Enhancing Energy Efficiency of D-MIMO Networks: Scalable Clustering and Deep Learning-based Power Control

Wilker de O. Feitosa *, Igor M. Guerreiro *, Fco. Rodrigo P. Cavalcanti *,
Juno V. Saraiva *, Maria Clara R. Lobão *, Yuri C. B. Silva *, Gabor Fodor †‡
Wireless Telecom Research Group (GTEL), Federal University of Ceará, Fortaleza 60455-760, Brazil.*
Ericsson Research, 16480 Stockholm, Sweden, Division of Decision and Control,†
KTH Royal Institute of Technology, 11428 Stockholm, Sweden.‡
e-mails: {wilker, igor, rodrigo, juno, clara, yuri}@gtel.ufc.br *, gabor.fodor@ericsson.com†

***Abstract*—Distributed multiple-input and multiple-output (D-MIMO) technology is a promising candidate to be integrated in beyond fifth generation networks offering uniform quality of service along the network coverage and higher overall system capacity. It leverages the cooperation of many antenna arrays spread over the coverage area that jointly and coherently serve user equipment devices. While the spectrum efficiency benefits of D-MIMO (sometimes also referred to as cell-free technology) are well documented, the corresponding energy efficiency (EE) of such networks has received more attention in recent years as concerns about sustainability become central in future 6G systems. In this context, this paper proposes and investigates the combined effect of intelligent access point clustering and power control to enhance D-MIMO's EE. While many works on D-MIMO assume that the whole network serves every user, we consider scalability and massive MIMO constraints to address the practical issues of a fully connected network in terms of high computational complexity, elevated signaling bandwidth, and limitations of MIMO's spatial degrees of freedom. To this end, we propose a resource-aware graph-based clustering method combined with a deep-learning-based power control. Comprehensive computer simulations demonstrate that the proposed strategy significantly enhances the network's EE, while producing comparable spectral efficiency performance to a fully connected scenario, while also outperforming a state-of-the-art existing clustering approach.**

***Index Terms*—Energy efficiency, D-MIMO, clustering, power control, graph reinforcement learning.**

I. Introduction

The next generation of mobile communication systems, namely sixth generation (6G), has been designed to provide higher data rates, more reliable communication and higher energy efficiency than in current fifth generation (5G) [1]. In order to meet these demands, 6G systems have a few challenges to overcome. Among those challenges, practical distributed multiple-input and multiple-output (D-MIMO) deployment and scalable power control are immediate concerns [2].

This work was supported by Ericsson Research (technical coop. contracts UFC.50 and UFC.53), by Coordenação de Aperfeiçoamento de Pessoal de Nivel Superior - Brasil (CAPES) - Finance Code 001. Gabor Fodor was also supported by the SSF FUS21-0004 SAICOM project. Igor, Rodrigo and Yuri were also supported by Brasil 6G project (01245.010604/2020-14, RNP and MCTI)

D-MIMO architecture is a more efficient alternative to that of traditional cellular systems, as it provides better coverage, higher data rates and more reliable connection [3]. This architecture consists of a large number of access points (APs) equipped with small antenna arrays distributed over a relatively small coverage area, all connected to a coordination central processing unit (CPU) via a backhaul link. The APs operate in a cooperative manner often performing joint coherent transmission or reception.

An important concern regarding D-MIMO systems is their scalability in terms of computational complexity and signaling [4]. In order to address these issues, the APs form user-centric clusters. The channel estimation, as well as the combining schemes, are designed based on local channel state information (CSI), lowering the signaling load and improving scalability through the network [3].

Reinforcement learning (RL) techniques have had success in solving complex non-convex problems in a wide range of domains [5]–[7]. Regarding power allocation, deep reinforcement learning (DRL) and the graph reinforcement learning (GRL) techniques have shown considerable advances in solving both sum-rate and max-min fairness problems [8], [9].

In this work, the clustering problem in D-MIMO network with focus on the scalability and energy efficiency is addressed. To this end, a graph-based clustering algorithm is proposed and used along with a DRL power control solution. More specifically, since the power control problem depends on the clustering structure of the network, this work starts by tackling the clustering problem in a way that efficiently assigns the APs to the user equipment devices (UEs) aiming at providing strong and reliable connections while keeping the network usage to a minimum. Then, when the power control algorithm starts, the UEs transmit only to APs that best serve them, making the DRL solution act on fewer links with stronger signal-to-interference-plus-noise ratio (SINR), effectively reducing its search space. As a consequence, the training time is also shortened. This coupling of graph-based techniques with DRL methods is also jointly referred as GRL. The system performance is evaluated in terms of spectral

ISBN 978-3-903176-73-7

efficiency (SE) and energy efficiency (EE).

The results show that the clustering solution outperforms the state of the art, providing around 0.25 bps/Hz higher SE at the 50th percentile. The proposed scheme is also superior in terms of EE up until the 60th percentile. Moreover, the DRL solution provides gains in both key performance indicators (KPIs) for all users and the combination of our proposed algorithm with the DRL approach shows increasing benefits for UEs with poor channel conditions. Finally, the proposed GRL scheme exhibits all those benefits with a considerable smaller training time compared to other scenarios (typically one fifth of their training time).

II. System Model

The uplink (UL) of a D-MIMO system is considered, composed of L APs equipped with a uniform linear array (ULA) of N antennas and K single-antenna UEs. $\mathbf{\Omega}$ is the AP-UE association matrix related to the clusterization, with $\boldsymbol{\omega_l}$ being the l-th row, and its norm represents the number of connections of the l-th AP. The APs are connected via ideal backhaul links to a CPU and the network operates in time division duplex mode.

The system processing is implemented in a partially distributed manner, where the symbol estimation is done locally by the APs, which then send their estimates to the CPU for final decoding. This is essential to ensure the scalability of the network. Given that the APs estimate the symbols transmitted by the UEs, they cannot serve more than N UEs at the same time. Because of this, the ideal D-MIMO network where all the APs serve all the UEs simultaneously becomes unfeasible. Hence, to overcome this limitation, the user-centric clustering strategy is applied. It is also worth noting that for the degrees of freedom (DoF) of the AP to be equal to N, the AP must have at least N radio frequency (RF) chains. In the scope of this work, all the APs employ fully-digital beamforming and have N RF chains.

