Deep Reinforcement Learning for 5G Networks: Joint Beamforming, Power Control, and Interference Coordination

Faris B. Mismar, *Senior Member, IEEE,* Brian L. Evans, *Fellow, IEEE,* and Ahmed Alkhateeb, *Member, IEEE*

Abstract

The fifth generation of wireless communications (5G) promises massive increases in traffic volume and data rates, as well as improved reliability in voice calls. Jointly optimizing beamforming, power control, and interference coordination in a 5G wireless network to enhance the communication performance to end users poses a significant challenge. In this paper, we formulate the joint design of beamforming, power control, and interference coordination to maximize the signal to interference plus noise ratio (SINR) and solve the non-convex problem using deep reinforcement learning. By using the greedy nature of deep Q-learning to estimate future benefits of actions, we propose an algorithm for voice bearers in sub-6 GHz bands and data bearers in millimeter wave (mmWave) frequency bands. The algorithm exploits reported SINR from connected users, the transmit powers of the base stations, and the coordinates of the connected users to improve the performance measured by coverage and sum-rate capacity. The proposed algorithm does not require the channel state information and removes the need for channel estimation. Simulation results show that our algorithm outperforms the link adaptation industry standards for sub-6 GHz voice bearers and approaches the optimal limits for mmWave data bearers for small antenna sizes in realistic cellular environments.

Index Terms

F. B. Mismar and B. L. Evans are with the Wireless Networking and Communications Group, Dept. of Electrical and Comp. Eng., The University of Texas at Austin, Austin, TX, 78712, USA. e-mail: faris.mismar@utexas.edu, bevans@ece.utexas.edu. A. Alkhtaeeb is with the School of Electrical, Computer and Energy Engineering at Arizona State University, Tempe, AZ 85287, USA. email: alkhateeb@asu.edu.

A preliminary version of this work was presented at the 2018 Asilomar Conf. on Signals, Systems and Computers [1].

Deep reinforcement learning, power control, interference coordination, beam coordination, self-organizing network

I. Introduction

The massive growth in traffic volume and data rate continues to evolve with the introduction of *fifth generation of wireless communications* (5G). Also evolving is enhanced voice call quality with better reliability and improved codecs. Future wireless networks are therefore expected to meet this massive demand for both the data rates and the enhanced voice quality. In an attempt to learn the implied characteristics of inter-cellular interference and inter-beam interference, we propose an online learning based algorithm based on a *reinforcement learning* (RL) framework. We use this framework to derive a near-optimal policy to maximize the end-user SINR. The importance of reinforcement learning in power control has been demonstrated in [1]–[3]. Power control in voice bearers makes them more robust against wireless impairments, such as fading. It also enhances the usability of the network and increases the cellular capacity.

A. Prior Work

Performing power control and beamforming in both uplink and downlink was studied in [4]–[7]. Power control and beamforming were jointly solved in [7] using optimization, but with no regards to scattering or shadowing, which are critical phenomena in *millimeter wave* (mmWave) propagation.

The industry standards adopted the method of *almost blank subframe* (ABS) to resolve the co-channel inter-cell interference problem in LTE where two base stations interfere with one another [8]. While ABS works well in fixed beam antenna patterns, the dynamic nature of beamforming reduces the usefulness of ABS [9].

An online learning algorithm for link adaptation in *multiple-input multiple-output* (MIMO) bearers was studied in [2]. The algorithm computational complexity was comparable to existing online learning approaches, but with minimal spatial overhead. Further, the algorithm adapted to the change of channel distribution quickly.

Interference avoidance in a heterogeneous network was studied in [3]. A Q-learning framework for the coexistence of both macro and femto BSs was proposed. The feasibility of decentralized self-organization of these BSs was established where the femtocells inteference towards the macro BSs was mitigated. The use of Q-learning was also proposed in [1]. The framework

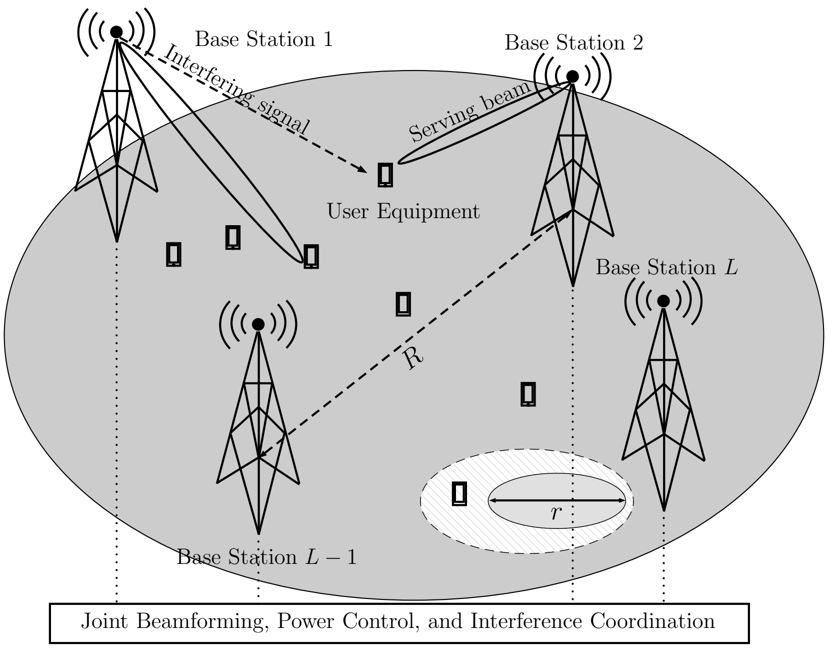

Fig. 1. Performing joint beamforming and power control on the signal from the serving base station while coordinating interference from the other BS. There are L base stations with an inter-site distance of R and a cell radius of r.

focused on packetized voice power control in a multi-cell indoors environment. It exploits the use of semi-persistent scheduling in order to establish a virtual sense of a dedicated channel. This channel enabled the power control of the downlink to ensure enhanced voice clarity compared to industry standards.

Joint power control in massive MIMO was introduced in [4]. This approach led to a reduced overhead due to a limited exchange of channel state information between the BSs participating in the joint power control. The joint power control scheme led to enhanced performance measured by the SINR. In the uplink direction, power control in beamforming was studied in [5]. An optimization problem was formulated to maximize the achievable sum rate of the two users while ensuring a minimal rate constraint for each user. Using reinforcement learning to solve the problem for the uplink is computationally expensive and can cause a faster depletion of the *user equipment* (UE) battery. We on the other hand focus on the downlink and on interference cancellation alongside power control.

Over the last two years, the use of deep learning in wireless communications has been studied in [6], [10]–[12]. The specific use of deep reinforcement learning to perform power control for mmWave was studied in [6]. This approach was proposed as an alternative to beamforming in improving the *non-line of sight* (NLOS) transmission performance. The power allocation problem to maximize the sum-rate of UEs under the constraints of transmission power and quality targets was solved using deep reinforcement learning. In this solution, a convolutional neural network was used to estimate the Q-function of the deep reinforcement learning problem. In [10], a policy that maximizes the successful transmissions in a dynamic correlated multichannel access environment was obtained using deep Q-learning. The use of deep convolutional neural networks was proposed in [11] to enhance the automatic recognition of modulation in cognitive radios at low SINRs. A deep learning classifier that was able to reliably jam transmissions through power control was studied in [12].

B. Contribution

In this paper, we introduce a different approach to power control, where we not only control the transmit power of the *base station* (BS), but also coordinate the transmit powers of the interfering base stations. This approach allows us to control the SINR through controlling the interference instead of the usual control of transmit power levels. As a result of this apparent conflict, a race condition emerges, where the serving BS of a given user is an interfering BS of another user. Therefore, while power control requests an increase in power for a given BS and a given user, the interference coordination may simultaneously request a decrease in power for that same BS. In our previous work [1], we focused on power control on the downlink for voice users by changing the serving BS transmit power. However, for SINR target computations, we only derived an upper bound on interference and used it in our computations.

