Multi-agent Deep Reinforcement Learning for Multi-Cell Interference Mitigation

Madan Dahal and Mojtaba Vaezi
Department of Electrical and Computer Engineering
Villanova University, Villanova, PA 19085, USA
Email: {mdahal, mvaezi}@villanova.edu

2023 57th Annual Conference on Information Sciences and Systems (CISS) | 978-1-6654-5181-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/CISS56502.2023.10089622

***Abstract*—Multi-cell interference management techniques typically require sharing channel state information (CSI) among all cells involved, making the algorithms ineffective for practical uses. To overcome this shortcoming, an interference mitigation technique that does not require explicit CSI or coordination among neighboring cells is developed in this paper. The algorithm leverages distributed deep reinforcement learning to this end and delivers a faster and more spectrally-efficient solution than state-of-the-art centralized techniques. An important aspect of our proposed solution is that it scales very well with the number of cells in the network. The effectiveness of the proposed algorithm is verified by simulation over millimeter-wave networks with two to seven cells. Interestingly, the penalty for not sharing CSI decreases as the number of cells increases. In particular, for a 7-cell network, the proposed algorithm without sharing CSI achieves 92% of the spectral efficiency obtained by sharing CSI.**

I. INTRODUCTION

Spectral efficiency improvement is a crucial objective for modern *multi-cell* networks that employ tight or global frequency reuse. This tight frequency reuse creates *inter-cell interference*, also known as co-channel interference, which hinders high throughput and spectral efficiency in current network [1]–[3]. Inter-cell interference management has been studied extensively in the literature [1]–[4]. *Interference alignment* [1], [2] is a recent breakthrough in interference management that is much more efficient than time-division multiple access in theory. However, it has not entered wireless standards as it requires *global* channel state information (CSI), i.e., the CSI of the interfering cells should be known by the serving cell, which is not practical [3]. Coordinated multi-point (CoMP) [5] is another well-known solution to the inter-cell interference problem in which data and CSI are shared among neighboring cellular base stations (BSs) to coordinate their transmissions in the downlink and jointly process the received signals in the uplink. CoMP requires a high-speed backhaul network for enabling the exchange of information between the BSs [6].

As a cutting-edge tool, deep learning holds significant potential in offering solutions to complex problems. Particularly, *deep reinforcement learning (DRL)*, a sub-field of deep learning, has been successfully used to solve several communication problems in the physical layer, such as beamforming, power allocation, and interference cancellation in various settings [7], [8]. In reinforcement learning [9] an agent learns to interact with an environment (the multi-cell network in this work) by taking a sequence of actions to maximize a cumulative reward, e.g., the spectral efficiency or any other desired quantity. Centralized DRL where the training is done at a central position and all the BSs shares their information, performs well as it requires information from all cells. This performance, though, comes with two unfavorable trade-offs: first, the algorithm must be optimized across all cooperating BSs, which results in extremely high computational complexity; second, it needs a significant exchange of network CSI between cooperating BSs [10]. Consequently, these methods do not scale well for deployment in real-world wireless networks as they require ensuring this strict level of cooperation and information sharing across the network.

Distributed DRL (multi-agent DRL), where training is done at individual BS, has recently received a lot of attention since it can reduce the need for information exchange between the agents. The problem may be approached as a non-cooperative game in which the BSs attempt to develop an acceptable power distribution plan utilizing best-response dynamics without having access to the entire network CSI [11]. In [12]–[14] multi-agent DRL approach is used to solve the problem of maximizing the spectral efficiency but all methods require coordination between cooperating BSs.

In this paper, we propose a distributed DRL algorithm to be employed by all BSs to mitigate inter-cell interference in a multi-cell network with limited information sharing between the neighboring cells and by only relying on the power measurement at the desired cell and user coordinates. The algorithm works for an arbitrary number of cells. The goal is to maximize the spectral efficiency of the network, manifested by network sum-capacity, through optimizing transmit power and beamforming vectors. Simulation results show that the proposed algorithm can be almost as effective as the centralized technique. More importantly, the spectral efficiency scales with the number of cells, and its value using the distributed technique gets closer to that with centralized technique as the number of cells increases.

The remainder of the paper is organized as follows. The system model and problem formulation of a multi-cell wireless network are described in Section II. In Section III, distributed DRL-based interference management is designed. Section IV discusses the training setup and simulation results. The paper is concluded in Section V.

