2021 IEEE Globecom Workshops (GC Wkshps) | 978-1-6654-2390-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/GCWkshps52748.2021.9681985

Deep Reinforcement Learning for Joint Spectrum and Power Allocation in Cellular Networks

Yasar Sinan Nasir and Dongning Guo
Department of Electrical and Computer Engineering
Northwestern University, Evanston, IL 60208.

***Abstract*—A wireless network operator typically divides its radio spectrum into a number of subbands and reuse them to serve traffic in many cells. To mitigate co-channel interference, allocation of spectrum and power resources needs to be adapted to time-varying channel and traffic conditions throughout the network. Standard model-based network utility maximization is severely limited by the computational complexity and the difficulty of acquiring instantaneous global channel state information. In this paper, a learning-based method is proposed to optimize discrete subband allocations and continuous power allocations using (generally delayed and inaccurate) channel state information in local and nearby cells. For these two types of allocations, two complementary deep reinforcement learning algorithms are designed to be executed and trained simultaneously to maximize a joint objective. Simulation results show that the proposed method outperforms a state-of-the-art fractional programming algorithm as well as a previous solution based on deep reinforcement learning.**

I. Introduction

In today's cellular networks, the spectrum is divided into many subbands. Each cellular device suffers from the co-channel interference caused by nearby access points which use the same subbands. The interference can be particularly severe with dense, irregularly placed access points. Joint subband selection and transmit power control is a crucial tool for interference mitigation.

For the single band scenario, state-of-the-art optimization methods such as fractional programming (FP) [1] have been applied to the power control problem to reach a near-optimal allocation. We assume that the number of subbands is much less than the number of cellular devices and that each link can occupy at most one subband at a time. Therefore, the joint subband selection and power allocation problem involves *mixed integer programming* [2].

Conventional optimization-based schemes such as fractional programming are model-driven and require a mathematically tractable and sufficiently accurate model [3]. Furthermore, such a scheme is in general centralized and requires instantaneous global channel state information (CSI). A centralized solution's computational complexity does not scale well for a large number of cellular devices. Therefore, its implementation is quite challenging in a practical scenario where network and channel conditions vary.

Recently, there has been extensive research on reinforcement learning based transmit power control which is purely data-driven [3]. For the single band scenario, deep Q-learning has been considered on a "centralized training and distributed execution" framework in [4]–[6]. Since deep Q-learning applies only to discrete power control, the continuous transmit power domain had to be quantized in [4]–[6] which may introduce a quantization error as discussed in [7], [8]. Reference [7] first showed the performance in [5] can be improved by quantizing the transmit power using logarithmic step size instead of linear step size, and propose replacing deep Q-learning algorithm by an actor-critic learning algorithm called deep deterministic policy gradient that applies to continuous power control.

For the multiple band scenario, Tan *et al.* [2] have proposed to train a single deep Q-network that jointly handles both subband selection and transmit power control. One major drawback of this approach is that the action space is the Cartesian product of available subbands and quantized transmit power levels. Therefore, the deep Q-network output layer size and the number of state action pairs to be visited for convergence during training do not scale well with increasing number of subbands. Moreover, the joint deep Q-learning approach is not directly applicable to a problem that includes both discrete and continuous variables. To overcome these challenges, we propose a novel approach that consists of two layers, where the bottom layer is responsible for continuous power allocation with deep deterministic policy gradient, and the top layer schedules discrete subbands by adapting deep Q-learning. Using simulations, we evaluate the proposed learning scheme by comparing it with the joint deep Q-learning approach and the fractional programming algorithm in terms of convergence rate and sum-rate performance.

II. System Model

In this paper, we consider a cellular network with N links that are placed in K cells and share M subbands. We denote the set of link and subband indexes by $\mathcal{N} = \{1, \dots, N\}$ and $\mathcal{M} = \{1, \dots, M\}$, respectively. Link n is composed of receiver n and its transmitter n. Transmitter n is placed at the corresponding cell center that includes receiver n within its cell boundaries. We consider a fully synchronized time slotted system with a fixed slot duration of T. We assume that all transmitters and receivers are equipped with a single antenna.

This material is based upon work supported by the National Science Foundation under Grants No. CCF-1910168, No. CNS-2003098, AST-2037838, and AST-2037852 as well as a gift from Intel Incorporation.

978-1-6654-2390-8/21/$31.00 ©2021 IEEE

Due to relative scarcity of available spectrum, K tends to be much larger than M, i.e., $K \gg M$. We let each link pick one subband at the beginning of each time slot.

Similar to [9], our channel model is composed of two parts: large and small scale fading. For simplicity, we assume that the large-scale fading is same across all subbands, whereas the small-scale fading is frequency selective, i.e., different across all subbands [2]. Within each subband, small-scale fading is assumed to be block-fading and flat. Let $g_{n\to l,m}^{(t)}$ denote the downlink channel gain from transmitter n to receiver l on subband m in time slot t:

$$g_{n\to l,m}^{(t)} = \beta_{n\to l} \left| h_{n\to l,m}^{(t)} \right|^2, \quad t = 1, 2, \dots, \tag{1}$$

where $\beta_{n\to l}$ is the large-scale fading that includes both path loss and log-normal shadowing, and $h_{n\to l,m}^{(t)}$ is the small-scale Rayleigh fading. We assume that the large-scale fading remains the same through many time slots. Note that in case of mobile receivers, a time index can be associated with $\beta_{n\to l}$.