We propose a *deep reinforcement learning* (DRL) approach to resolve the mentioned race condition. We further perform joint beamforming with the power control and interference coordination where applicable. We do so through simultaneously coordinating the transmit powers of both the serving and interfering BSs. This joint activity can take place at a central location, or at one of the base stations, as shown in Fig. 1. We adopt the grid of beams beamforming approach and perform downlink *power control and interference coordination* (PCIC) without the connected handsets sending these commands to the serving or interfering BS. Rather, the BSs autonomously compute their PCIC commands based on RL. The PCIC commands are issued on behalf of a single handset at any given discrete time step.

C. *Paper Organization and Notation*

The remainder of this paper is organized as follows. In Section II, we describe the network model, the system model, and the channel model in detail. Section III outlines the problem formulation and motivates the importance of using reinforcement learning in such problems. In Section IV, we discuss deep reinforcement learning and its usage in solving our problem. In Section V we propose deep RL-based algorithms to perform coordinated PCIC with for voice bearers in sub-6 GHz bands. Section VI extends the idea to joint beamforming and PCIC but for mmWave data bearers. In Section VII, we show the proposed performance measurement quantities to benchmark our algorithms. Section VIII shows the results of our proposed algorithms based on the selected performance measures and a discussion about these results. We conclude the paper in Section IX.

Notation: Boldface lower and upper case symbols represent column vectors and matrices, respectively. Calligraphic letters are for sets. The Hermitian transpose is $(\cdot)^{\mathsf{H}}$. The cardinality of a set is $|\cdot|$. The expectation operator is $\mathbb{E}[\cdot]$. The indicator function $\mathbb{1}_{(\cdot)}$ is equal to one if the condition in the parentheses is true and zero if false. $[\cdot]_{i,j}$ is the element in row i and column j of a matrix. Finally, an M-by-N matrix whose elements are real or complex numbers is $\mathbb{R}^{M\times N}$ or $\mathbb{C}^{M\times N}$.

II. Network, System, and Channel Models

In this section, we describe the adopted network, system, and channel models.

A. *Network Model*

We consider an *orthogonal frequency division multiplexing* (OFDM) multi-access downlink cellular network of L *base stations* (BS). This network is comprised of a serving BS and at least one interfering BS. We adopt a downlink scenario, where a BS is transmitting to one *user equipment* (UE). The BSs have an intersite distance of R and the UEs are randomly scattered in their service area. The cell radius is $r > R/2$ to allow overlapping of coverage. Voice bearers run on sub-6 GHz frequency bands while the data bearers use mmWave frequency band. We employ analog beamforming for the data bearers to compensate for the high propagation loss due to the higher center frequency.

B. System Model

Considering the network model in Section II-A, and adopting a multi-antenna setup where each BS employs a *uniform linear array* (ULA) of M antennas and the UEs have single antennas, the received signal at the UE from the ℓ-th BS can be written as

$$y_\ell = \mathbf{h}_{\ell,\ell}^{\mathsf{H}} \mathbf{f}_\ell x_\ell + \sum_{b \neq \ell} \mathbf{h}_{\ell,b}^{\mathsf{H}} \mathbf{f}_b x_b + n_\ell \tag{1}$$

where $x_\ell, x_b \in \mathbb{C}$ are the transmitted signals from the ℓ-th and b-th BSs, and they satisfy the power constraint $\mathbb{E}[|x_\ell|^2] = P_{\text{TX},\ell}$ (similarly for b). The $M \times 1$ vectors $\mathbf{f}_\ell, \mathbf{f}_b \in \mathbb{C}^{M \times 1}$ denote the adopted downlink beamforming vectors at the ℓ-th and b-th BSs, while the $M \times 1$ vectors $\mathbf{h}_{\ell,\ell}, \mathbf{h}_{\ell,b} \in \mathbb{C}^{M \times 1}$ are the channel vectors connecting the user at the ℓ-th BS with the ℓ-th and b-th BSs, respectively. Finally, $n_\ell \sim \mathcal{N}_{\mathbb{C}}(0, \sigma^2)$ is the received noise at the user sampled from a complex Normal distribution with zero-mean and variance σ^2.

The first term in (1) represents the desired received signal, while the second term represents the interference received at the user due to the transmission from the other BSs.

Beamforming vectors: Given the hardware constraints on the mmWave transceivers, we assume that the BSs use analog-only beamforming vectors, where the beamforming weights of every beamforming vector $\mathbf{f}_\ell, \ell = 1, 2, ..., L$ are implemented using constant-modulus phase shifters, i.e., $[\mathbf{f}_\ell]_m = e^{j\theta_m}$. Further, we assume that every beamforming vector is selected from a beamsteering-based beamforming codebook $\mathcal{F}$ of cardinality $|\mathcal{F}| = N_{\text{CB}}$, with the n-th element in this codebook defined as

$$\begin{aligned} \mathbf{f}_n &:= \mathbf{a}(\theta_n) \\ &= \frac{1}{\sqrt{M}} \left[1, e^{jkd\cos(\theta_n)}, ..., e^{jkd(M-1)\cos(\theta_n)}\right]^\top, \end{aligned} \tag{2}$$

where d and k denote the antenna spacing and the wave-number, while θ_n represents the steering angle. Finally, $\mathbf{a}(\theta_n)$ is the array steering vector in the direction of θ_n. The value of θ_n is obtained by dividing the the antenna angular space between 0 and π radians by the number of antennas M.

Power control and interference coordination: Every BS ℓ is assumed to have a transmit power $P_{\text{TX},\ell} \in \mathcal{P}$, where $\mathcal{P}$ is the set of candidate transmit powers. We define the set of the transmit powers as the power offset above (or below) the BS transmit power. Our choice of the transmit power set $\mathcal{P}$ is provided in Section VIII-A. This choice of $\mathcal{P}$ follows [13].

Power control and interference coordination take place over a semi-dedicated channel. For voice, this is facilitated through the semi-persistent scheduling, which creates a virtual sense of a dedicated channel as we mentioned earlier. For data bearers, the use of beamforming provides a dedicated beam for a given UE, through which power control and interference coordination takes place.

C. Channel Model

In this paper, we adopt a narrow-band geometric channel model, which is widely considered for analyzing and designing mmWave systems [14]–[16]. With this geometric model, the downlink channel from a BS b to the user in BS ℓ can be written as

$$\mathbf{h}_{\ell,b} = \frac{\sqrt{M}}{\rho_{\ell,b}} \sum_{p=1}^{N^p_{\ell,b}} \alpha^p_{\ell,b} \mathbf{a}\left(\theta^p_{\ell,b}\right)^{\mathsf{H}}, \tag{3}$$

where $\alpha^p_{\ell,b}$ and $\theta^p_{\ell,b}$ are the complex path gain and angle of departure (AoD) of the p-th path, and $\mathbf{a}(\theta^p_{\ell,b})$ is the array response vector associated with the AoD, $\theta^p_{\ell,b}$. Note that $N^p_{\ell,b}$ which denotes the number of channel paths is normally a small number in mmWave channels compared to sub-6 GHz channels [17], [18], which captures the sparsity of the channels in the angular domain. Finally, $\rho_{\ell,b}$, represents the path-loss between BS b and the user served in the area of BS ℓ. Note that the channel model in (3) accounts of both the LOS and NLOS cases. For the LOS case, we assume that $N^p_{\ell,b} = 1$.

We define $P_{\text{UE}}[t]$ as the received downlink power as measured by the UE over a set of *physical resource blocks* (PRBs) at a given time t as

$$P^{\ell,b}_{\text{UE}}[t] = P_{\text{TX},b}[t] \left|\mathbf{h}^{\mathsf{H}}_{\ell,b}[t]\mathbf{f}_b[t]\right|^2 \tag{4}$$

where $P_{\text{TX},b}$ is the PRB transmit power from BS b. Next, we compute the received SINR for the UE served in BS ℓ at *transmit time interval* (TTI) t as follows:

$$\gamma^{\ell}_{\text{DL}}[t] = \frac{P_{\text{TX},\ell}[t]|\mathbf{h}^{\mathsf{H}}_{\ell,\ell}[t]\mathbf{f}_\ell[t]|^2}{\sigma^2 + \sum_{b\neq\ell} P_{\text{TX},b}[t]|\mathbf{h}^{\mathsf{H}}_{\ell,b}[t]\mathbf{f}_b[t]|^2}. \tag{5}$$

This is the received SINR that we will optimize in our paper in Sections V and VI.