978-1-6654-5181-9/23/$31.00 ©2023 IEEE

II. System Model and Problem Formulation

A. System Model

We consider a downlink cellular network consisting of L cells, each of which consists of a multi-antenna BS and a single-antenna user equipment (UE). The users share the same frequency range in all cells, resulting in inter-cell interference. We assume that all BSs are equipped with a uniform linear array having M antenna elements. Hence, the received signal at a given UE at cell $\ell \in \{1, \ldots, L\}$ can be written as

$$y_\ell = \mathbf{h}_{\ell,\ell}^H \mathbf{w}_\ell x_\ell + \sum_{j \neq \ell} \mathbf{h}_{\ell,j}^H \mathbf{w}_j x_j + n_\ell, \tag{1}$$

where $\mathbf{h}_{\ell,\ell}$ and $\mathbf{h}_{\ell,j} \in \mathbb{C}^{M\times 1}$ are the downlink channel vectors between the user and ℓth and jth BSs respectively, $\mathbf{w}_\ell$ and $\mathbf{w}_j \in \mathbb{C}^{M\times 1}$ are analog beamforming vector, $n_\ell \in \mathcal{N}(0, \sigma^2)$ is the noise at the UE sampled from a complex Normal distribution with zero-mean and variance σ^2, and $x_j \in \mathbb{C}$ is the transmitted symbol from jth BS, where the transmitted signal must satisfy the average power constraints $\mathbb{E}[|x_j|^2] = P_j$, where P_j is the transmit power of BS j. Each UE decodes only the message from its associated BS, and thus, signals from the other BSs are viewed as interference. The signal-to-interference-plus-noise-ratio (SINR) and the sum achievable rate of UE located at cell ℓ are given by

$$\gamma_\ell = \frac{P_\ell |\mathbf{h}_{\ell,\ell}^H \mathbf{w}_\ell|^2}{\sigma^2 + \sum_{j\neq\ell} P_j |\mathbf{h}_{\ell,j}^H \mathbf{w}_j|^2}, \tag{2}$$

$$C_\ell = \log_2(1 + \gamma_\ell), \tag{3}$$

in which P_ℓ and P_j are the transmit power of the serving BS and the jth interfering BS.

We adopt the geometric channel model for the channels. If we assume there are N_p paths between the UE and each BS and each path from the jth BS to the user has a complex gain of $\alpha_{\ell,j,i}$, $(i \in \{1, ..., N_p\})$ and angle of departure (AoD) is $\phi_{\ell,j,i}$, then $\mathbf{h}_{\ell,j}$ can be written as

$$\mathbf{h}_{\ell,j} = \frac{\sqrt{M}}{\rho_{\ell,j}} \sum_{i=1}^{N_P} \alpha_{\ell,j,i} \mathbf{a}^*(\phi_{\ell,j,i}), \tag{4}$$

where $\mathbf{a}^*(\phi_{\ell,j}^i)$ is the array response vector associated with the angle of departure, and $\rho_{\ell,j}$, represent the path loss between the jth BS and user at cell ℓ. For a two-dimensional channel model, the transmit antenna array is described by its *array steering vector*. The steering vector $\mathbf{a}(\theta)$ depends on the angular directions of the departing plane wave, and for an M-element *uniform linear array* is given by

$$\mathbf{a}(\theta) = \left[1, e^{-j2\pi \frac{d}{\lambda}\cos(\theta)}, \ldots, e^{-j2\pi \frac{d}{\lambda}(M-1)\cos(\theta)}\right]^T, \tag{5}$$

where $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ is a physical angle of departure and d and λ, respectively, are the antenna spacing and the wavelength of operation. Due to the hardware constraints on large-scale multiple-antenna systems, the BSs normally use pre-defined

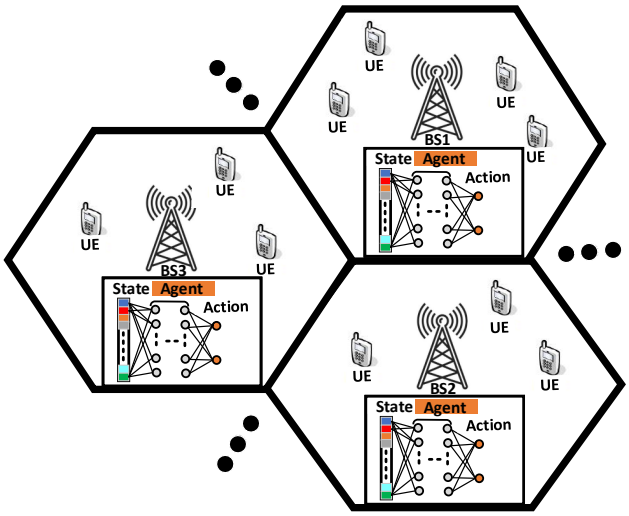


Fig. 1: Distributed DRL where the BSs simultaneously and independently determine their power and beamforming vectors.