A. Propagation Model

The channel between UE k and AP l is modelled as a Rician channel, denoted as $\mathbf{h}_{k,l} \in \mathbb{C}^N$ and described as:

$$\mathbf{h}_{k,l} = \left[\sqrt{\frac{\bar{K}_{k,l}}{\bar{K}_{k,l}+1}} \mathbf{a}_{k,l} + \sqrt{\frac{1}{\bar{K}_{k,l}+1}} \mathbf{h}_{k,l}^{(w)} \right] \sqrt{\beta_{k,l}}\,, \quad (1)$$

where $\bar{K}_{k,l}$ is the Rician K-factor, $\mathbf{h}_{k,l}^{(w)} \sim \mathcal{N}(0, \mathbf{R}_{k,l})$ is the non-line of sight (NLOS) component composed by a spatially correlated Rayleigh fading with $\mathbf{R}_{k,l} \in \mathbb{C}^{N \times N}$ being the spatial correlation matrix defined by [10]. The term $\mathbf{a}_{k,l}$ represents the antenna array steering vector that corresponds to the line of sight (LOS) component of the channel, and is given by $\mathbf{a}_{k,l} = \left[u_{k,l}^1 \dots u_{k,l}^M \right]^T$, with:

$$u_{k,l}^m = e^{j\frac{2\pi}{\lambda}(m-1)d\sin\theta_{k,l}\cos\phi_{k,l}}\,, \quad (2)$$

where $\theta_{k,l}$ and $\phi_{k,l}$ are the angle of arrival (AoA) for elevation and azimuth, respectively, d the antenna spacing and λ the wave length. Finally, $\beta_{k,l}$ represents the large-scale propagation parameters and is defined as:

$$\beta_{k,l} = 10^{\frac{\mathrm{PL}_{k,l}+\mathrm{SH}_{k,l}}{10}}\,, \quad (3)$$

where $\mathrm{PL}_{k,l}$ is the path-gain and $\mathrm{SH}_{k,l}$ is the lognormal correlated shadowing with standard deviation of σ_{SH}.

All large-scale parameters as well as the Rician K-factor and the probability of a UE having a LOS component are modelled according to the technical specification [11].

B. Channel Estimation

In this work, a channel estimation based on pilot transmission is performed. Consider $\tau_p \leq K$ orthogonal pilot sequences of length τ_p. The estimation is implemented as described in [3], in which the signal received by the AP is correlated with the normalized pilot sequence transmitted by the UEs, resulting in:

$$\mathbf{z}_{t_k,l} = \sum_{i \in \mathcal{P}_k} \sqrt{p_i \tau_p} \mathbf{h}_{i,l} + \mathbf{n}_{t_k,l}\,, \quad (4)$$

where $\mathbf{z}_{t_k,l}$ is the signal received by the AP l arriving from the users that transmitted the pilot t_k, $\mathcal{P}_k$ is the set of UEs that share the pilot t_k, p_i is the uplink transmitted power of UE i, and $\mathbf{n}_{t_k,l} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_N)$ is the noise at the receiver normalized by the pilot sequence t_k. By using an minimum mean square error (MMSE) estimator and defining $\mathbf{\Psi}_{t_k,l} = \mathbb{E}\{\mathbf{z}_{t_k,l}\mathbf{z}_{t_k,l}^H\}$, the channel estimates are computed using the correlated signal $\mathbf{z}_{t_k,l}$, resulting in:

$$\hat{\mathbf{h}}_{k,l} = \sqrt{p_k \tau_p} \mathbf{R}_{k,l} \mathbf{\Psi}_{t_k,l}^{-1} \mathbf{z}_{t_k,l}\,. \quad (5)$$

C. Receiver Processing and SE

Considering the following signal received by the l-th AP:

$$\mathbf{y}_l = \sum_{k=1}^{K} \mathbf{h}_{k,l} s_k + \mathbf{n}_l\,, \quad (6)$$

where $\mathbf{y}_l \in \mathbb{C}^N$ is the baseband signal received by the AP and s_k is the symbol sent by the user k. A combiner must be applied by the AP so the symbol can be decoded. Given that the D-MIMO system operates inherently in a distributed manner, the combiner is determined by each AP using only its local estimates. The optimal MMSE combiner that maximizes the SE when only the local information of the AP is available is described in [12]. However, the complexity of this model grows with the number of users in the network. In order to maintain the scalability of the processing operation, a partial version of the local combiner was developed in [3]. The local partial MMSE (LP-MMSE) is defined as:

$$\mathbf{v}_{k,l} = p_k \left(\sum_{i \in \mathcal{D}_l} p_i \left(\hat{\mathbf{h}}_{i,l} \hat{\mathbf{h}}_{i,l}^H + \mathbf{C}_{i,l} \right) + \sigma^2 \mathbf{I}_N \right)^{-1} \mathbf{D}_{k,l} \hat{\mathbf{h}}_{k,l}\,, \quad (7)$$

where $\mathbf{C}_{i,l} = \mathbb{E}\{\hat{\mathbf{h}}_{i,l}\hat{\mathbf{h}}_{i,l}^H\}$, $\mathcal{D}_l$ is the group of UEs served by the AP l and $\mathbf{D}_{k,l}$ is a block-diagonal matrix related to the cluster of arrays that serves the user k [4].