III. Problem Formulation

Our objective is to jointly optimize the beamforming vectors and the transmit power at the L BSs to maximize the achievable sum rate of the users. We formulate the joint beamforming, power control, and interference coordination optimization problem as

$$\begin{aligned} \underset{\substack{P_{\text{TX},\ell}[t],\ \forall \ell \\ \mathbf{f}_\ell[t],\ \forall \ell}}{\text{maximize}} \quad & \prod_{\ell} \gamma_{\text{DL}}^{\ell}[t] \\ \text{subject to} \quad & P_{\text{TX},\ell}[t] \in \mathcal{P}, \qquad \forall \ell, \\ & \mathbf{f}_\ell[t] \in \mathcal{F}, \qquad \forall \ell, \\ & \gamma_{\text{DL}}^{\ell}[t] \geq \gamma_{\text{DL,target}}. \end{aligned} \tag{6}$$

where $\gamma_{\text{DL,target}}$ denotes the target SNR of the downlink transmission. This problem is a non-convex optimization problem due to the non-convexity of the constraints. Solving this problem using classical (non-machine learning techniques) would normally require an exhaustive search over the large space to find candidate solutions. In this paper, we propose to solve this challenge by leveraging deep learning tools that can avoid the exhaustive search while achieve high SINRs. In particular, adopting deep learning (and more specifically deep reinforcement learning) is motivated by the following points:

1) We do not require the knowledge of the channels in order to find the optimal beamforming vector.
2) We minimize the involvement of the UE in sending feedback to the BS. In particular, the UE sends back its received SINR along with its coordinates, while the agent handles the power control and interference coordination commands to the involved BSs.
3) The optimal coordination of joint beamforming, power control, and interference coordination when multiple BSs are involved is prohibitively expensive. The use of RL offers near-optimal distributed coordination of the control overhead of multiple BS in linear time in L.
4) Having explicit PCIC commands sent by the UE to the serving and interfering BSs requires a modification to the current industry standards [13].

Next, we provide a brief overview on deep reinforcement learning in Section IV before delving into the proposed algorithm in Sections V and VI.

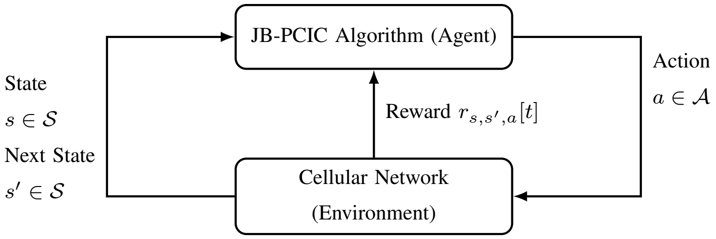

Fig. 2. The agent-environment interaction in reinforcement learning.

IV. A Primer on Deep Reinforcement Learning

In this section, we describe *deep reinforcement learning* (DRL), which is a special type of reinforcement learning that was introduced in [19]. Reinforcement learning is a machine learning technique that enables an *agent* to discover what action it should take to maximize its expected future *reward* in an interactive *environment*. The interaction between the agent and the environment is shown in Fig. 2.

Since we adopt reinforcement learning algorithms in this paper, as described in Section V and VI, the next description focuses on reinforcement learning. In particular, DRL exploits the ability of deep neural networks to learn better representations than handcrafted features and act as a universal approximator of functions.

Reinforcement learning elements: Reinforcement learning has several elements [20]. These elements interact together, and are as follows:

- *Observations*: Observations are continuous measures of the properties of the environment and are written as a p-ary vector $\mathbf{O} \in \mathbb{R}^p$, where p is the number of properties observed.
- *States*: The state $s_t \in \mathcal{S}$ is the discretization of the observations at time step t. Often, states are also used to mean observations.
- *Actions*: An action $a_t \in \mathcal{A}$ is one of the valid choices that the agent can make at time step t. The action changes the state of the environment from the current state s to the target state s'.
- *Policy*: A policy $\pi(\cdot)$ is a mapping between the state of the environment and the action to be taken by the agent. We define our stochastic policy $\pi(a \,|\, s) : \mathcal{S} \times \mathcal{A} \to [0, 1]$.
- *Rewards*: The reward signal $r_{s,s',a}[t]$ is obtained after the agent takes an action a when it is in state s at time step t and moves to the next state s'.
- *State-action value function*: The state-action value function under a given policy π is denoted $Q_\pi(s, a)$. It is the expected discounted reward when starting in state s and selecting an action

a under the policy π.

These elements work together and their relationship is governed by the objective to maximize the future discounted reward for every action chosen by the agent, which causes the environment to transition to a new state. The policy dictates the relationship between the agent and the state. The value of the expected discounted reward is learned through the training phase.

If $Q_\pi(s, a)$ is updated every time step, then it is expected to converge to the optimal state-action value function $Q_\pi^*(s, a)$ as $t \to \infty$ [20]. However, this may not be easily achieved. Therefore, we use a function approximator instead aligned with [19]. We define a neural network with its weights at time step t as $\mathbf{\Theta}_t \in \mathbb{R}^{u \times v}$ as in Fig. 3. Also, if we define $\boldsymbol{\theta}_t := \operatorname{vec}(\mathbf{\Theta}_t) \in \mathbb{R}^{uv}$, we thus build a function approximator $Q_\pi(s, a; \boldsymbol{\theta}_t) \approx Q_\pi^*(s, a)$. This function approximator is neural network based and is known as the *Deep Q-Network* (DQN) [19]. The neural network is trained through adjusting $\boldsymbol{\theta}$ at every time step t to reduce the mean-squared error loss $L_t(\boldsymbol{\theta}_t)$:

$$\underset{\boldsymbol{\theta}_t}{\text{minimize}} \qquad L_t(\boldsymbol{\theta}_t) := \mathbb{E}_{s,a}\left[(y_t - Q_\pi(s, a; \boldsymbol{\theta}_t))^2\right] \tag{7}$$

where $y_t := \mathbb{E}_{s'}[r_{s,s',a} + \gamma \max_{a'} Q_\pi(s', a'; \boldsymbol{\theta}_{t-1}) \,|\, s_t, a_t]$ is the estimated function value at time step t when the current state and action are s and a respectively.

Deep reinforcement training phase: In the training phase of the DQN, the weights $\boldsymbol{\theta}_t$ in the DQN are updated after every iteration in time t using the *stochastic gradient descent* (SGD) algorithm on a minibatch of data. SGD starts with a random initial value of $\boldsymbol{\theta}$ and performs an iterative process to update $\boldsymbol{\theta}$ using a step size $\eta > 0$ as follows:

$$\boldsymbol{\theta}_{t+1} := \boldsymbol{\theta}_t - \eta \nabla L_t(\boldsymbol{\theta}_t). \tag{8}$$

The training of the DQN is facilitated by "experience replay" [21]. The experience replay buffer $\mathcal{D}$ stores the experiences at each time step t. An experience e_t is defined as $e_t := (s_t, a_t, r_{s,s',a}[t], s_t')$. We draw samples of experience at random from this buffer and perform minibatch training on the DQN. This approach offers advantages of stability and avoidance of local minimum convergence [19]. The use of experience replay also justifies the use of off-policy learning algorithms, since the current parameters of the DQN are different from those used to generate the sample from $\mathcal{D}$.

We define the state-action value function estimated by the DQN $Q_\pi^*(s, a)$ as

$$Q_\pi^*(s_t, a_t) := \mathbb{E}_{s'}\left[r_{s,s',a} + \gamma \max_{a'} Q_\pi^*(s', a') \,\middle|\, s_t, a_t\right], \tag{9}$$

which is known as the Bellman equation. Here, γ: $0 < \gamma < 1$ is the *discount factor* and determines the importance of the predicted future rewards. The next state is s' and the next action is a'. Our goal using DQN is to find a solution to maximize the state-action function $Q^{*}_{\pi}(s_t, a_t)$.