beamforming codebooks (such as DFT codebooks [15]) that scan all possible directions for data transmission. Let $\mathcal{W}$ represents the *beamforming codebook* adopted by the BSs. For r-bit quantized phase shifters

$$\mathbf{w} = \frac{1}{\sqrt{M}} \left[e^{j\theta_1}, \ldots, e^{j\theta_M}\right]^T, \tag{6}$$

where the phase shift θ_m, $1 \leq m \leq M$, is selected from a finite set $\mathbf{\Theta}$ with $N = 2^r$ possible discrete values uniformly drawn from $[0, \pi]$. That is, $\mathbf{\Theta} = [0, \frac{\pi}{N}, \frac{2\pi}{N}, \ldots, \frac{(N-1)\pi}{N}]$. To simplify the design, one may use constant modulus constraints on the entries of the beamforming vector [15]. In such a case, $\mathbf{w} = \mathbf{a}(\theta)$ where θ is an angle in $\mathbf{\Theta}$. We use $N = M$ in this paper.

B. Problem Formulation

Sum achievable rate, or simply sum-rate, is a common measure of spectral efficiency in cellular networks. Considering this, our goal is to find the arguments that maximize the network sum-rate we can solve

$$\begin{aligned} \max_{P_\ell, \mathbf{w}_\ell} \quad & \sum_{\ell=1}^{L} C_\ell \\ \text{s.t.} \quad & P_\ell \in \mathcal{P}, \quad \mathbf{w}_\ell \in \mathcal{W}, \end{aligned} \tag{7}$$

in which $\mathcal{P}$ is the set of possible transmit powers and $\mathcal{W}$ is *beamforming codebook* from which $\mathbf{w}_\ell$ is selected.

The above optimization problem is nonconvex and is hard to solve due to the constant modulus constraints. Most of the recent methods will need global CSI knowledge, i.e., $\mathbf{h}_{\ell,j}$ for all ℓ and j. This is not, however, practical since CSI overhead will consume a big portion of the bandwidth when L is large and reduce the spectral efficiency.

In the following, we proposed a distributed DRL method that requires knowing only the serving cell channel $\mathbf{h}_{\ell,\ell}$ or P_ℓ, the power of the serving cell. That is, it does not need to know the global CSI ($\mathbf{h}_{\ell,j}$ for $j \neq \ell$) or there is no need for coordination between the BSs, which is a big advance.

III. Distributed DRL-based Interference Management

A. Motivation for distributed DRL-based Solution

A centralized interference management requires information exchange, which grows exponentially with the number of cells. Besides, the state and action spaces' expansion sizes cause convergence to happen very slowly. This problem, often known as the "curse of dimensionality," makes the centralized DRL technique unsuitable for large wireless networks since the corresponding training periods grow impractically long. From a computational standpoint, multi-agent techniques are appealing since the network only computes the actions for one agent, avoiding the issue of the state and action spaces necessarily growing in size with a centralized approach. Training a single deployment strategy across all BSs appears to enable network scaling without necessarily requiring longer training times. As a result, rather than using single-agent techniques, multi-agent approaches are typically used to handle large-scale reinforcement learning issues. We structure the learning process such that these agents learn under a shared incentive of the network's spectral efficiency. We consider each BS in the network to be a separate agent as shown in Fig. 1 with no access to network information.

B. System Operation

In this subsection, we discuss how to acquire the power measurements to evaluate the objective function of (7). The power of the signal received from the intended transmitter as well as the power of interference caused by other transmitters must be measured specifically. UE$_\ell$ can coordinate with the serving BS$_\ell$ to determine when it is transmitting, and this coordination might be used to provide the necessary power measurements. To evaluate γ_ℓ in (2), the UE first measures the interference plus noise level

$$I + N = \sum_{j \neq \ell} P_j |\mathbf{h}_{\ell,j}^H \mathbf{w}_j|^2 + \sigma^2, \tag{8}$$

when the serving BS is not transmitting. To get the interference plus noise level, we can use zero power CSI reference signal (CSI-RS) in 5G New Radio [16]. Then, when the serving BS starts transmission, the UE measure the signal plus interference plus noise level

$$S + I + N = P_\ell |\mathbf{h}_{\ell,\ell}^H \mathbf{w}_\ell|^2 + \sum_{j \neq \ell} P_j |\mathbf{h}_{\ell,j}^H \mathbf{w}_j|^2 + \sigma^2, \tag{9}$$

The receive power of the UE can hence be determined by subtracting two power measurements, and the SINR can be approximately obtained from (2), by using these measured powers. This measured SINR is sent back to the serving BS.