We adopt Jake's fading model to describe $h_{n\to l,m}^{(t)}$ [9]. Accordingly, the small-scale fading for each channel follows a first-order complex Gauss-Markov process:

$$h_{n\to l,m}^{(t)} = \rho h_{n\to l,m}^{(t-1)} + \sqrt{1-\rho^2} e_{n\to l,m}^{(t)}, \tag{2}$$

where the correlation between two successive fading blocks $\rho = J_0(2\pi f_d T)$ with $J_0(.)$ being the zeroth-order Bessel function of the first kind depending on the maximum Doppler frequency f_d. Besides, $h_{n\to l,m}^{(0)}$ and the channel innovation process $e_{n\to l,m}^{(1)}, e_{n\to l,m}^{(2)}, \dots$ are independent and identically distributed circularly symmetric complex Gaussian random variables with unit variance. The cells are agnostic to the specific fading statistics a priori.

We use binary variables $\alpha_{n,m}^{(t)}$ to indicate the subband selection of link n in time slot t. If link n selects subband m, we have $\alpha_{n,m}^{(t)} = 1$ and $\alpha_{n,j}^{(t)} = 0$ for every $j \neq m$. We denote the transmit power of transmitter n in time slot t as $p_n^{(t)}$. The signal-to-interference-plus-noise at receiver n on subband m in time slot t is given by

$$\gamma_{n,m}^{(t)} = \frac{\alpha_{n,m}^{(t)} g_{n\to n,m}^{(t)} p_n^{(t)}}{\sum_{l\neq n} \alpha_{l,m}^{(t)} g_{l\to n,m}^{(t)} p_l^{(t)} + \sigma^2}, \tag{3}$$

where σ^2 is the additive white Gaussian noise power spectral density at receiver n. Assuming normalized bandwidth, the downlink spectral efficiency achieved by link n on subband m during time slot t is

$$C_{n,m}^{(t)} = \log\left(1 + \gamma_{n,m}^{(t)}\right). \tag{4}$$

III. Problem Formulation

Denoting subband and power vectors in time slot t as $\boldsymbol{\alpha}^{(t)} = \left[\alpha_{1,1}^{(t)}, \alpha_{1,2}^{(t)}, \dots, \alpha_{N,M}^{(t)}\right]^{\mathsf{T}}$ and $\boldsymbol{p}^{(t)} = \left[p_1^{(t)}, \dots, p_N^{(t)}\right]^{\mathsf{T}}$, respectively, we formulate the sum-rate maximization problem as [2], [10]:

$$\begin{aligned}
\underset{\boldsymbol{p}^{(t)}, \boldsymbol{\alpha}^{(t)}}{\text{maximize}} \quad & \sum_{n=1}^{N} C_n^{(t)} && \text{(P1a)}\\
\text{subject to} \quad & 0 \leq p_n^{(t)} \leq P_{\max}, \forall n \in \mathcal{N}, && \text{(P1b)}\\
& \alpha_{n,m}^{(t)} \in \{0,1\}, \forall n \in \mathcal{N}, \forall m \in \mathcal{M}, && \text{(P1c)}\\
& \sum_{m\in\mathcal{M}} \alpha_{n,m}^{(t)} = 1, \forall n \in \mathcal{N}, && \text{(P1d)}
\end{aligned}$$

where $C_n^{(t)} = \sum_{m=1}^{M} C_{n,m}^{(t)}$ is link n's achieved spectral efficiency, and (P1b) restricts the transmit power to be non-negative and no larger than $P_{\max}$.

Unfortunately, (P1) is in general non-convex and requires *mixed integer programming* to be carried out for each time slot as channel varies. Even for a given subband selection $\boldsymbol{\alpha}^{(t)}$, this problem has been proven to be NP-hard [10]. Conventional algorithms such as fractional programming are centralized solutions to (P1), but these algorithms still require many iterations to converge and their computational complexity does not scale well with increasing number of links. Besides that, obtaining instantaneous global CSI in a centralized controller and sending the allocation decisions back to all transmitters is difficult in practice.