Often compared with deep Q-learning is the tabular version of Q-learning [20]. Despite the finite size of the states and action space, tabular Q-learning is slow to converge is because its convergence requires the state-action pairs to be sampled infinitely often [20], [22]. Further, tabular RL requires a non-trivial initialization of the $\mathbf{Q} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$ table to avoid longer convergence times [23]. We discuss tabular Q-learning in Section V.

Policy selection: In general, Q-learning is an off-policy reinforcement learning algorithm. An off-policy algorithm means that a near-optimal policy can be found even when actions are selected according to an arbitrary exploratory policy [20]. Due to this, we choose a near-greedy action selection policy. This policy has two modes:

1) *exploration*: the agent tries different actions at random at every time step t to discover an effective action a_t.
2) *exploitation*: the agent chooses an action at time step t that maximizes the state-action value function $Q_{\pi}(s, a; \boldsymbol{\theta}_t)$ based on the previous experience.

In this policy, the agent performs exploration with a probability ϵ and exploitation with probability of $1 - \epsilon$, where ϵ: $0 < \epsilon < 1$ is a hyperparameter that adjusts the trade-off between exploration and exploitation. This trade-off is why this policy is also called the ϵ-greedy action selection policy.

At each time step t, the UEs move at speed v and the agent plays a certain action a_t from its current state s_t. The agent receives a reward $r_{s,s',a}[t]$ and moves to a target state $s' := s_{t+1}$. We call the period of time in which an interaction between the agent and the environment takes place an *episode*. One episode has a duration of T time steps. An episode is said to have *converged* if within T time steps the target objective was fulfilled.

In our DQN implementation, we particularly keep track of the UE coordinates. When UE coordinates are reported back to the network and used to make informed decisions, the performance of the network improves [24]. Therefore, UE coordinates need to be part of the DRL state space $\mathcal{S}$.

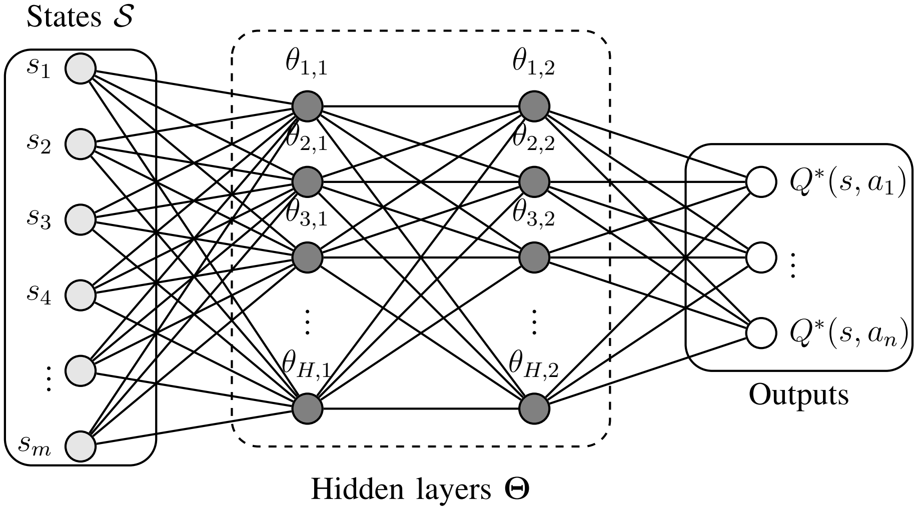

Fig. 3. Structure of the deep Q-network used for the implementation of the algorithms with two hidden layers each of dimension H. Here, $(u, v) = (H, 2)$, $|\mathcal{S}| = m$, and $|\mathcal{A}| = n$.

V. Deep Reinforcement Learning in Voice Power Control and Interference Coordination

In this section, we describe our proposed voice power control and interference coordination reinforcement learning algorithm as well as the baseline solutions which we compare our solution against. First, we describe the fixed power allocation algorithm, which is the industry standard algorithm today, then the implementation of the algorithm using tabular implementation of Q-learning. Finally, we show our proposed algorithm.

1) Fixed Power Allocation: We introduce the *fixed power allocation* (FPA) power control as a baseline algorithm which allows to set the transmit signal power at a specific value. No interference coordination is implemented in FPA. Total transmit power is simply divided equally among all the PRBs and is therefore constant:

$$P_{\text{TX},b}[t] := P_{\text{BS}}^{\max} - 10\log N_{\text{PRB}} + 10\log N_{\text{PRB},b}[t] \qquad \text{(dBm)} \tag{10}$$

where N_{PRB} is the total number of physical resource blocks in the BS and $N_{\text{PRB},b}$ is the number of available PRBs to the UE in the b-th BS at the time step t.

FPA with adaptive modulation and coding is the industry standard algorithm [13]. Based on UE-originating reports, the serving BS attempts to change the *modulation and code schemes* (MCS) of the transmission. This change, also known as the link adaptation, results in an improved effective SINR and a reduction in the voice packet error rate.

2) Tabular RL: We use a tabular setting of Q-learning (also known as vanilla Q-learning) to implement the algorithm for voice communication. In a tabular setting, the state-action value function $Q_\pi(s_t, a_t)$ is represented by a table $\mathbf{Q} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$. There is no neural network involvement and the Q-learning update analog of (9) is defined as:

$$\begin{aligned} Q_\pi(s_t, a_t) &:= (1-\alpha)Q_\pi(s_t, a_t) + \\ &\alpha\left(r_{s,s',a} + \gamma \max_{a'} Q_\pi(s', a')\right) \end{aligned} \tag{11}$$

where $Q_\pi(s_t, a_t) := [\mathbf{Q}]_{s_t,a_t}$. Here, $\alpha > 0$ is the learning rate of the Q-learning update and defines how aggressive the experience update is with respect to the prior experience. Besides the issue with convergence times, this algorithm has a tendency to diverge or oscillate [1], [23]. Computationally, the tabular setting suits problems with small ULA size $M = 1$ better than DQN, and maintaining a table $\mathbf{Q}$ is possible.

3) Proposed Algorithm: We propose Algorithm 1 which is a DRL-based approach. This algorithm performs both power control and interference coordination without the UE sending explicit power control or interference coordination commands. This use of the DQN may provide a lower computational overhead compared to the tabular Q-learning depending on the number of states and the depth of the DQN [23].

The main steps of Algorithm 1 are as follows:

- Select an optimization action at a time step t.
- If the periodicity $T_\text{periodicity}$ is fulfilled at t, select a beamforming vector from $\mathcal{F}$.
- Otherwise, select a power control or interference coordination action from $\mathcal{P}$.
- Assess the impact on $\gamma^\ell_\text{DL, eff}[t]$.
- Reward the action taken based on the impact on $\gamma^\ell_\text{DL, eff}[t]$ and its distance from $\gamma_\text{DL, target}$ or $\gamma_\text{DL, min}$.
- Train the DQN based on the outcomes.