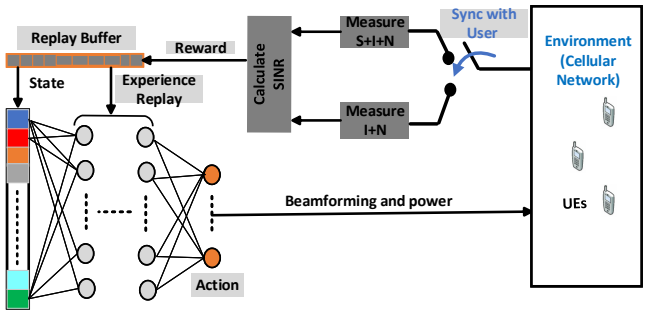


Fig. 2: An illustration of the operational flow of the proposed distributed DRL solution for interference management, where the signal power is estimated by the UE.

C. Distributed deep Q-network (DQN) Algorithm

Our goal is to develop an algorithm for interference management without coordination between cooperating BSs. In the following, we describe the main elements of a DRL system and specify them in our proposed algorithm. Besides the *agent* (each BS), and the environment there are three other basic concepts in DRL: state, action, and reward.

- The *state* $\mathbf{s}_{\ell,t} \in \mathcal{S}$ represents the features extracted by agent ℓ from the environment that describes the current situation. It is what agent ℓ observes at time step t
  - $x_\ell[t]$: the x coordinates of the UE of cell ℓ.
  - $y_\ell[t]$: the y coordinates of the UE of cell ℓ.
  - $P_\ell[t]$: the transmit power of the BS of cell ℓ.
  - $\mathbf{w}_\ell[t]$: the index of beamforming vector code book of the BS of cell ℓ, all at time t.
- The action $\boldsymbol{a}_{\ell,t} \in \mathcal{A}$ is the move taken by an agent ℓ within the environment at time step t. The action $\boldsymbol{a}_{\ell,t}$ will advance the state $\mathbf{s}_{\ell,t}$ to $\mathbf{s}_{\ell,t+1}$. In our problem, actions are to change the power and beamforming vector of BS. To be more specific, the action $\boldsymbol{a}_{\ell,t}$ taken by each agent is a binary vector of length 2 ($\boldsymbol{a}_{\ell,t} \in \mathbb{R}^2$) in the following form

$$\boldsymbol{a}_{\ell,t} = \{ \underbrace{a_1}_{\text{power control}}, \underbrace{a_2}_{\text{beamforming}} \}, \tag{10}$$

  and element of the action is either '0'or '1'. More specifically, for any ℓ, we have
  - $a_1 = 0$: decrease the transmit power of BS ℓ by 1dB.
  - $a_1 = 1$: increase the transmit power of BS ℓ by 1dB.
  - $a_2 = 0$: step down the beamforming codebook index of BS ℓ.
  - $a_2 = 1$: step up the beamforming codebook index of BS ℓ.

  The change in the transmit power of the BS ℓ at time step t due to agent ℓ is given by

$$P_\ell[t] := \min\left(P_{BS}^{\max}, P_\ell[t-1] + PC_\ell[t]\right), \tag{11}$$

in which $P_{BS}^{\max}$ is the maximum allowable power of the BS and is set the same for all BSs and $PC_j[t]$ is the power control command at BSℓ which is $+1$dB or -1dB depending on the action related to that command. For beamforming, we start with a random beamforming vector in the codebook (random index) and move to the previous or next beamforming vector in the codebook. It is seen that, by taking action $\boldsymbol{a}_{\ell,t}$, the agent ℓ is changing the beamforming vectors as well as transmit power for serving BS.

- The *reward* is an incentive mechanism that tells each agent ℓ the consequence of an action. The agent's final objective is to maximize the total cumulative reward. The setting of the reward is based on the objective function (7) and is given as

$$r_{t+1} = \sum_{\ell=1}^{L} C_\ell, \tag{12}$$

where C_ℓ is the achievable rate received by the UE at cell ℓ when action $\boldsymbol{a}_{\ell,t}$ is taken by the agent ℓ.

- The *episode* (E) is a time frame within which all the agent interacts with the environment. Each episode has T time steps.

These elements interact with each other and are governed by the goal of maximizing the future discounted reward for every action taken by the agent ℓ that changes the environment.