IV. A Deep Reinforcement Learning Framework

A. Overview of Reinforcement Learning

Model-free reinforcement learning [11] is a trial-and-error process where an agent interacts with an unknown environment in a sequence of discrete time steps to achieve a task. At time t, agent first observes the current state of the environment which is a tuple of relevant environment features and is denoted as $s^{(t)} \in \mathcal{S}$, where $\mathcal{S}$ is the set of possible states. It then takes an action $a^{(t)} \in \mathcal{A}$ from an allowed set of actions $\mathcal{A}$ according to a policy which can be either stochastic, i.e., π with $a^{(t)} \sim \pi(\cdot|s^{(t)})$ or deterministic, i.e., μ with $a^{(t)} = \mu(s^{(t)})$ [12]. Since the interactions are often modeled as a Markov decision process, the environment moves to a next state $s^{(t+1)}$ following an unknown transition matrix that maps state-action pairs onto a distribution of next states, and the agent receives a reward $s^{(t+1)}$. Overall, the above process is described as an experience at $t+1$ denoted as $e^{(t+1)} = \left(s^{(t)}, a^{(t)}, r^{(t+1)}, s^{(t+1)}\right)$. The goal is to learn a policy that maximizes the cumulative discounted reward at time t, defined as

$$R^{(t)} = \sum_{\tau=0}^{\infty} \gamma^{\tau} r^{(t+\tau+1)}, \tag{5}$$

where $\gamma \in (0,1]$ is the discount factor.

Next, we introduce two reinforcement learning methods that are used in the proposed design.

Q-learning [11] is a popular reinforcement learning method that learns an action value function $Q(s,a)$. Let $\pi(a|s)$ be the probability of taking action a conditioned on the current state being s. Assuming a stationary setting, the Q-function under a π is the expected cumulative discounted reward when action a is taken in state s:

$$Q^{\pi}(s,a)=\mathbb{E}_{\pi}\left[R^{(t)}\middle|s^{(t)}=s,a^{(t)}=a\right]. \tag{6}$$

Assuming the optimal policy $\pi^*(a|s)$ be equal to 1 for the most favorable action a^* that maximizes $Q^{\pi^*}(s,a)$ for a given state s, the optimal Q-function satisfies the Bellman equation:

$$Q^{\pi^*}(s,a)=\mathcal{R}(s,a)+\gamma\sum_{s'\in S}\mathcal{P}^{a}_{ss'}\max_{a'}Q^{\pi^*}(s',a'), \tag{7}$$

where $\mathcal{R}(s,a)=\mathbb{E}\left[r^{(t+1)}\middle|s^{(t)}=s,a^{(t)}=a\right]$ is the expected reward of taking action a at state s, and $\mathcal{P}^{a}_{ss'}=\Pr\left(s^{(t+1)}=s'\middle|s^{(t)}=s,a^{(t)}=a\right)$ is the transition probability from state s to next state s' with action a. The classical Q-learning algorithm uses a lookup table to represent the Q-function values and employs the fixed-point relation in (7) to iteratively update these values. However, the classical lookup table approach is not practical for continuous or large discrete state spaces.

To overcome this drawback, deep Q-learning replaces the lookup table with a deep neural network which is called deep Q-network and expressed as $q(s,a;\boldsymbol{\psi})$ with $\boldsymbol{\psi}$ being its parameters [13]. As described in [13, Fig. 1], its input layer is fed by a given state s, and each port of its output layer gives the Q-function value for input s and corresponding action output. Deep Q-learning is an off-policy learning method that stores the past experiences in an experience replay memory denoted as $\mathcal{D}$ in the form of $e=(s,a,r',s')$. A small value for the maximum size of this memory, $|\mathcal{D}|$, will result with over-fitting, while a large value will slow down learning. Additionally, deep Q-learning adopts "quasi-static target network" technique that implies creating a target network with parameters $\boldsymbol{\psi}_{\text{target}}$ to predict the target values in the following mean-squared Bellman error:

$$L(\boldsymbol{\psi},\mathcal{D})=\mathbb{E}_{(s,a,r',s')\sim\mathcal{D}}\left[\left(y(r',s')-q\left(s,a;\boldsymbol{\psi}\right)\right)^2\right], \tag{8}$$

where the target $y(r',s')=r'+\gamma\max_{a'}q\left(s',a';\boldsymbol{\psi}_{\text{target}}\right)$. To minimize (8), $\boldsymbol{\psi}$ is updated by sampling a random mini-batch $\mathcal{B}$ from $\mathcal{D}$ and running gradient descent by

$$\nabla_{\boldsymbol{\psi}}\frac{1}{|\mathcal{B}|}\sum_{(s,a,r',s')\in\mathcal{B}}\left(y(r',s')-q\left(s,a;\boldsymbol{\psi}\right)\right)^2. \tag{9}$$

Each iteration is followed by updating $\boldsymbol{\psi}_{\text{train}}$ by $\boldsymbol{\psi}$. During the training, instead of fully exploiting the updated policy, the learning agent applies the ϵ-greedy strategy which takes a random action with a probability of ϵ for exploration.