The use of $T_\text{periodicity}$ ensures that the UE receives an updated beam whenever a PCIC action is made. Power control for the serving BS b is described as

$$P_{\text{TX},b}[t] = \min(P^\text{max}_\text{BS}, P_{\text{TX},b}[t-1] + \text{PC}_b[t]) \tag{12}$$

We add one more condition for the interference coordination on the interfering BS ℓ as

$$P_{\text{TX},\ell}[t] = \min(P^\text{max}_\text{BS}, P_{\text{TX},\ell}[t-1] + \text{IC}_\ell[t]) \tag{13}$$

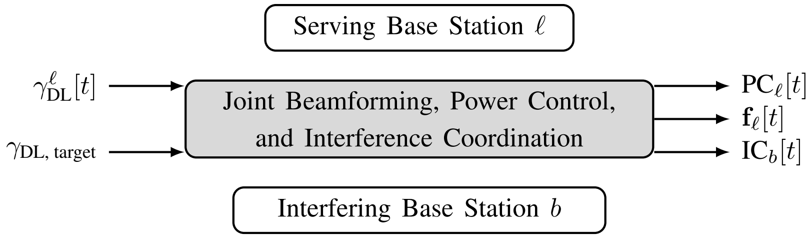
Serving Base Station ℓ
$\gamma^{\ell}_{\text{DL}}[t]$
$\gamma_{\text{DL, target}}$
Joint Beamforming, Power Control, and Interference Coordination
$\text{PC}_{\ell}[t]$
$\mathbf{f}_{\ell}[t]$
$\text{IC}_{b}[t]$
Interfering Base Station b

Fig. 4. Downlink joint beamforming, power control, and interference coordination module.

where the role of the BS (serving vs interfering) can change based on the UE. IC and PC commands are actually the same, but the role of the BS makes one an interferer (which needs coordination) and the other a server (which needs power control). We model the PCIC algorithm using deep Q-learning as shown in Algorithm 1. Our proposed algorithm solves (6).

Different from [1], we use adaptive coding for all three algorithms of voice where for a given code rate $R \leq 1$, we compute the effective SINR as $\gamma^{\ell}_{\text{DL, eff}}[t] = \gamma^{\ell}_{\text{DL}}[t]/R$. We choose to fix the modulation since voice bearers do not typically require high data rates [1]. This effective SINR $\gamma^{\ell}_{\text{DL, eff}}[t]$ is the quantity we optimize in Algorithm 1.

For FPA, the runtime complexity is $\mathcal{O}(1)$. For tabular Q-learning PCIC, the runtime complexity is $\mathcal{O}(|\mathcal{S}^{\text{voice}}||\mathcal{A}^{\text{voice}}|)$ [23], where $|\mathcal{S}^{\text{voice}}|, |\mathcal{A}^{\text{voice}}|$ are the state and action sets for voice bearers. 2) For deep Q-learning PCIC, the runtime complexity is at least in $\mathcal{O}(k(\boldsymbol{\Theta})|\mathcal{A}^{\text{voice}}|)$ [25], where $k(\boldsymbol{\Theta})$ is a multiplicative function of the hidden layers $\boldsymbol{\Theta}$. We observe that our proposed deep Q-learning based algorithm to have the highest runtime among all three.

VI. Deep Reinforcement Learning in mmWave Beamforming Power Control and Interference Coordination

In this section, we present our proposed algorithms and quantitatively describe the changes in the SINR as a result of the movement of the UEs and optimization actions of the RL-based algorithm.

A. Proposed Algorithm

We propose an DRL-based algorithm where the beamforming vectors ans transmit powers at the base stations are controlled to maximize the objective function in (6). More specifically, the proposed RL solution first selects the beamforming vectors from the available codebook $\mathcal{F}$ then controls the transmit powers for a duration $T_{\text{periodicity}}$.

```
Algorithm 1: Joint Beamforming and PCIC (JB-PCIC)
Input: The DL received SINR measured by the UEs.
Output: Sequence of beamforming, power control, and interference coordination commands to solve (6).
Initialize time, states, actions, fault handling register, and replay buffer $\mathcal{D}$.
repeat
    repeat
        $t := t + 1$
        Observe current state $s_t$.
        $\epsilon := \max(\epsilon \cdot d, \epsilon_{\text{min}})$
        Sample $r \sim \text{Uniform}(0, 1)$
        if $r \leq \epsilon$ then
            Select an action $a_t \in \mathcal{A}$ at random.
        else
            Select an action $a_t = \arg\max_{a'} Q_\pi(s_t, a'; \boldsymbol{\theta}_t)$.
        end
        if $t \bmod T_{\text{periodicity}} = 0$ then
            Perform action $a_t$ if it is a beamforming action.
        else
            Perform action $a_t$ if it is a power control the serving cell or coordinate the interference of the neighbors.
        end
        Compute $\gamma^{\ell}_{\text{DL, eff}}[t]$ and $r_{s,s',a}[t]$ from (16), (17).
        if $\gamma^{\ell}_{\text{DL, eff}}[t] < \gamma_{\text{min}}$ then $r_{s,s',a}[t] := r_{\text{min}}$
        if $\gamma^{\ell}_{\text{DL, eff}}[t] \geq \gamma_{\text{DL, target}}$ then $r_{s,s',a}[t] := r_{s,s',a}[t] + r_{\text{max}}$
        Observe next state $s'$.
        Store experience $e[t] \triangleq (s_t, a_t, r_{s,s',a}, s')$ in $\mathcal{D}$.
        Minibatch sample from $\mathcal{D}$ for experience $e_j \triangleq (s_j, a_j, r_j, s_{j+1})$.
        Set $y_j := r_j + \gamma \max_{a'} Q_\pi(s_{j+1}, a'; \boldsymbol{\theta}_t)$
        Perform SGD on $(y_j - Q_\pi(s_j, a_j; \boldsymbol{\theta}_t))^2$ to find $\boldsymbol{\theta}^*$
        Update $\boldsymbol{\theta}_t := \boldsymbol{\theta}^*$ in the DQN and record loss $L_t$
        $s_t := s'$
    until $t \geq T$
    Average the losses for this episode.
until convergence
```

TABLE I

REINFORCEMENT LEARNING HYPERPARAMETERS

Parameter	Value	Parameter	Value
Discount factor γ	0.995	Exploration rate decay d	0.9995
Initial exploration rate ϵ	1.000	Minimum exploration rate $(\epsilon_{\text{min}}^{\text{voice}}, \epsilon_{\text{min}}^{\text{bf}})$	(0.15,0.10)
Number of states	8	Number of actions $(\lvert\mathcal{A}^{\text{voice}}\rvert, \lvert\mathcal{A}^{\text{bf}}\rvert)$	(8,10)
Deep Q-Network width H	8	Deep Q-Network depth	2

First, selecting the beamforming vector is performed as follows. The agent searches the beamforming codebook using circular increments:

$$n \mapsto \mathbf{f}_n[t] \colon n := (n+1) \bmod M \tag{14}$$

for BSs b and ℓ independently. We monitor the change in $\gamma_{\text{DL}}^{\ell}$ as a result of the change in the beamforming vector. We assume a code rate of $R = 1$ in computing $\gamma_{\text{DL, eff}}^{\ell}$.

After the beamforming vectors are selected for s given UE, the agent can perform power control over the next $T_{\text{periodicity}} - 1$ time steps to control the transmit power of the BS to this UE (or the interference coordination of other BSs) is made. The selection of the transmit power is governed by (12) and (13), both of which define the set $\mathcal{P}$.

For our proposed algorithm, the runtime complexity of the deep reinforcement learning is in $\mathcal{O}(k(\mathbf{\Theta})L|\mathcal{F}|)$ [25].

B. Optimal Beamforming

The optimal beamforming and PCIC algorithm is an exhaustive search in the Euclidean space $\mathcal{P}\times\mathcal{F}$ per BS to optimize the SINR. While the size of $\mathcal{P}$ can be selected regardless of the number of the antennas in the ULA M, the size of $\mathcal{F}$ is directly related to M. This algorithm solves (6). This algorithm may perform well for small M and small number of BSs L. However, we observe that with large M the search time grows in $\mathcal{O}((|\mathcal{P}||\mathcal{F}|)^L)$, compared with the proposed algorithm, which is linear in L as shown earlier.

Since beamforming and PCIC can be applied to either voice or data by modifying the action space $\mathcal{A}$, we call our algorithm the *joint beamforming, power control, and interfernce coordination* (JB-PCIC) algorithm.

VII. PERFORMANCE MEASURES

In this section we introduce the performance measures we use to benchmark our algorithms.