After receiving all the estimates, the CPU linearly decodes the symbol information based on weights per-UE and per-AP. Let $\mathbf{g}_{k,i} = [\mathbf{v}_{k,1}^H \mathbf{h}_{i,1} \dots \mathbf{v}_{k,L}^H \mathbf{h}_{i,L}]^T$ be the combined channels between the UE k and each of the APs. The large scale fading decoding weights that maximize the SINR can be found with the following expression [3]:

$$\mathbf{w}_k = p_k \left(\sum_{i=1}^{K} p_i \mathbb{E}\{\mathbf{g}_{k,i}\mathbf{g}_{k,i}^H\} + \mathbf{F}_k + \tilde{\mathbf{D}}_k \right)^{-1} \mathbb{E}\{\mathbf{g}_{k,k}\}\,, \quad (8)$$

where $\tilde{\mathbf{D}}_k \in \mathbb{N}^{L\times L}$ is a diagonal matrix in which the (l,l)-th element is 0 if the AP l serves the user k and 1 otherwise, included to ensure that the term inside the parentheses is invertible; $\mathbf{F}_k = \text{diag}\left(\mathbb{E}\{||\mathbf{D}_{k,1}\mathbf{v}_{k,1}||^2\} \dots \mathbb{E}\{||\mathbf{D}_{k,L}\mathbf{v}_{k,L}||^2\}\right) \in \mathbb{C}^{L\times L}$. This expression leads to the maximum achievable SINR:

$$\text{SINR}_k = p_k \mathbb{E}\{\mathbf{g}_{k,k}^H\} \Bigg(\sum_{i=1}^{K} p_i \mathbb{E}\{\mathbf{g}_{k,i}\mathbf{g}_{k,i}^H\} - p_k \mathbb{E}\{\mathbf{g}_{k,k}\}\mathbb{E}\{\mathbf{g}_{k,k}^H\} + \mathbf{F}_k + \tilde{\mathbf{D}}_k \Bigg)^{-1} \mathbb{E}\{\mathbf{g}_{k,k}\}\,. \quad (9)$$

The k-th user SE is determined as:

$$\text{SE}_k = \left(1 - \frac{\tau_p}{\tau_c}\right) \log_2\left(1 + \text{SINR}_k\right)\,, \quad (10)$$

where τ_c is the coherence block length. Lastly, the sum SE over all UEs is denoted as:

$$\epsilon = \sum_{k=1}^{K} \text{SE}_k\,. \quad (11)$$

D. Power Consumption Model and EE

In order to lower the energy consumed by the network, after the clustering formation process, an AP selection phase is performed, in which a subset of the available APs are used for communication and the rest are put in a *sleep* mode. In the sleep mode, an AP does not serve any user. It dissipates less power than when it is active and can be quickly turned *active* again. The AP selection scheme employed in this work occurs basically by putting all the APs not involved in the communication process, e.g., not participating in any cluster, in a *sleep* mode while all the others are in *active* mode; in which their total consumed power takes into consideration the number of *active* RF chains and the network throughput as a whole.

This work adopts the following power consumption model, based on [13]:

$$P_{Tu}(\mathbf{v}) = P_{Tu}^{\text{fix}} + \sum_{k=1}^{K} \frac{\tau_u}{\tau_c} \frac{p_k \omega_k}{\alpha_k} + B \sum_{l=1}^{L} \left(\xi_l^{\text{AP}} + \xi_l^{\text{FH}}\right)\epsilon_g\,, \quad (12)$$

in which P_{Tu} is the total power consumed in the UL transmission considering the APs, the UEs and the power consumed for using the fronthaul network to send information to the CPU; P_{Tu}^{fix} is the fixed power cost to maintain the system powered on, ω_k is the power control coefficient on the k-th UE, α_k is the amplifier efficiency for the same UE; B is the total communication bandwidth, ξ_l^{AP} and ξ_l^{FH} are the power used to send a gigabit through an AP and a fronthaul link connected to it, and lastly, ϵ_g is the sum SE in Gbps/Hz.

$$P_{Tu}^{\text{fix}} = \frac{\tau_u}{\tau_c}\Bigg[\sum_{k=1}^{K} P_k^{\text{UE,fix}} + \sum_{l=1}^{L} \bar{\omega_l}\Big(P_l^{\text{FH,fix}} + P_l^{\text{AP,fix}} + L_l P_l^{\text{AP,chain}}\Big) + \sum_{l=1}^{L} \bar{\omega_l}\Big(P_{l\text{sleep}}^{\text{FH,fix}} + P_{l\text{sleep}}^{\text{AP,fix}} + (N - L_l) P_{l\text{sleep}}^{\text{AP,chain}}\Big)\Bigg]\,, \quad (13)$$

where $P_k^{\text{UE,fix}}$ is the fixed power to maintain a UE on; $P_l^{\text{FH,fix}}$, $P_l^{\text{AP,fix}}$ and$P_l^{\text{AP,chain}}$ are the fixed power cost for maintaining the backhaul link, an AP and one RF chain of an AP on; and $P_{l\text{sleep}}^{\text{FH,fix}}$, $P_{l\text{sleep}}^{\text{AP,fix}}$ and $P_{l\text{sleep}}^{\text{AP,chain}}$ are their sleep mode counterparts, L_l is the number of active RF chains in the l-th AP, $\bar{\omega_l} = \text{sgn}(\sum_{k=1}^{k} \omega_{l,k})$, and sgn$(\cdot)$ is the sign function. Finally, the EE can be defined as:

$$\text{EE} = \frac{B\epsilon}{P_{Tu}}. \quad (14)$$

III. Problem definitions

The user-centric clustering formation is a process carried out by the CPU. In this task, the CPU chooses the set of APs that will serve a given user. This solution tends to be based on long term statistics to avoid network signaling overhead and allow stable links to UEs. When considering AP selection and channel estimation, the clustering arrangement of the network highly impacts the SE (via the number of receiving antennas and available CSI) and EE (via the number of active APs and RF chains). Considering the concept of DoF, and that the AP performs the channel estimation and part of the symbol processing, each AP can only serve N UEs [14]. Since the EE is the main focus of this work, for clustering we consider the following optimization problem:

$$\begin{aligned}
&\underset{\mathbf{\Omega} = \{\omega_{l,k}\}}{\text{maximize}} && \text{EE} && \text{(15a)}\\
&\text{subject to} && a_{l,k} \in \{0,1\}, \forall_{l,k}, && \text{(15b)}\\
&&& \text{SE}_k \geq \text{SE}_{target}, \forall_k, && \text{(15c)}\\
&&& ||\boldsymbol{\omega_l}|| \leq N. && \text{(15d)}
\end{aligned}$$

This problem, like the one in [15], is a combinatorial integer problem that, when the constraints (15c) and (15d) are taken out, has a feasible region consisting of 2^{LK} discrete points. When considering those constraints, a great number of these solutions become unfeasible, and, given the highly irregular and non-convex form of both SE and EE expressions, the problem described in (15) poses as a nondeterministic polynomial time (NP) hard problem. Since optimally solving (15) is not possible, this work will focus on low-complexity scalable

TABLE I
GRL TECHNIQUES ON WIRELESS NETWORKS.