On the other hand, to overcome the challenge of applying deep Q-learning to continuous action spaces. Reference [14] had proposed an actor-critic learning scheme called deep deterministic policy gradient. It iteratively trains a critic network, defined by ϕ, to represent an action-value function, and uses the critic network to train an actor network, defined by $\boldsymbol{\theta}$, that parameterizes a deterministic policy. We define the deterministic policy as $\mu:\mathcal{S}\rightarrow\mathcal{A}$, and for a given state s, the action is determined by $a=\mu(s;\boldsymbol{\theta})$. Hence, the target policy μ^* satisfies the Bellman property:

$$Q^{\mu^*}(s,a)=\mathcal{R}(s,a)+\gamma\sum_{s'\in S}\mathcal{P}^{a}_{ss'}Q^{\mu^*}(s',\mu^*(s')). \tag{10}$$

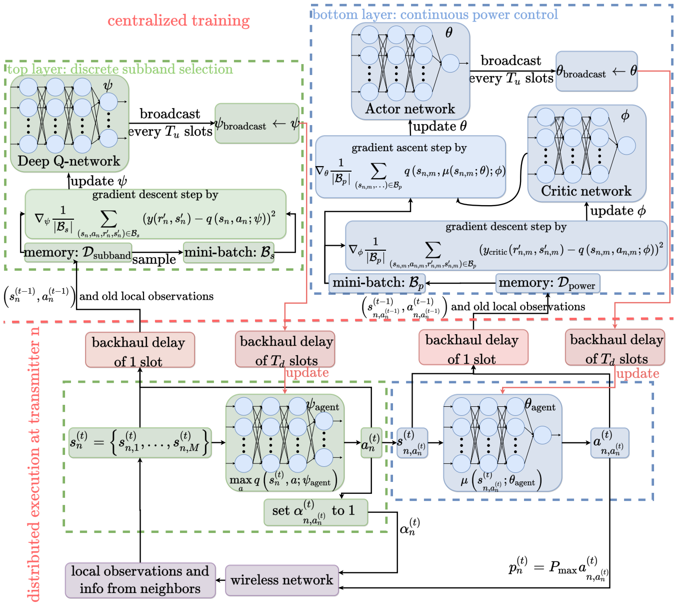

centralized training
bottom layer: continuous power control
top layer: discrete subband selection
θ
Actor network
broadcast
every T_u slots
$\theta_{\text{broadcast}}\leftarrow\theta$
update θ
ψ
Deep Q-network
broadcast
every T_u slots
$\psi_{\text{broadcast}}\leftarrow\psi$
update ψ
gradient ascent step by
$\nabla_{\theta}\frac{1}{|\mathcal{B}_p|}\sum_{(s_{n,m},\dots)\in\mathcal{B}_p}q\left(s_{n,m},\mu(s_{n,m};\theta);\phi\right)$
ϕ
Critic network
update ϕ
gradient descent step by
$\nabla_{\psi}\frac{1}{|\mathcal{B}_s|}\sum_{(s_n,a_n,r'_n,s'_n)\in\mathcal{B}_s}\left(y(r'_n,s'_n)-q\left(s_n,a_n;\psi\right)\right)^2$
memory: $\mathcal{D}_{\text{subband}}$
sample
mini-batch: $\mathcal{B}_s$
gradient descent step by
$\nabla_{\phi}\frac{1}{|\mathcal{B}_p|}\sum_{(s_{n,m},a_{n,m},r'_{n,m},s'_{n,m})\in\mathcal{B}_p}\left(y_{\text{critic}}\left(r'_{n,m},s'_{n,m}\right)-q\left(s_{n,m},a_{n,m};\phi\right)\right)^2$
mini-batch: $\mathcal{B}_p$
memory: $\mathcal{D}_{\text{power}}$
$\left(s_n^{(t-1)},a_n^{(t-1)}\right)$ and old local observations
$\left(s^{(t-1)}_{n,a_n^{(t-1)}},a^{(t-1)}_{n,a_n^{(t-1)}}\right)$ and old local observations
distributed execution at transmitter n
backhaul delay of 1 slot
backhaul delay of T_d slots
update
backhaul delay of 1 slot
backhaul delay of T_d slots
update
$s_n^{(t)}=\left\{s_{n,1}^{(t)},\dots,s_{n,M}^{(t)}\right\}$
ψ_{agent}
$\max_a q\left(s_n^{(t)},a;\psi_{\text{agent}}\right)$
$a_n^{(t)}$
set $\alpha^{(t)}_{n,a_n^{(t)}}$ to 1
$\alpha_n^{(t)}$
$s^{(t)}_{n,a_n^{(t)}}$
θ_{agent}
$\mu\left(s^{(t)}_{n,a_n^{(t)}};\theta_{\text{agent}}\right)$
$a^{(t)}_{n,a_n^{(t)}}$
$p_n^{(t)}=P_{\max}a^{(t)}_{n,a_n^{(t)}}$
local observations and info from neighbors
wireless network

Fig. 1: Diagram of the proposed power control algorithm.

Similar to deep Q-learning, the critic network is trained by minimizing the mean-squared Bellman error defined in (8). However, compared to the deep Q-network, the critic network has only one output that gives a Q-function value estimate for a given state and action input. In addition, the target in (8) becomes $y_{\text{critic}}(r',s')=r'+\gamma q\left(s',\mu(s';\boldsymbol{\theta});\boldsymbol{\phi}_{\text{target}}\right)$.