TABLE II

Joint Beamforming Power Control Algorithm – Radio Environment Parameters

Parameter	Value	Parameter	Value
Base station (BS) maximum transmit power $P_{\text{BS}}^{\max}$	46 dBm	Downlink frequency band	(2100 MHz, 28 GHz)
Cellular geometry	circular	Cell radius r	(350, 150) m
Propagation model (voice, bf)	(COST231, [26])	User equipment (UE) antenna gain	0 dBi
Antenna gain ($G_{\text{TX}}^{\text{voice}}, G_{\text{TX}}^{\text{bf}}$)	(11, 3) dBi	Inter-site distance R	(525, 225) m
Max. number of UEs per BS N	10	Number of multipaths N_p	(15, 4)
Probability of LOS $p_{\text{LOS}}^{\text{voice}}, p_{\text{LOS}}^{\text{bf}}$	(0.9, 0.8)	UE average movement speed v	(5, 2) km/h
Number of transmit antennas $M^{\text{voice}}, M^{\text{bf}}$	(1,$\{4, 8, 16, 32, 64\}$)	Radio frame duration $T^{\text{voice}}, T^{\text{bf}}$	(20, 15) ms

A. *Convergence*

We define convergence ζ in terms of the episode at which the target SINR is fulfilled over the entire duration of T for all UEs in the network. We expect that as the number of antennas in the ULA M increase, the convergence time ζ will also increase. In voice, convergence as a function of M is not applicable, since we only use single antennas. For several random seeds, we take the aggregated percentile convergence episode.

B. *Coverage*

We build a *complement cumulative distribution function* (CCDF) of $\gamma_{\text{DL}}^{\ell}$ following [27] by running the simulation many times and changing the random seed, effectively changing the way the users are dropped in the network.

C. *Sum-rate capacity*

Using $\gamma_{\text{DL}}^{\ell}$, we compute the average sum-rate capacity as

$$C = \frac{1}{T} \sum_{t=1}^{T} \sum_{\ell} \log_2(1 + \gamma_{\text{DL}}^{\ell}[t]) \tag{15}$$

which is an indication of the data rate served by the network. We then obtain the median sum-rate capacity resulting from computing (15) over many episodes.

VIII. Simulation Results

In this section, we evaluate the performance of our RL-based proposed solutions in terms of the performance measures in Section VII. First, we describe the adopted setup in Section VIII-A before delving into the simulation results in Sections VIII-B and VIII-C.

A. Setup

We adopt the network, signal, and channel models in Section II. The users in the urban cellular environment are uniformly distributed in its coverage area. The users are moving at a speed v with both log-normal shadow fading and small-scale fading. The cell radius is r and the inter-site distance $R = 1.5r$. The users experience a probability of line of sight of p_{LOS} and receive a radio frame duration as shown in Table II. We set the target SINRs as:

$$\gamma_{\text{DL, target}}^{\text{voice}} := 3\,\text{dB},$$

$$\gamma_{\text{DL, target}}^{\text{bf}} := 20 + 10\log M\,\text{dB}$$

and set the minimum SINR at -3 dB below which the episode is declared aborted and the data session is unable to continue (i.e., dropped).

The hyperparameters required to tune the RL-based model are shown in Table I. We refer to our source code [28] for further implementation details. Further, we run Algorithm 1 on the cellular network with its parameters in Table II. The simulated states $\mathcal{S}$ are setup as:

$$\begin{aligned} (s_t^0, s_t^1) &:= \text{UE}_\ell(x[t], y[t]), & (s_t^2, s_t^3) &:= \text{UE}_b(x[t], y[t]), \\ s_t^4 &:= P_{\text{TX},\ell}[t], & s_t^5 &:= P_{\text{TX},b}[t], \\ s_t^6 &:= \mathbf{f}_n^\ell[t], & s_t^7 &:= \mathbf{f}_n^b[t], \end{aligned}$$

where (x, y) are the Cartesian coordinates of the given UE.

The simulated action space $\mathcal{A}$ contains:

- a_t^0: increase the transmit power of BS ℓ by 1 dB.
- a_t^1: increase the transmit power of BS ℓ by 3 dB.
- a_t^2: decrease the transmit power of BS ℓ by 3 dB.
- a_t^3: decrease the transmit power of BS b by 1 dB.
- a_t^4: decrease the transmit power of BS b by 3 dB.
- a_t^5: increase the transmit power of BS b by 1 dB.
- a_t^6: increase the transmit power of BS b by 3 dB.
- a_t^7: Step up the beamforming codebook index of BS ℓ.
- a_t^8: Step up the beamforming codebook index of BS b.

Here, we can infer that $\mathcal{P} = \{\pm 1, \pm 3\}$ dB offset from the transmit power. The actions to increase and decrease BS transmit powers are implemented as in (12) and (13). We introduce 3-dB power steps for voice only to compensate for not using beamforming. Therefore, we define the action subspaces $\mathcal{A}^{\text{voice}} := \{a_1, a_2, a_4, a_6\}$ are for the voice scenario only and $\mathcal{A}^{\text{bf}} := \{a_7, a_8\}$ are for the beamforming scenario only.

The reward we use in our proposed algorithms is divided into two tiers: 1) based on the timing of the action taken and 2) based on whether the target SINR has been met or the SINR falls below the minimum.

For the voice bearers, the reward is defined as

$$r^{\text{voice}}_{s,s',a}[t] := 3\mathbb{1}_{2\leq a_t\leq 5} + \mathbb{1}_{a_t\in\{0,1,6,7\}}, \qquad \forall s, s' \tag{16}$$

which rewards the agent more when it chooses to lower power than increase power. For the data bearers, the reward is

$$\begin{aligned} r^{\text{bf}}_{s,s',a}[t] := (t \bmod T_{\text{periodicity}})\mathbb{1}_{0\leq a_t\leq 3}+ \\ [15-(t \bmod T_{\text{periodicity}})]\mathbb{1}_{a_t\in\{4,5\}}, \qquad \forall s, s' \end{aligned} \tag{17}$$

where in both equations either a penalty r_{min} or a maximum reward r_{max} is added based on whether the minimum γ_{min} has been violated or $\gamma_{\text{DL, target}}$ has been achieved as shown in Algorithm 1. Here, it is also clear that the agent is rewarded more for searching in the beamforming codebook than attempting to power up or down. The reward changes with how far the time step t is from the periodicity interval, to enable the beamforming to take place early in the radio frame, and have power control subsequently follow that change.

We abort the episode if any of the constraints in (6) becomes inactive. At this stage, the RL agent receives a reward $r_{s,s',a}[t] := r_{\text{min}}$.

In our simulations, we use an NVIDIA GeForce GPU-based server. We further use TensorFlow on Python 3.6 in the implementation of our DQN. Our minibatch sample size is 32.

B. Outcomes

1) Convergence. Every time step is 1 ms, which is equivalent to the NR TTI duration. Even when the convergence episode is 1,378 for $M = 64$, this actually corresponds to 1.378 seconds of network time.

We also find that as the size of the ULA M increases, the number of episodes required converge increases. This is justified since the number of attempts to traverse the beamforming codebook increases almost linearly with the increase of M.

2) Coverage: for voice bearers we observe that the coverage as defined by the SINR CCDF improves everywhere. For data bearers, the coverage improves where the SINR monotonically

increases with the increase in M. While one may argue that obtaining more samples for $M = 64$ would enhance the characteristic of the CCDF, the reality is that its convergence takes the longest, making obtaining more samples a tedious task.

3) Sum-rate: the median sum-rate capacity increases logarithmically as a result of the increase of M, which is justified using (5) and (15).

C. Figures

Fig. 5 shows the CCDF of the effective SINR $\gamma_{\text{DL, eff}}$ for the voice PCIC algorithms, where we see that the FPA algorithm has the worst performance, which is expected since FPA has no power control or interference coordination. The performance of the tabular Q-learning and the deep Q-learning PCIC implementations are therefore better than FPA. We observe that deep Q-learning outperformed tabular Q-learning PCIC algorithm, and this is because deep Q-learning convergence to a better solution was not impeded by the choice of a initialization of the state-action value function, unlike the tabular Q-learning approach.

The coverage CCDF shows that the gap between $M = 4$ and $M = 16$ curves is much narrower than the gap between $M = 16$ and $M = 64$ curves in Fig. 6. This is justified since the SINR gain is a function of the antenna size M and the difference between $M = 16$ and $M = 64$ is much wider.