Technique (RL + Graph-based technique)	**Description**
sparse RL + factor graphs (solved by max-plus algorithm) [16].	Apply sparse RL to solve the coordination problem in a distributed sensor network and then applies factor graphs to improve speed and achieve cooperation in multiple sub-tasks.
Covers a variety of techniques and combinations [17].	A survey which covers several RL methods, Graph-based techniques and their combination and provides application in problems like routing, scheduling and resource allocation.
REINFORCE algorithm + Graph neural networks (GNNs) [8].	Deal with power control problem using Multi-agent RL and then uses GNN to improve generalization and convergence speed.
Deep Q-Network + Graph Coloring Algorithm [18].	Assign sub-bands to user-clusters using graph coloring to mitigate the intracell interference and DRL to mitigate intercell interference.
Rainbow DRL + graph attention network (GAT) [19].	Deal with problem of Vehicle routing and scheduling with a Rainbow DRL architecture using GAT for policy processing.

heuristic solutions that yield a good result when compared with the state of the art.

Another problem considered in this work is the power allocation of the UEs. Taking advantage of the problem structure and aiming at increasing the SE of the clustered network, the following optimization is presented:

$$\underset{\mathbf{p}_k}{\text{maximize}} \quad \mathbb{E}_\pi \left[\sum_{k=1}^{K} \gamma \text{SE}_k \right] \tag{16a}$$

$$\text{subject to} \quad 0 \leq p_k \leq p, \forall_k, \tag{16b}$$

$$\text{SE}_k \geq \text{SE}_{target}, \forall_k, \tag{16c}$$

where γ is the reward discount factor and $\mathbf{p}_k = [p_1, p_2, \ldots, p_k]$. This problem is modeled using concepts of RL and is stated as to maximize the expected discounted reward over a policy π, being subject to a lower and upper power limit and with an SE target per user. The policies, for this case, are defined as the transmit power for each user. The power control problem, due to the complex nature of the SE formula, is a stochastic continuous problem with a large solution space and no trivial solutions, which makes it a good fit with DRL algorithms. Note that both problems are non-convex with large search spaces, which makes a direct comparison with convex optimization unfeasible.

IV. PROPOSED SOLUTION

Given the problems described in Section III, we propose the following solution: the use of a graph-based heuristic for the clustering problem, and a DRL algorithm based on the actor-critic framework for power control, as depicted in Fig. 1. The combination of graph theory based algorithms and DRL techniques result in the emerging field of GRL, a recent research topic that has drawn a lot of attention in the last few years [20]. Some solutions found in the literature that are implemented using GRL are described in Table I. The proposed heuristic is called the dual tree clustering (DTC) algorithm. Its modelling comes from considering the combinatorial nature of problem in (15) and addressing its constraints efficiently in a resource-aware manner. Before going into details of our proposed solution, this section also presents a quick overview of the GRL field with a focus on its applications in wireless networks.

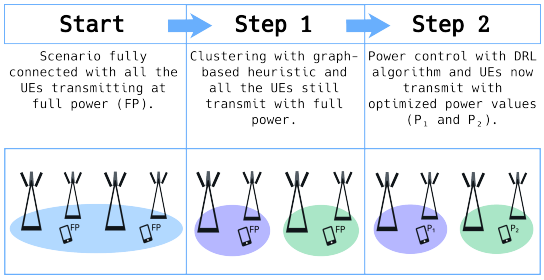

Fig. 1. Illustration of our proposed scheme.

A. GRL Overview

Given the large applicability and proven efficacy of the RL techniques in dealing with complex and non-linear problems, and more specifically the DRL methods, these algorithms have been explored to solve combinatorial problems [21] and to deal with graph mining tasks [20]. Due to the success of DRL techniques in handling those tasks, new algorithms involving graph theory, RL and neural networks were developed [17]. In general, the merge of those topics is called graph reinforcement learning. GRL algorithms have many forms, components, neural network architectures and their own ways to represent information [20], [22].

In this work, we will focus on one of the earliest developed techniques in GRL. Basically, we take a complex problem and break it into two phases: a structural/combinatorial phase and a second phase that can be modelled as a Markov decision process (MDP). The combinatorial phase is usually solved by a heuristic or by a classical combinatorial algorithm and the second phase is solved by a DRL algorithm. This approach works by using the first phase as a way of limiting the state and the action space of the DRL algorithm, making it easier to train and, in most cases, leading to better results. Some examples of this technique can be seen in [16], [18], [19].

B. Dual-Tree Clustering

The DTC algorithm is the solution we propose for the combinatorial/structural phase that solves the clustering problem. It is a graph-inspired combinatorial heuristic that forms clusters based on the available RF chains in each AP and also take into consideration the pilots of each user. A graph is a data structure formed by two disjoint sets (V, E), in which V represents an object and E a connection between two objects; in this context, a tree is a graph that is connected (no object is isolated) and has no cycles in its structure [23].