It is expected that $Q_\ell^\pi(\mathbf{s}_{\ell,t}, \boldsymbol{a}_{\ell,t})$ will converge to the optimal state-action value functions as $t \to \infty$ as it is updated at each time step [9]. However, it could be difficult to achieve. Therefore, a neural network-based function approximator aligned with [17] is used in this paper. We define $\boldsymbol{\Theta}_{\ell,t} \in \mathbb{R}^{u \times v}$ to represent the weights of neural networks at time steps t for each agent ℓ, where u is the number of hidden nodes and v is the number of layers. We define $\boldsymbol{\theta}_{\ell,t} \triangleq \text{vec}(\boldsymbol{\Theta}_{\ell,t}) \in \mathbb{R}^{uv}$ and use this as a function approximator. We choose the sigmoid activation function to compute the hidden layer values and has the following form

$$f(x) = \frac{1}{1 + e^{-x}}, \tag{13}$$

For every agent ℓ, DQN with the initial weight $\boldsymbol{\theta}_{\ell,t}$ is adjusted at every time step t to reduce the error via the mean-squared error loss function $L_{\ell,t}(\boldsymbol{\theta}_{\ell,t})$

$$\min_{\boldsymbol{\theta}_{\ell,t}} L_{\ell,t}(\boldsymbol{\theta}_{\ell,t}) \triangleq \mathbb{E}_{\mathbf{s}_{\ell,t}, \boldsymbol{a}_{\ell,t}} \left[(y_{\ell,t} - Q_\ell^\pi(\mathbf{s}_{\ell,t}, \boldsymbol{a}_{\ell,t}; \boldsymbol{\theta}_{\ell,t}))^2 \right], \tag{14}$$

in which

$$y_{\ell,t} := \mathbb{E}_{\mathbf{s}_{\ell,t}, \boldsymbol{a}_{\ell,t}} [r_{\ell,t+1} + \alpha \max_{\boldsymbol{a}_{\ell,t+1}} Q_\ell^\pi(\mathbf{s}_{\ell,t+1}, \boldsymbol{a}_{\ell,t+1}; \boldsymbol{\theta}_{\ell,t-1} | \mathbf{s}_{\ell,t}, \boldsymbol{a}_{\ell,t})]$$

is the estimated function value at time step t given state $\mathbf{s}_{\ell,t}$ and have an action $\boldsymbol{a}_{\ell,t}$. We will try to reduce this loss in every iteration for agent ℓ. The objective of the distributed DQN algorithm is to find a solution that optimizes the state-action value function for each agent ℓ.

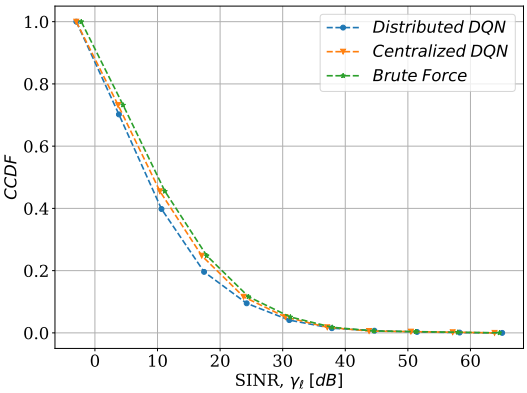


Fig. 3: Coverage plots for different methods for $M = 4$.

D. Training and Evaluation

The algorithm has two phases: *training and testing* phases. During the training phase, the agents are trained offline before it becomes active in the network. In this phase, the weights of the neural network are optimized using the stochastic gradient descent algorithm on the batches of the dataset taken from the replay buffer R each agent ℓ. Having a replay buffer allows each agent ℓ to use a more diverse mini-batch for performing updates during the training process. It also allows each to take larger mini-batch sizes B. Further, by sampling at random from the replay buffer, the updates to the neural network will have low variance since the data entering the optimization method look independent and identically distributed.