In Fig. 7, it takes longer convergence time as M increases. This is due to the longer time required for the agent to search through a grid of beams of size $|\mathcal{F}|$, which are typically narrower at large M. This causes the agent to spend longer time to meet the target SINR especially with UEs moving at speed v. The achieved SINR is proportional to the ULA antenna size M as shown in Fig. 8. This is expected as the beamforming gain is $||\mathbf{f}_b||^2 \leq M$. Consequently, the required transmit power from the BS will decrease, which is also shown in the figure.

Fig. 8 also shows the relative performance of JB-PCIC compared with the optimal algorithm. We observe that the performance gap of the transmit power of the base stations is narrow all across M even though the transmit power is not the optimization objective. However, the performance gap of the SINR is small for small antenna sizes $M = 4$, but continues to widen as the antenna size M increases. This is because the DQN may have become stuck at a local minimum.

Finally, Fig. 9 shows the sum-rate capacity of both the JB-PCIC and optimal algorithms. Similarly, the performance gap widens with larger M for the same reason discussed in Fig. 8.

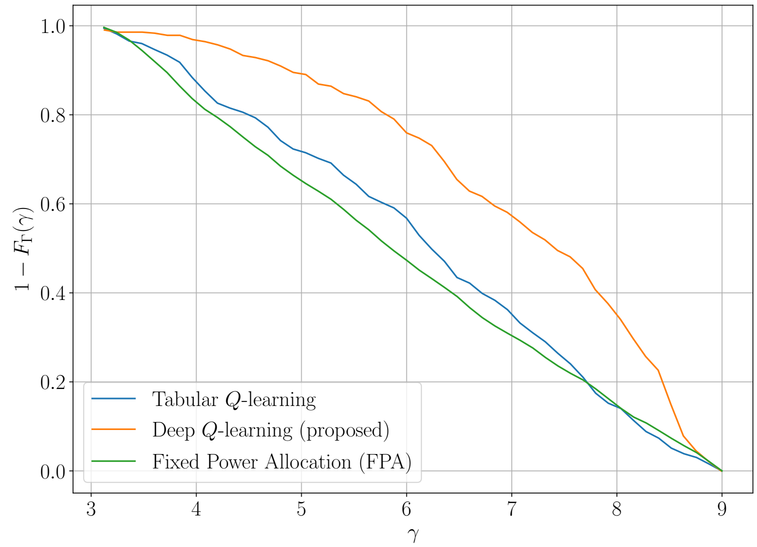

Fig. 5. Coverage CCDF plot of $\gamma_{\text{DL, eff}}^{\text{voice}}$ for three different voice power control and inteference coordination algorithms.

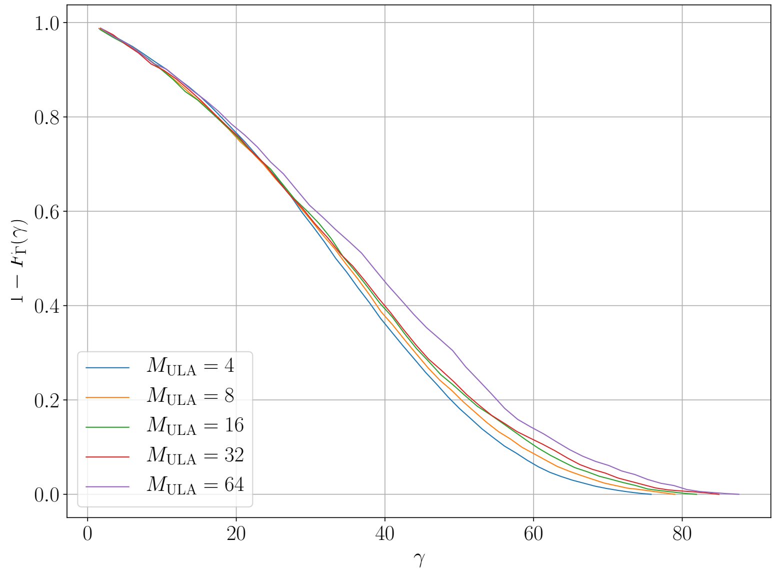

Fig. 6. Coverage CCDF plot of the effective SINR $\gamma_{\text{DL, eff}}$ for the proposed deep Q-learning algorithm vs. the number of antennas M.

IX. Conclusion

In this paper, we seek to maximize the downlink SINR in a multi-access OFDM cellular network from a multi-antenna base station to single-antenna user equipment. The user equipment experiences interference from other multi-antenna base stations. Our system uses sub-6 GHz frequencies for voice and mmWave frequencies for data. We assume that each base station can

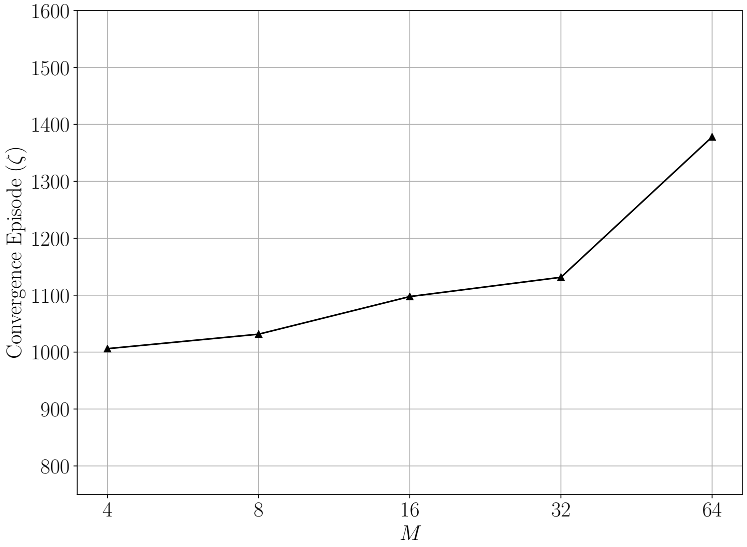

Fig. 7. The convergence time for the proposed deep Q-learning algorithm as a function of the number of antennas M.

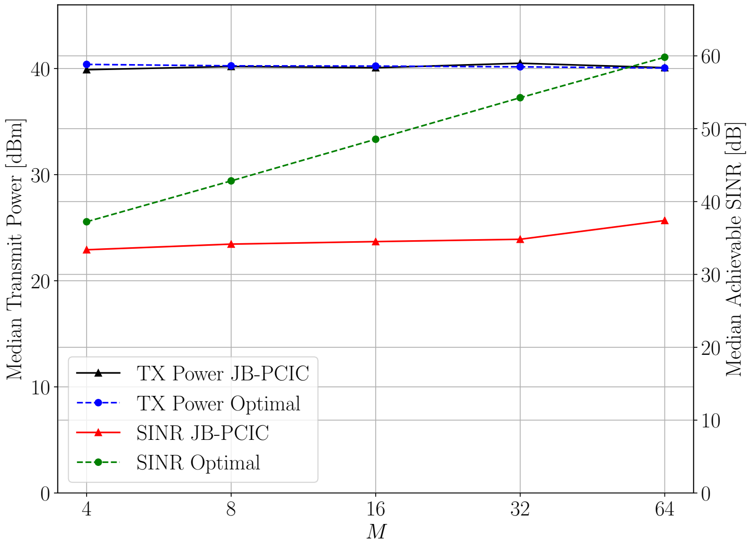

Fig. 8. Achievable SINR and transmit power for both the optimal and proposed JB-PCIC algorithms as a function of the number of antennas M.

select a beamforming vector from a finite set. The power control commands are also from a finite set. We show that a closed-form solution does not exist, and that finding the optimum answer requires an exhaustive search. An exhaustive search has a runtime that is exponential in the number of base stations.