The DTC is based on the iteration of two trees formed by the available resources on the network, the UEs and the pilot information. This algorithm is performed by the CPU, which is the root of both trees. In the first tree, also known as the UE tree, its branches represent the pilot sequence and their leaves the UEs associated to each pilot. In the second tree, called the resources tree, its branches represent the APs available to serve one or more users, and their leaves represent the available RF chains at the AP. The logic behind DTC is:

1) Each branch of the UEs' tree has a copy of the associated resources;
2) For each UE on a branch, the c APs with the best gain are assigned to the user and those APs are cut off from this pilot branch.
3) For each time that an AP is chosen to serve a UE, one of its RF chains are cut off from the original resources tree. If all RF chains are pruned from an AP, this AP is no longer available to serve new UEs.
4) After iterating through the whole UE tree, the remaining RF chains on the resources tree are considered in sleep mode and if the AP has all of its RF chain available, the whole AP is put in sleep mode.

Observe that the output of the DTC is a candidate solution for solving problem (15) and that its clusters always obey the restriction (15d).

C. Deep Learning-based Power Control

RL is a field of study concerned with the problem of one or more learning agents interacting with an environment in order to achieve a goal. The second phase of our two-phase solution can be mathematically modelled as an MDP [24]. To properly understand the chosen power control algorithm, a few definitions are in order. The MDP framework models an interface between the agent and its environment, in which the agent receives a state from the environment and, based on this state, chooses an action. Depending on both state and action, the agent receives a reward and then senses a new state to repeat the process as much as needed. A policy π is a mapping, either stochastic or deterministic, that, given a state and an action, returns the probability of that action being taken in that state. An action-value function for policy π, $q_\pi(s, a)$, is a function that returns the expected reward value of taking an action in a given state under policy π. In order to solve the learning problem, the agent must find a policy that maximizes the action-value function of the problem. In the case of the power control problem presented in (16), the state and action space for the agent to learn are both large and continuous, which make them unfeasible for classical RL algorithms to solve. A well known solution to address large state and action space is to use deep learning techniques into RL methods to be able to provide a good generalization of the whole space using only a subset of it [25]. This combination of techniques is often called DRL. These DRL methods also deal efficiently with continuous spaces [26].

For the DRL algorithm the choice was the deep deterministic policy gradient (DDPG) algorithm due to its proven capacity to solve sum-rate maximization problems [27], [28] similar to problem in (16). The DDPG algorithm is an actor-critic based off-policy learning method, proposed by [29]. Initially, two sets of neural networks are created. The first set is the actor $\mu(s|\theta^\mu)$ and the critic $Q(s, a|\theta^Q)$ with weights θ^μ and θ^Q. The second set is called target network, formed by the target actor $\mu'(s|\theta^{\mu'})$ and the target critic $Q'(s, a|\theta^{Q'})$, with $\theta^\mu = \theta^{\mu'}$ and $\theta^Q = \theta^{Q'}$. The second set is used to calculate the target values and their weights will be slowly updated by interacting with the first set. After the creation of the networks, a replay buffer *R* is created. Then, for each episode[1], the algorithm initializes a random process $\mathcal{N}$ for action exploration, receives an initial state s_0 and enters in the main training loop. In the main training loop, for each step t in the episode, an action is selected according to the current actor policy and exploration noise, $a_t = \mu(s_t|\theta^\mu) + \mathcal{N}_t$, then the action is performed providing a reward r_t and a new state s_{t+1}. The transition (s_t, a_t, r_t, s_{t+1}) is stored in *R*. After the current policy is evaluated, the algorithm will update their networks' weights. First, a mini-batch of *N* transitions is sampled from *R*; then, the critic is updated with the loss calculated as:

$$L = \frac{1}{N}\sum_i (r_i + \gamma Q'\left(s_{i+1}, \mu'\left(s_{i+1}|\theta^\mu\right)'|\theta^Q\right)' - Q\left(s_i, a_i|\theta^Q\right))^2. \tag{17}$$

The actor policy is updated by:

$$\nabla_{\theta^\mu} Q = \frac{1}{N}\sum_i \nabla_a Q\left(s, a|\theta^Q\right)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu}\mu\left(s|\theta^\mu\right)|_{s_i}, \tag{18}$$

and finally, the target networks are soft updated[2] using:

$$\begin{aligned} \theta^{Q'} &= \tau\theta^Q + (1-\tau)\theta^{Q'} \\ \theta^{\mu'} &= \tau\theta^\mu + (1-\tau)\theta^{\mu'} \end{aligned}, \tag{19}$$

finishing then the training loop and the episode in the last t.

D. State Space, Action Space and Reward Function

The specifics of our power control problem can be addressed in the modelling of the state and action, in which, the algorithm in Section IV-C will act on and in the reward function used to evaluate its actions. For our scenario, described in Section II, the full state s_t in a step, similarly as in [28], is defined as a vector of the user's SE, $s_t = [\mathrm{SE}_1, \mathrm{SE}_2, \ldots, \mathrm{SE}_k]$. The action is simply the vector with the power of each user, $a_t = [\rho_1, \rho_2, \ldots, \rho_k]$. Lastly, the reward function is given by $\Gamma_t = wr_t^1 + (1-w)r_t^2$, where w is a weight factor between r_t^1 and r_t^2, with

$$r_t^1 = \sum_{k=1}^{K} I_k^t \mathrm{SE}_k \tag{20}$$

and

$$r_t^2 = -\sum_{k=1}^{K}(1 - I_k^t)(\mathrm{SE}_{target} - \mathrm{SE}k), \tag{21}$$

[1]In this context, a set of iterations with a random initialization.

[2]Also called Polyak averaging, in which τ is a weight parameter.

where I_k^t is equal to 1 if ($\text{SE}_k \geq \text{SE}_{target}$) and 0 otherwise. This reward function has two main goals, the first one, in (20), is to benefit the UEs that have an SE above the target and the second, in (21), is to punish the UEs that do not achieve the target SE.

V. Simulation Results

In this section, the performance of the D-MIMO network described in Section II using the proposed solutions is evaluated in terms of SE and EE. To benchmark our methods, an ideal fully connected architecture (FCA) D-MIMO network is used as reference for our KPIs, along with the dynamic cooperation clustering (DCC) algorithm [4].

TABLE II
Simulation parameters.