Since $q(s,a;\boldsymbol{\phi})$ is differentiable with respect to action, caused by action space being continuous, the policy parameters are simply updated by the following gradient:

$$\nabla_{\boldsymbol{\theta}}\frac{1}{|\mathcal{B}|}\sum_{(s,\dots)\in\mathcal{B}}q\left(s,\mu(s;\boldsymbol{\theta});\boldsymbol{\phi}\right). \tag{11}$$

Note that a noise term is added to the deterministic policy output for exploration during training.

B. Local Information and Neighborhood Sets

We next describe the extent of the local information at transmitter n at the beginning of time slot t. In each time slot, transmitter n has two types of neighborhood sets for each subband. The first set is called "interferers" that consists of c indexes and is denoted as $\mathcal{I}^{(t)}_{n,m}$. For subband m, transmitter n first divides nearby transmitters into two groups whether they used subband m during time slot $t-1$ or not in order to prioritize the transmitters that occupy subband m. Then, it sorts each group according to the interfering channel strength at receiver n from their transmitters during time slot $t-1$ by descending order, i.e., $g^{(t-1)}_{i\rightarrow n,m}$. Lastly, the first c sorted nearby transmitters forms $\mathcal{I}^{(t)}_{n,m}$.

The second set is the set of "interfered receivers" that consists of c indexes and is defined as $\mathcal{O}_{n,m}^{(t)}$. Again, each nearby receiver j is first divided into two groups based on $\alpha_{j,m}^{(t-1)}$. The sorting criteria within each group becomes the potential significance of the interference strength at receiver j from transmitter n during time slot $t-1$, i.e., $g_{n\to j,m}^{(t-1)}\left(\sum_{l\in\mathcal{N},l\neq j}\alpha_{l,m}^{(t-1)}g_{l\to j,m}^{(t-1)}p_l^{(t-1)}+\sigma^2\right)^{-1}$.

Compared to [5], we follow simpler practical constraints on the available local information to be used in the state set design, as our main goal is to show the usefulness of the proposed approach. At the beginning of time slot t, transmitter n has access to the most recent local information gathered at receiver n for each subband m such as $g_{n\to n,m}^{(t)}$, $g_{i\to n,m}^{(t)}$ $\forall i\in\mathcal{I}_{n,m}^{(t)}$, and the sum interference power at receiver n which is measured just before the new policy decisions at the beginning of time slot t and can be formulated with subband and power allocation from time slot $t-1$ and interfering channel gains from time slot t as $\sum_{l\in\mathcal{N},l\neq n}\alpha_{l,m}^{(t-1)}g_{l\to n,m}^{(t)}p_l^{(t-1)}$. Conversely, the measurements gathered at nearby receivers are delayed by one time slot, e.g., $g_{n\to j,m}^{(t-1)}$ $\forall j\in\mathcal{O}_{n,m}^{(t)}$. Apart from the channel related measurements, we assume that each interfered and interferer neighbor also sends crucial key performance indicators delayed by one time slot due to network latency, e.g., its achieved spectral efficiency during last slot.

C. Proposed Multi-Agent Learning Scheme

In order to allow distributed execution, each link, specifically, each transmitter, operates as an independent learning agent by treating other agents as part of its local environment. Hence, our approach is based on multiple learning agents, rather than a single learning agent that controls the entire action space whose dimensions will grow exponentially with the total number of links. The single learning agent approach has similar drawbacks as the conventional centralized optimization algorithms in terms of complexity and cost of communication. In contrast, the proposed multi-agent approach is easily scalable to larger networks and can operate with just local information after training.

At the beginning of each time slot, each agent successively executes two policies to determine its associated subband and transmit power level. The reinforcement learning component at the top layer is a deep Q-network that is responsible for the subband selection. The bottom layer uses deep deterministic policy gradient algorithm to train the actor network responsible for agent's transmit power level decisions. As described in Fig. 1, the actor network at the bottom layer requires the subband decision of the top layer to determine its state input before setting agent's transmit power.

We next describe key components of the proposed design:

1) **Action set design:** All agents have the same pair of action spaces. The top layer uses a discrete action space that consists of subband indexes, i.e, $a_n^{(t)}\in\mathcal{A}_{\text{subband}}=\{1,\dots,M\}=\mathcal{M}$. Hence, we denote the subband selection of agent n for time slot t as $a_n^{(t)}$. The bottom layer has a continuous action space defined as $\mathcal{A}_{\text{power}}=[0,1]$. Since the bottom layer is executed after the top layer, we denote its action as $a_{n,a_n^{(t)}}^{(t)}$. We later multiply it by $P_{\max}$ to get $p_n^{(t)}=P_{\max}a_{n,a_n^{(t)}}^{(t)}$.