We define the state-action value function estimated by the DQN in agent ℓ, $Q_\ell^\pi(\mathbf{s}_{\ell,t}, \boldsymbol{a}_{\ell,t})$ as

$$Q_\ell^\pi(\mathbf{s}_{\ell,t}, \boldsymbol{a}_{\ell,t}) = \mathbb{E}\left[r_{\ell,t+1} + \alpha Q_\ell^\pi(\mathbf{s}_{\ell,t+1}, \boldsymbol{a}_{\ell,t+1}) | \mathbf{s}_{\ell,t}, \boldsymbol{a}_{\ell,t} \right], \tag{15}$$

This is also known as the Bellman equation, in which α is a discount factor whose range is $[0, 1]$, $\mathbf{s}_{\ell,t+1}$ and $\boldsymbol{a}_{\ell,t+1}$ are the new state and action, respectively, $r_{\ell,t+1}$ is the reward achieved when moving to the new state

At each round of the training process, each agent ℓ strikes a balance between exploring the environment and exploiting the knowledge of best action accumulated through such exploration. We adopt an ϵ-greedy policy [9], where $\epsilon := \max(\epsilon\delta, \epsilon_{\min})$ is the exploration rate, δ is the exploration decay rate, and $\epsilon_{\min}$ is the minimum exploration rate. The exploration rate decays in every episode until it reaches $\epsilon_{\min}$. We exploit if $p > \epsilon$ where p is randomly drawn from Unif$(0, 1)$; we explore otherwise. Based on the selected action $\boldsymbol{a}_{\ell,t+1}$, each agent ℓ computes its reward function according to (12). A summary of the training phase is given in Algorithm 1.

After the convergence of the training, we use the optimized weights for the evaluation (testing) of the DRL algorithm to assess the quality of the learned policy [18]. The evaluation can be performed during training or after that. In this phase, agent ℓ chooses its actions greedily (no exploration) for each

Algorithm 1 Training phase of the proposed DQN algorithm

Randomly initialize network $Q_\ell^\pi(\mathbf{s}_{\ell,t}, \boldsymbol{a}_{\ell,t})$ with weight $\boldsymbol{\theta}_{\ell,t}$
Initialize time, states, actions, minibatch size B and R
for episode 1 to E do
  for t=1 to T do
    for ℓ=1 to L do
    Receive observation state $\boldsymbol{s}_{\ell,t}$ for BS ℓ
      Step 1: Action selection
      if $p > \epsilon$ then
        $arg\max_{\mathbf{a}_{\ell,t+1}} Q_\ell^\pi(\mathbf{s}_{\ell,t}, \boldsymbol{a}_{\ell,t+1}; \boldsymbol{\theta}_{\ell,t})$
      else
        randomly chosen from $\mathcal{A}$
      Step 2: Reward calculation
      Calculate rewards based on (12)
      Store $(\boldsymbol{s}_{\ell,t}, \boldsymbol{a}_{\ell,t}, r_{\ell,t})$ in $R_{\ell,t}$
    end for
    Calculate the network sum-rate, $r_t = \sum_{\ell=1}^{L} r_{\ell,t}$
    Step 3: DQN update
    for ℓ=1 to L do
      Observe the next state $\boldsymbol{s}_{\ell,t+1}$
      Store $(r_t, \boldsymbol{s}_{\ell,t+1})$ in $R_{\ell,t}$
      Sample a random minibatch of size b
      $b = \min(B, T(E-1)+t)$, from $R_{\ell,t}$
      Set $y_{\ell,t} = \left[r_{\ell,t+1} + \alpha \max_{\boldsymbol{a}_{\ell,t+1}} \mathrm{Q}_\ell^\pi(\mathbf{s}_{\ell,t+1}, \boldsymbol{a}_{\ell,t+1}; \boldsymbol{\theta}_{\ell,t})\right]$
      Perform SGD on $(y_{\ell,t} - Q_\ell^\pi(\mathbf{s}_{\ell,t}, \boldsymbol{a}_{\ell,t}; \boldsymbol{\theta}_{\ell,t}))^2$ to find $\boldsymbol{\theta}_{\ell,t}^*$
      Update $\boldsymbol{\theta}_{\ell,t} = \boldsymbol{\theta}_{\ell,t}^*$ in the DQN
    end for
  end for
end for

state. Actions for agent ℓ are a sequence of power control and beamforming selection to solve (7).

IV. Training Setup and Simulation Results

The training setup, simulation details, performance measures and numerical results are demonstrated in this section.

A. Network Setup and Performance Measures

We consider an L-cell network with hexagonal geometry each with a cell radius of $112m$ and inter-site distance $D = 225m$. The operation frequency is 28 GHz and the propagation model is COST231 [19]. UEs are uniformly distributed within each cell and move at a speed of 2 km/h. In (6) to (2), where needed $d = \frac{\lambda}{2}$, $N_p = 4$ with probability 0.8 and $N_p = 1$ (line of sight channel) with probability 0.2, and radio frame duration $T = 10ms$. The initial position of the UEs, the initial power of the BSs, and initial the beamforming vectors are selected randomly. In order to plot the effective SINR, we set the minimum SINR as $\gamma_{\min} = -3$ dB which represents the minimum SINR for any user in the cellular network. If the SINR falls below the minimum value, the episode aborts which means the call is dropped. The training parameters of the distributed DQN are listed in Table I.