Parameter	Value
Coverage Area	500x500 m
Number of APs	$L = 25$
Number of antennas per AP	$N = 4$
Number of users	$K = 10$
Carrier frequency	f_c = 2 GHz
Communication bandwidth	W = 20 MHz
UE and AP heights	1.5 m, 11.5 m
Max UL power per UE	p = 100 mW
Antenna spacing	$d = (1/2)\,\lambda$
Number of pilots	τ_p = 10
Coherence block length	τ_c = 200
UEs σ_{sh} (LOS, NLOS)	4, 8.2
Power amplifier efficiency α_k^{UE}	0.3
Power per AP ξ_l^{AP}, UE ξ_l^{UE}, FH ξ_l^{FH}	0.25, 0.25, 0.25 W/Gbps
$P_l^{\text{AP,fix}}$, $P_{l\,\text{sleep}}^{\text{AP,fix}}$	6, 0.8 W
$P_l^{\text{AP,chain}}$, $P_{l\,\text{sleep}}^{\text{AP,chain}}$	0.15, 0.02 W
$P_l^{\text{FH,fix}}$, $P_{l\,\text{sleep}}^{\text{FH,fix}}$	5, 0.5 W
$P_l^{\text{UE,fix}}$	0.75 W

The FCA scheme is a D-MIMO network, in which all APs serve all the UEs at the same time. Note that when an AP serves all the UEs, its combiner scheme converges to the optimal MMSE combiner [12]; thus providing improved interference mitigation.

As a well-known algorithm and practical solution for user-centric clustering, the DCC framework [4] presents the following logic: first, it sets the AP with the best channel gain as the master; then, it assigns other APs that serve only users with orthogonal pilots and that are within a channel gain *threshold* range of the master AP, given by [master gain, master gain + threshold].

This algorithm tends to yield higher SE than other heuristics based on channel gain or distance due to each AP serving users with orthogonal pilots, and thereby reducing the interference within each cluster. To cope with our scenario of MIMO DoF, the DCC algorithm will also limit the maximum number of connections to an AP to its number of RF chains.

The simulated network is configured according to the parameters in Table II, while the hyper parameters of the DRL algorithm are set according to to Table III. The neural networks used by the DDPG consist of two hidden layers of 64 neurons each, with layer normalization applied at each layer. The APs

TABLE III
DDPG hyper parameters.

Hyper Parameters	Value
Soft update	0.005
Learning rate for the networks	1e-3
Buffer size	1e6
Reward discount factor	0.5
Reward weight	0.5
Batch size	50
SE target	6 bps/Hz

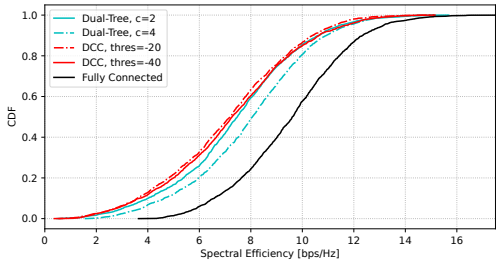

Fig. 2. CDF of SE per UE for various cluster configurations.

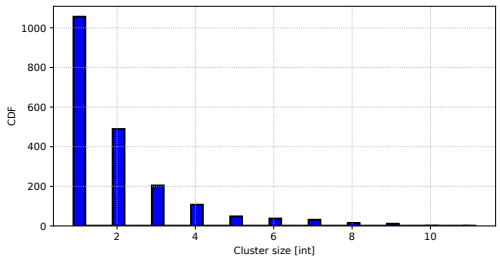

Fig. 3. Histogram of the cluster size for DCC with -20 dB threshold.

are equally spaced along a two dimensional (2D) grid, while the UEs are uniformly distributed over the same grid. In order to compare the EE of the clusters in a fair manner, when DCC is applied, all the resources not selected to integrate a cluster, both AP and its RF chains, will be considered in sleep mode.

A. Clustering Without Power Control

We start by comparing the two clustering solutions in terms of SE and EE. This comparison will not make use of the DDPG algorithm for power control; instead, all UEs transmit at full power. Starting from this scenario helps us to understand the behavior of each algorithm and allows us to isolate the contribution of each step of the proposed solution.

Fig. 2 shows the SE per UE of the network under two parameterizations for each algorithm, and for the FCA scheme. Recall that when using the DCC, the cluster size is a consequence of the threshold range set by the master AP. As indicated by Fig. 2, increasing this threshold increases the number of connections made by a UE, it does not necessarily improves the SE. As for the DTC, the cluster size is fixed as an input parameter, and it is observable that doubling the cluster size shifts most of the SE CDF to the right, which indicates an improvement of the SE for most of the UEs.

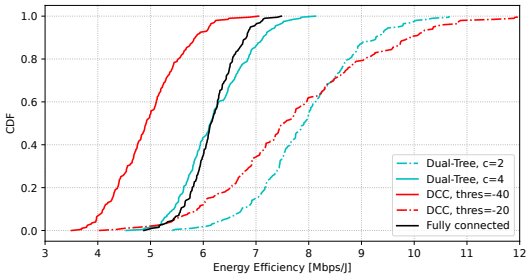

Fig. 4. CDF of EE network for various cluster configurations.

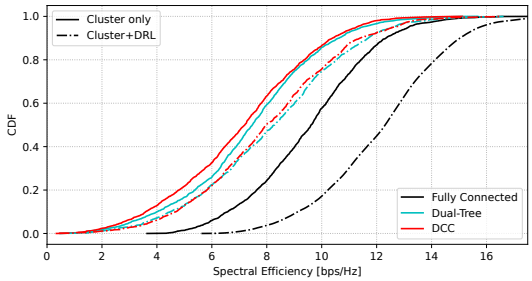

Fig. 5. CDF of SE per UE for various cluster configurations.

This contrast occurs due to how the DCC forms clusters: As the APs are assigned depending on the channel gain, when a smaller threshold is chosen the UEs with higher channel gain quickly grow their clusters, and the UEs with poor channel conditions experience only a small (if any) improvement in their connections. if any. In order to visualize this effect, Fig. 3 shows a histogram of the cluster size of the UEs when the threshold of -20 dB is used.