2) **State set design:** To be used in the state, all agents rank the subbands at the beginning of each time slot according to their direct channel gain to the total interference power ratio. We denote the rank as $z_{n,m}^{(t)}$. Now we describe the state of agent n on subband m at time t as:

$$\begin{aligned} s_{n,m}^{(t)}=\Bigg\{&\alpha_{n,m}^{(t-1)}p_n^{(t-1)},C_n^{(t-1)},z_{n,m}^{(t)},g_{n\to n,m}^{(t)},\\ &\sum_{l\neq n}\alpha_{l,m}^{(t-1)}g_{l\to n,m}^{(t)}p_l^{(t-1)},\Big\{g_{i\to n,m}^{(t)},\alpha_{i,m}^{(t-1)}p_i^{(t-1)},\\ &C_i^{(t-1)},z_{i,m}^{(t-1)}\Big|\forall i\in\mathcal{I}_{n,m}^{(t)}\Big\},\Big\{g_{n\to j,m}^{(t-1)},g_{j\to j,m}^{(t-1)},\\ &C_j^{(t-1)},z_{j,m}^{(t-1)},\sum_{l\neq j}\alpha_{l,m}^{(t-1)}g_{l\to j,m}^{(t-1)}p_l^{(t-1)}\Big|\forall j\in\mathcal{O}_{n,m}^{(t)}\Big\}\Bigg\}. \end{aligned} \tag{12}$$

Since the top layer does the subband decisions that requires information from all subbands, it should have a broader environment view than the bottom layer. Thus, for the top layer, we define agent n's state as $s_n^{(t)}=\left\{s_{n,1}^{(t)},\dots,s_{n,M}^{(t)}\right\}$. Then, the bottom layer uses $s_{n,a_n^{(t)}}^{(t)}$ as its input.

3) **Reward Function Design:** Both learning layers collaboratively aim to maximize the objective in (P1a). Consequently, they share the same reward function that describes the overall contribution of agent's combined subband and power decisions on the sum-rate objective. This includes agent's own spectral efficiency and a penalty term depending on its externalities to its interfered neighbors on subband $a_n^{(t)}$ [5]. For the reward function, we first compute the externality of agent n to interfered $j\in\mathcal{O}_{n,a_n^{(t)}}^{(t+1)}$ during time slot t as

$$\pi_{n\to j}^{(t)}=C_{j\setminus n,a_n^{(t)}}^{(t)}-C_{j,a_n^{(t)}}^{(t)}, \tag{13}$$

where $C_{j\setminus n,a_n^{(t)}}^{(t)}$ is the spectral efficiency of j without the interference from agent n on subband $a_n^{(t)}$ during slot t:

$$C_{j\setminus n,a_n^{(t)}}^{(t)}=\log\left(1+\frac{\alpha_{j,a_n^{(t)}}^{(t)}g_{j\to j,a_n^{(t)}}^{(t)}p_j^{(t)}}{\sum_{l\neq n,j}\alpha_{l,a_n^{(t)}}^{(t)}g_{l\to j,a_n^{(t)}}^{(t)}p_l^{(t)}+\sigma^2}\right). \tag{14}$$

Next, we define the reward of agent n as

$$r_n^{(t+1)}=C_{n,a_n^{(t)}}^{(t)}-\sum_{j\in\mathcal{O}_{n,a_n^{(t)}}^{(t+1)}}\pi_{n\to j}^{(t)}. \tag{15}$$

4) **Centralized Training:** Since multi-agent setting violates the environment stationary assumption of the underlying Markov decision process discussed in Section IV-A, there is an extensive research to develop multi-agent learning frameworks with good empirical performance, but rarely

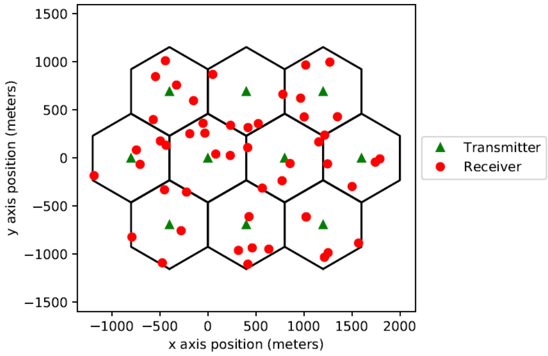


Fig. 2: A network configuration example.

with theoretical guarantees [15]. We follow some recently emerged multi-agent learning concepts like transfer learning and parameter sharing that increase the stability and convergence rate by taking advantage of the fact that agents are learning together [16]. Therefore, our method encourages stability by training global policy parameters shared across the network and trained by a centralized trainer that gathers experiences of all agents. As shown in Fig. 1, centralized training stores two experience-replay memories for each layer: $\mathcal{D}_{\text{subband}}$ and $\mathcal{D}_{\text{power}}$. At time t, the most recent experience at $\mathcal{D}_{\text{subband}}$ and $\mathcal{D}_{\text{power}}$ from agent n is $e^{(t-1)}_{n,\text{subband}} = \left(s^{(t-2)}_n, a^{(t-2)}_n, r^{(t-1)}_n, s^{(t-1)}_n\right)$ and $e^{(t-1)}_{n,\text{power}} = \left(s^{(t-2)}_{n,a^{(t-2)}_n}, a^{(t-2)}_{n,a^{(t-2)}_n}, r^{(t-1)}_n, s^{(t-1)}_{n,a^{(t-2)}_n}\right)$, respectively, due to the backhaul delay of 1 time slot. Note that the next state in $e^{(t-1)}_{n,\text{power}}$ is with respect to the old subband selection $a^{(t-2)}_n$.