TABLE I: The distributed DQN parameters for agent ℓ.

Parameters	Value
α	0.995
Initial, ϵ	1.000
ϵ_{min}	0.1
Learning rate	0.01
u	24
v	2
δ	0.995
DQN batch size, B	32

Spectral efficiency (measured by achievable sum-rate) is the main performance evaluation measure. We evaluate the average network sum-rate by

$$R_{\text{sum}} = \frac{1}{E} \sum_{e=1}^{E} \sum_{\ell=1}^{L} C_\ell, \tag{16}$$

where E is the total number of episodes. Another performance measure is overall network coverage, evaluated by the *complementary cumulative distribution function (CCDF)* of the SINR (γ_ℓ for all cells).

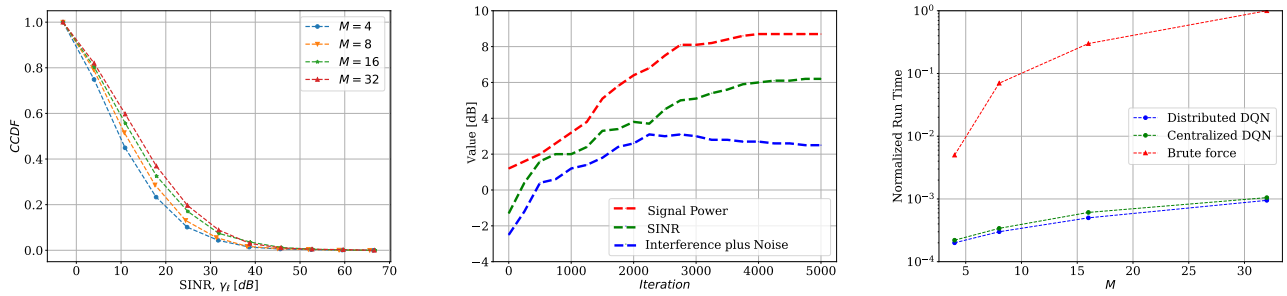
B. Results

In Fig. 3, the CCDFs of γ_ℓ for different algorithms are compared with the brute force method for $M = 4$. The proposed distributed DQN results in a solution which is close to that of the centralized DQN. For example, using the proposed distributed DQN 18% of the time the UEs have SINR > 20 dB, while this number is about 20% for the centralized DQN. This is a good result as in the distributed approach there is no the between cooperating BSs. In Fig. 4(a), we see that as M increases, the probability of having higher SINRs increases which is related to the fact that the array gain increases with M. This figure shows that the algorithm scales with the BS antennas. In Fig. 4(b), we see that the signal power increases as the number of iteration increases.

The normalized run time of different algorithms are compared in Fig. 4(c). We can see that distributed DQN is faster than other algorithms as there is no information sharing between the cells. Finally, Fig. 5 shows that the sum-rate of the proposed distributed DQN algorithm scales with the number of cells. The average per cell achievable rates are is 3.15, 3.30, 3.61 and 3.86 bps/Hz for $L = 2, 3, 5$ and 7, respectively. This graph shows that with universal frequency reuse and without coordination, the network capacity can be increased almost linearly with L.

V. Conclusions

We have proposed a distributed DQN-based interference management in multi-cell mmWave networks. The goal is to maximize the network sum-rate without sharing CSI information between the cells, rather only by relying on the certain power measurements at the desired cell. The BSs select their beamforming vector and power command from finite sets. The input features of each agent are its



(a) $L = 2$ and $M = 4$ (b) $L = 3$ and $M = 4$ (c) $L = 3$

Fig. 4: (a) Coverage plot of the proposed distributed DQN for different number of antennas M, (b) SINR improves by number of iterations, and (c) normalized run times for the different methods as a function of number of M.