Examining the EE CDF in Fig. 4, it is clear that for both algorithms, lowest cluster sizes yield highest efficiency. This occurs because when the APs and RF chains are not utilized, they are put in sleep mode, and the smaller the cluster size, the less RF chains are needed by the network. Moreover, when comparing the two solutions in terms of EE, we can notice an interesting effect for the case of cluster size 2 and threshold value -20 dB. Until the 60th percentile the DTC is the most energy efficient scheme, but after that, the DCC becomes more efficient. Looking at Fig. 3, we can see that roughly half of the UEs are only allowed to have a single AP connections. Thus, the cases that outperform the DTC are more likely to have four or more APs in their cluster size.

B. Clustering and Power Control

Having gained some understanding the dynamics of each algorithm, in this subsection we study how they interact with the learning-based power control. Similar to the full power case, Fig. 5 shows the SE per UE for the DCC, the DTC and FCA, with and without the DDPG as power control. First, we can observe that all algorithms with power control present a significant improvement on their respective SE curves, with the FCA being the most prominent. This happens due to the convergence of its combiner to the optimal form, leading the interference of the network to a lower level compared with the clustering scenarios in [3]. This enables the DDPG algorithm to deal with a simpler scenario to learn and efficiently increase its SE.

Regarding the clustering algorithms, both of them improve in a similar manner at SE. This happens because both of achieving the SE target at around their 20th percentile, which suggests that for most cases the solution by the DRL technique is superior due to producing a positive reward and thereby advancing the gradient at a slower pace.

Fig. 6, presents a remarkable dynamic. Similarly to Fig. 4, the DTC under the DDPG power control performs better than

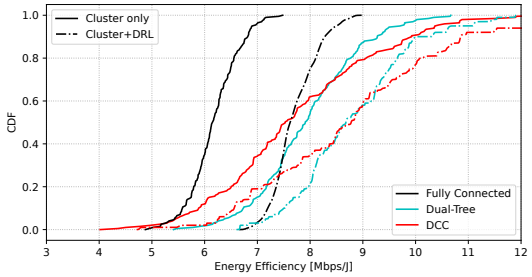

Fig. 6. CDF of EE network for various cluster configurations.

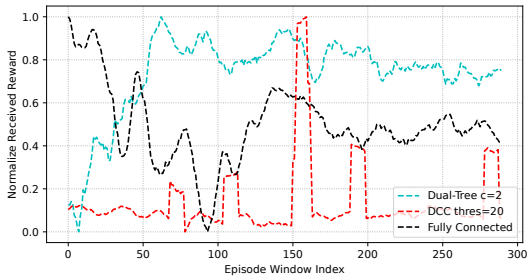

Fig. 7. The normalized *moving averaged* reward over the episodes.

the DCC under same conditions until the 60th percentile of their CDFs. Even so, at low percentiles, the DTC not only has a larger advantage to DCC in comparison to the full power case. It also starts its CDF together with the FCA case, and outperforms it at every percentile. These results make the DTC and DDPG pair the preferred choice when the EE is the important KPI.

Finally, as briefly commented in Section IV, the combination of DTC and DDPG can be classified as a GRL approach, and their synergy goes beyond providing a suitable solution to the problem in (16). Observing Fig. 7, we can study the normalized reward of DDPG over its episodes. A moving average of 20 samples was used to filter the statistical variations of the reward signal. At a first glance, we can observe that all three approaches learn quite differently. The DTC and the FCA scheme seem to converge to a small range of values along the episodes while the DCC presents a more chaotic behavior. This happens due to the variability of the scenario: both the DTC and the FCA schemes have a fixed number of links, while DCC changes them based on the channel. This

makes DCC a more complex scenario to learn, yielding a highly varying reward. Furthermore, comparing the FCA and the DTC schemes, we note that the FCA scheme forces all endpoints to communicate, turning the power control to a challenging problem, since there are more UEs transmitting to each AP. The DTC scheme, on the other hand, has a fixed number of links and a smaller number of connections between the endpoints, and produces a good response to Problem (15). Thus, it inherently contributes to solving Problem (16), which helps the learning process of DDPG. Moreover, as Fig. 7 indicates, the DTC scheme converges around the 60th episode, which is one fifth of the total training time.

VI. Conclusions

In this paper, resource allocation problems in D-MIMO networks were studied. Specifically, the AP clustering and the uplink power allocation problems were addressed and evaluated in terms of both EE and SE. To address both problems, a resource-aware clustering algorithm was proposed and used along with a DRL based power control. Practical networking aspects were considered such as channel estimation and MIMO DoFs.

Our results indicate that the proposed solution outperforms the state-of-the-art DCC algorithm when no power control is used, both in terms of the SE and, for the low and middle percentile, in the EE. When the learning-based power control solution is used, the DTC and DDPG pair provides superior EE in most of the cases, while showing better convergence properties compared to the DCC and FCA solutions.

The perspectives for this work include coupling the DTC with other possible DRL solutions like soft actor critic (SAC) and proximal policy optimization (PPO), modeling a multi-agent form of DDPG for this problem and integrating the DTC scheme even tighter with the DDPG by allowing DRL to modify the structural part of the problem in an alternating manner and thereby providing a more robust GRL approach.

References

[1] D. G. S. Pivoto, T. T. Rezende, M. S. P. Facina, *et al.*, "A detailed relevance analysis of enabling technologies for 6G architectures," *IEEE Access*, vol. 11, pp. 89 644–89 684, 2023.

[2] M. Matthaiou, O. Yurduseven, H. Q. Ngo, D. Morales-Jimenez, S. L. Cotton, and V. F. Fusco, "The road to 6G: Ten physical layer challenges for communications engineers," *IEEE Communications Magazine*, vol. 59, no. 1, pp. 64–69, 2021.