During time slot t, the centralized training runs one gradient step for each policy. As described in Fig 1, it broadcasts most recent versions of ψ and θ once per T_u time slots. The broadcasting takes T_d time slots to finish, again due to the backhaul delay.

V. Simulation Results

In this section, we compare the performance of the proposed learning approach with some conventional optimization methods and joint learning as the number of subbands increases.

Throughout the simulations, we choose two network sizes of $(K, N) = (5 \text{ cells}, 20 \text{ links})$ and $(10 \text{ cells}, 50 \text{ links})$, respectively. As described in Fig. 2, we consider homogeneous hexagonal cells of 400 meters radius with each cell having equal number of uniformly randomly placed receivers. We vary the number of subbands M from 1 to 10. Following the LTE standard, we set the distance dependent path loss to $128.1 + 37.6 \log_{10}(d)$ (in dB), where d is transmitter-to-receiver distance in km. The log-normal shadowing standard deviation is 10 dB. We set $f_d = 10$ Hz, $T = 20$ ms, $P_{\text{max}} = 38$ dBm, and $\sigma^2 = -114$ dBm. Similar to [5], the signal-to-interference-plus-noise ratio is capped at 30 dB in the calculation of the spectral efficiency in (4) due to practical constraints on front end's dynamic range.

We compare the proposed approach with four benchmarks. The first is the joint learning approach as proposed in [2]. We discretize the transmit power into 10 levels. The second is called the 'ideal FP'. It runs the fractional programming algorithm with an assumption of full instant CSI. The first scenario ignores any delay during the execution of centralized optimization or passing the optimization outcomes to the transmitters. On the other hand, the third benchmark is called the 'delayed FP' and assumes one time slot delay to run the fractional programming algorithm. In the final benchmark, each transmitter just picks a random subband and transmit power at the beginning of every time slot.

We divide training into four episodes with each running for 5,000 time slots. At the beginning of each episode, we randomly sample a new deployment, and we reset the exploration and learning rate parameters. For faster convergence, we replace the noise term added to the deterministic policy output with Q-learning's ϵ-greedy algorithm. We have made the source code (including the implementation and hyperparameters) available at [17]. For better stability, we ensure that the bottom layer has higher learning rate than the top layer, and it uses a higher initial value of ϵ, but with a higher decay rate. The fine-tuning of the ϵ value is important to avoid converging to undesired situations in which all agents want to transmit with P_{max} or with zero power.

In Fig. 3, we show the training convergence of the proposed and joint reinforcement learning scheme. For $M = 2$ subbands, as shown in Fig. 3a, their convergence rates are quite close. However, when we increase the number of subbands, the joint learning approach is not able to keep up with the proposed approach in terms of training convergence. This is mainly caused by the increased size of the joint learning's action space and increased deep Q-network output layer complexity. Next, we test the performance of the trained policies on several randomly generated deployments in Table I. Testing shows that a pretrained policy is still usable on new deployments and the proposed approach is better scalable than the benchmarks.

VI. Conclusion and Future Work

We have demonstrated a novel multi-agent reinforcement learning framework for the joint subband selection and power control problem. With centralized training and distributed execution, only local information is needed by the agent under practical constraints. In addition, as the number of subbands increases, the proposed learning approach has better training convergence and higher sum-rate performance than the joint learning approach. For future work, we are looking into better and easily tunable training and exploration schemes to better adapt to the environment non-stationarity of the multi-agent setting.