To avoid an exhaustive search, we developed a joint beamforming, power control, and inter-

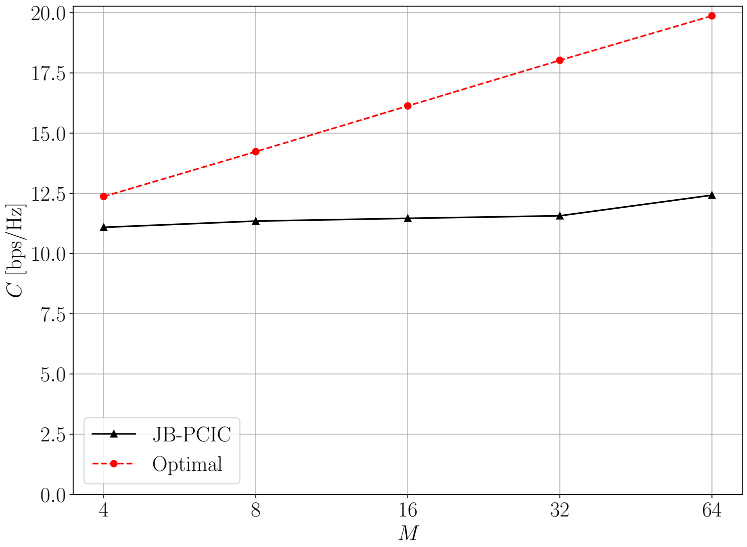

Fig. 9. Sum-rate capacity of the convergence episode as a function of the number of antennas M.

ference coordination algorithm (JB-PCIC) using deep reinforcement learning. The near-optimal SINR values achieved are higher than those achieved by industry standard algorithms. For voice communication, the proposed algorithm outperformed both the tabular and the fixed power allocation algorithms due to its faster convergence. The runtime complexity of the proposed algorithm is the product of the number of possible actions, number of base stations and number of base station antennas. That is, the runtime complexity is linear in each quantity.

Our proposed algorithm for joint beamforming, power control and interference coordinations requires that the UE sends its coordinates and its received SINR every millisecond to the base station. The proposed algorithm, however, does not require channel state information, which removes the need for channel estimation and the associated training sequences. Moreover, the overall amount of feedback from the UE is reduced because the UE would not need to send explicit commands for beamforming vector changes, power control, or interference coordination.

References

[1] F. B. Mismar and B. L. Evans, “Q-Learning Algorithm for VoLTE Closed Loop Power Control in Indoor Small Cells,” in *Proc. Asilomar Conference on Signals, Systems, and Computers*, Oct. 2018.

[2] S. Yun and C. Caramanis, “Reinforcement Learning for Link Adaptation in MIMO-OFDM Wireless Systems,” in *Proc. IEEE Global Telecommunications Conference*, Dec. 2010.

[3] M. Bennis and D. Niyato, “A Q-learning Based Approach to Interference Avoidance in Self-Organized Femtocell Networks,” in *Proc. IEEE Globecom Workshops*, Dec. 2010.

[4] J. Choi, "Massive MIMO With Joint Power Control," *IEEE Wireless Communications Letters*, vol. 3, no. 4, pp. 329–332, Aug. 2014.

[5] L. Zhu, J. Zhang, Z. Xiao, X. Cao, D. O. Wu, and X. Xia, "Joint Power Control and Beamforming for Uplink Non-Orthogonal Multiple Access in 5G Millimeter-Wave Communications," *IEEE Transactions on Wireless Communications*, vol. 17, no. 9, pp. 6177–6189, Sep. 2018.

[6] C. Luo, J. Ji, Q. Wang, L. Yu, and P. Li, "Online Power Control for 5G Wireless Communications: A Deep Q-Network Approach," in *Proc. IEEE Int. Conf. on Commun.*, May 2018.

[7] F. Rashid-Farrokhi, L. Tassiulas, and K. J. R. Liu, "Joint optimal power control and beamforming in wireless networks using antenna arrays," *IEEE Trans. on Commun.*, vol. 46, no. 10, pp. 1313–1324, Oct. 1998.

[8] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Overall description," 3rd Generation Partnership Project (3GPP), TS 36.300, Jan. 2019. [Online]. Available: http://www.3gpp.org/dynareport/36300.htm

[9] R. Kim, Y. Kim, N. Y. Yu, S. Kim, and H. Lim, "Online Learning-based Downlink Transmission Coordination in Ultra-Dense Millimeter Wave Heterogeneous Networks," *IEEE Trans. on Wirel. Commun.*, vol. 18, no. 4, pp. 2200–2214, Mar. 2019.

[10] S. Wang, H. Liu, P. H. Gomes, and B. Krishnamachari, "Deep Reinforcement Learning for Dynamic Multichannel Access in Wireless Networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 2, pp. 257–265, Jun. 2018.

[11] Y. Wang, M. Liu, J. Yang, and G. Gui, "Data-Driven Deep Learning for Automatic Modulation Recognition in Cognitive Radios," *IEEE Trans. on Veh. Technol.*, vol. 68, no. 4, pp. 4074–4077, Apr. 2019.

[12] T. Erpek, Y. E. Sagduyu, and Y. Shi, "Deep learning for launching and mitigating wireless jamming attacks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 1, pp. 2–14, Mar. 2019.

[13] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures," 3rd Generation Partnership Project (3GPP), TS 36.213, Dec. 2015. [Online]. Available: http://www.3gpp.org/dynareport/36213.htm

[14] A. Alkhateeb, O. El Ayach, G. Leus, and R. W. Heath Jr., "Channel estimation and hybrid precoding for millimeter wave cellular systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 831–846, Oct. 2014.

[15] R. W. Heath Jr., N. Gonzalez-Prelcic, S. Rangan, W. Roh, and A. Sayeed, "An overview of signal processing techniques for millimeter wave MIMO systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 3, pp. 436–453, April 2016.

[16] P. Schniter and A. Sayeed, "Channel Estimation and Precoder Design for Millimeter Wave Communications: The Sparse Way," in *Proc. Asilomar Conference on Signals, Systems and Computers*, Nov. 2014.

[17] T. Rappaport, F. Gutierrez, E. Ben-Dor, J. Murdock, Y. Qiao, and J. Tamir, "Broadband millimeter-wave propagation measurements and models using adaptive-beam antennas for outdoor urban cellular communications," *IEEE Transactions on Antennas and Propagation*, vol. 61, no. 4, pp. 1850–1859, Apr. 2013.

[18] T. S. Rappaport, R. W. Heath Jr, R. C. Daniels, and J. N. Murdock, *Millimeter Wave Wireless Communications*. Pearson Education, 2014.

[19] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari with Deep Reinforcement Learning," *NIPS Deep Learning Workshop*, 2013. [Online]. Available: http://arxiv.org/abs/1312.5602

[20] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*. The MIT Press, 1998.

[21] L.-J. Lin, "Reinforcement Learning for Robots Using Neural Networks," Ph.D. dissertation, Carnegie-Mellon University, Pittsburg, PA, 1993.

[22] M. Simsek, A. Czylwik, A. Galindo-Serrano, and L. Giupponi, "Improved Decentralized Q-learning Algorithm for Interference Reduction in LTE-femtocells," in *Proc. Wireless Advanced*, Jun. 2011.

[23] F. B. Mismar, J. Choi, and B. L. Evans, “A Framework for Automated Cellular Network Tuning with Reinforcement Learning,” *IEEE Trans. on Commun., to appear*, Jun. 2019.

[24] F. B. Mismar and B. L. Evans, “Partially Blind Handovers for mmWave New Radio Aided by Sub-6 GHz LTE Signaling,” in *Proc. IEEE International Conference on Communications Workshops*, May 2018.

[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion *et al.*, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, 2011.

[26] 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. on Wirel. Commun.*, vol. 15, no. 10, pp. 6939–6947, Oct. 2016.

[27] T. Bai and R. W. Heath Jr., “Coverage and Rate Analysis for Millimeter-Wave Cellular Networks,” *IEEE Transactions on Wireless Communications*, vol. 14, no. 2, pp. 1100–1114, Feb. 2015.

[28] F. B. Mismar. (2019) Source code. [Online]. Available: https://github.com/farismismar/Deep-Reinforcement-Learning-for-5G-Networks