[3] Ö. T. Demir, E. Björnson, and L. Sanguinetti, "Foundations of user-centric cell-free massive MIMO," *Foundations and Trends in Signal Processing*, vol. 14, no. 3-4, pp. 162–472, 2021.

[4] E. Björnson and L. Sanguinetti, "Scalable cell-free massive MIMO systems," *IEEE Transactions on Communications*, vol. 68, no. 7, pp. 4247–4261, Jul. 2020.

[5] J. Cerviño, J. A. Bazerque, M. Calvo-Fullana, and A. Ribeiro, "Multi-task reinforcement learning in reproducing kernel Hilbert spaces via cross-learning," *IEEE Transactions on Signal Processing*, vol. 69, pp. 5947–5962, 2021.

[6] J. Tang, X. Chen, Z. Li, *et al.*, "Log-regularized dictionary-learning-based reinforcement learning algorithm for GNSS positioning correction," *IEEE Internet of Things Journal*, vol. 11, no. 8, pp. 15 022–15 037, 2024.

[7] A. Elgabli, H. Khan, M. Krouka, and M. Bennis, "Reinforcement learning based scheduling algorithm for optimizing age of information in ultra reliable low latency networks," in *Proc. IEEE Symposium on Computers and Communications (ISCC)*, 2019, pp. 1–6.

[8] L. M. Amorosa, M. Skocaj, R. Verdone, and D. Gündüz, "Multi-agent reinforcement learning for power control in wireless networks via adaptive graphs," in *Proc. IEEE International Conference on Communications*, 2024, pp. 2968–2973.

[9] M. Rahmani, M. Bashar, M. J. Dehghani, *et al.*, "Deep reinforcement learning-based sum rate fairness trade-off for cell-free mMIMO," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 5, pp. 6039–6055, 2023.

[10] E. Björnson, J. Hoydis, and L. Sanguinetti, *Massive MIMO Networks: Spectral, Energy, and Hardware Efficiency*. NOW Publishing, 2017.

[11] 3GPP, *TR 38.901: Study on channel model for frequencies from 0.5 to 100 GHz*, Release 15, report, 2019.

[12] E. Björnson and L. Sanguinetti, "Making cell-free massive MIMO competitive with MMSE processing and centralized implementation," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 77–90, Sep. 2019.

[13] J. García-Morales, G. Femenias, and F. Riera-Palou, "Energy-efficient access-point sleep-mode techniques for cell-free mmwave massive mimo networks with non-uniform spatial traffic density," *IEEE Access*, vol. 8, pp. 137 587–137 605, 2020.

[14] E. Björnson, C.-B. Chae, R. W. H. J. au2, *et al.*, *Towards 6G MIMO: Massive spatial multiplexing, dense arrays, and interplay between electromagnetics and processing*, 2024. arXiv: 2401.02844 [cs.IT].

[15] N. Ghiasi, S. Mashhadi, S. Farahmand, S. M. Razavizadeh, and I. Lee, "Energy efficient AP selection for cell-free massive MIMO systems: Deep reinforcement learning approach," *IEEE Transactions on Green Communications and Networking*, vol. 7, no. 1, pp. 29–41, 2023.

[16] Z. Zhang and D. Zhao, "Clique-based cooperative multiagent reinforcement learning using factor graphs," *IEEE/CAA Journal of Automatica Sinica*, vol. 1, no. 3, pp. 248–256, 2014.

[17] N. Vesselinova, R. Steinert, D. F. Perez-Ramirez, and M. Boman, "Learning combinatorial optimization on graphs: A survey with applications to networking," *IEEE Access*, vol. 8, pp. 120 388–120 416, 2020.

[18] C. Ge, S. Xia, Q. Chen, and F. Adachi, "Learning based on graph: A joint interference coordination for cluster-wise distributed MU-MIMO," *IEEE Communications Letters*, vol. 27, no. 3, pp. 871–875, 2023.

[19] Q. Xing, Y. Xu, and Z. Chen, "A bilevel graph reinforcement learning method for electric vehicle fleet charging guidance," *IEEE Transactions on Smart Grid*, vol. 14, no. 4, pp. 3309–3312, 2023.

[20] M. Nie, D. Chen, and D. Wang, "Reinforcement learning on graphs: A survey," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 7, no. 4, pp. 1065–1082, 2023.

[21] H. Dai, E. B. Khalil, Y. Zhang, B. Dilkina, and L. Song, "Learning combinatorial optimization algorithms over graphs," in *Proc. International Conference on Neural Information Processing Systems*, ser. NIPS'17, 2017, pp. 6351–6361, ISBN: 9781510860964.

[22] Y. Liu, J. Xia, S. Zhou, *et al.*, *A survey of deep graph clustering: Taxonomy, challenge, application, and open resource*, 2023. arXiv: 2211.12875 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2211.12875.

[23] R. Diestel, *Graph Theory* (Electronic library of mathematics). Springer, 2005, ISBN: 9783540261834.

[24] L. Luo, J. Zhang, S. Chen, X. Zhang, B. Ai, and D. W. K. Ng, "Downlink power control for cell-free massive MIMO with deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 6, pp. 6772–6777, 2022.

[25] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*, 2nd. MIT Press, 2018.

[26] D. Bertsekas, *Reinforcement Learning and Optimal Control* (Athena Scientific optimization and computation series). Athena Scientific, 2019.

[27] X. Guo, Z. Li, P. Liu, *et al.*, "A novel user selection massive MIMO scheduling algorithm via real time DDPG," in *Proc. IEEE Global Communications Conference*, 2020, pp. 1–6.

[28] X. Zhang, M. Kaneko, V. An Le, and Y. Ji, "Deep reinforcement learning-based uplink power control in cell-free massive MIMO," in *Proc. IEEE 20th Consumer Communications & Networking Conference (CCNC)*, 2023, pp. 567–572.

[29] T. P. Lillicrap, J. J. Hunt, A. Pritzel, *et al.*, *Continuous control with deep reinforcement learning*, 2019. arXiv: 1509.02971 [cs.LG]. [Online]. Available: https://arxiv.org/abs/1509.02971.