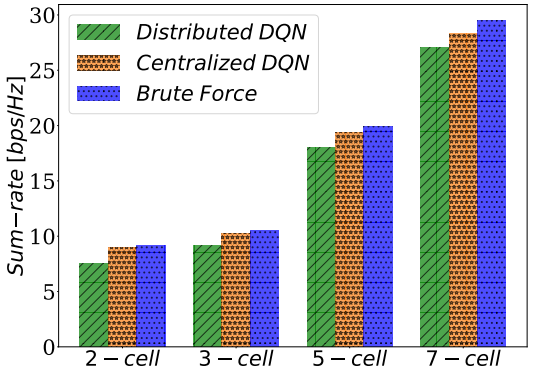


Fig. 5: Network sum-rate for the proposed distributed DQN, centralized DQN, and brute force method, for different numbers of cells.

user's coordinates, BS power, and beamforming vector. The output has a sequence of interference management along with power control and beamforming that optimize the objective function. Our proposed algorithm almost reaches the spectral efficiency obtained by centralized DQN researching among all possible beamforming vectors and BS powers. Also, the performance of the algorithm improves as the number of cells increases. Furthermore, the complexity of the method is much lower, which makes it promising to be implemented in practice

REFERENCES

[1] M. A. Maddah-Ali, A. S. Motahari, and A. K. Khandani, "Communication over MIMO X channels: Interference alignment, decomposition, and performance analysis," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3457–3470, 2008.

[2] S. A. Jafar, *Interference alignment: A new look at signal dimensions in a communication network*. Now Publishers Inc, 2011.

[3] O. El Ayach, S. W. Peters, and R. W. Heath, "The practical challenges of interference alignment," *IEEE Wireless Commun.*, vol. 20, no. 1, pp. 35–42, 2013.

[4] S. Sun, Q. Gao, Y. Peng, Y. Wang, and L. Song, "Interference management through CoMP in 3GPP LTE-advanced networks," *IEEE Wireless Commun.*, vol. 20, pp. 59–66, 2013.

[5] D. Lee, H. Seo, B. Clerckx, E. Hardouin, D. Mazzarese, S. Nagata, and K. Sayana, "Coordinated multipoint transmission and reception in LTE-advanced: deployment scenarios and operational challenges," *IEEE Commun. Mag*, vol. 50, no. 2, pp. 148–155, 2012.

[6] W. Shin, M. Vaezi, B. Lee, D. J. Love, J. Lee, and H. V. Poor, "Non-orthogonal multiple access in multi-cell networks: Theory, performance, and practical challenges," *IEEE Commun. Magazine*, vol. 55, no. 10, pp. 176–183, 2017.

[7] F. B. Mismar, B. L. Evans, and A. Alkhateeb, "Deep reinforcement learning for 5G networks: Joint beamforming, power control, and interference coordination," *IEEE Trans. Commun*, vol. 68, no. 3, pp. 1581–1592, 2020.

[8] M. Dahal and M. Vaezi, "Deep reinforcement learning for interference management in millimeter-wave networks," in *Proc. Asilomar Conf. Signals Syst. Comput*, 2022.

[9] R. S. Sutton and A. G. Barto, *Reinforcement learning: An Introduction*. MIT Press, 2018.

[10] A. A. Khan, R. Adve, and W. Yu, "Optimizing multicell scheduling and beamforming via fractional programming and hungarian algorithm," in *IEEE GC Wkshps*, pp. 1–6, 2018.

[11] P. de Kerret, S. Lasaulce, D. Gesbert, and U. Salim, "Best-response team power control for the interference channel with local CSI," in *Proc. IEEE Int. Conf. Commun. (ICC)*, pp. 4132–4136, 2015.

[12] Y. S. Nasir and D. Guo, "Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks," *IEEE J. Sel. Areas Commun*, vol. 37, no. 10, pp. 2239–2250, 2019.

[13] L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. on Syst., Man, and Cybern, Part C*, vol. 38, no. 2, pp. 156–172, 2008.

[14] N. Zhao, Z. Liu, and Y. Cheng, "Multi-agent deep reinforcement learning for trajectory design and power allocation in multi-UAV networks," *IEEE Access*, vol. 8, pp. 139670–139679, 2020.

[15] S. Hur, T. Kim, D. J. Love, J. V. Krogmeier, T. A. Thomas, and A. Ghosh, "Millimeter wave beamforming for wireless backhaul and access in small cell networks," *IEEE Trans. Commun*, vol. 61, no. 10, pp. 4391–4403, 2013.

[16] 3GPP, "Study on new radio access technology: Physical layer aspects," Tech. Rep. 38.802, 2017, version 14.2.2.

[17] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv:1312.5602*, 2013.

[18] D. L. Poole and A. K. Mackworth, *Artificial Intelligence: Foundations of Computational Agents*. Cambridge University Press, 2010.

[19] A. I. Sulyman, A. Alwarafy, G. R. MacCartney, T. S. Rappaport, and A. Alsanie, "Directional radio propagation path loss models for millimeter-wave wireless networks in the 28-, 60-, and 73-GHz bands," *IEEE Trans. Wireless Commun.*, vol. 15, no. 10, pp. 6939–6947, 2016.