future optically-supported wireless networks with the Eindhoven University of Technology. He was involved in many European research projects. He has supervised 42 Ph.D. students. He has authored or coauthored around 300 articles. His current research interests include machine learning in radio access network architectures, radio-over-fiber, mm-wave technologies, the energy-harvesting IoT, and V2X communication. He was a Reviewer for many projects.



SONIA HEEMSTRA DE GROOT received the M.Sc. degree in electrical engineering from the Philips International Institute/NUFFIC and the Universidad Nacional de Mar del Plata, Argentina, The Netherlands, and the Ph.D. degree in electrical engineering from the University of Twente, The Netherlands, in 1990. She was a Chief Scientist with the Twente Institute for Wireless and Mobile, from 2003 to 2014. Since 2012, she has been a Full Professor with the Eindhoven University of

Technology, where she holds the Part-Time Chair in heterogeneous network architectures. In September 2016, she was the Director of the Center for Wireless Technology, Eindhoven. She has held an Assistant position and an Associate Professor position with the University of Twente. She also held a Full-Professor position with the Delft University of Technology, in personal and ambient networking. She was a Senior Researcher with Ericsson EuroLab, The Netherlands. She has Co-Founded the Twente Institute for Wireless and Mobile. Her research interests include wireless and mobile communications, 5G, vehicular networks, wireless indoor communications, the Internet of Things, and wireless security.

• • •