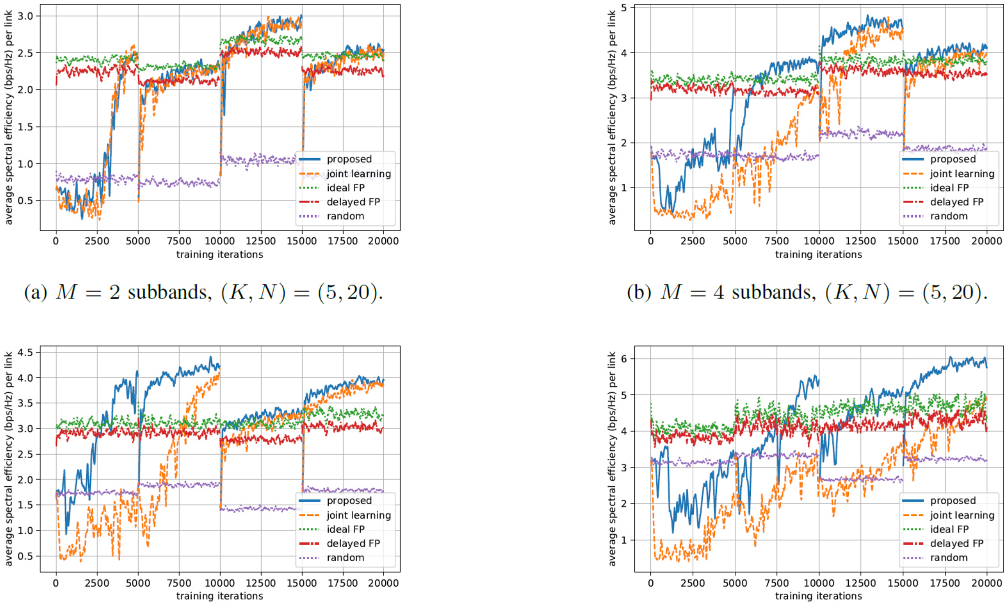
References

[1] K. Shen and W. Yu, "Fractional programming for communication systems—part i: Power control and beamforming," *IEEE Transactions on Signal Processing*, vol. 66, no. 10, pp. 2616–2630, May 2018.

[2] J. Tan, Y. C. Liang, L. Zhang, and G. Feng, "Deep reinforcement learning for joint channel selection and power control in D2D networks," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2020.

TABLE I: Testing results.

		average sum-rate performance in bps/Hz per link					output layer size		average
(K, N)	M	reinforcement learning		other schemes			reinforcement learning		iterations
(cells, links)	subbands	proposed	joint	ideal FP	delayed FP	random	proposed	joint	FP
$(5, 20)$	1	1.51	1.50	1.58	1.46	0.41	1 + 1	10	70.30
	2	2.63	2.64	2.66	2.46	0.99	2 + 1	20	102.08
	4	4.57	4.38	3.81	3.57	2.12	4 + 1	40	122.15
$(10, 50)$	1	1.26	1.26	1.31	1.21	0.25	1 + 1	10	72.83
	2	2.08	2.10	2.08	1.92	0.59	2 + 1	20	96.32
	4	3.34	3.34	2.90	2.68	1.31	4 + 1	40	185.93
	5	3.79	3.76	3.18	2.94	1.64	5 + 1	50	206.38
	10	5.71	4.41	4.44	4.08	2.99	10 + 1	100	287.70



(a) $M = 2$ subbands, $(K, N) = (5, 20)$.

(b) $M = 4$ subbands, $(K, N) = (5, 20)$.

(c) $M = 5$ subbands, $(K, N) = (10, 50)$.

(d) $M = 10$ subbands, $(K, N) = (10, 50)$.

Fig. 3: Training convergence.

[3] Z. Qin, H. Ye, G. Y. Li, and B. F. Juang, "Deep learning in physical layer communications," *IEEE Wireless Communications*, vol. 26, no. 2, pp. 93–99, 2019.

[4] E. Ghadimi, F. D. Calabrese, G. Peters, and P. Soldati, "A reinforcement learning approach to power control and rate adaptation in cellular networks," in *IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–7.

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

[6] F. Meng, P. Chen, and L. Wu, "Power allocation in multi-user cellular networks with deep Q learning approach," in *IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.

[7] F. Meng, P. Chen, L. Wu, and J. Cheng, "Power allocation in multi-user cellular networks: Deep reinforcement learning approaches," *IEEE Transactions on Wireless Communications*, vol. 19, no. 10, pp. 6255–6267, 2020.

[8] Y. S. Nasir and D. Guo, "Deep actor-critic learning for distributed power control in wireless mobile networks," in *the 54th Asilomar Conference on Signals, Systems, and Computers*, 2020, pp. 398–402.

[9] L. Liang, J. Kim, S. C. Jha, K. Sivanesan, and G. Y. Li, "Spectrum and power allocation for vehicular communications with delayed csi feedback," *IEEE Wireless Communications Letters*, vol. 6, no. 4, pp. 458–461, Aug 2017.

[10] Z. Q. Luo and S. Zhang, "Dynamic spectrum management: Complexity and duality," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 1, pp. 57–73, Feb 2008.

[11] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. Cambridge, MA, USA: MIT press, 2018.

[12] J. Achiam, "Spinning up in deep reinforcement learning," https://spinningup.openai.com, 2018.

[13] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[14] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv e-prints*, p. arXiv:1509.02971, Sep. 2015.

[15] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi, "Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications," *IEEE Transactions on Cybernetics*, pp. 1–14, 2020.

[16] J. K. Gupta, M. Egorov, and M. Kochenderfer, "Cooperative multi-agent control using deep reinforcement learning," in *International Conference on Autonomous Agents and Multiagent Systems*. Springer, 2017, pp. 66–83.

[17] Y. S. Nasir and D. Guo, "TensorFlow code for deep reinforcement learning for joint spectrum and power allocation in cellular networks," https://github.com/sinannasir/Spectrum-Power-Allocation, 2020.